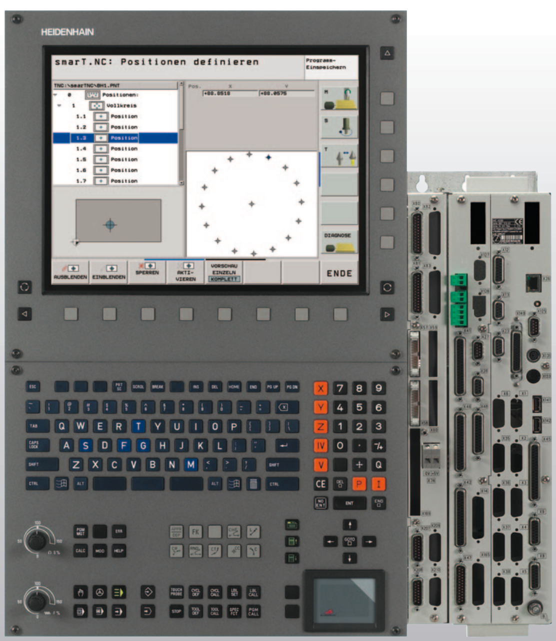
HEIDENHAIN

Technical Manual



iTNC 530

NC Software
340 490-06
340 491-06
340 492-06
340 493-06

February 2011

Content

1 Update Information No. 26 – Revised Version

1 Update Information No. 27

1 Update Information No. 28

1 Update Information No. 29

1 Update Information No. 30

2 Introduction

3 Mounting and Electrical Installation

4 Machine Parameters

5 Modules, Markers and Words

6 Configuring the Axes and Spindle

7 CC 424(B) Controller Unit

8 Machine Interfacing

9 PLC Programming

10 Data Interfaces

11 iTNC 530 with Windows 2000/Windows XP

12 Error Messages

13 Subject Index

1 Update Information

1.1General Information

Update Information for the iTNC 530 appears at irregular intervals, often as part of a new software version. This is preliminary information in PDF format, containing brief descriptions of new software functions as well as new hardware components. After the Update Information has been published, the new items are included in the iTNC 530 Technical Manual.

The Technical Manual and each Update Information are saved in the HEIDENHAIN FileBase on the Internet, where registered users can access them at http://portal.heidenhain.de.

Registered users of the HEIDENHAIN FileBase on the Internet receive an e-mail notification when a new Update Information appears.

This version of the Technical Manual includes all 24 Update Information notifications that have appeared so far, meaning that the contents of this Technical Manual for the iTNC530 correspond to the scope of functions of software version 34049x06.

1 Update Information No. 25 – Introduction of HSCI

1.1 Overview

1.1.1 Released service packs

The following service packs were released for **340 49x-05**:

- Service pack 2: January 2009
- Service pack 2: March 2009
- Service pack 3: June 2009
- Service pack 4: July 2009
- Service pack 5: November 2009
- Service pack 6: April 2010

1.1.2 Released NC software

The following NC software was released for the iTNC 530 with HSCI and MC 6xxx/CC 6xxx:

- NC software 606 420-01 and 606 421-01 July 2010
- NC software 606 422-01 and 606 423-01 July 2010

1.2 iTNC 530 with HSCI

1.2.1 Important notes

- **HSCI support in software version 340 49x-05**
  HSCI systems operating with NC software 340 49x-05 during the introductory phase must migrate to the new NC software 606 42x-01. The iTNC 530 control with HSCI must be operated with NC software 606 42x-01.
- **NC software 340 49x-06 and NC software 606 42x-01**
  The new NC software 606 42x-01 of the iTNC 530 with HSCI will be introduced for the new MC 6xxx/CC 6xxx hardware at around the same time as the NC software 340 49x-06 of the iTNC 530 for the MC 42x/CC 42x hardware.
  The separately issued Update Information No. 24 describes all of the new software functions of the two NC software versions to be introduced. If there are differences between the two versions, these will be pointed out. New features that will be introduced only for NC software 606 42x-01 because of its greater processing power are described in this Update Information No. 25 about the introduction of HSCI.
- **Documentation of NC software 606 42x-01**
  For the documentation of the new iTNC 530 hardware generation for HSCI, please refer to this Update Information for the time being. The same applies to the special features of NC software 606 42x and its differences from NC software 340 49x. All other functions are almost identical to those of NC software 340 49x-06 and are described in the present Technical Manual for the iTNC 530. The new hardware MC 6xxx with CC 6xxx has the same features as the previous hardware MC 422C with CC 424B. HEIDENHAIN will inform you when a separate manual is available for the iTNC 530 with HSCI, NC software 606 42x-01.
- **Introduction of HSCI for the iTNC 530**
  The new HEIDENHAIN control design with HSCI will be introduced with software 606 42x-01 for the iTNC 530. In this design, communication between the control components takes place via the HSCI digital interface. HSCI stands for HEIDENHAIN Serial Controller Interface and is based on the 100BaseT Ethernet standard so well known from network connections. A special interface component and real-time protocol, both developed by HEIDENHAIN, make short cycle times for data transfer possible.
  Today’s MC 4xx main computers of the iTNC 530 are always mechanically connected to the CC 4xx controller unit, and are installed in the electrical cabinet. The new MC 6xxx main computers and CC 61xx controller units with HSCI interface can be installed in separate locations. A specialized version of the MC can be incorporated directly into the operating panel. The TFT flat-panel display and the main computer are then one unit, meaning that the problem of excessive cable lengths is no longer an issue on large machines. Main advantages of the control design with HSCI:
  - High noise immunity due to digital communication between components
  - Hardware basis for integrated Functional Safety (FS)
  - Simpler wiring
  - Connection of EnDat 2.2 encoders
  - Inverters connected as before via well-proven PWM interface
  - Hardware platform for flexible and scalable control system
  - Greater cable lengths of the complete system
  - Number of control loops can be increased (up to 18 axes and 1 spindle)
  - More PLC inputs and outputs
  - Controller units can be installed elsewhere

- **New HeROS 5 operating system**
  With the introduction of the new NC software 606 42x-01, the operating system will also be migrated to the new HeROS 5 (HEIDENHAIN Real Time Operating System) version. The migration will make it possible to integrate even more powerful functions in the control software.
  First of all, HeROS 5 will also include a PDF viewer, which will make it possible to open PDF files right on the control. Furthermore, many operating system dialogs (e.g. time setting or network settings) will be displayed in a standardized format in the future.
- **Introduction of Functional Safety (FS) for the iTNC 530**
  The new HSCI hardware also makes it possible to set up HEIDENHAIN control systems with integrated Functional Safety (FS). However, the full range of features will not yet be available in the first software version for the iTNC 530. Your contact person at HEIDENHAIN will be glad to answer any questions concerning the iTNC 530 with Functional Safety. This documentation describes the HSCI hardware for systems without Functional Safety.
  The special features of the hardware and software for Functional Safety are described in the Technical Manual for the iTNC 530 with Functional Safety (FS), which will be available soon.

> Note
>
> The NC software 606 42x is not yet approved for applications that use the integrated Functional Safety (FS) of the control! A separate approval is required for the use of integrated Functional Safety (FS) according to EN ISO 13849-1!

- **Software option #46 (Python OEM Process)**
  The new and enhanced JH library version 3.0 of the HEIDENHAIN functions for Python will be introduced with software version 606 42x-01 and the HeROS 5 operating system, respectively. Also, Python version 2.3.4 will no longer be available and will be replaced by Python version 2.6. Due to these changes the Python scripts that were created for the JH library 2.0 running under HeROS 4 cannot run with library 3.0. You will find more detailed information in the "PLC" section on page 33 ff.
- **Updating the HSCI configuration**
  HEIDENHAIN recommends updating all IOC files and EAZ projects with the IOconfig version 2.2.02. This is the only way to make sure that you can use all of the features provided by the new HSCI components, see page 134.

Machine parameters

- **Only CC 61xx: MP108.x – Assignment of the axes to the drive-control motherboards**
  With MP108.x, you assign the axes to the individual drive-control motherboards of the CC 61xx in the HSCI system. The HSCI address of the respective drive-control motherboard is entered in MP108.x. The HSCI address is obtained from the position of the drive-control motherboard in the HSCI system. However, the HSCI address to be entered only depends on the drive-control motherboards in the system. I/O units (PLs) and machine operating panels (MBs) are not taken into account. This means that for the first drive-control motherboard, you have to enter the address 0 in MP108, regardless of whether I/O units or machine operating panels are located before the CC in the HSCI chain.
  MP120.x is used for further assignment of the axes to the outputs of the respective drive-control motherboard.
  Input: 0 to 3
- **Only CC 61xx: MP109.x – Assignment of the spindles to the drive control motherboards**
  With MP109.x, you assign the spindles to the individual drive-control motherboards of the CC 61xx in the HSCI system. The HSCI address of the respective drive-control motherboard is entered in MP109.x. The HSCI address is obtained from the position of the drive-control motherboard in the HSCI system. However, the HSCI address to be entered only depends on the drive-control motherboards in the system. I/O units (PLs) and machine operating panels (MBs) are not taken into account. This means that for the first drive-control motherboard, you have to enter the address 0 in MP108, regardless of whether I/O units or machine operating panels are located before the CC in the HSCI chain.
  MP121.x is used for further assignment of the spindles to the outputs of the respective drive-control motherboard.
  Input: 0 to 3
- **Removed as of CC 61xx: MP115, MP116**
  If a CC 61xx or a later version is used, then MP115, MP116 will be replaced by MP118 and MP119. MP118 (for axes) and MP119 (for spindles) are used to configure the position encoder inputs for specific axes.
- **Only CC 61xx: MP118.x – Position encoder input for axes**
  MP118.x is used as a replacement for MP116. In MP118.x, the position encoder input for the axes is configured for specific axes.
  Input: %xxxx
  - Bit 0: Position encoder input 1 V_{PP}
    0: 1 V_{PP}
    1: Reserved
  - Bit 1: Reserved
  - Bit 2: Input frequency of the position encoder input
    With 1 V_{PP}:
    0: 27 kHz
    1: 400 kHz
  - Bit 3: Analog or digital position encoder input
    0: Analog encoder signal control (1 V_{PP})
    1: Digital encoder signal control (EnDat 2.2)

- **Only CC 61xx: MP119.x – Position encoder input for spindles**
  MP119.x is used as a replacement for MP116. In MP119.x, the position encoder input for the spindles is configured for specific axes.
  Input: %xxxx
  - Bit 0: Position encoder input 1 V_{PP}
    0: 1 V_{PP}
    1: Reserved
  - Bit 1: Reserved
  - Bit 2: Input frequency of the position encoder input
    With 1 V_{PP}:
    0: 27 kHz
    1: 400 kHz
  - Bit 3: Analog or digital position encoder input
    0: Analog encoder signal control (1 V_{PP})
    1: Digital encoder signal control (EnDat 2.2)
- **Expanded for CC 61xx: MP120.x – Nominal speed value outputs of axes**
  With the new CC 61xx controllers, the expanded MP120.x is used to assign the outputs to the axes. Entering X51 to X56, X80 to X85 in MP120.x also provides all further connector assignments to the axis, because the assignment between speed encoder, PWM output (power module output on the UEC) and position encoder is a permanent assignment:
  - When using a CC 61xx in conjunction with HEIDENHAIN inverters

Input of MP120.x/MP121.x: PWM output	**51**	**52**	**53**	**54**	**55**	**56**
Input: Speed encoder	X15	X16	X17	X18	X19	X20
Input: Position encoder (input in MP110.x/MP111.x)	X201	X202	X203	X204	X205	X206

  - When using a UEC

Input in MP120.x: Speed output: PWM output	**80**	**81**	**82**	**83**	**84**	**85**
Input: Speed encoder	X15	X16	X17	X18	X19	X20
Input: Position encoder	X201	X202	X203	X204	X205	X206

Expanded for CC 61xx: MP121.x – Nominal speed value outputs of spindles

With the new CC 61xx controllers, the expanded MP121.x is used to assign the outputs to the spindles. Entering X51 to X56, X80 to X85 in MP121.x also provides all further connector assignments to the spindle, because the assignment between speed encoder, PWM output (power module output on the UEC) and position encoder is a permanent assignment:

- When using a CC 61xx

Input of MP120.x/MP121.x: PWM output	**51**	**52**	**53**	**54**	**55**	**56**
Input: Speed encoder	X15	X16	X17	X18	X19	X20
Input: Position encoder (input in MP110.x/MP111.x)	X201	X202	X203	X204	X205	X206

- When using a UEC

Input in MP121.x: Speed output: PWM output	**X80**	**X81**	**X82**	**X83**	**X84**	**X85**
Input: Speed encoder	X15	X16	X17	X18	X19	X20
Input: Position encoder	X201	X202	X203	X204	X205	X206

- **Removed for CC 61xx: MP2050 – Drive enabling function I32**
  MP2050 has no function in the CC 61xx. In the HSCI system, input I32 is replaced by input -ES.B (Emergency Stop input 2), which, however, behaves different from I32. The input -ES.B behaves in the same way as input -ES.A and is used to evaluate the emergency stop chain.
- **Removed for CC 61xx: MP2150 – Signal for power fail**
  MP2150 has no function in the CC 61xx. The evaluation of the two signals -PF.PS.AC and -PF.PS.DC is activated by MP2195 bit#7 and bit#8, respectively.
- **Removed for CC 61xx: MP2192, MP2194 – LIFTOFF configuration**
  The CC 424(B) controller board was used to define voltage thresholds in MP2192 and MP2194, which were used to configure the LIFTOFF function. These voltage thresholds are now defined by the HSCI CC 61xx controller unit itself, depending on the power supply module being used. The associated machine parameters MP2192 and MP2194 are no longer required. If you do not need the LIFTOFF function, you can use bit#9 in MP2195 to deactivate monitoring of the DC-link voltage.
  The CC 61xx makes the following distinction: It takes into account whether the power supply module is:
  - a regenerative power supply module with servo-controlled DC-link voltage, or
  - a nonregenerative power supply module where, due to changes in the line voltage and load on the DC link, voltage fluctuations must be considered.
- **New for CC 61xx: MP2195 bit#9 – Monitoring the DC-link voltage Uz or LIFTOFF function**
  With the CC 61xx, the LIFTOFF function is no longer configured in MP2192 and MP2194, but it is automatically activated with the monitoring of the DC-link voltage Uz. If you do not need the LIFTOFF function, you can use bit#9 in MP2195 to deactivate monitoring of the DC-link voltage.
  Input: Bit#9 = 0: Uz monitoring and LIFTOFF function are active
  Bit#9 = 1: Uz monitoring and LIFTOFF function are not active
- **New for CC 61xx: MP2196 – Designator for power supply module**
  A designator used to identify the supply modules (MP2198.x) in diagnostic or error messages can be entered in MP2196.
  Input: P,Q, R or T
- **New for CC 61xx: MP2000.x – Performance of control loop**
  With MP2000.x, you can switch control loops for specific axes from single-speed to double-speed for higher controller performance (software option #49). Machine parameter MP7610.x is therefore no longer required for the CC 6xxx .
  Input: 0: Single-speed axis
  1: Double-speed axis
- **Enhanced for the CC 61xx: MP2206.x – Type of encoder**
  Input: 9: Aligned rotary encoder with EnDat 2.2 interface
  10: Nonaligned rotary encoder with EnDat 2.2 interface
  11: Linear encoder with EnDat 2.2 interface
  12: Reserved

- **New for CC 61xx: MP2221.x bit#10 – Handling of linear and synchronous motors**
  With the CC 61xx, a new way of handling linear and torque motors by the controller unit was introduced. Together with AFC, this makes it possible to attain higher milling power. However, if you want to handle the motors in such a way as the CC 424(B) or CC 422 does, this can be configured in MP2221 bit#10.
  Input: 0: New way of handling
  1: Handling in the same way as the CC 424(B) or CC 422
- **Removed for CC 61xx: MP2234.x – Internal triggering of the motor brakes via the PWM interface**
  MP2234.x has no function in the CC 61xx. If the value entered in MP2230.x is not equal to 0, the brake is automatically triggered and the motor brake test is performed. However, a system with Functional Safety (FS) is required for the test to be performed.
- **New for CC 61xx: MP2440.x – Cutoff frequency of feedforward current controller in Hz**
  For a description of MP2440, see page 102.
  Input: 0 Hz to 5000.0 Hz
  0: Feedforward is deactivated (default setting)
- **New for CC 61xx: MP2561.x bit#0 – Maximum input frequency for motor encoders**
  Switching the maximum input frequency for motor encoders to 800 kHz may be necessary for high-speed motors or spindles, depending on the resolution of the motor encoder. The precondition is that the encoder be suitable for this frequency range and still be able to provide a sufficient signal amplitude at 800 kHz.
  Format: %xxxxxxxxxxxxxxxx
  Input: Bit 0 – Maximum input frequency of motor encoder
  0: 400 kHz
  1: 800 kHz
- **Removed for CC 61xx: MP2607.x, MP2608.x**
  These machine parameters have no function in the CC 61xx.
- **New for CC 61xx: MP2912.x – Settings for master-slave torque control**
  The new machine parameter MP2912.x simplifies configuration of the torque-adjustment controller output (=speed compensation value). There are only two input options for the new MP:
  - MP2912.x = 0
    The output of the torque-adjustment controller (= speed compensation value) is distributed evenly to master and slave.
  - MP2912.x = 1
    The output of the torque-adjustment controller (= speed compensation value) is only distributed to the slave.

Previous settings:	With CC 61xx:
MP2930.x = 0	MP2912.x = 0
MP2930.x = 100	MP2912.x = 1

- **Removed for CC 61xx: MP2920.x, MP2930.x – Settings for master-slave torque control**
  The two machine parameters MP2920.x (Factor for variable torque distribution) and MP 2930.x (Speed compensation ratio for master-slave torque control) are no longer required for the CC 61xx. Free distribution between –100% and +100%, which was possible with MP2930 that is no longer valid, has actually not been used. Instead, only the two special cases MP2930 = 0 or MP2930 = 100 occurred. As a result, the following improvement was made: The previous configuration will be replaced by the new machine parameter MP2912.x.
- **New for CC 61xx: MP2450.x – Dead-time compensation**
  MP 2450 is used to deactivate (0) or activate (1) the dead-time compensation. For a description of MP2450, see page 102.
  Input: 0 or 1
- **New for HSCI: MP4020 bit#14 – Configuration of PLC module interface**
  Up to 18 NC axes are available in an HSCI system. The input option in the token AXISNUMBER of the OEM.SYS has therefore been increased from 14 to 18. Up to now, the PLC interface also supported only 14 axes. To be able to use the increased number of axes, you must define bit#14 of machine parameter MP4020.
  Input:
  - MP4020 bit#14 = 0
    PLC module interface for 14 axes plus spindle
    (axes = bit 0 to bit 14, spindle = bit 15).
  - MP4020 bit#14 = 1
    PLC module interface for 18 axes plus spindle
    (axes = bit 0 to bit 17, spindle = bit 31).

Note

Please keep in mind that the parameters of PLC modules with bit-encoded axis masks or constants for axes and spindles are set and interpreted in different ways, depending on MP4020 bit#14.

- **New for HSCI: MP4044 – Set PLC output after shutdown**
  In an HSCI system, machine parameter MP4042 is replaced by MP4044. In MP4044, you can specify a PLC output of an HSCI system, which will be set after shutdown. This makes it possible to configure the shutdown of the machine. The shutdown function is activated as before via MP4040 and MP4041. However, in an HSCI system, any PLC output can be used for this function.
  Input: Symbolic name of PLC output
  0: Cannot be evaluated, results in an error message
- **Removed for HSCI: MP4060.x, MP4061.x – Delayed switch-off of outputs**
  In a system with HSCI, these two machine parameters are replaced by settings in the IOconfig software tool for PCs. The values entered in IOconfig for the delay time of outputs replace MP4060.x and MP4061.x.

- **New for CC 61xx: MP4132 – Axis-specific drive enabling**
  A CC 61xx does not have an X150 connector for axis-specific or axis-group-specific drive enabling. In MP 4132.0 to MP4132.7, you can enter up to eight numbers of PLC inputs or symbolic PLC operands that simulate the inputs of X150. The axes are assigned to the individual axis groups in MP2040.0 to MP2040.7.
  Input: 0 to 20000
  [Number of the PLC input or symbolic PLC operand]
  -1: Function inactive
- **Enhanced for HSCI: MP7230.0 – NC conversational language**
  In controls with HSCI, the language of operating-system texts is also configured in Machine Parameter MP7230.x. The operating-system settings defined in the Language menu item in the JH menu are overwritten by the machine-parameter settings after the control has been restarted.
- **Removed for HSCI: MP7392.x – Settings for screen saver**
  In controls with HSCI, the screen saver is configured using the XFCE window manager.
- **MP7645 index 3 – Reserved for HR 550**
  Reserved for the introduction of the HR 550FS wireless handwheel
- **MP7620 bit#9 – HSCI keyboard with three potentiometers**
  If you are using an HSCI keyboard with three potentiometers, MP7620 bit#9 must be set. This activates the three potentiometers for feed-rate, spindle-speed and rapid-traverse override.
  Input: 0 = Keyboard with two potentiometers
  1 = HSCI keyboard with three potentiometers
- **Removed for HSCI: MP7691 – TRACE settings**
  In controls with HSCI, the TRACE settings are replaced by the HE Logging settings. These are settings for internal diagnostic files (FILO memory) for finding errors. These files can only be evaluated by HEIDENHAIN. The machine parameters were replaced by the following settings:
  - MP7691.0 (OS-Trace) by **`FTrace`**
  - MP7691.1 (TCP-Trace) by **`TCPDump`**
  - MP7691.2 (NC-Trace), MP7691.3 (Kernel-Trace) by **`Syslog`**

Expanded PLC module interface for 18 axes

Note

Please keep in mind that the parameters of PLC modules with bit-encoded axis masks or constants for axes and spindles are set and interpreted in different ways, depending on MP4020 bit#14.

If the expanded PLC module interface is active, the spindle bit moves to bit#31 and the axis number (constant) for the spindle to K 31 for the following PLC modules:

PLC module	Function	If MP4020 bit#14 = 0	If MP4020 bit#14 = 1
9038	Read general axis information	K 15	K 31
9065	Status of the commissioning function	Bit#15	Bit#31
9158	Maximum torque	K 15	K 31
9128	Torque limiting by the PLC	K15	K31
9129	Status of torque limiting by the PLC	K15	K31
9149	Set field angle/Read via PLC	K15	K31
9157	Drive controller status	Bit#15	Bit#31
9159	Advance status report: Drives will be switched off	Bit#15	Bit#31
9161	Enable the drive controller	Bit#15	Bit#31
9162	Status request of the drive controller	Bit#15	Bit#31
9163	Switching the operating modes	K15	K31
9164	Read the actual speed value of the motor	K15	K31
9165	Sample the current motor temperature	K15	K31
9166	Momentary utilization of the drive motor	K15	K31
9168	Interrogate the commissioning status	Bit#15	Bit#31
9169	Axes for which I32 does not switch off the drives	Bit#15	Bit#31
9170	Find the current torque	K15	K31
9220	Renewed traversing of the reference marks	K15	K31
9314	Limitation of drive power	Bit#15	Bit#31

Some PLC words are also affected by the conversion to the expanded PLC module interface. For the PLC words concerned (e.g. W1024), the expansion to 32 bits makes a conversion to double words (e.g. W1100) necessary so that all pieces of information can be included. The previous words continue to be valid, but they do not contain all of the 32 bits. The previous words will therefore be copied to a double word with 32 bits.

16-bit word W	**32-bit double word D**	**Description**
1024	1100	Axis enabling
1026	1104	Axes in position
1028	1108	Axes in motion
1030	1112	Current direction of traverse
1032	1116	Reference marks not yet traversed
1034	1120	Positive software limit switch was approached
1036	1124	Negative software limit switch was approached
1038	1128	Prepare to open the position control loop
1040	1132	Axis-specific opening of the position control loop
1042	1136	Deactivate monitoring functions
1044	1140	Actual-to-nominal value transfer
1046	1144	Manual traverse in positive direction
1048	1148	Manual traverse in negative direction
1050	1152	Incremental jog positioning in positive direction
1052	1156	Incremental jog positioning in negative direction
1054	1160	Reference end position
1056	1164	Lubrication pulse: Value in MP4050.x exceeded
1058	1168	Reset the accumulated distance (lubrication)
1060	1172	Axis-specific feed-rate enable
1062	1176	Lock the handwheel for specific axes

The PLC modules 9160 (status request for temperature monitoring and I2t monitoring) and 9168 (interrogate commissioning status) are further special cases. If the expanded PLC module interface is active, the two PLC modules generate an error message and can no longer be used. Please use PLC module 9066 (status of HEIDENHAIN power supply unit) instead of PLC module 9160, and PLC module 9065 (status of commissioning function) instead of PLC module 9168. With software version 340 49x-05, the corresponding functions were added to PLC modules 9066 and 9065.

For PLC modules 9040/9041 (reading the axis coordinates), a sufficient number of double words—matching the maximum possible number of machine axes (entry AXISNUMBER in the OEM.SYS)—must be reserved for saving the axis coordinates. If you do not reserve sufficient memory area for the axis coordinates, the following memory area might be overwritten by modules 9040 and 9041.
As the expanded PLC module interface also requires a maximum area of 32 double words, you can simply reserve the memory area for 32 double words as an alternative to the entry AXISNUMBER in order to prevent data from being lost by overwriting the memory area.

- **M4641 – HSCI status marker**
  The PLC marker M4641 shows the current status of the HSCI connection. If the marker is set, the HSCI bus operates without error.
- **M4260 – "Control-is-ready" acknowledgment (CR 34980)**
  The marker M4260 is set when an EMERGENCY STOP is triggered (the signals ES.A, ES.B, ES.A.HW, ES.B.HW become active). The status of the marker is ambiguous until the self-test of the control, including the EMERGENCY STOP test, has been completed, because EMERGENCY STOP is also checked during the tests.
- **Software option #46 (Python OEM Process)**
  Scripts that were created with JH library 2.0 must be adapted to the requirements of the new interface 3.0. Please note the following important points:
  - Notification command for monitoring the complete table for changes:
    `jh.Get(handle)` (instead of "**#**")

    The function always returns exactly one data block.
    If a subscribe returns more than one data block, this function can be used to iterate over the data blocks.
    The first call returns the first data block, and each further call returns the next data block of the interrogation.
    The virtual iterator is reset every time before the notify callback is called.
  - Access to tables has been changed:
    The callback function of a **`jh.Subscribe(notify = self.Callback)`** will transfer the type of change in the table.
    The new parameter **`event`** must be defined for every callback function.
    e.g.:
    ```
    def Callback(dict, event , param=None)
    print dict
    import jh handle =
    jh.Subscribe(ident=¿\\PLC\\program\\symbol\\glo-
    bal\\M_global_marker_1¿, notify=Callback, onChange=True)
    ```
  - Renamed function for Python-Python communication jh.gtk.Socket:
    `jh.gtk.Socket` was changed to **`jh.gtk.TcpComm`**
    `SocketError` was changed to **`LoLevelError`**
    `jh.gtk.Socket.Server` was changed to **`jh.gtk.TcpComm.TcpCommServer`**
    `jh.gtk.Socket.Client` was changed to **`jh.gtk.TcpComm.TcpCommClient`**
  - Functions that have been removed from the Python-Python communication:
    The two functions **`jh.gtk.Socket.Client.Receive`** and **`jh.gtk.Socket.Server.Receive`** were removed from the JH library 3.0, because using them has frequently caused problems in the program run.

Module 9132 Selection of override potentiometer

In a control system with more than one HSCI machine operating panel or keyboard unit, you can use Module 9132 to define which potentiometers are to be active. The selected operating panel or keyboard unit uses the potentiometer values to calculate the override.

If you use more than one operating station or machine operating panel, the PLC program must ensure that only one of the two operating devices is active at any one time so as to avoid danger to the operator.

Condition:

- The override potentiometers are selected by selecting the machine operating panel to which the potentiometers are connected.
- The numbering of the machine operating panels starts with the number 0 and is based on the arrangement or sequence in the IOC configuration or the HSCI string. A machine operating panel or PLB 6001 at connector X501 of the MC always receives the last, i.e. the highest number of the machine operating panels in the HSCI system.
- After the control has been restarted and the PLC has been restarted/ compiled, the potentiometers on the machine operating panel with the number 0 are active. If you want the previously active operating panel to be active again, the PLC program must write the corresponding information to the nonvolatile memory and evaluate it during a restart of the PLC.
- Basically, there is no switchover of operation between the machine operating panels or the keyboard units. Only the evaluation of the potentiometers is switched.

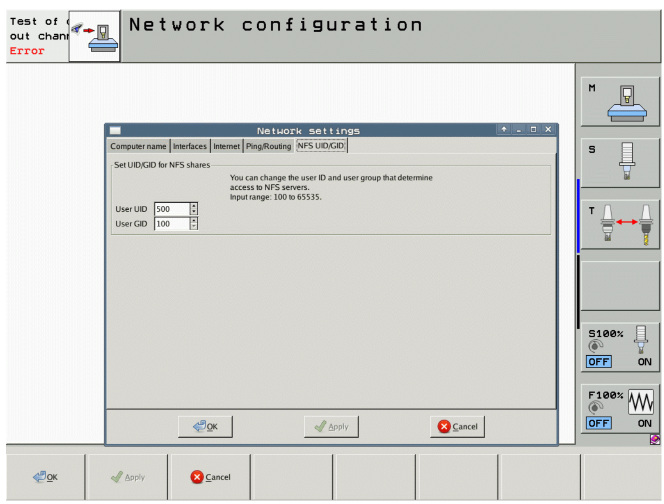
Call:

```
PS     K/B/W/D   <Number of machine operating panel>
                 0 ... n
PS     K/B/W/D   <Reserved>
PS     K/B/W/D   <Reserved>
CM     9132
```

Error recognition:

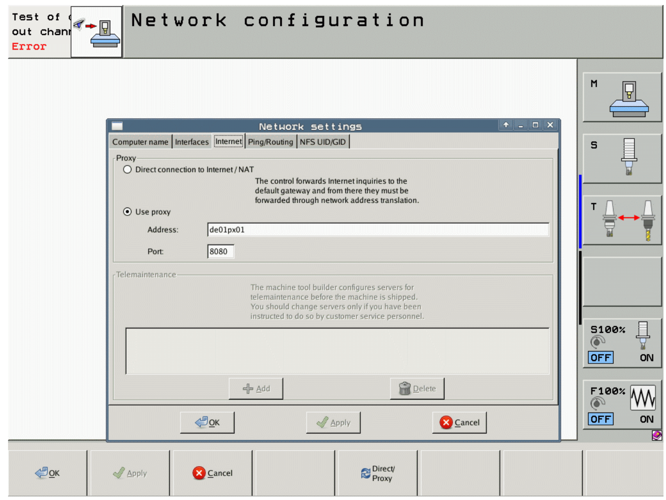
Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Programmed MB machine operating panel does not exist

- **Calculation of intersections for Dynamic Collision Monitoring (DCM)**
  With the new MC 62xx main computers, up to 200 intersection calculations are possible for collision monitoring. The collision objects themselves and the number of collision objects must be defined in such a way that no more than 200 intersection calculations are required. Otherwise the increased calculation efforts can cause increased block processing times. The intersection calculations apply to the complete collision monitoring model, including the monitoring of tool-carrier kinematics and fixtures. You therefore need to save a sufficient number of intersection calculations when describing the actual machine so that the max. permissible number of intersection calculations is not exceeded if you also want to monitor tool-carrier kinematics and fixtures.
- **Only CC 61xx: New entries in the motor table (motor.mot)**
  With the CC 61xx, the following entries in "Encoder in use (SYS)" have been introduced for encoders with EnDat 2.2 interface:
  - 9 = Rotary encoder with EnDat 2.2 interface (aligned)
  - 10 = Rotary encoder with EnDat 2.2 interface (nonaligned)
  - 11 = Linear encoder with EnDat 2.2 interface
- **Network settings UID/GID**
  With HeROS 5, too, the configuration possibilities for UID and GID will be available on a tab in the Network settings window after you have entered the code number NET123:



Tab	Option	Meaning
`NFS UID/ GID`	Entry of user ID and user group for access to the NFS server	
	`User ID:`	Definition of which user identification the control uses to access files in the network. Your network specialist determines this value.
	`Group ID:`	Definition of which group identification the control uses to access files in the network. Your network specialist determines this value.

- **Network settings for Internet access**
  With HeROS 5, the control can access the Internet. After entering the code number NET123, you will find the settings for Internet access on a tab in the Network settings window. These settings are used for remote maintenance and are used as default settings by the control's web browser (Firefox). However, you can also use your own program-specific settings to configure the web browser.



Tab	Option	Meaning
`Internet`	Settings for Internet access (proxy) and remote maintenance:	
	`Proxy`	■ **`Direct connection to Internet / NAT`** The control forwards Internet inquiries to the default gateway and from there they must be forwarded through network address translation (e.g. if a direct connection to a modem is available) ■ **`Use proxy`** These settings are made by your network specialist. • **`Address:`** Enter the address of the Internet router in the network • **`Port:`** Enter the port number
	`Telemaintenance`	Function reserved for future expansion

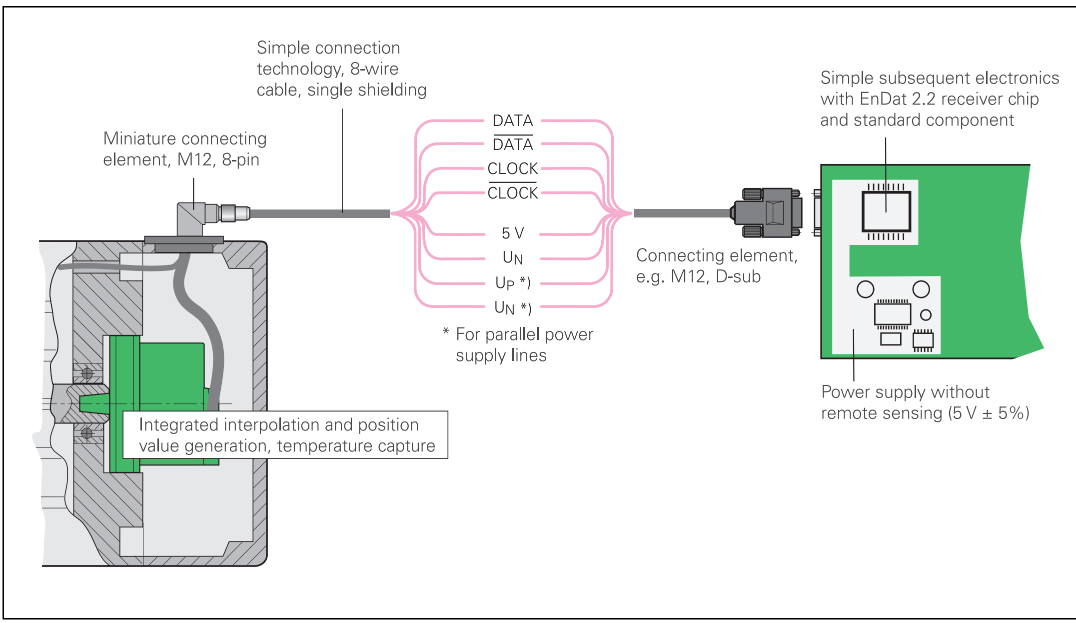
1.3 HSCI

1.3.1 Introduction

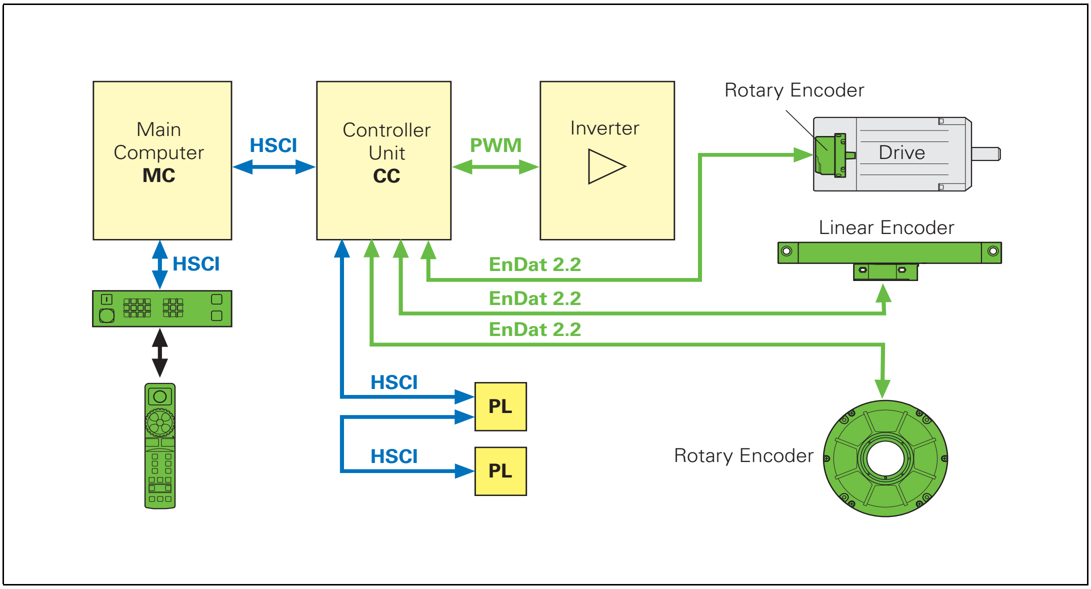
The MC is connected to the CC controller unit and the PLB 6xxx PLC basic modules in the electrical cabinet via **HSCI** (HEIDENHAIN Serial Controller Interface). The MB machine operating panel is also connected via HSCI. The connection of the various control components via HSCI offers numerous benefits, including:

- Simple and uncomplicated wiring
- High noise immunity
- Simple commissioning
- Comprehensive yet straightforward possibilities for diagnostics

The iTNC 530 is prepared for the connection of incremental and absolute position and speed encoders. EnDat 2.2, which is purely digital and compatible to version 2.1, makes it possible to very rapidly transmit highly resolved position values over long cable lengths. An overview of EnDat 2.2:



Overview of the purely digital control architecture with HSCI and EnDat 2.2:



The **iTNC 530** is designed for connection of a compact inverter or the modular inverter system. A complete control package, including drives and HEIDENHAIN motors, can be delivered (see the "Inverter Systems and Motors" Technical Manual).

1.3.2 HSCI interface

The individual control components communicate with each other via the HSCI connection (HEIDENHAIN Serial Controller Interface). A connection via HSCI is only permitted for HEIDENHAIN components that are part of the machine tool's control system. In addition, the HSCI connecting cable may only be installed in a protected manner (e.g. within the electrical cabinet, cable ducts).

The following features characterize the HSCI connection:

- Based on standard 100BaseT Ethernet hardware
- Telegrams of the HSCI connection are not compatible with the Ethernet
- Line structure
- Only one master in the system (MC), all other devices are HSCI slaves

Different addresses are assigned to the individual participants in the HSCI network. The addresses are assigned dynamically during booting of the MC. The HSCI addresses of the participants are formed from an HSCI address (8 bits) and a device type address (6 bits).
After the machine has undergone acceptance testing, the nominal configuration of the control is saved in the IOC file on the control’s hard disk. This nominal configuration contains the assignment of the device-type address and serial number of the device to the individual HSCI addresses. The momentary configuration is ascertained during startup of the system by requesting the serial numbers. The momentary configuration is compared with the nominal configuration. If there is a deviation, the machine operator is prompted to check the configuration.

The following applies to the assignment of the HSCI address:

- The HSCI address (bus address) is the result of the device's position in the bus
- The master (MC) always has the HSCI address 0
- The HSCI addresses of the slaves at X500 result from their position in the bus:
  - 1. Second device after the master (MC): Bus address 1
  - 2. Second device after the master (MC): Bus address 2
  - etc.
- The slave at X501 of the MC always receives the last, i.e. the highest, HSCI address in the system.

The device type address is for internally distinguishing between connected HSCI participants. Each device type (MC, CC, PL, MB, etc.) is assigned a type specification that is used to address all HSCI participants of this type.

1.3.3 Special features of the software

Firmware

HSCI components have their own firmware, which must match the currently installed NC software version. Every time the control software is started, the NC software checks the firmware versions of the individual HSCI components. After a software update, during initial operation or after replacing a hardware component a firmware update may be required.

> Note
>
> The control must not be switched off during a firmware update!

If the NC software detects that a firmware update is necessary, you will be prompted by a dialog box to start the firmware update. Press the **`Start update`** button in the dialog box for the update to be run automatically. The firmware updates are included in the NC software update package. When the firmware versions of all HSCI participants have been updated, the control boots until the "Power interrupted" message appears.

After the firmware update you are prompted to switch off the control via the main switch and leave it off for 60 seconds. This time must be maintained to ensure that all components concerned are completely reset. For technical reasons, some of the individual HSCI components must be updated successively. This makes it necessary to switch off the control repeatedly via the main switch.

Configuration

During startup, a control in an HSCI system expects the complete configuration of the HSCI system in the form of an IOC file. The IOconfig software for PCs is used to configure the system, the PLC input/output systems (PL6xxx, UEC11x, MB machine operating panel, etc.) and the PROFIBUS components. The inputs and the keys of the handwheel cannot be configured freely. The key information of the handwheels continue to be mapped on to the known PLC markers.

With IOconfig, the components of an HSCI/PROFIBUS system are specified in a project and their arrangement is described. The actual sequence of the components must be identical with the arrangement of the components in the IOC file. An MB machine operating panel or a PLB 6001, which is connected to X501, must be arranged as the last participant of the HSCI system. On the basis of this data, IOconfig generates the IOC file (*.IOC) that contains all relevant configuration data for the HSCI system. You then transfer the IOC file to the control. Enter the path and name of the project file (e.g. **`IOCCFG = PLC:\IOC\Structur.ioc`**) in **`IOCCFG =`** in the OEM.SYS file. When the control is next booted, the file is read in and evaluated by the control.

> Note
>
> All information required for using the IOconfig PC software and configuring HSCI and PROFIBUS components is contained in the IOconfig Technical Information, which is available for registered customers from the HEIDENHAIN FileBase on the Internet.

During start-up of the control, the nominal configuration of the IOC file is compared with the actual configuration. If there is a difference, you will be informed of it in a dialog box. Press the **Accept** button to confirm the new configuration and continue start-up of the control.

In addition to the configuration, the serial numbers of the individual HSCI components are also saved. During start-up they are compared with those of the detected hardware. Any difference must be checked and accepted by the user.

The control also boots without the IOC file or the entry "IOCCFG =" in the OEM.SYS. The control then automatically detects all components connected to the HSCI chain and the current configuration—without the PLC input/output systems, however. Press the **Accept** button in the dialog box to confirm the detected configuration and continue start-up of the control. However, further configuration of the control is not possible. In the **Programming and Editing** mode of operation, you can start BUS DIAGNOSIS after pressing the MOD key and the DIAGNOSIS soft key. The control will display the structure of the detected HSCI system. If required, you can use this information for the configuration with IOconfig.

Note

HEIDENHAIN recommends updating all IOC files and EAZ projects with the IOconfig version 2.2.02. This is the only way to make sure that you can use all of the features provided by the new HSCI components, see page 134.

1.3.4 HeROS 5 operating system

Boot manager

A selection menu for selecting the operating mode is provided temporarily when the control is booting. If you press one of the arrow keys of the ASCII keyboard, the booting process is interrupted at this point and the control expects you to select an operating mode ("runlevel"). If the booting process is not interrupted at this point, the control continues booting in **Runlevel 5** and starts the NC control software. You can select one of the following operating modes:

- **Runlevel 5: Start Control** (standard mode of operation)
  After the HeROS 5 operating system has booted, the NC control software is started automatically. The iTNC 530 is started in the standard condition.
- **Runlevel 4: Service Mode (graphics)**
  This operating mode is only for trained service personnel!
  After the HeROS 5 operating system has booted, the booting process is stopped. The NC control software is not started. The iTNC 530 is now in the graphic desktop of the operating system.
- **Runlevel 3: Service Mode (text)**
  This operating mode is only for trained service personnel!
  After the HeROS 5 operating system has booted, the booting process is stopped. The NC control software is not started. The iTNC 530 is in a text console of the operating system.

Select the desired operating mode with the arrow keys and press the ENTER key to start it.

HE Logging

In controls with HSCI, the TRACE settings in MP7691.x are replaced by the **HE Logging** settings. These are settings for internal diagnostic files (FILO memory) for finding errors. These files can only be evaluated by HEIDENHAIN. You can access **HE Logging** as follows if HEIDENHAIN service personnel requests you to do so:

- ▶ Make the taskbar at the bottom of the screen visible by moving the mouse pointer over it.
- ▶ Press the green HEIDENHAIN button to open the JH menu.
- ▶ Select the **Diagnostics** menu item
- ▶ Select the **HE Logging** menu item

This takes you to the **HE Logging** menu containing tabs for possible settings for **Syslog**, **FTrace** and **TCPDump**. These possibilities for diagnostics and the associated settings are to be used only together with HEIDENHAIN service personnel.

Proceed as follows if HEIDENHAIN requests you to send one of the diagnosis files:

- ▶ Open the corresponding tab **Syslog**, **FTrace** or **TCPDump**
- ▶ Under **View** at the bottom of the respective tab, select the file requested by HEIDENHAIN.
- ▶ Press the **Copy** button to the right of the file selection in the **View** settings. This opens the **Savepath...** file browser with which you can select the target folder for the diagnostic file. Keep in mind that the file is saved in the folder that is open in the file browser.
- ▶ Press the **Copy** button to save the diagnostic file in the open target folder. There you will later find the diagnostic file, can copy it from the control (e.g. with TNCremo) and send it by e-mail.

The folder **mnt** under **file system** is the folder that contains the HEIDENHAIN partitions TNC, PLC and SYS.

Tools

The HeROS 5 operating system provides additional software tools that make it possible to also open external file types and edit some of these file types. These tools are not software tools from HEIDENHAIN. They are tools from external suppliers. HEIDENHAIN will not provide any support or guarantee for the proper functioning of these tools. The following tools are available:

- File manager
  File manager of the operating system
- Mousepad
  Text editor for opening and editing TXT and INI files
- PDFViewer
  Viewer software for opening PDF files
- Ristretto
  Viewer software for opening GIF, JPG, PNG and BMP files
- Gnumeric
  Editor for opening and editing XLS and CSV files
- Firefox
  Web browser for opening HTML files

If file types that are not known to the NC control software are selected in the file manager of the control, then the operating system automatically uses the corresponding tools to open the file types it knows. The tools themselves can only be opened in the PLC mode of operation. Proceed as follows:

- While in the Programming and Editing operating mode, press the MOD key.
- Enter the code number 807 667 to switch to the PLC operating mode.
- Make the taskbar at the bottom of the screen visible by moving the mouse pointer over it.
- Press the green HEIDENHAIN button to open the JH menu.
- Select the **Tools** menu item.
- Start the desired tool.

Operating-system password

Certain accesses to or changes in the operating system are password-protected. The code number 807667 or the PLCPASSWORD defined in the OEM.SYS are required for these areas.

1.3.5 Emergency stop monitoring

External emergency stop

For the Emergency Stop routine, controls with HSCI but without Functional Safety (FS) have a "control-is-ready" output X9/3a (MC.RDY or STO.A.G) and the two Emergency Stop inputs ES.A (X9/7a) and ES.B (X9/7b). A zero signal at ES.A or ES.B triggers an Emergency Stop reaction. However, only ES.A causes the "External Emergency Stop" message to be displayed in order to avoid that the message is displayed twice. The previous PLC inputs I3 (ES.A) and I32 (ES.B) are replaced in HSCI systems by ES.A and ES.B (ES: Emergency Stop).

During an external emergency stop, you have to switch off ES.A and ES.B through PLC Module 9161.

Axis enabling

On controls with HSCI but without Functional Safety (FS), a pure axis enable, which up to now has been realized over I32, can only be realized through the function of the former X150. However, a CC 61xx does not have an X150 connector for axis-specific or axis-group-specific drive enabling.

But instead, you can now enter in machine parameters MP4132.0 to MP4132.7 eight numbers of those PLC inputs that simulate the inputs of X150. If one of these PLC inputs is switched off, the associated drives are decelerated on the provided braking ramp (usually at the limit of current). Over PLC module 9161 you have to switch off the axes of the affected axis groups. To do so, you can use PLC module 9157 to scan the corresponding PLC inputs.

MP2040.0 to MP2040.7 continue to be used for the assignment of axes to individual axis groups.

Emergency stop test

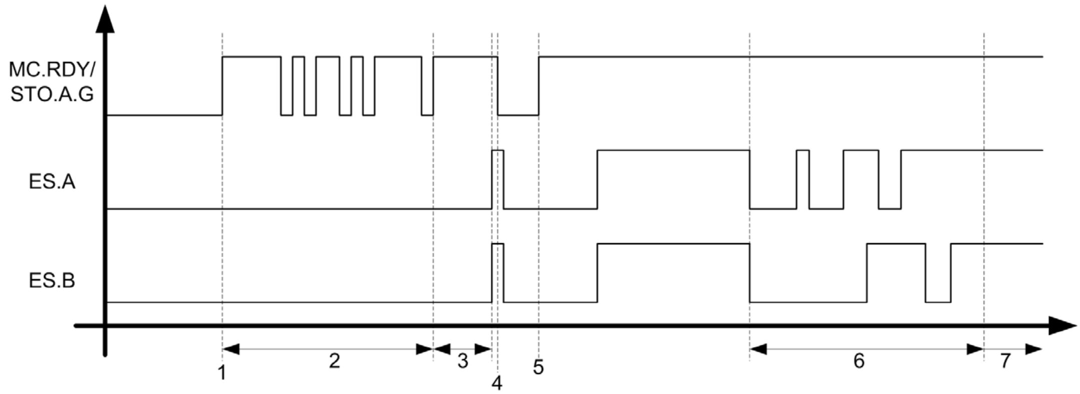
The Emergency Stop test was expanded for HSCI controls without Functional Safety (FS). After the emergency stop test or a self-test is started, internal signals of the HSCI participants are now tested for proper function in the first phase. In a further, second phase the emergency stop test and brake test is conducted with external signals (ES.A, ES.B, STO.A.G).

But it still applies, that, in the event of an error, a drop-off of the control-is-ready output (MC.RDY or STO.A.G) always triggers an emergency stop.

Note

The circuitry recommended by HEIDENHAIN is illustrated in the Basic Circuit Diagram.

Time diagram of essential signals after the control is booted and during the self test:



Step	Function	Screen display
1	Start of the self test, immediately after compiling the PLC program	Pop-up window **Self test** **HSCI components are being tested**
2	Phase 1 of the self test: Release and detection of essential internal signals are tested. In this phase of the self test, the signals STO.A.G (X9/3a) and STOS.A.G (X9/2a) are set and deleted several times.	
3	Waiting for machine control voltage	**RELAY EXTERNAL DC VOLTAGE MISSING**
4	Detection of the control voltage and cut-off of the STO.A.G/STOS.A.G signal by the NC software. ES.A/ES.B must switch to zero within one second.	**EMERGENCY STOP test**
5	Switch-on of the STO.A.G/STOS.A.G signal by the NC software.	
6	Phase 2 of the self test: Release and detection of the emergency stop signals ES.A and ES.B are tested internally. No hardware terminals are switched!	
7	Normal control operation Control voltage is on, STO.A.G/STOS.A.G output and ES.A/ES.B are at "1".	**TRAVERSE REFERENCE POINTS**

The following error messages can occur during the test:

- **`Timeout during self test`**
  At least one HSCI participant has not answered a request or has not correctly detected a signal condition to be tested. Possible causes:
  - HSCI participant/device is defective
  - HSCI cabling is faulty
  - The hardware components used and/or software are not compatible with each other.
- **`Error during self test`**
  The sequence of the individual test steps and processes in the self test do not fulfill the requirements. A signal condition to be tested is not in the required initial condition. Possible causes:
  - HSCI participant/device is defective
  - HSCI cabling is faulty
  - The hardware components used and/or software are not compatible with each other.
- **`Error in self test`**
  Has same causes as **`Error during self test`**. However, after the cause of error is corrected (e.g. by closing the guard door), the test can be continued without the control having to restart.

1.3.6 Topology and cables

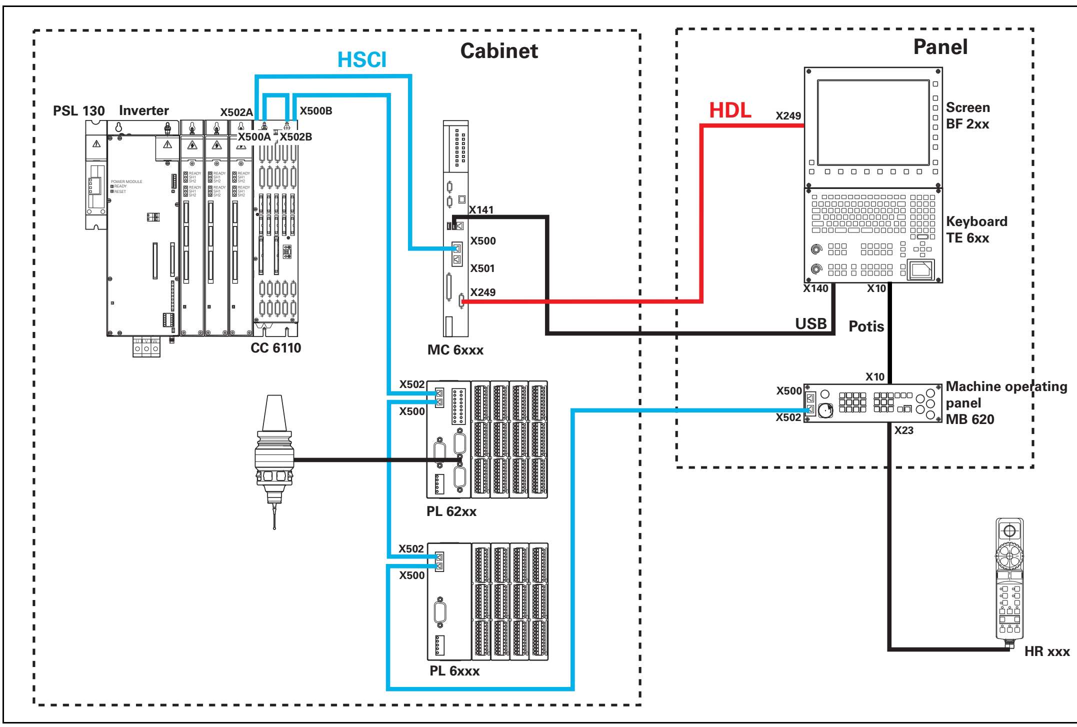
Topology

The HSCI slaves are connected—in series—to the connector of the synchronized HSCI output X500 of the MC 6xxx main computer. As a continuation, connector X502 is always the HSCI input to the HSCI slaves and X500 the HSCI output to the next HSCI slave.

The nonsynchronized second HSCI output X501 of the MC 6xxx can be used to connect a machine operating panel or a PLB 6001. However, do not connect any devices other than a machine operating panel or a PLB 6001 to X501 (no other or further HSCI components).

The HSCI components are connected via special shielded HSCI cables adapted for the increased demands of machine tool environments.

Terminating resistors are not required in the HSCI system.



The HSCI network is to be arranged as an "open" ring (line structure). Connector X500 on the MC is the beginning of the HSCI network. The HSCI connection is then led to X502 of the next HSCI participant (CC, PL or MB), where X500 again serves as the output for continuing the HSCI network to the next participant (X502). The last HSCI participant of the ring is automatically aware of its position (X500 remains vacant), and internally closes the ring of the HSCI network.

Note

A star configuration of the HSCI network (e.g. by using a hub) is not possible.

The table shows the maximum permissible number of each HSCI participant. The system allows a maximum of 18 axes plus 2 spindles (20 control loops):

<table>
<tr><th>HSCI component</th><th colspan="2">Maximum number</th></tr>
<tr><td>MC
(HSCI master)</td><td colspan="2">1 in the system</td></tr>
<tr><td>CC
(HSCI slave)</td><td colspan="2">4 controller motherboards,
(distributed to the CCs 61xx as desired)</td></tr>
<tr><td>MB 6xx(FS),
PLB 6001 (FS)
(HSCI slave)</td><td>2 in the system</td><td rowspan="3">Total number of 9 components must not be exceeded in this case</td></tr>
<tr><td>PLB 62xx (FS)
(HSCI slave)</td><td>1 in the system
(not with UEC 11x)</td></tr>
<tr><td>PLB 61xx (FS),
PLB 62xx (FS)
(HSCI slave)</td><td>7 in the system</td></tr>
<tr><td>HR (FS) handwheel
(at the MB 6xx or
PLB 6001)</td><td colspan="2">1 in the system</td></tr>
<tr><td>PLD-H xx-xx FS
(in PLB 6xxx FS)</td><td colspan="2">8 in the system</td></tr>
<tr><td>PLD-H xx-xx
(in PLB 6xxx (FS))</td><td colspan="2">64 in the system</td></tr>
</table>

Danger

If you use more than one operating station or machine operating panel, the PLC program must ensure that only one of the two operating devices is active at any one time so as to avoid danger to the operator.

The order of the HSCI participants at X500 of the MC is freely selectable. Each HSCI participant is assigned its HSCI address based on its position in the HSCI chain, see page 42.

Pin layout of the HSCI cable:

ID 618893-xx			
Female	**Color**	**Pin layout**	**Female**
1	White/Green	Data	1
2	Green	Data	2
3	White/Orange	Data	3
4	Vacant	Vacant	4
5	Vacant	Vacant	5
6	Orange	Data	6
7	Vacant	Vacant	7
8	Vacant	Vacant	8

Cable – Specifications

Device	ID	Max. bend radius (rigid configuration)	Max. bend radius (frequent flexing)	Cable diameter
HSCI	618 893-xx	≥ 40 mm	≥ 100 mm	ø 6.8 mm
HDL	625 901-xx	≥ 60 mm	≥ 150 mm	ø 11.4 mm
Position 1 V_{PP}	298 429-xx, 298 430-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
Position 1 V_{PP}	310 199-xx, 309 783-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Position EnDat	332 115-xx, 323 897-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Position EnDat	313 791-xx, 332 790-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
Analog output	290 110-xx, 290 109-xx	≥ 40 mm	[a]	ø 7.3 mm
TS 220	274 543-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
SE 640, SE 540	310 197-xx, 517 518-xx	≥ 40 mm, ≥ 10 mm	≥ 100 mm, ≥ 50 mm	ø 8 mm ø 4.5 mm
HR 130, HR 410 (VL)	281 429-xx	≥ 20 mm	[a]	ø 5.6 mm
HR 410 (VL)	296 466-xx	≥ 20 mm	[a]	ø 5.6 mm
HR 410	296 687-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
RS-232, 9-pin	355 484-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
RS-232, 9-pin (VL extension cable)	366 964-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
RS-232, 25-pin	365 725-xx	≥ 40 mm	≥ 100 mm	ø 7.1 mm
RS-232, 25-pin (VL extension cable)	274 545-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
USB[b]	354 770-xx	≥ 20 mm	≥ 75 mm	ø 4.5 mm
USB (with hub)[c]	624 775-xx	Cable like 354 770-xx, hub: ø ~ 20 mm, length ~ 115 mm		

a. Conditionally resistant to frequent flexing and torsion
b. These USB cables support USB 1.1 and USB 2.0
c. The hubs integrated in the USB cable (ID 624 775-xx) only support USB 1.1

Cables – Instructions for wiring

Keep the following in mind for wiring inside the electrical cabinet:

The stray magnetic field of a KDR 1xx commutating reactor can disturb conductors routed in its proximity. This means that no conductors (such as dc-voltage lines and signal lines), except for the power connections of the motor and the connections between the UVR 1xx and KDR 1xx, should be routed in the proximity of the KDR 1xx. HEIDENHAIN recommends leaving a space of 20 cm around the commutating reactor. This distance is to be maintained regardless of the KDR 1xx used.

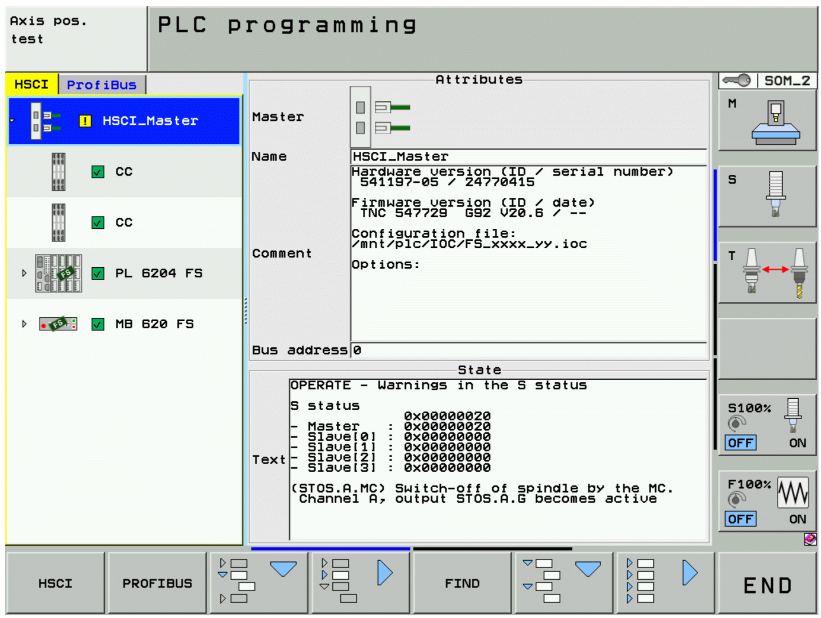
1.3.7 Bus diagnosis

The control features functions for diagnosis of the HSCI or PROFIBUS system.

You can access the functions as follows:

- Switch to the **Programming** mode of operation
- Press the MOD key.
- Press the DIAGNOSIS soft key.
- Press the BUS DIAGNOSIS soft key.

Example screen:



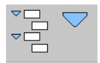
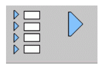
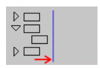
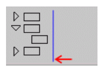
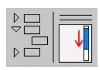
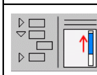
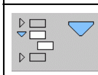
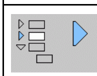
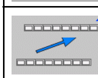
In Diagnosis mode, the structure of the HSCI/PROFIBUS system as well as the details of the HSCI/PROFIBUS components can be displayed.

The **HSCI** and **PROFIBUS** tabs above the left window are only shown if both bus systems are connected to the control. In this case you can switch between the bus systems by pressing the HSCI and PROFIBUS soft keys.

The detected, actually connected hardware at the HSCI bus (actual configuration) is displayed. The control also knows from the IOC file the previously configured hardware configuration to be expected (nominal configuration). The HSCI bus diagnosis compares the actual configuration to the nominal configuration and reveals deviations. At HSCI bus positions where a deviation was found, the actually configured hardware (**IOC:**) is displayed behind the detected connected hardware (dialog, e.g. **/IOC: MB 620**). This provides you with a direct comparison of the actual configuration and the nominal configuration in order to support you in finding errors. HSCI components that are missing from the actual configuration (e.g. due to an interruption in the HSCI bus) are identified by a corresponding symbol.

Soft keys for HSCI/ PROFIBUS diagnostics

The following soft keys are available in the main menu for bus diagnosis:

Soft key	Submenu	Function
HSCI		Select the HSCI or PROFIBUS bus system
PROFIBUS		
		Open the HSCI/PROFIBUS tree
		Shrink the HSCI/PROFIBUS tree
		Move the separating line (enlarge/reduce the window size)
		
MORE FUNCTIONS		Opens the submenu with additional functions
		Move the state window up or down
		
		Open/Close HSCI/PROFIBUS slaves
		
	FIND	Find an HSCI/PROFIBUS component
		Return to the previous soft key row
END		Exit the BUS diagnostics

Navigation with the arrow keys of the operating panel:

- ↓, ↑ Select an HSCI/PROFIBUS component
- → (or + key) Open the HSCI/PROFIBUS component for the connected modules or terminals to appear
- ← (or – key) Close the HSCI/PROFIBUS component

Screen contents

"HSCI/PROFIBUS" window (at left)

- Arrangement and designation of the HSCI/PROFIBUS components and terminals
- Status of the HSCI/PROFIBUS components and terminals:

 OK

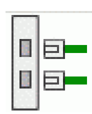
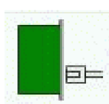
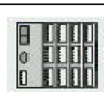
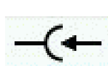
 Error (further information in the text window)

 Warning (further information in the text window)

 Undefined state (further information in the text window)

"Attributes" window (at upper right)

Configuration parameters of the selected HSCI/PROFIBUS component (see table).

Field	Function	Formed from...
Designation and symbol of the HSCI/PROFIBUS component		
	 HSCI master	
	 PROFIBUS master	
	 Slave	
	 (Slave) module, AS-i slave	
	 Terminal	
Name	Name of the slave, slave module or terminal	- "Name" entered for the slave / slave module - "Symbolic name" entered for the terminal

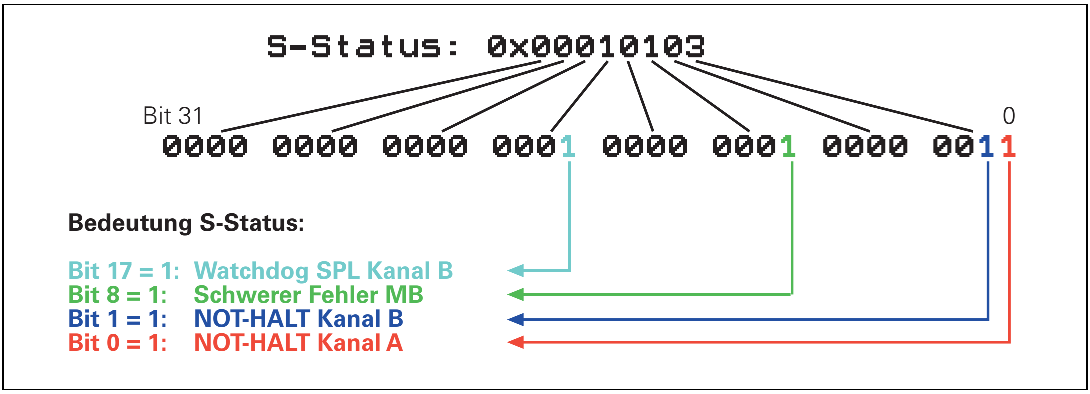
Field	Function	Formed from...
Comment	Name of the slave (for slaves and slave modules) or terminal description	Master entries: • "Configuration file" Master/slave entries: • "Name" • "Hardware ID" • "Hardware serial number" • "Firmware ID" • "Firmware timestamp" • "Configuration file" Terminal entries: • "Comment" • "Pin name" • "Plan page" • "Order number"
Option	Designation of the option	Slave / slave module entered in "block"
Bus address	HSCI/PROFIBUS address	Slave entered in "HSCI/PROFIBUS address"
Slot	Slot number of the slave module	
Terminal	Terminal number	Determined from the terminals entered in "I/O offset" and "I/O bit"
Operand	Operand designation	Terminal entered in "PLC operand address"
Order no.	Order number	"Order number" entered for the terminal
Plan page	Wiring diagram page	"Page" entered for the terminal
Pin name	Terminal name	"Pin name" entered for the terminal

"State" window (at lower right)

Current status of selected HSCI/PROFIBUS component.

- **Value:** Operand value of the terminal in decimal and hexadecimal code
- **Text:** Error texts that appear when an error occurs; otherwise, no display. The texts displayed are error-specific texts from the HDD/GSD file or the error information configured in IOconfig.
- Display of S status (31 bits) in HEX code. The meaning is displayed in abbreviated form below the S status.

You can use the following tables to look up the meaning of the individual S status bits. To do this, you need to convert the displayed HEX code to a binary number. Example:



S status bit	Signal	Meaning
0	–ES.A	Emergency stop channel A (SS1)
1	–ES.B	Emergency stop channel B (SS1)
2	–ES.A.HW	Emergency stop channel A, handwheel (SS1); no function in controls without Functional Safety (FS)
3	–ES.B.HW	Emergency stop channel B, handwheel (SS1); no function in controls without Functional Safety (FS)
4	–STO.A.MC.WD	Watchdog of MC software, switch-off of inverters, A channel (SS1F, with Functional Safety (FS): switch-off of FS outputs)
5	–STOS.A.MC	Spindle is switched off by the MC, A channel, STOS.A.G is triggered (CC: switch-off of spindle); no function in controls without Functional Safety (FS)
6	–STO.B.CC.WD	Watchdog of CC software, switch-off of inverters, B channel (SS1F)
7	–SMC.A.WD	"Fast" watchdog of MC software (SS1); alarm on CC, which initiates the deceleration of the axes
8	–SPL.WD	With FS: Multi-channel watchdog of SPL firmware (A/B channel); serious error of PL (SS1F) Without FS: Single-channel watchdog of PL firmware
9	–SMOP.WD	With FS: Multi-channel watchdog of SMOP firmware (A/B channel); serious error of MOP machine operating panel (SS1F) Without FS: Single-channel watchdog of MOP firmware (machine operating panel)

S status bit	Signal	Meaning
10	–PF.PS.AC	Power supply of inverter too low (parameterized LIFT OFF function in some cases)
11	–PF.PS.DC	DC-link voltage U_Z too low (CC: SS1)
12	–PF.BOARD	Error in the supply voltage of the respective module (SS1F)
13	–N0	Internal S status bit (CC: SS1)
14	–REQ.SS2	Alarm (SS2)
15	–	Reserved

The following additional status bits are available for an external PL:

S status bit	Signal	Meaning
16	–SPL.A.WD	SPL watchdog, channel A
17	–SPL.B.WD	Only in controls with Functional Safety (FS): SPL watchdog, channel B
18	PGOOD.NC	Voltage monitoring of NC reports an error
19	PGOOD.PLC	Voltage monitoring of PLC reports an error
20	–INT	Internal interrupt
21..31	1	Reserved

The following additional status bits are available for an external MB machine operating panel:

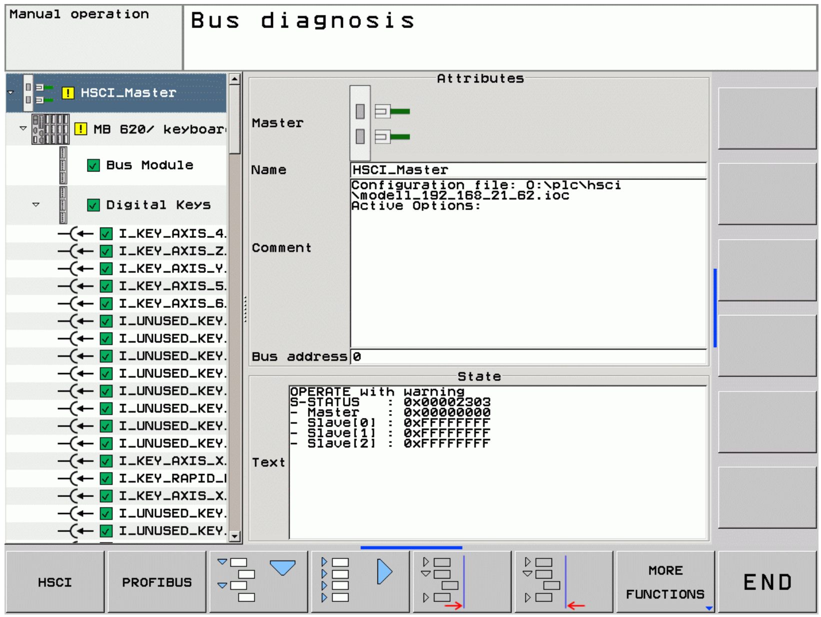
S status bit	Signal	Meaning
16	–SMOP.A.WD	SMOP watchdog, channel A
17	–SMOP.B.WD	Only in controls with Functional Safety (FS): SMOP watchdog, channel B
18	PGOOD.A	Voltage monitoring of channel A reports an error
19	PGOOD.B	Voltage monitoring of channel B reports an error
20	1	Reserved
21..31	1	Reserved

Overview of HSCI/PROFIBUS I/O configuration

The number of inputs and outputs transmitted via HSCI/PROFIBUS are listed in the HSCI/PROFIBUS diagnosis.

It can thus, for example, be tested whether further Profibus slaves can be connected without exceeding the maximum number of 252 input bytes and 252 output bytes.

In order to see this overview, select the master to be displayed in the HSCI/PROFIBUS diagnosis.



1.3.8 Supply voltages in the HSCI system

Two separate 24 V power supplies must be used to supply the **+24 V power** to the individual control components in the HSCI system: +24 V NC and +24 V PLC.

HEIDENHAIN recommends using the dc-link buffered PSL 13x for supplying power to the HSCI components, See "PSL13x Low-Voltage Power Supply Unit" on page 145.

If you are using the UEC 11x, the main computer and other NC components are usually supplied via the 24 V power supply unit of the UEC (connection X90). A PSL 130 is not necessary if the total current consumption of the NC supply of all HSCI components does not exceed 3.5 A.

External PL assemblies with HSCI (e.g. the PL 6xxx or the PL in the UxC 11x) consist of an HSCI part (bus module and logic) and a PLC part (PLC input/output assemblies). Due to the topology of the HSCI system, the 24 V NC voltage supplied to the HSCI part must comply with the requirements for double basic insulation according to EN 50 178 (PELV) All other NC components with HSCI interface (e.g. main computer and machine operating panel) must also be supplied with 24 V NC voltage with double basic insulation. The reason for the double basic insulation is electrical safety, e.g. accessibility of connecting elements supplied with +24 V NC voltage.

PLC components, such as motor holding brakes and solenoid valves, usually have simple basic insulation. The PLC part must therefore be powered by another +24 V supply voltage. The two supply voltages must not be connected to each other. The double basic insulation of the NC power supply is removed through "mixed operation," i.e. +24 V NC voltage with double basic insulation is connected to PLC components with simple basic insulation. This is not permitted in an HSCI system.

The following components are powered by +24 V NC supply voltage:

- MC 6xxx main computer unit
- BF 2xx TFT visual display unit
- MB 6xx machine operating panel
- TE 6xx keyboard unit

Protective Extra Low Voltage (PELV) according to EN 61800-5-1 must be complied with by the power supply unit for the +24 V NC supply voltage.

The following components are powered by +24 V PLC supply voltage:

- PLB 6xxx input/output module
- PLD digital slot for PLB
- PLA analog slot for PLB
- Motor holding brakes, further components in the PLC circuit

Note

HEIDENHAIN recommends also using a power supply unit complying with Protective Extra Low Voltage (PELV) according to EN 61800-5-1 for the +24 V PLC power supply, although the circuit has only ELV status. In addition, HEIDENHAIN recommends connecting the 0 V PLC supply voltage to protective earth (PE). This is not strictly required according to the VDE standards. However, it provides additional safety in the event of insulation failure in the PLC circuit.

The CC 6xxx controller unit is supplied by the X69 supply bus of the HEIDENHAIN supply module and X74 (+5 V).

Danger

- The +24 V NC supply voltage (PELV system according to EN 50178) is required to be safely separated voltage for the entire HSCI system and must not be connected to the +24 V PLC supply voltage (ELV) of the system.
- Protective Extra Low Voltage (PELV) according to EN 61800-5-1 must be complied with for the +24 V NC power supply of the machine.
- VDE 0160/EN 50178 is to be observed for the +24 V NC voltage lines and cable routing. Lines or cables for safely separated electric circuits must therefore have double or reinforced insulation between the wire and the surface if they are routed without spatial separation from other cables and lines.
- Due to the structure of the PLC area in the HSCI system, the +24 V PLC supply voltage is a voltage with basic isolation (ELV as per EN 61800-5-1).
- The 0 V signal of the NC power supply must be connected by a 6 mm^2 conductor to the machine's central functional ground (B).
- The 0 V signal of the PLC power supply must be connected by a 6 mm^2 conductor to the machine's central protective ground (PE).

The signal ground is used for functional-equipotential bonding. The signal-ground connections (B) of the HEIDENHAIN control components must be connected to the central functional ground of the machine (minimum cross section 6 mm^2).
The 0-V PLC and all of the protective-ground connections of the HEIDENHAIN control components must be connected separately from the signal-ground connections to the central protective ground (PE) of the machine (minimum cross section 6 mm^2).
The central signal ground and the central protective ground must be connected with each other once for the machine! The cross section of this conductor must be at least as large as the largest cross section of the conductors for connecting the components used to protective ground or functional ground, see page 155.

Note

The line cross section of the +24 V NC power supply must be designed for the power consumption of the connected devices. EN 60204-1 lists the protection provided by line cross sections.

Minimum cross section of the +24 V NC power supply: 0.75 mm^2

Note

The motor brakes are controlled by 24 V PLC voltage. The trigger circuit and the brake itself are usually separated from the line power only by basic insulation according to EN 618100-5-1 (also EN 50178). Also, other add-on devices that are controlled by PLC circuits usually have only basic insulation from the line power.

The CC 61xx controller unit is supplied with a power of **+5 V** by the power supply units via supply bus X69 and connector X74, see page 112.

For information on the power supply units, refer to the "Inverter Systems and Motors" Technical Manual.

Device	Load capacity
UVR 1xxD, UE 2xxD	20.00 A via X74 10.00 A via X69

Device	Load capacity
UEC 11x	3.5 A via X90 (+ 24-V NC)

Device	Current consumption of the 5 V supply
CC 6106/6 control loops	3.80 A
	Consisting of: 1 drive-control motherboards: 2.00 A 2 drive-control expansion boards: 0.90 A each
CC 6108/8 control loops	5.80 A
	Consisting of: 2 drive-control motherboards: 2.00 A 2 drive-control expansion boards: 0.90 A each
CC 6110/10 control loops	6.70 A
	Consisting of: 2 drive-control motherboards: 2.00 A 3 drive-control expansion boards: 0.90 A each
UEC drive control board	2.50 A
LS, LB	0.15 A
ERN, ROD, RON	0.20 A
Absolute rotary encoders	0.25 A (+0.085 A with line-drop compensator)
Absolute angle encoders	0.35 A (+0.085 A with line-drop compensator)
LC	0.30 A (+0.085 A with line-drop compensator)

Note

Please also refer to the Technical Manual for your control and the "Inverter Systems and Motors" Technical Manual.

X101: NC power supply

The MC 61xx is supplied with the +24 V NC (machine control voltage) of the machine, for example by the PSL 130.

Power supply: Minimum absolute value: +20.4 V–
Maximum absolute value +28.8 V–

Pin layout:

Connecting terminals at X101	Pin layout	Fuse
+	+24 V NC	7-A safety fuse integrated in the MC
–	0 V NC	

Power consumption of the MC 6241:40 W

Power consumption of the MC 6222:60 W

Attention

Ensure that either the dc-link power supply unit is switched off or the line power is disconnected before connecting the power cables!

1.3.9 Power supply for PLC outputs

The PLC outputs of the PLB 6xxx are powered by the 24-V control voltage (PLC) of the machine (in accordance with VDE 0551). The power to the PLC outputs is supplied via the corresponding terminals on the respective I/O module connectors for PLC outputs.

The control voltage must be smoothed with a minimum 1000 µF at a rated current capacity of 150 µF/A. At a current load of 15 A, for example, this corresponds to a capacity of 2250 µF.
If the PSL 13x is used as 24 V– supply unit, this additional smoothing is not necessary.

Note

HEIDENHAIN recommends the PSL 13x (575 047-01) as 24 V– power supply unit, see page 145.

EN 61 131-2:1994 permits:

- Minimum absolute value: 20.4 V–
- Maximum absolute value: 25.4 V– at 200 W power output
- Maximum absolute value: 28.8 V– at 100 W power output

Attention

Use only original replacement fuses.

Power consumption

If half of the outputs are switched at the same time, the following are the values for power consumption:

PL 6xxx: approx. 485 W
UEC 11x: 48 W

Power output

The maximum permissible output of a PLD-H xx-xx-xx is 200 W.

Rated operating current per output

UEC 11x: 0.150 A

PLD-H xx-xx-xx: 2 A
Simultaneity with a supply voltage of 25.4 V:
4 outputs with 2 A each
8 outputs with 1 A each
Total current:
Out0 to Out7: ≤ 8 A
Out0 to Out3: ≤ 4 A
Out4 to Out7: ≤ 4 A

For all PLD-H xx-xx-xx it must be remembered that a total current of 8 amperes maximum per slot (PLD-H) must not be exceeded! This applies regardless of the number of PLD-H outputs.

1.3.10 Power supply for PLB 6xxx

Note

The control cyclically monitors the supply voltage of the PL 6xxx.

Power consumption of the PL 6xxx via X3, +24 V NC: max. 48 W

Power consumption of the PL 6xxx via X3, +24 V PLC: max. 21 W

The power to the PLC outputs is also supplied via the corresponding terminals on the respective I/O module connectors for PLC outputs. The power consumption of the +24 V PLC via X3 and the power consumption of the PLC outputs add to each other.

X3: +24 V NC, +24 V PLC power supply

Pin layout for X3: Power supply for logic circuit and PLC outputs

Connecting terminal	Pin layout
1 (top terminal)	+ 24 V NC
2	0 V NC (ground +24 V NC)
3	⏚ Protective ground Minimum wire cross section of the power cables for 24 V PLC
4	+ 24 V PLC
5 (bottom terminal)	0 V PLC (ground +24 V PLC)

1.3.11 Power supply for control-is-ready signal

X9: Power supply for control-is-ready signal

The control-is-ready output (X9/3a) is powered by 24 V PLC.

Pin layout:

Connecting terminal X9	Pin layout
1a	+24 V PLC
2b	0 V PLC

1.4 Overview of Components

Hardware components		ID
MC 6241	Main computer 1.8 GHz with HDR, electrical cabinet version without Profibus	573 398-xx
MC 6241	Main computer 1.8 GHz with HDR, electrical cabinet version with Profibus	653 220-xx
MC 6222	Main computer with 15-inch TFT display, 1.8 GHz with SSDR, operating-panel version, without Profibus	634 109-xx
MC 6222	Main computer with 15-inch TFT display, 1.8 GHz with SSDR, operating-panel version, with Profibus	634 113-xx
HDR iTNC	Hard disk for MC 6x41, 80 GB, NC software 606 420-01	682 272-01
HDR iTNC	Hard disk for MC 6x41 (export version), 80 GB, NC software 606 421-01	682 272-51
SSDR iTNC	Solid State Disk for MC 6222, 32 GB, NC software 606 420-01	736 591-01
SSDR iTNC	Solid State Disk for MC 6222 (export version), 32 GB, NC software 606 421-01	736 591-51
SIK iTNC	SIK for MC 62xx, single-processor version, incl. SW option 2	586 084-01
SIK iTNC	SIK for MC 62xx, single-processor version, incl. SW option 2 (export version)	586 084-51
BF 250	15-inch TFT display with HDL connection	599 916-xx
CC 6106	Controller unit for HSCI for max. 6 control loops	662 636-xx
CC 6108	Controller unit for HSCI for max. 8 control loops	662 637-xx
CC 6110	Controller unit for HSCI for max. 10 control loops	662 638-xx
Connecting cables	Ribbon cable for connecting 4 x connectors X69 of the CC (2 x CC 6108, 2 x CC 6110, CC 6108 to the CC 6110, 4 x CC 6106)	325 816-22
UEC 111	Controller unit with inverter and PLC, 4 control loops	625 777-xx
UEC 112	Controller unit with inverter and PLC, 5 control loops	625 779-xx
PSL 130	Low-voltage power supply unit, 750 W, for +24 VNC and +24 V PLC	575 047-xx
PSL 135	Low-voltage power supply unit, 750 W, for +24 V NC, +24 V PLC and +5 V NC	627 032-xx
MS 110	Mounting housing for multi-row configuration	658 132-xx
MS 111	Mounting case for multi-row assembly, additional connection for 24 V supply to the fan	673 685-xx
TE 620	Keyboard unit without touchpad	625 806-xx
TE 630	Keyboard unit with touchpad	617 976-xx
TE 635Q	TE with touchpad and integrated MB for HSCI connection	617 975-xx

Hardware components		ID
MB 620	Machine operating panel for HSCI connection	617 973-xx
PLB 6001	HSCI adapter for OEM-specific machine operating panel, 64 digital inputs, 32 digital outputs	668 792-xx
PLB 6104	PLB for HSCI, 4 slots	591 828-xx
PLB 6106	PLB for HSCI, 6 slots	630 058-xx
PLB 6108	PLB for HSCI, 8 slots	630 059-xx
PLB 6204	PLB for HSCI, 4 slots, with system module	591 832-xx
PLB 6206	PLB for HSCI, 6 slots, with system module	630 054-xx
PLB 6208	PLB for HSCI, 8 slots, with system module	630 055-xx
PLD-H 16-08-00	PL for PLB 6xxx: 16 digital inputs, 8 digital outputs	594 243-xx
PLD-H 08-16-00	PL for PLB 6xxx: 8 digital inputs, 16 digital outputs	650 891-xx
PLA-H 08-04-04	PL for PLB 6xxx, 8 x ±10 V inputs, 4 x ±10 V analog outputs, 4 x PT 100 inputs	675 572-xx
PL empty housing	Empty housings for slots of a PL 6xxx	383 022-11
HSCI cable	HSCI connecting cable	618 893-xx
HDL cable	Connecting cable for connection to BF	625 901-xx

If you want to design the control system in accordance with the new standard ISO 13849-1, you need the corresponding validation values of the individual control components for calculating the required performance level. This also applies to the inverters and power supply modules used in a system with HSCI but without integrated Functional Safety (FS). Only devices whose index or version number is greater than or equal to the device version numbers listed below in the right column are permitted for use in accordance with ISO 13849-1.

Furthermore, in HSCI systems with integrated Functional Safety (FS) you may use only inverters or power supply modules that have been certified for use in such systems. Please take this into account when configuring your machine. Suitable devices are listed below in the middle column.

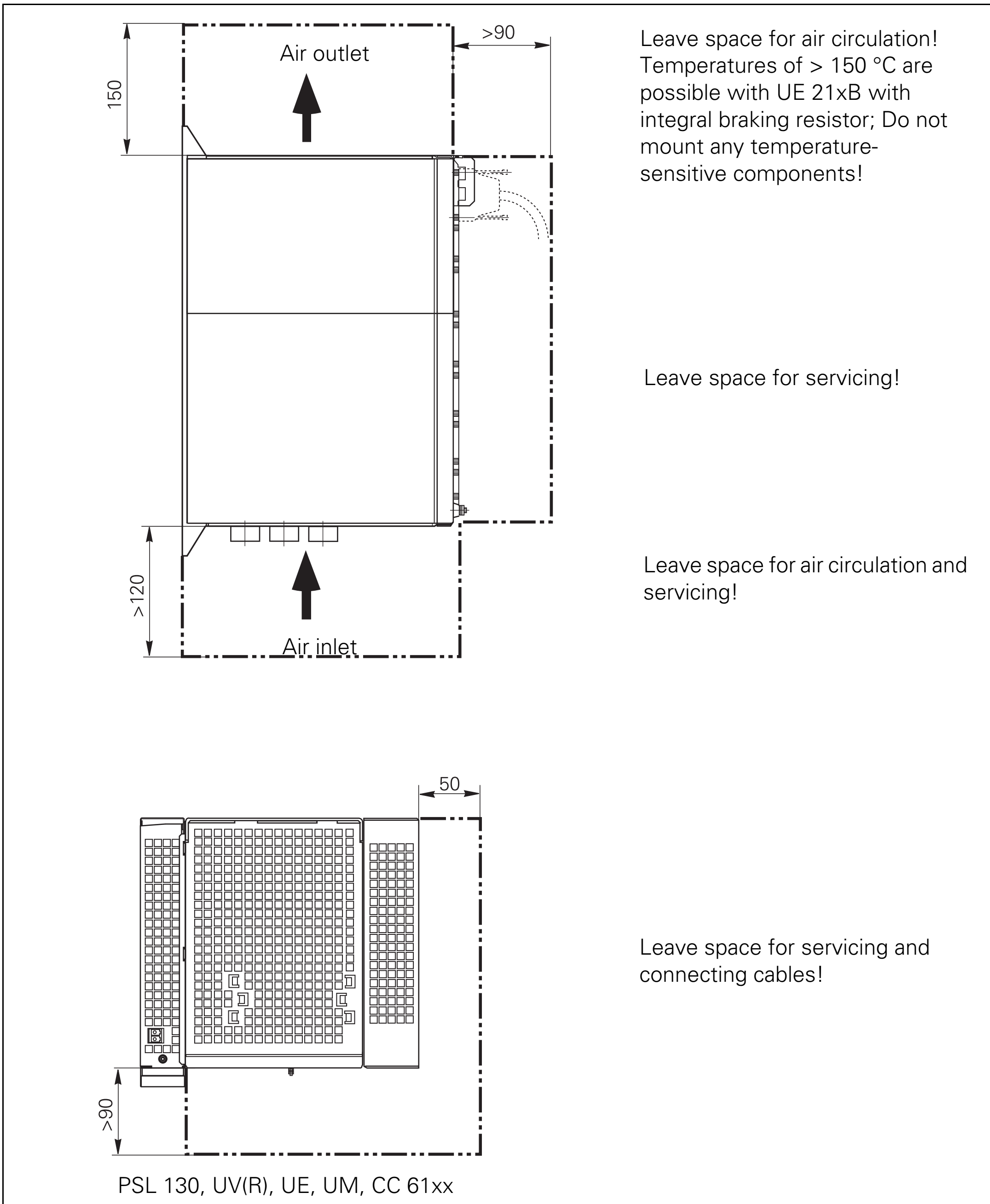
Below you will find an overview of the devices that—according to ISO 13849— are permitted for use in systems with and without FS.

Device designation	**Device ID for systems with integrated FS**	**Device ID for systems without integrated FS**
Inverter modules		
UM 116D	667954-01	542998-01
UM 116DW	667946-01	369629-01 index B
UM 115D	671566-01	387852-01 index E
UM 114D	671288-01	510509-01 index E
UM 113D	730435-01	518703-01 index B
UM 112D	731984-01	519971-01 index C
UM 122D	667633-01	519972-01 index C
UM 121BD	667942-01	513037-01 index C
UM 111BD	671968-01	513035-01 index E
UM 121D	667838-01	392319-01 index F
UM 111D	667945-01	392318-01 index F
Power supply modules		
UVR 120D	728252-01	390188-01 index K
UV 130D	728250-01	389311-01 index E
UVR 130D	728248-01	377639-01 index K
UVR 140D	728253-01	390281-01 index N
UVR 150D	728255-01	390421-01 index P
UVR 160D	728257-01	530341-01 index G
UVR 160DW	728258-01	560106-01 index G
Non-regenerative compact inverters		
UE 210D	733 421-01	558302-01 index C
UE 211D	733 423-01	558303-01 index C
UE 212D	733 424-01	558304-01 index C
UE 230D	733 425-01	558305-01
UE 240D	733 426-01	558306-01
UE 241D	733 427-01	558307-01
UE 242D	733 428-01	558308-01
UE 110	Not yet available	375713-02 index B

Device designation	Device ID for systems with integrated FS	Device ID for systems without integrated FS
UE 111	Not yet available	375714-02 index B
UE 112	Not yet available	375715-02 index B
Regenerative compact inverters		
UR 242D	Not yet available	536565-01 index A
UR 230D	Not yet available	536561-01 index A
UR 240D	Not yet available	536564-01 index B

1.5 Mounting Position of MC 6xxx, CC 61xx, UV xxx, UM xxx, UE 2xx B

> **Attention**
>
> When mounting, please observe proper minimum clearance, space requirements, length and position of the connecting cables.



If filtered air is blown into the electrical cabinet for cooling purposes, the standard EN 61800-5-1 applies.

Danger

Be sure to take the measures required for preventing dust or water from entering the electrical cabinet or the housing.
Dust depositing inside electrical devices may cause them to fail and impair the safety of the system. Max. contamination level 2 is permitted for the components.

All electric and electronic control components must be installed in an environment (e.g. electrical cabinet, housing) that fulfills the requirements of protection class IP54 (dust and splash-proof protection) in order to fulfill the requirements of contamination level 2.

1.6 Storage and Operating Temperatures

1.6.1 Limit values

Device	Air approaching the device in the panel / electrical cabinet	Temperature range outside the panel / electrical cabinet
MC 6xxx in panel without HDR	0°C to +50°C	0°C to +45°C (no direct exposure to sunlight)
MC 6110, MC 6120	0°C to +50°C	0°C to +45°C (no direct exposure to sunlight)
MC 6210	0°C to +50°C	0°C to +45°C (no direct exposure to sunlight)
MB 6xx, TE 6xx, PLB 6001	0°C to +50°C	0°C to +45°C
HR 4xx, HR 5xx		0°C to +45°C
MC 6x41	+5°C to +40°C	
PLB 61x, PLB 62x, PLD-H, PLA-H	+5°C to +40°C	
BF 2xx	0°C to +50°C	0°C to +45°C (no direct exposure to sunlight)
HDR hard disk	+5°C to +55°C	0°C to +45°C (no direct exposure to sunlight)
CompactFlash card	0°C to +70°C	0°C to +45°C (no direct exposure to sunlight)

1.6.2 Limit value for temperature inside the panel

Because the MC 62xx is installed in the panel, additional heat is generated that must be emitted to the surroundings through the panel wall. The required temperature difference depends on the design and the size of the panel.

The maximum permissible temperature of the air surrounding the panel is 45°C. However, the temperature of the air inside the panel, which flows into the HEIDENHAIN devices (e.g. the MC), is decisive for the HEIDENHAIN devices integrated in the panel.

> **Attention**
>
> The temperature of the air inside the panel, which flows into the HEIDENHAIN devices, must not exceed a maximum temperature of +50 °C.

The panel must be designed in such a way that this maximum temperature of +50 °C is not exceeded. Please verify this through an appropriate temperature measurement.

With the recommended temperature limits, active cooling should not be required for a typical panel.

1.6.3 Humidity during operation

> **Attention**
>
> Condensation on the electronics is not permitted!

Condensation can form, for example, if warm, moist air flows along cool surfaces of the electronics. Therefore, cooling units with discontinuous (e.g.two-position) temperature control must not be used. The resulting cyclic changes in temperature and humidity can cause condensation on the cool surfaces of the electronics. Furthermore, you must prevent condensate from reaching the electronics and ensure that it can drain away without causing damage.

1.6.4 Storage temperatures

For all control components: –20 °C to +60 °C.

NiMH rechargeable batteries in the HR 550FS: –20 °C to +21 °C at a charge status of 40%.

Humidity during storage of NiMH rechargeable batteries
NiMH rechargeable batteries in the HR 550FS: Relative humidity 5 < 60% rF

1.6.5 Limit values for ambient conditions

HEIDENHAIN specifies the range of application 2 for the use of its control products.

Furthermore, the following limit values apply:

Characteristic values during operation	**Limit values to be maintained**	**Standard to be complied with**
Vibration	+/– 0.075mm, 10 Hz to 41 Hz; 5 m/s2, 41 Hz to 500 Hz;	DIN EN 60068-2-6
Shock	50 m/s2, 11 ms	DIN EN 60068-2-27
Relative air humidity	75% in continuous operation; 95% for not more than 30 days a year (randomly distributed)	
IP protection	IP 54 for HW, TFT, MB, TE	DIN EN 60529
ESD 61000-4-2	Severity level 3	DIN EN 6100-4-2
HF field 61000-4-3	Severity level 3	DIN EN 6100-4-3
Burst 61000-4-4	Severity level 3	DIN EN 6100-4-4
Surge 61000-4-5	Severity level 3	DIN EN 6100-4-5
Conducted disturbances	Severity level 3	DIN EN 6100-4-6

1.7 MC 6xxx Main Computer, HDR and SIK

1.7.1 General information

There is a new MC 6xxx hardware for the standard (single-processor) versions of the iTNC 530 for connection via HSCI.

Properties of the MC 6241 (comparable with MC 422C):

- Pentium M with 1.8 GHz
- 1 GB RAM
- Main computer unit for single-processor version
- Electrical cabinet version
- Slot for HDR with real-time HeROS operating system

Properties of the MC 6222 (comparable with MC 422C):

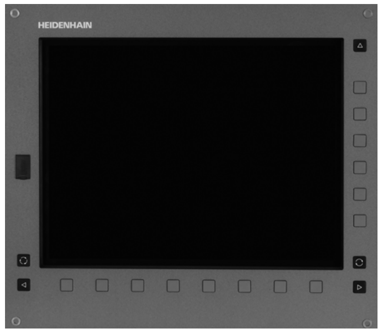
- Pentium M with 1.8 GHz
- 1 GB RAM
- Main computer unit for single-processor version
- Operating-panel version
- Slot for SSDR with real-time HeROS operating system

Main computer	ID
MC 6241 without Profibus	573 398-xx
MC 6241 with Profibus	653 220-xx
MC 6222 without Profibus	634 109-xx
MC 6222 with Profibus	634 113-xx

Please note that iTNC 530 software earlier than version 606 42x-01 cannot be run on the MC 6xxx.

Note

HSCI systems that ran on NC software 340 49x-05 during the introductory phase must migrate to the new NC software 606 42x-01.

MC 6222 main computer		
The iTNC 530 with HSCI comprises 2 components: ■ MC 6xxx main computer (MC = Main Computer) ■ CC 6xxx controller unit (CC = Controller Computer) ■ Weight: 5.8 kg, including SSDR		

MC 6241 main computer	
The iTNC 530 with HSCI comprises 2 components: ■ MC 6xxx main computer (MC = Main Computer) ■ CC 6xxx controller unit (CC = Controller Computer)	

Cycle times of main computer (block processing):

- MC 62xx: 0.5 ms
- MC 61xx: 3.6 ms (0.5 ms with software option 9)

HDR

HDR hard disk for iTNC 530 with HSCI	ID
Export version (MC 6x41)	682 272-51
Standard version (MC 6x41)	682 272-01

The following sizes are valid for the partitions on HDR hard disks for the iTNC 530 that were delivered with the NC software for HSCI:

Partition	Contents	Size
SYS	System files	2 GB (for multiple software versions in packed format)
PLC	OEM files	1 GB
TNC	User files	64 GB

SSD

SSDR Solid State Disk for iTNC 530 with HSCI	ID
Export version (MC 6222)	736 591-51
Standard version (MC 6222)	736 591-01

The following sizes are valid for the partitions on SSD Solid State Disks for the iTNC 530 that were delivered with the NC software for HSCI:

Partition	Contents	Size
SYS	System files	2 GB (for multiple software versions in packed format)
PLC	OEM files	1 GB
TNC	User files	21.4 GB

SIK

SIK with NC software license for MC 61xx (single processor)		ID (standard)	ID (export)
iTNC 530 with HSCI			
	For 4 control loops with CC 61xx Without software options	617 776-01	617 776-51

SIK with NC software license for MC 62xx (single processor)		ID (standard)	ID (export)
iTNC 530 with HSCI			
	For 4 control loops with CC 61xx Without software options	586 048-01	586 048-51

Further control loops can be enabled either as groups or individually. The combination of control-loop groups and individual control loops makes it possible to enable any number of control loops. Up to 20 control loops are possible.

In addition to the four control loops of the basic version, further control loop groups can be enabled through software options. There are control-loop groups with four and with eight control loops:

SW option: Control loop groups:	ID
#77: 4 additional control loops	634 613-01
#78: 8 additional control loops	634 614-01

In addition to the four control loops of the basic version and the control-loop groups, up to eight individual control loops can be enabled through software options. These can also be enabled in addition to the control-loop groups:

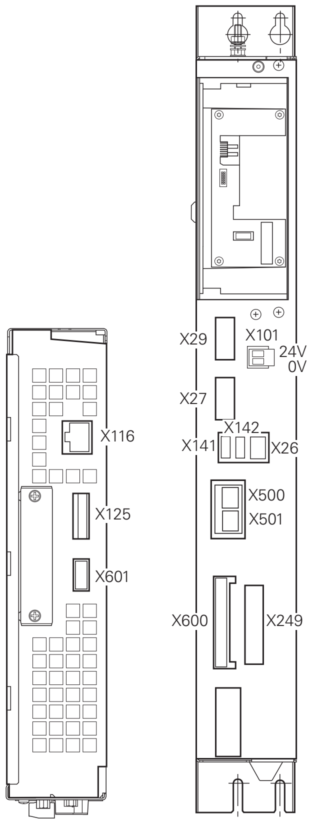
SW option: Control loop groups:	ID
#0: 1st additional control loop	354 540-01
#1: 2nd additional control loop	353 904-01
#2: 3rd additional control loop	353 905-01
#3: 4th additional control loop	367 867-01
#4: 5th additional control loop	367 868-01
#5: 6th additional control loop	370 291-01
#6: 7th additional control loop	370 292-01
#7: 8th additional control loop	370 293-01

The control loops of the CC 61xx can operate selectably as single-speed or double-speed axes. This requires software option 49:

SW option:	ID
#49: Double speed for control loops	632 223-01

1.7.2 Connection overview

MC 6x41

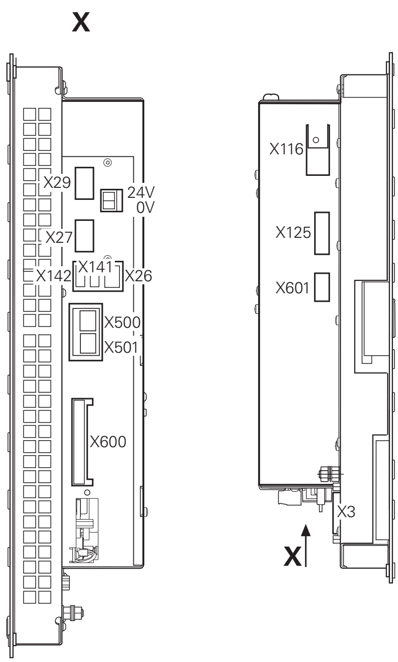


Pin layout	Connector	Function	Page
	X29	Reserved	
	X26, X116	Ethernet data interface, X116 on bottom	82
	X27	RS-232-C/V.24 data interface	
	X141, X142	USB 2.0 interface (Type A)	82
	X101	+24 V NC power supply	64
	X500	HSCI output 1 (synchronized) to CC, PL, MB (not with MC 6x5x)	50
	X501	HSCI output 2, only to MB or PLB 6001 (not with MC 6x5x)	50
	X600	Reserved	
	X249	HDL interface for visual display unit	83
	X121	MC 6x4x: Profibus (option)	
	X125	SIK (System Identification Key)	
	X601	Reserved	
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

MC 6222



Pin layout	Connector	Function	Page
	X29	Reserved	
	X26, X116	Ethernet data interface	82
	X27	RS-232-C/V.24 data interface	
	X141, X142	USB 2.0 interface (Type A)	82
	X101	+24 V NC power supply	64
	X500	HSCI output 1 (synchronized) to CC, PL, MB (not with MC 6x5x)	50
	X501	HSCI output 2, only to MB or PLB 6001 (not with MC 6x5x)	50
	X600	Reserved	
	X601	Reserved	
	X116	Reserved	
	X121	MC 6222: Profibus (option)	
	X125	SIK (System Identification Key)	
	X3	Connection for screen soft keys	184
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

1.7.3 USB interface (USB 2.0)

X141, X142, X143, X144

Pin layout for USB connection (Type A):

USB connection (female) 4-pin	Pin layout
1	+5 V
2	USBP–
3	USBP+
4	GND

Note

If USB components that are connected to one of the USB ports require more than 0.5 A, a separate power supply becomes necessary for these components. One possibility is the USB hub (USB 2.0) from HEIDENHAIN (582 884 02).

If a USB hub is connected to one of the USB ports, the maximum permissible length of the USB cable (ID 624 775-xx) is reduced to 20 m.

USB hub

The power supply for the USB hub must comply with EN 50 178, 5.88 requirements for "low voltage electrical separation."

For more information on the USB interface, see the iTNC 530 Technical Manual.

1.7.4 Ethernet interface

X26, X116

- Maximum cable length:
  - Unshielded: 100 m
  - Shielded: 400 m
- Network topology: Star configuration

This means a hub serves as a central node that establishes the connection to the other participants.

Danger

The Ethernet interfaces of the MC 6xxx comply with the requirements of PELV ("low voltage electrical separation") according to EN 61800-5-1 and are powered internally by 24 V NC. All devices connected to these Ethernet interfaces must comply with the requirements of SELV or PELV according to EN 61800-5-1.

For more information on the Ethernet interface, see the iTNC 530 Technical Manual.

1.7.5 HDL interface

X249 Pin layout:

MC 6xxx, X249		Connecting cable ID 625 901-xx	BF 2xx, X2	
25-pin connection	**Pin layout**		**25-pin connection**	**Pin layout**
1	TD2+		1	TD2+
2	TD2–		2	TD2–
3	TD1+		3	TD1+
4	TD1–		4	TD1–
5	TD0+		5	TD0+
6	TD0–		6	TD0–
7	TC+		7	TC+
8	TC–		8	TC–
9	Do not assign		9	Do not assign
10	Do not assign		10	Do not assign
11	+ 5 V		11	Do not assign
12	TxD+		12	RxD+
13	TxD–		13	RxD–
14	TD2S		14	TD2S
15	Do not assign		15	EQSEL1
16	TD1S		16	TD1S
17	Do not assign		17	GND for EQSEL
18	TD0S		18	TD0S
19	Do not assign		19	EQSEL0
20	TCS		20	TCS
21	Do not assign		21	Do not assign
22	Do not assign		22	Do not assign
23	GND		23	GND
24	RxD+		24	TxD+
25	RxD–		25	TxD–

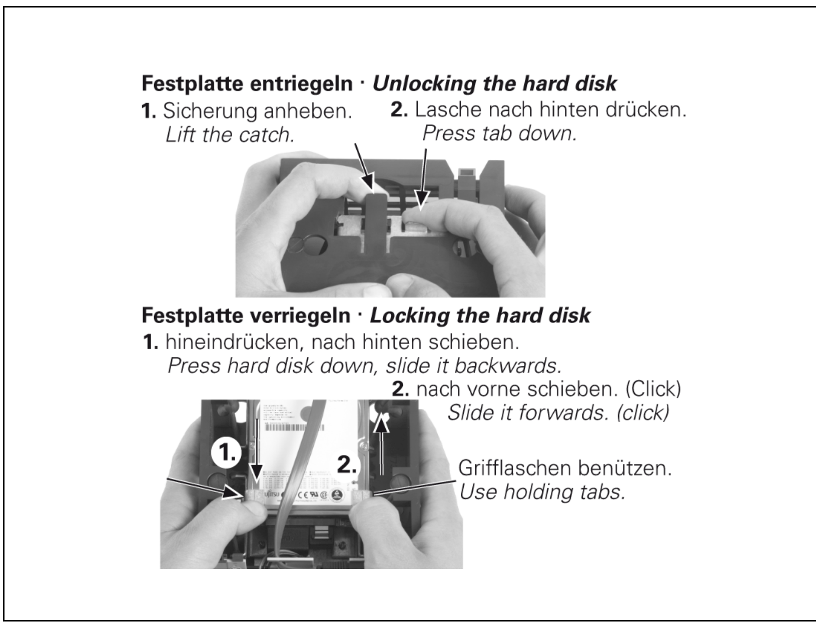
1.7.6 Handling the HDR hard disk

Shipping brace of the hard disk

The HDR hard disks of the MC 6x4x are equipped with a shipping brace. Before putting the iTNC 530 into service, the shipping brace of the hard disk must be removed.

> Attention
>
> Do not transport the HDR with the MC 6x4x after you have installed the HDR. If the entire machine is being transported, or the MC is being transported inside the electrical cabinet, the shipping brace for the hard disk is usually not required. However, if the possibility exists that the hard disk could be subject to increased shock or vibration loads, then you must remove the hard disk from the MC for transport, reinstall the shipping brace in the HDR, and send the HDR separately in the original packaging.

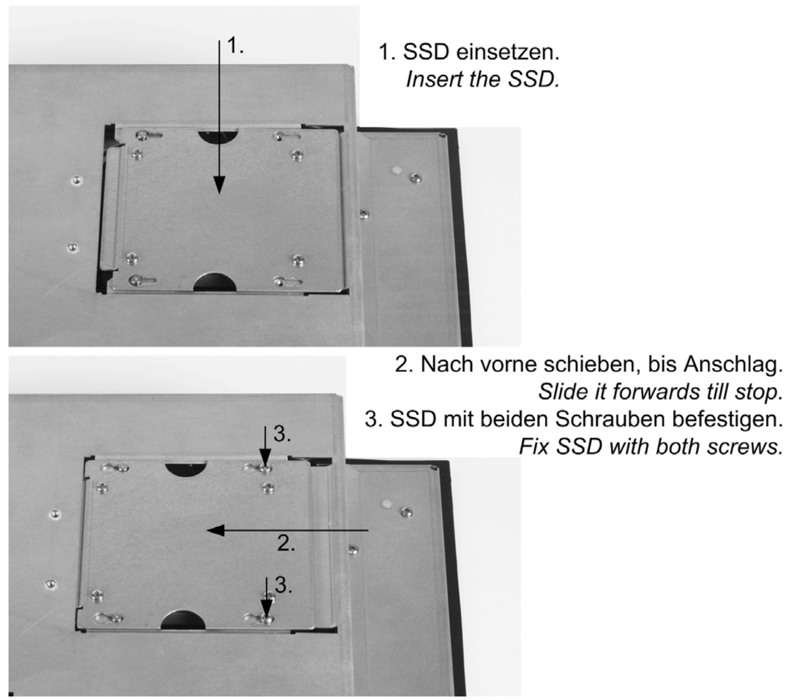
Should servicing become necessary (i.e. the HDR is being shipped on its own), the hard disk must be secured with the shipping brace.



> Note
>
> When handling the HDR disk and the SIK, ensure electrostatic discharge protection. Improper handling can result in damage to the components or assemblies due to ESD!

1.7.7 Mounting the Solid State Disk (SSD)

A special shipping brace is not required for the Solid State Disk (SSD). The SSD is to be mounted as follows:



> **Note**
>
> When handling the Solid State Disk (SSD), ensure electrostatic discharge protection. Improper handling can result in damage to the components or assemblies due to ESD!

1.7.8 Buffer battery

Note

Make a data backup before changing the buffer battery.

Danger

When exchanging the buffer battery, remember:

- Switch off the machine and the TNC.
- The buffer battery may be exchanged only by trained personnel.
- When exchanging the buffer battery, ensure electrostatic discharge protection. Improper handling can result in damage to the components or assemblies due to ESD!

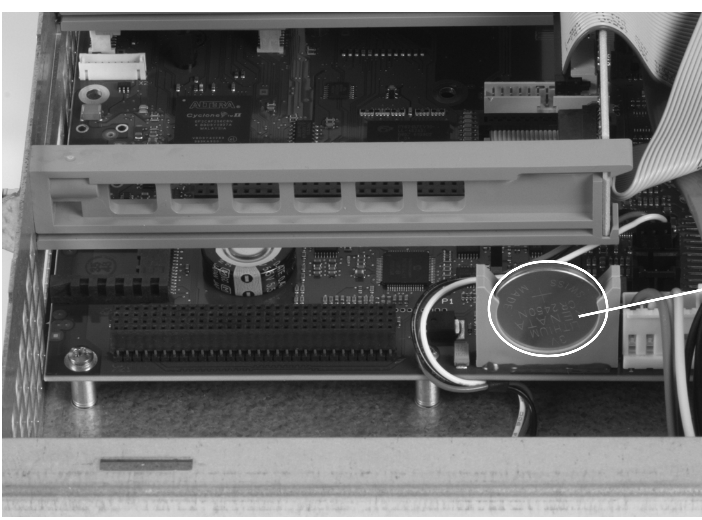
Battery type: 1 Lithium battery, type CR 2450N (Renata), ID 315 878-01

If the voltage of the buffer battery falls below 2.6 V, the error message **`Exchange buffer battery`** appears. If the voltage does not rise above 2.6 V again, the error message is reactivated after 30 minutes.

To exchange the battery:

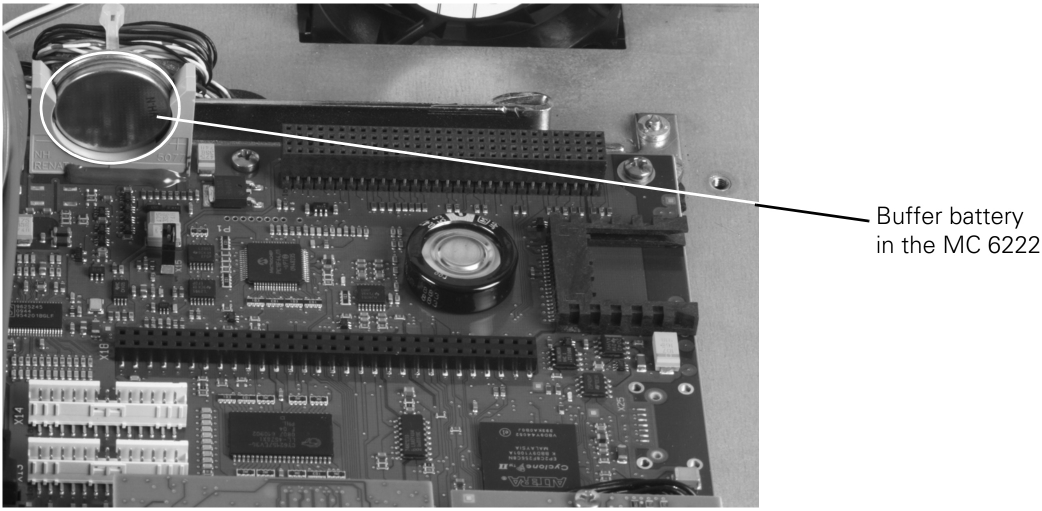
- The buffer battery is on the motherboard of the MC 6xxx.
- Loosen the screws securing the housing cover of the MC 6xxx.
- Remove the cover.
- The buffer battery is at the border of the PCB:

In the MC 6x41, it is next to the slot for the HDR:



Buffer battery in MC 6x41

In the MC 6222, it is at the rear close to the fans:

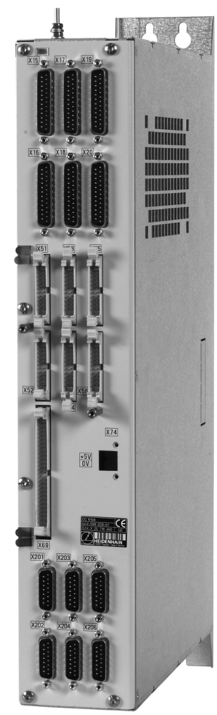


- Exchange the buffer battery. The new battery can be inserted in only one position.

1.8 CC 61xx Controller Unit

1.8.1 General information

<table>
<tr><th colspan="2">CC 6xxx controller unit</th></tr>
<tr><td>CC 6106 with 6 control loops
consists of:
1 drive-control motherboard and
2 drive-control expansion boards
It is equipped with:
■ 6 PWM outputs
■ 6 speed encoder inputs (1 Vpp or EnDat 2.2)
■ 6 position encoder inputs (1 Vpp or EnDat 2.2)
■ 2 SPI expansion slots
■ Power supply through UV(R) power supply unit
■ Weight: 4.1 kg

ID 662 636-xx</td><td></td></tr>
</table>

CC 6108 with 8 control loops consists of:
2 drive-control motherboards and
2 drive-control expansion boards
It is equipped with:

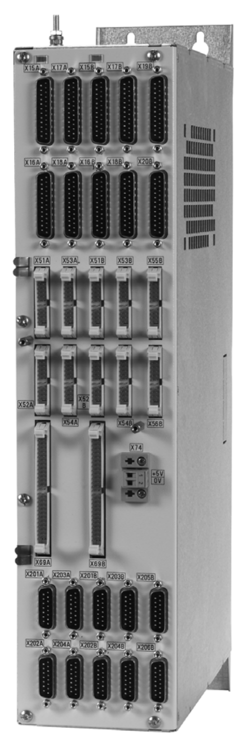
- 8 PWM outputs
- 8 speed encoder inputs (1 Vpp or EnDat 2.2)
- 8 position encoder inputs (1 Vpp or EnDat 2.2)
- 4 SPI expansion slots
- Power supply through UV(R) power supply unit
- Weight: 4.7 kg

ID 662 637-xx

CC 6110 with 10 control loops consists of:
2 drive-control motherboards and
3 drive-control expansion boards
It is equipped with:

- 10 PWM outputs
- 10 speed encoder inputs (1 Vpp or EnDat 2.2)
- 10 position encoder inputs (1 Vpp or EnDat 2.2)
- 4 SPI expansion slots
- Power supply through UV(R) power supply unit
- Weight: 4.8 kg

ID 662 638-xx



Difference between CC 61xx and CC 424

Function	CC 61xx	CC 424
Hardware	Position encoder inputs X201(A) to X206(A) and X201B to X206B on the CC 61xx	Position encoder inputs X201 to X206 and X207 to X210 on the CC 424
Assignment of speed encoder inputs to the PWM outputs	Permanent assignment, position encoder inputs are also permanently assigned. Always in one row, e.g. X16A, X51A and X202A are permanently assigned to each other.	Permanent assignment, MP112.x removed
MP108.x, MP109.x	Assignment of axes and spindles to drive-control motherboards	Not present
MP120.x, MP121.x	Assignment of axes and spindles to the speed outputs (X15 to X18, X15 to X20)	Assignment of axes and spindles to speed outputs
MP115.x, MP116.x	MP115.x and MP116.x are removed. They are replaced by MP 118.x for axes and MP119.x for spindles.	When using an MC 42x(B) without position encoder inputs, MP115.x is removed. Otherwise MP115.x applies to the position encoder inputs on the MC 42x(B); MP116.x applies to the position encoder inputs on the CC 424
Encoders with EnDat 2.2 interface	Encoders with EnDat 2.2 interface can be operated without analog encoder signals. The position is evaluated purely digitally via a serial data protocol.	Encoders can only be operated via analog encoder signals.
MP118.x, MP119.x	Axis-specific configuration of the position encoder inputs	Not present
PWM frequency	Same as CC 424, only the drive-control boards for which the same PWM frequency must be set have changed, see page 93. Only MP2182.x = 1 is not supported.	Can be set via MP2180.x (switchable during operation); the calculation of the current-controller cycle time must be adapted via MP2182.x
Control loops can be switched from single speed to double speed for higher controller performance	See "Structure and performance of the controller unit"on page 93.	
Control-loop cycle times (at 5000-Hz PWM frequency) (position/speed/current)	Single-speed: 200 µs/200 µs/100 µs Double-speed: 100 µs/100 µs/100 µs (with position encoder) 100 µs/100 µs/100 µs (without position encoder)	Single-speed: 200 µs/200 µs/100 µs Double-speed: 200 µs/100 µs/100 µs (with position encoder) 100 µs/100 µs/100 µs (without position encoder)

Function	CC 61xx	CC 424
Following error in the jerk phase (MP2606.x)	Same as CC 424	Typical input values: 0.001 to 0.005
Stick-slip friction compensation (MP2610.x, MP2612.x, MP2614.x)	Same as CC 424	Feed-rate independent; MP2610.x same meaning as previously (effective values, readjustment necessary), MP2612 has new meaning MP2614.x is new
Stick-slip friction compensation (MP1511.x, MP1512.x)	Same as CC 424	MP1511.x and MP1512.x can only be used for analog axes. MP2610.x, MP2612.x and MP2614.x should be used for digital axes.
Multiplication factor for k_V factor and kink point	Same as CC 424	MP1820.x and MP1830.x are only supported for analog axes.
Filter in the speed control loop	Same as CC 424	MP2530.x, MP2540.x, MP2550.x and MP2560.x removed, New machine parameters MP2542.x to MP2546.x, MP2552.x to MP2556.x, MP2562.x to MP2566.x, MP2572.x to MP2576.x, MP2560.x has new meaning
Master-slave torque control	In master-slave-torque mode, the PWM outputs of the master and slave axes must always be operated on the same DSP (meaning the same drive-control motherboard for the CC 61xx), i.e. the master and slave axes must be operated with the same power. More slave axes are possible for the CC 61xx than before, e.g. up to five slave axes are possible for a CC for six axes.	As of 340 422-06: The PWM outputs of the master and slave axes must always be operated on the same DSP ("Single speed" setting).
Gantry axes and master-slave-torque control	The PWM outputs of the master and slave axes in gantry mode of operation do not need to be operated on the same DSP (with CC 61xx, they can therefore be distributed to different CCs).	
Reading the absolute value of encoders with EnDat interface	Same as CC 424	The absolute value can be read out again via the **PASS OVER REFERENCE** soft key or via Module 9220 (i.e. after the exchange of milling heads).

Function	CC 61xx	CC 424
MP2220.x	Same as CC 424	Bit 4: Monitoring for excessive temperature Bit 5: Monitoring for insufficient temperature Bit 6: Reserved Bit 7: Monitoring of encoder input frequency Bit 8: Adjust mechanical offset by gradually increasing the k_V factor
MP2250.x, MP2252.x	Same as CC 424	Field angle determination for nonaligned encoders
MP7602	PLC cycle time is a multiple of the HSCI cycle time (3 ms). Minimum possible value is therefore 12 ms.	PLC cycle time [ms]
MP7600.0	Same as CC 424	MP7600.x removed, path interpolation fixed at 3 ms (does not influence the position controller cycle)
Display in internal oscilloscope and in TNCopt	Same as CC 424	Effective values
LIFTOFF	LIFTOFF can also be used in conjunction with non-regenerative HEIDENHAIN inverters.	LIFTOFF can only be used in conjunction with regenerative HEIDENHAIN inverters.

1.8.2 Structure and performance of the controller unit

Structure of the new CC 61xx and UEC 11x controller units

There is no backplane between the CC and MC for the CC 61xx and MC 6xxx. The CCs continue to be supplied via X69 and X74. The MCs of the new generation are supplied with +24-V NC voltage via X101, independently of the CCs.

A new DSP processor is used on the new CC 61xx and the UEC 11x. A single one of these DSPs can regulate up to six control loops, with the same controller performance as the CC 424 (B).

Note

Due to the new structure of the CC 61xx, not all of the axis configurations of the CC 424 can directly be implemented on the CC 61xx! Please note in particular the changes in the available double-speed control loops of a CC 61xx.

The CC 61xx family has a modular structure. The addressed DSP is on the motherboard of the controller. Every drive-control motherboard is an HSCI participant in the HSCI system. These drive-control motherboards have two control loops, and can therefore control two axes. The same applies to each drive-control expansion board, which does not have its own DSP, however. A drive-control expansion board has two control loops, but can only be used in combination with a drive-control motherboard (DSP).
As a result, a maximum of two drive-control expansion boards can be controlled by one drive-control motherboard (DSP). This configuration results in a maximum of six axes that a single DSP can control. If more than six axes are to be controlled, then another drive-control motherboard is necessary, providing another DSP. All axes that are controlled by a DSP belong to a controller group.

If two drive-control motherboards (two DSPs) are housed in one CC (this means two HSCI participants in one housing), then the inputs and outputs of the CC are uniquely assigned to one of the two controller groups (DSPs) via the letters A and B.
Index A means that these inputs and outputs are controlled by the first DSP (first drive-control motherboard). All inputs/outputs that are assigned to a DSP belong to a controller group (here: controller group A). Index B means that these inputs/outputs are controlled by the second DSP (second drive-control motherboard), and therefore belong to controller group B.

For configuring the controller units via machine parameters, the HSCI address of the respective drive-control motherboard must be entered in MP108.x or MP109.x . The HSCI address is obtained from the position of the drive-control motherboard in the HSCI system. The first drive-control motherboard after X500 of the MC is given the address 0, etc. However, the HSCI address to be entered only depends on the drive-control motherboards in the system. I/O units (PLs) and machine operating panels (MBs) are not taken into account. This means that for the first drive-control motherboard, you have to enter the address 0 in MP108, regardless of whether PLs or MBs are located before the CC in the HSCI chain.

There is one LED on each drive-control motherboard indicating its HSCI address relevant for MP108.x and MP109.x by a blink code.

Also, on the CC 61xx the inputs and outputs have permanent assignments to each other. Switching of the inputs and outputs, as with the CC 424, is not possible here.
Example:
X51, X15, X201 are permanently assigned to each other.
X53A, X17A, X203A are permanently assigned to each other.
etc.

See the table for the assignments. The assignment within each row is permanent. Switching between the rows is not possible with the CC 61xx.

Speed output: PWM output (CC 61xx / UEC 11x)	**Input: Speed encoder**	**Input: Position encoder**
X51(A/B) / X81	X15(A/B)	X201(A/B)
X52(A/B) / X82	X16(A/B)	X202(A/B)
X53(A/B) / X83	X17(A/B)	X203(A/B)
X54(A/B) / X84	X18(A/B)	X204(A/B)
X55(A/B) / X85	X19(A/B)	X205(A/B)
X56(A/B) / X86	X20(A/B)	X206(A/B)

Combination of several CC 61xx controller units

Two drive-control motherboards are housed in one CC 6108 or CC 6110. Both boards must be connected to the supply bus via the connectors X69A and X69B. A short ribbon cable (ID 325 816-15) with three connectors for X69 is therefore included with either of the two CCs. Two of the connectors are connected to X69A and X69B of the CC, respectively. The third connector of the cable serves to establish the connection to X69 of the power supply module. This cable must also be used to operate two CC 6106 controller units arranged next to each other.

However, if you want to arrange several CC 61xx controller units next to each other in one row, you need to replace the short ribbon cable by an additional accessory cable (ID 325 816-22) with five connectors. This cable is required for the following configurations: 2 x CC6108, 2 x CC 6110, CC 6110 with CC 6108 or 4 x CC 6106. The four connectors X69 of the CC are then connected via the additional accessory cable. The fifth connector of the cable is used to establish the connection to X69 of the power supply module.

Two drive-control motherboards are housed in one CC 6108 or CC 6110. The HSCI bus must be forwarded from one board to the other via the connectors X500A and X502B. A short HSCI cable (approx. 20 cm) is therefore included with either of the two CCs.

PWM frequencies with the CC 61xx

The same PWM frequency must be set for both PWM outputs of a drive-control motherboard or a drive-control expansion board in MP2180.x.

- Drive control board 1: X51, X52
- Drive control board 2: X53, X54
- Drive control board 3: X55, X56

PWM frequencies of the UEC 1xx

The same PWM frequency must be set for both PWM outputs of a drive-control motherboard or a drive-control expansion board in MP2180.x.

- Drive control board 1: X80, X81
- Drive control board 2: X82, X83
- Drive control board 3: X84

Use of several UV(R) power supply modules

As of software version 606 42x-0, it is basically possible to supply several CC 61xx units separately via several power supply units. The NC software determines the number of power modules existing in the system and finds out which power module is connected to which CC 61xx. The UV that is connected to the first CC in the HSCI chain is referred to below as the "first" power supply module. However, there are still the following constraints for this function in software version 606 42x-0:

- Plug & play is possible for power modules only if they are connected to the first power supply module
- Plug & play is possible only for the first of the power supply modules
- Error conditions of all UVs are recognized and lead to the necessary reactions. However, if several UVs are used, the error messages still apply globally to all of the UVs. In this software version it is therefore not possible to identify which UV has caused the error.

- MP2195 for suppressing the error messages from the UV does not yet apply to specific power supply modules, and the settings in MP2195 are globally effective for all UVs.
- When ERR.IZ.GR is sent, the spindle power limiting defined in MP2220.x works if the spindle is included in the increased DC-link voltage. This means that the power of the spindle is limited only if a spindle is connected to the UV that reports the error.
- The parameters in MP108/MP109 and MP2199 cannot be defined independently of each other. MP108/MP109 is used to assign the axes/spindles to the drive-control motherboards. MP 2199 is used to assign the axes/spindles to the power supply modules. All axes/spindles whose power modules are connected to a power supply module via the DC link may be assigned only to those drive-control motherboards that are connected to the same power supply module via the supply bus. This assignment must be remembered and taken into account when defining the parameters. If this assignment is not considered, a correct reaction to individual error messages and alarms is not possible and the power of the spindle cannot be limited in the event of ERR.IZ.GR.

Figure 1: Permissible parameterization

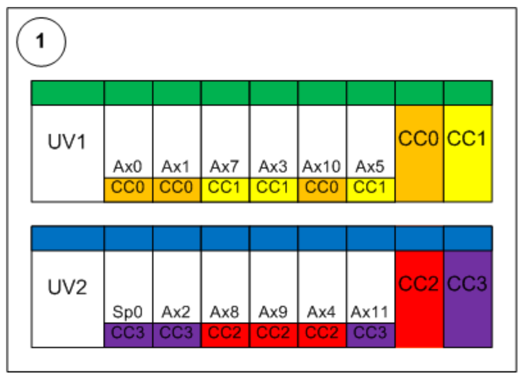
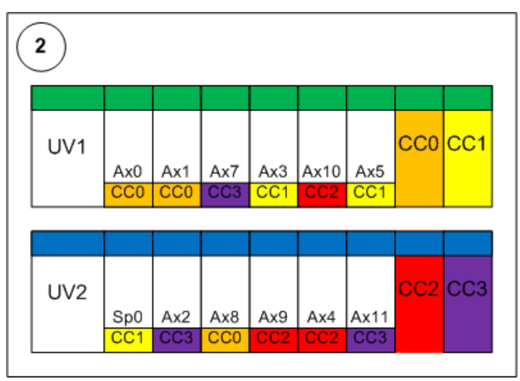


Figure 2: Impermissible parameterization



Performance of the CC 61xx

Depending on the required machine or controller performance, it may be necessary to drive an axis with increased computing power of the controller: this is then referred to as a double-speed axis. On the CC 61xx controllers, there is a DSP on every drive-control motherboard. The UEC 11x controller also has a DSP, with which it controls all the axes of the UEC 11x.

A DSP has sufficient computing power to control up to six single-speed axes or three double-speed axes. A double-speed axis requires the computing power of two single-speed axes. This enables you, for example, to achieve excellent results regarding the workpiece surface quality or speed optimization.

Single-speed control loops are used for:

- Spindles
- Conventional axes

With the CC 61xx, single-speed control loops can normally also be used for:

- Linear motors
- Torque motors

Double-speed control loops are used for:

- High-frequency spindles
- "Axes that are difficult to control"

The Technical Manual of your control describes the conditions under which a double-speed control loop must be used. All control loops are defined as single-speed control loops in the software default settings.

With the CC 61xx controller generation, single-speed or double-speed axes are no longer configured on the specific DSP (CC 424), but individually in MP2000.x.

Note

CC 61xx double-speed control loops require software option 49!

With the CC 61xx, it is possible to distribute double-speed axes controlled by one DSP over any of the outputs of the controller groups. The only restriction is the maximum controller performance of the parent DSP, which must not be exceeded. For example, with a CC 6106 up to any three outputs may be used for double-speed axes.

Double-speed control loops are needed particularly for linear drives in order to achieve proper speed/position control factors required by high-efficiency mechanical systems.

High-speed spindle motors requiring PWM frequencies greater than 5 kHz with short current controller cycle times can be operated only with double-speed control loops.

Please note that the maximum number of available control loops may be reduced through the configuration of double-speed control loops.

This leads to the following possible configurations for a CC 6106:

Composition of the CC 6106 controller unit:

Controller group (one DSP) for up to six single-speed (SS) axes or three double-speed (DS) axes					
Drive-control motherboard (RB)		Drive-control expansion board 1 (RE_1)		Drive-control expansion board 2 (RE_2)	
X51 (SS)	X52 (SS)	X53 (SS)	X54 (SS)	X55 (SS)	X56 (SS)

Possible configurations for the CC 6106:

Number of single-speed axes:	**Number of double-speed axes:**
6 single-speed	0 double-speed
4 single-speed	1 double-speed
2 single-speed	2 double-speed
0 single-speed	3 double-speed

Composition of the CC 6108 controller unit:

Controller group A (one DSP) for max. four SS or three DS				**Controller group B (one DSP) for max. four SS or three DS**			
CB A		CE 1		CB B		CE 1	
X51A (SS)	X52A (SS)	X53A (SS)	X54A (SS)	X51B (SS)	X52B (SS)	X53B (SS)	X54B (SS)

Possible configurations for the CC 6108, each for controller group A and B:

Number of single-speed axes:	**Number of double-speed axes:**
4 single-speed	0 double-speed
3 single-speed	1 double-speed
2 single-speed	2 double-speed
1 single-speed	2 double-speed
0 single-speed	3 double-speed

Composition of the CC 6110 controller unit:

Controller group A (one DSP) for max. four SS				**Controller group B (one DSP) max. six SS**					
CB A		CE 1		CB B		RE 1		RE 2	
X51A (SS)	X52A (SS)	X53A (SS)	X54A (SS)	X51B (SS)	X52B (SS)	X53B (SS)	X54B (SS)	X55B (SS)	X56B (SS)

Possible configurations for the CC 6110 for controller group A:

Number of single-speed axes:	**Number of double-speed axes:**
4 single-speed	0 double-speed
3 single-speed	1 double-speed
2 single-speed	2 double-speed
1 single-speed	2 double-speed
0 single-speed	3 double-speed

Possible configurations for the CC 6110 for controller group B:

Number of single-speed axes:	**Number of double-speed axes:**
6 single-speed	0 double-speed
5 single-speed	0 double-speed
4 single-speed	1 double-speed
3 single-speed	1 double-speed
2 single-speed	2 double-speed
1 single-speed	2 double-speed
0 single-speed	3 double-speed

Composition of the UEC 11x:

DSP on drive-control motherboard for up to five single-speed (SS) axes					
Controller group 1 (on drive-control motherboard RB)		Controller group 2 (drive-control expansion board 1 CE 1)		Controller group 3 (drive-control expansion board 2 CE 2)	
X80 (SS) Spindle	X81 (SS) Axis	X82 (SS) Axis	X83 (SS) Axis	X84 (SS) Axis	Vacant

Useful configurations of UEC 11x:

Number of single-speed axes:	**Number of double-speed axes:**
5 single-speed	0 double-speed
4 single-speed (X80, X81, X82, X83)	1 double-speed (X84)
3 single-speed axes (X82, X83, X84)	1 double-speed spindle (X80)

1.8.3 New functions of the CC 61xx

- Through a changed algorithm, the CC 61xx achieves a lower drive utilization for synchronous and linear motors, provided that the stall current (I-0) is higher than the rated current (I-N) of the motor in the motor table. As a result, the use of automatic feed control (AFC) attains higher milling power.
  - Compatibility mode:
    If AFC reference values that were recorded with the CC 424(B) or CC 422 are to be used with the CC 61xx, then you can switch to the old algorithm over MP2221 bit#10.
- With the CC 61xx and the UEC 1xx, the LIFTOFF function can also be performed if non-regenerative UV power supply modules are being used. In this case, the controller unit itself must measure the Uz dc-link voltage. This results in the following voltage thresholds for LIFTOFF:
  At a voltage of 400 V the initialization of LIFTOFF is reported to the MC and the spindle is retracted from the workpiece, at a voltage of 350 V the spindle is decelerated systematically in order to maintain the dc-link voltage, at a voltage of 300 V the powerfail signal is set by the controller unit.
  The 24-V PLC power supply must be maintained for a least a second. HEIDENHAIN offers the PSL 13x or CML 110 (Capacitor Module for Low Voltage) for this.

Expanded current controller bandwidth (only CC 61xx)

For axes with rigidly connected mechanics it may make sense to adjust the current control loop using the maximum possible bandwidth to attain high gains in the superimposed velocity control loop. For the CC 61xx controller unit there are two procedures for increasing the bandwidth of the current control loop:

- Using machine parameter MP2450 to increase the controller factors in the current control loop,
- and proportionally differentiating feedforward (D feedforward) to optimize the command action of the current control loop (MP2440).

Increased controller factors in the current control loop:

The dead time in the current control loop is the decisive factor for the attainable controller factors, and therefore for the attainable bandwidth. With MP2450, you can activate compensation of this dead time. After activation, you have to repeat the current controller adjustment. Usually you can then set considerably higher current controller factors. The higher the controller factors, the larger the bandwidth of the current controller. The bandwidth of the current control loop can be evaluated with the Bode diagram function of the TNCopt commissioning and diagnostic software (see TNCopt User's Manual).

However, the noise in the current—and as a result, the loudness of the drive—increase when the current controller bandwidth is increased.

D feedforward for optimization of the current controller command action:

With **MP2440**, you activate feedforward with proportionally differentiating behavior for the torque-generating current. This way you achieve an optimized command action of the current controller.

The parameter **MP2440** indicates the cutoff frequency in Hertz [Hz] up to which feedforward is active. When this cutoff frequency is reached, feedforward is deactivated by a low-pass element. It is therefore also referred to as PDT_1 feedforward in the following.

The following table shows the maximum possible cutoff frequency depending on the PWM frequency of the drive. The higher the cutoff frequency, the larger the bandwidth of the current controller regarding its command action. At the same time, however, the noise in the current—and as a result, the loudness of the drive— increase. If you enter 0 Hz, feedforward is deactivated.

Maximum values for parameter **MP2440** as a function of the PWM frequency:

PWM frequency	Maximum value for MP_MP2440
3333 Hz	800 Hz
4000 Hz	960 Hz
5000 Hz	1200 Hz
6666 Hz	1600 Hz
8000 Hz	1920 Hz
10000 Hz	2400 Hz

Adjustment of the current controller if increased current controller factors are active over MP2450:

Note

For standard drives, it is usually not necessary to increase the current controller bandwidth. In general, the increase is only useful if the drives are connected directly without an interconnected gear.

If PDT_1 feedforward or the increase of the current controller factors is activated, you must keep in mind that the noise in the current increases when the bandwidth is increased. You have to find a compromise between the

- increase of the current controller bandwidth and
- the development of noise in the drive.

To adjust the increased current controller factors, proceed as follows:

- To activate the function, enter MP2450 = 1 for the corresponding parameter of the axis.
- Readjust the current controller with TNCopt (Bode diagram function; see TNCopt User's Manual).

Note

If you activate the PDT_1 feedforward by entering a value in parameter **MP2440**, it is not necessary to readjust the current controller. The PDT_1 feedforward does not have a noticeable effect on the current control loop but rather only on the superimposed speed control loop. For axes with rigidly connected mechanics, you attain higher controller factors in the speed control loop if the PDT_1 feedforward is activated.

EnDat 2.2 encoders

Encoders with EnDat 2.2 interface can be operated without analog encoder signals. The position is evaluated purely digitally via a serial data protocol.

For **motor encoders with EnDat 2.2 interface,** the control requires the following values for calculating the signal period:

> Note
>
> Encoders with EnDat 2.2 interface do not supply any incremental signals. Only a purely digital measured value is transferred to the control. The number of increments per encoder revolution (MP332.x) must therefore be set to the value "1".

The controller unit assumes that the rotary encoder has only one line per encoder revolution. Enter the following data:

- M331.x: For each axis, enter the distance traversed per motor revolution.
- MP332.x: Enter the value "1".
- MP1054.x: For each axis, enter the distance traversed per motor revolution.
- Enter the value "1" in the **STR** column (line count of the motor encoder) of the motor table.

> Note
>
> Large input values in MP331.x and MP332.x cannot be read by the PLC!

> Note
>
> HEIDENHAIN recommends:
>
> The optimum manufacturing precision can only be attained with linear encoders. A control without linear encoders finds the axis position through the pitch of the ball screw. The problem is, the ball screw gets hot during machining and expands. And the result is a position measurement that deviates from the actual position, causing error. With linear encoders, the control always determines the correct slide position. This means that your machine positions its axes with constantly high precision.
>
> More information is available on the Internet at:
>
> http://www.heidenhain-shows-the-way.eu

1.8.4 Specifications

Machine interfacing		iTNC 530
Regulation with CC 6xxx / UEC 1xx		
	Position control resolution	$\frac{\text{Signal period}}{4096}$ or encoder resolution (EnDat 2.2 interpol.)
	Path interpolation	3ms
	Fine interpolation	Cycle time for fine interpolation = Cycle time of position controller
	Cycle time of current controller	PWM frequency — Cycle time if MP2182.x = 0: 3333 Hz — 150 µs 4000 Hz — 120 µs 5000 Hz — 100 µs 6666 Hz — 75 µs (option 49) 8000 Hz — 60 µs (option 49) 10000 Hz — 50 µs (option 49)
	Cycle time of speed controller	■ Speed controller cycle time = 2 · current controller cycle time ■ Unless PWM frequency ≤ 5 kHz and double-speed performance, then: Speed controller cycle time = Current controller cycle time
	Cycle time of position controller	Position controller cycle time = Speed controller cycle time

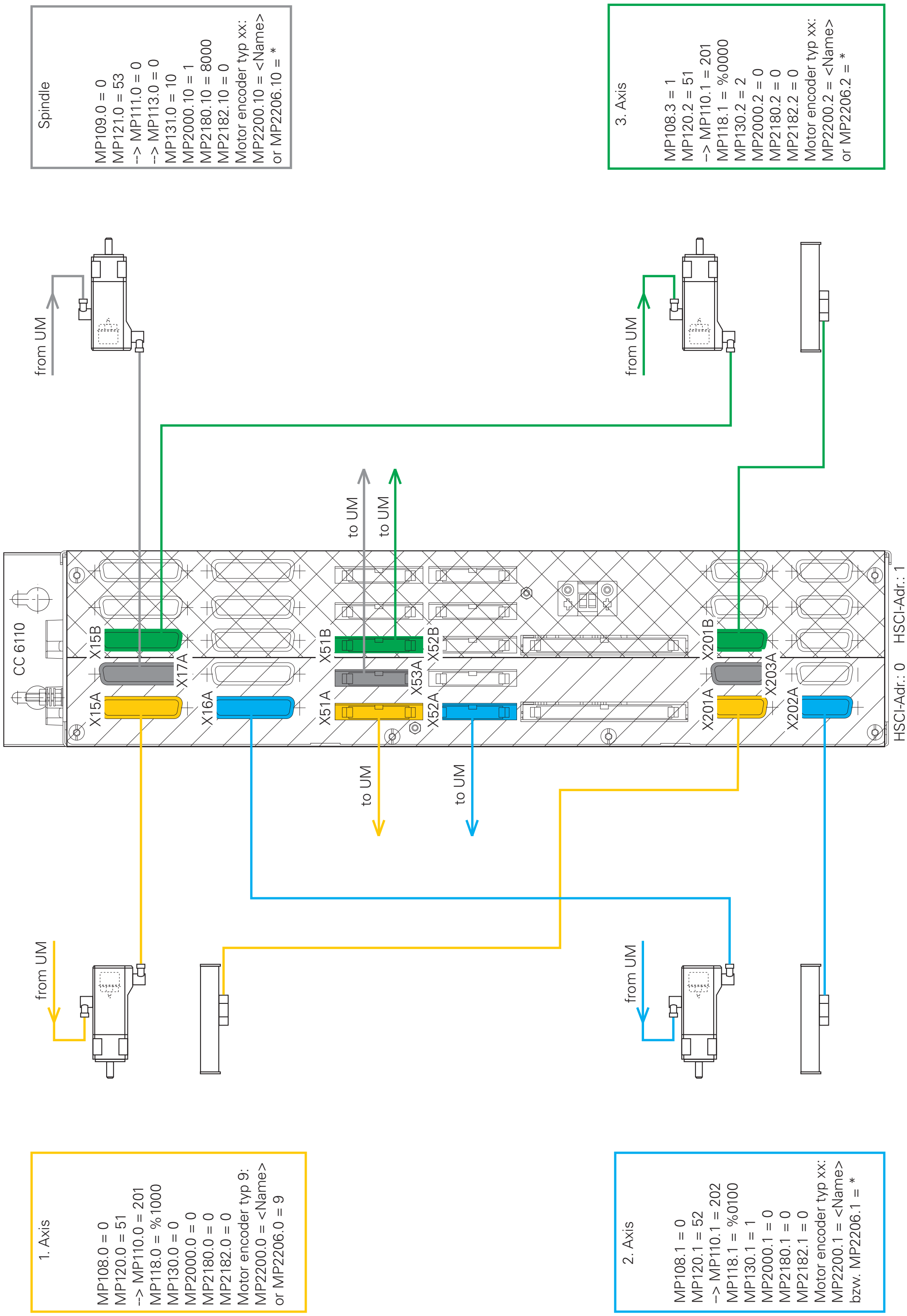
Machine interfacing	**iTNC 530**
Maximum motor speed	$n_{max} = \frac{f_{PWM} \cdot 60000\ min^{-1}}{p \cdot 5000\ Hz}$ n_{max}: Maximum motor speed [min^{-1}] f_{PWM}: PWM frequency [Hz] p: Number of pole pairs The following PWM frequencies are available: 3333 Hz, 4000 Hz, 5000 Hz With option 49: 6666 Hz, 8000 Hz, 10 000 Hz
or by way of:	Maximum signal frequency of motor encoder = 400 kHz or 800 kHz (depending on MP2561.x bit#0) Calculation of maximum motor speed: $n_{max} = \frac{f_{max} \cdot 60000\ [s/min]}{ELC}$ n_{max}: Maximum motor speed [min^{-1}] f_{max}: Maximum signal frequency of motor encoder [kHz] ELC: Encoder line count Example 1 $n_{max} = \frac{400kHz \cdot 60000\ [s/min]}{2048}$ nmax = 11718.75 min^{-1} Example 2 $n_{max} = \frac{800\ kHz \cdot 60000\ [s/min]}{2048}$ nmax = 23437.5 min^{-1}
Maximum feed rate by way of:	Maximum signal frequency of the position encoder: ■ At 1 Vpp: 27 kHz or 400 kHz ■ At 11 μA_{PP}: 27 kHz or 140 kHz Calculation of the maximum feed rate Fmax: $F_{max} = SP \cdot f_{max} \cdot 60^{-3} \left[\frac{m \cdot s}{\mu m \cdot min}\right]$ F_{max}: Maximum feed rate [m/min] f_{max}: Maximum signal frequency of position encoder [kHz] SP: Signal period of the position encoder [kHz] Example: $F_{max} = 20\ um \cdot 400\ kHz \cdot 60^{-3} \left[\frac{m \cdot s}{\mu m \cdot min}\right]$ Fmax = 480 m/min

1.8.5 Example configuration

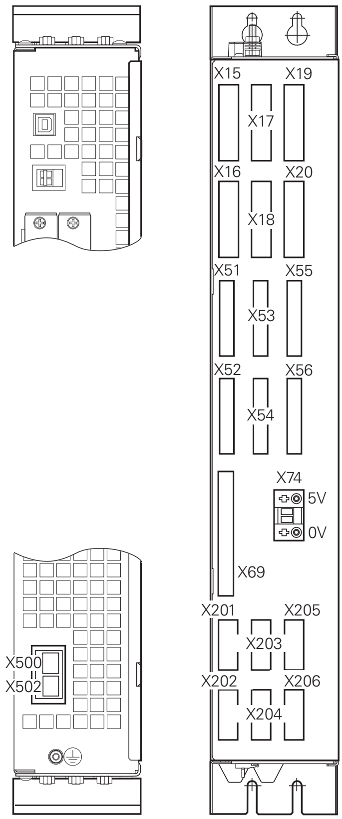
The machine tool consists of 3 axes plus spindle. The following conditions apply:

- Spindle:
  - High-performance spindle
  - 8 kHz PWM frequency
- 1st axis
  - EnDat 2.2 speed encoder
  - 5 kHz PWM frequency
  - EnDat 2.2 position encoder
- 2nd axis:
  - EnDat 2.1 speed encoder
  - 5 kHz PWM frequency
  - 1 Vpp position encoder
- 3rd axis:
  - 1 Vpp speed encoder
  - 5 kHz PWM frequency
  - No position encoder

Connection overview and example MP configuration of the CC 6110:

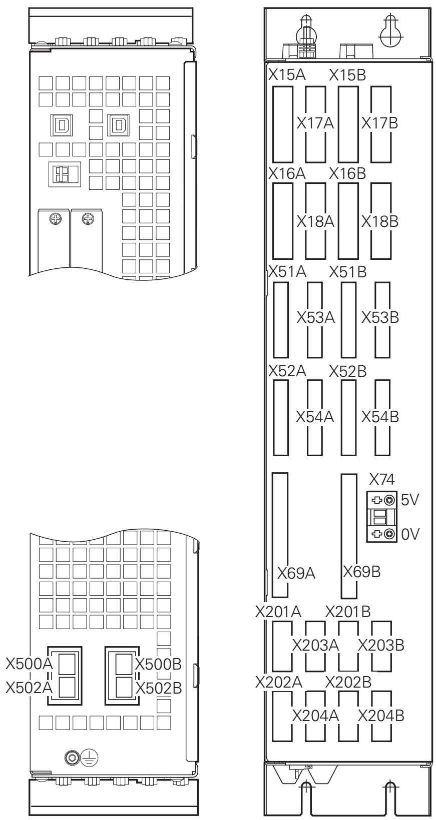


1.8.6 Connection overview

CC 6106, controller unit with 6 control loops and HSCI interface			
Pin layout	**Connector**	**Function**	**Page**
	X15 to X20	Speed encoder	
	X51 to X56	PWM output	
	X69	Supply bus	
	X201 to X206	Position encoder	
	X500	HSCI output	50
	X502	HSCI input	50
	–	SPI slot 1 (on bottom, reserved for expansion modules)	
	–	SPI slot 2 (on bottom, reserved for expansion modules)	
	X74	+ 5 V supply	112
	X7	Bridge for signal ground (on bottom)	112
	⏚	Protective ground	

> **Attention**
>
> Do not engage or disengage any connecting elements while the unit is under power!

CC 6108 controller unit with 8 control loops and HSCI interface

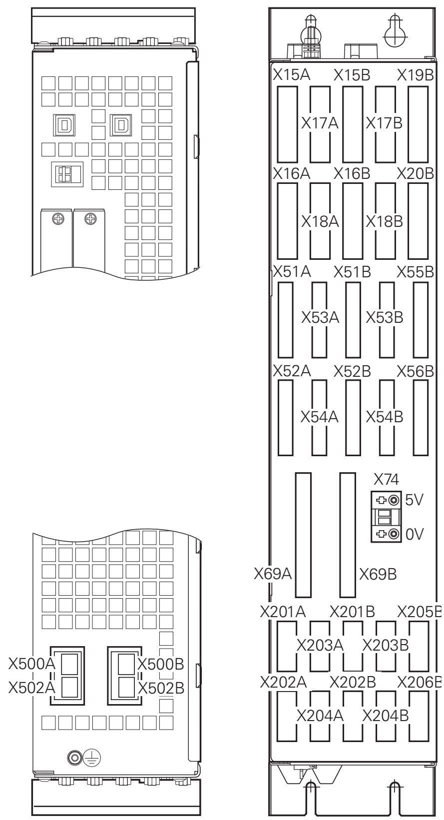


Pin layout	Connector	Function	Page
	X15A - X18A	Speed encoder Drive-control motherboard A	
	X15B - X18B	Speed encoder Drive-control motherboard B	
	X51A - X54A	PWM output Drive-control motherboard A	
	X51B - X54B	PWM output Drive-control motherboard B	
	X69A	Supply bus Drive-control motherboard A	
	X69B	Supply bus Drive-control motherboard B	
	X201A - X204A	Position encoder Drive-control motherboard A	
	X201B - X204B	Position encoder Drive-control motherboard B	
	X500A	HSCI output Drive-control motherboard A	50
	X502A	HSCI input Drive-control motherboard A	50
	X500B	HSCI output Drive-control motherboard B	50
	X502B	HSCI input Drive-control motherboard B	50
	X74	+ 5 V supply	112
	–	SPI slot 1 (on bottom, reserved for expansion modules)	
	–	SPI slot 2 (on bottom, reserved for expansion modules)	
	X7	Bridge for signal ground (on bottom)	112
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

CC 6108 / CC 6110 controller unit with 8 / 10 control loops and HSCI interface



Pin layout	Connector	Function	Page
	X15A - X18A	Speed encoder Drive-control motherboard A	
	X15B - X20B	Speed encoder Drive-control motherboard B	
	X51A - X54A	PWM output Drive-control motherboard A	
	X51B - X56B	PWM output Drive-control motherboard B	
	X69A	Supply bus Drive-control motherboard A	
	X69B	Supply bus Drive-control motherboard B	
	X201A - X204A	Position encoder Drive-control motherboard A	
	X201B - X206B	Position encoder Drive-control motherboard B	
	X500A	HSCI output Drive-control motherboard A	50
	X502A	HSCI input Drive-control motherboard A	50
	X500B	HSCI output Drive-control motherboard B	50
	X502B	HSCI input Drive-control motherboard B	50
	X74	+ 5 V supply	112
	–	SPI slot 1 (on bottom, reserved for expansion modules)	
	–	SPI slot 2 (on bottom, reserved for expansion modules)	
	X7	Bridge for signal ground (on bottom)	112
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

X74: +5 V supply

Connecting terminal X74	Pin layout
1	+5 V from the UV, UE supply module (X74)
2	0 V

Attention

The +5 V supply via X74 from the supply module is mandatory for the CC 61xx!

If the system includes several CCs 61xx that are connected to the same supply bus (X69) via a UV supply module, only the last CC 61xx (usually the unit at the extreme right) must be connected to the UV via X74. The other CC 61xx units are then supplied via supply bus X69.

If several CC 61xx units are supplied by more than one UV supply module, which means that they are connected to different supply bus systems (X69), then the last CC 61xx (usually the unit at the extreme right) of the respective supply bus must also be supplied with the additional +5 V of the UV via X74.

Please check whether the +5 V supply of all drive control motherboards is ensured when initially configuring the control. The voltage is displayed in the DriveDiag diagnosis tool. On the "Voltages and currents" tab for the drive control boards, you will find the +5 V supply voltage. The value of this voltage should not be below +4.90 V.

X7: Bridge for signal ground

Connecting terminal X7	Pin layout
1	Connection for signal ground
2	Connection on housing

In shipping condition of the CC 61xx, the signal ground (pin 1) is connected to the housing (pin 2) over an external bridge. If only one CC 61xx is in the system, it ensures the correct signal-ground connection.

If there are two or more CC 61xx units in the system that are connected over a UV power module to the same supply bus (X69), this external signal-ground bridge can stay connected with only one CC 61xx. Disengage these bridges on all other CCs in order to prevent ground loops.

If there are two or more CC 61xx units powered over more than one UV power module and are therefore connected with different supply bus systems (X69), then the external signal-ground bridge is to be connected with only one each CC 61xx of the respective supply bus. In order to prevent ground loops, disengage this bridge for the signal ground on all other CC 61xx units that are on a common supply bus (X69).

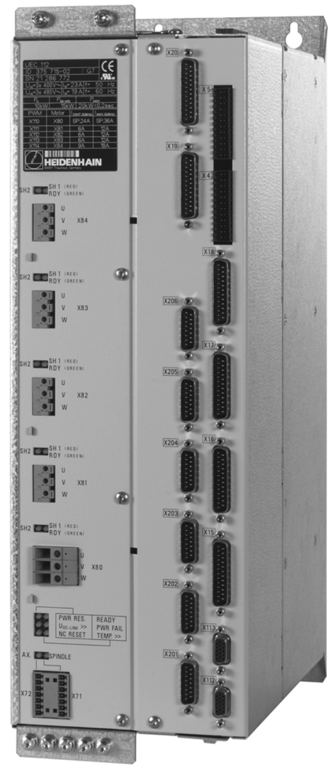
1.9 UEC 11x Controller Unit with Integrated Inverter and PLC

<table>
<tr><td colspan="2">UEC 11x</td></tr>
<tr><td>
Controller unit with integrated inverter and PLC for up to 5 control loops. Compact unit for machines with limited number of axes and low power demands.

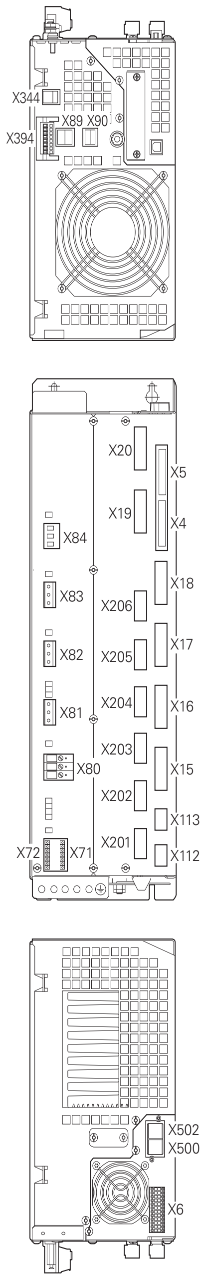
It is equipped with:

■ HSCI interface
■ 4 (UEC 111) or 5 (UEC 112) speed encoder inputs
■ 4 (UEC 111) or 5 (UEC 112) position encoder inputs
■ Connection for 3 axes plus spindle (UEC 111) or
Connection for 4 axes plus spindle (UEC 112)
■ Braking resistor
■ 38 PLC inputs, 23 PLC outputs (expandable via PL 61xx)
■ Integrated power supply unit 24 V NC / 3.5 A for supplying the HSCI components

UEC 111 with 4 control loops
Without FS: ID 625 777-xx

UEC 112 with 5 control loops
Without FS: ID 625 779-xx
</td><td>


</td></tr>
</table>

UEC 11x, controller unit with integrated inverter and PLC			
Pin layout	**Connector**	**Function**	**Page**
	X4, X5	PLC inputs	142
	X6	PLC outputs	144
	X15 to X19	Speed encoder	
	X31	Supply voltage for UEC 11x (3 x 400 V ± 10%)	116
	X71	Spindle safety relay (pulse inhibitor for spindle)	118
	X72	Axes safety relay (pulse inhibitor for axes)	118
	X80	Motor connection for spindle (24 A rated current at 3.3 kHz)	118
	X81	Motor connection axis 1 (6 A rated current at 3.3 kHz)	118
	X82	Motor connection axis 2 (6 A rated current at 3.3 kHz)	118
	X83	Motor connection axis 3 (9 A rated current at 3.3 kHz)	118
	X84	Motor connection axis 4 (6 A rated current at 3.3 kHz)	118
	X89	Braking resistor	119
	X90	24-V– NC output / 3.5 A	119
	X112	TS touch-trigger probe	131
	X113	TT touch-trigger probe	131
	X201 to X205	Position encoder	
	X500	HSCI output	50
	X502	HSCI input	50
	⏚	Protective ground M5	–

Attention

Do not engage or disengage any connecting elements while the unit is under power!

General information

Number of available control loops

- UEC 111: 4 control loops
- UEC 112: 5 control loops

Specifications:

Specifications	UEC 112			UEC 111		
	3 axes	**1 axis**	**Spindle**	**2 axes**	**1 axis**	**Spindle**
Supply voltage	3 x 400 V~ ± 10% (50 Hz to 60 Hz)					
DC-link voltage	565 V– (with supply voltage of 400 V)					
Power loss	Approx. 450 W			Approx. 450 W		
Rated current at a PWM frequency of 3333 Hz	6.0 A	9.0 A	24.0 A	6.0 A	9.0 A	24.0 A
4000 Hz	5.5 A	8.3 A	22.0 A	5.5 A	8.3 A	22.0 A
5000 Hz	5.0 A	7.5 A	20.0 A	5.0 A	7.5 A	20.0 A
6666 Hz	4.2 A	6.3 A	16.8 A	4.2 A	6.3 A	16.8 A
8000 Hz	3.6 A	5.5 A	14.6 A	3.6 A	5.5 A	14.6 A
10000 Hz	3.0 A	3.0 A	12.2 A	3.0 A	3.0 A	12.2 A
Peak power 6-40%[a]	18 kW			18 kW		
DC-link power	14 kW			14 kW		
Maximum current[b] at a PWM frequency of 3333 Hz	12.0 A	18.0 A	36.0 A	12.0 A	18.0 A	36.0 A
4000 Hz	11.0 A	16.5 A	33.0 A	11.0 A	16.5 A	33.0 A
5000 Hz	10.0 A	15.0 A	30.0 A	10.0 A	15.0 A	30.0 A
6666 Hz	8.4 A	12.6 A	25.2 A	8.4 A	12.6 A	25.2 A
8000 Hz	7.3 A	11.0 A	21.9 A	7.3 A	11.0 A	21.9 A
10000 Hz	6.0 A	6.0 A	18.3 A	6.0 A	6.0 A	18.3 A
Integral braking resistor						
Continuous power	2.1 kW			2.1 kW		
Peak power[c]	27 kW			27 kW		
Resistance	18 ohms			18 ohms		
Load capacity (+24 V NC)	3.5 A			3.5 A		
Weight	Approx. 20 kg			Approx. 20 kg		

a. Spindle: 40% cyclic duration factor for duty cycle time of 10 minutes (S6-40%)
b. Axes: 0.2 s cyclic duration factor for duty cycle time of 10 s with 70% rated current preload
Spindle: 10 s cyclic duration factor for duty cycle time of 60 s with 70% rated current preload
c. 1.5 % cyclic duration factor for duration of 120 s

X31: UEC power supply

Danger

Danger of electrical shock!
The UEC 11x controller unit must be opened only by HEIDENHAIN service engineers.
Do not engage or disengage any terminals while they are under power.

Note

IEC 61800-5-1 requires a non-detachable connection to the line power supply.

Note

If the power supply is other than 400 V, an autotransformer is required. It must comply at least with the connection specifications of the subsequent compact inverter.

With a power supply of 400 V, the inverter voltage U_Z is 565 V–, and with a power supply of 480 V it is 678 V–.

For information on the power connection, refer to the Technical Manual for "Inverter Systems and Motors."

Connecting terminals	UEC111, UEC 112
Operation with 400 V~	
L1	400 V~ ± 10% 50 Hz to 60 Hz
L2	
L3	
	Cable / single conductor (HT wire): 6 mm^2 (AWG 10) Single conductor H07 V2-K: 4 mm^2 (AWG 10) Line fuse: 25 A (gR) Siemens Sitor type Grounding terminal: ≥ 10 mm^2 (AWG 6)
	Tightening torque for connecting terminals: 0.7 Nm (6.5 to 7 psi)
Operation with 480 V~	
L1	480 V~ ± 10% 50 Hz to 60 Hz
L2	
L3	
	Cable / single conductor (HT wire): 6 mm^2 (AWG 10) Single conductor H07 V2-K: 4 mm^2 (AWG 10) Line fuse: 25 A (gR) Siemens Sitor type Grounding terminal: ≥ 10 mm^2 (AWG 6)
	Tightening torque for connecting terminals: 0.7 Nm (6.5 to 7 psi)

X80: Spindle motor
X81: Axis motor 1
X82: Axis motor 2
X83: Axis motor 3
X84: Axis motor 4

Connection:

Connecting terminals	Pin layout
U	Motor connection U
V	Motor connection V
W	Motor connection W

For information on synchronous motors, asynchronous motors and power cables, refer to the Technical Manual on Inverter Systems and Motors, "Motors for Axis and Spindle Drives" chapter.

X71: Safety relay for spindle
X72: Safety relay for axes

For information on the wiring and function, see the Basic Circuit Diagram for your control.

Connecting terminals X71 to X72	Pin layout
1	+24 V output (max. 250 mA)
2	0 V
3	+24 V input for U_Z ON, Axis ON, Spindle ON
4	Do not assign
5	Do not assign
6[a]	Normally closed contact (OE1, OE1A or OE1S)
7[a]	Normally closed contact (OE2, OE2A or OE2S)

a. Max. 125 V

Attention

A recovery diode is required in the proximity of inductive loads, e.g. relay or contactor coils.

X89: Braking resistor

Connection at the UEC 11x:

Connecting terminal X89 UE 11x	Pin layout	PW 21x	PW 1x0(B); connecting terminal X1
1	$+U_Z$	RB1	1
2	Switch to $–U_Z$	RB2	2

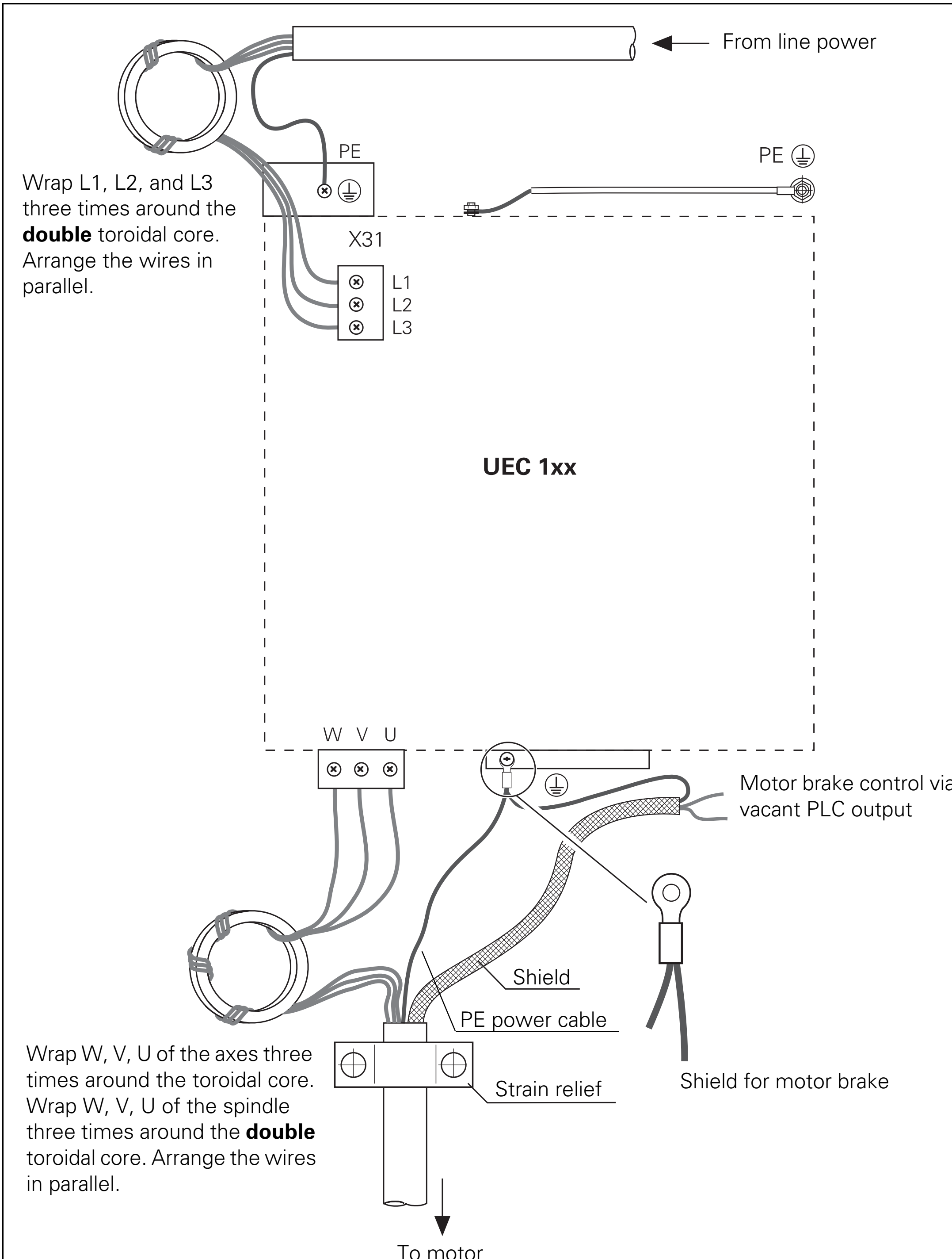
X90: 24 V output

Connecting terminal X90	Pin layout
+	+24 V (max. 3.5 A)
–	0 V

Device	Load capacity of 24 V NC supply (X90)
UEC 11x	3.5 A

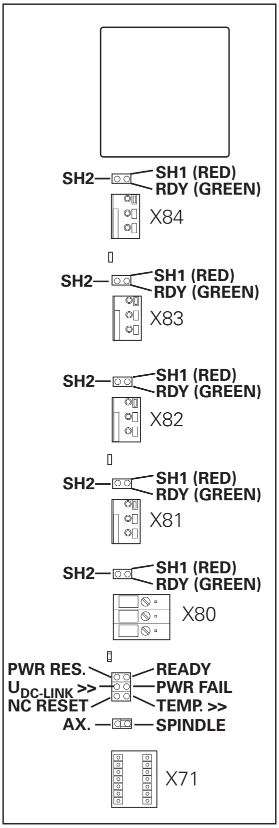
Mounting the toroidal cores

To suppress occurrence of conducted interference, toroidal cores must be mounted in the motor leads (X80 to X84), in the voltage supply lead (X31) and in the lead to the optional, external braking resistor (X89).



1.9.1 Meaning of the LEDs

On the front of the UEC 11x are several LEDs for functional control, with the following meaning:

UEC 11x	LED	Meaning	Signal direction	Signal
	NC RESET	Reset signal from the MC unit to the UEC	MC → UEC	$\overline{\text{RES.LE}}$
	PWR FAIL	U_Z too low, U_Z < 410 V (e.g. failure of a phase under load, power < 290 V)	UEC → MC	$\overline{\text{PF.PS}}$
	PWR RES	Reset signal from the UEC to the MC computer unit	UEC → MC	$\overline{\text{RES.PS}}$
	READY	Inverter ready	UEC → MC	RDY
	TEMP >>	Temperature of heat sink too high (> 100 °C)	UEC → MC	$\overline{\text{ERR.TEMP}}$
	$U_{\text{DC LINK}}$ >>	U_Z too high (> approx. 850 V); power modules are switched off	UEC → MC	$\overline{\text{ERR.UZ.GR}}$
	X 71 SPINDLE	Safety relay for spindle on	–	–
	X 72 AX	Safety relay for axes on	–	–
	SH1 (RED) RDY (GREEN)	Safe stop 1; no enable from control (main contactor not active, DSP error, PLC error with Emergency Stop, hardware or software error of MC, CC) Axis/Spindle enabled	MC → UEC UEC → MC	$\overline{\text{SH1B}}$ RDY
	SH2	Safe stop 2; no drive enable from control (e.g. by the PLC, active via external signal or SH1)	MC → UEC	$\overline{\text{SH2}}$

1.10 PL 6xxx

1.10.1 General information

The PLC inputs and outputs of the iTNC 530 with HSCI are available via the external modular PL 6xxx PLC input/output systems.

The PL 6xxx consists of the PLB 6xxx basic module and one or more I/O modules. The basic modules are connected to the MC main computer via the HSCI interface.

Note

The MC 6xxx main computer of the iTNC 530 does **not** have an integrated PLC, and has **no** connections for TS or TT touch probes. In order to operate the control, at least the PL 62xx system PL (when using a CC 6106) or the UEC 11x controller unit with integrated inverter and PLC is necessary (the system PL is integrated in the UEC).

The PL 6xxx units are configured with the PC software IOconfig.

<table>
<tr><th colspan="2">PL 62xx system PL</th></tr>
<tr><td>System PL, consisting of PLB 620x basic module and I/O modules.
■ One module must be in the HSCI system if no UEC 11x is used
■ Available with 4, 6 or 8 slots
■ HSCI interface
■ Connections for TS and TT touch probes
■ Safety-relevant PLC inputs/outputs
They are mounted on standard NS 35 rails (DIN 46 227 or EN 50 022)</td><td></td></tr>
</table>

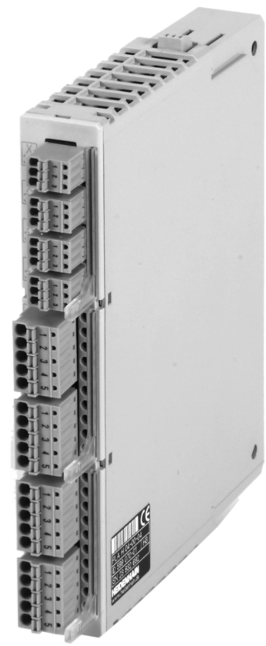
PL 61xx expansion PL	
Expansion PL, consisting of PLB 620x basic module and I/O modules. ■ Available with 4, 6 or 8 slots ■ HSCI interface ■ Up to 7 PL 61xx can be present in the HSCI system For an overview of the optionally available I/O modules, See "I/O modules" on page 124. They are mounted on standard NS 35 rails (DIN 46 227 or EN 50 022)	

1.10.2 I/O modules

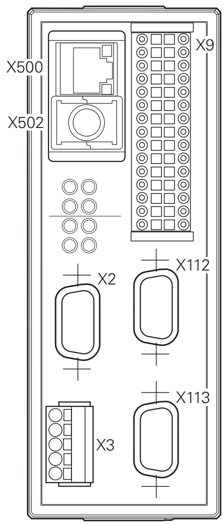
I/O modules are available with digital and analog inputs and outputs. For partially occupied PLB basic modules, the unused slots must be occupied by an empty housing (ID 383 022-11).

<table>
<tr><th colspan="2">PLD-H xx–xx–xx</th></tr>
<tr><td>Digital I/O module:
■ PLD-H 16-08-00:
I/O module with 16 digital inputs and 8 digital outputs
■ PLD-H 08-16-00:
I/O module with 8 digital inputs and 16 digital outputs</td><td></td></tr>
</table>

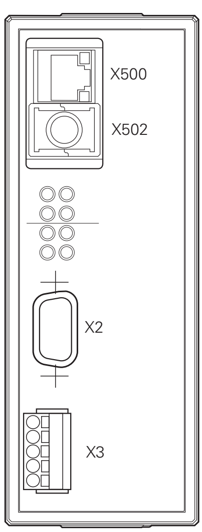
<table>
<tr><th colspan="2">PLA-H xx–xx–xx</th></tr>
<tr><td>Analog I/O module:

■ PLA-H 08-04-04:
Analog module with 4 analog inputs for PT 100 thermistors and 8 analog inputs ± 10 V and 4 analog outputs ± 10 V.</td><td></td></tr>
</table>

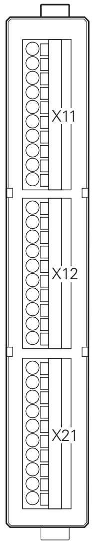
1.10.3 Connection overview of PLB 62xx

System module			
Pin layout	**Connector**	**Function**	**Page**
	X500	HSCI output	50
	X502	HSCI input	50
	X9	Safety-related PLC inputs/outputs	129
	X2	Reserved	
	X3	+ 24 V NC, +24 V PLC power supply	130
	X112	TS or TT touch trigger probe	
	X113	TS or TT touch trigger probe	
	Diagnosis (meanings of the LEDs): ■ Green (LEDs to the right): status of PL reports OK ■ Yellow (LEDs to the left): group message with error to PL ■ Alternating flashing of green and yellow LEDs in top row indicates a faulty HSCI connection		

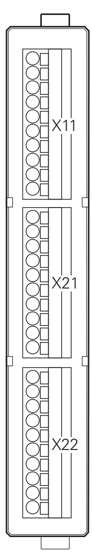
1.10.4 Connection overview of PLB 61xx

Expansion module			
Pin layout	**Connector**	**Function**	**Page**
	X500	HSCI output	50
	X502	HSCI input	50
	X2	Reserved	
	X3	+ 24 V NC, +24 V PLC power supply	130
	Diagnosis (meanings of the LEDs): ■ Green (LEDs to the right): status of PL reports OK ■ Yellow (LEDs to the left): group message with error to PL ■ Alternating flashing of green and yellow LEDs in top row indicates a faulty HSCI connection		

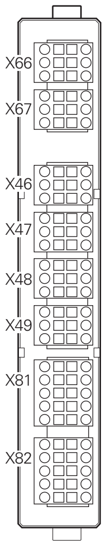
1.10.5 PLD/PLA connection overviews

<table>
<tr><td colspan="4">PLD-H 16-08-00</td></tr>
<tr><th>Pin layout</th><th>Connector</th><th>Function</th><th>Page</th></tr>
<tr><td rowspan="4">
</td><td>X11</td><td>PLC inputs</td><td>136</td></tr>
<tr><td>X12</td><td>PLC inputs</td><td>136</td></tr>
<tr><td>X21</td><td>PLC outputs</td><td>137</td></tr>
<tr><td colspan="3">Diagnosis (meanings of the LEDs):
■ Red (X11/pin 1) status LED
• Flashes: status of I/O module OK
• Permanently on or off: error on I/O module
■ Yellow (per output): Status of the output

Error recognition:
■ Short circuit: A short circuit is reported when a current >= 20 A flows for approximately 3 ms. Both the output-specific message and the group message are modal.
After the short circuit has been removed, the PLC must reset the output before it can be activated again.
■ Open circuit operation (line break): With load currents <= 300 mA, the PLD 16-8 reports a line breakage.</td></tr>
</table>

<table>
<tr><td colspan="4">PLD-H 08-16-00</td></tr>
<tr><th>Pin layout</th><th>Connector</th><th>Function</th><th>Page</th></tr>
<tr><td rowspan="4">
</td><td>X11</td><td>PLC inputs</td><td>136</td></tr>
<tr><td>X21</td><td>PLC outputs</td><td>137</td></tr>
<tr><td>X22</td><td>PLC outputs</td><td>137</td></tr>
<tr><td colspan="3">Diagnosis (meanings of the LEDs):
■ Red (X11/pin 1) status LED
• Flashes: status of I/O module OK
• Permanently on or off: error on I/O module
■ Yellow (per output): Status of the output

Error recognition:
■ Short circuit: A short circuit is reported when a current >= 20 A flows for approximately 3 ms. Both the output-specific message and the group message are modal.
After the short circuit has been removed, the PLC must reset the output before it can be activated again.
■ Open circuit operation (line break): With load currents <= 300 mA, the PLD 08-16 reports a line breakage.</td></tr>
</table>

PLA-H 08-04-04			
Pin layout	**Connector**	**Function**	**Page**
	X66 ... X67	±10 V analog outputs	140
	X46 ... X49	±10 V analog inputs	139
	X81 ... X82	PT 100 analog inputs	140

1.10.6 X9: Safety-related PLC inputs/outputs

Pin layout of PLB 620x:

The triggering outputs at X9 each supply up to 150 mA of output current. The only exceptions are the two outputs –STOS.A.G and -STO.A.G with max. 2 A of output current.

Seven outputs and twelve inputs are available at X9 of a PLB 620x for free use. Further PLC inputs/outputs must be realized by means of I/O modules.

Terminal	Signal design. NEW	OLD connector/signal design. (MC 42xC)	Assignment / Function
1a	24 V.A	X34	24 V supply of the outputs MC.RDY, O.0 to O.2
2a	Do not assign	–	–
3a	MC.RDY[a]	–SH1A (safe stop) X41.34 / O33	24 V output: (safe torque off) "Control is ready"
4a	O.0		24 V outputs (high-side driver)
5a	O.1		
6a	O.2		
7a	–ES.A	–NE1 / X42.4 / I3 Acknowledgment: "Control is ready"	24 V input Emergency Stop input 1
8a	I.0		24 V inputs (PLC)
9a	I.1		
10a	I.2		
11a	I.3		
12a	I.4		
13a	I.5		
14a	–PF.PS.AC	–PF.PS.AC (signal on X69)	24 V outputs for powerfail
15a	–PF.PS.DC	–PF.PS.ZK (signal on X69)	
1b	24 V.B	X44	24 V supply of the outputs O.3 to O.6
2b	0 V		0 V PLC for all I/Os
3b	O.3[a]		24 V outputs (high-side driver)
4b	O.4		
5b	O.5		
6b	O.6		
7b	–ES.B	–NE2 / X42.33 / I32 "Drive enabling"	24 V input Emergency Stop input 2
8b	I.6		24 V inputs (PLC)
9b	I.7		
10b	I.8		
11b	I.9		
12b	I.10		
13b	I.11		
14b	–SP.REF+	X30	Optocoupler input, Spindle ref.
15b	–SP.REF–	X30	

a. 2-A outputs

X3: +24 V NC, +24 V PLC power supply

Pin layout of X3:

Connecting terminal	Pin layout
1 (top terminal)	+ 24 V NC
2	0 V NC (ground +24 V NC)
3	⏚ Protective ground Minimum wire cross section of the power cables for 24 V PLC
4	+ 24 V PLC
5 (bottom terminal)	0 V PLC (ground +24 V PLC)

X112/X113: Triggering touch probe

> **Note**
>
> The touch probes are connected to the PLB 620x or UEC 11x PLC system module at X112 (TS) and X113 (TT).

> **Note**
>
> For the PLB 62xx up to variant -02, please note:
>
> To connect a TT touch probe to the iTNC 530, you have to connect the touch probe adapter, ID 667 674-01, to X113.

X112/X113 pin layout on PLB 62xx as of variant-03 and UEC 11x:

(D-sub, 15-pin, 3-row)

> **Note**
>
> The interface complies with the requirements of EN 60204-1:2006 for protective extra-low voltage (PELV).

Female	Assignment X112 (TS)	Assignment X113 (TT)
1	Trigger signal	Trigger signal
2	$\overline{\text{Trigger signal}}$[a]	$\overline{\text{Trigger signal}}$a
3	TS ready	Do not assign
4	$\overline{\text{Battery warning}}$	$\overline{\text{Battery warning}}$
5	+ 5 V NC (+/–5 %)	+ 5 V NC (+/–5 %)
6	Start TS	Do not assign
7	Do not assign	Start TT
8	0 V NC	0 V NC
9	0 V NC	0 V NC
10	+ 24 V NC	+ 24 V NC
11	Do not assign	TT ready
12	Do not assign	Do not assign
13	Do not assign	Do not assign
14	Do not assign	Do not assign
15	Do not assign	Do not assign

a. Stylus at rest means logic level HIGH.

Wire colors of adapter cable ID 633 608-xx for X112/X113 to TS or TT:

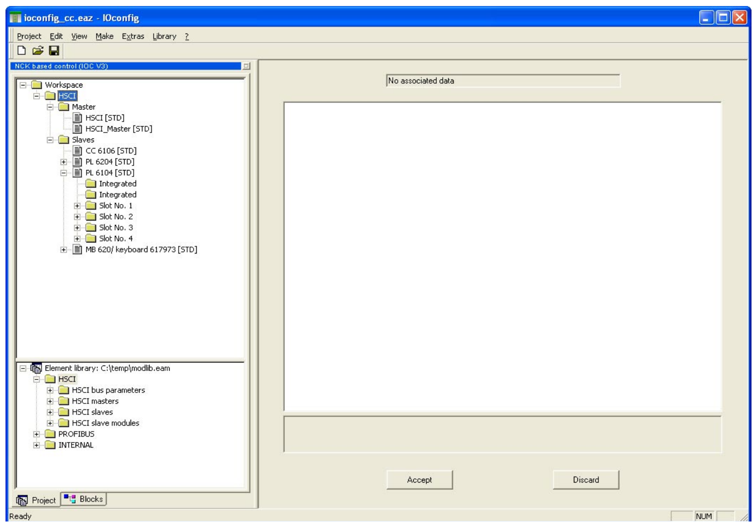
X112/X113 on PLB 62xx or UEC 11x	Adapter cable 633 608-xx		
Female (D-sub)	**Male (D-sub)**	**Color**	**Female (M12)**
1	1	Not assigned	
2	2	Pink	4
3	3	Green	5
4	4	Gray	6
5	5	Not assigned	
6	6	BL	3
7	7	White	7
8	8	Not assigned	
9	9	White/Green	1
10	10	Brown/Green	2
11	11	Brown	8
12 to 15	12 to 15	Not assigned	

1.10.7 Configuration with IOconfig

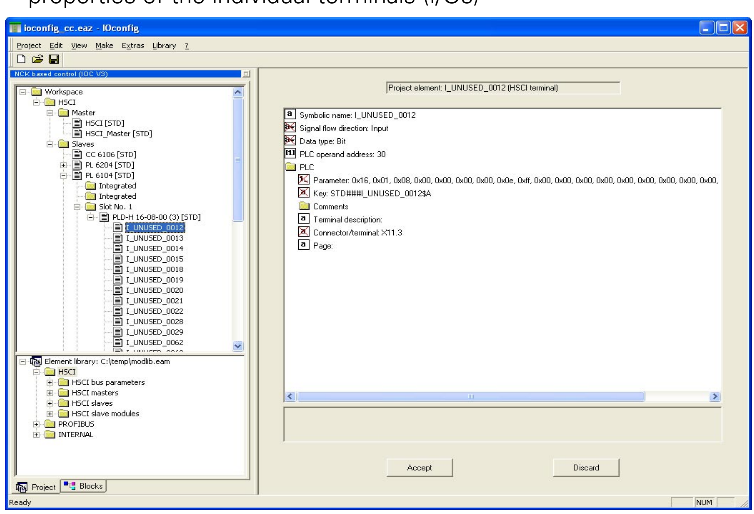
The IOconfig program for PCs is used to configure the new input/output modules connected via the HSCI network. The program has previously been used to describe a PROFIBUS system, and now it also includes the HSCI system. If desired, IOconfig creates the configuration file (IOC file) and symbol definition file (DEF file) for the control or the PLC program.

The following properties (and others) of the HSCI system are defined with IOconfig:

- Position of all input/output modules in the HSCI system (PL 6xxx (FS) with associated PLD-H and PLA-H modules, MB 6xx (FS))



- Definition of all I/Os freely available to the OEM on PLD-Hs, PLA-Hs and MB 6xx (FS)
- Assignment of symbolic names, PLC operand addresses and other properties of the individual terminals (I/Os)



HEIDENHAIN provides device files (HDD files) for all HSCI components with the technical characteristics of these components. These HDD files are of critical importance for the configuration of the HSCI system. You should therefore make sure that you always use the latest version of IOconfig and the HDD files for the configuration.

When configuring a project, the IOconfig program takes the description of the HSCI components from the HDD file and transfers it to the configuration file (IOC file).

Note

HEIDENHAIN recommends updating all IOC files and EAZ projects with the IOconfig version 2.2.02. This is the only way to ensure that you can use all of the features provided by the new HSCI components.

For more information about using IOconfig, please refer to the program's on-line help.

Updating the HSCI configuration

To update the HSCI configurations, proceed as follows:

Install the IOconfig version 2.2.02 or a later IOconfig version. In the HEIDENHAIN FileBase, you will find the current version of the IOconfig software for PCs for downloading. With IOconfig version 2.2.02, however, changes to the HDD files are not transferred automatically.

If new HDD files exist on the PC, IOconfig 2.x does not update these files automatically until these files:

- are copied into the directory in which IOconfig searches for HDD files (to be defined in Extras/Options and Project/Settings), and
- are read in with the HDD import function or imported via an IOC import of GSD/HDD files.

HDD files cannot be imported in any other way, e.g. when EAZ files are loaded. Any existing new HDD files will therefore be ignored.

Do the following steps manually in order to be able to use all available features of new HSCI components:

1. Transfer the HDD files

Check where the standard directory and the project directory for HDD files are located. The two directories should usually be identical:

- Extras -> Options: Standard directory for HDD library: xxxxxxx
- Project -> Project Settings: Directory of HDD library: xxxxxxx

Move or delete all HDD files being used.

Then copy all current HDD files from IOconfig (e.g. from C:\Programme\HEIDENHAIN\IOconfig\HDD) to the standard directory or the project directory of the HDD library.

2. Update the EAZ project

The only way to ensure that all HDD files specified in the IOC file are evaluated and used for generating the HSCI components in an IOC project is exporting and importing an IOC file. These steps are required to be able to detect new components, diagnostic information and parameters in new HDD files.
The essential steps are creating an IOC file and re-importing this IOC file. These steps ensure that all diagnostic information is updated and all new parameterization possibilities become visible.

If the message **`Missing HDD files`** is displayed during import, press **`CONTINUE`** to continue the process. Then the slave is inserted, but it will not have any diagnostic or parametric properties.
This can happen if the names of HDD files were changed and the IOC file still contains the previous file names. A new IOconfig version with current HDD files will usually not find these old files. **If it finds them nevertheless, then move the old HDD files and make sure that you do not use them anymore.**

Something similar can occur if the name of an HDD file is correct, but the name of the slave contained in it was changed and the IOC file still contains the old name of the slave. The error message **`Missing HDD files`** will then not be displayed, but the diagnostic and parametric properties of the slave will be missing.

In such cases, the slave in question must be deleted and reinserted from the correct HDD file. The new functions provided by IOconfig 2.2.02 for this purpose can be used to transfer the terminal names during the Delete/Paste actions.

3. Check the machine operating panels and keyboard units

If the **`Override`** module is shown for a MB/TE slave, then the respective slave must be deleted and reinserted from the correct HDD file. The new functions provided by IOconfig 2.2.02 for this purpose can be used to transfer the terminal names during the Delete/Paste actions.

1.10.8 PLC inputs/outputs

If the holding brakes of motors are not driven exclusively over the inverter outputs for brakes, a protective circuit in the form of a varistor must be used. Due to the inductance of the holding brakes and possibly used relays, a voltage peak that may exceed 1000 V occurs when the exciting current is switched off. This may destroy other electronics, such as connected PLC inputs/outputs.

A protective circuit is not necessary only if the holding brakes are driven exclusively over the inverter outputs for brakes, because the voltage is limited internally by electronic switches.

PLC inputs

Pin layout on the PLD-H xx-xx-xx input/output module:

Note

The 0 V terminals of X11, X12 and X14 of the PLD-H are connected internally. These connections are used for potential bonding of the electronics and for operating the LEDs. Since only a low current is to be led away (max. 50 mA), it is sufficient to establish only one 0 V connection (preferably at X11).

X11: PLC inputs channel A										
Pin layout	**Terminal**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PL 6xxx Slot 1	0 V PLC	0 V PLC	I0.A	I1.A	I2.A	I3.A	I4.A	I5.A	I6.A	I7.A

X12: PLC inputs channel A										
Pin layout	**Terminal**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PL 6xxx Slot 1	0 V PLC	0 V PLC	I8.A	I9.A	I10.A	I11.A	I12.A	I13.A	I14.A	I15.A

The designation "Channel A" or ".A" for the PLC inputs is relevant only for I/O modules with Functional Safety (FS).

Fast PLC inputs

Only the first four slots of a PL 6xxx can be used for fast PLC inputs. The fifth slot and the successive slots (on PL 6x06, PL 6x08) must not be defined as fast PLC inputs.

The configuration of fast PLC inputs in the HSCI system corresponds to the previous configuration using machine parameters.

PLC outputs **Pin layout on the PLD-H xx-xx-xx input/output module:**

X21: PLC outputs, channel A										
Pin layout	**Terminal**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PL 6xxx Slot 1	O0.A	O1.A	O2.A	O3.A	O4.A	O5.A	O6.A	O7.A	24 V PLC for O0.A to O3.A	24 V PLC for O4.A to O7.A

X22: PLC outputs, channel A										
Pin layout	**Terminal**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PL 6xxx Slot 1	O8.A	O9.A	O10.A	O11.A	O12.A	O13.A	O14.A	O15.A	24 V PLC for O8.A to O11.A	24 V PLC for O12.A to O15.A

"Channel A" or ".A" of PLC outputs needs to be entered only for I/O modules with Functional Safety (FS).

Output signals and addresses

The switching outputs are transistor outputs with current limitation.

Please note:

- Permissible load: Resistive load—inductive load only with quenching diode parallel to inductance
- PLD H: The outputs are short-circuit proof.

Output signals:

	PLD-H
Min. output voltage for "1" signal	3 V below supply voltage

Note

The switching outputs need a minimum load of 5 mA.
They conform to EN 61131-2.

Attention

PLC outputs must neither be connected to a 24 V supply, nor to other PLC outputs with a difference in potential. Otherwise, the voltage present at the PLC outputs is transmitted to the power supply. As a result, the PLC outputs that can be switched off may nevertheless be supplied with this voltage.

1.10.9 Analog PLC inputs/outputs

Note

The interfaces of the PLA-H 08-04-04 module are electrically separated from the 230 V line power in accordance with EN 50178.

Specifications:

Power consumption of PLA-H 08-04-04 under full load: 4.5 W

Analog inputs

Voltage range:	–10 V to +10 V
Input resistance:	> 40 kΩ
Resolution:	10 mV

Analog outputs

Voltage range:	–10 V to +10 V
Load impedance:	> 5 kΩ
Output current:	< 2 µA
Resolution:	10 mV

Inputs for Pt 100 thermistors

Constant current:	4.096 mA
Temperature range:	0 °C to 100 °C
Resolution:	0.01 °C, increment: 0.03 °C

X31 to X34: Analog inputs

Pin layout

Connecting terminals	Pin layout
1	–10 V to +10 V (input)
2	0 V (reference potential)
3	Shield

X46 to X49: Analog inputs

Pin layout

Connecting terminals	Pin layout
1a/1b	–10 V to +10 V (input)
2a/2b	0 V (reference potential)
3a/3b	Shield

X51 to X54: Analog output

Pin layout

Connecting terminals	Pin layout
1	–10 V to +10 V (output)
2	0 V (reference potential)
3	Shield

X66 to X67: Analog output

Pin layout

Connecting terminals	Pin layout
1a/1b	–10 V to +10 V (output)
2a/2b	0 V (reference potential)
3a/3b	Shield

X71 to X74: Connection for Pt 100

Pin layout:

Connecting terminals	Pin layout
1	I+ Constant current for PT 100
2	U+ Measuring input for PT 100
3	U– Measuring input for Pt 100
4	I– Constant current for Pt 100
5	Shield

X81 to X82: Connection for Pt 100

Pin layout:

Connecting terminals	Pin layout
1a/1b	I+ Constant current for PT 100
2a/2b	U+ Measuring input for PT 100
3a/3b	U– Measuring input for Pt 100
4a/4b	I– Constant current for Pt 100
5a/5b	Shield

1.10.10 Switching inputs 24 V– (PLC)

Input signals and addresses

Input signals of the switching inputs of the PLD-H xx-xx-xx and the UEC 11x:

Voltage range	PLD-H 16-08-00 and UEC 11x
"1" signal: U_i	11 V to 28.8 V
"0" signal: U_i	–3 V to 2.2 V

Current ranges	PLD 16-08-00 and UEC 11x
"1" signal: U_i	2.0 mA to 6.1 mA
"0" signal: I_i when U_i = 3.2 V	0.3 mA

Input signals of the switching inputs of MB 6xx or connector X9 of a PL 62xx:

Voltage range	PLD-H 16-08-00 and UEC 11x
"1" signal: U_i	11 V to 28.8 V
"0" signal: U_i	–3 V to 2.2 V

Current ranges	PLD 16-08-00 and UEC 11x
"1" signal: U_i	2.1 mA to 6.0 mA
"0" signal: I_i when U_i = 3.2 V	0.43 mA

1.11 UEC 11x: Digital PLC Inputs/Outputs

X4: PLC inputs

Connection on the front of the UEC 11x:

Terminal	Signal designation	Assignment / Function
1a	+24 V PLC.01	24 V supply of the outputs MC.RDY, O16 to O22
2a	+24 V PLC.02	24 V supply of the outputs O8 to O15
3a	+24 V PLC.03	24 V supply of the outputs O0 to O7
4a	0 V PLC	0 V for all I/Os
5a	–REF.SP	Input for spindle reference signal
6a	0 V PLC	0 V for all I/Os
7a	I12	24 V inputs
8a	I13	
9a	I14	
10a	I15	
11a	I16	
12a	I17	
1b	I0	24 V inputs
2b	I1	
3b	I2	
4b	I3	
5b	I4	
6b	I5	
7b	I6	
8b	I7	
9b	I8	
10b	I9	
11b	I10	
12b	I11	

X5: PLC inputs

Connection on the front of the UEC 11x:

Terminal	Signal designation	Assignment / Function
1a	I30	24 V inputs
2a	I31	
3a	I32	
4a	I33	
5a	I34	
6a	I35	
7a	I36	
8a	I37	
9a	–ES.A	+24 V input Acknowledgment: "Control is ready"
10a	–ES.B	24 V input "Drive enabling"
11a	Do not assign	
12a	Do not assign	
1b	I18	24 V inputs
2b	I19	
3b	I20	
4b	I21	
5b	I22	
6b	I23	
7b	I24	
8b	I25	
9b	I26	
10b	I27	
11b	I28	
12b	I29	

Note

If the PLC inputs integrated in the UEC 11x do not suffice for your application, you can connect up to seven additional external PL 61xx expansion PLs via the HSCI interface.

X6: PLC outputs

Connection on the top of the UEC 11x:

Terminal	Signal designation	Assignment / Function
1a	O4	24 V outputs, can be switched off via terminal X4.3a (+24 V PLC.03)
2a	O5	
3a	O6	
4a	O7	
5a	O12	24 V outputs, can be switched off via terminal X4.2a (+24 V PLC.02)
6a	O13	
7a	O14	
8a	O15	
9a	O20	24 V outputs, cannot be switched off
10a	O21	
11a	O22	
12a	–MC.RDY	24 V output "Control is ready"
1b	O0	24 V outputs, can be switched off via terminal X4.3a (+24 V PLC.03)
2b	O1	
3b	O2	
4b	O3	
5b	O8	24 V outputs, can be switched off via terminal X4.2a (+24 V PLC.02)
6b	O9	
7b	O10	
8b	O11	
9b	O16	24 V outputs, cannot be switched off
10b	O17	
11b	O18	
12b	O19	

Note

If the PLC outputs integrated in the UEC 11x do not suffice for your application, you can connect up to seven additional external PL 61xx expansion PLs via the HSCI interface.

1.12 PSL13x Low-Voltage Power Supply Unit

1.12.1 General information

PSL 130 power supply unit for HSCI components with +24-V power supply when using a HEIDENHAIN inverter system.

The **PSL 130** power supply unit was conceived in order to be able to provide the HSCI components of the iTNC 530 with +24-V NC voltage and +24-V PLC voltage.

The output voltages of the **PSL 130** fulfill the requirements for Protective Extra Low Voltage (PELV) according to EN 50178. The power supply unit is powered with line voltage (L1, L2) and the DC-link voltage Uz. This is used to produce the +24-V NC and +24-V PLC output voltages.

ID 575 047-xx PSL 130

PSL 135 power supply unit for supplying the HSCI components in a double-row configuration.

The **PSL 135** power supply unit was conceived in order to be able to provide the HSCI components of the iTNC 530 with +24-V NC voltage, + 24-V PLC voltage and +5 V.

The output voltages of the **PSL 135** fulfills the requirements for Protective Extra Low Voltage (PELV) according to EN 50178. The power supply unit is powered with line voltage (L1, L2) and the DC-link voltage Uz. This is used to produce the +24-V NC, +24-V PLC and +5-V output voltages.

ID 627 032-xx PSL 135

The two 24 V output voltages of the PSL 13x are generated by two separate power supplies. The + 24 V NC and + 24 V PLC voltages are separated from each other by basic insulation and fulfill the requirements of EN 61800-5-1 for low voltage electrical separation.

The 0 V line of the NC supply voltage must be connected separately to the central grounding point of the machine (= central functional ground). The 0 V NC voltage in the PSL 130 is therefore connected internally with the outward conductor to signal ground (= central functional ground). An outward connector to protective ground will not be available for the 0 V PLC voltage until variant 03. With the variants up to and including variant 02, the 0 V PLC must be tapped at terminal X90.4 and led to the central protective ground. In variant 03 and later variants, an outward connector at the PSL will be available for connecting the 0-V PLC voltage, too.

With the PSL 135, the connections to the central functional ground and protective ground must be realized over the 0 V connections. The respective outward connections will not be available until the next hardware variants of the PSL 135.

The +24-V NC voltage is separated from the line power by double insulation. The +24-V PLC voltage is separated from the line power by basic insulation. Because of the connection to the PLC part of the HSCI system the 24 V PLC is a supply voltage with basic insulation. This voltage must therefore not be linked with other voltages (24-V NC).

The +5 V NC supply voltage for X74 of the PSL 135 is taken from the +24 V NC supply voltage and is therefore galvanically connected with it.

Danger

- For the entire HSCI system, the +24 V NC power supply voltage is required to be safely separated voltage and must not be connected with the +24 V PLC voltage of the system.
- Protective Extra Low Voltage (PELV) according to EN 61800-5-1 must be complied with for the +24 V NC power supply of the machine.
- The 0 V signal of the PLC power supply must be connected by a 6 mm^2 conductor to the machine's central protective ground (PE).
- Refer to your control's grounding diagram!

1.12.2 Specifications

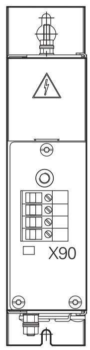
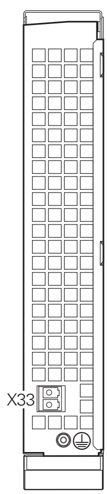
Specifications	PSL 130	PSL 135
Power supplies: at X33 (L1, L2)	400 V~ ± 10% 50 Hz	
Power supplies: DC-link power bar or X31	400 V– to 750 V–	
Protection	Page 152	
Output voltages: Accuracy of the +24 V NC Accuracy of the +24 V PLC Accuracy of the +5 V NC	+24 V NC: +/–5 % +24 V PLC: Variations depending on the load, between 20 V and 28 V - - -	+24 V NC: +/–5 % +24 V PLC: Variations depending on the load, between 20 V and 28 V +5-V NC: +/– 5% (power supply unit is adjusted to 5.2 V)
Output power	24 V NC output: max. 500 W 24 V PLC output: max. 500 W Total power output: max. 750 W	24 V NC output: max. 350 W 24 V PLC output: max. 500 W 5 V NC output: max. 100 W +/–15 V at X69: max. 50 W Total power output: max. 750 W
24-V NC output current	Max. 20 A	Max. 14.5 A
24-V PLC output current	Max. 20 A	Max. 20 A
5-V NC output current	- - -	Max. 20 A
Total 24 V output current	Continuous load: max. 31 A Max. load time 1 s: max. 41 A Max. load time 0.1 s: max. 46 A The outputs are short-circuit proof and switch off automatically when overloaded.	
5 V output current	- - -	Max. 20 A The outputs are short-circuit proof and switch off automatically when overloaded.
Power consumption	Max. 1000 W	
Degree of protection	IP 20	
Module width	50 mm	
Weight	2.1 kg	
ID number	575 047-xx	627 032-xx

It is possible to connect both 24 V output voltages of the PSL 13x in parallel. In this way the PSL supplies only a maximum output power of 750 W, which fulfills the requirements for PELV according to EN 61800-5-1. However, because of the parallel circuit, a PSL 130 supplies only an output voltage of +24 V, which can then be used as desired for the 24 V NC voltage or the 24 V PLC voltage. An additional PSL 13x can be used for the respective other voltage.

Danger

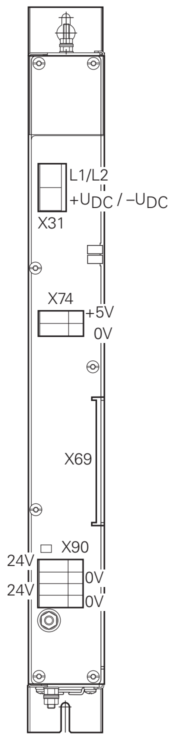
When using it to supply the PLC, the common 0 V signal must be connected by a conductor of at least 6 mm^2 to the machine's central protective ground (PE).

1.12.3 Connection overview

PSL 130 pin layout	Connector	Function	Page
 	Conductor bar	Connection of DC-link voltage Uz	151
	B – Signal ground	Signal ground (the connection is connected internally with 0-V NC)	154
	X90	Output for control voltages: ■ Terminal 1: +24 V NC (double insulation from line power) ■ Terminal 2: 0 V NC (ground +24 V NC) ■ Terminal 3: +24 V PLC (basic insulation from line power) ■ Terminal 4: 0 V PLC (ground +24 V PLC)	154
	X33	Input voltages L1, L2	152
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

PSL 135 pin layout	Connector	Function	Page
	X31	Input voltages L1, L2 and connection of the DC link voltage Uz	153
	X74	Output for control voltages: ■ Terminal 1: +5 V ■ Terminal 2: 0 V	154
	X69	Power supply and control signals for CC 61xx (for X69 on CC)	
	X90	Output for control voltages: ■ Terminal 1: +24 V NC (double insulation from line power) ■ Terminal 2: 0 V NC (ground +24 V NC) ■ Terminal 3: +24 V PLC (basic insulation from line power) ■ Terminal 4: 0 V PLC (ground +24 V PLC)	154
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

DC-link connection

Since the power to the PSL 130 is supplied through the DC-link, the voltage fed into the dc-link by the motors that are still running can be used during line voltage failures. The PSL 130 uses the supply voltage buffered via the DC-link to maintain the power supply for the control until the inverter system has been shut down properly by the control.

If the PSL 130 is connected over litz wires, a wire cross section of at least 1.5 mm^2 must be used. Fuses or a motor protection switch of 6.3 A or greater for conductor protection must be selected depending on the wire cross section used. The PSL 130 is protected internally by a fuse (4 A).

Connecting terminals	Pin layout
$-U_Z$	DC-link voltage –
$+U_Z$	DC-link voltage +

HEIDENHAIN offers insulated conductor bars if you want to position the PSL130 next to the left of the UVR inverter and connect it to the DC-link via conductor bars. Two conductor bars are required for each connection. The conductor bars are shipped in shown in packaging units. Also, this position makes it possible to connect the grounding conductor of the PSL via conductor bars. Therefore, angulated and non-insulated conductor bars are also included in this packaging unit. This makes it possible to continue using the straight conductor bar included with the UVR for the grounding conductor connection from the UVR to the inverters via the right side.

Length	For connection to	ID (conductor bar set)
150 mm	UVR 120D/130D	687 669-xx
200 mm	UVR 140D/150D	687 670-xx
250 mm	UVR 160D/160DW	687 671-xx

Note

Tightening torque for the screws of the DC-link conductor bars = 3.5 Nm.

X33: Input voltage of the PSL 130

Supply voltage: 400 V ± 10%

Connection:

Connecting terminal	Pin layout
L1	Phase 1 / 400 V~ ±10% / 50 Hz to 60 Hz
L2	Phase 2 / 400 V~ ±10% / 50 Hz to 60 Hz
	Protective ground (YL/GN), ≥ 10 mm^2
	Connecting lead: Wire cross section: 1.5 mm^2 (AWG 16) **Conductor protection:** Fuses or a motor protection switch of 6.3 A or greater depending on the wire cross section used. **Line fuse:** Internal protection of the PSL (4 A).
Tightening torque: for the connecting terminals 0.5 to 0.6 Nm **Grounding terminal:** ≥ 10 mm^2 (AWG 6) **Strain relief:** Ensure that the connecting cables are not subject to excessive strain.	

Through the connection to 400 V (L1, L2) via X33, the output voltages of the PSL 130 are available as soon as the machine's main switch has been turned on. These voltages are indispensable to be able to boot the control.

Note

HEIDENHAIN recommends connecting the PSL 130 power supply unit to the U_Z DC-link voltage and the 400 V supply voltage (X33).

X31: Input voltage of the PSL 135

Supply voltage: 400 V ± 10% or 400 V– to 750 V–

Connection:

Connecting terminal	Pin layout
L1	Phase 1 / 400 V~ ±10% / 50 Hz to 60 Hz
L2	Phase 2 / 400 V~ ±10% / 50 Hz to 60 Hz
+UDC	400 V– to 750 V–
–UDC	0 V–
	Protective ground (YL/GN), ≥ 10 mm^2
	Connecting lead: Wire cross section: 1.5 mm^2 (AWG 16) **Conductor protection:** Fuses or a motor protection switch of 6.3 A or greater depending on the wire cross section used. **Line fuse:** Internal protection of the PSL (4 A).
Tightening torque: for the connecting terminals 0.5 to 0.6 Nm **Grounding terminal:** ≥ 10 mm^2 (AWG 6) **Strain relief:** Ensure that the connecting cables are not subject to excessive strain.	

Through the connection to 400 V (L1, L2) via X31, the output voltages of the PSL 135 are available as soon as the machine's main switch has been turned on. These voltages are indispensable to be able to boot the control.

Since the power to the PSL 135 is supplied through the dc-link, the voltage fed into the dc-link by the motors that are still running can be used during line voltage failures. The PSL 135 uses the supply voltage buffered via the DC-link to maintain the power supply for the control until the inverter system has been shut down properly by the control.

Note

HEIDENHAIN recommends connecting the PSL 135 power supply unit to the U_Z DC-link voltage and the 400 V supply voltage (X31).

X90: Output voltage of the PSL 13x

Output voltages: + 24 V (2 x)

Connection:

Note

The interface complies with the requirements of EN 60204-1:2006 for protective extra-low voltage (PELV).

Connecting terminal	Pin layout
Terminal 1 (top)	+ 24 V NC
Terminal 2	0 V NC (ground +24 V NC)
Terminal 3	+ 24 V PLC
Terminal 4 (bottom)	0 V PLC (ground +24 V PLC)
Tightening torque: for the connecting terminals 0.5 to 0.6 Nm **Strain relief:** Ensure that the connecting cables are not subject to excessive strain.	

X74: Output voltage of the PSL 135

Output voltages: + 5 V

Connection:

Note

The interface complies with the requirements of EN 60204-1:2006 for protective extra-low voltage (PELV).

Connecting terminal	Pin layout
Terminal 1 (top)	+ 5 V NC
Terminal 2	0 V NC (also ground +24 V NC)
Tightening torque: for the connecting terminals 0.5 to 0.6 Nm **Strain relief:** Ensure that the connecting cables are not subject to excessive strain.	

Signal ground

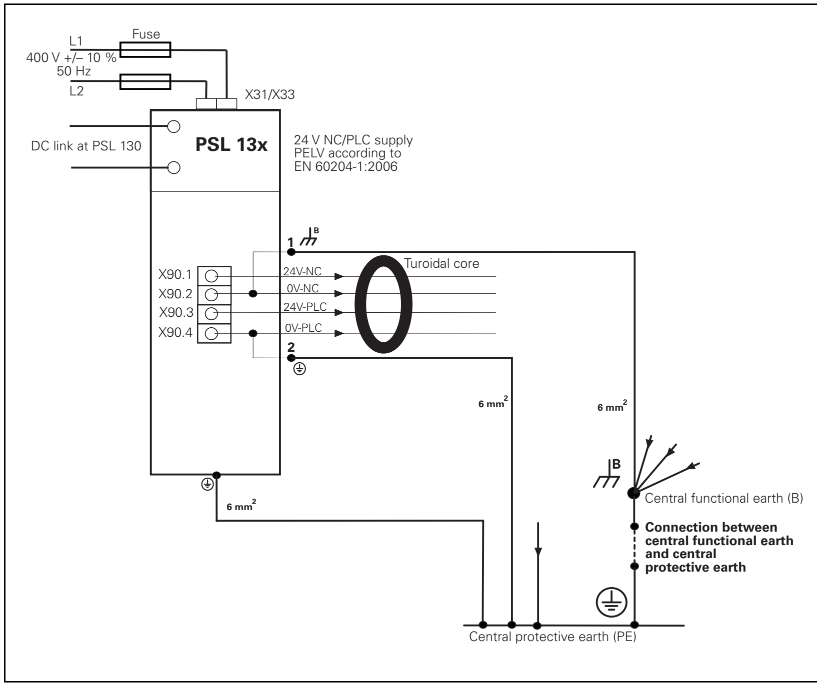
Connections for signal ground (= functional ground), which are connected internally in the PSL 13x to the 0-V NC.

Connection:

Note

Conductor cross section of at least 6 mm^2 for connecting to signal ground (= central functional ground).

Power connection



The 0 V line of the NC supply voltage must be connected separately to the central grounding point of the machine (= central functional ground). The 0 V NC voltage of the PSL 130 is therefore connected internally with the outward conductor to signal ground (= central functional ground).

An outward connector to protective ground will not be available for the 0 V PLC voltage until variant 03. With the variants up to and including variant 02, the 0 V PLC must be tapped at terminal X90.4 and led to the central protective ground. In variant 03 and later variants, an outward connector at the PSL will be available for connecting the 0-V PLC voltage, too.

With the PSL 135, the connections to the central functional ground and protective ground must be realized over the 0 V connections. The respective outward connections will not be available until later hardware variants of the PSL 135.

If a suitable type of network (see Technical Manuals for Inverters and Motors) is used, the inverter system from HEIDENHAIN and the PSL 13x are connected to the main power line without an additional isolating transformer.

If an isolating transformer is required for the inverter system due to the type of network, then the PSL 13x must also be powered via the isolating transformer of the inverter system.
If the line voltage is 3 x 480 V~ and inverters suitable for this voltage (UE compact inverters) are used, then the PSL 13x must be powered via an additional autotransformer. The required output voltage of the autotransformer is 400 V~ +0%/–15%.

Conducted interference

To suppress conducted interference, the conductors for the 24 NC and 24 V PLC output voltages (X90 of the PSL) must be passed through a toroidal core (ID 309 694-07, inside diameter 14 mm). There are different ways to do this depending on the PSL variant:

- **Variant 02 of the PSL 130, variant 01 of the PSL 135**
  The toroidal core is included with the PSL. The output lines of connector X90 (24 V NC, 24 V PLC) must be passed through the toroidal core in order to suppress conducted interference. The lines must not be wound around the toroidal core, however.
- **Variant 03 of the PLS 130**
  The toroidal core is firmly secured to the outside of the PSL housing. The output lines of connector X90 (24 V NC, 24 V PLC) must be passed through the toroidal core in order to suppress conducted interference. The lines must not be wound around the toroidal core, however.
- **Variant 02 of the PLS 135**
  The toroidal core is integrated in the unit. No further external measures for noise suppression must be taken.

1.13 MS 110 / MS 111 Mounting Case for Double-Row Configuration

1.13.1 General information

MS 110 without additional 24-V supply ID 658 132-xx MS 110	
MS 111 with additional 24-V supply ID 673 685-xx MS 111	

Sometimes limited space prevents the control and inverter system from being mounted in the same row in a machine's electrical cabinet, meaning that they must be mounted in two separate rows. In other cases the design calls for a second electrical cabinet to house the inverter system. This means that the distribution and arrangement of the components can be very different from case to case.

In order to establish an electrical connection (immune to noise) between the components of the inverter system, the MS 1xx mounting cases are needed. The ribbon cables (unit bus, PWM lines, supply bus) from the other components are connected to the MS mounting cases, and shielded round cables of the appropriate lengths connect the MS mounting cases with each other.

In some cases, in order to ensure that the power supply for the fans of the inverters is maintained under all circumstances, it is also necessary to feed 24 V from an external power supply unit to the unit bus (since this is handled by the unit bus).

In most cases the additional 24 V are not needed, since the UVR 1xxD provides enough current for the fans.

If this is the case, and a double-row configuration is used, then two MS 110 mounting cases are necessary (see basic circuit diagram). For the current consumption of the fans, refer to the "Inverter Systems and Motors" Technical Manual. Based on these values you can calculate whether feeding in the additional 24 V is necessary.

For inverter systems with many powerful UM 1xxD power modules, it might be the case that the current provided by the UV(R)1xx power supply unit for the fans of the UMs does not suffice to guarantee safe and reliable operation of the fans. The sum of the currents must not exceed the maximum current provided by the UV(R).

If it is exceeded, then an MS111 must be used in the inverter row where the current consumption is very high.

With the MS 111, the 24 V from an external 24-V power supply unit are fed to the unit bus X79 in order to ensure reliable operation of the fans, and therefore the reliable cooling of the inverters. The 24-supply that is routed via X79C is interrupted internally at the MS 111.

Note

With the MS 111, an additional power supply unit must be used for the 24-V power supply of the fans.

1.13.2 Double-row configuration

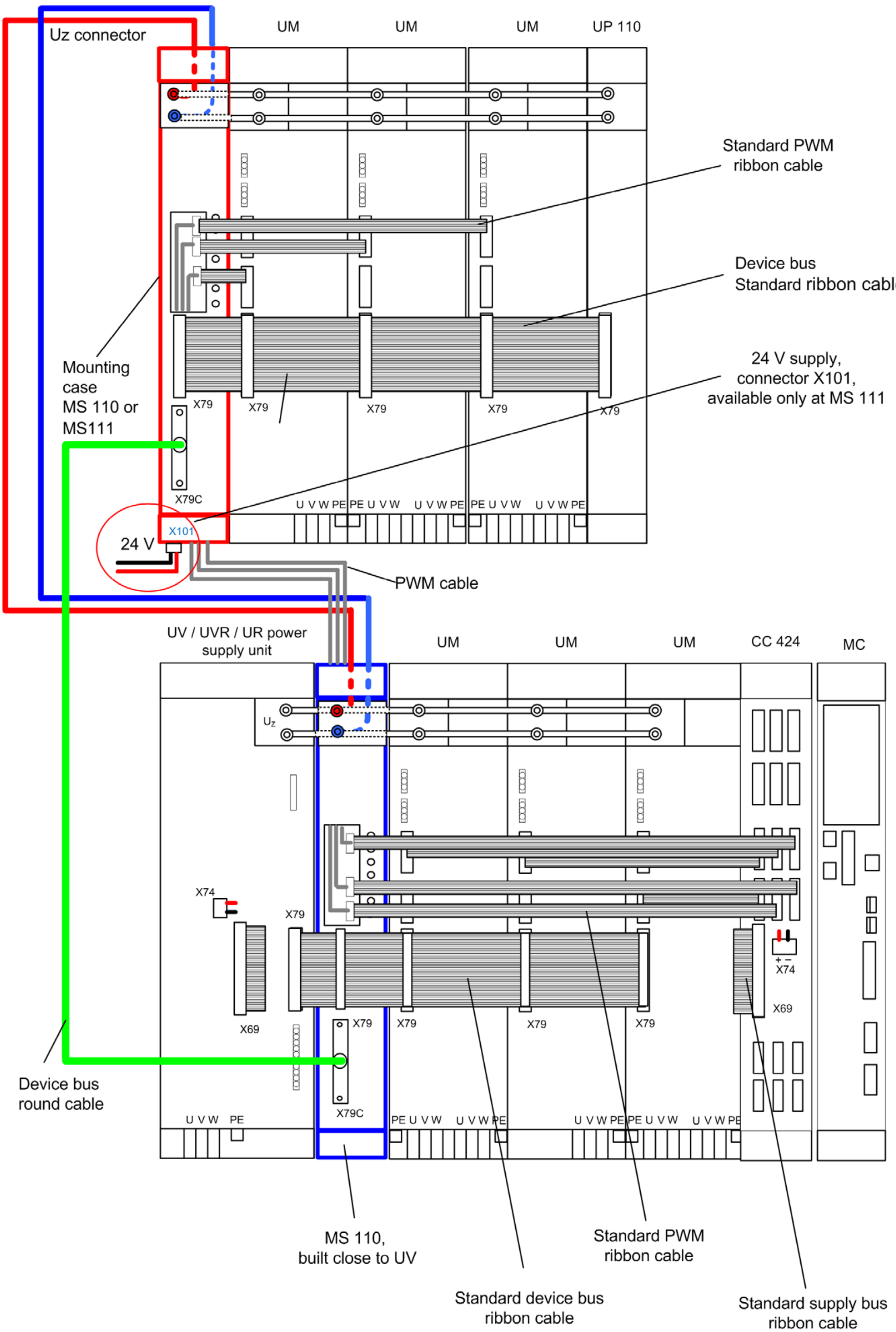
Components and cables for double-row configuration:

Component/Cable	ID
Unit bus cable (shielded, round) with 37-pin D-sub at both ends; max. length: 3 m	ID 664 023-xx
PWM cable (round) with ribbon connector at both ends; max. length: 5 m	ID 664 332-xx
Supply bus cable (round) with ribbon connector at both ends; max. length: 5 m (only necessary if the UV(R) 1xxD is not in the same row as the CC/MC)	ID 361 508-xx
Litz wire for DC link (16 mm^2, shielded, color: red); max. length: 3 m	ID 655 440-xx
Litz wire for DC link (16 mm^2, shielded, color: blue); max. length: 3 m	ID 655 438-xx

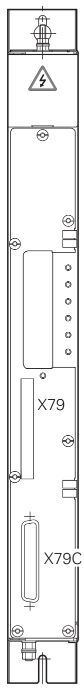
When using a double-row configuration, please keep the following in mind:

- The litz wires used for the DC-link connection of the power modules in the "second row" must not be longer than 3 m.
- Litz wires with 16-mm² cross section make a DC-link current of approx. 67 A possible. In a regenerative system, this results in approx. 35 kW of continuous power for the system connected by these wires.
- In a nonregenerative system the resulting maximum power is approx. 25 kW.
- Use fast-acting semiconductor fuses for protection of the UV(R) 1xxD on the primary side.
- The length of the unit bus ribbon cable must not exceed 1 m!
- If necessary, place the MS 110 or MS 111 in the "second row" in the center of the UMs.
- When calculating the length of the ribbon cables, make sure to include the module width of the MS110 or MS111.

Basic circuit diagram for double-row configuration:

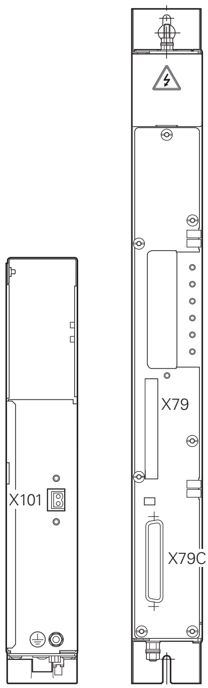


1.13.3 Connection overview

MS 110 pin layout	Connector	Function	Page
	DC-link conductor bar	400 V– to 750 V–	
	Screw fastening	For fastening the PWM round cable shields	
	X79	Unit bus (connection for ribbon cable)	163
	X79C	Unit bus (connection for round cable)	164
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

MS 111 pin layout	Connector	Function	Page
	DC-link conductor bar	400 V– to 750 V–	
	Screw fastening	For fastening the PWM round cable shields	
	X79	Unit bus (connection for ribbon cable)	163
	X79C	Unit bus (connection for round cable)	164
	LED	+24-V supply for fans is available	
	X101	Connection for +24-V supply for fans	165
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

X79: Unit bus Connection:

40-pin ribbon connector	Pin layout	
1a to 3b	0 V *1	These voltages must not be linked with other voltages (only basic insulation)!
4a	+24 V *1	
4b	+24 V *1	
5a	+15 V *1	
5b	+24 V *1	
6a	+15 V *1	
6b	+15 V *1	
7a to 8b	Do not assign	
9a	Reserved (SDA)	
9b	Do not assign	
10a	Reserved (SCL)	
10b	$\overline{\text{ERR.TEMP}}$	
11a	$\overline{\text{PF.PS}}$	
11b	0 V	
12a	$\overline{\text{RES.PS}}$	
12b	0 V	
13a	$\overline{\text{PWR.OFF}}$	
13b	0 V	
14a	5 V FS (spindle enable)	
14b	0 V	
15a	5 V FA (axis enable)	
15b to 16b	0 V	
17a and 17b	–15 V	
18a and 18b	+15 V	
19a to 20b	+5 V	

Danger

The interface complies with the requirements of EN 61800-5-1 for low voltage electrical separation (except for 1a to 6b).

X79C: Unit bus

Connection:

<table>
<tr><th>Round cable connector 37-pin</th><th>Pin layout</th><th></th></tr>
<tr><td>17 -19, 35 -37</td><td>0 V *1</td><td rowspan="7">These voltages must not be linked with other voltages (only basic insulation)!</td></tr>
<tr><td>16</td><td>+24 V *1</td></tr>
<tr><td>34</td><td>+24 V *1</td></tr>
<tr><td>15</td><td>+15 V *1</td></tr>
<tr><td>33</td><td>+24 V *1</td></tr>
<tr><td>14</td><td>+15 V *1</td></tr>
<tr><td>32</td><td>+15 V *1</td></tr>
<tr><td>12, 13</td><td>Do not assign</td><td></td></tr>
<tr><td>11</td><td>Reserved (SDA)</td><td></td></tr>
<tr><td>30, 31</td><td>Do not assign</td><td></td></tr>
<tr><td>29</td><td>Reserved (SCL)</td><td></td></tr>
<tr><td>10</td><td>$\overline{\text{ERR.TEMP}}$</td><td></td></tr>
<tr><td>28</td><td>$\overline{\text{PF.PS}}$</td><td></td></tr>
<tr><td>9</td><td>0 V</td><td></td></tr>
<tr><td>27</td><td>$\overline{\text{RES.PS}}$</td><td></td></tr>
<tr><td>0</td><td>0 V</td><td></td></tr>
<tr><td>26</td><td>$\overline{\text{PWR.OFF}}$</td><td></td></tr>
<tr><td>25</td><td>0 V</td><td></td></tr>
<tr><td>7</td><td>5 V FS (spindle enable)</td><td></td></tr>
<tr><td>6</td><td>5 V FA (axis enable)</td><td></td></tr>
<tr><td>5, 23, 24</td><td>0 V</td><td></td></tr>
<tr><td>4, 22</td><td>–15 V</td><td></td></tr>
<tr><td>3, 21</td><td>+15 V</td><td></td></tr>
<tr><td>1, 2, 20</td><td>+5 V</td><td></td></tr>
</table>

Danger

The interface complies with the requirements of EN 61800-5-1 for low voltage electrical separation (except for 1a to 6b).

X101: Power supply Pin layout:

Connecting terminals	Pin layout
+	+24 V NC
–	0 V NC

1.14 MB 620 Machine Operating Panel

MB 620

MB 620:

- HSCI interface
- 21 snap-on (exchangeable) keys
  The key functions are freely definable via the PLC
- 8 PLC inputs and 8 PLC outputs
- Two holes for additional keys or keylock switches
- Weight: 1.2 kg
- Fulfills IP54 degree of protection when installed

Controls and displays:

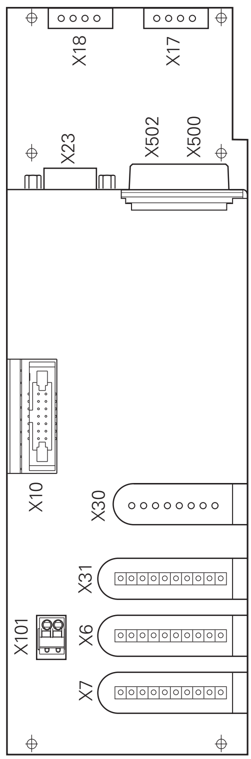
- 12 axis keys
- 16 function keys
- NC Start[1]
- NC Stop[1]
- EMERGENCY STOP button
- Control voltage On[1]

1) Keys illuminated

ID 617 973-xx

For machines with up to six axes, HEIDENHAIN offers the MB 620 machine operating panel with HSCI interface.

On the underside of the machine operating panel are terminal strips bearing the PLC inputs as well as the PLC outputs. Also, connection X23 (175) for HR handwheels is on the underside of the MB 620.



Pin layout for MB 620	Connector	Function	Page
	X17	Emergency stop (MB)	168
	X18	Emergency stop (MB)	168
	X500	HSCI output	50
	X502	HSCI input	50
	X6	PLC inputs	169
	X7	PLC outputs	168
	X10	Interface to keyboard and potentiometers	171
	X23	Handwheel connection	175
	X30	Connection for handwheel adapter	172
	X31	Permissive button, NC start, NC stop	170
	X101	24 V NC power supply	171
	⏚	Protective ground	

Danger

Please note that the outputs of connector X7 are powered internally by +24 V NC, and therefore supply +24 V NC at HIGH level.

For the entire HSCI system, the +24 V NC supply voltage is required to be safely separated voltage. The +24 V NC supply voltage must not, under any circumstances, be connected with the +24 V PLC supply voltage, because this removes the double basic insulation.

Each of the switching outputs at X7 supplies up to 150 mA of output current and are provided for driving the lamps on the MP 620.

X17/X18: EMERGENCY STOP on MB

Connectors X17 and X18 are electrically parallel.

In the MB 6xx without FS, the connectors X17 and X18 do not support dual-channel evaluation of the EMERGENCY STOP button. These inputs must therefore not be used for evaluating EMERGENCY STOP!

Note

With the MB 620 without FS, the EMERGENCY STOP must be wired externally in the EMERGENCY STOP chain as before.

X7: PLC outputs

Pin layout:

Connecting terminals	Pin layout
1	O0 (illumination for the NC Start key)[a]
2	O1 (illumination for the NC Stop key)[a]
3	O2 (illumination for the Control voltage ON key)[a]
4	O3
5	O4
6	O5
7	O6
8	O7
9	+24 V NC (available here)
10	0 V NC (available here)

a. With standard wiring

Ampacity of the outputs: Maximum 150 mA per output

Danger

Please note that the outputs of connector X7 are powered internally by +24 V NC, and therefore supply +24 V NC at HIGH level.

For the entire HSCI system, the +24 V NC supply voltage is required to be safely separated voltage. The +24 V NC supply voltage must not, under any circumstances, be connected with the +24 V PLC supply voltage, because this removes the double basic insulation.

Each of the switching outputs at X7 supplies up to 150 mA of output current and are provided for driving the lamps on the MP 620.

X6: PLC inputs

Pin layout for MB 620 without FS:

Connecting terminals	Pin layout
1	I0
2	I1
3	I2 (control voltage ON, CVO)[a]
4	I3
5	I4
6	I5
7	I6
8	I7
9	Reserved (do not use)
10	Reserved (do not use)

a. With standard wiring

Danger

Please note that the MB 620 is powered by +24 V NC.

For the entire HSCI system, the +24 V NC power supply voltage is required to be safely separated voltage. It must also be safely separated from the +24 V PLC!

Danger

Please note that the outputs of connector X7 are powered internally by +24 V NC, and therefore supply +24 V NC at HIGH level.

For the entire HSCI system, the +24 V NC supply voltage is required to be safely separated voltage. The +24 V NC supply voltage must not, under any circumstances, be connected with the +24 V PLC supply voltage, because this removes the double basic insulation.

Each of the switching outputs at X7 supplies up to 150 mA of output current and are provided for driving the lamps on the MP 620.

X31: MB 620, permissive button / NC Start / NC Stop

Standard wiring of connection X31 for MB 620 without FS:

Connecting terminals	Pin layout
1	Reserved (do not use)
2	Reserved (do not use)
3	Reserved (do not use)
4	Reserved (do not use)
5	NC start[a]
6	Reserved (do not use)
7	NC start power supply (+24-V NC)[a]
8	NC stop [a]
9	Reserved (do not use)
10	NC stop power supply (+24-V NC)[a]

a. With standard wiring

NC Start and NC Stop are normally-open contacts on the MB 620 (FS).

Danger

Please note that the MB 620 is powered by +24 V NC.

For the entire HSCI system, the +24 V NC supply voltage is required to be safely separated voltage. The +24 V NC supply voltage must not, under any circumstances, be connected with the +24 V PLC supply voltage, because this removes the double basic insulation.

X10: Interface to keyboard and potentiometers

Pin layout:

Connecting terminals	Pin layout
1a	Potentiometer 1
2a	Potentiometer 3
3a	Do not assign
4a	Do not assign
5a	Do not assign
6a	+ 5 V
7a	0 V
1b	Potentiometer 2
2b	- - -
3b	Do not assign
4b	Do not assign
5b	- - -
6b	+ 5 V
7b	0 V

X101: Power supply

Pin layout:

Connecting terminals	Pin layout
1	+24 V NC
2	0 V NC

Power consumption of the operating panel units without HR handwheel and controlled inputs/outputs:

Power consumption of the MB 620:4.0 W

Power consumption of the PLB 6001:5.0 W

X30: Handwheel connection, permissive button / emergency stop

Connection X30 not used on MB 620 without FS.

Note

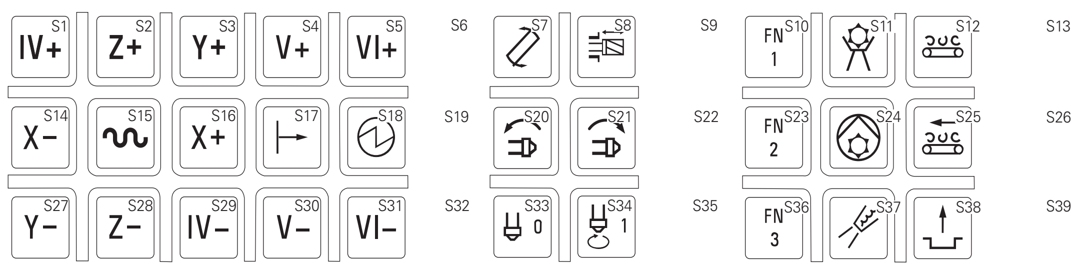
With the MB 620 without Functional Safety (FS), the permissive buttons and the EMERGENCY STOP of the handwheel must be wired externally in corresponding safety circuits as before.

Danger

Please note that the MB 620 is powered by +24 V NC.

For the entire HSCI system, the +24 V NC supply voltage is required to be safely separated voltage. The +24 V NC supply voltage must not, under any circumstances, be connected with the +24 V PLC supply voltage, because this removes the double basic insulation.

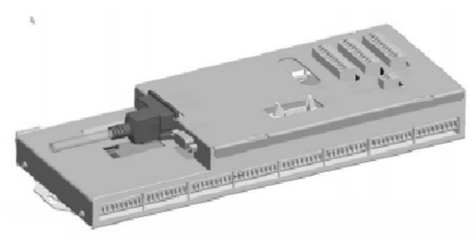
Machine operating panel: Key assignment



1.15 HSCI Adapter for PLB 6001 OEM-Specific Machine Operating Panel

<table>
<tr><th colspan="2">PLB 6001</th></tr>
<tr><td>

- HSCI interface
- Connection for HR xxx
- 64 PLC inputs, 32 PLC outputs for keys / key illumination
- Connection for spindle-speed and feed-rate override potentiometer
- Screw fastening or top-hat-rail mounting
- Weight: 1.2 kg

ID 668 792-xx
</td><td></td></tr>
</table>

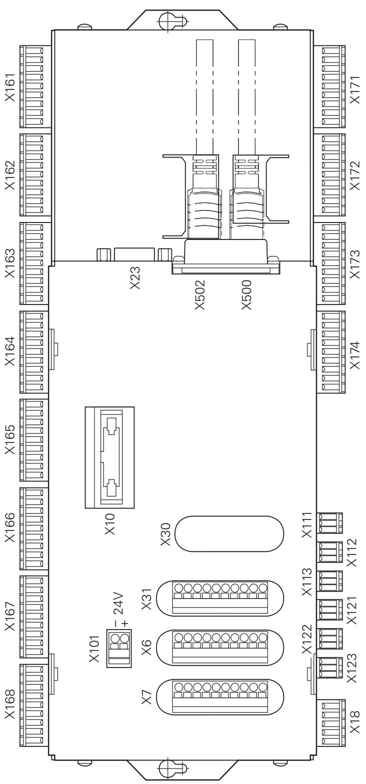
HEIDENHAIN offers the PLB 6001 HSCI adapter with HSCI interface for connecting an OEM-specific machine operating panel.

On the underside of the machine operating panel are terminal strips bearing the PLC inputs as well as the PLC outputs. Also, connection X23 (175) for HR handwheels is on the underside of the PLB 6001.

Danger

Please note that the PLB 6001 is supplied with +24 V NC.

For the entire HSCI system, the +24 V NC supply voltage is required to be safely separated voltage. The +24 V NC supply voltage must not, under any circumstances, be connected with the +24 V PLC supply voltage, because this removes the double basic insulation.

PLB 6001 pin layout	Connector	Function	Page
	X500	HSCI output	50
	X502	HSCI input	50
	X6	PLC inputs	169
	X7	PLC outputs	168
	X10	Interface to keyboard and potentiometers	171
	X18	Reserved	
	X23	Handwheel connection	175
	X30	Reserved	
	X31	Permissive button, NC start, NC stop	170
	X101	24 V NC power supply	171
	X111	Potentiometer connection 1	176
	X112	Potentiometer connection 2	176
	X113	Potentiometer connection 3	176
	X121	Potentiometer connection 4	176
	X122	Potentiometer connection 5	176
	X123	Potentiometer connection 6	176
	X161	PLC inputs I0 to I7	175
	X162	PLC inputs I8 to I15	175
	X163	PLC inputs I16 to I23	175
	X164	PLC inputs I24 to I31	175
	X165	PLC inputs I32 to I39	175
	X166	PLC inputs I40 to I47	175
	X167	PLC inputs I48 to I55	175
	X168	PLC inputs I56 to I63	175
	X171	PLC outputs O0 to O7	176
	X172	PLC outputs O8 to O15	176
	X173	PLC outputs O16 to O23	176
	X174	PLC outputs O24 to O31	176
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

X23: Handwheel connection

> Danger
>
> The connector for the handwheel on the machine operating panel, as well as the connector on the handwheel itself, may be removed only by trained and qualified personnel, even if it can be removed without using a tool.
> If the handwheel connector is removed, only basic insulation from line power (230 V) is provided!

X161 to X168: PLC inputs

X161: PLC inputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	0 V PLC	I0	I1	I2	I3	I4	I5	I6	I7

X162: PLC inputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	0 V PLC	I8	I9	I10	I11	I12	I13	I14	I15

X163: PLC inputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	0 V PLC	I16	I17	I18	I19	I20	I21	I22	I23

X164: PLC inputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	0 V PLC	I24	I25	I26	I27	I28	I29	I30	I31

X165: PLC inputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	0 V PLC	I32	I33	I34	I35	I36	I37	I38	I39

X166: PLC inputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	0 V PLC	I40	I41	I42	I43	I44	I45	I46	I47

X167: PLC inputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	0 V PLC	I48	I49	I50	I51	I52	I53	I54	I55

X168: PLC inputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	0 V PLC	I56	I57	I58	I59	I60	I61	I62	I63

X171 to X174: PLC outputs

X171: PLC outputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	O0	O1	O2	O3	O4	O5	O6	O7	24 V PLC for O0 to O7

X172: PLC outputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	O8	O9	O10	O11	O12	O13	O14	O15	24 V PLC for O8 to O15

X173: PLC outputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	O16	O17	O18	O19	O20	O21	O22	O23	24 V PLC for O16 to O23

X174: PLC outputs									
Pin layout	**Terminal**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
PL 6001	O24	O25	O26	O27	O28	O29	O30	O31	24 V PLC for O24 to O31

X111 to X123: Potentiometer connection

Pin layout:

Connecting terminals	Pin layout
1	0 V potentiometer
2	Potentiometer arm
3	+5 V potentiometer

1.16 TE 620 Keyboard Unit

General information

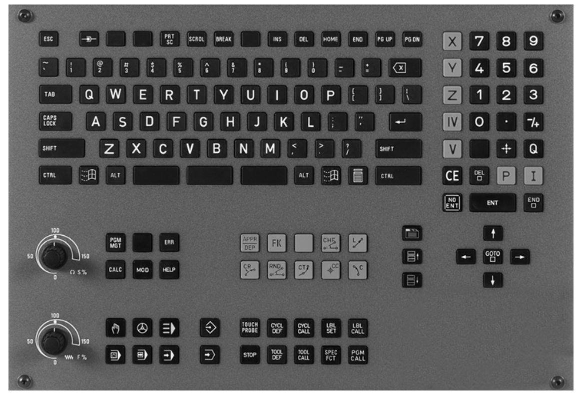
New NC operating panel without touchpad

Technical characteristics:

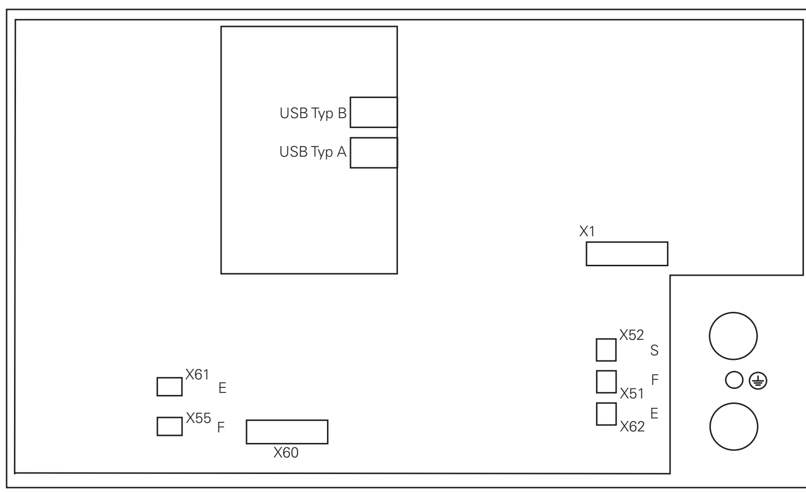
NC operating panel:
Same features as TE 630

- The IV and V keys, the blank key to the left of V and the blank key above MOD in the operating panel are snap-ons.
- USB interface
- Weight: 2.2 kg
- Fulfills IP54 degree of protection when installed

ID 625 806-xx TE 620



Connection overview



Connector	Function
USB Type B	USB connection to BF 2xx. The TE 6xx is connected to the USB hub of the BF 2xx. The maximum permissible cable length for this USB connection is 3 m. The USB hub of the TE 6xx only supports USB 1.1.
USB Type A	Freely available USB connection (USB 1.1) Maximum load capacity of USB output: 1 x 100 mA
X1	Screen soft keys of BF 2xx X3 via ribbon cable to keyboard unit X1
X60	Potentiometer of keyboard unit via ribbon cable to machine operating panel X10
X52 (S)	Connection for spindle-speed override potentiometer

X51 (F)	Connection for feed-rate override potentiometer
X62 (E)	Connection for rapid-traverse override potentiometer, alternative to X61 (not with TE 63x)
X55	Connection for feed-rate override potentiometer, alternative to X51 (not with TE 63x)
X61	Connection for rapid-traverse override potentiometer
⏚	Protective ground (M5)

The TE keyboard units are supplied with power via the USB port.

1.17 TE 630 Keyboard Unit

General information

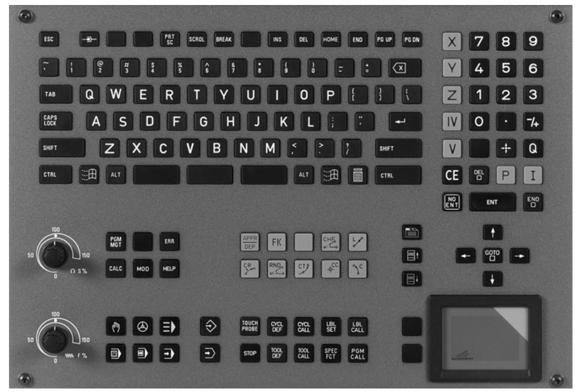
New NC operating panel with touchpad

Technical characteristics:

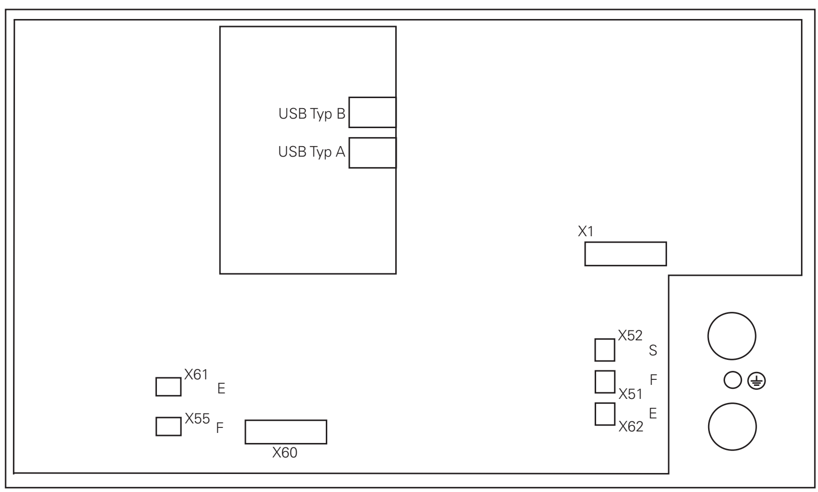
NC operating panel:
Same features as TE 620

- The IV and V keys, the blank key to the left of V and the blank key above MOD in the operating panel are snap-ons.
- USB interface
- Weight: 2.2 kg
- Fulfills IP54 degree of protection when installed

ID 617 976-xx TE 630



Connection overview



Connector	Function
USB Type B	USB connection to BF 2xx. The TE 6xx is connected to the USB hub of the BF 2xx. The maximum permissible cable length for this USB connection is 3 m. The USB hub of the TE 6xx only supports USB 1.1.
USB Type A	Freely available USB connection (USB 1.1) Maximum load capacity of USB output: 1 x 100 mA
X1	Screen soft keys of BF 2xx X3 via ribbon cable to keyboard unit X1
X60	Potentiometer of keyboard unit via ribbon cable to machine operating panel X10
X52 (S)	Connection for spindle-speed override potentiometer

X51 (F)	Connection for feed-rate override potentiometer
X62 (E)	Connection for rapid-traverse override potentiometer, alternative to X61 (not with TE 63x)
X55	Connection for feed-rate override potentiometer, alternative to X51 (not with TE 63x)
X61	Connection for rapid-traverse override potentiometer
⏚	Protective ground (M5)

The TE keyboard units are supplied with power via the USB port.

Possible configurations for the potentiometers of a TE 6xx

The override potentiometers of a TE 6xx can be configured in machine parameter MP7620 bit#5 and bit#9. If bit#5 is set in MP7620, the potentiometers are evaluated as feed-rate and rapid-traverse override potentiometers. If an HSCI TE keyboard unit with three override potentiometers (spindle, rapid traverse and feed rate) is used, bit#9 must be set in MP7620. If bit#5 and bit#9 are both set at the same time, an error message is triggered due to invalid configuration.

For the evaluation of the potentiometers, the following distinctions are made between HSCI and non-HSCI hardware:

- TE for systems without HSCI
  Configuration with feed-rate and rapid-traverse overrides (MP7620, bit#5). The spindle-speed override is used as rapid-traverse override. The NC keys are used to realize the spindle-speed override. If hardware for systems without HSCI is used, bit#9 is ignored.
- TE for systems without HSCI
  Configuration for three potentiometers with feed-rate, spindle-speed and rapid-traverse overrides (MP7620, bit#5). This configuration has no effect. The potentiometers are evaluated, as usual, as feed-rate and spindle-speed overrides. Bit#9 is ignored.
- TE for systems with HSCI
  Configuration with feed-rate and rapid-traverse overrides (MP7620, bit#5). The spindle-speed override input **S** (X52) on the rear of the TE is not evaluated. The rapid-traverse override must be connected to the provided connector **E** (X61, X62). There are no NC keys for spindle-speed override. Machine operating panel keys must be used instead, and the function must be realized in the PLC program (the PLC program must transmit the information to W764).
  The + and 100% keys for spindle-speed override can be snapped on to an HSCI machine operating panel. Unlike with keyboards for systems without HSCI, these keys are no longer available as NC keys, but as PLC inputs. They are not evaluated automatically by the NC software. The PLC program must evaluate the keys and transmit the calculated value to the NC (W764; spindle-speed override). The NC software limits this value depending on MP3310.0 (max. spindle-speed override value) and MP3310.1 (minimum spindle-speed override value).
- TE with three potentiometers for systems with HSCI
  Configuration for three potentiometers with feed-rate, spindle-speed and rapid-traverse overrides (MP7620, bit#9): Feed-rate, spindle-speed and rapid-traverse overrides are evaluated.

1.18 TE 635Q Keyboard Unit

General information

The new operating panel is the TE 630 plus the machine operating keys.

Technical characteristics:

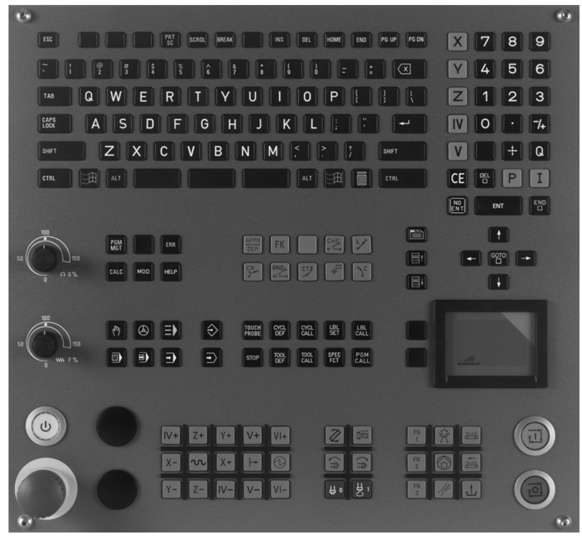
- Weight: 3.3 kg
- Fulfills IP54 degree of protection when installed

NC operating panel:
Same features as TE 630

Machine operating panel:

- 6 axis-direction keys
- 16 function keys
- Keys for NC start and NC stop (illuminated)
- Keys for spindle start and spindle stop
- All keys in the machine operating panel are snap-on keys.
- EMERGENCY STOP key
- Key for control voltage ON (RAFI key, illuminated)
- Two bore holes (22 mm) for additional RAFI buttons (shipped blocked with a cover) or keylock switches
- HSCI interface

ID 617 975-xx TE 635Q



Machine operating panel

The description of the TE 635Q machine operating panel is the same as that of the MB 620, see page 166.

Danger

Please note that the TE 635Q is powered by +24-V NC.

For the entire HSCI system, the +24 V NC supply voltage is required to be safely separated voltage. The +24 V NC supply voltage must not, under any circumstances, be connected with the +24 V PLC supply voltage, because this removes the double basic insulation.

Keyboard unit

The description of the TE 635Q control panel corresponds to the description of the TE 630, see page 179.

1.19 BF 250 15-Inch TFT Screen

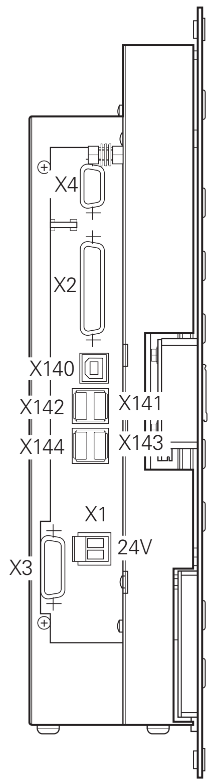
General information

Thanks to its HDL interface, the new 15-inch TFT screen is compatible with all MC 6xxx main computers with HDL output.

Technical characteristics:

- Resolution: 1024 x 768 pixels
- 1 x 8 soft keys, 1 x 6 soft keys for PLC
- 3 soft keys for switching soft-key rows
- Key for screen layout and operating mode switching
- Additional USB interface (USB 2.0) on the front of the visual display unit
- Integrated USB hub (USB 2.0) with 4 USB interfaces on back of the visual display unit
- HDL connection
- Weight: 5.2 kg
- Fulfills IP54 degree of protection when installed

ID 599 916-xx BF 250

<table>
<tr><td colspan="4">BF 250 15-inch TFT screen</td></tr>
<tr><th>Pin layout</th><th>Connector</th><th>Function</th><th>Page</th></tr>
<tr><td rowspan="7">
</td><td>X4</td><td>Reserved</td><td></td></tr>
<tr><td>X2</td><td>HDL connection</td><td>184</td></tr>
<tr><td>X140</td><td>USB input (Type A)</td><td>184</td></tr>
<tr><td>X141 to X144</td><td>USB 2.0 output (Type A)</td><td>184</td></tr>
<tr><td>X1</td><td>+24 V power supply</td><td>184</td></tr>
<tr><td>X3</td><td>Connection for soft keys</td><td>184</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

X1: +24 V power supply for BF 2xx

Connection for the +24 V power supply of the BF 2xx. The screen can be powered by +24 V NC or +24 V PLC through the integrated power supply unit of the BF 2xx.

Power supply: Minimum absolute value: +20.4 V–
Maximum absolute value: +28.8 V–

Pin layout:

Connecting terminals at X1	Pin layout
+ / 1	+24 V NC
– / 2	0 V NC

Power consumption of the BF 250: 50 W

X2: HDL connection

Port for the HEIDENHAIN display link (HDL) connection for the MC 6xxx connector X249, see page 83.

X3: Connection of screen soft keys

Connection of the screen soft keys to the TE 6xx keyboard unit connector X1 via ribbon cable.

X140: USB input

Connection for the integrated USB hub to the MC 6xxx connector X141 or X142. The USB hub of the BF 2xx supports USB 2.0.

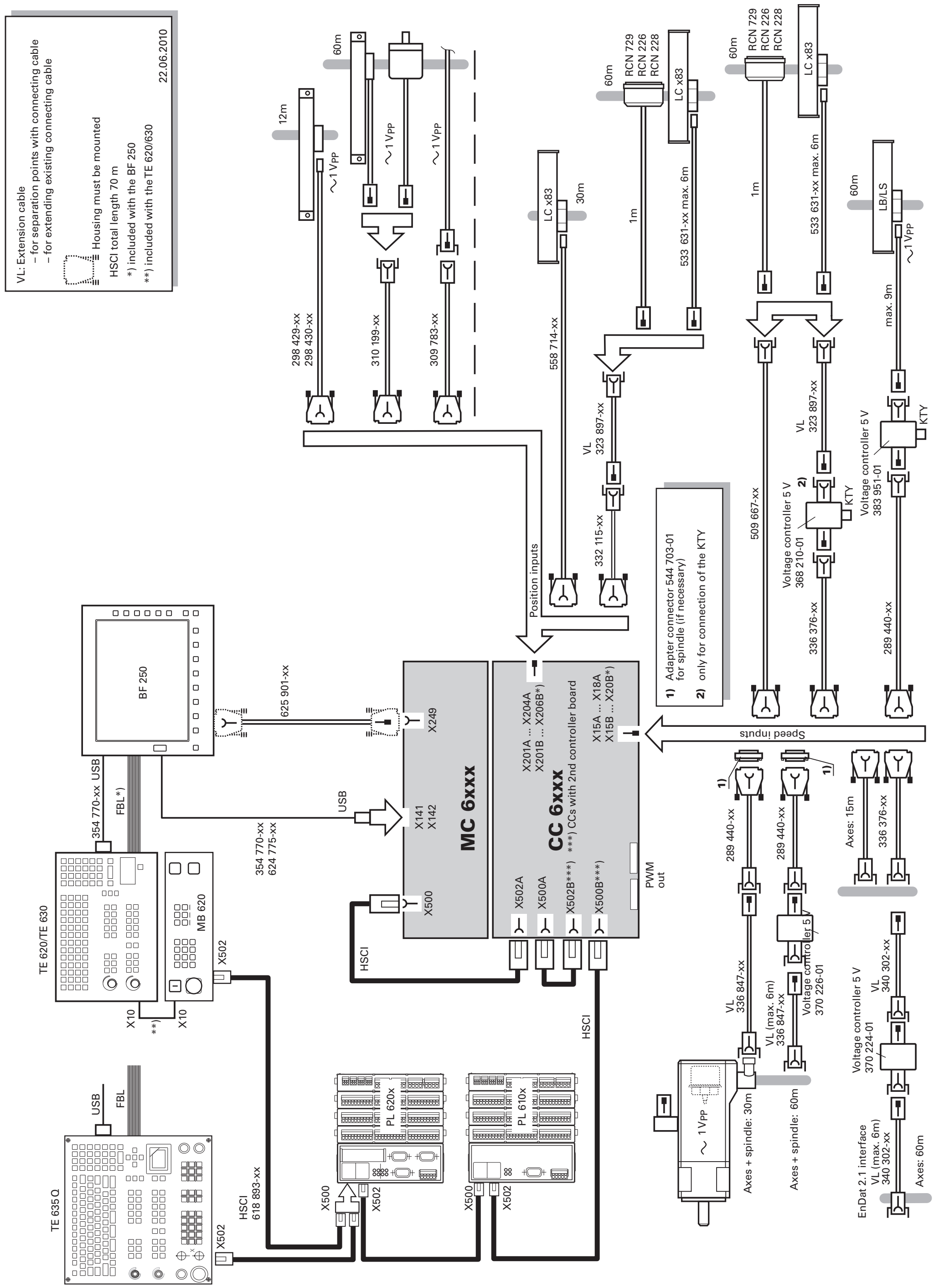
X141 to X146: USB output

BF 250: Five freely available connections for additional USB devices. One of them on the front of the BF 250.

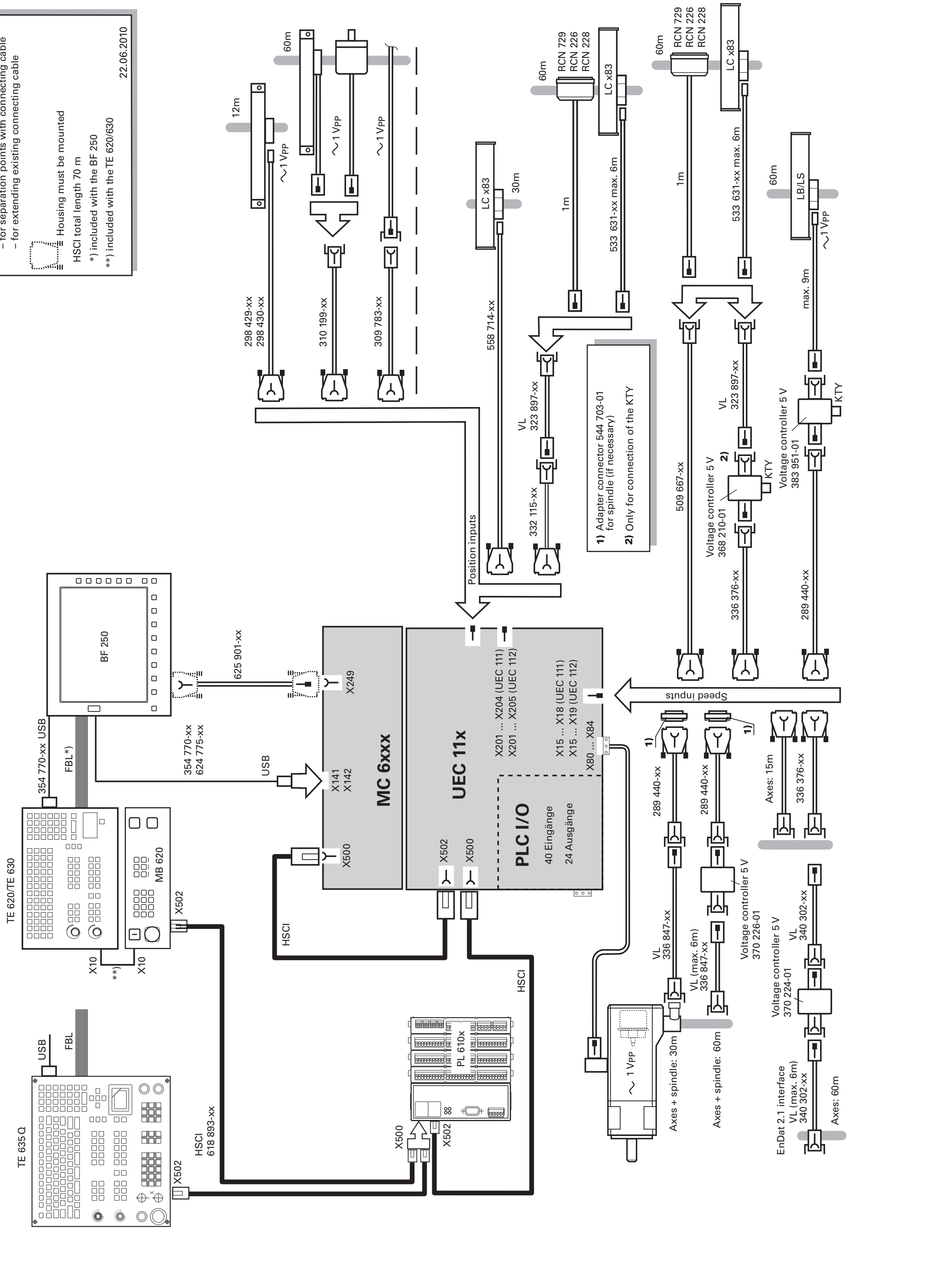
Maximum load of the five USB outputs of the BF 250:
4 x 500 mA, 1 x 100 mA, distributed as desired

1.20 Cable Overviews

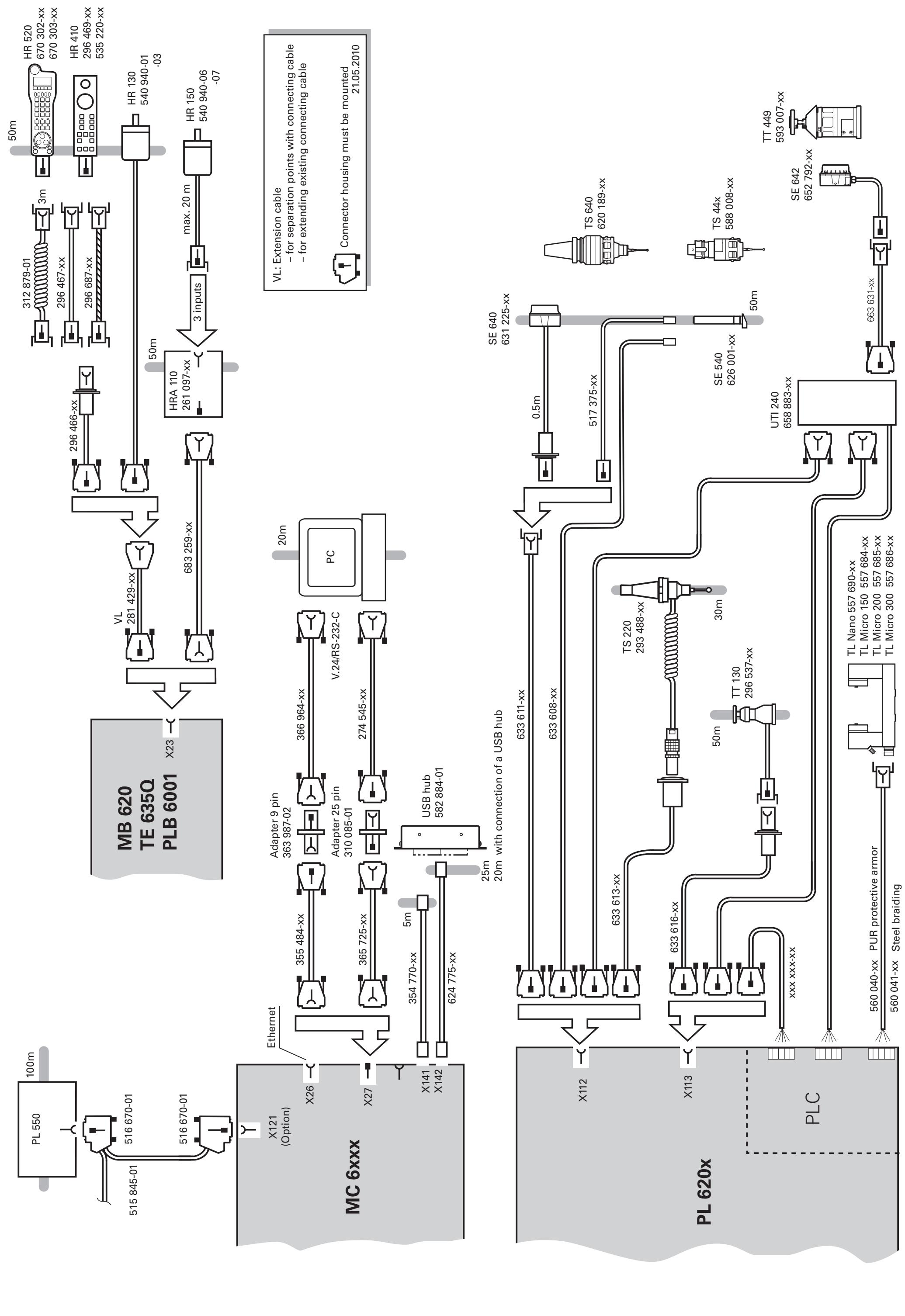
1.20.1 Control Systems with CC; MC in Electrical Cabinet



Basic configuration



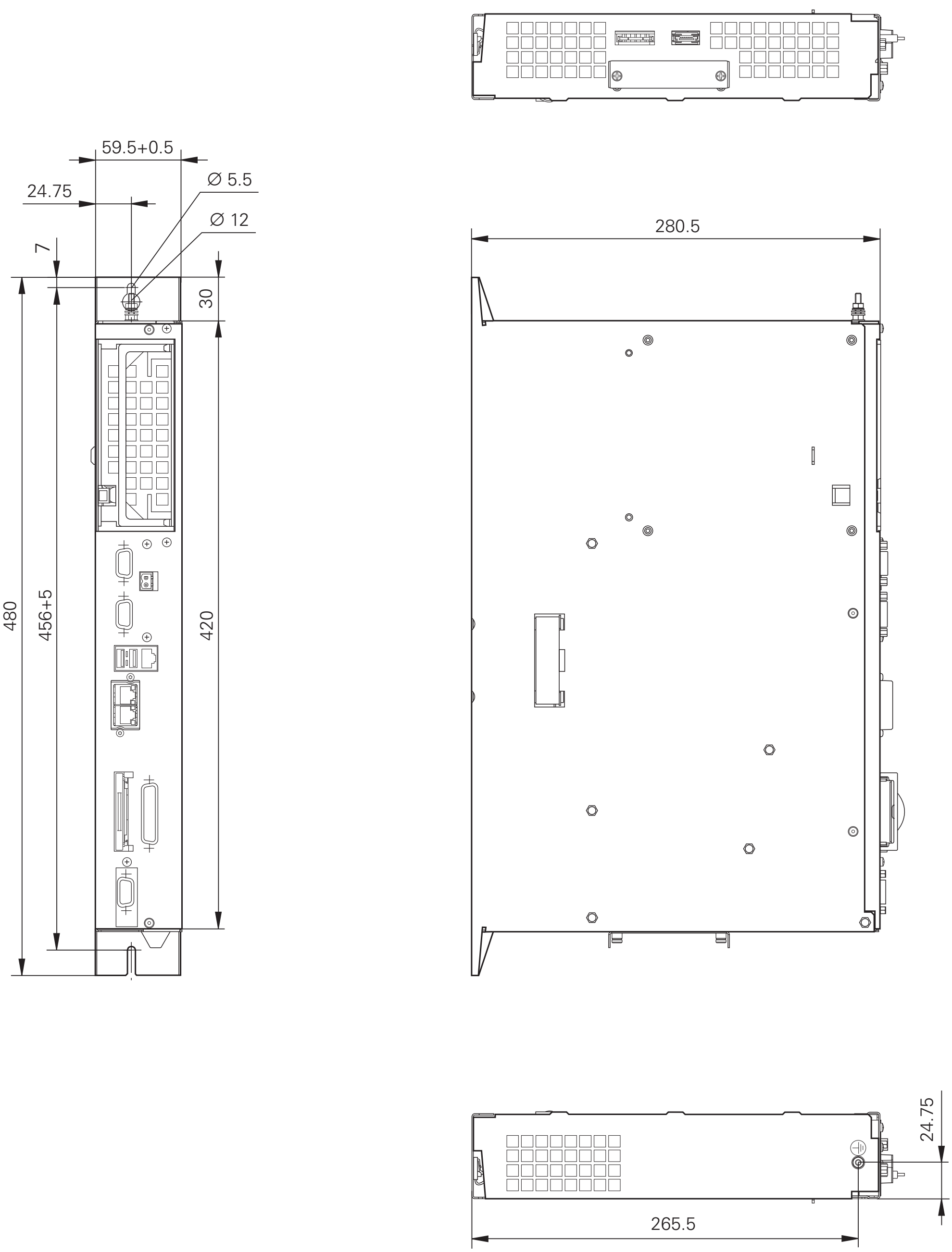
1.20.3 Accessories



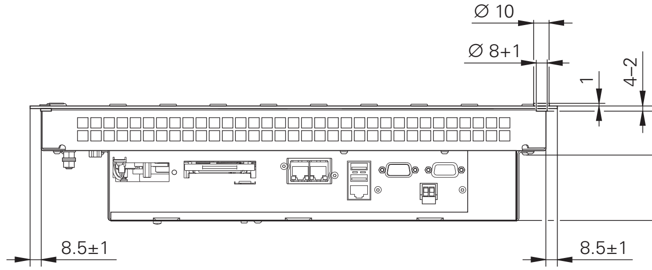
1.21 Dimensions

> **Note**
>
> All dimensions are in millimeters [mm].

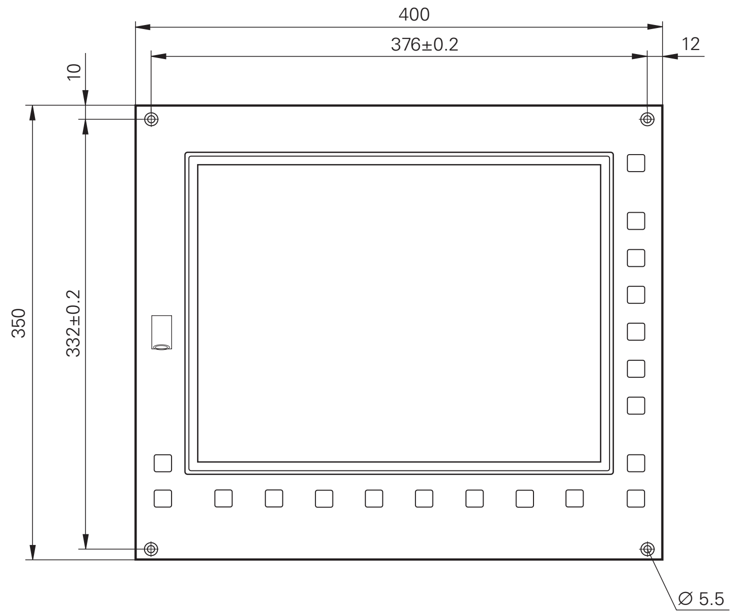
1.21.1 MC 6x41



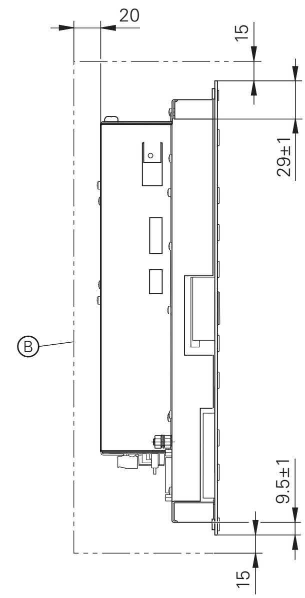
1.21.2 MC 6222



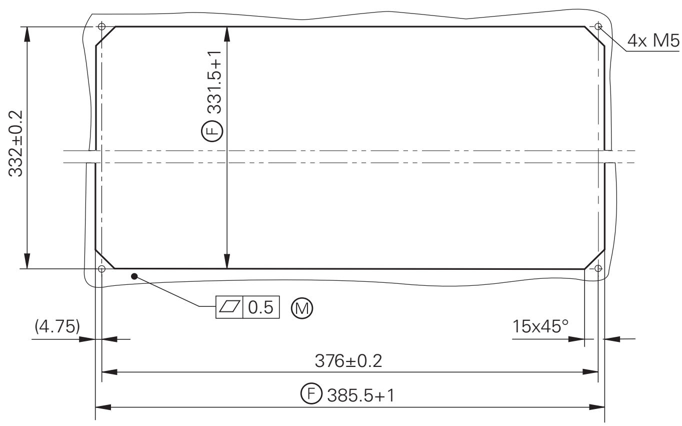
Ø 10
Ø 8+1
1
4–2
34+2
84+2
8.5±1
8.5±1



400
376±0.2
12
10
350
332±0.2
Ø 5.5

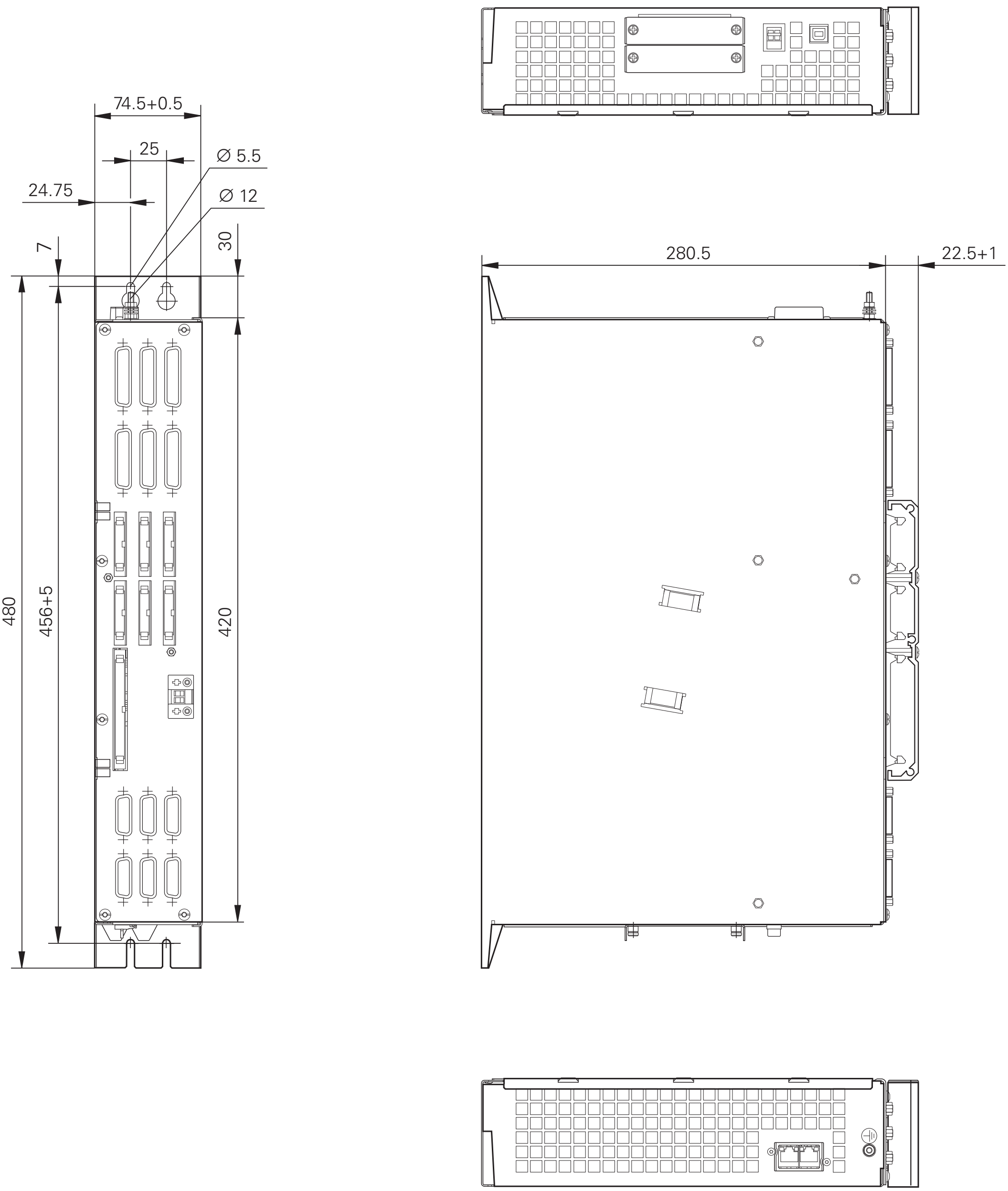


20
15
29±1
B
9.5±1
15

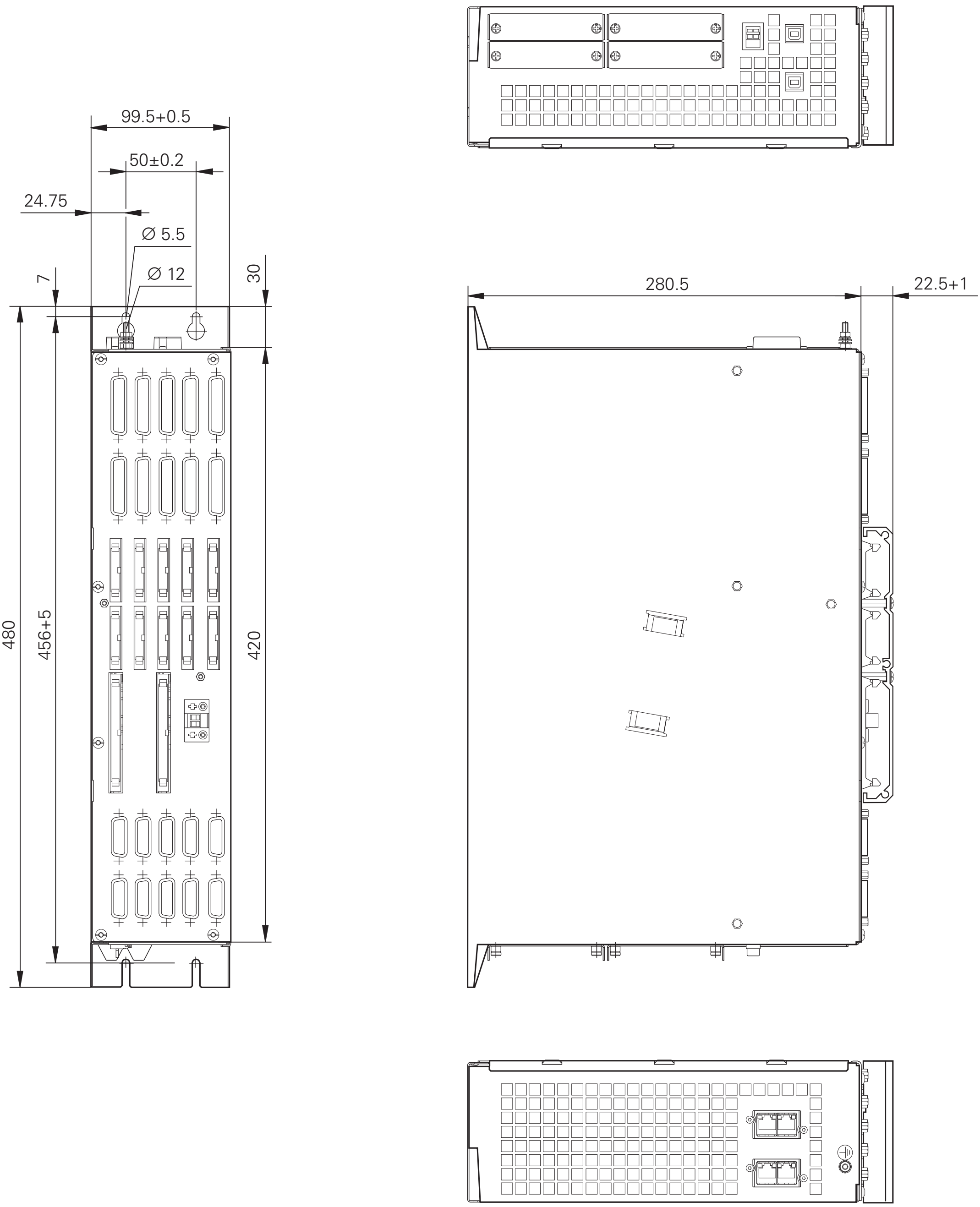


4x M5
332±0.2
F 331.5+1
0.5
M
(4.75)
15x45°
376±0.2
F 385.5+1

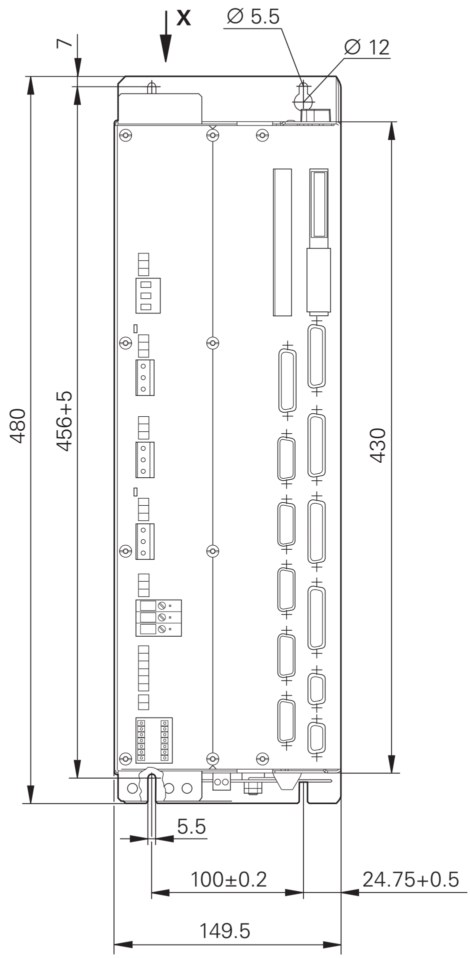
1.21.3 CC 6106



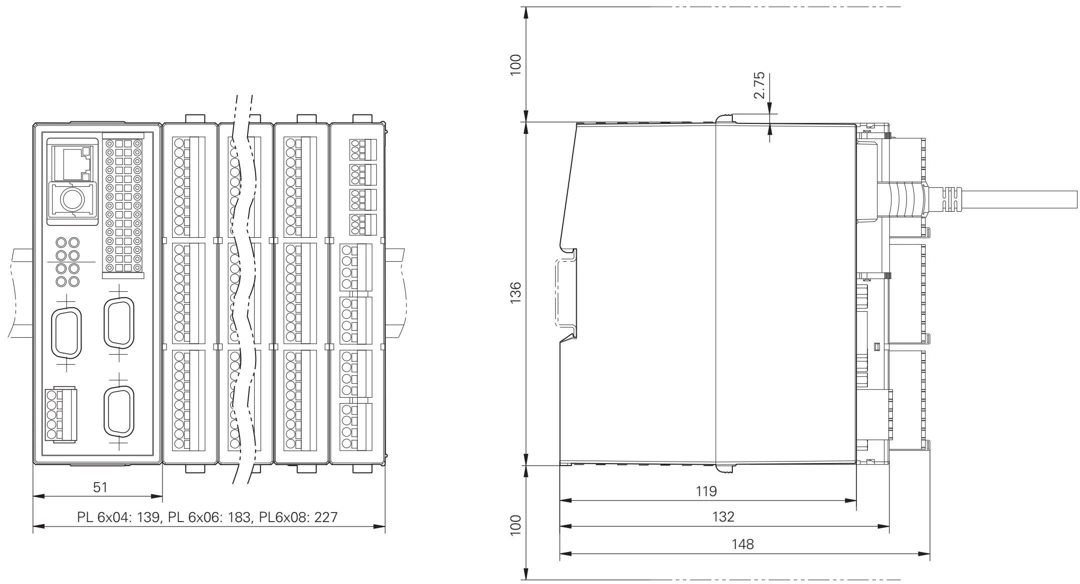
1.21.4 CC 6108 / CC 6110



1.21.5 UEC 11x

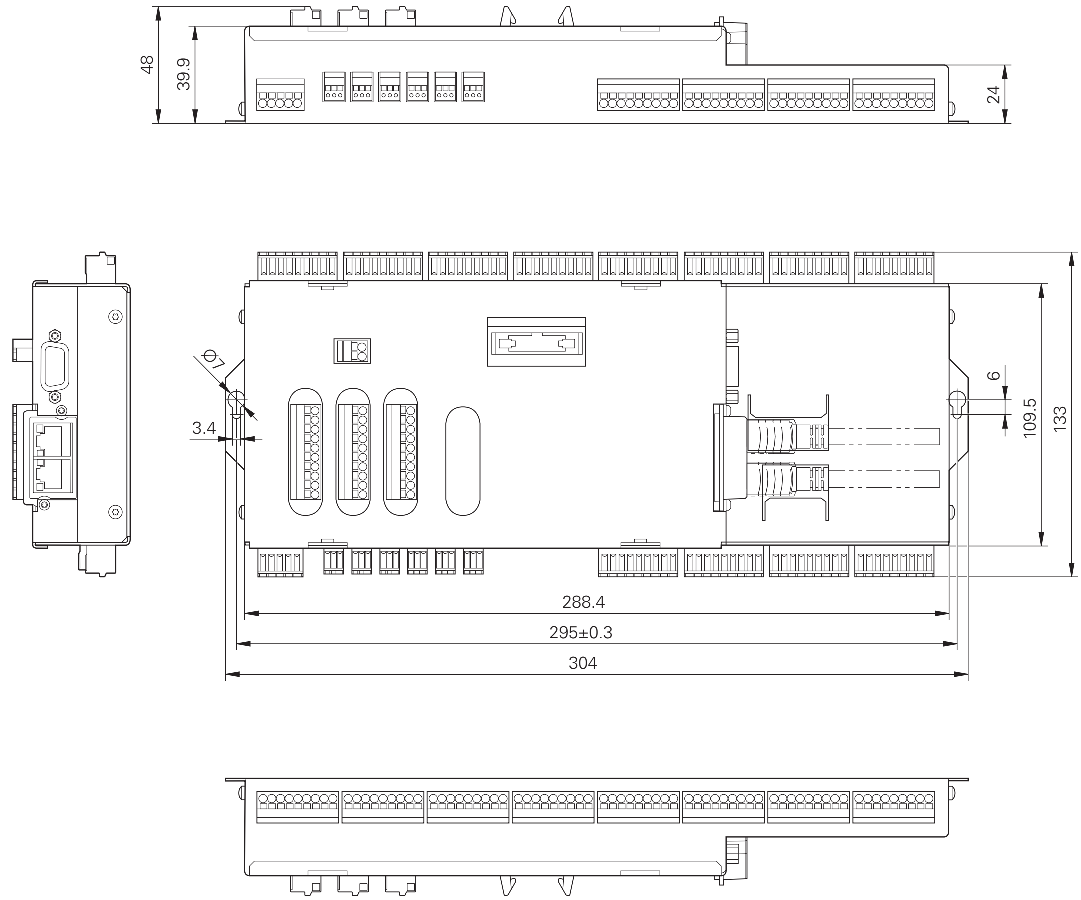


1.21.6 PLB 6xxx (FS)

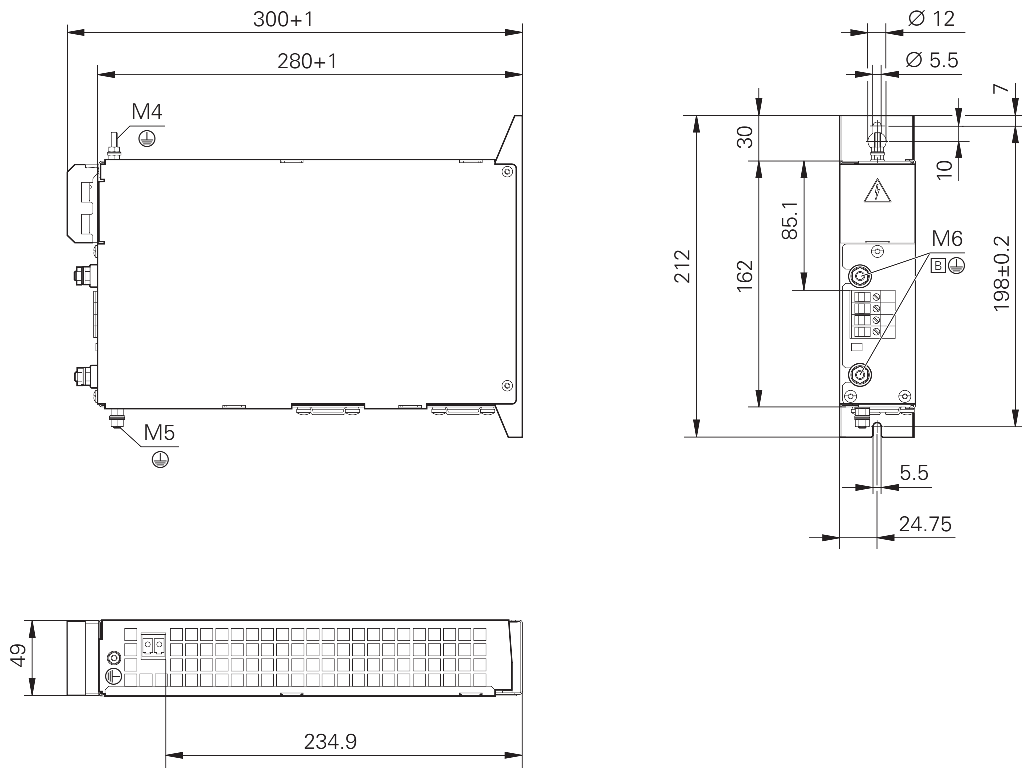


Dashed line: Space for air circulation

1.21.7 PLB 6001 (FS)

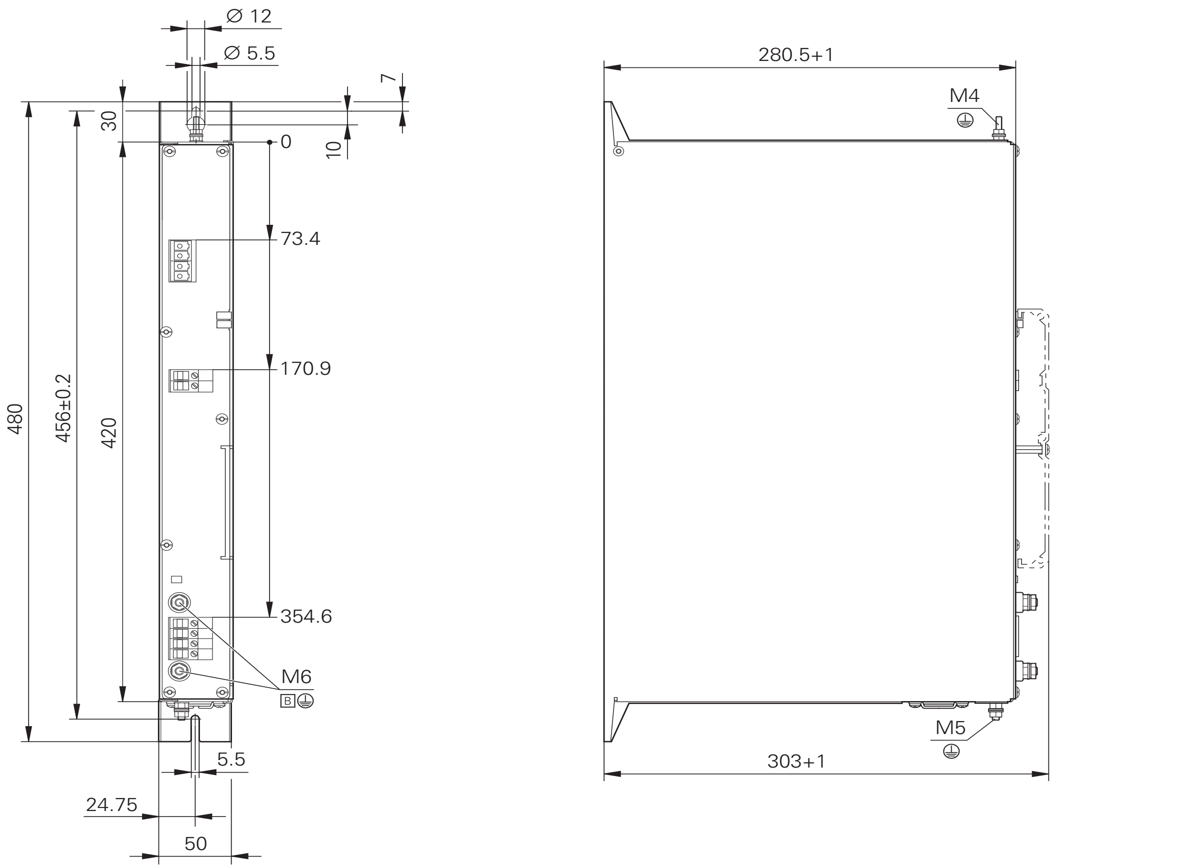


1.21.8 PSL 130



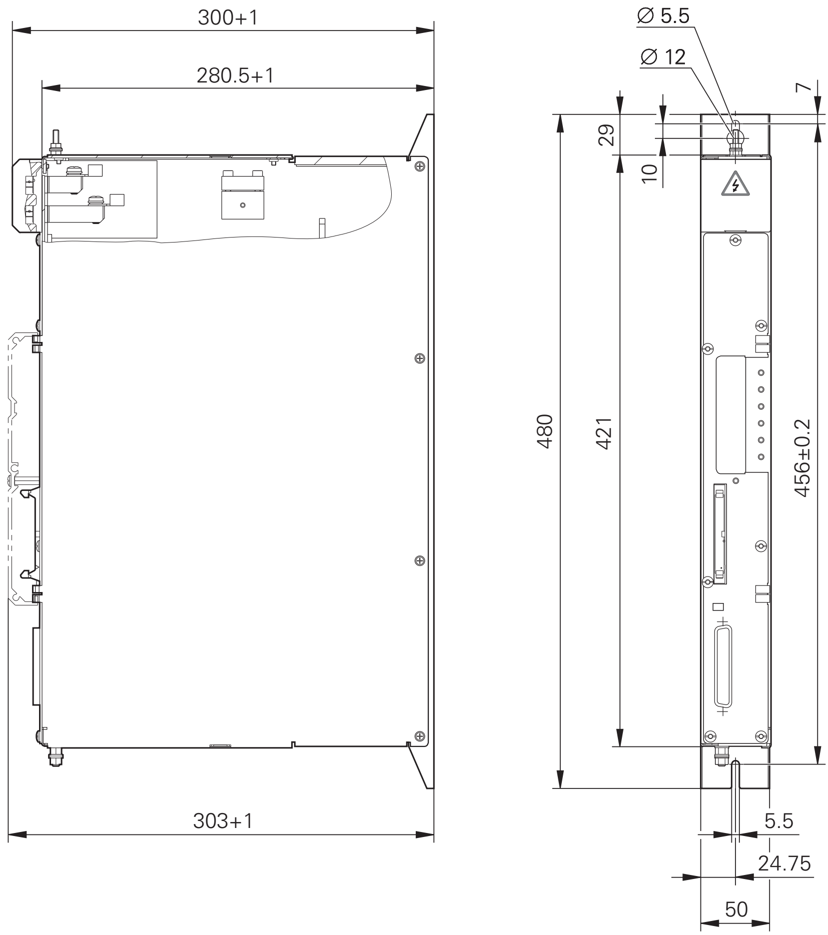
300+1
280+1
M4
M5
Ø 12
Ø 5.5
7
30
10
85.1
M6
B
212
162
198±0.2
5.5
24.75
49
234.9

1.21.9 PSL 135

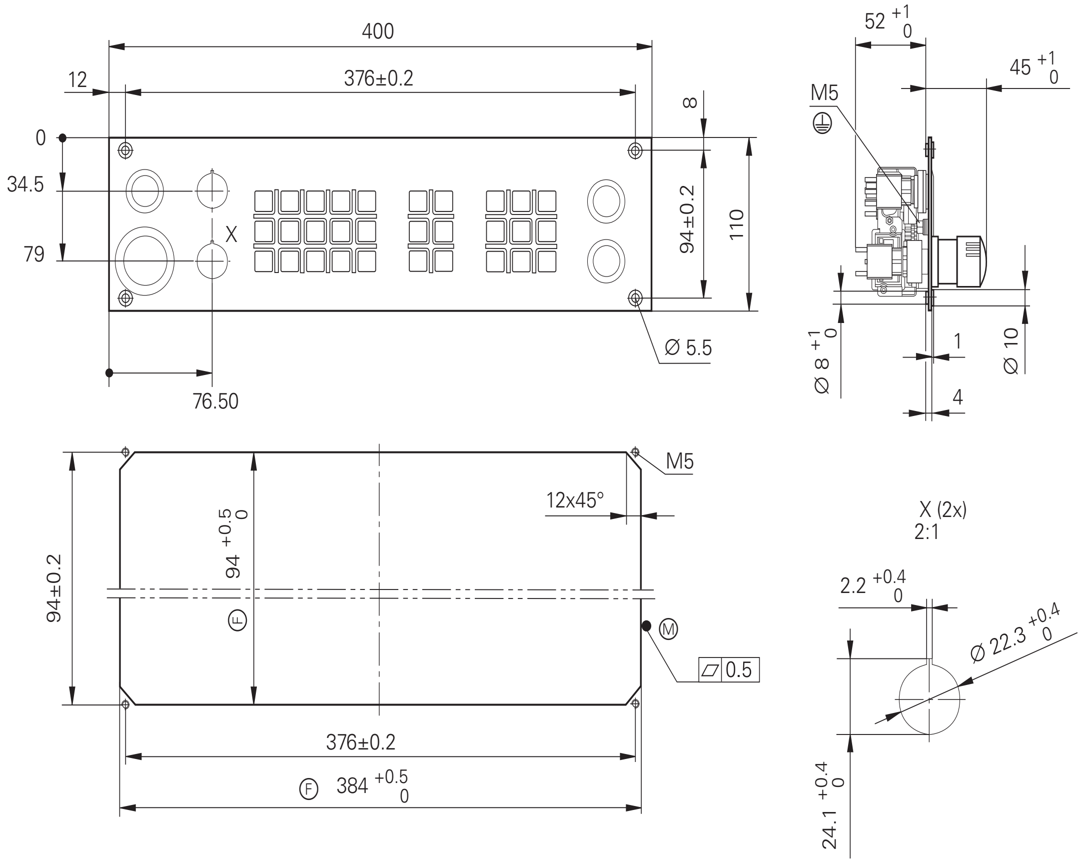


Ø 12
Ø 5.5
7
30
10
0
73.4
170.9
354.6
480
456±0.2
420
M6
5.5
24.75
50
280.5+1
M4
M5
303+1

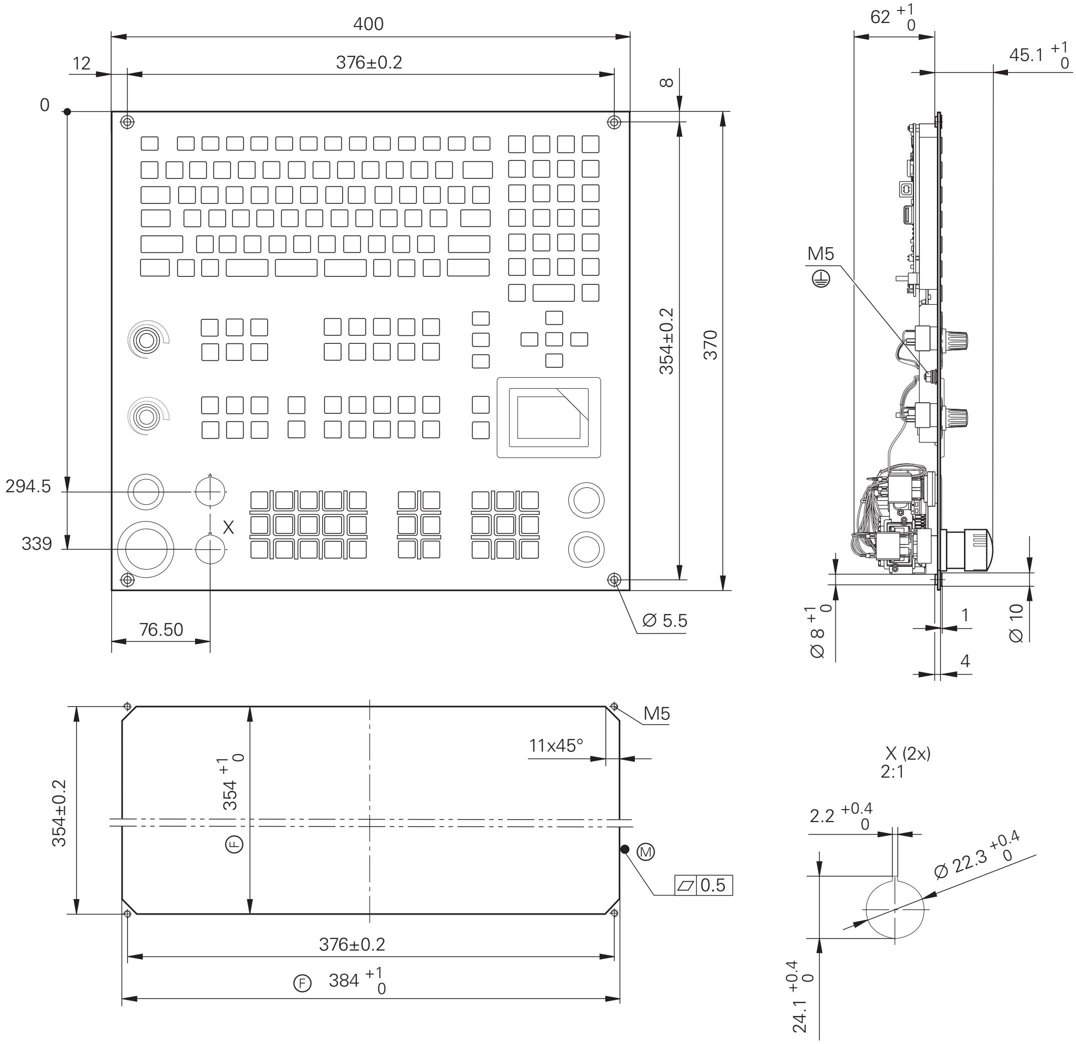
1.21.10 MS 11x



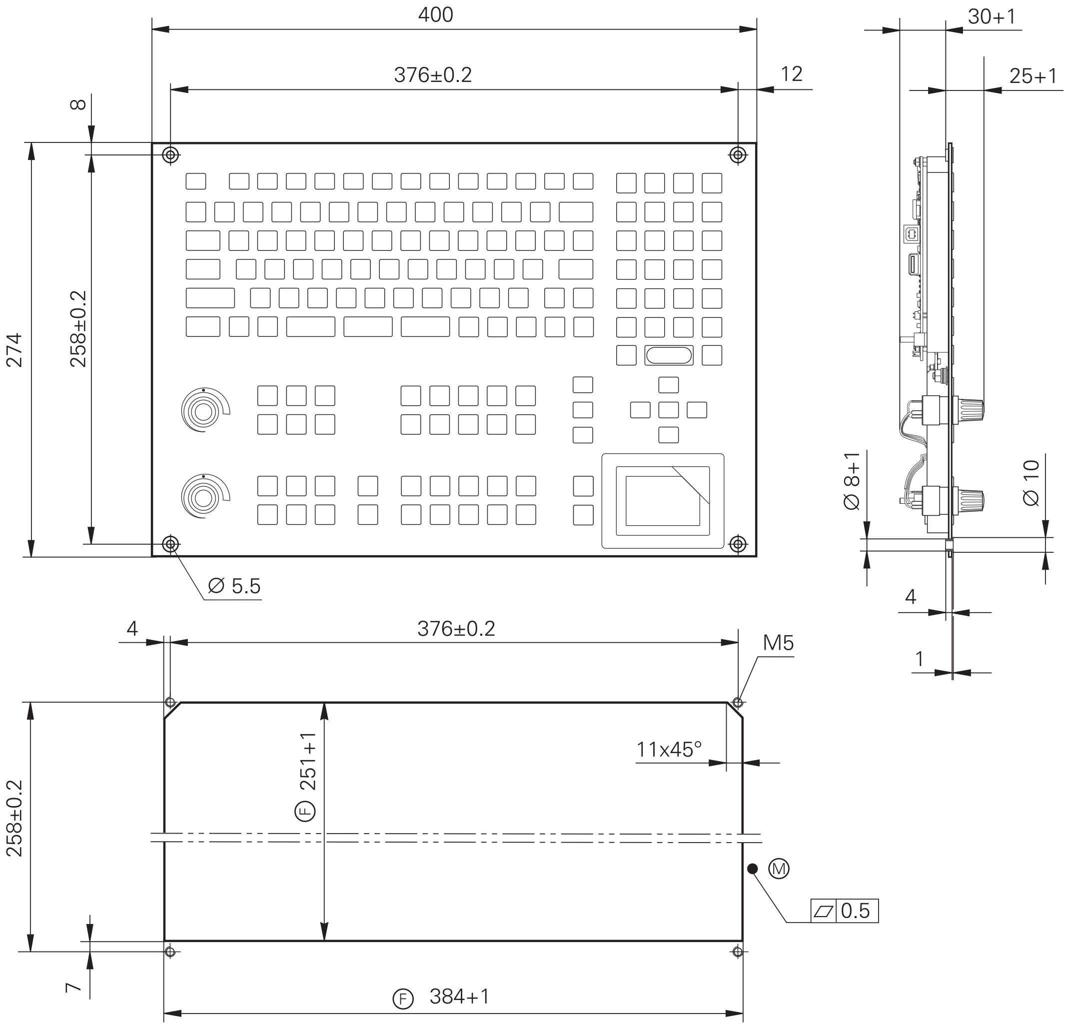
1.21.11 MB 620


400
12
376±0.2
0
34.5
79
X
8
94±0.2
110
Ø 5.5
76.50
52 +1 0
45 +1 0
M5
Ø 8 +1 0
1
Ø 10
4
M5
12x45°
94 +0.5 0
94±0.2
F
M
0.5
376±0.2
F 384 +0.5 0
X (2x)
2:1
2.2 +0.4 0
Ø 22.3 +0.4 0
24.1 +0.4 0

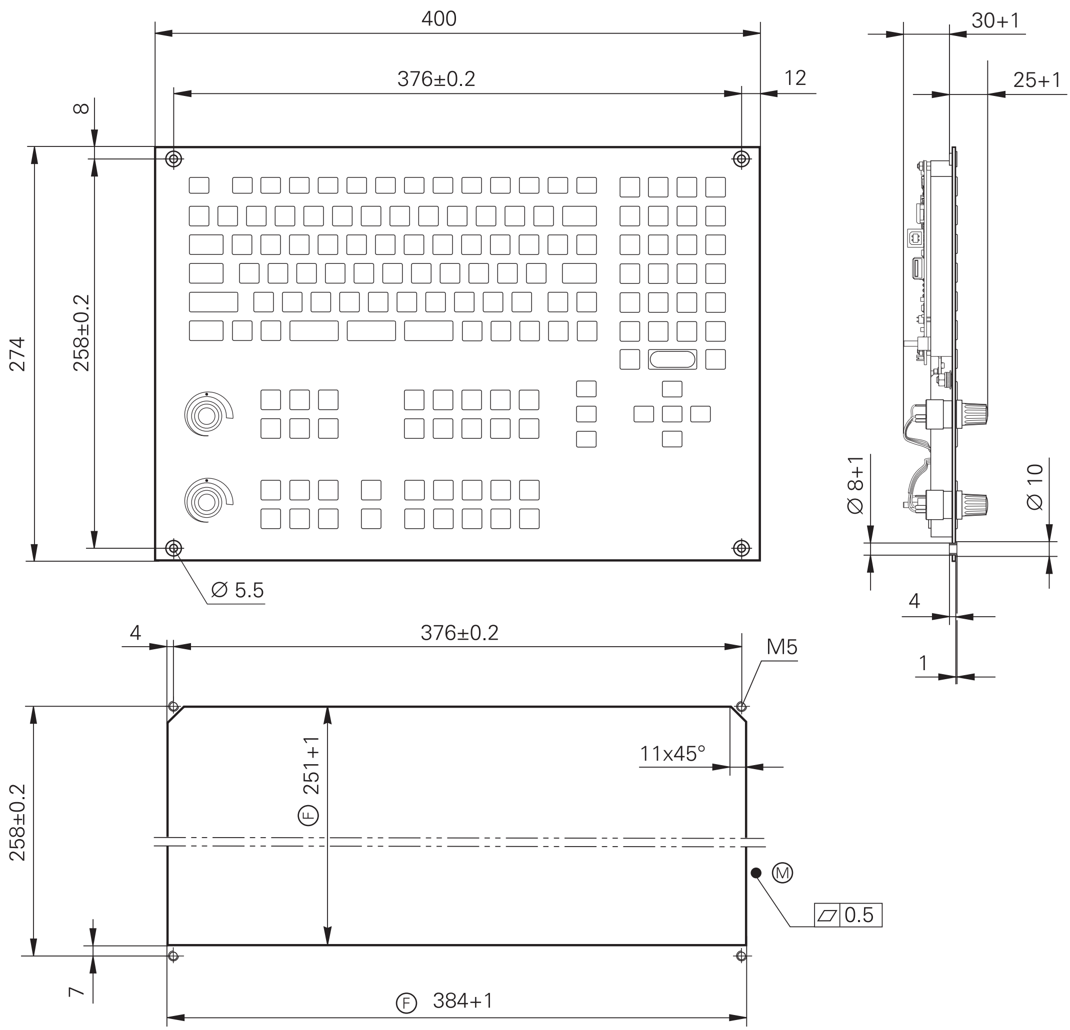
1.21.12 TE 535Q



1.21.13 TE 630

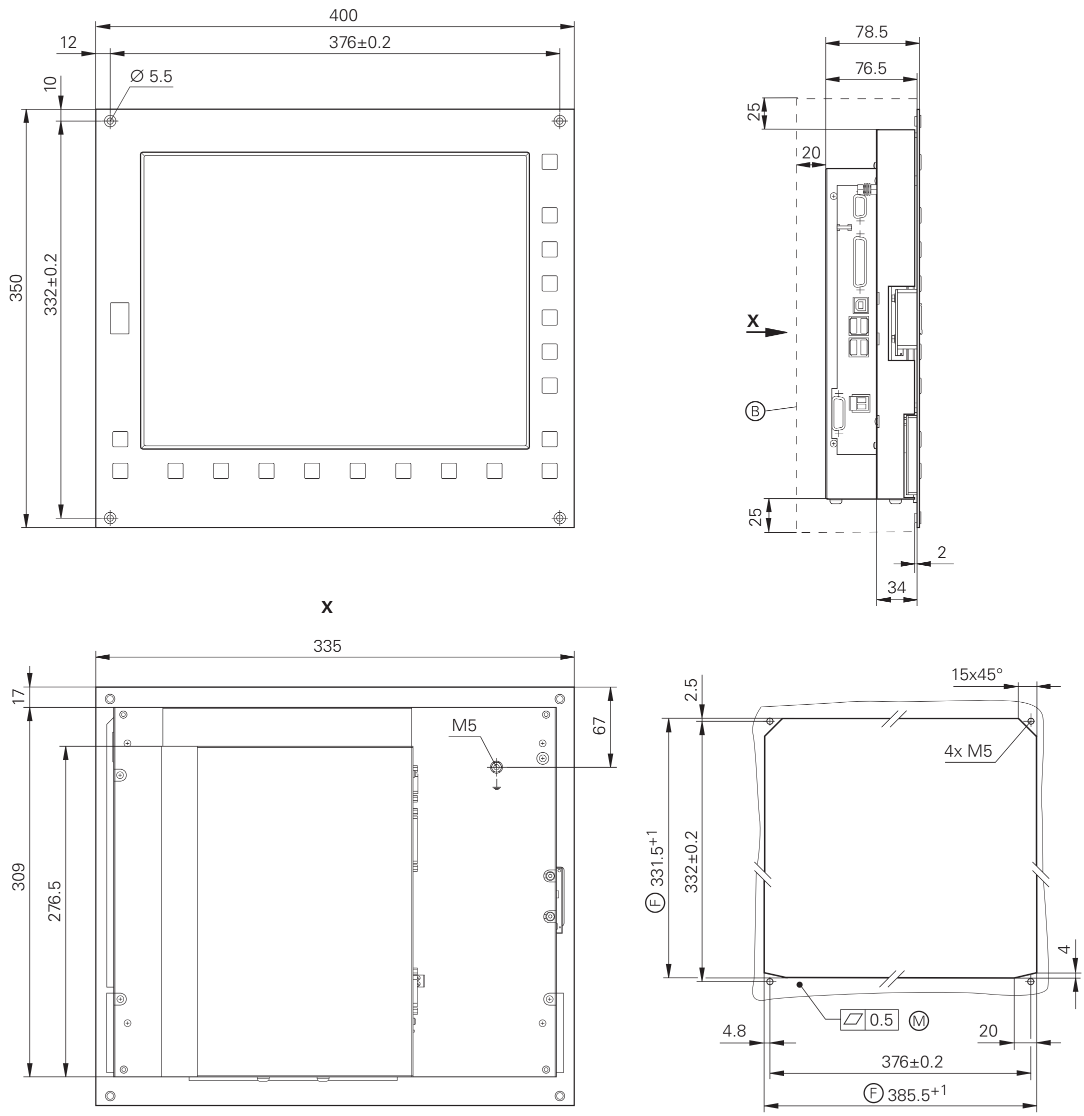

400
376±0.2
12
8
274
258±0.2
Ø 5.5
30+1
25+1
Ø 8+1
Ø 10
4
1
4
376±0.2
M5
11x45°
251+1
258±0.2
0.5
7
384+1

1.21.14 TE 620



400
376±0.2
12
8
274
258±0.2
Ø 5.5
30+1
25+1
Ø 8+1
Ø 10
4
1
4
376±0.2
M5
11x45°
258±0.2
F 251+1
M
0.5
7
F 384+1

1.21.15 BF 250



B = Space for air circulation

F = Front panel opening

M = Mounting surface

1 Update Information No. 26 – Revised Version

1.1 Overview

1.1.1 Released service packs

The following service packs were released for **340 49x-05**:

- Service pack 6: May 2010

The following service packs were released for **340 49x-06**:

- No service pack up to now

The following service packs were released for **606 42x-01**:

- Service pack 1: August 2010

1.2 NC Software 340 49x-xx, 606 42x-01

1.2.1 Important notes

Changed behavior of markers for "control-in-operation" (control-in-operation symbol):

Attention

The changed behavior might result in damage to the machine!

The marker M4176 (control-in-operation symbol is on or blinking) is handled differently by the NC software as of the software versions listed below. This removes ambiguity of the marker's function, but the PLC programs must be adapted to the new behavior.

Affected software versions:

- NC software 340 49x-05 with service pack 06
- NC software 340 49x-06 without service pack
- NC software 606 42x-01 without service pack or with service pack 01

Previous behavior:

The marker M4175 (control-in-operation symbol is blinking) and the marker M4176 (control-in-operation symbol is on or blinking) were cleared simultaneously at an internal stop. The two markers remained reset even during a subsequent RUNCANCEL macro (NCMACRO.SYS). The markers signalized that the control was not in operation although the RUNCANCEL macro was still running.

New behavior with the above-mentioned software versions:

The marker M4175 (control-in-operation symbol is blinking) and the marker M4176 (control-in-operation symbol is on or blinking) are no longer cleared simultaneously at an internal stop. The marker M4176 remains set during a subsequent RUNCANCEL macro, whereas marker M4175 is already reset. This shows that the control is still in operation during the RUNCANCEL macro. The marker M4176 is not cleared until execution of the RUNCANCEL macro has been completed.

HEIDENHAIN recommends:

The changed behavior can cause problems in step sequences (e.g. tool change) in the PLC program. If you have used the markers M4175 and M4176 in the PLC program to initialize step sequences at an internal stop, the changed behavior might lead to the step sequences not being reset as expected. This can lead to an inconsistent condition and might cause damage to the machine.

The change also affects the basic PLC program from HEIDENHAIN. Machine manufacturers who are using the basic program must modify the affected program section in the module LIBRARY.SRC as follows:

```
[LIBRARY.SRC]
#define /s    ML_manual_mode_interruption                M
#define /s    ML_internal_STOP                           M
...
;External/Internal STOP
L       NP_M4176_control_in_operation_on_or_flashes ;row changed
R       NP_M4185_internal_STOP                      ;row inserted

L       NP_M4221_PET_table_NC_stop                  ;row inserted
ON      PN_M4560_NC_stop_0_active                   ;row inserted
A       NP_M4176_control_in_operation_on_or_flashes ;row inserted
A       MG_manual_mode                              ;row inserted
S       ML_manual_mode_interruption                 ;row inserted

L       MG_pulse_internal_stop                      ;row inserted
O       MG_automatic_mode                           ;row inserted
R       ML_manual_mode_interruption                 ;row inserted

L       NP_M4185_internal_STOP                      ;row inserted
AN      ML_internal_STOP                            ;row inserted
O[                                                  ;row inserted
 L       ML_manual_mode_interruption                ;row inserted
 AN      NP_M4176_control_in_operation_on_or_flashes;row inserted
]                                                   ;row inserted
=       MG_pulse_internal_stop                      ;row inserted

L       NP_M4185_internal_STOP                      ;row inserted
=       ML_internal_STOP                            ;row inserted

L       NP_M4175_control_in_operation_flashes
AN      ML_control_operates_flashes
AN      NP_M4158_mode_block_scan
=       MG_pulse_external_stop

L       NP_M4176_control_in_operation_on_or_flashes
AN      NP_M4175_control_in_operation_flashes
=       MG_control_in_operation_on
=N      MG_release_PLC_strobe

L       MG_control_in_operation_on
A       ML_control_operates_flashes
AN      NP_M4157_mode_restore_position
AN      NP_M4193_RUNCANCEL_macro_active             ;row inserted
=       MG_NC_start_pulse
```

```
;IFT instruction for MG_BA_Manual has been removed
;two rows from the IFT instruction remain
L       NP_M4175_control_in_operation_flashes
=       ML_control_operates_flashes

L       ML_mode_restore_position
AN      MG_key_NC_stop
AN      NP_M4221_PET_table_NC_stop
R       MG_pulse_external_stop

L       NP_M4157_mode_restore_position
=       ML_mode_restore_position

;NC STOP / NC START
...
```

Attention

- Check the functions of the markers M4175 and M4176 in your PLC program.
- If required, modify the PLC program with respect to the changed behavior of the markers.
- Test the behavior of the PLC program on the machine.
- If required, conduct an update of the PLC program on the affected machines in the field.

If you need assistance in evaluating the situation, please contact the responsible HEIDENHAIN service agency.

Further action:

With the following service packs, the previous behavior of the markers (see above) is reintroduced as default behavior:

- Software 340 49x-05 as of service pack 07
- Software 340 49x-06 as of service pack 02
- Software 606 42x-01 as of service pack 02

The service packs are scheduled to be available for download from our FileBase in October 2010.

Note

There will be no service pack 01 for NC software 340 49x-06. The first service pack of NC software 340 49x-06 will be SP 02.

With the above-mentioned service packs, a new error message will be displayed in the event of error. If – due to the resetting of marker M4176 (control-in-operation symbol is on or blinking) while the RUNCANCEL macro is running – impermissible PLC reactions (e.g. range switchover, PLC positioning movements, overwriting of machine parameters) are triggered in the PLC program, the fatal error message "**PLC strobe not allowed (M4193)**" will be issued.

With the above-mentioned service packs or higher, the new PLC marker M4193 (RUNCANCEL macro is active) will be available to adapt the PLC program. The marker is set if a RUNCANCEL macro is active.

Also, bit #17 in Machine Parameter MP7680 will be available with these service packs or higher. This new bit enables you to choose between the previous behavior (bit #17 = 0) and the new behavior (bit #17 = 1) of the markers. MP7680 bit #17:

Input:	0: M4175, M4176 are cleared simultaneously at an internal stop
	1: M4176 remains set during RUNCANCEL macro

If the error message "**PLC strobe not allowed (M4193)**" appears with these service packs or higher, HEIDENHAIN recommends that you change to the new behavior of the markers by settingMP7680 bit #17 and adapting the PLC program. If no error message occurs, the PLC program does not need to be modified.

1 Update Information No. 27

1.1 Overview

1.1.1 Released service packs

The following service packs were released for **340 49x-05**:

- Service pack 6: May 2010

The following service packs were released for **340 49x-06**:

- No service pack up to now

The following service packs were released for **606 42x-01**:

- Service pack 1: August 2010

1.2 NC Software 340 492-xx, 340 493-xx

1.2.1 Important notes

Attention

Freely definable tables are not converted correctly to ASCII during a software update to software version 340 49x-06.
This behavior causes loss of the freely definable tables!

Affected software versions:

- NC software 340 492-03, 340 493-03
- NC software 340 492-04, 340 493-04
- NC software 340 492-05, 340 493-05 up to and including service pack 04

Description of the problem:

The problem occurs only on dual-processor controls with the software versions listed above. Freely definable tables are not converted correctly to ASCII during a software update of the above-mentioned versions to software version 340 49x-06. This cannot be undone and the freely definable tables are lost.

HEIDENHAIN recommends:

The problem can be avoided by taking one of the following three actions:

1. Manual conversion of the files

Manually convert all files to ASCII before running the software update:

- Switch to the **Machine Parameter Programming** mode of operation by entering the code number 95148.
- Press the MOD key and then the **UPDATE DATA** soft key.
- Press the first soft key **Convert Bin -> ASC** to start the manual conversion to ASCII.
- Then you can run the desired software update.

2. Backup of all *.TAB files

- Using TNCremoNT, make a full backup of at least all *.TAB files on the **PLC:** and **TNC:** before the software update.
- Run the desired software update.
- Transfer the *.TAB backup files to the control again.
- After the software update, delete all defective *.TA% files on the **PLC:** and **TNC:**

3. Install service pack 05 or a higher SP of software versions 340 492-05, 340 493-05

- Based on the software versions concerned, first install service pack 05 or higher of software versions 340 492-05, 340 493-05.
  In service packs 05 or higher the problem has been solved, i.e. starting from SP 05 or higher, software versions 340 492-06, 340 493-06 can be installed.
- Run the desired software update.

1 Update Information No. 28

1.1 Overview

1.1.1 Released service packs

The following service packs were released for **340 49x-05**:

- Service pack 06: May 2010

The following service packs were released for **340 49x-06**:

- No service pack up to now

The following service packs were released for **606 42x-01**:

- Service pack 01: August 2010

1.2 NC Software 340 49x, 606 42x

Attention

Starting immediately, please ship all machines with iTNC 530 and software versions 340 49x-06 and 606 42x-01 only with service pack 02!

Software versions 340 49x and 606 42x are affected by the problems described in the following. They concern “Space characters in path and file names" on page 210 and “Brake test for synchronized axes" on page 211.

If you need assistance in evaluating the situation regarding one of these issues, please contact the responsible HEIDENHAIN service agency.

1.2.1 Space characters in path and file names

According to the HEIDENHAIN user documentation, file names must not include space characters. As a rule, this should also apply to path names. If space characters were used nevertheless for path and file names in NC programs, macros, cycles etc., the following problem occurs now in software versions 340 49x-06 and 606 42x-01:

Description of the problem

In the software versions listed below, the space characters in path and file names are misinterpreted. This misinterpretation always occurs after a block scan and restoration of the machine status through macros if any NC program or cycle was executed since the last machine start-up and if the corresponding path or file name includes a space character.
The misinterpretation causes jumps within any program (e.g. CALL LBL) to an incorrect destination. An erroneous jump can trigger an error message or lead to the machining operation being continued at an incorrect spot. This problem can cause damage to the machine, but does not occur with other software versions.

Affected software versions:

- NC software 340 49x-06
- NC software 606 42x-01

HEIDENHAIN recommends:

Starting immediately, please ship all machines with iTNC 530 and software versions 340 49x-06 and 606 42x-01 only with service pack 02! Run a software update on the affected machines in the field, as well.

Further action:

The misinterpretation is eliminated with the following service packs:

- Software 340 49x-06 as of service pack 02
- Software 606 42x-01 as of service pack 02

The service packs are scheduled to be available for download from our FileBase in December 2010.

> Note
>
> There will be no service pack 01 for NC software 340 49x-06. The first service pack of NC software 340 49x-06 will be SP 02.

If you need assistance or have any questions, please contact the responsible HEIDENHAIN service agency.

1.2.2 Brake test for synchronized axes

General information:

The brake test is activated (> 0) or deactivated (= 0) separately for each drive via MP2230.x. During the brake test, an additional test torque defined via a multiplier from MP2230.x is applied for the motor stall current. This test torque puts an additional load on the holding brake of the drive. Only if the brake withstands this load is the axis prevented from moving during the brake test and the brake test is evaluated as "passed".

The control determines the algebraic sign of the test torque individually for each drive depending on the holding torque. On drives without gravitational load or with closed brake and thus without current holding torque, the sign of the test torque cannot be determined exactly.
This is a problem during the brake test of synchronized axes if different algebraic signs are determined for the master and slave drives. In rare cases this can lead to the test torques of the master and slave drives working against each other so that the synchronized axis is strained. In this case, the result of the brake test cannot be based on the real holding force of the brake, which means that the brake test result is not meaningful.
Besides this, the algebraic sign must be opposite to the current holding torque on gravity-loaded axes. Otherwise the power stage must provide not only the holding torque but also the test torque against gravity and in unfavorable circumstances, this can result in a shut-down by the power stages because of overload.

HEIDENHAIN is adapting the brake test for synchronized axes as follows:

Modified behavior during synchronized axes brake test:

The function reads the machine configuration to detect which drives are operated as synchronized axes and which must therefore be treated separately in the brake test. MP850.x is used to configure drives to a synchronized axis. The brakes and drives of the synchronized axis are tested simultaneously. It is ensured, however, that the same algebraic sign is used for the test torque of all drives.
The sign is determined for all drives of the synchronized axis based on the entry for the holding torque of the master in MP2630.x. If no value has been entered in MP2630.x, the current holding torque of the master is used.

There are two possibilities for starting the brake test. In both cases the function described is used to test the brakes of synchronized axes simultaneously:

- Automatic brake test
  The brake test takes place automatically during the power-up test of the control, as soon as all drives of the respective synchronized axis have been switched on.
- Brake test via PLC module
  The brake test can be activated axis-specifically by the PLC program with PLC module 9143. What is new in the service packs listed below is that the slave drives of a synchronized axis can be tested simultaneously with the master. Previously it was not possible to perform the brake test of slave drives via the PLC module. As a prerequisite for the brake test of a synchronized axis, all drives of the axis must be switched on and the brakes must be opened.

HEIDENHAIN will introduce the modified brake test as described above with the following service packs:

- NC software 606 42x-01 SP 02 (December 2010)
- NC software 340 49x-06 SP 02 (December 2010)
- NC software 340 49x-05 SP 07 (December 2010)
- NC software 340 49x-04 SP 09 (January 2011)
- NC software 340 49x-03 SP 11 (January 2011)

New behavior as of 340 490-06 SP02 and 606 42x-01 SP02:
Testing the brakes of a synchronized axis successively

The above-mentioned service packs 02 will also make it possible to activate a changed brake test sequence via MP860 bit #2. With MP860 bit #2 = 1 the drives of a synchronized axis are tested successively rather than simultaneously. Due to this, the brake test is performed individually for all drives of a synchronized axis.

Each brake or drive of the synchronized axis is then sequentially tested with the specified test current. For the brakes and drives that are not part of the momentary test, but are configured as connected to the drive to be tested, the current is set during the test so that the drive is not moved. The brakes of these drives must be opened for this. This way, only the brakes of an individual drive are tested each time, without the other drives or brakes of the synchronized axis having an effect on the test.

To be able to use this brake test sequence, the individual brakes of the synchronized axis must not be combined. Individual control of the brakes must be possible.

There are two possibilities for starting the brake test. In both cases the function described above is used to test the brakes of synchronized axes sequentially:

- Automatic brake test
  The brake test takes place automatically during the power-up test of the control, as soon as all drives of the respective synchronized axis have been switched on.
- Brake test via PLC module
  The brake test can be activated axis-specifically by the PLC program with PLC module 9143. What is new in the below-mentioned service packs is that the slave drives of a synchronized axis can be tested successively after the master. Previously it was not possible to perform the brake test of slave drives via the PLC module. As a prerequisite for the brake test of a synchronized axis, all drives of the axis must be switched on and the brakes must be opened.

Due to the modified sequence of the brake test, more time is necessary to perform the test for all axes. You may have to consider this for your PLC program if you monitor the sequence times of the power-up test or the brake test.

HEIDENHAIN will introduce the brake test for synchronized axes as described above with the following service packs:

- NC software 606 42x-01 SP 02 (December 2010)
- NC software 340 49x-06 SP 02 (December 2010)

Conditions for the brake test:

As a prerequisite for the brake test of a synchronized axis, all drives of the axis must be switched on and the brakes must be opened. The test can only be performed if all relevant drives are switched on.

> Note
>
> Before performing the brake test, ensure via the PLC program that all drives of a synchronized axis are switched on and the holding brakes are opened.

For slave drives for which the brake test has been disabled via MP2230.x, the current is adjusted so that the drive is not moved during the test of the other drives of the synchronized axis.

As the algebraic sign of the test torque cannot be determined before the drives are servo-controlled and the brakes are opened, an appropriate waiting time must be specified for the start of the brake test of the synchronized axes. The time set in MP2309.x is used for this. The value for MP2309.x must equal the time that passes until the brake is really open after the controller has been switched on. The time entered in MP2309.x must be equal for all drives of a synchronized axis. For CC 424 and CC 61xx controller units the time in MP2309.x is taken into account each time the controller is switched on. For CC 422 controller units the time in MP2309.x is only taken into account the first time the controller is switched on, in order to delay the brake test by this time. The delay is defined as 200 ms for software versions that do not support MP2309.x.

The following applies in general to the brake control: If the brakes are not controlled by inverters but by the PLC, PLC module 9159 (drive controllers are switched off) gives the command for the PLC program to close the brakes during the test.

HEIDENHAIN recommends:

> Note
>
> For all machines on which a brake test for synchronized axes is to be performed, HEIDENHAIN recommends installing the above-mentioned service packs to be able to use the new behavior of the brake test. Moreover, if software versions 340 49x-06 and 606 42x-01 are used, the sequential brake test of synchronized axes should be activated (MP860 bit #2 = 1).
>
> - If required, modify the PLC program with respect to the conditions and the changed behavior of the brake test.
> - Test the behavior of the PLC program and the brake test on the machine.
> - If required, run an update of the NC software and the PLC program on the affected machines in the field.

If you also need the modified brake test for synchronized axes for software versions for which no service pack is currently planned, please contact the responsible HEIDENHAIN service agency.

Machine parameters and PLC module:

MP_860.x	**Synchronous control**
Input:	Bit 2: Brake test for synchronized axes 0 = Testing the brakes of a synchronized axis simultaneously 1 = Testing the brakes of a synchronized axis successively (new behavior, as of software versions 340 49x-06 SP02 and 606 42x-01 SP02)
MP_2230.x	**Multiplier for motor current during test of motor brake**
Input:	0.100 to 30.000 [· motor stall current] 0: No test of motor brakes, or motor without brake Recommended: 1.3 · M_L / M_0
MP_2232.x	**Maximum permissible path during test of motor brakes**
Input:	0 to 10.0000 [mm] or [°]
MP_2309.x	**Controller parameters adjusted to closed brake**
Input:	0: Not active 0.001 to 5.000 [s]
MP_2630.x	**Holding current**
Input:	-100.000 to +100.000 [A]

Module 9143 Activate the brake test

This module can start an axis-specific brake test with the configuration from the machine parameters or with values other than those in MP2230 and MP2232.

Constraints:

- Synchronized axes
  For synchronized axes, only the brake test of the master can be configured and requested via the PLC module. If a brake test for an associated slave drive of the synchronized axis is configured via MP2230.x, then the slaves are tested automatically with the master. The settings in the machine parameters are used for the brake test of the slave drives.
  In order to start the brake test of synchronized axes via PLC module 9143, all drives of a synchronized axis must be switched on via the PLC program before the brake test can be performed. If a participating drive is not switched on, the brake test is stopped with the error message **`8330 Brake test was aborted`**.
- Programming the module in a submit job blocks other submit jobs until the test is completed.
- The PLC module automatically passes the processing time to other spawn and submit processes.

Call:

PS	K/B/W/D	<Axis number> 0 = 1st axis, 1 = 2nd axis, etc.
PS	K/B/W/D	<Multiplier for motor stall current> Value in 1/1000 or 0: Multiplier from MP2230 (default)
PS	K/B/W/D	<Permissible traverse path> Value in 0.1 [um] or 0: Default MP2232
CM	9143	
PL	B/W/D	<Status/Error> 0: Brake OK 1: Brake defective 2: Invalid axis or negative values for rated current or traverse path 3: Call during running NC program or during other PLC jobs 4: Call was made from a cyclic PLC program 5: Error during data exchange 6: Not permitted for controls with functional safety FS 7: Drive not ready 8: Brake test was aborted (e.g. by emergency stop)

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis programmed (invalid axis number, not a closed-loop axis, axis currently open-loop axis or slave axis) or negative values for the traverse path or current are programmed
	8	Module not permitted for controls with functional safety FS
	20	Module was not called in a spawn job or submit job
	21	Call during program run or during other active PLC jobs for the programmed axis
	40	Drive not ready
	45	Canceled due to error during data exchange or due to external influences (e.g. emergency stop)

Module 9159 Advance status report: Drives will be switched off

Call:

CM	9159	
PL	W/D	<Drives, in bit code, that are switched off in the time defined in MP2308>

1.2.3 New Functions

HEIDENHAIN will introduce the described, new functions only with the following service packs:

- NC software 606 42x-01 SP 02 (December 2010)
- NC software 340 49x-06 SP 02 (December 2010)
- NC software 340 49x-05 SP 07 (December 2010)

Profibus telegram repetitions

With the above-mentioned software versions, the HEIDENHAIN Profibus master can execute telegram repetitions if there are errors in the Profibus communication. This function is activated via the entry **`__JHDP_RETRY=x`** in MP4000.x. To do so, the x must be replaced by the required maximum number of telegram repetitions, e.g. **`__JHDP_RETRY=5`**. The permissible range for x is 0 to 15. If a number above 15 is entered, it is automatically limited to 15. Previously, the occurrence of telegram repetitions immediately triggered an Emergency Stop. This is still the default setting.

When activating Profibus telegram repetitions it must be noted, however, that the occurrence of telegram repetitions influences the time response of the entire Profibus data transfer. This can have massive effects on the time response of the PLC program, as the updating times of the inputs and outputs can vary considerably.

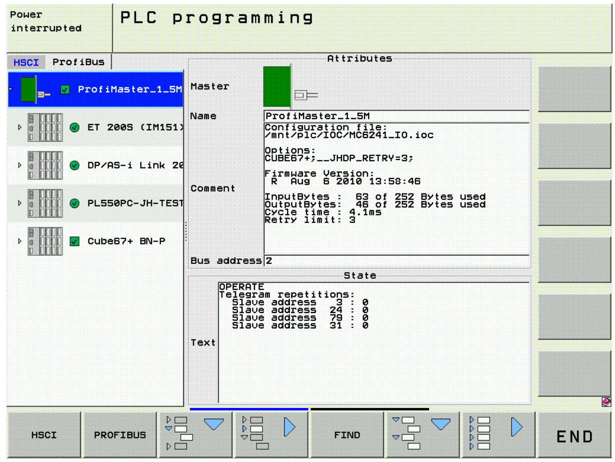
Attention

Do not activate telegram repetitions in PLC programs that depend on fixed update intervals for the inputs and outputs.
Repetitions of Profibus telegrams can cause delays of one or more PLC cycles while reading PLC inputs.

The PLC program does not recognize the occurrence of telegram repetitions below the set maximum number.

If the number of required telegram repetitions exceeds the configured maximum number, an error message with an Emergency Stop reaction is triggered.

To detect potential problems in the bus system, the telegrams that required repetitions are counted in the Profibus master for each Profibus slave. This information, as well as the maximum permissible number of repetitions, is provided in the Profibus diagnosis:



1 Update Information No. 29

1.1 Overview

1.1.1 Released service packs

The following service packs were released for **340 49x-06**:

- Service pack 02: December 2010

The following service packs were released for **606 42x-01**:

- Service pack 02: December 2010

1.2 New Hardware – HR 550FS Wireless Handwheel

Another product of the new HR 5xx handwheel family, the HR 550FS wireless handwheel for the iTNC 530 (HSCI), is now available in addition to the HR 520. The standard iTNC 530 (HSCI) supports the use of electronic handwheels. The handwheels are connected to X23 of the MC, or to X23 of the machine operating panel in the case of HSCI.

A wireless handwheel system consists of the HR 550FS mobile part and the HRA 551FS handwheel adapter. The HR 550FS has a rechargeable battery for mobile operation on the machine. During operation it continuously communicates via radio with the HRA 551FS handwheel adapter, which communicates directly with the control via a serial cable connection.

The HRA 551FS handwheel adapter, which functions both as receiving and charging station for the mobile part, features an M23 connector system for connection to the control via a standard handwheel adapter. If the wireless handwheel system is damaged, the dummy plug for the emergency stop circuit (ID 271 958-03) can be used so that the operational readiness of the machine can be restored quickly.

To ensure the necessary and unambiguous assignment of the HR 550FS to the HRA 551FS, a pairing is created during commissioning. Once they have been paired, a wireless handwheel and a handwheel adapter can only be operated with the assigned device. As a precautionary measure, paired devices must be marked by the stickers (in five different colors) included in the items supplied so that they can be identified visually (the soft-key label strip for the mobile part must have the same color as the round sticker for the handwheel adapter). After successful pairing, they must be affixed to the wireless handwheel system. The available colors are gray, blue, red, green and yellow.

The wireless handwheel system can be used in many different places thanks to the 2.4 GHz ISM radio band, which is freely accessible almost all over the world. The HR 550FS complies with the internationally accepted standards. For radio communication, the HEIDENHAIN HR 550FS uses 2.4 GHz ZigBee transmission technology with a special HEIDENHAIN communications protocol.

<table>
<tr><th colspan="2">HR 550FS handwheel with wireless transmission</th></tr>
<tr><td>Portable electronic handwheel with wireless transmission:
■ Graphic display, resolution: 128 x 64 pixels, 6-line display
■ 6 NC keys
■ 6 PLC keys with LEDs that are controlled through the PLC
■ 2 override potentiometers (feed rate and spindle speed)
■ 2 permissive buttons
■ Exchangeable snap-on keys for PLC functions and maintenance
■ Integrated emergency stop button
■ Vibration alarm when leaving the radio range
■ Battery warning
■ Display of radio field strength
■ Weight: Approx. 1 kg
Without mechanical detent:
ID 598 515-xx
With mechanical detent:
ID 606 622-xx
Accessories (1 x included with the HR 550FS):
Handwheel batteries
(battery pack, 4-tray)
ID 623 166-xx</td><td></td></tr>
<tr><td>HRA 551FS
Handwheel adapter (docking tray) with integrated charger for handwheel batteries
ID 731 928-xx</td><td></td></tr>
</table>

1.2.1 HR 550 wireless handwheel

The HR 550FS wireless handwheel must always be used together with the HRA 551FS handwheel adapter. The handwheel and the handwheel adapter communicate with each other via radio transmission. The handwheel adapter is connected to X23 of the control system via the connecting cable.

When the handwheel is located in the HRA, contacts on the HRA and on the rear of the HR are used to communicate via serial data transmission instead of via radio transmission. The function of the HR 550FS remains the same, regardless of whether it communicates via radio transmission or via the handwheel adapter.

The HRA handwheel adapter features an integrated charger for the handwheel batteries. In addition, the safety-related signals (emergency stop, permissive buttons) are translated into relay contacts through the HRA. They must be used to connect the emergency stop button and the permissive buttons of the handwheel to the control and the associated safety circuits.

Danger

If more than one control with a portable HR 550FS handwheel is located in a factory hall, a room or a working area, the handwheels must be identified by unique color markings. The color marking must be applied to the handwheel **and** the associated machine tool at a clearly visible location. The marking on the handwheel and the machine tool must enable the machine operator to identify unambiguously which handwheel belongs to which machine tool. The color marking must ensure that no confusion occurs. A set of different markers is included with the wireless handwheel.

In addition, the machine operator must be informed that he must ensure/verify every time before using the handwheel that he uses the correct handwheel for the respective machine tool.

If the machine operator moves with the HR too far away from the HRA, a vibration alarm announces that transmission is weak before the machine operator has actually left the radio range. If the machine operator leaves the radio range anyway, or if radio communication is interrupted for other reasons, the HRA relay contacts for emergency stop and for the permissive buttons will open. You must ensure through appropriate wiring that this triggers an emergency stop on the control.

HEIDENHAIN also recommends that you always place the wireless handwheel into the HRA 551FS adapter when you are not using it. This prevents you from confusing the different handwheels. It also ensures that the handwheel battery is recharged and prevents an unexpected emergency stop reaction triggered by an empty battery.

Specifications	HR 550FS with HRA 551FS
Radio range (max. distance between the HR and the HRA)	Max. 20 m in direct line of sight (depending on the ambient conditions)
Ambient temperature during operation	0 °C to + 40 °C
Ambient temperature for charging process of battery	+ 5 °C to + 40 °C (recharging the battery at a temperature of < + 10 °C may reduce the battery life)
Storage temperature of NiMH rechargeable battery	–20 °C to +21 °C with a state of charge of 40 %
Power supply for HRA 551FS	12 V
Display resolution	128 x 64 pixels
Power consumption	Max. 10 W during charging phase
Batteries	Pack with 4 x AA batteries NiMh 4.8 V / 2400 mAh
Complete charging time	Approx. 150 minutes
Life of rechargeable battery	Approx. 16 hours with a utilization of 50 %
Radio frequency	2.4 GHz ISM band
Transmitter power	10 mW
Usable number of channels	16 (ZigBee)
Number of possible HRs with wireless transmission	Only one wireless HR per machine tool
Degree of protection	IP 54 for HR 551FS IP 54 for HR 550FS
Service life of the product	20 years
Max. voltage for relay contacts in HRA 551FS	250 V ~
Max. current for relay contacts in HRA 551FS	3.6 A
Reaction times of emergency-stop and permissive-button safety functions	Time until the reaction of the relay contacts: max. 170 ms; typically 60 ms
FCC-ID	YJKHR550FS
Weight of HR 550	1 kg
ID for HR 550FS without mech. detent	598 515-xx
ID for HR 550FS with mech. detent	606 622-xx
ID for HRA 551FS	731 928-xx

When using a wireless handwheel and the HRA handwheel adapter at X23, you must ensure that the assignment between the handwheel and the HRA is unambiguous. The wireless handwheel serial number and the HRA serial number are used for mutual assignment. When the control to which the HR and the HRA are connected is switched on for the first time, both units exchange their serial numbers and store them. This process is called "pairing." To do this, the wireless handwheel must be located in the HRA. If this is the case, the HR and the HRA communicate only over the serial interface.
Both serial numbers (HRA and HR) will then also be saved in the control. If you remove the HR from the HRA during the initialization process, the process will be canceled and a corresponding error message will be issued.

During the emergency stop test and when the connection to the wireless handwheel is set up, the system ensures that only the HR and the HRA are addressed, whose serial numbers have been assigned unambiguously to each other. If a difference is found in one of the comparisons of serial numbers, the connection will be terminated. The relays for emergency stop and the permissive buttons remain open during the complete process.

Switching the HR 550FS on/off

To switch the HR 550FS off or on, press the CTRL key and the medium soft key simultaneously. When switching the HR 550FS on, you must also press one of the two permissive buttons. The HR 550FS does not run a power-up test and can therefore not be used until a permissive button has been pressed. During the power-up test, the background illumination of the HR 550FS is active, but the display remains blank. The permissive buttons are tested at the end of the power-up test. None of the permissive buttons may be pressed at this time. In the event of error, an error message appears on the display, informing you that a permissive button is pressed. As a result, the power-up test cannot be completed. The display is filled with characters as soon as the power-up test has been completed successfully, and the "OFF-LINE" message appears on the display. If an error is found during the power-up test, an error message appears on the display and the service department must be informed.

The handwheel is configured in a menu that you open by pressing the SET UP WIRELESS HANDWHEEL soft key. After you have pressed the MOD key, this soft key is displayed in the second soft-key row in the Programming and Editing mode of operation if machine parameter MP7640 has been set to the value of 12. The following functions are available for configuration:

- Assigning the handwheel to a specific handwheel holder
- Setting the radio channel
- Analyzing the frequency spectrum for determining the optimum radio channel
- Selecting the transmitter power
- Statistical information on the transmission quality

For configuring the HR 550FS handwheel, refer to the chapter "Configuring the HR 550FS Wireless Handwheel" in the User's Manual for the control.

The functions that can be defined for the soft keys on the HR 550FS are the same as for the HR 520(FS).

After the wireless handwheel has been configured and switched on, it can be activated via the handwheel activation key. Machine operation then switches from the machine operating panel to the handwheel. The machine tool builder must ensure through the PLC program that the machine can be operated by only one operating unit at any one time. Machine operating panel and handwheel must never be active at the same time. The display of the wireless handwheel and the display on the screen inform the operator when the wireless handwheel is active.

1.2.2 Radio transmission regulations

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

- This device must not cause harmful interference, and
- this device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy. If the equipment is not installed and used in accordance with the instructions, the equipment may cause harmful interference to radio communication. There is no guarantee, however, that such interference will not occur in a particular installation.

Note

This wireless handwheel from HEIDENHAIN must be installed and used in strict accordance with the manufacturer's instructions as described in this technical documentation and the User's Manual for your control. Any other installation or use will violate FCC Part 15 regulations. Modifications not expressly approved by Dr. JOHANNES HEIDENHAIN GmbH could void your authority to operate the equipment.

This device must not be co-located or operated in conjunction with any other antenna or transmitter.

1.2.3 General information

Thanks to the wireless connection between the handwheel and the control, the HR 550FS wireless handwheel system and the HRA 551FS handwheel adapter provide maximum ease of use to the machine operator. The wireless communication uses the public, and therefore freely available, 2.4 GHz radio band via a special radio transmission protocol from HEIDENHAIN.

The wireless handwheel system attains a high and certified safety standard because of its dual-channel transmission protocol. The safety functions of the HR 550FS fulfill the requirements of performance level d, category 3, according to EN ISO 13849-1. Single errors do not lead to the loss of the safety functions. The service department must be notified if an error occurs. Only the functions of the permissive buttons and the emergency stop switch are classified as safety functions of the HR 550FS.

The safety functions must not be bypassed or deactivated in some other way.

Modifying or rebuilding the HRA 551FS and HR 550FS is not permitted and may lead to a loss of the safety functions.

The machine operator is responsible for identifying the area in which the wireless handwheel may be used. The wireless handwheel itself does not provide any configuration possibilities or limit values for defining a certain maximum range of use.

Because a handwheel is always connected with the movement of axes, the operator must be able to trigger an emergency stop of the machine by means of the handwheel at any time. This results in a special situation for the emergency stop switch of the HR 550FS wireless handwheel system. An unsafe situation in the radio communication must always trigger a safe reaction from the control. Therefore, an interference in the radio communication also means an emergency stop of the HRA 551FS as reaction.

An unexpected emergency stop of the machine is undesirable and may cause damage to the contour of the workpiece being machined. It is therefore essential to ensure a noise-free radio communication. Reliable operation of the wireless handwheel always requires planning the coexistence of the radio users active in the frequency and radio range concerned. If this cannot be guaranteed because there is no vacant radio channel available, we strongly advise against using the wireless handwheel system. HEIDENHAIN cannot guarantee noise-free radio communication of the HR 550FS and HRA 551FS wireless handwheel system.

Even if an available radio channel has been found and the proper functioning of the system has been ascertained, it is possible that the availability situation of the 2.4 GHz ISM band will change over a longer period of time. If this situation cannot be corrected by changing the wireless handwheel system or the competing radio system to another channel, it may become necessary to revert to using a handwheel with cable.

The following situations lead to an emergency stop through the wireless handwheel system:

- The HR 550FS wireless handwheel is placed into the wrong HRA 551FS handwheel adapter.
- The HR 550FS wireless handwheel is switched off outside of the HRA 551FS handwheel adapter.
- The radio communication between the HR 550FS wireless handwheel and the HRA 551FS handwheel adapter is interrupted.
- The charge of the handwheel battery becomes too low.

Note

- The power-up test of the HR 550FS must be repeated within no more than 168 hours. The HR 550FS handwheel and the HRA 551FS handwheel adapter must be turned off and on again to conduct the test. It must be ensured by the PLC program that this requirement is met.
- To guarantee the HR's emergency-stop and permissive-button safety functions, the respective contacts must be connected to the emergency stop chain and further safety circuits (if present) or safe inputs of the control.

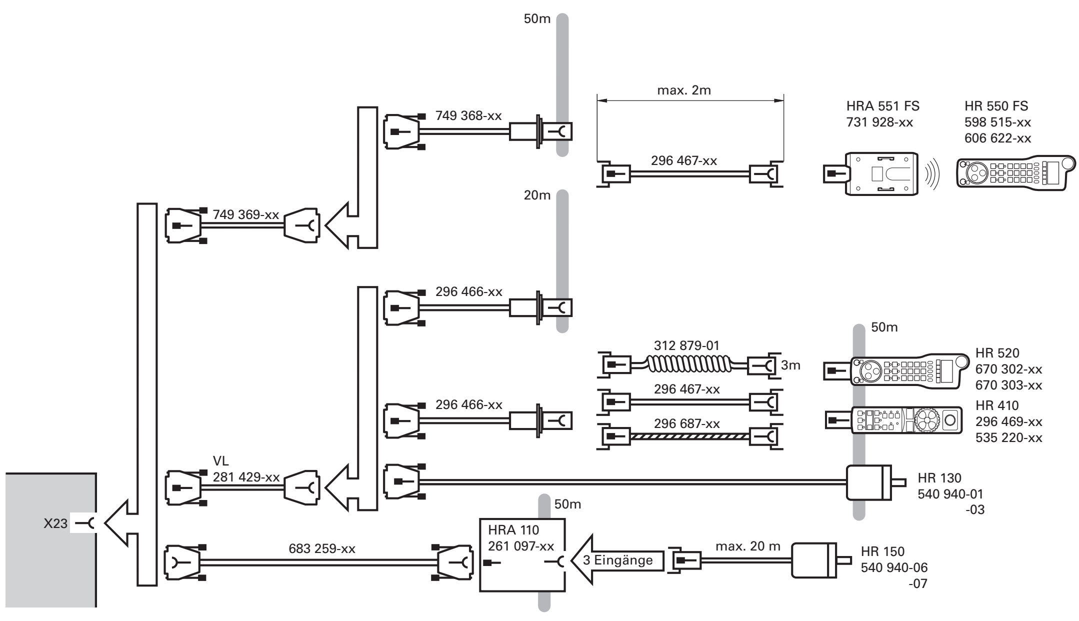
The handwheel requires no maintenance work as long as the HR 550FS can be operated in conjunction with the HRA 551FS without disturbances or error messages. In the event of a fault, please inform your service agency. The only expendable part of the HR 550FS are the rechargeable batteries. If you notice a marked decrease in battery life, it's time to exchange them (handwheel batteries, ID 623 166-02).

1.2.4 Connecting the wireless handwheel system

A wireless handwheel system is always connected to the control via the M23 plug connection of the HRA 551FS. The advantage of this is that the wireless handwheel system is also supplied with power via this connection so that no additional supply lines must be provided.

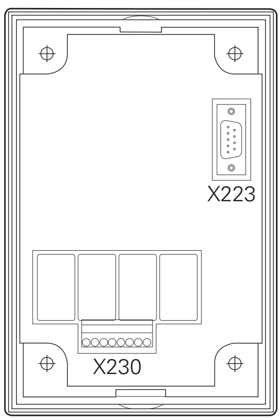
The serial communication, dual-channel emergency stop and dual-channel permissive buttons of the HR 550FS mobile part are routed from the HRA 551FS (M23 plug connection with handwheel cable, ID 296 467-xx) via the corresponding handwheel adapter, ID 749 368-xx, to the control hardware. For this purpose, special, short handwheel cables (ID 296 467-xx) in lengths of 1.0 m, 1.5 m and 2.0 m were introduced. The reason for this is the charging current required for the HR 550FS mobile part. Due to the small cross section of the handwheel cable it can be transmitted only over short distances. This means that longer cables with ID 296 467-xx cannot be used for the wireless handwheel system. If longer handwheel cables are required, HEIDENHAIN should be consulted. A maximum distance of 50 m between X23 (MC 4xx or MB 620/PLB 6001 for HSCI) and the HRA 551FS base station can be attained by using the adapter cable ID 749 368-xx and the optional extension cable ID 749 369-xx (with a handwheel cable, ID 296 467-xx, with a length of max. 2.0 m).

Please note that the new adapter cables, ID 749 368-xx, and extension cables, ID 749 369-xx, will not be available until April 2011. Until then, the wireless handwheel system must be connected with the currently available cables, which limits the attainable cable length to the base station to 22 m!



Use of available cable connections:
For cable lengths up to 22 m, the existing cable connections for HEIDENHAIN handwheels of the HR 4xx and HR 5xx series can be used to retrofit a wireless handwheel system. The existing handwheel connecting cables must be checked regarding their length and the possibility of connecting a wireless handwheel system. Please refer to the cable overview shown above for the wireless handwheel system.

HR 551FS – Connector overview

<table>
<tr><td colspan="4">HRA 551FS</td></tr>
<tr><th>Internal pin layout</th><th>Connector</th><th>Function</th><th>Page</th></tr>
<tr><td rowspan="3">
</td><td>X223</td><td>Serial handwheel interface</td><td>1 – 229</td></tr>
<tr><td>X230</td><td>Emergency stop, permissive buttons of handwheel</td><td>1 – 230</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

X223: Serial interface on the HRA

Connecting cable for serial interface from X23 of the MB to X223 in the HRA, ID 683 259-xx.

Pin layout:

HRA 551FS	
9-pin D-sub connector	**Pin layout**
1	Do not assign
2	0 V
3	Do not assign
4	+12 V
5	Do not assign
6	DSR–
7	RxD
8	TxD
9	Do not assign
Housing	External shield

X230: Emergency stop, permissive buttons on the HRA

Pin layout:

HRA 551FS	
8-pin Phoenix terminal	**Pin layout**
1	Permissive button contact A / terminal 1
2	Permissive button contact A / terminal 2
3	Permissive button contact B / terminal 1
4	Permissive button contact B / terminal 2
5	Emergency stop contact A / terminal 1
6	Emergency stop contact A / terminal 2
7	Emergency stop contact B / terminal 1
8	Emergency stop contact B / terminal 2

The relay contacts for the permissive buttons and emergency stop button on the HRA are four individual, normally open contacts.

- Behavior of the emergency stop contacts
  When the emergency stop button is not pressed, the relays are actuated, and therefore the relay contacts are closed. If an emergency stop is triggered, the control voltage of the relays will be switched off and the relay contacts will open. For the control and the associated safety circuits, the relay contacts have the same effect as normally closed contacts.
  If the batteries of the wireless handwheel are empty or if the wireless handwheel is out of range, the relay contacts are open.
- Behavior of the contacts for the permissive buttons
  Depending on the setting in MP7645.3, the permissive buttons are represented as two normally open contacts, or as one normally open contact and one normally closed contact. This makes it possible to adapt the handwheel (permissive buttons) to the requirements of the planned safety concept of the respective machine.
  HEIDENHAIN recommends setting the MP7645.3 to 1 to make the handwheel permissive buttons short-circuit proof. To ensure compatibility with previous machine designs, MP7645.3 is set to 0 by default.
  - MP7645.3 = 0:
    When neither of the two permissive buttons is pressed, both relays are without control voltage, and therefore the relay contacts are open. When at least one permissive button is pressed, both relays are actuated and both relay contacts close.
  - MP7645.3 = 1:
    When neither of the two permissive buttons is pressed, one of the relays is without control voltage, and relay contact A is open. When at least one of the permissive button is pressed, the relay is actuated and relay contact A closes. When neither of the two permissive buttons is pressed, control voltage is applied to the other relay and relay contact B is closed. When at least one of the permissive button is pressed, the relay is not actuated anymore and relay contact B opens. MP7645.3 must be set to 1 for machines with HEIDENHAIN Functional Safety.

Note

The interfaces comply with the requirements of EN 60204-1:2006 for "protective extra-low voltage (PELV)."

Pin layout for the various extension cables, adapter cables, connecting cables, and the handwheel:

Extension cable ID 281 429-xx			**Adapter cable ID 296 466-xx**			**Connecting cable ID: see "Introduction" chapter**			**HRA 551FS**	
D-sub connector (male) 9-pin		**D-sub cnnctr. (female) 9-pin**	**D-sub connector (male) 9-pin**		**Cplg.mntg.base (female) (5+7)-pin**	**Cnnctr. (male) (5+7)-pin**		**Cnnctr. (female) (5+7)-pin**		
Hsg.	Shield	Housing	Hsg.	Shield	Housing	Hsg.	Shield	Housing	Hsg.	Internal X230
2	White	2	2	White	E	E	White	E	E	
4	Brown	4	4	Brown	D	D	Brown	D	D	
6	Yellow	6	6	Yellow	B	B	Yellow	B	B	
7	Gray	7	7	Gray	A	A	Gray	A	A	
8	Green	8	8	Green	C	C	Green	C	C	
					6	6	Black	6	6	8
					7	7	RD/BL	7	7	7
					5	5	Red	5	5	5
					4	4	Blue	4	4	6
					2	2	WH/GN	2	2	3
					3	3	BN/GN	3	3	1
					1	1	GY/PK	1	1	4, 2
				WH/BN	3	Contacts A + B				
				WH/YL	2	Contact B (permissive button)				
				WH/GN	1	Contact A (permissive button)				
				WH/BL	1	Contact A/Terminal 1 (EMERGENCY STOP)				
				WH/RD	2	Contact A/Terminal 2 (EMERGENCY STOP)				
				YL/BK	3	Contact B/Terminal 1 (EMERGENCY STOP)				
				WH/BK	4	Contact B/Terminal 2 (EMERGENCY STOP)				

Note

The interfaces comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

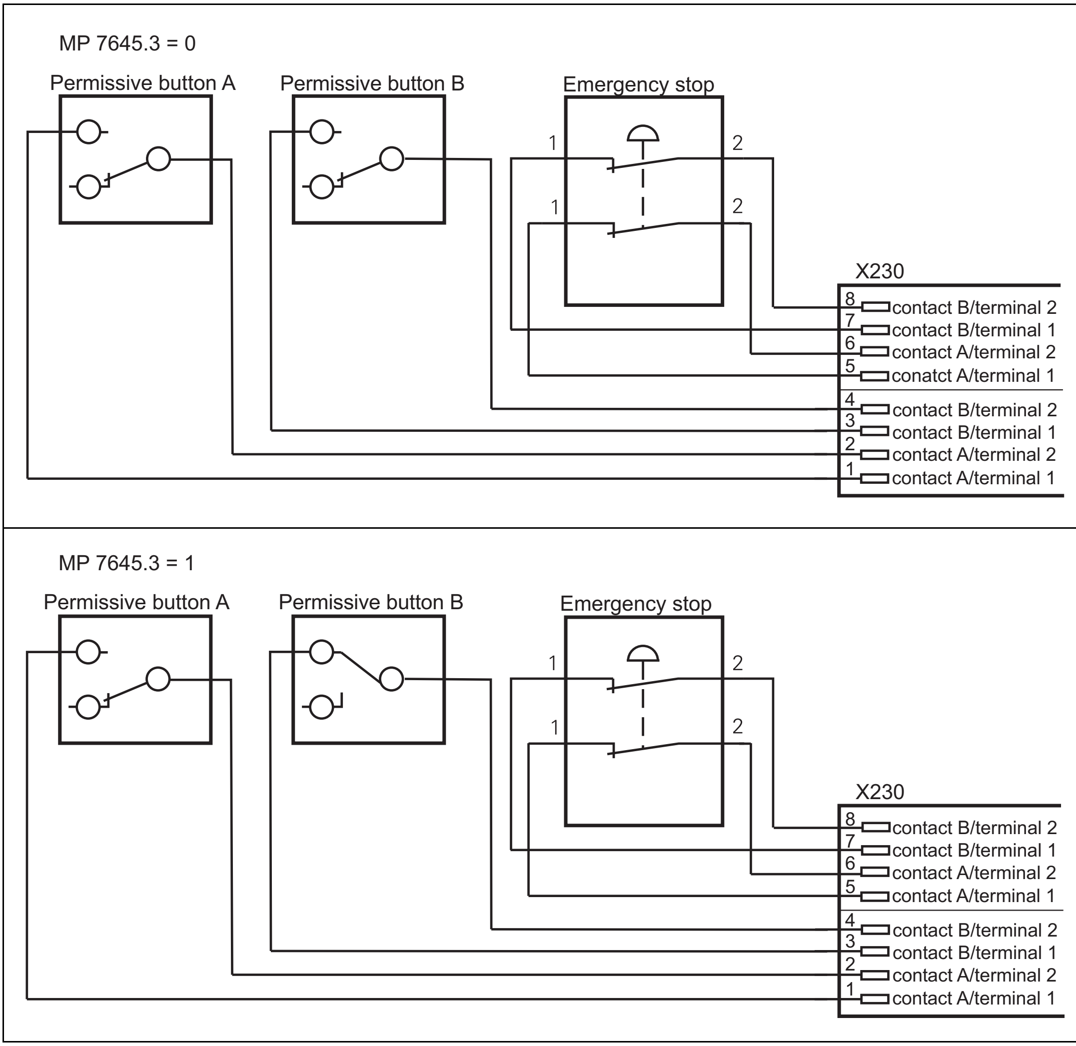
Danger

Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

The adapter includes plug-in terminal strips for the contacts of the emergency stop button and permissive button (max. load 24 V–, 1.2 A).

The plug-in terminal strips are supplied together with the adapter cable. If you have an immediate need for these terminal strips, they can be ordered in advance. See the "Additional components" table below.

Internal wiring of the contacts for the emergency stop button and permissive button:



Additional components		ID
Connecting cables		
	Spiral cable	312 879-01
	Normal cable	296 467-xx
	With metal armor	296 687-xx
Plug-in terminal strips for advance ordering		
	4-pin terminal block	266 364-12

1.2.5 Settings

- Enter MP7640 = 12 (HR 550FS).
- In MP7641, specify whether you are using an HR 5x0 with or without detent, and whether the keys on the handwheel are to be evaluated by the NC or PLC.

All settings that can be defined in the machine parameters of the HR 520 can also be used for the HR 550FS. The PLC markers of the HR 520 also match those of the HR 550FS.

All keys of the HR 5x0 are evaluated by the NC. Certain keys are mapped to PLC markers. The six LEDs of the HR 5x0 can be controlled by the specified PLC markers.

F1	F2	F3	F4	F5
	X	Y	Z	
	IV	V	VI	
	↑	Handwheel active/ inactive	↓	
	– (M4667)	Rapid traverse (M4663)	+ (M4666)	
	Spindle start (M4664) LED on (M4684)	Actual position capture LED on (M4689)	NC start (M4661) LED on (M4681)	
	Spindle stop (M4665) LED on (M4685)	Ctrl (M4668) LED on (M4688)	NC stop (M4662) LED on (M4682)	

1.2.6 Prerequisites for use of the wireless handwheel system

The following conditions must be taken into account for the operation of a wireless handwheel system:

- The HR 550FS wireless handwheel system with the HRA 551FS is currently only available in combination with the iTNC 530 control.
- The iTNC 530 requires NC software version 340 49x-06 or 606 42x-01 (HSCI), or higher.
- The use of radio equipment in the 2.4 GHz range must be generally permitted at the intended place of use (end user's site).
- Country-specific and customer-specific requirements and regulations for radio equipment at the intended place of use of the wireless handwheel system must be taken into account.
- At the intended place of use, every wireless handwheel system from HEIDENHAIN needs a suitable and vacant radio channel in the 2.4 GHz ISM band. If this cannot be guaranteed, we advise against using the wireless handwheel system (see "Important notes on the radio network of the wireless handwheel system" on page 235 and see "Checking the radio environment at the intended place of use" on page 236).
- If several wireless handwheel systems from HEIDENHAIN are to be used within a certain range, a vacant channel must be selected for each individual wireless handwheel system.
- If there are not enough vacant radio channels available for the wireless handwheel systems and identical channel numbers have to be used for the wireless handwheel systems because of other ISM band participants, a minimum distance of 200 m must be maintained between the HRA 551FS base stations. In addition, the transmitting and receiving power of the wireless handwheel system can be reduced during commissioning. Three power stages are available for this. They are set using the wireless handwheel configuration dialog on the control.

Ideally, "coexistence planning" for all ISM band participants is available for the intended place of use of the wireless handwheel system. The coexistence planning includes all radio participants and the radio channel or radio range they are using. This planning can be used to determine whether a vacant radio channel is available for the wireless handwheel, and therefore whether a wireless handwheel system can be used.

1.2.7 Important notes on the radio network of the wireless handwheel system

The HEIDENHAIN wireless handwheel system uses the ISM band in the frequency range of 2.405 GHz to 2.480 GHz up to a maximum transmission power of 10 mW. As the ISM band generally is a freely accessible frequency range worldwide (country-specific restrictions must be checked individually for the country of destination), many radio systems have been developed for this range. As other radio systems can disturb the radio transmission of the wireless handwheel system, a vacant radio channel (ZigBee channels 11 to 26) is always required to ensure noise-free radio communication between the HR 550FS and HRA 551FS.

Since a handwheel is always connected with the movement of axes, the operator must be able to trigger an emergency stop of the machine by means of the handwheel at any time. This results in a special situation for the emergency stop switch of the HR 550FS wireless handwheel system. An unsafe situation in the radio communication must always trigger a safe reaction from the control. Therefore, an interference in the radio communication also means an emergency stop at the HRA 551FS as reaction. An unexpected emergency stop of the machine is undesirable and may cause damage to the contour of the workpiece being machined. It is therefore essential to ensure a noise-free radio communication. Reliable operation of the wireless handwheel always requires planning the coexistence of the radio users active in the frequency and radio range concerned. If this cannot be guaranteed because there is no free radio channel available, we advise against using the wireless handwheel system.

HEIDENHAIN cannot guarantee noise-free radio communication of the HR 550FS wireless handwheel system with the HRA 551FS. Even if an available radio channel has been found and the proper functioning of the system has been ascertained, it is possible that the availability situation of the 2.4 GHz ISM band will change later on. If this situation cannot be corrected by changing the wireless handwheel system or the competing radio system to another channel, it may become necessary to revert to using a handwheel with cable. HEIDENHAIN therefore offers the HR 520 handwheel with cable as an alternative.

If a wireless handwheel is to be operated in an unsafe radio environment anyway, the HR 550FS mobile part must be placed into the charging position of the HRA 551FS if a critical situation occurs, in which a workpiece could be damaged through an emergency stop. The contacts on the HR 550FS and the HRA 551FS provide a direct contact between the mobile part and the base station, and therefore communication is independent of the radio connection.

1.2.8 Checking the radio environment at the intended place of use

The following procedure is very important to determine whether a wireless handwheel system can be used in a certain radio environment:

Procedure

- Clarification of whether the HEIDENHAIN wireless handwheel system is allowed to use the 2.4 GHz ISM band in the customer's area (country-specific and company-internal regulations at the intended place of use and the end user's site must be taken into account).
- Inquiry about known radio participants at the end user's site. Ideally, the end user can provide information regarding the utilization of the 2.4 GHz ISM band at the intended place of use of the wireless handwheel.
- If the customer cannot provide exact information about which devices are using the 2.4 GHz ISM band at the intended place of use, we recommend performing measurements with a suitable spectrum analyzer and a WLAN detector. HEIDENHAIN recommends performing the measurements using a spectrum analyzer to obtain a utilization profile of the 2.4 GHz ISM band, see page 237. It is also possible to use a WLAN-capable laptop and the "Network Stumbler" PC software which is available free of charge on the Internet, see page 237.
- Selection and determination of a vacant radio channel for the wireless handwheel system.
- If no vacant radio channel can be found despite the above-mentioned measures, we advise against using the wireless handwheel system.

WLAN networks

In production environments, the ISM band is most frequently used by WLAN networks. Due to their transmitting/receiving characteristics and the transmitting power to be expected, these networks are not suitable for being operated in the same frequency range as a wireless handwheel system. A wireless handwheel system must always be operated outside the frequency range of a WLAN network.

Evaluation of the radio environment

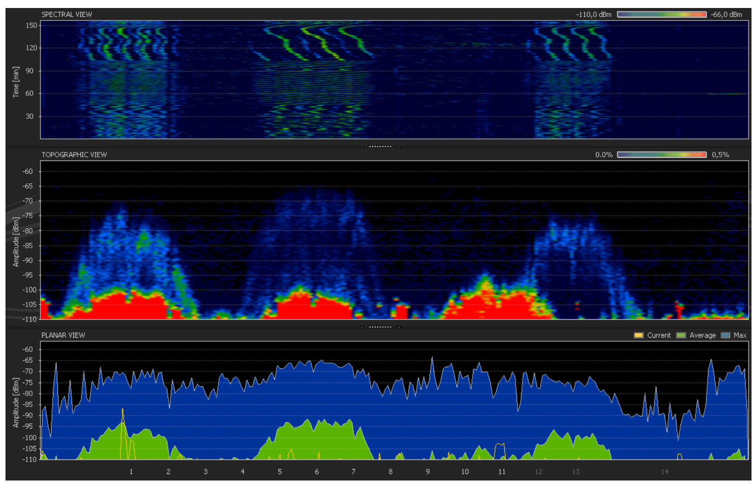
The radio environment must be evaluated at the intended place of use under the specific local conditions. All radio devices in the environment should be active for the evaluation.

HEIDENHAIN recommends performing the measurements using a "Wi-Spy WLAN USB 2.4x Spectrum Analyzer" to obtain a utilization profile of the 2.4 GHz ISM band. This USB device for PCs together with the associated Chanalyzer 3.4 software makes it possible to appropriately visualize the radio activity in the frequency spectrum and to identify other WLAN networks that could interfere with the connection.

The "Network Stumbler" software for PCs can be used as an alternative. However, it is only capable of detecting WLAN networks. The identified WLAN networks are displayed in an easy-to-read list. The channel used by the network is listed next to the name of the WLAN network. A radio channel outside the frequency range of a WLAN network must be selected for the wireless handwheel system.

Figure 1: Spectrum analyzer view of radio environment

Three WLAN networks have been identified in this example environment. In this example, the wireless handwheel must be set on a vacant radio channel outside the frequency range of the WLAN networks.



Note

In the dialog on controls with software 606 42x-01 or 340 49x-06, the channels are designated by numbers from 0 to 15; these numbers do not correspond to the official ZigBee channel numbers. WiFi (WLAN) networks have channel frequencies and also channel designations, which, however, differ from those of the wireless handwheel system (ZigBee), see page 239.

Figure 2: Possible position of radio channel

Three WLAN networks have been identified in this example environment. In this example, the wireless handwheel could be set on the vacant ZigBee channel 15 outside the WLAN networks. The diagram shows the typical curve of a ZigBee radio device (white curve), such as the HR 550FS. Another possibility would be one of the vacant ZigBee channels 14, 20, 21 or 26.

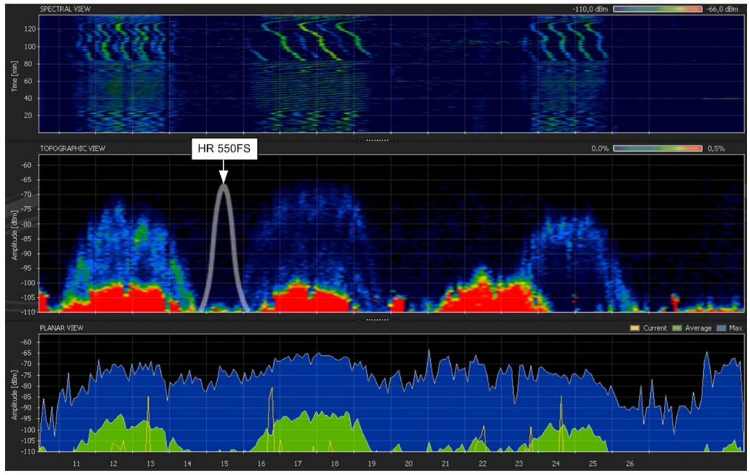
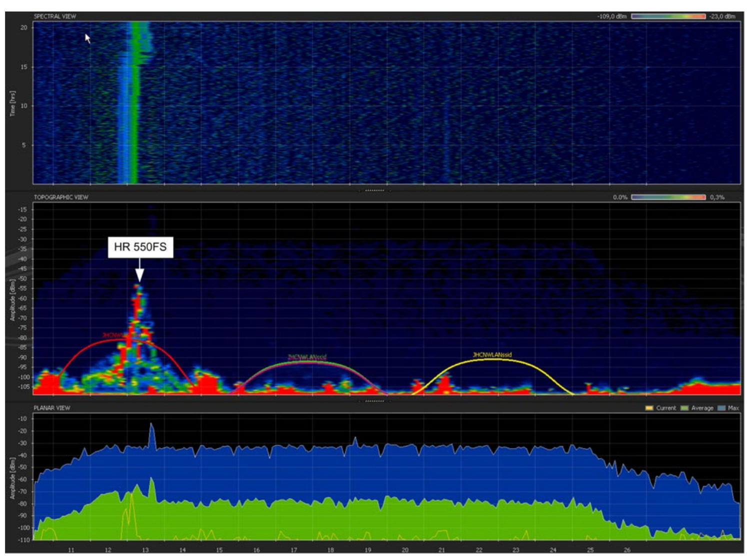


Figure 3: Radio channel of wireless handwheel collides with WLAN

In this example, the radio channel of the wireless handwheel is located in the range of the first WLAN network. This makes noise-free operation of the wireless handwheel impossible. Another channel must be selected for the wireless handwheel. In this case, ZigBee channel 20, 25 or 26 would be possible.

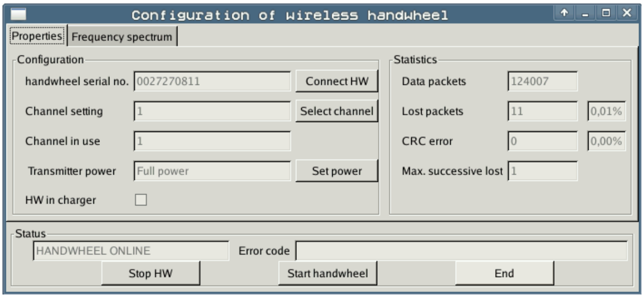


1.2.9 Information in the handwheel configuration dialog

The general procedure for the configuration of the HR 550FS handwheel is described in the iTNC 530 User's Manual.

To access the handwheel configuration dialog, proceed as follows:

- Press the MOD key
- Scroll through the soft-key row
- Press the SET UP WIRELESS HANDWHEEL soft key



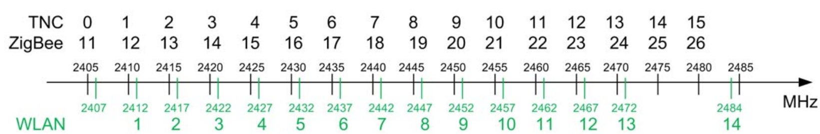
HEIDENHAIN generally recommends that in **`Channel setting`** you manually define the radio channel to be used. To select a vacant radio channel, please follow the information provided in this document. We advise against using the **`Best channel`** setting because it does not allow planning for the radio environment, and therefore unexpected disturbances during the operation of the wireless handwheel system may occur.

Channel setting

In the dialog on controls with software 606 42x-01 or 340 49x-06, the channels are designated by numbers from 0 to 15; these numbers do not correspond to the ZigBee channel numbers. Channel 0 at 2.405 GHz in the dialog on the control corresponds to ZigBee channel 11; channel 1 at 2.410 GHz in the dialog on the control corresponds to ZigBee channel 12, and so on. With the next software versions, the ZigBee channel numbers will also be displayed in the dialog on the control. The diagram below shows the relationship between the channel designations in the dialog on the control and the ZigBee channel numbers.

WiFi (WLAN) networks have channel frequencies and also channel designations, which, however, differ from those of the wireless handwheel system (ZigBee). For example, the WLAN channel 13 is at 2.472 GHz, and the wireless handwheel channel 13 is at 2.415 GHz. The diagram below shows the relationship between the channel designations in the dialog on the control, the ZigBee channel numbers and the WIFI (WLAN) channels. In addition, a detected WLAN channel has a channel width of 20 MHz, which must be taken into account in the coexistence planning. The four ZigBee channels next to a WLAN channel must therefore not be used in order to ensure that 20 MHz of channel width remain available.

Example: WLAN on channel 6
As a result, the ZigBee channels 16, 17, 18 and 19 must not be used. A wireless handwheel should be set only to ZigBee channel 15 or 20, which corresponds to channel 4 or 9 in the dialog on the control.



Formula for the next vacant wireless handwheel channel (TNC channel) depending on a WLAN channel:

TNC channel^{+} = WLAN channel + 3
TNC channel^{-} = WLAN channel – 2

Example:

TNC channel^{+} = 6 + 3 = 9
TNC channel^{-} = 6 – 2 = 4

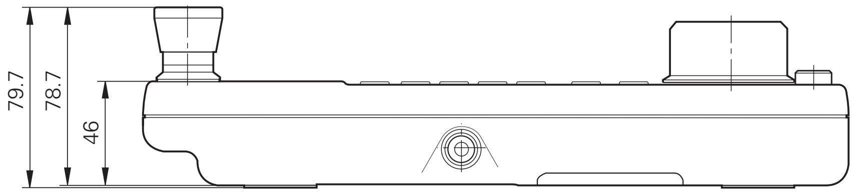
Radio connection during operation

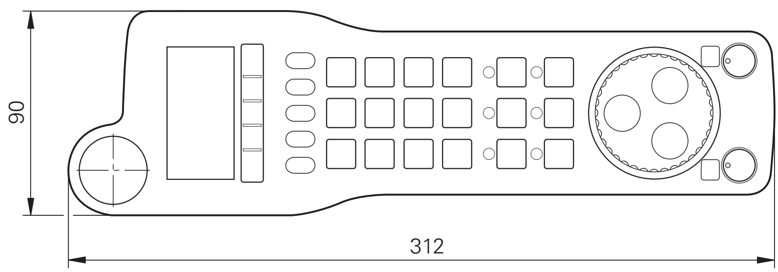
After installation and during normal operation, the following options of the handwheel configuration dialog on the control provide information about the quality of the radio connection:

- **Data packets**
  Number of data packets transmitted by radio transmission since the last activation of the wireless handwheel.
- **Lost packets** – **Number**
  Number of data packets lost since the last activation of the wireless handwheel. For example, if this value shows that more than two packets are lost within 10 seconds during radio transmission (wireless handwheel is outside the handwheel adapter), the limit range has been reached. The radio connection should be checked. You should change to another radio channel.
- **Lost packets** – **Percentage**
  Percentage of data packets lost since the last activation of the wireless handwheel.
  - Value < 0.05 %
    The radio connection functions properly.
  - Value between 0.05 % and 0.1 %
    The radio connection functions properly, but it is in the limit range. The radio connection should be checked. You should change to another radio channel.
  - Value > 0.1 %
    The radio connection is not reliable. We advise against operating the wireless handwheel on this channel. The radio connection should be checked. You must change to another radio channel.
- **Max. successive lost**
  Number of data packets that were lost successively, one after the other. If four successive data packets are lost, an emergency stop is triggered.
  - Value = 1
    The radio connection functions properly.
  - Value = 2
    The radio connection functions properly, but it is in the limit range. The radio connection should be checked. You should change to another radio channel.
  - Value = 3
    The radio connection is not reliable. We advise against operating the wireless handwheel on this channel. The radio connection should be checked. You must change to another radio channel.

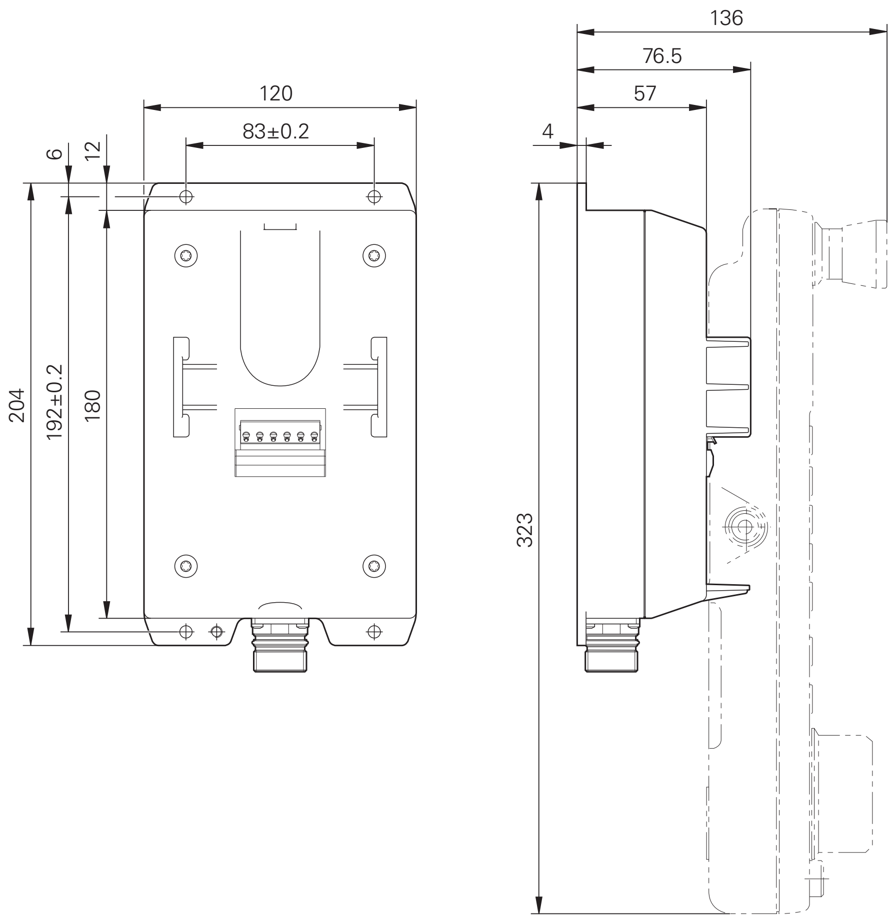
1.2.10 Dimensions

HR 550FS





HRA 551FS



1.3 More Information

With software version 340 492-06 SP02/340 493-06 SP02, the following Microsoft updates and patches were released by HEIDENHAIN for Windows XP:

- Microsoft Internet Explorer 8
- Microsoft service pack 3 for Windows XP
- In addition, Microsoft patches:
  KB981793, KB954459, KB956802, KB952069, KB923561, KB952004, KB956572, KB960803, KB959426, KB961501, KB970238, KB973815, KB973540, KB973507, KB973869, KB971657, KB960859, KB956744, KB971961, KB956844, KB975025, KB954155, KB974112, KB974571, KB969059, KB958869, KB974392, KB974318, KB973904, KB972270, KB978706, KB975713, KB978037, KB977914, KB975560, KB979309, KB978601, KB980232, KB981332, KB977816, KB979402, KB978338, KB978542, KB975562, KB979482, KB978695, KB980195, KB980218, KB2229593, KB2286198, KB981852, KB2160329, KB980436, KB981997, KB2079403, KB2115168, KB982214, KB982665

1 Update Information No. 30

1.1 Overview

1.1.1 Released service packs

The following service packs were released for **340 49x-04**:

- Service pack 09: January 2011

The following service packs were released for **340 49x-05**:

- Service pack 07: January 2011

The following service packs were released for **340 49x-06**:

- Service pack 01: was skipped
- Service pack 02: December 2010

The following service packs were released for **606 42x-01**:

- Service pack 01: August 2010
- Service pack 02: December 2010

1.2 NC Software 340 49x-06, 606 42x-01

Attention

In software versions 340 49x-06 SP 02 and 606 42x-01 SP 02, the brake control function might, in rare cases, be provided with incorrect information during the brake test for synchronized axes, which could cause hanging axes to fall down.

Affected software versions:

- NC software 340 49x-06, service pack 2
- NC software 606 42x-01, service pack 2

Description of the problem:

The brake test for synchronized axes was modified in the affected software versions. The newly implemented behavior makes it possible to detect which drives are operated together as a sychnronized axis and must be handled separately in the brake test. If a fault resulting in the drives being switched off (e.g. fault from the PLC) is initiated during these tests, incorrect information might be supplied to the brake control function if all of the conditions listed below are fulfilled:

- The brake test (MP2230.x) is active for servo drives belonging to a synchronized axis (MP850.x).
- The entries in MP2308.x (overlap time for braking) for the servo drives of a synchronized axis are almost identical (deviation < 50 ms).

The problem does not depend on the entry in MP860 bit#2.

If the event described above occurs, incorrect information is supplied for brake output X344 on the HEIDENHAIN power module, and therefore the brake is not closed when the servo drive is switched off.
Also, the PLC module 9159 (drive controllers are switched off) transmits an

incorrect status message to the PLC program regarding the closing of the brakes during the brake test. In this case, the status message of the PLC module also remains equal to zero during the overlap time for braking. During evaluation in the PLC program this can result in the brake being released while the servo drive is switched off, and therefore cause hanging axes to fall down.

However, the standstill monitoring function of the MC usually responds when a fault occurs. This fault initiates an internal emergency stop and leads to the switch-off of the „control-is-ready“ output after no more than 10 ms. As a result, the brakes are activated via the relay K1 if the machine is wired according to the HEIDENHAIN basic circuit diagram.

HEIDENHAIN recommends:

Starting immediately, please ship all machines with iTNC 530 and software versions 340 49x-06 and 606 42x-01 only with service pack 03 if you are using the brake test for synchronized axes! Also run a software update on the affected machines in the field.

Further procedure:

The problem is corrected with the following service packs:

- Software 340 49x-06 as of service pack 03
- Software 606 42x-01 as of service pack 03

The service packs should be available for download from our FileBase in February 2011.

If you need assistance or have any questions, please contact the responsible HEIDENHAIN service agency.

2 Introduction

2.1 General Information

2.1.1 Meaning of the symbols used in this manual

Danger

Failure to comply with this information could result in most serious or fatal injuries, and/or in substantial material damage.

Attention

Failure to comply with this information could result in injuries and interruptions of operation, including material damage.

Note

Tips and tricks for operation as well as important information, for example about standards and regulations as well as for better understanding of the document.

2.1.2 Proper operation

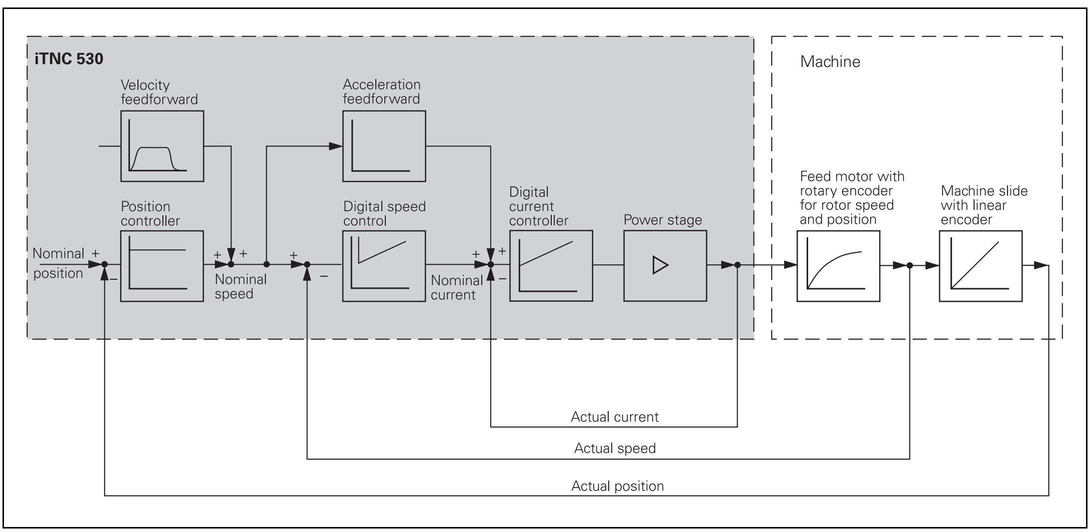
The described components may only be installed and operated as described in this manual. Commissioning, maintenance, inspection and operation are only to be performed by trained personnel.

HEIDENHAIN contouring controls and their accessories are designed for integration in milling, drilling and boring machines, and machining centers.

The **iTNC 530** features integral digital drive control and controls the power stages via PWM signals.

Integration of the drive controllers in the iTNC 530 offers the following advantages:

- All the software is contained centrally in the NC; this means that the individual components of the NC, such as feed axes, spindle, NC and PLC, are optimally matched.
- High control quality, because the position controller, speed controller and current controller are combined into one unit.
- The same functions are available for commissioning, optimizing and diagnosing feed drives as well as spindles.



The **iTNC 530** offers digital control for up to 14 axes and 2 spindles at spindle speeds up to 40 000 min^{-1}.

The **iTNC 530** is designed for connection of a compact or modular inverter system. A complete control package, including drives and HEIDENHAIN motors, can be delivered (see the "Inverter Systems and Motors" Technical Manual).

2.1.3 Trained Personnel

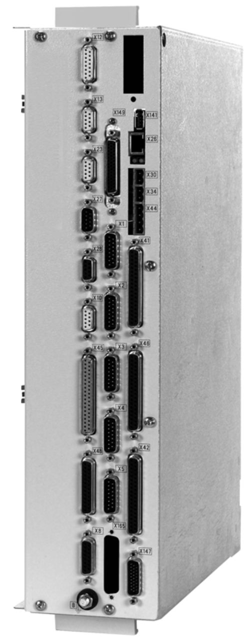
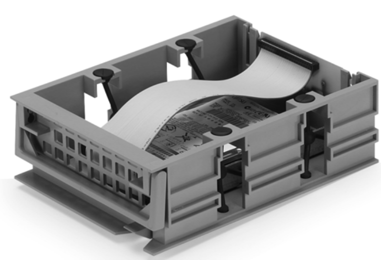
Trained personnel in the sense of this manual means persons who are familiar with the installation, mounting, commissioning, and operation of the HEIDENHAIN components. Furthermore, electrical engineering work on the system may be carried out only by trained electrical engineering technicians or persons trained specifically for the respective application.

Basically, persons who perform work on HEIDENHAIN components must meet the following requirements:

- They must have been trained or instructed in the standards of safety engineering.
- They must be familiar with the use of appropriate safety equipment (clothing, measuring systems).
- They should be skilled in first-aid practice.

2.2 Overview of Components

2.2.1 Main computer, hard disk and SIK

MC main computer	
The iTNC 530 comprises two components: ■ MC 42x(C) main computer (MC = Main Computer) ■ CC 42x(B) controller unit (CC = Controller Computer) The MC 42x(C) main computer is available in two versions: ■ Standard version MC 422C ■ MC 420 basic version with 5 position encoder inputs and reduced performance range. However, these functions can be activated with two code numbers. See page 299 for the performance range.	 MC 422 C  MC 420
The main computer consists of three components: ■ MC 42x(C) main computer ■ HDR hard disk ■ SIK system identification key	 HDR  SIK

Main computer (standard version)	Signal inputs	ID of the MC for BF 120 monitor	Replaces ID
MC 422C (as of software 340 490/491-02 service pack 05 or 340 422/423-14)			
Without position encoder inputs (for CC 424(B))	–	587 929-01	387 173-01
5 position encoder inputs	Position: 1 V_{PP}/ EnDat	587 932-01	387 181-01
10 position encoder inputs		587 934-01	387 189-01
MC 422C DP (with Windows XP)			
Without position encoder inputs (for CC 424(B))	–	631 209-01	387 175-01
5 position encoder inputs	Position: 1 V_{PP}/ EnDat	631 215-01	387 183-01
10 position encoder inputs		631 217-01	387 191-01

Main computer (basic version)	Signal inputs	ID of the MC for BF 120 monitor	ID of the MC for BF 150 screen
MC 420			
5 position encoder inputs	Position: 1 V_{PP}/ EnDat	–	515 929-01

HDR hard disk for iTNC 530	ID
Export version (MC 422C, MC 420)	524 571-51
Standard version (MC 422C, MC 420)	524 571-01
Export version with Windows XP (MC 422C DP)	617 969-51
Standard version with Windows XP (MC 422C DP)	617 969-01

SIK with NC software license for standard version with MC 422C		ID (standard)	ID (export)
iTNC 530			
	For 4 control loops for CC 4xx with 6/8 control loops For 7 control loops for CC 4xx with 10/12/14 control loops	389 764-01	389 764-51
	For 5 control loops for CC 4xx with 6/8 control loops For 8 control loops for CC 4xx with 10/12/14 control loops	389 764-02	389 764-52
	For 6 control loops for CC 4xx with 6/8 control loops For 9 control loops for CC 4xx with 10/12/14 control loops	389 764-03	389 764-53
	For 10 control loops for CC 4xx with 10/12/14 control loops	389 764-04	389 764-54
	For 11 control loops for CC 4xx with 12/14 control loops	389 764-05	389 764-55
iTNC 530 with Windows XP			
	For 4 control loops for CC 4xx with 6/8 control loops For 7 control loops for CC 4xx with 10/12/14 control loops	389 769-01	389 769-51
	For 5 control loops for CC 4xx with 6/8 control loops For 8 control loops for CC 4xx with 10/12/14 control loops	389 769-02	389 769-52
	For 6 control loops for CC 4xx with 6/8 control loops For 9 control loops for CC 4xx with 10/12/14 control loops	389 769-03	389 769-53
	For 10 control loops for CC 4xx with 10/12/14 control loops	389 769-04	389 769-54
	For 11 control loops for CC 4xx with 12/14 control loops	389 769-05	389 769-55

SIK with NC software license for basic version with MC 420		ID (standard)	ID (export)
iTNC 530			
	4 control loops without software options 1 + 2	510 085-01	510 085-51
	5 control loops without software options 1 + 2	510 085-02	510 085-52
	6 control loops without software options 1 + 2	510 085-03	510 085-53

Designation of MC 42x(B,C) and CC 42x

ID numbers of MC 42x(B,C) and CC 42x:

359 632-01

BasicIDnumber — Variant

The basic ID number indicates hardware differences.
This first digit of the variant number indicates hardware changes.

Variant	Changes to the MC 422
xxx xxx-y1	Initial version
xxx xxx-y2	Main computer revised (not for dual-processor version)

Variant	Changes to the MC 420
xxx xxx-01	Initial version

Variant	Changes to the MC 422B
xxx xxx-01	Initial version

Variant	Changes to the CC 422
xxx xxx-01	Initial version (speed controller → SH1, current controller → SH2)
xxx xxx-02	Modified controller (MC → SH1, speed controller → SH2)

Variant	Changes in the CC 424(B)
xxx xxx-01	Initial version

Variant	Changes to the MC 422C
xxx xxx-01	Initial version

Variant	Changes to the MC 422C DP
xxx xxx-01	Initial version

Important notes **MC 422C:**

> Attention
>
> Regarding the MC 422C hardware, please note:
>
> - The new MC 422C hardware only runs as of software 340 490/491-02 service pack 5 or software 340 422/423-14 or higher.
> - Software 340 49x-02 service pack 07 is available as a full version in order to go back to software 340 49x-02 from software 340 49x-03 on an MC 422C.
> - If you accidentally install a lower software version in connection with the MC 422C, then a corresponding error message appears when the control is booted. The boot procedure will be aborted.
> - If you have difficulties rebuilding, your control doesn't boot, or you want to exchange a defective MC 422B for an MC 422C, please contact your HEIDENHAIN service agency.

Proceed as follows when replacing an MC 422B with an MC 422C:

- ▶ When using the MC 422B, install at least the 340 490/491-02 SP5 or 340 422/423-14 software on the hard disk.
- ▶ Completely switch off your machine after you have finished the installation.

> Attention
>
> Do not engage or disengage any connecting elements while the unit is under power!

- ▶ Remove the MC 422B hardware. Remove the HDR on which the new software was installed.
  Please refer to the notes in the Technical Manual.
- ▶ Install the HDR in the new MC 422C hardware.
- ▶ Install the MC 422C and reengage all connecting elements. Please refer to the connection overview of the MC 422C.

Important notes about the MC 422C DP:

Attention

In regard to the MC 422C DP hardware, please note the following:

- The new MC 422C DP hardware only runs on software version 340 492/ 493-04 or higher.
- If you accidentally install a lower software version in connection with the MC 422C DP, then a corresponding error message appears when the control is booted. The boot procedure will be aborted.
- If you have difficulties rebuilding, your control doesn't boot, or you want to exchange a defective MC 422B DP for an MC 422C DP, please contact your HEIDENHAIN service agency.

Note

On the MC 422C DP, X129 and X126 must be connected with each other via an RJ 45 cable (Ethernet cable).
If this connection is missing or was removed during operation, the main computer must be shut down and switched off with the main switch to ensure that it is not under power. The HeROS computer cannot be rebooted until the connection between X129 and X126 has been restored.

When exchanging an MC 422B DP for an MC 422C DP, please proceed as follows:

- Use the TNCremoNT software for PCs to create a full backup of the PLC partition and the TNC partition of the MC 422B DP.
- Completely switch off your machine after you have created the backup.

Attention

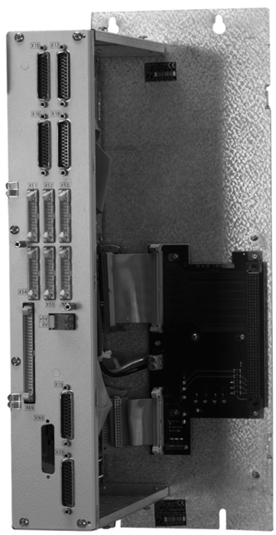
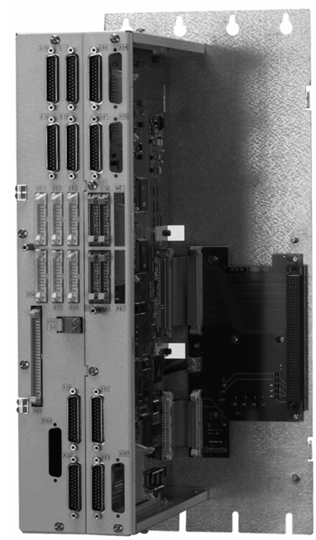
- Do not engage or disengage any connecting elements while the unit is under power!
- Do not in any case exchange the hard disk of an MC 422B DP for that of an MC 422C DP, or vice versa. Because of the different Windows operating systems, this causes faulty configurations that cannot be fixed.

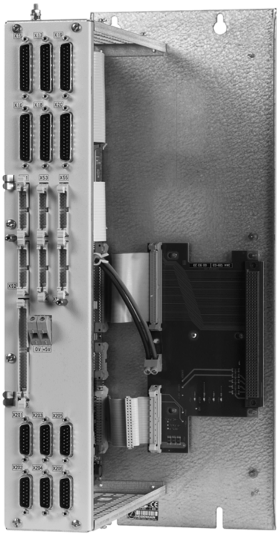
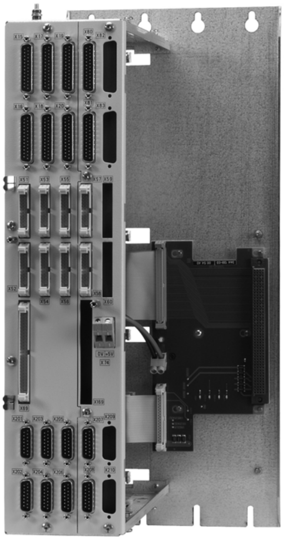
- Remove the MC 422B DP hardware.
  Please refer to the notes in the Technical Manual.
- Install the MC 422C DP with the appropriate new HDR and reconnect all connecting elements. Refer to the connection overview of the MC 422C DP.
- Completely boot the MC 422C DP until the control application is running. The boot process for the control application ("iTNC – Control Panel") may take somewhat longer than that of the MC 422B DP.
- Restore the full backup of the PLC partition and the TNC partition to the new HDR of the MC 422C DP.
- Reboot the control.

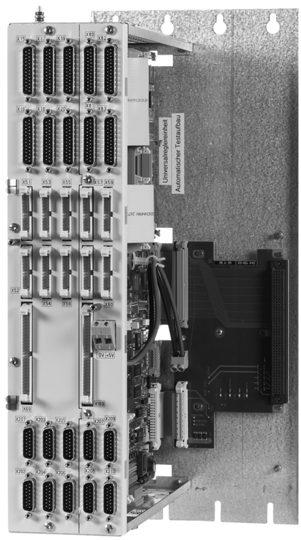
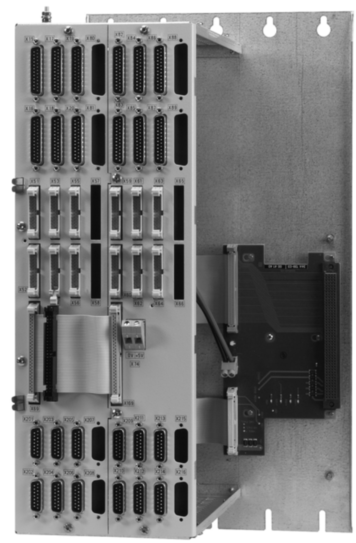
2.2.2 CC controller units

<table>
<tr><th colspan="2">CC 422 controller unit</th></tr>
<tr><td>CC 422 with 6 control loops

It is equipped with:
- 6 PWM outputs
- 6 speed encoder inputs</td><td>CC 422 with 10 (12) control loops

It is equipped with:
- 10 (12) PWM outputs
- 10 (12) speed encoder inputs</td></tr>
<tr><td></td><td></td></tr>
</table>

CC 424B controller unit	
CC 422B with 6 control loops It is equipped with: - 6 PWM outputs - 6 speed encoder inputs - 6 position encoder inputs	**CC 424B** with 8 control loops (as of SW 340 49x) It is equipped with: - 8 PWM outputs - 8 speed encoder inputs - 8 position encoder inputs
	
CC 422B with 10 control loops It is equipped with: - 10 PWM outputs - 10 speed encoder inputs - 10 position encoder inputs	**CC 424B** with 12 control loops (as of SW 340 49x) It is equipped with: - 12 PWM outputs - 12 speed encoder inputs - 12 position encoder inputs

	
CC 424B with 14 control loops (as of software 340 49x-02) It is equipped with: - 14 PWM outputs - 14 speed encoder inputs - 12 position encoder inputs	
An image was not available at press time.	

Controller unit	Signal inputs	Enabled control loops	Possible analog control loops	ID of the CC 422
CC 422				
Max. 6 digital speed control loops	1 V_{PP}/ EnDat	Depends on SIK	6 additional	359 651-xx
Max. 10 digital speed control loops			5 additional	359 652-xx
Max. 12 digital speed control loops			3 additional	359 653-xx

Controller unit	Signal inputs	Enabled control loops	Possible analog control loops	ID of the CC 424B
CC 424B				
Max. 6 digital control loops	1 V_{PP}/ EnDat	Depends on SIK	6 additional[a]	580 501-xx
Max. 8 digital control loops			6 additional[a]	580 510-xx
Max. 10 digital control loops			5 additional[a]	580 503-xx
Max. 12 digital control loops			3 additional[a]	580 511-xx
Max. 14 digital control loops			1 additional[a]	580 512-xx

a. In the CC 424B, analog control loops can be realized only in connection with an MC 42x(B,C) with position encoder inputs.

Further control loops can be enabled in addition to the control loops of the respective SIK version. Digital or analog control loops (as of NC software 340 420-04) can be enabled. There are software options for control-loop groups with four and with eight control loops:

SW option: Control loop groups:	ID
#77: 4 additional control loops	634 613-01
#78: 8 additional control loops	634 614-01

In addition to the four control loops of the basic version and the control-loop groups, up to eight individual control loops can be enabled through software options. These can also be enabled in addition to the control-loop groups:

SW option: Control loop groups:	ID
#0: 1st additional control loop	354 540-01
#1: 2nd additional control loop	353 904-01
#2: 3rd additional control loop	353 905-01
#3: 4th additional control loop	367 867-01
#4: 5th additional control loop	367 868-01
#5: 6th additional control loop	370 291-01
#6: 7th additional control loop	370 292-01
#7: 8th additional control loop	370 293-01

2.2.3 UV 106B power supply unit

General information

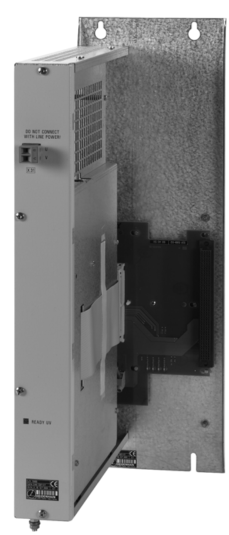
UV 106B power supply unit for analog HEIDENHAIN contouring controls

The **UV 106B** power supply unit was designed so that the iTNC 530 could be used with a compact, coordinated system for analog nominal shaft-speed interfaces (+/– 10 V).

It supplies the iTNC 530 with the supply voltages necessary for operation.

The **UV 106B** (ID 546 581-01) is being introduced as a replacement for the **UV 106** (ID 366 572-11).

ID 546 581-01 UV 106B



UV 106B

Specifications

Specifications	**UV 106B**
Supply voltage (at X31)	400 V~ ± 10%[a] 50 Hz
Protection	6.3 A (gR) Siemens SITOR type 6.3 A (gRL) SIBA type
Load capacity (5 V)	20 A
Power consumption	Max. 400 W
Degree of protection	IP 20
Module width	159 mm
Weight	4 kg
ID number	546 581-xx

a. An isolating transformer is not necessary for connecting the UV 106B.

2.2.4 UV 105B (non-HEIDENHAIN inverter systems)

General information

The **UV 105B** (ID 532 581-01) was designed solely for the use of HEIDENHAIN controls in connection with non-HEIDENHAIN inverter systems. It is essential for the supply voltages of the HEIDENHAIN control units.

UV 105B power supply unit for the operation of HEIDENHAIN controls with non-HEIDENHAIN inverter systems An LED showing the readiness of the supply voltage is located on the front of the UV 105B. ID 532 581-01 UV 105B	 UV 105B

2.2.5 UV 105 power supply unit

UV 105 power supply unit

The UV 105 serves to supply the power to the CC 42x if a non-HEIDENHAIN inverter is used, or, if required, to supply additional power if HEIDENHAIN inverter components are used (See "Power Supply for the iTNC 530" on page 351).

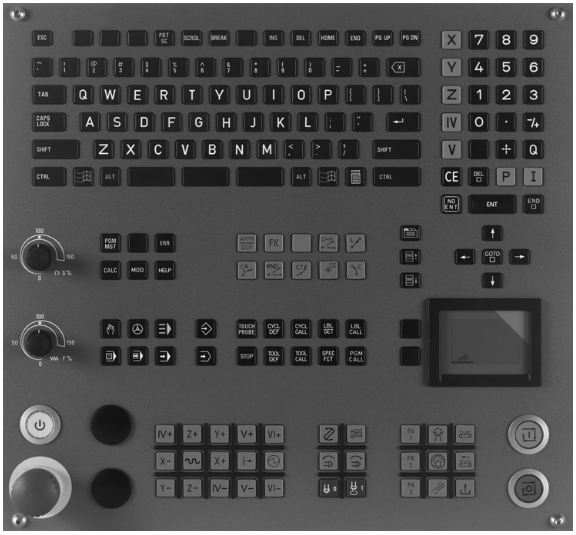
If a non-HEIDENHAIN inverter system is used, the adapter connector is connected to X69 of the UV 105.

The cover for the UV 105 and the adapter connector for X69 are included in the items supplied.

ID 344 980-xx	UV 105
ID 349 211-01	Adapter connector for X69

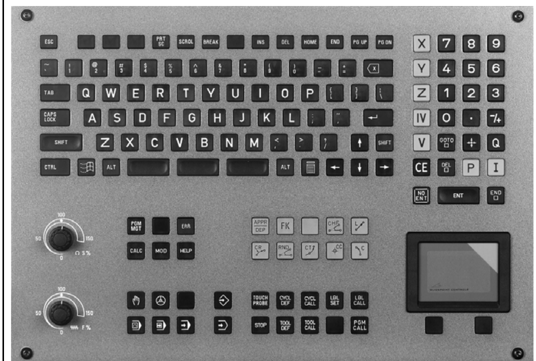
Variant	Changes to UV 105
xxx xxx-01	Initial version
xxx xxx-02	Modification for double-row configuration
xxx xxx-12	Version only for HEIDENHAIN inverters
xxx xxx-13	Version for HEIDENHAIN and non-HEIDENHAIN inverters
xxx xxx-14	Leads and ribbon cables elongated

2.2.6 Keyboard units and monitors

TE 535Q operating panel The NC keys of the operating panel are identical to those of the TE 530B. In addition, machine operating keys were integrated. **Machine operating panel:** ■ 6 axis-direction keys ■ 16 function keys ■ Keys for NC start and NC stop (illuminated) ■ Keys for spindle start and spindle stop ■ All keys in the machine operating panel are snap-on keys. ■ EMERGENCY STOP button ■ Key for control voltage ON (RAFI key, illuminated) ■ Two bore holes (22 mm) for additional RAFI buttons (shipped blocked with a cover) or keylock switches ID 547 577-xx TE 535Q	

TE 530 operating panel with touchpad

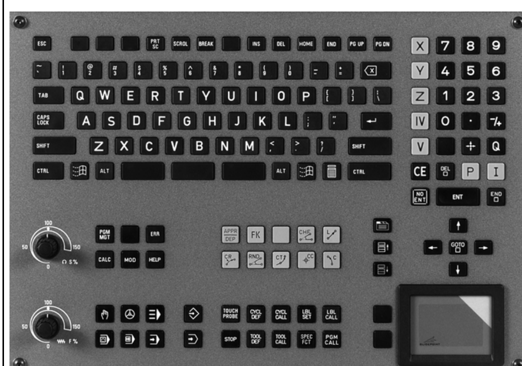
TE 530 ID 359 906-01



TE 530B operating panel with touchpad

Function keys for the new **smarT.NC** operating mode, as well as the new SPEC FCT key for calling special TNC functions.
TE 530B for **smarT.NC**
and **SPEC FCT** key ID 519 441-11

The IV and V keys are snap-ons, and can be switched (See "Key symbols" on page 272).



Note

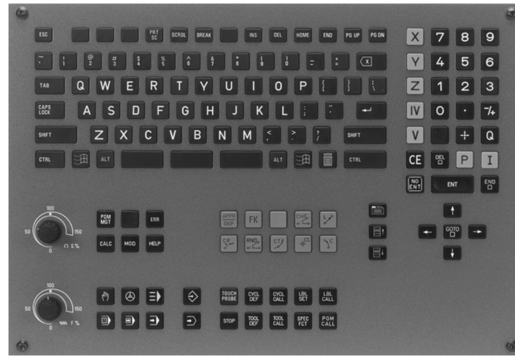
Please note the following limitations when using a TE with a touchpad:

- The length of the required USB cable is limited to 30 m.
- When using two operating panels, which are switched with the BTS 1xx, only one touchpad may be active (touchpad at X142 of the MC).

TE 520B operating panel with function keys for the new alternative `smarT.NC` operating mode, as well as the new SPEC FCT key for calling special TNC functions. This operating panel corresponds to the TE 530B, but without the touchpad.

ID 535 835-01 TE 520B

The IV and V keys are snap-ons, and can be switched (See "Key symbols" on page 272).

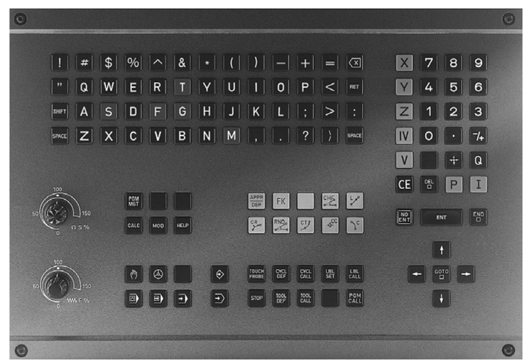


TE 420 operating panel

ID 313 038-12

The IV and V keys are snap-ons, and can be switched (See "Key symbols" on page 272).

Horizontal rows to match the design of the flat-panel display
ID 316 343-01



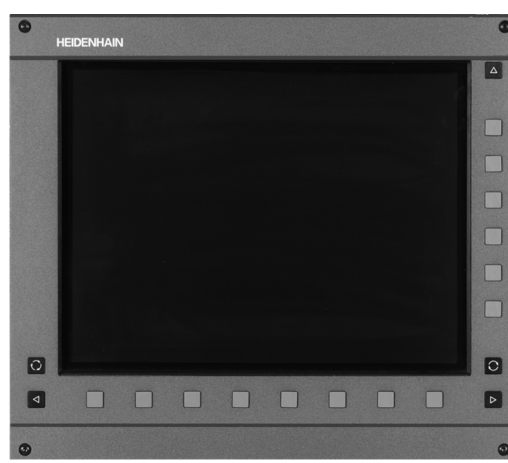
BF 150 visual display unit

15.1-inch color flat panel display (1024 x 768 pixels) with the following keys:

- 1 x 8 soft keys, 1 x 6 soft keys for PLC
- 3 soft keys for switching soft-key rows
- Screen layout
- Operating mode switchover

ID 353 522-04

Horizontal rows for the design
ID 339 516-02 (bottom)
ID 339 516-04 (top)



BF 120 visual display unit (discontinued as of software 340 49x) 10.4-inch color flat panel display (640 x 480 pixels) with the following keys: ■ 8 soft keys ■ 2 soft keys for switching soft-key rows ■ Screen layout ■ Operating mode switchover ID 313 506-02	

Variant	Changes in the BF 150
xxx xxx-01	Initial version
xxx xxx-02	New housing front
xxx xxx-03	New display model (compatible to previous versions)
xxx xxx-04	New display model (compatible to previous versions)

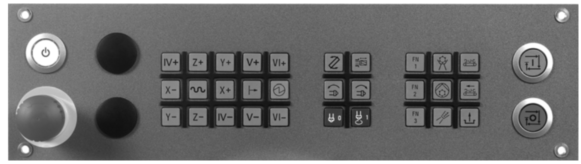
BTS 1x0 screen-keyboard switching unit With the BTS 1x0, it is possible to connect two monitors and two operating panels to an MC 42x(B,C). ID 353 544-01 BTS 150 (2 x BF 150) ID 329 965-02 BTS 120 (2 x BF 120)	

MB 520 machine operating panel
It has the same machine operating panel functions as those integrated in the TE 535Q.

Machine operating panel:

- 6 axis-direction keys
- 16 function keys
- Keys for NC start and NC stop (illuminated)
- Keys for spindle start and spindle stop
- All keys in the machine operating panel are snap-on keys.
- EMERGENCY STOP button
- Key for control voltage ON (RAFI key, illuminated)
- Two bore holes (22 mm) for additional RAFI buttons (shipped blocked with a cover) or keylock switches

ID 628 040-xx MB 520



MB 420 machine operating panel

Machine operating panel with snap-on (exchangeable) keys (See "Key symbols" on page 272).

Key assignment:

- Emergency stop
- Machine control voltage
- NC start / NC stop
- Direction keys for 5 axes
- Rapid traverse
- Coolant
- Spindle start
- Spindle stop
- 7 keys for machine functions
  - FN 1 to FN 5 (standard assignment)
  - Retract axis, tool change, unlock tool, menu selection→, unlock door, rinse water jet, chip removal (assignment for HEIDENHAIN basic PLC program)

ID 283 757-33	Standard assignment
ID 293 757-45	Machine key assignment for HEIDENHAIN basic PLC program

2.2.7 Handwheels

All handwheels are available with and without detent. For handwheels with detent, the cogging torque prevents movements of the handwheel due to motions or vibrations of the machine. On handwheels without detent, this is prevented by a defined holding torque.

Handwheels with detent feature 100 detent positions per revolution, i.e. every 3.6°. The increment per step is specified as follows:

- HR 130: By the user via conversational programming on the control
- HR 410: By the user on the HR 410 via 3 different step increments predefined by the machine manufacturer
- HR 420, HR 520: By the user on the HR 420 or HR 520

Handwheels with detent are supported as of software versions 340 422-09, 340 423-09, 340 480-09, 340 481-09 and 340 49x-01

<table>
<tr>
<td>

HR 410 handwheel

Portable electronic handwheel with snap-on (exchangeable) keys, (See "Key symbols" on page 272).

- Five axis selection keys
- Keys for traverse direction
- Keys for preset feeds
- Actual-position-capture key
- Three keys for machine functions (definable via PLC)
  - Spindle right/left/stop
  - NC start/stop, spindle start; (for HEIDENHAIN basic PLC program)
- Two permissive buttons (24 V)
- Emergency stop button (24 V)
- Magnetic holding pads

ID 296 469-54HR 410 handwheel (spindle right/ left/stop)

ID 296 469-55HR 410 handwheel (NC start/stop, spindle start)

ID 535 220-55HR 410 handwheel (NC start/stop, spindle start) with detent

</td>
<td>

</td>
</tr>
</table>

HR 420 handwheel

Portable electronic handwheel with

- Display for operating mode, actual position value, programmed feed rate and spindle speed, error messages
- Spindle speed and feed-rate override
- Selection of axes via keys or soft keys
- Actual position capture
- NC start/stop
- Spindle start/stop
- Keys for traverse direction
- Two permissive buttons (24 V)
- Emergency stop button (24 V)
- Magnetic holding pads
- Mount for attaching the handwheel to the machine

ID 375 239-01HR 420 handwheel

ID 512 367-01 HR 420 handwheel with detent

HR 520 handwheel

- Display for operating mode, actual position value, programmed feed rate and spindle speed, error messages
- Graphic display,
  Resolution: 128 x 64 pixels, 6-line display
- Spindle speed and feed-rate override
- Selection of axes via keys or soft keys
- Actual position capture
- NC start/stop, spindle start/stop
- 6 freely programmable PLC keys with LED
- Keys for traverse direction
- Exchangeable snap-on keys for PLC functions and maintenance
- Integrated permissive key and emergency stop button (24 V)
- Magnetic holding pads
- Mount for attaching the handwheel to the machine (ID 591 065-xx)
- Weight: Approx. 1 kg

ID 670 302-xx HR 520 handwheel

ID 670 303-xx HR 520 handwheel with detent

ID 312 879-01Connecting cable to cable adapter (spiral cable 3 m)
ID 296 467-xxConnecting cable to cable adapter (normal cable)
ID 296 687-xxConnecting cable to cable adapter (with metal armor)
ID 296 466-xxAdapter cable to MC 42x(B,C)
ID 281 429-xxExtension to adapter cable
ID 271 958-03Dummy plug for emergency stop circuit

HR 130 handwheel

Panel-mounted handwheel

ID 254 040-05	With ergonomic knob, radial cable outlet
ID 540 940-01	Same as above, but with detent

HRA 110 handwheel adapter

For connecting up to three **HR 150** handwheels to the TNC.
The axes and the subdivision factor are selected via rotary switch.

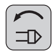
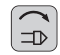
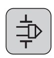
The most important characteristics of the HR 150 are:

- 11 µA_{PP} signals, line count 5000
- 1 m cable with 9-pin connector (male)
- Cable outlet for axial and radial use
- Ergonomic knurled control knob
- Version with mechanical detent, 100 positions per 360°

ID 261 097-03	HRA 110
ID 540 940-06	HR 150 with mechanical detent
ID 540 940-07	HR 150 without mechanical detent
ID 270 908-01	HRA selector switch

2.2.8 Key symbols

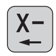
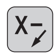
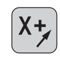
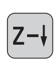
Key symbols for the spindle

Key	Description Print/Background ID	Key	Description Print/Background ID
	Spindle stop White/Red 330 816-08		Spindle start White/Green 330 816-09
	Spindle direction left Black/Gray 330 816-40		Spindle direction right Black/Gray 330 816-41
	Spindle stop White/Red 330 816-47		Spindle start White/Green 330 816-46
	Clamp the spindle Black/Gray 330 816-48		

Key symbols with axis designations

Key	Description Print/Background ID	Key	Description Print/Background ID
X	X Black/Orange 330 816-24	Y	Y Black/Orange 330 816-36
Z	Z Black/Orange 330 816-25	A	A Black/Orange 330 816-42
B	B Black/Orange 330 816-26	C	C Black/Orange 330 816-23
U	U Black/Orange 330 816-43	V	V Black/Orange 330 816-38
W	W Black/Orange 330 816-45	IV	IV Black/Orange 330 816-37

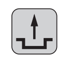
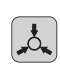
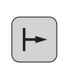
Axis direction keys for the principal axes

Key	Description Print/Background ID	Key	Description Print/Background ID
	X– Black/Gray 330 816-63		X+ Black/Gray 330 816-64
	X– <– Black/Gray 330 816-18		X+ –> Black/Gray 330 816-17
	X’– –> Black/Gray 330 816-0W		X’+ <– Black/Gray 330 816-0V
	X– <– Black/Gray 330 816-0N		X+ –> Black/Gray 330 816-0M
	Y– Black/Gray 330 816-67		Y+ Black/Gray 330 816-68
	Y’– –> Black/Gray 330 816-21		Y’+ <– Black/Gray 330 816-20
	Y– <– Black/Gray 330 816-0P		Y+ –> Black/Gray 330 816-0R
	Y– –> Black/Gray 330 816-0D		Y+ <– Black/Gray 330 816-0E
	Z– Black/Gray 330 816-65		Z+ Black/Gray 330 816-66
	Z– <– Black/Gray 330 816-19		Z+ –> Black/Gray 330 816-16
	Z’– –> Black/Gray 330 816-0L		Z’– <– Black/Gray 330 816-0K

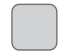
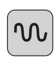
Key symbols for axis-direction keys for rotary and additional linear axes

Key	Description Print/Background ID	Key	Description Print/Background ID
A–	A– Black/Gray 330 816-95	A+	A+ Black/Gray 330 816-96
B–	B– Black/Gray 330 816-97	B+	B+ Black/Gray 330 816-98
C–	C– Black/Gray 330 816-99	C+	C+ Black/Gray 330 816-0A
U–	U– Black/Gray 330 816-0B	U+	U+ Black/Gray 330 816-0C
V–	V– Black/Gray 330 816-70	V+	V+ Black/Gray 330 816-69
W–	W– Black/Gray 330 816-0G	W+	W+ Black/Gray 330 816-0H
IV–	IV– Black/Gray 330 816-71	IV+	IV+ Black/Gray 330 816-72

Key symbols for machine functions

Key	Description Print/Background ID	Key	Description Print/Background ID
SPEC FCT	Special function Black/Gray 330 816-0X	FCT A	Function A White/Black 330 816-30
FCT B	Function B White/Black 330 816-31	FCT C	Function C White/Black 330 816-32
FN 1	Function 1 Black/Gray 330 816-73	FN 2	Function 2 Black/Gray 330 816-74
FN 3	Function 3 Black/Gray 330 816-75	FN 4	Function 4 Black/Gray 330 816-76
FN 5	Function 5 Black/Gray 330 816-77		Unlock door Black/Gray 330 816-78
	Unlock door Black/Gray 330 816-79		Coolant Black/Gray 330 816-80
	Coolant (internal) Black/Gray 330 816-0S		Coolant (external) Black/Gray 330 816-0T
	Rinse water jet Black/Gray 330 816-81		Spotlight Black/Gray 330 816-82
	Chip removal Black/Gray 330 816-83		Chip conveyor Black/Gray 330 816-84
	Tool change Black/Gray 330 816-89		Tool changer left Black/Gray 330 816-85
	Tool changer right Black/Gray 330 816-86		Unlock tool Black/Gray 330 816-87
	Unlock tool Black/Gray 330 816-88		Lock tool Black/Gray 330 816-94
	Lock tool Black/Gray 330 816-0U		Retract axis Black/Gray 330 816-91

Other key symbols

Key	Description Print/Background ID	Key	Description Print/Background ID
	No symbol –/Black 330 816-01		No symbol –/Gray 330 816-61
	NC start White/Green 330 816-11		NC stop White/Red 330 816-12
	NC start White/Green 330 816-49		NC stop White/Red 330 816-50
	Feed rate 1 Black/Gray 330 816-33		Feed rate 2 Black/Gray 330 816-34
	Rapid traverse Black/Gray 330 816-35		Permissive button White/Green 330 816-22
	Permissive button Black/Gray 330 816-90		Actual position capture White/Black 330 816-27
	– White/Black 330 816-28		+ White/Black 330 816-29
	Menu selection –> Black/Gray 330 816-92		Menu selection <– Black/Gray 330 816-93
	0 Black/Gray 330 816-0Y		

2.2.9 Touch probes

TT 130 tool touch probe

Triggering touch probe for measuring tools.

ID 296 537-xx	TT130
ID 335 332-xx	Adapter cable for connection to the MC 42x(B,C)

TT 140 tool touch probe

Triggering touch probe with rated break point of the connection pin for the probe head and optical deflection display. An additional connection pin is delivered with the touch probe.

ID 527 797-03	TT140
ID 559 758-01	Connection pin
ID 335 332-xx	Adapter cable for connection to the MC 42x(B,C)

TS 220 Touch Probe

Triggering touch probe with cable connection for signal transmission for machines with manual tool change. For workpiece setup and measurement during machining.

ID 293 488-xx	TS 220
ID 633 613-xx	Adapter cable for connection to the system PL or the UEC

TS 740, TS 640, TS 444, TS 440 touch probes

Triggering touch probes with infrared transmission, for workpiece setup and measurement during machining. For machines with automatic tool changer.

- TS 440 with compact dimensions
- TS 444 with alternative battery-free power supply via compressed air through the spindle head
- TS 640 with wide-range infrared transmission and long operating time
- TS 740 with high probing accuracy and repeatability, and low probing forces

The infrared transmission is established between the TS touch probe and the SE transceiver unit. The following SE units can be combined with the TS touch probes:

- SE 640 for integration in the machine workspace
- SE 540 for integration in the spindle head

ID 573 757-xx	TS 740
ID 620 189-xx	TS 640
ID 620 046-xx	TS 440
ID 588 008-xx	TS 444
ID 631 225-xx	SE 640 transmitter-receiver unit
ID 626 001-xx	SE 540 transmitter/ receiver unit

TS 640, TS 740

TS 440, TS 444

SE 640

SE 540

2.2.10 MS110, MS111 installation kit

MS 110 without additional 24 V supply

Sometimes limited space prevents the control and inverter system from being mounted in the same row in a machine’s electrical cabinet, meaning that they must be mounted in two separate rows. In other cases the design calls for a second electrical cabinet to house the inverter system. This means that the distribution and arrangement of the components can be very different from case to case.

In order to establish an electrical connection (immune to noise) between the components of the inverter system, the MS 1xx installation kits are needed.

In most cases the additional 24 V are not needed, since the UVR 1xxD provides enough current for the fans and the MS 110 is sufficient.

ID 658 132-xx MS 110

MS 111 with 24 V supply

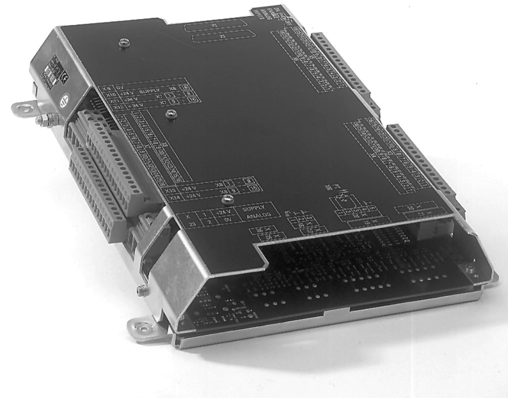
For inverter systems with many powerful UM 1xxD power modules, it might be the case that the current provided by the UV(R)1xx power supply unit for the fans of the UMs does not suffice to guarantee safe and reliable operation of the fans.

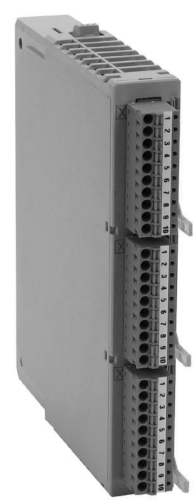
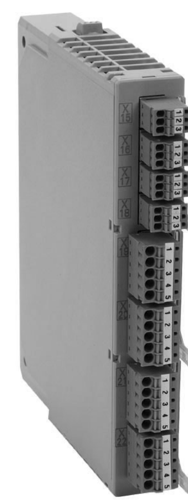
If it is exceeded, then an MS111 must be used in the inverter row where the current consumption is very high.

ID 673 685-xx MS 111

Variant	Changes to MS 11x
xxx xxx-01	Initial version

2.2.11 Other accessories

PL 410B PLC input/output unit for additional PLC inputs and outputs ID 263 371-12 — 64 inputs, 31 outputs ID 263 371-02 — 64 inputs, 31 outputs, 4 analog inputs ± 10 V, 4 inputs for Pt 100 thermistors **PL 405B** PLC input/output unit ID 263 371-22 — 32 inputs, 15 outputs	
PL 510 PLC input/output unit This is a modular I/O system for the expansion of PLC inputs and outputs. The PL 510 consists of the **PLB 510** basic module and the following components: ■ **PLD 16-8** I/O module with 16 digital inputs and 8 digital outputs ■ **PLA 4-4** analog module with 4 analog inputs for Pt 100 thermistors and 4 ±10 V analog inputs ■ Empty housing for partial assembly The PL 510 can be mounted on a top hat rail (NS 35 EN 50022). The PL 510 equipped (completely or partially) with PLD 16-8 is compatible with PL 410B/ PL 405B.	
PLB 510 with 4 slots ID 358 849-01 — PLB 510	

PLB 511 with 6 slots ID 556 941-01 PLB 511	
PLB 512 with 8 slots ID 557 125-01 PL 512	
360 916-01 PLD 16-8 input/output module 366 423-01 PLA 4-4 analog module 383 022-01 Empty housing	  PLD 16-8 PLA 4-4

Further components	ID
Adapters for encoder signals	
TTL (HEIDENHAIN layout) / 1 V_{PP}	317 505-01
TTL (SIEMENS layout) / 1 V_{PP}	317 505-02
11 μA_{PP} to MC 42x(B,C)	317 505-05
11 μA_{PP} / 1 V_{PP}	313 119-01

CML 110 capacitor module for 24 V power supply

Specifications

- Supply voltage: 24 V
- Capacitance: 8.3 F
- Max. charging current: 2.4 A
- Internal resistance (discharge)

Maximum:	156 mOhm
Typically:	65 mOhm

- Discharge current: 30 A

574 087-02	CML 110

USB hub for integration in the operating panel

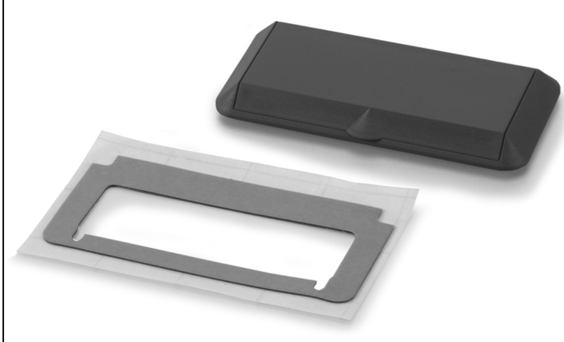
Specifications

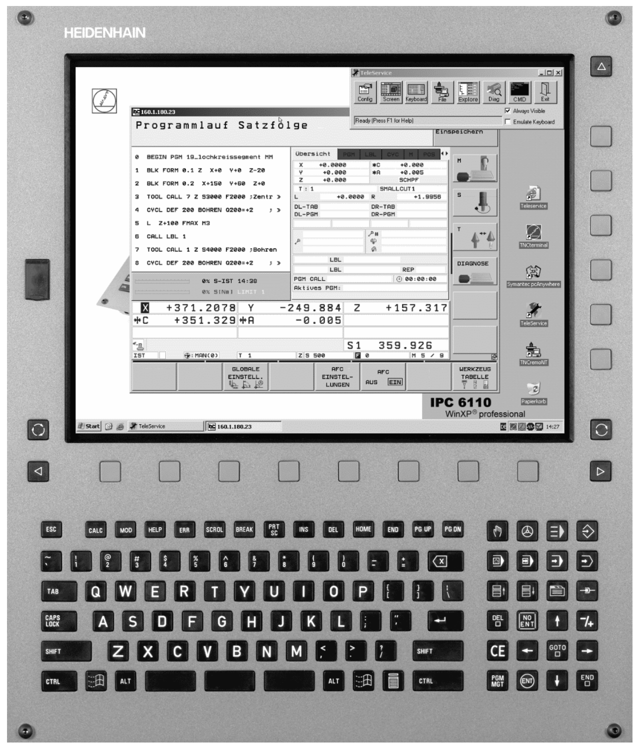
- Supply voltage: 24 V
- Inputs:
  X1 – Power supply 24 V / max. 300 mA
  (as per IEC 742/VDE0551)
  X140 – USB-B input
- USB-A output on rear:
  X141, X142
- USB-A output on front:
  X143, X144
- Maximum load capacity of the outputs:
  2 x 100 mA, 2 x 500 mA, any distribution

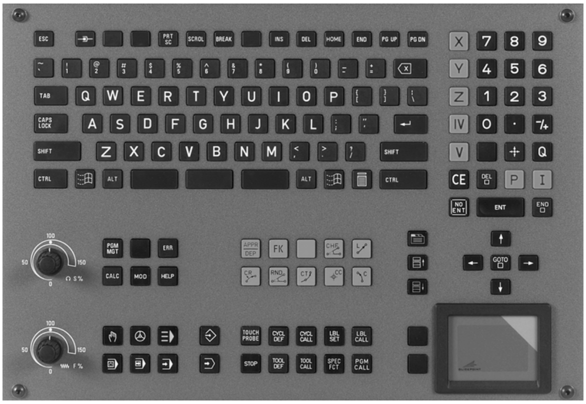
ID 582 884-01	USB hub

Cover cap (IP 52)

ID 508 921-01	Cover cap



IPC 6110 main computer	
ID 597968-02	■ Main computer ■ Integrated ASCII keyboard with additional keypad ■ 15-inch screen with soft keys ■ 3x USB connections ■ Slot for CompactFlash memory card, types I and II ■ Ethernet connection ■ V.24/RS-232-C data interface ■ Power supply connection
Controls on the IPC 6110 ■ English ASCII keyboard with function keys ■ 24-key keyboard block with snap-on keys ■ 8 horizontal soft keys ■ 6 vertical soft keys ■ 2 keys for horizontal menu change ■ Key for vertical menu change ■ Key for view change ■ Key for changing screen layout	

IPC 6120 kit	
ID 664 01 "IPC 6120 Adv. Kit" (with TE 630 and CFR) ID 664 01 "IPC 6120 Basic Kit" (with TE 620 and CFR)	■ Main computer ■ Separate TNC operating panel TE 630 or TE 620 ■ 15-inch screen ■ 3x USB connections ■ Slot for CompactFlash memory card, types I and II ■ Ethernet connection ■ V.24/RS-232-C data interface ■ Power supply connection ■ CFR TNCterm
Controls on the IPC 6120 (can be used only as a kit with TE 6xx) ■ Complete TNC operating panel with programming keys ■ Two override potentiometers (can be accessed for switching with analog inputs, not included in delivery) ■ Mouse pad with keys (only "IPC 6120 Adv. Kit") ■ 8 horizontal soft keys ■ 6 vertical soft keys ■ 2 keys for horizontal menu change ■ Key for vertical menu change ■ Key for view change ■ Key for changing screen layout	 

2.2.12 Documentation

HEIDENHAIN offers the following User's Manuals for the control:

- User's Manual, Conversational Format: ID 670 387-xx.
- User’s Manual for ISO programming: ID 670 391-xx.
- User's Manual, Cycle Programming: ID 670 388-xx.
- User's Manual, smarT.NC Programming: ID 533 191-xx.

The HEIDENHAIN inverters and motors for the iTNC 530 are described in the "Inverter Systems and Motors" Technical Manual.

The components required for operating the iTNC 530 with non-HEIDENHAIN inverter systems are described in "Technical Information for Operation of SIMODRIVE and POWER DRIVE Inverter Systems."

2.3 Brief Description

Specifications	iTNC 530
MC 422C	
	■ Processor: Pentium III with 800 MHz ■ 512 MB RAM ■ Bus frequency: 133 MHz ■ 5 or 10 position encoder inputs 1 V_{PP} or EnDat, or without position encoder inputs (for CC 424(B)) ■ Unambiguous identification of MC 422C through System Identification Key (SIK)
MC 422C DP (with Windows XP)	
	■ Real-time computer: Pentium III with 800 MHz ■ Windows computer: Pentium M with 1.8 GHz ■ Main memory of real-time computer: 512 MB RAM ■ Main memory of Windows computer: 512 MB RAM ■ Same dimensions as the MC 422B DP ■ 2 x USB ■ 5 or 10 position encoder inputs 1 V_{PP} or EnDat, or without position encoder inputs (for CC 424(B)) ■ Unambiguous identification of MC 422C through System Identification Key (SIK)
MC 420	
	■ Processor: Celeron with 400 MHz ■ 512 MB RAM ■ Bus frequency: 100 MHz ■ Five position encoder inputs 1 V_{PP} or EnDat ■ Unambiguous identification of MC 420 through System Identification Key (SIK)
CC 422	
	All speed encoder inputs 1 V_{PP} or EnDat
	■ 6 speed encoder inputs with 1 V_{PP} or EnDat for axes and spindle(s) Basic version: 4 speed control loops ■ 10 speed encoder inputs with 1 V_{PP} or EnDat for axes and spindle(s) Basic version: 7 speed control loops ■ 12 speed encoder inputs with 1 V_{PP} or EnDat for axes and spindle(s) Basic version: 7 speed control loops
CC 424(B)	
	All position and speed encoder inputs 1 V_{PP} or EnDat
	■ 6 position and 6 speed encoder inputs with 1 V_{PP} or EnDat for axes and spindle(s) Basic version: 4 speed control loops ■ 10 position and 10 speed encoder inputs with 1 V_{PP} or EnDat for axes and spindle(s) Basic version: 7 speed control loops

Specifications		iTNC 530
Options		
		Additional control loops or software options can be enabled by entering a code number.
Display		
		■ 15.1-inch TFT color flat-panel display ■ 10.4-inch TFT color flat-panel display
Program memory		
		Hard disk with > 6 GB
Input resolution and display step		
		Up to 0.1 µm for linear axes Up to 0.0001° for angular axes
Interpolation		
	Straight line	5 of 14 axes
	Straight line (with MC 42x(B,C) E[a])	4 of 14 axes
	Circle	■ 2 of 14 axes ■ 3 of 14 axes with tilted working plane
	Helix	Superimpositioning of circular and straight paths
	Spline (software option 2)	Cubic splines can be executed
Block processing time		
		0.5ms Basic version: 3.6 ms (0.5 ms with software option 2)

a. Export version

Machine interfacing	iTNC 530
Feedback control with CC 422	
Position control resolution	$\frac{\text{Signal period}}{1024}$
Path interpolation	1.8ms
Fine interpolation	–
Cycle time of position controller	Minimum 1.8 ms
Cycle time of speed controller	600 µs
Cycle time of current controller	PWM frequency — Cycle time 3333 Hz — 150 µs 4166 Hz — 120 µs 5000 Hz — 100 µs 6666 Hz — 75 µs 8333 Hz — 60 µs 10000 Hz — 50 µs
Maximum motor speed	$n_{max} = \frac{f_{PWM} \cdot 60000\ \text{min}^{-1}}{p \cdot 5000\ \text{Hz}}$ n_{max}: Maximum motor speed [min^{-1}] f_{PWM}: PWM frequency [Hz] p: Number of pole pairs The following PWM frequencies are available: 3333 Hz, 4000 Hz, 5000 Hz, 6666 Hz, 8000 Hz, 10000 Hz
or by way of:	Maximum signal frequency of the motor encoder = 400 kHz $n_{max} = \frac{f_{max} \cdot 60000\ [\text{s/min}]}{\text{ELC}}$ n_{max}: Maximum motor speed [min^{-1}] f_{max}: Maximum signal frequency of the motor encoder [kHz] ELC: Encoder line count Example: $n_{max} = \frac{400\text{kHz} \cdot 60000\ [\text{s/min}]}{2048}$ $n_{max} = 11718.75\ \text{min}^{-1}$

Machine interfacing	iTNC 530
Maximum feed rate by way of:	Maximum signal frequency of the position encoder: ■ At 1 V_{PP}:27 kHz or 400 kHz ■ At 11 μA_{PP}:27 kHz or 140 kHz Calculation of the maximum feed rate F_{max}: $F_{max} = SP \cdot f_{max} \cdot 6 \cdot 10^{-3} \left[\frac{m \cdot s}{\mu m \cdot min}\right]$ F_{max}: Maximum feed rate [m/min] f_{max}: Maximum signal frequency of the position encoder [kHz] SP: Signal period of the position encoder [kHz] Example: $F_{max} = 20\ \mu m \cdot 400\ kHz \cdot 6 \cdot 10^{-3} \left[\frac{m \cdot s}{\mu m \cdot min}\right]$ F_{max} = 480 m/min

Machine interfacing		iTNC 530
Regulation with CC 424(B)		
	Position control resolution	$\frac{\text{Signal period}}{1024}$
	Path interpolation	3ms
	Fine interpolation	200 µs/100 µs[a]
	Cycle time of current controller	PWM frequency — Cycle time if MP2182.x = 0: 3333 Hz — 150 µs 4000 Hz — 120 µs 5000 Hz — 100 µs 6666 Hz — 75 µs (MP7610 = Double-speed) 8000 Hz — 60 µs (MP7610 = Double-speed) 10000 Hz — 50 µs (MP7610 = Double-speed)
	Cycle time of speed controller	■ Speed controller cycle time = 2 · current controller cycle time ■ Unless PWM frequency ≤ 5kHz and double-speed performance, then: Speed controller cycle time = Current controller cycle time
	Cycle time of position controller	■ Single-speed: Position controller cycle time = Speed controller cycle time ■ Double-speed without position encoder (single-encoder systems): Position controller cycle time = Speed controller cycle time ■ Double-speed with position encoder (dual-encoder systems): Position controller cycle time = 2 · speed controller cycle time
	Maximum motor speed	$n_{max} = \frac{f_{PWM} \cdot 60000\ \text{min}^{-1}}{p \cdot 5000\ \text{Hz}}$ n_{max}: Maximum motor speed [min^{-1}] f_{PWM}: PWM frequency [Hz] p: Number of pole pairs The following PWM frequencies are available: 3333 Hz, 4000 Hz, 5000 Hz, 6666 Hz, 8000 Hz, 10000 Hz
	or by way of:	Maximum signal frequency of the motor encoder = 400 kHz $n_{max} = \frac{f_{max} \cdot 60000\ [\text{s/min}]}{\text{ELC}}$ n_{max}: Maximum motor speed [min^{-1}] f_{max}: Maximum signal frequency of the motor encoder [kHz] ELC: Encoder line count Example: $n_{max} = \frac{400\ \text{kHz} \cdot 60000\ [\text{s/min}]}{2048}$ $n_{max} = 11718.75\ \text{min}^{-1}$

Machine interfacing	iTNC 530
Maximum feed rate by way of:	Maximum signal frequency of the position encoder: ■ At 1 V_{PP}: 27 kHz or 400 kHz ■ At 11 μA_{PP}: 27 kHz or 140 kHz Calculation of the maximum feed rate F_{max}: $F_{max} = SP \cdot f_{max} \cdot 6 \cdot 10^{-3} \left[\frac{m \cdot s}{\mu m \cdot min}\right]$ F_{max}: Maximum feed rate [m/min] f_{max}: Maximum signal frequency of the position encoder [kHz] SP: Signal period of the position encoder [kHz] Example: $F_{max} = 20\ \mu m \cdot 400\ kHz \cdot 6 \cdot 10^{-3} \left[\frac{m \cdot s}{\mu m \cdot min}\right]$ $F_{max} = 480$ m/min

a. Single-speed/double-speed

Machine interfacing	iTNC 530
Error compensation	
	■ Linear and nonlinear axis error ■ Backlash ■ Reversal peaks during circular movements ■ Hysteresis ■ Thermal expansion ■ Stick-slip friction ■ Sliding friction
Monitoring functions	
	■ Amplitude of encoder signals ■ Edge separation of encoder signals ■ Absolute position for encoders with distance-coded reference marks ■ Following error ■ Movement monitoring ■ Standstill monitoring ■ Nominal speed value ■ Checksum of safety-related functions ■ Power supply ■ Buffer battery ■ Operating temperature ■ Run time of PLC program ■ Motor current ■ Motor temperature ■ Temperature of power stage ■ DC-link voltage
Integral PLC	
PLC memory	2 GB on hard disk
Program format	Statement list
RAM process memory	512 KB
PLC cycle time	10.8 ms (can be set)
PLC inputs, 24 V–	56 (additional inputs as option)
PLC outputs, 24 V–	31 (additional outputs as option)
Analog inputs, ±10 V	3 (additional analog inputs as option)
Analog outputs, ±10 V	13
Inputs for thermistors	3 (additional inputs as option)

Machine interfacing	iTNC 530
Commissioning aids	
	■ Oscilloscope ■ Trace function ■ Table function ■ Logic diagram ■ Log ■ TNCopt PC software
Interfaces	
	■ One RS-232-C/V.24 and one RS-422/V.11, each with max. 115 Kbps ■ Expanded interface with LSV-2 protocol for external operation of the iTNC over the interface with HEIDENHAIN software TNCremo. ■ Fast Ethernet interface 100BaseT
Permissible temperature range	Operation: 0 °C to +40 °C Storage: –35 °C to +65 °C

User functions

User functions	iTNC 530
Program input	HEIDENHAIN conversational and ISO formats
Fixed cycles	■ Drilling/boring cycles for drilling, peck drilling, reaming, boring, counterboring, tapping with or without floating tap holder ■ Cycles for milling internal and external threads ■ Roughing and finishing rectangular and circular pockets ■ Cycles for clearing level and inclined surfaces ■ Cycles for milling linear and circular slots ■ Hole patterns on circle and line ■ Contour pockets—also with contour-parallel machining ■ Contour train ■ OEM cycles (special cycles developed by the machine tool builder) can also be integrated
Touch probe cycles	■ Touch probe calibration ■ Compensation of workpiece misalignment, manual or automatic ■ Datum setting, manual or automatic ■ Automatic workpiece measurement ■ Cycles for automatic tool measurement
Contour elements	■ Straight line ■ Chamfer ■ Circular path ■ Circle center point ■ Circle radius ■ Tangentially connected arc ■ Corner rounding
Approaching and departing the contour	■ Via straight line: tangential or perpendicular ■ Via circular arc

User functions	iTNC 530
FK free contour programming	FK free contour programming in HEIDENHAIN conversational format with graphic support for workpiece drawings not dimensioned for NC
Parallel operation	Creating a program with graphical support while another program is being run
3-D machining	■ Feed rate reduction during plunging (M103) ■ Jerk-free path control ■ HSC filter and advanced HSC filter ■ 3-D tool compensation through surface normal vectors ■ Automatic compensation of machine geometry when working with tilted axes (M114, M115, M128, M129, M130) ■ Changing the angle of the tilting head with the electronic handwheel during program run. The position of the tool tip does not change. ■ Keeping the tool normal to the contour ■ Tool radius compensation perpendicular to traversing and tool direction ■ Spline interpolation

User functions	iTNC 530
Rotary table machining	■ Programming a contour on a cylindrical surface as if on a plane ■ Feed rate in mm/min (M116)
Q parameters — programming with variables	■ Mathematical functions =, +, –, *, /, sin α, cos α, angle α from sin α and cos α, $\sqrt{a}$, $\sqrt{a^2 + b^2}$ ■ Logical comparisons (=, =/, <, >) ■ Calculating with parentheses ■ tan α, arc sine, arc cosine, arc tangent, a^n, e^n, ln, log, absolute value of a number, the constant π, negation, truncation of digits before or after the decimal point ■ Functions for calculation of circles
Programming aids	■ Calculator ■ Context-sensitive help function for error messages ■ Graphical support for programming cycles ■ Comment blocks in the NC program
Position data	■ Nominal positions for straight lines and circles in Cartesian or polar coordinates ■ Incremental or absolute dimensions ■ Display and entry in mm or inches ■ Display of the handwheel path during machining with handwheel superimpositioning
Tool compensation	■ Tool radius in the working plane and tool length ■ Radius-compensated contour look-ahead for up to 99 blocks (M120) ■ Three-dimensional tool-radius compensation for changing tool data without having to recalculate an existing program
Tool tables	Multiple tool tables with any number of tools

User functions	iTNC 530
Cutting-data tables	For automatic calculation of spindle speed and feed rate from tool-specific data (cutting speed, feed rate per tooth)
Constant contouring speed	■ Relative to the path of the tool center ■ Relative to the tool cutting edge (M109, M110, M111)
Program jumps	■ Subroutines ■ Program-section repeat ■ Calling any program as subroutine
Coordinate transformation	■ Datum shift, rotation, mirroring ■ Scaling factor (axis-specific) ■ Tilting the working plane
Actual position capture	Actual positions can be transferred directly into the NC program
Program verification graphics	Graphic simulation before a program run, even while another program is running ■ Plan view, view in three planes, 3-D view ■ Magnification of details

User functions	iTNC 530
Programming graphics	In the **Programming and Editing** operating mode, the contours of the NC blocks are drawn (2-D pencil-trace graphics), also while another program is being run.
Program Run graphics	Graphical simulation of executed program in plan view, three planes and 3-D view
Machining time	■ Calculation of machining time in the **Test Run** operating mode ■ Display of the current machining time in the Program Run operating modes
Returning to the contour	■ Mid-program startup in any block in the program, returning the tool to the calculated nominal position to continue machining ■ Program interruption, contour departure and return
Datum tables	Multiple datum tables
Pallet tables	Tool-oriented or workpiece-oriented execution of pallet tables with any number of entries for selection of pallets, part programs and datums

Accessories

Accessories	iTNC 530
Electronic handwheels	■ One portable HR 410 handwheel, or ■ One portable HR 420 handwheel with display, or ■ One panel-mounted HR 130 handwheel, or ■ Up to three panel-mounted HR 150 handwheels via the HRA 110 handwheel adapter
Touch probes	■ TS 220 triggering 3-D touch probe with cable connection, or ■ TS 440, TS 640 triggering 3-D touch probe with infrared transmission ■ TT 130 triggering 3-D touch probe for tool measurement
Data transfer software	TNCremoNT, TNCremo
PLC software developing environment	PLCdesignNT
Software for creating cycle structures	CycleDesign
Software for remote diagnosis	TeleService
Software for putting digital control loops into operation	TNCopt
Software for diagnosis of digital drive systems	TNCdiag
PLC input/output unit	Up to four PL 410B/PL 510 or one PL 405B
	PL 410B version 1: Additional 64 PLC inputs and 31 PLC outputs per PL PL 410B version 2: Additional 64 PLC inputs and 31 PLC outputs as well as 4 analog inputs ±10 V and 4 inputs for thermistors per PL PL 405B: Additional 32 PLC inputs and 15 PLC outputs per PL PL 510: Four slots for PLD 16-8 (16 PLC inputs and 8 PLC outputs) and PLA 4-4 (four ±10 V analog inputs and 4 inputs for thermistors)

Accessories	iTNC 530
IPC 6110/6120	See the separate Technical Manual for a detailed description. ■ Processor: Intel Celeron, 400 MHz ■ 512 MB SDRAM ■ Bus frequency: 100 MHz ■ 15-inch screen, resolution: 1024 x 768 pixels ■ 3 x USB connections (1x front, 2x rear - USB 1.1) ■ Slot for CompactFlash memory card, types I and II ■ Ethernet connection, 100 Mbps ■ V.24/RS-232-C data interface ■ Power supply: 85 to 264 VAC ■ Frequency: 47 to 440 Hz ■ Power consumption: 60 W ■ Cooling system: fan ■ Degree of protection for computer and TE 6xx: Front panel IP 54, rear panel IP 20 ■ Weight: 7.7 kg

Software options

Software options	iTNC 530
#8: Software option 1 ID 367 591-01 (standard for the MC 422B,C)	■ Cylindrical surface interpolation ■ Feed rate in mm/min ■ Tilting the working plane ■ Circular interpolation in 3 axes with tilted working plane
#9: Software option 2 ID 367 590-01 (standard for the MC 422B,C)	■ **HSC path control**: • Special position value filters for optimal path control **(`HSC filter`** and **`advanced HSC filter`)** • **Cycle 32:** Additional options in Cycle 32 for roughing and finishing mode, and tolerances for rotary axes ■ **3-D compensation** • of the tool length and diameter via surface normal vectors (**`DL/DR`** in the tool table) • via radius compensation perpendicular to the tool direction (**`RL/RR`** in the NC program) • Programming with **`LN`** blocks (machine-neutral 3-D programming with various options) ■ **3-D machining** • **TCPM:** Tool Center Point Management – Maintain position of the tool tip when positioning with tilting axes (via **`M128`** in the NC program) and with additional options (via **`Function TCPM`** in the NC program) • **M114:** Automatic correction of machine geometry when positioning tilting axes • **M144**: Compensating the machine's kinematic configuration for ACTUAL/NOMINAL positions at end of block ■ Keeping the tool normal to the contour ■ Straight-line interpolation in 5 axes (export license required) ■ Spline interpolation: Execution of splines (third-degree polynomials) ■ 0.5-ms block processing time
#18: HEIDENHAIN DNC ID 526 451-01	■ All functions of HEIDENHAIN DNC are available (see "Remo Tools SDK" documentation) ■ The following functions of the LSV2 ActiveX control become available (all other options do not require option #18): • OpenChatWindow • RunProgram • SetOverride • SetPreset • TransmitChatText • TransmitPLCCommand • TransmitPLCString • HostFunction • EventReceived • NCMsgReceived • PLCMsgReceived

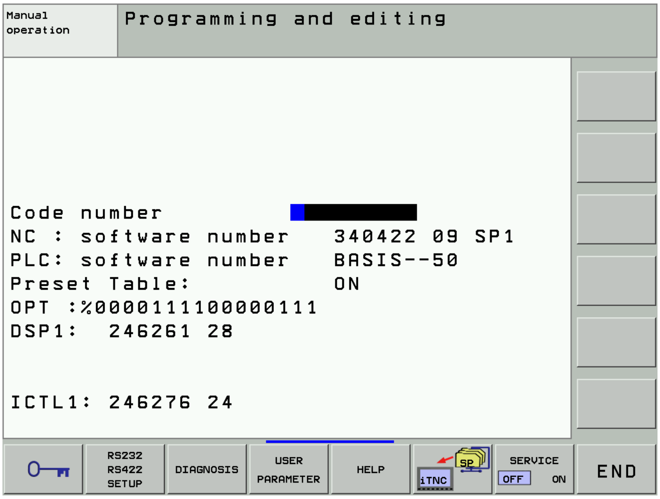
Software options	iTNC 530
#40: DCM – Dynamic Collision Monitoring ID 526 452-01	■ Dynamic collision monitoring of defined machine components
#41: Additional Language: ID 530 184-01	■ Additional conversational languages: • Slovenian • Norwegian • Slovak • Latvian • Korean • Estonian • Turkish • Romanian • Lithuanian
#42: DXF Converter ID 526 450-01	■ Conversion of DXF files (also just parts of files) for editing in HEIDENHAIN conversational programming or with smarT.NC (see iTNC 530 User's Manual)
#44: Global PGM Settings ID 576 057-01	■ Possibility of superimposing various coordinate transformations and settings in the program-run modes of operation (see iTNC 530 User's Manual)
#45: AFC - Adaptive Feed Control ID 579 648-01	■ In adaptive feed control, after a teach-in cut the TNC automatically controls the feed rate as a function of the respective spindle power consumption.
#46: Python OEM Process ID579 650-01	■ Possibility of running a Python process on the control
#48: KinematicsOpt ID630 916-01	■ Performing an initial measurement and optimizing the active kinematics
#52: KinematicsComp ID661 879-01	■ Three-dimensional compensation of geometry errors, thermally induced errors and dynamic errors
#53: Upgrade functions (Feature Content Level) ID 529 969-01	■ Enabling of expanded, useful functions in updates of NC software. However, these are available without option #53 for the initial installation (see iTNC 530 User's Manual)
#92: 3D-ToolComp ID 679 678-01	■ 3-D tool radius compensation, which uses a compensation-value table to define angle-dependent delta values that describe the tool deviation from the ideal circular shape.
#93: Extended Tool Management ID 676 938-01	■ The enhanced tool management function additionally includes the tooling list table and the T usage order table.
#101 to #130: OEM option 1 to OEM option 30 ID 579 651-01 to 579 651-30	■ Possibility for the OEM to enable his applications for the end user via the SIK from HEIDENHAIN

2.4 Software

2.4.1 Designation of the software

The iTNC 530 features a separate software for the NC and the PLC. The NC software is identified with an eight-digit number.

If you press the MOD key in any operating mode, you can display the ID numbers of the NC software, the DSP software (DSP1, DSP2) and the current-controller software (ICTL1, ICTL2). An installed service pack is shown by **SPx** after the ID number of the NC software.



Model

The iTNC 530 is shipped with the following NC software:

Standard	Export	Comment
340 420-xx	340 421-xx	iTNC 530
340 422-xx	340 423-xx	iTNC 530
340 500-xx	340 501-xx	Service pack for 340 480-xx or 340 481-xx
340 480-xx	340 481-xx	iTNC 530 with Windows 2000
340 502-xx	340 503-xx	Service pack for 340 480-xx or 340 481-xx
340 490-xx	340 491-xx	iTNC 530
340 492-xx	340 493-xx	iTNC 530 with Windows XP

Due to restrictions on the export of the iTNC 530, HEIDENHAIN can also supply a special export version. This export version differs from the standard control though the installed NC software type. HEIDENHAIN releases a new NC software type whenever it introduces extensive new functions.

2.4.2 PLC software

The PLC software is stored on the hard disk of the iTNC. You can order a PLC basic program directly from HEIDENHAIN. With the PLC development software **PLCdesignNT,** the PLC program can very easily be adapted to the requirements of the machine.

2.4.3 Additional Control Loops or Software Options

For each MC 42x(B,C), only the minimum number of control loops is enabled. If you need additional control loops, you must enable them by entering a code number. These additional control loops are not bound to a certain machine parameter index. The definition as to whether a control loop is used is entered as a value ≠ 0 in MP120.x (nominal speed value outputs to the axes) and MP121.x (nominal speed value outputs to the spindles).

Each MC 42x(B,C) can clearly be identified by the SIK (System Identification Key). You will find the SIK number on the outside of the MC 42x(B,C) housing (below the ID label) and on the SIK board. To install the SIK, See "Handling of the HDR Hard Disk and the SIK" on page 335.

If you wish to enable additional control loops or software options, please contact HEIDENHAIN for the code number. HEIDENHAIN can give you the code number after you state your SIK number.

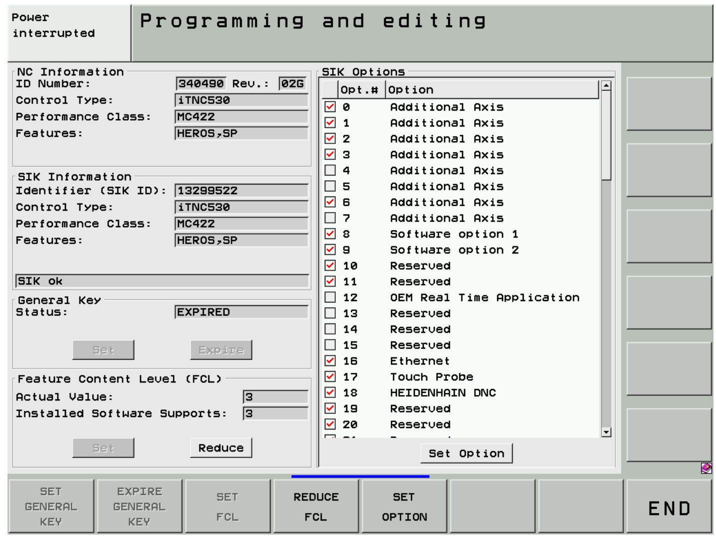
Note

If you replace the MC 42x(B,C), you must also replace the SIK in order to ensure that the enabled control loops or software options will also be enabled on the new MC 42x(B,C).

In order to enable additional control loops or software options, proceed as follows:

- While in the `Programming and Editing` operating mode, press the MOD key.
- Enter the code number SIK and confirm your entry with the ENT key.

The following display will appear:



The display gives you the following information and possibilities for settings:

NC Information:

Display	Meaning
ID Number	Software
Rev.	Software version
Control Type	Control model
Performance Class	Type of main computer
Features	Characteristics of the control

SIK Information:

Display	Meaning
Identifier (SIK ID)	SIK number
Control Type	Type of control
Performance Class	Type of main computer
Features	Characteristics of the SIK
SIK ok, **wrong SIK (Control Type mismatch),** **wrong SIK (Features mismatch),** **wrong SIK (Performance Class mismatch),** **no SIK (Programming Station)** or **no SIK**	Status of the SIK

General Key:

The General Key permits you to enable and test all new Feature Content Level functions and software options. The General Key is valid for 90 days after the first enabling. After these 90 days have expired, the General Key can only be used again after the software version on your control has been updated.

Display	Meaning	
Status	**NONE**	General Key was not used yet for this software version.
	dd.mm.yyyy	Date up to which all options will be available. It is not possible to enable them again after this date.
	EXPIRED	General Key has expired for this software version. Enabling is not possible.
Set	Pressing the **Set** button or the **SET GENERAL KEY** soft key opens a window in which you press the **Apply** button or the **SET GENERAL KEY** button again to enable all options for **90 days.** If this is done successfully, the **General Key has been set** message appears and the expiration data of the General Key is shown in the **Status** field.	
Expire	Pressing the **Expire** button or the **EXPIRE GENERAL KEY** soft key opens a window in which you press the **Apply** button or the **EXPIRE GENERAL KEY** button again to expire the General Key immediately. Be aware that it is then no longer possible to enable the General Key for this software version!	

Feature Content Level (FCL):

> **Danger**
>
> If you reduce the Feature Content Level, then all new FCL functions and expanded features will be set to the desired lower version. Only error fixes remain active. This action can only be reversed by entering a valid code number again.

Display	Meaning
Actual Value	Current version of the Feature Content Level
Installed Software Supports	Highest possible FCL with the current software
Set	Pressing the **Set** button or the **SET FCL** soft key opens a window in which you can enter the code number for the desired Feature Content Level under **Enter Key Code.** HEIDENHAIN can give you the code number after having been informed of the SIK number. Then press the **Set FCL** button or the **SET FCL** soft key to confirm the entry. If this is done successfully, the message **Feature Content Level has been set** appears.
Reduce	Pressing the **Reduce** button or the **REDUCE FCL** soft key opens a window in which you can enter the desired **lower** Feature Content Level under **New (lower) FCL.** Please note that this action can only be reversed by entering a valid code number again. If you really want to reduce the Feature Content Level, then press the **Apply** button or the **REDUCE FCL** soft key to confirm the entry. If this is done successfully, the message **Feature Content Level has been reduced** appears.

SIK Options:
All available software options and their corresponding numbers are listed in a table. The check marks in the first column indicate which software options have been enabled on your control.

To enable additional software options, proceed as follows:

- Press the MOD key and enter the code number **SIK** to display the input mask for SIK functions.
- In the table on the right side of the screen, use the arrow keys or the mouse to select the software option you want to enable.
- Pressing the **Set Option** button or the **SET OPTION** soft key opens a window in which you can enter the code number for the desired software option under **Enter Key Code.** HEIDENHAIN can give you the code number after you state your SIK number.
- Enter the code number and confirm the entry by pressing the **Apply** button or the **SET OPTION** soft key.
- If this is done successfully, the message **Option <number> has been set** appears, and the option is checked in the table.

When you leave this table of SIK functions by pressing the **END** soft key or the END key, you are requested to reboot the control if you have made any changes. Press the emergency stop button and press the **Reboot Now!** button or the **REBOOT NOW** soft key to reboot the control and activate the changes.

To display the corresponding number of machine-parameter indexes:

- In OEM.SYS, enter the code word **PWMPARAMETER =** followed by the required number of machine-parameter indexes MP2xxx.y for the current and speed controller.
- In OEM.SYS, enter the code word **AXISNUMBER =** followed by the required number of remaining machine-parameter indexes.

Status of options that have been set

Module 9067 Status of software settings

PLC Module 9067 enables you to request status information about software settings. Module 9067 can currently be used to interrogate the software options set in the SIK.

Call:

PS	B/W/D/K	<Mode> 0: Interrogate whether SW option is set in the SIK
PS	B/W/D/K	<Number> If mode is 0: Number of SIK option
CM	9067	
PL	B/W/D	<Status> Status of SIK option (if mode is 0): 0: Not set 1: Set

Error recognition:

Marker	Value	Meaning
M4203	0	Function was performed correctly
	1	Error code in W1022
W1022	1	Invalid value for number
	2	Invalid value for mode

Temporary enabling of an option

You have the possibility of enabling software options with a temporary code number for a limited period of time. You can define a time between 10 to 90 days for enabling the software options. However, each option can only be enabled once with a temporary code number.

If you want to enable a software option temporarily on the control via the temporary key, proceed in the same way as for the standard enabling of software options. Press the **Set Option** button or the SET OPTION soft key. This opens a window in which you can enter the code number for the desired software option in **Enter Key Code.**

If the software option was enabled successfully, the expiration date of the temporary enabling is shown in the **Expires** column under **SIK Options.** After the defined period has expired, the entry in the **Expires** column will change to **EXP**, meaning "expired." The software option is then no longer available.

A software option can be enabled for an unlimited period at any time by means of the code number, which you will receive from HEIDENHAIN after stating the SIK number.

HEIDENHAIN would like to point out that it is not possible to use the OEM-specific options with the SIKs of the first generation. If you encounter any problems in this respect, please contact your HEIDENHAIN service agency.

You can generate the temporary code number with the TNCOEMOption tool for PCs. The tool is available for cost-free download from our HEIDENHAIN FileBase.

When you generate the code number, you must specify the number of the respective software option, the number of days you want to enable the option, and an optional OEM Key. The OEM Key provides protection against unauthorized persons generating code numbers for your machines with the help of the PC tool in order to enable software options without your approval. However, it is not essential to specify an OEM Key.

The OEM Key for generating the code number must be identical to the OEM Key on your control.

The OEM Key was added to the display and management of the SIK options, Feature Content Level and General Key on the control. Press the MOD key and enter the code number **SIK** to display the input form for SIK functions:

- **OEM Key**
  The OEM Key on the control can only be used for enabling software options with a temporary code number if the same OEM Key was specified for the generation of the temporary code number.
  If the OEM Key on the control is not identical to the one used for generating the code number, the software option will not be enabled. Once the OEM Key has been set, it cannot be reset.

Display	Meaning	
OEM Key for temp. options	**NONE**	OEM Key was not set
	SET	OEM Key was set
Set OEM Key	Press the **Set OEM Key** button or the **SET OEM KEY TEMP. OPT.** soft key to open a window in which you can enter the OEM Key. Keep in mind that this process cannot be undone. The OEM Key may consist only of numbers. Confirm your entry by pressing the **Apply** button, or by pressing the **SET OEM KEY TEMP. OPT.** soft key again. If the OEM Key was set successfully, the message **OEM Key has been set** appears, and the status in **OEM Key for temp. options** changes to **SET.**	

Displaying the status of an option

The status of an option (enabled or not) can be displayed either as a menu by entering the code number SIK (see previous pages), or bit-encoded after pressing the MOD key in the **OPT** line.

Each bit has the following meaning:

OPT: %xxxxxxxxxxxxxxxx

Bit 0: 1st additional control loop
to
Bit 7: 8th additional control loop
Bit 8: Software option 1
Bit 9: Software option 2
Bit 10: Reserved
to
Bit 15: Reserved

Options for the OEM

As of software 340 49x-04, the SIK options #101 to #130 are available as OEM-specific options. You can assign your own OEM-specific software options to these options.
This makes it possible for you to enable your own applications (e.g. cycles) for the end user via the SIK from HEIDENHAIN.

The options can be enabled by entering a code number. HEIDENHAIN can give you the code number after having been informed of the SIK number. You also have the possibility of creating a temporary key for these OEM-specific options in order to enable the options for a period of max. 90 days.

HEIDENHAIN would like to point out that it is not possible to use the OEM-specific options with the SIKs of the first generation. If you encounter any problems in this respect, please contact your HEIDENHAIN service agency.

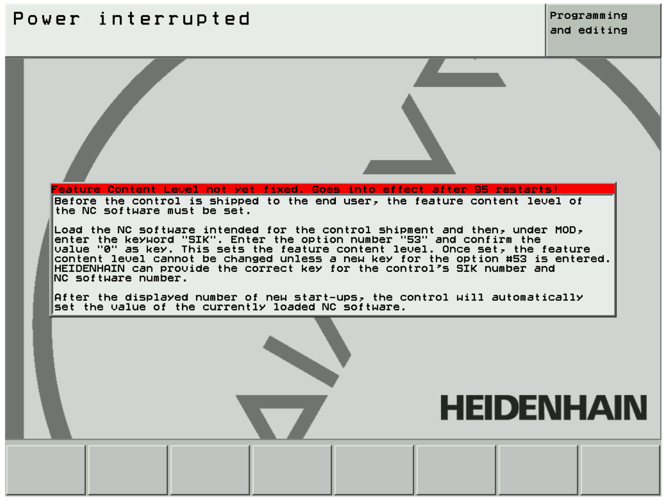
The new PLC module 9067 enables you to request the status of the SIK options. Use this module to request information from the SIK about whether certain software options are enabled via the SIK. Depending on the response, you can activate the software option via the PLC program.

2.4.4 Upgrade Functions (Feature Content Level)

In the past, each new NC software version contained error fixes as well as expanded functions. Users who wanted only the NC software update to eliminate the errors often felt bothered by the expanded functions. For this reason, error fixes and expanded functions are now handled separately within the software.

If a new NC software is later loaded as an update onto a machine with NC software 340 490-01, then as the default setting only the error fixes contained will be effective. The **upgrade functions** will at first remain inactive. The upgrade functions can then be enabled by entering a code number. HEIDENHAIN can give you the code number after having been informed of the SIK number and NC software version.

- The **upgrade functions** are defined as "feature content level" (FCL) in the SIK under option #53.
- The first time an NC software with upgrade functionality is installed on a control (i.e. no FCL has been set in the SIK), then the entire scope of functions can be used (including the **upgrade functions**).
  The FCL is then automatically set after 255 restarts, or by entry of the code number 0 under **option #53** (See "Additional Control Loops or Software Options" on page 302), and all upgrade functions belonging to this software version are enabled as well.
  A note appears asking to confirm the installed NC software as the initial version, or if another initial software version is to be installed.
- If the FCL has already been set in the SIK of a control, then after an update (e.g. from software 340 490-02 to -03), the new **upgrade functions** of the newer software version can only be used after entry of a code number from HEIDENHAIN under **option #53** (See "Additional Control Loops or Software Options" on page 302).
- After pressing the MOD key, the current status of the FCL is displayed in addition to the software versions.
- The FCL is incremented with each new version of the NC software.
- If the **upgrade functions** are enabled via the FCL for a software version, then all **upgrade functions** of this software version and all its predecessors are available. For example, if the FCL for version 340 490-03 is set, then all **upgrade functions** from version 340 490-02 are also available immediately.
- If a newer software version, e.g. 340 490-06, is simply loaded onto a control, then the already existing **upgrade functions** remain available, but the **upgrade functions** of the newer version cannot be used. They must be enabled by entering a new code number.



Availability of feature content levels:

Software version	Available FCL	Notes on the FCL
340 49x-01	FCL function not supported	
340 49x-02	FCL 02	
340 49x-03	FCL 03	The TNCguide FCL function and Asian languages require at least 256 MB RAM
340 49x-04	FCL 04	
340 49x-05	FCL 04	No new FCL functions introduced

2.4.5 NC software exchange on the iTNC 530

Note

- The NC software must be exchanged only by trained personnel.
- For exchanging the NC software, HEIDENHAIN provides packed files (*.zip) with the NC software. The packed files are transferred to the hard disk of the control and unpacked. The packed files remain on the hard disk of the control.

General information

- The following files are necessary for an update, and must be located in the same directory:
  - **setup.zip**
  - **setup.omf** (setup.exe through 340 49x-02)
  - **setup.elf** (setup.exe as of 340 49x-03, as an alternative to **setup.omf**)
  - **setup.ini** (this is to be created by the OEM, and is only necessary if the update is to occur according to a specific sequence)

  Please note that this currently means that the version of the software to be installed is not displayed in advance.

Note

You can download the two files required for the update (**setup.zip**, **setup.elf**) from the HEIDENHAIN Filebase and unzip it in your update directory.

- Software updates and service packs are loaded in the same manner.
- Automated updating is possible. If your directories are structured appropriately and you use a **setup.ini** control file, this method makes it possible to update the control automatically when it is booted (See "Automated update (setup.ini)" on page 314).
- Copying of the update files (setup.elf/exe and setup.zip) to the system partition occurs automatically as part of the update program in **directories** with the following naming convention:
  - iTNC without Windows
    - Software update: **SYS:\zip\<ID no.>_<ver>**
    - Service pack: **SYS:\zip\<ID no.>_<ver>_SP<n>**
  - iTNC with Windows
    - Software update: **C:\Program Files\install\<ID no.>_<ver>**
    - Service pack: **C:\Program Files\install**<ID no.>_<ver>_SP<n>

Attention

No manual changes may be made to these directories, since they might be required for restoring earlier software versions.

- Before the update, a query appears asking whether the necessary binary to ASCII conversion should be performed. HEIDENHAIN recommends always performing this conversion. Only files whose binary version has **changed** are converted. As part of the recommended procedure, immediately thereafter a query appears if there is not enough memory space:
  - Cancel if not enough memory
  - Delete largest or oldest files first
  After successful installation, these ASCII files are automatically reconverted to binary format.
  This procedure can be automated in the control file mentioned above.
- The NC software has been prepared in such a manner that when an update is performed or a service pack loaded as of software 340 490-02, the PLC program and PLC partition can be updated as well, according to the requirements of the OEM. When the NC software is updated, the OEM uses the HEIDENHAIN PC software PLCdesignNT to add all necessary files to the **setup.zip** archive. These files are copied to the appropriate locations during an update.

Note

The support necessary for this in PLCdesignNT has been available since version 2.3. It is not possible to update only the PLC data but not the NC software.

Automated update (setup.ini)

Automated updating is possible with the iTNC 530. In order to automate an update as much as possible, a control file with the name **setup.ini** is necessary in addition to the **setup.zip** and **setup.elf** or **setup.omf** (.exe with Windows control) files. If your directories are structured appropriately and you use this **setup.ini** control file, this method makes it possible to update the control automatically when it is booted. If there is a certain "install" directory on your iTNC containing a **setup.ini** control file in addition to the update files, an update is performed automatically according to the instructions in this control file.
The following directories are checked during booting for the presence of a control file:

- iTNC without Windows
  - **TNC:\install**
  or, if a USB memory device is connected
  - **USB0:\install** (**USB0:** first partition of the first USB memory device)
- iTNC with Windows
  - **D:\install**
  or, if a USB memory device is connected
  - **G:\install** (**G:** corresponds to the drive letter of the USB memory device – network drives are not permitted!)

As of software version 340 49x-03 you can save two different update procedures for automatic software updating on one USB stick. This makes it possible to use just one USB stick for updating iTNCs with and without a Windows operating system. In addition to the **install** directory, the following directories are also checked for the presence of the setup files when the control is booted:

- iTNC without Windows
  - **USB0:\installsingle** (**USB0:** first partition of the first USB memory device)
- iTNC with Windows
  - **G:\installdual** (**G:** corresponds to the drive letter of the USB memory device)

The **setup.ini** file can be created with a simple text editor.

An example of a setup.ini file:

```
Interactive=1
Confirm=1
Language=GERMAN
SavePlc=TNC:\backup\340490_002.zip
```

The following settings are possible:

Parameter	Description
Interactive=[0,1]	Deletion of the NC software archive (old software versions) and binary to ASCII conversion **must be confirmed by the user.** **0:** No **1:** Yes [default]
Confirm=[0,1]	**Start of update and reboot process must be confirmed by the user** **0:** No **1:** Yes [default]
ServiceRequest=[0,1]	A service request is triggered after an update **(only if remote diagnosis is active)** **0:** No [default] **1:** Yes
ConvertToAscii=[0,1] (only if Interactive=0)	Binary to ASCII conversion with **automated** update process (not with service pack) **0:** No **1:** Yes [default]
CopyToSys=[0,1]	The setup archive to be installed is copied to the directory SYS:\zip (iTNC with Windows: C:\Program Files\install). This makes it possible to return to this software version later. **0:** No **1:** Yes [default]
DeleteFiles= [DATE,SIZE,CANCEL] (only if Interactive=0)	Procedure during binary to ASCII conversion during automated update if there is not enough memory available on the TNC or PLC partition **DATE:** Delete oldest files first **SIZE:** Delete largest files first **CANCEL:** Cancel the update [default]

Parameter	Description
DeleteArchives=[DATE,SIZE,CANCEL] (only if Interactive=0)	Procedure is there is not enough room on the SYS partition for the update **DATE:** Delete oldest archive first **SIZE:** Delete largest archive first **CANCEL:** Cancel the update [default]
Language=[ENGLISH, GERMAN] (only if Confirm=0)	Language for the dialog guidance during the update if no user activities are required for starting the update and rebooting (Confirm=0) **GERMAN:** German dialog text **ENGLISH:** English dialog text [default]
DelSource=[0,1]	Delete the source files (setup.zip, setup.ini, setup.elf/exe) once the update has completed successfully **0:** No [default] **1:** Yes
DeleteIni=[0,1]	Delete the setup.ini file after a successful update. **0:** No [default] **1:** Yes
SavePlc=<name>	If the software is updated from 340 49x-02 or higher to a newer version, then the entire PLC partition can be stored as a ZIP file in binary format. This makes it possible to restore this software state including the PLC files. Here you enter the path and file name for the ZIP file in which the entire PLC partition is saved in binary format. Please consider the software version when assigning the file name. After an update, the **-SETUP** code number is entered on a single-processor control, and the **Setup Back** button is pressed in the iTNC Control Panel on a dual-processor control in order to reverse an update. The previous software version is then reinstalled. The PLC partition is restored from the *.zip file that was entered under **SavePlc=** in the setup.ini file. Please note that a subdirectory in which you want to save the *.zip file must be created on the TNC partition before the update, otherwise the update is canceled when backing up the old PLC partition. Only updates from one full software version to another can be reversed. You cannot reverse the installation of service packs.
RestorePlc=<name>	The Restore function is used to restore the PLC data to a certain software state. Here you enter the path and file name for the ZIP file containing the PLC data state (in binary format) appropriate to the NC software version to be installed. This should be the ZIP file which was saved with the SavePlc function, containing the corresponding PLC state and saved on the PLC partition.

Automated updating of machine parameters

If a PLC:_mpupdate directory is created, then files that automatically update or expand the active and selected MP or OEM.SYS files when the control is started can be saved here.
Files containing the name `merge.* (merge.mp` and `merge.oem.sys)` expand the MP and OEM.SYS files by the entries contained in them. Files containing the name `overwrite.* (overwrite.mp` and `overwrite.oem.sys)` contain updated entries for the corresponding files, and overwrite entries with the same names in the MP and OEM.SYS files.
This means that MP subfiles cannot be considered.

Information about the OEM cycles

Keep the following information in mind if you switch from software 340 422-xx to 340 490-xx or when you use OEM cycles in the HEIDENHAIN cycle tree instead of in an OEM cycle tree:

- Download and install the current CycleDesign software version from the HEIDENHAIN FileBase
- Open the existing CycleDesign project with the current CycleDesign version and adapt it if necessary.
- Menu item **File** > **Change variant...**
- In the **New variant** pull-down menu, select the entry with the newest version of the NC software and confirm with **OK**.
- Confirm also the subsequent Update Information with **OK**.
- Menu item **File** > **Save**
- Then transfer the file (*.cdf) to the control, start the software update and check its functions.

Attention

Without this procedure, after a software update to the version 340 49x the error message "Key nonfunctional" will appear when you press the CYCL DEF key.
The iTNC also does not recognize these cycles, and so ERROR blocks are inserted in the NC program. These ERROR blocks must be deleted manually!

Your *.cdf file and the appropriate *.cdc for the HEIDENHAIN cycles are still in the folder PLC:\JH\ on the control after the NC software exchange. You can find more information in the User's Manual or in the online help for CycleDesign.

Manual NC software exchange as of software 340 490-02

Manual update

The following procedure is used to perform a manual update (without a **setup.ini** file in the setup directory) or an installation of a service pack:

- HEIDENHAIN recommends using the PC program TNCremoNT from HEIDENHAIN to make a backup for the control.
- Please check whether cycle projects (HEIDENHAIN or OEM cycles) must be adapted and transferred before the update! See "Information about the OEM cycles"on page 317.
- Connect the appropriate network drive or USB memory medium containing the files necessary for the update **(setup.zip**, **setup.elf)**. If the network drive is not displayed, please proceed as follows:
  - Enter the code number 95148 and confirm your entry with the ENT key.
  - While in the **Machine-parameter programming** mode, press the PGM MGT key.
  - Press the **NETWORK** soft key.
  - Select the PC to be connected with the cursor keys and press the **CONNECT DRIVE** soft key.
  - To exit the list of network drives, press the **END** soft key.
  - To exit the program manager, press the **END** soft key.
  - Press the END key to exit the **Machine-parameter programming** mode.
- In the **Programming and Editing** mode, press the **MOD** key
- Enter the code number **SETUP** and confirm your entry with the **ENT** key.
- In the upper part of the window, select the folder containing the setup files for the new NC software, and confirm with the **ENT** key. The file to be called for the installation **(setup.elf)** is displayed in the lower part of the window.
- To switch to the lower part of the window, press the **FILES** soft key. (Press the **PATH** soft key to return to the top part of the window.) In the lower part of the window, select the *.elf file of the new NC software with the cursor keys and press **ENT.** Following that, the NC software exchange will start.
- Use the cursor keys to select the language desired for the update guidance, and confirm your entry with the **ENT** key.
- Choose the desired action for the event that there is not enough memory available on the TNC or PLC partition for the binary to ASCII conversion:
  - **Cancel if not enough space:** The update procedure is canceled if there is not enough space available, and a message to this effect appears. In this case you must save the files (*.h, *.i, and tables) externally, if they are needed, and then remove them manually in order to make space for the conversion.
  - **Delete largest files first:** The largest files (*.h, *.i, and tables) are deleted until there is enough space for the conversion.
  - **Delete oldest files first:** The oldest files (*.h, *.i, and tables on the PLC and TNC partitions) are deleted until there is enough space for the conversion.
- If there is not enough space on the SYS partition, select deletion of the oldest setup files. This procedure is repeated until there is enough memory space available.
- Confirm the update actions listed and that are to be performed.
- After the update has finished successfully, confirm the restart of the control.
- Read-in files which you had saved to a PC using TNCremoNT.

- With the **COPY SAMPLE FILES** soft key, the HEIDENHAIN standard tables for cutting data, the tilting-axis geometry, and the M-function macros as well as a prototype for a freely definable table (contains only the column **Name**) can be copied into the corresponding directories.
- The NC software exchange is completed.

Note

If a **setup.ini** file exists in the setup directory when you update **manually,** then the update is performed according to the instructions in this file. See "Automated update (setup.ini)"on page 314.

Automated software exchange as of version 340 490-02

Automated update

An automated update is usually a part of a manual update. In addition, this means that you can use the control file to reduce the number of user actions necessary for the update to a minimum, and at the same time make a backup of the PLC partition.

However, before you perform the automated update, please check whether cycle projects (HEIDENHAIN or OEM cycles) must be adapted and transferred before the update! See "Information about the OEM cycles"on page 317.

If one of the following directories exists when an iTNC is booted,
- TNC:\install\
or, if a USB memory device is connected
- USB0:\install\ (**USB0:** first partition of the first USB memory device)
and if a **setup.ini** control file is saved in this directory, then an automated update is performed according to the instructions in this control file (See "Automated update (setup.ini)" on page 314).

Note

Please note that if automated updating is selected, the update program may start with a delay due to the file size.

Software exchange via remote operation

Update via remote operation

How to perform an update via remote operation:

- In the **Programming and Editing** mode of the iTNC, press the **MOD** key.
- On the iTNC, activate remote maintenance by pressing the **Service ON** soft key.
  A service request is triggered and a connection is established.
- Transmit the **setup.elf** and **setup.zip** files to a suitable directory on the control (e.g. **TNC:\update**).
- Continue the update via remote operation as described under "Manual update".

Notes on updating via remote operation

- If remote maintenance is active when the update is started, the remote maintenance is deactivated (this does not affect the current remote operation).
- After the reboot or after the update was canceled due to an error, a service request is triggered. The service request information indicates whether the update was successful.
- If the control does not boot due to missing or incorrect machine parameters, a service request is triggered until the "power interrupt" stage of the boot process is reached.

Procedure for exchanging the NC software up to and including 340 422-12

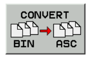
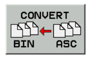
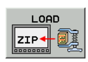
Before exchanging the NC software, ensure that the free space on the hard disk of the MC 42x(B,C) is at least 50% the size of the occupied space. If that is not the case, you must save the files to a PC, e.g., with the TNCremoNT data-transfer software for PCs.

Note

When the control starts, it checks whether there is enough space on the hard disk for system files. If not, the error message **Too many setup files** appears. In this event, delete any unnecessary setup files from the hard disk (See "Deleting the packed files of existing NC software" on page 325.)

- While in the **Programming and Editing** operating mode, press the MOD key.
- Enter the code number 95148 and confirm your entry with the ENT key.
- If you want to use the Ethernet interface for transferring the NC software from a PC, proceed as follows:
  - While in the **Machine-parameter programming** mode, press the PGM MGT key.
  - Press the **NETWORK** soft key.
  - Select the PC to be connected with the cursor keys and press the **CONNECT DRIVE** soft key.
  - To exit the list of network drives, press the **END** soft key.
  - To exit the program manager, press the **END** soft key.
- While in the **Machine-parameter programming** mode, press the MOD key.
- Press the **UPDATE DATA** soft key.
- The name and path of a log file can be entered after **Path =** in the header.
- Press the **BIN** → **ASC** soft key to convert the files on the hard disk from binary to ASCII format.

Soft keys for update functions

Soft key	Function
	Convert the files on the hard disk from binary format to ASCII format and save nonvolatile markers in the PLCMEM.A file.
	Convert the files on the hard disk from ASCII format to binary format and save nonvolatile markers in the PLCMEM.A file.
	Copy cutting-data tables, tables for tilting-axis geometry, and the table of M-function macros from the SYS partition into the corresponding directories of the TNC or PLC partition, and create prototypes of the tables.
	Activate or delete existing NC software.
	Exchange NC software.

Equivalent file name extensions in ASCII and binary format					
.H	.H%	.I	.I%	.T	.T%
.TCH	.TC%	.D	.D%	.P	.P%
.PNT	.PN%	.COM	.CO%	.CMA	.CM%

- Press the **NCV** → **iTNC** soft key.
- In the upper part of the window, you select the folder containing the *.zip file of the new NC software. The folder contents are displayed in the lower part of the window.
- To switch to the lower part of the window, press the **FILES** soft key. To return to the upper part of the window, use the **PATH** soft key. In the lower part of the window, select the *.zip file of the new NC software with the cursor keys and press ENT. Following that, the NC software exchange will start.
- All NC software versions that exist in the control are shown in the following list box.
- Select the new NC software with the arrow keys and press the **SELECT** soft key. The selected NC software is marked with an asterisk (*) in the **Sel** column. Confirm your selection with the **YES** soft key. The control activates the selected NC software and performs a reset. Press the END key, the **NO** soft key or the **END** soft key to exit the list box without making a new selection.
- If required, complete or delete the machine parameters.
- While in the **Programming and Editing** operating mode, press the MOD key.
- Enter the code number 95148 and confirm your entry with the ENT key.

- While in the **`Machine-parameter programming`** mode, press the MOD key.
- Press the **`UPDATE DATA`** soft key.
- Press the **`ASC`** → **`BIN`** soft key to convert the files on the hard disk from ASCII format back to binary format.
- Read-in files which you had saved to a PC.
- The NC software exchange is completed.
- With the **`COPY SAMPLE FILES`** soft key, the HEIDENHAIN standard tables for cutting data, the tilting-axis geometry, and the M-function macros as well as a prototype for a freely definable table (contains only the column **`Name`**) can be copied into the corresponding directories.

NC software exchange from standard version to export version (and vice versa)

The control version (iTNC 530 or iTNC 530 E) is stored in the SIK. If the NC software is exchanged on a software version different from that stored on the SIK, after the control starts up a message appears that the control can be operated only as a programming station. This message must be acknowledged.
Procedure for exchanging the NC software:

- Exchange the NC software as described above.

After the control powers up, **`Incorrect software version`** or **`Falsche Softwareversion`** appears.

- Switch off the control.
- Exchange the SIK for a new, appropriate SIK (for the ID, see page 250); for the location of the SIK in the MC 42x(B,C), See "Additional Control Loops or Software Options" on page 302.
- Switch on the control.

Since the new SIK has another SIK number, the options that are enabled on the old SIK must be re-enabled on the new SIK. After you state your SIK number, HEIDENHAIN can give you the code number for enabling the options.

To make it possible to identify the control from outside, after you indicate the ID and serial number of the control, you will receive a new ID label with the new data.

- Stick the new ID label with the new control designation on the MC42x(B,C) (E).

Attention

After you have changed the NC software from the standard version to the export version, you must delete the packed files of the standard version from the hard disk, since the packed files are also subject to export authorization.
After the export version has been started, a prompt appears, asking if the compressed files of the standard version are to be deleted from the hard disk. If you answer with YES, all compressed files with the names of the standard version are deleted from the hard disk.
The procedure for deleting compressed files in on page 325.

Activating existing NC software

Before activating existing NC software, ensure that the free space on the hard disk of the MC 42x(B,C) is at least 50% the size of the occupied space. If that is not the case, you must save the files to a PC, e.g., with the TNCremoNT data-transfer software for PCs.

- While in the **Programming and Editing** operating mode, press the MOD key.
- Enter the code number 95148 and confirm your entry with the ENT key.
- While in the **Machine-parameter programming** mode, press the MOD key.
- Press the **UPDATE DATA** soft key.
- The name and path of a log file can be entered after **Path =** in the header.
- Press the **BIN** → **ASC** soft key to convert the files on the hard disk from binary to ASCII format.

Equivalent file name extensions in ASCII and binary format					
.H	.H%	.I	.I%	.T	.T%
.TCH	.TC%	.D	.D%	.P	.P%
.PNT	.PN%	.COM	.CO%	.CMA	.CM%

- Press the **SELECT** soft key.
- All NC software versions that exist in the control are shown in the selection window that appears.
- Select the NC software to be activated with the arrow keys and press the **SELECT** soft key. The selected NC software is marked with an asterisk (*) in the **Sel** column. Confirm your selection with the **YES** soft key. The control activates the selected NC software and performs a reset. Press the END key, the **NO** soft key or the **END** soft key to exit the list box without making a new selection.
- If required, complete or delete the machine parameters.
- While in the **Programming and Editing** operating mode, press the MOD key.
- Enter the code number 95148 and confirm your entry with the ENT key.
- While in the **Machine-parameter programming** mode, press the MOD key.
- Press the **UPDATE DATA** soft key.
- Press the **ASC** → **BIN** soft key to convert the files on the hard disk from ASCII format back to binary format.
- The activation of the NC software is completed.
- With the **COPY SAMPLE FILES** soft key, the HEIDENHAIN standard tables for cutting data, the tilting-axis geometry, and the M-function macros as well as a prototype for a freely definable table (contains only the column **Name**) can be copied into the corresponding directories.

Deleting the packed files of existing NC software

- While in the **Programming and Editing** operating mode, press the MOD key.
- Enter the code number 95148 and confirm your entry with the ENT key.
- While in the **Machine-parameter programming** mode, press the MOD key.
- Press the **UPDATE DATA** soft key.
- Press the **SELECT** soft key.
- All NC software versions that exist in the control are shown in the following options display.
- Select the NC software to be deleted with the arrow keys and press the **DELETE** soft key to delete all packed NC software files. The currently active NC software is marked with an asterisk (*) in the **Sel** column. Confirm your selection with the **YES** soft key. Press the **NO** soft key or the **END** soft key to exit the list box without deleting an NC software.

> Note
>
> If the packed files of an NC software, including the currently active software, are deleted, the respective software cannot be activated via the selection window any longer. The software concerned must then again be transferred to the control (See "Procedure for exchanging the NC software up to and including 340 422-12" on page 321).
> The deletion of the packed files of the currently active NC software has no other effects.

Entries in the log file

If errors occur during conversion, the TNC will display error messages and record them in the log file. During the NC software switch, the name and path of a log file can be entered in the header after **Path =**; the extension .A must be used. If no entry is made in this line, the file TNC:\CVREPORT.A is created.

Each error message contains

- Error message
- Error code
- Error cause
- File concerned

Example:

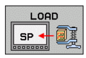
```
==================================================
ERROR              :REMANENT PLC DATA NOT RESTORED
ERRNO              :2
ERROR MESSAGE      :Program name not found
FILE               :PLCMEM.A
==================================================
```

Error message	Meaning
CANNOT OPEN DIRECTORY	File could not be opened
REMANENT PLC DATA NOT RESTORED	No access to the file PLCMEM.A
NOT ENOUGH SPACE	Too little free memory on the hard disk
CONVERSION BIN ASC FAILED	A binary file has an incorrect format (e.g., binary format from an old NC software)
CONVERSION ASC BIN FAILED	An ASCII file on the hard disk is incorrect

2.4.6 Installing a service pack

When needed, HEIDENHAIN prepares service packs for certain NC software versions. A service pack is loaded in addition to an already completely installed NC software.
When the control is started, a note regarding the installed service pack is shown. This can be replaced by a special logo (See "Powering up and shutting down the control" on page 1242).
The latest service pack always includes all changes from earlier service packs.

HEIDENHAIN recommends installing all released service packs.

Soft key	Function
	Installs a service pack.

Danger

When needed, HEIDENHAIN prepares service packs for the various versions of the NC software. Registered customers can download these service packs from the HEIDENHAIN FileBase on the Internet. Installation of a service pack via the already installed NC software implements important error fixes. Please ensure that the NC software always contains the latest service pack before you ship the machine.

Perform all tests required of the machine or the NC software again after having installed the service pack.

Attention

If a service pack has already been installed, it will not be possible to install a service pack with a lower index. Up to software 340 490-01, the person installing the service pack must check this. As of software 340 490-02, the software will check this during the installation of a service pack, and a message will be displayed if an error is found.

Note

A service pack may only be loaded onto completely installed NC software (files in binary format), and may only be done by trained personnel.

The service pack consists of packed files (*.zip). The packed files are transferred to the hard disk of the control and unpacked.

The installation of a complete software version **(setup.elf** or **setup.omf)** is also possible via the **LOAD SP** soft key.

It is not necessary to convert from binary format to ASCII format.

- If you want to use the Ethernet interface for transferring the service pack from a PC, proceed as follows:
  - While in the **Programming and Editing** operating mode, press the PGM MGT key.
  - Press the **NETWORK** soft key.
  - Select the PC to be connected with the cursor keys and press the **CONNECT DRIVE** soft key.
  - To exit the list of network drives, press the **END** soft key.
  - To exit the program manager, press the **END** soft key.
- While in the **Programming and Editing** operating mode, press the MOD key.
- Press the **LOAD SP** soft key.
- In the upper part of the window, select the folder containing the *.zip file of the service pack. The folder contents are displayed in the lower part of the window.
- To switch to the lower part of the window, press the **FILES** soft key. To return to the upper part of the window, use the **PATH** soft key. In the lower part of the window, select the *.zip file of the service pack with the cursor keys and press ENT. The installation of the service pack starts.
- Service pack installation finishes.

Note

When installing a service pack, it is not necessary to convert the binary format to ASCII format or vice versa, or to make a back-up of the non-volatile operands.

2.4.7 Data backup

HEIDENHAIN provides a data backup program called TNCBACK.EXE free of charge.

HEIDENHAIN recommends that the machine tool builder use the software TNCBACK.EXE to save all his **machine-specific data** to a floppy disk, and that he supply the disk with the machine. The disk must also contain the program TNCBACK.EXE.

The customer, too, can save his TNC data before exchanging the control. It is also advisable that the customer save all of the files and programs created on the iTNC at regular intervals. Data backup is described in detail in the "Readme" file, which is included on the disk.

3 Mounting and Electrical Installation

3.1 General Information

Attention

Keep the following in mind during mounting and electrical installation:

- National regulations for power installations
- Interference and noise immunity
- Conditions of operation
- Mounting attitude

3.1.1 Safety precautions

Danger

Ensure that the main switch of the control or machine is switched off when you engage or disengage connecting elements or connection clamps.

Danger

Ensure that the grounding conductor is connected. Interruptions in the equipment grounding conductor may cause damage to persons or property.

Danger

Incorrect or non-optimized input values can lead to faulty machine performance and therefore to serious injury to persons and damage to equipment. Modifications of the machine parameters should be done with caution and uncontrolled axis motions should be taken into account.

Attention

In order to be able to judge the behavior of an NC controlled machine, you need to have fundamental knowledge about drives, inverters, controls and encoders. Inappropriate use may cause considerable damage to persons or property.
HEIDENHAIN does not accept any responsibility for direct or indirect damage caused to persons or property through incorrect use or operation of the machine.

Danger

The interfaces for the PLC inputs/outputs, machine operating panel and PL connection comply with the requirements for basic insulation in accordance with **IEC 742 EN 50 178**.
Only units that comply with the requirements of **IEC 742 EN 50 178** for basic insulation may be connected, otherwise damage to persons or property may be caused. The maximum dc voltage mean value of the PLC inputs is 31 V.

3.1.2 Degrees of protection

The following components fulfill the requirements for IP54 (dust and splash-proof protection).

- Visual display unit (when properly installed)
- Keyboard unit (when properly installed)
- Machine operating panel (when properly installed)
- Handwheel

Danger

All electric and electronic control components must be installed in an environment (e.g. electrical cabinet, housing) that fulfills the requirements of protection class IP54 (dust and splash-proof protection).

3.1.3 Electromagnetic compatibility

This unit fulfills the requirements for Class A according to EN 55022 and is intended for operation in industrially zoned areas.

Protect your equipment from interference by observing the following rules and recommendations.

Likely sources of interference

Interference is mainly produced by capacitive and inductive coupling from electrical conductors or from device inputs/outputs, such as:

- Strong magnetic fields from transformers or electric motors
- Relays, contactors and solenoid valves
- High-frequency equipment, pulse equipment and stray magnetic fields from switch-mode power supplies
- Power lines and leads to the above equipment

Protective measures

- Keep a minimum distance of 20 cm from the MC 42x(B,C), CC 42x and its leads to interfering equipment.
- Keep a minimum distance of 10 cm from the MC 42x(B,C), CC 42x and its leads to cables that carry interference signals. For cables in metallic ducting, adequate decoupling can be achieved by using a grounded separation shield.
- Shielding according to IEC 61800-5-1.
- Use potential compensating lines with a cross section of 6 mm^2
- Use only genuine HEIDENHAIN cables, connectors and couplings.

3.1.4 ESD protection

Always assume that all electronic components and assemblies are endangered by electrostatic discharge (ESD).

To ensure protection from ESD, follow the precautionary measures described in IEC 61340-5-1, IEC 61340-5-2 and IEC 61340-4-1.

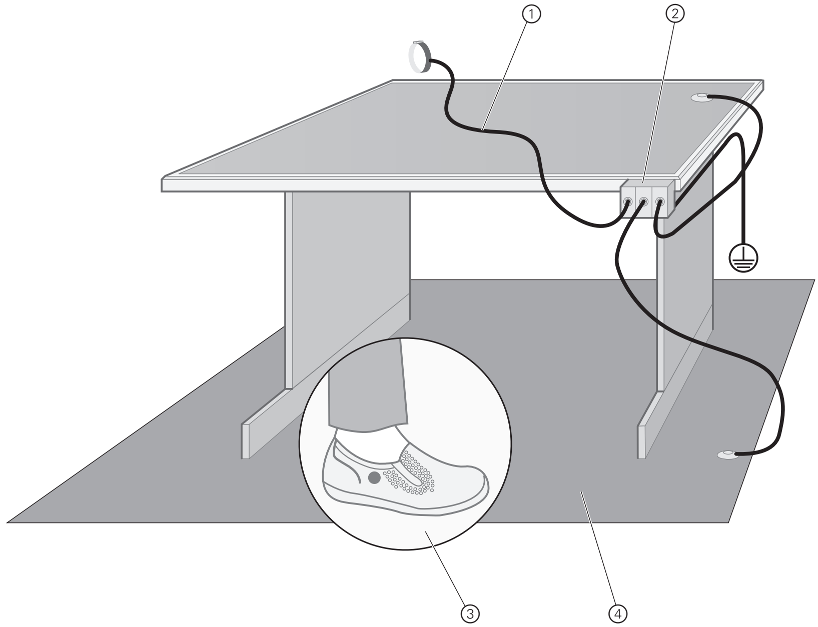
Note

Improper handling can result in damage to the components or assemblies due to ESD!

The following are some points covered in the above mentioned standards:

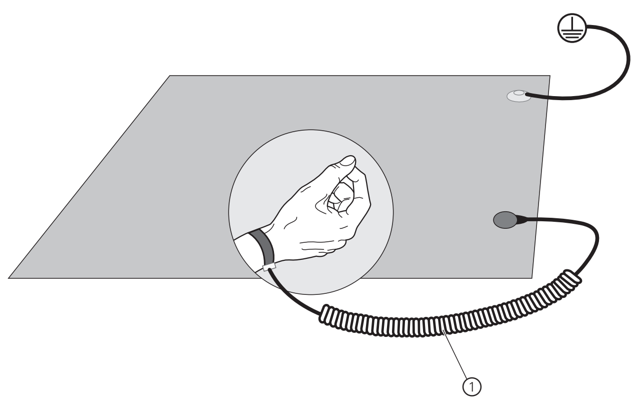
- When handling electrostatically endangered components or assemblies (e.g. exchange, installation, shipping), always comply with the precautionary measures described in IEC 61340-5-1, IEC 61340-5-2 and IEC 61340-4-1.
- Store and transport ESD-sensitive components in ESD protective containers.
- Ensure during handling the proper grounding of the working area (e.g. tool, workbench, packaging) and the person.
- Inspect the ESD protection system regularly.

The following figure shows how a suitable working area could look in accordance with IEC 61340.



- 1: Wristband with 1 MOhm grounding cable for grounding the person
- 2: Grounded connection for wristbands, floor mats, table mats etc. for equipotential bonding
- 3: Dissipative shoes
- 4: Dissipative flooring or floor mat

An important part of the working area is a suitable working surface with a wristband with 1 MOhm grounding resistance for personal grounding:



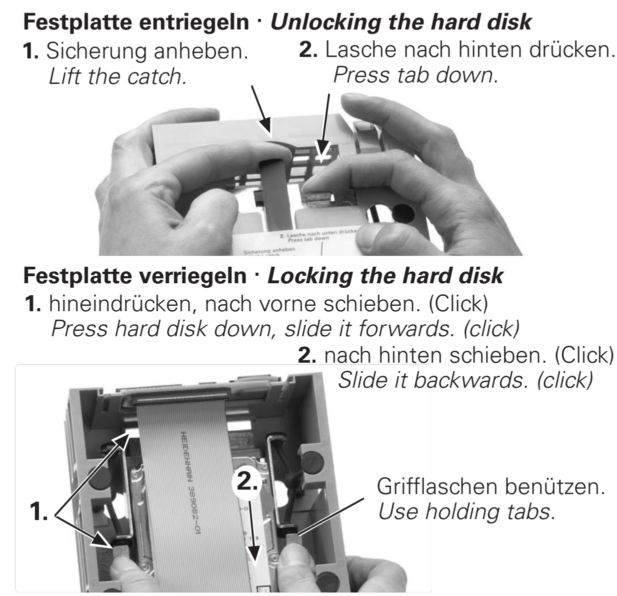
3.2 Handling of the HDR Hard Disk and the SIK

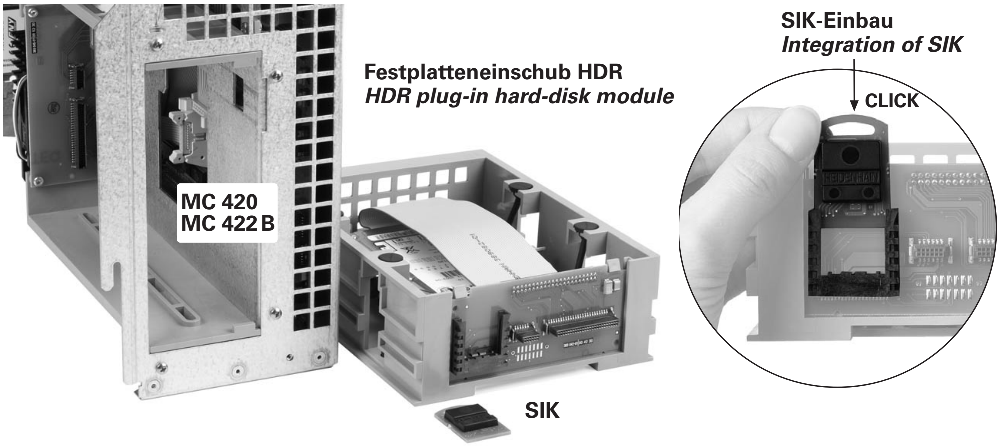
Shipping brace of the hard disk

The HDR hard disks of the MC 42x(B,C) are fitted with a shipping brace. Before putting the iTNC 530 into service, the shipping brace of the hard disk must be removed.

> Attention
>
> Do not transport the HDR with the MC 42x(B,C) after you have installed the HDR. If the entire machine is being transported, or the MC is being transported inside the electrical cabinet, the shipping brace for the hard disk is usually not required. However, if the possibility exists that the hard disk could be subject to increased shock or vibration loads, then you must remove the hard disk from the MC for transport, reinstall the shipping brace in the HDR, and send the HDR separately in the original packaging.

Should servicing become necessary (i.e. the HDR is being shipped on its own), the hard disk must be secured with the shipping brace.





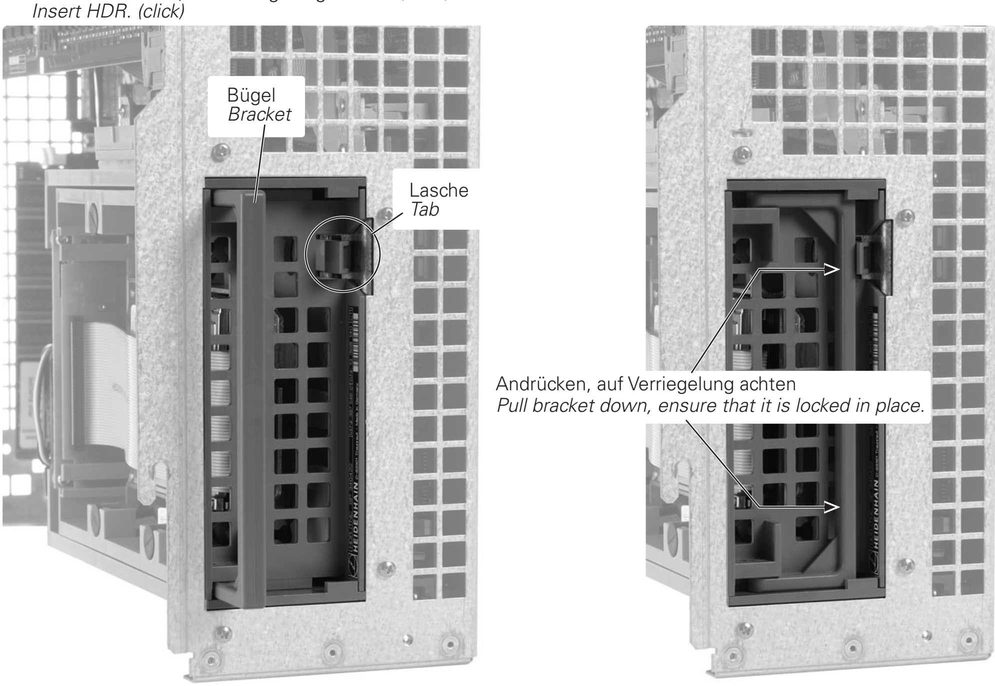
Achtung: Im Servicefall Festplatte verriegeln, SIK entnehmen und aufbewahren.
Note: *Should servicing become necessary, lock the hard disk, remove the SIK and keep it in a safe place.*

HDR-Einbau · *Installation of HDR*

1. SIK einbauen
Integrate SIK.

2. HDR einschieben, auf Verriegelung achten. (Click)
Insert HDR. (click)

3. Bügel umlegen und gegen Lasche andrücken. (Click)
Turn the bracket over and press it against the tab until it "clicks" in place.



HDR-Ausbau in umgekehrter Reihenfolge · *To remove the HDR, proceed in reverse order*

1. Bügel entriegeln (Click) und aufstellen.
Unlock the bracket so that it "unclicks" and pull it up.

2. Lasche drücken und HDR rausziehen.
Press the tab and pull out the HDR.

3. SIK entnehmen.
Remove the SIK.

3.3 Environmental Conditions

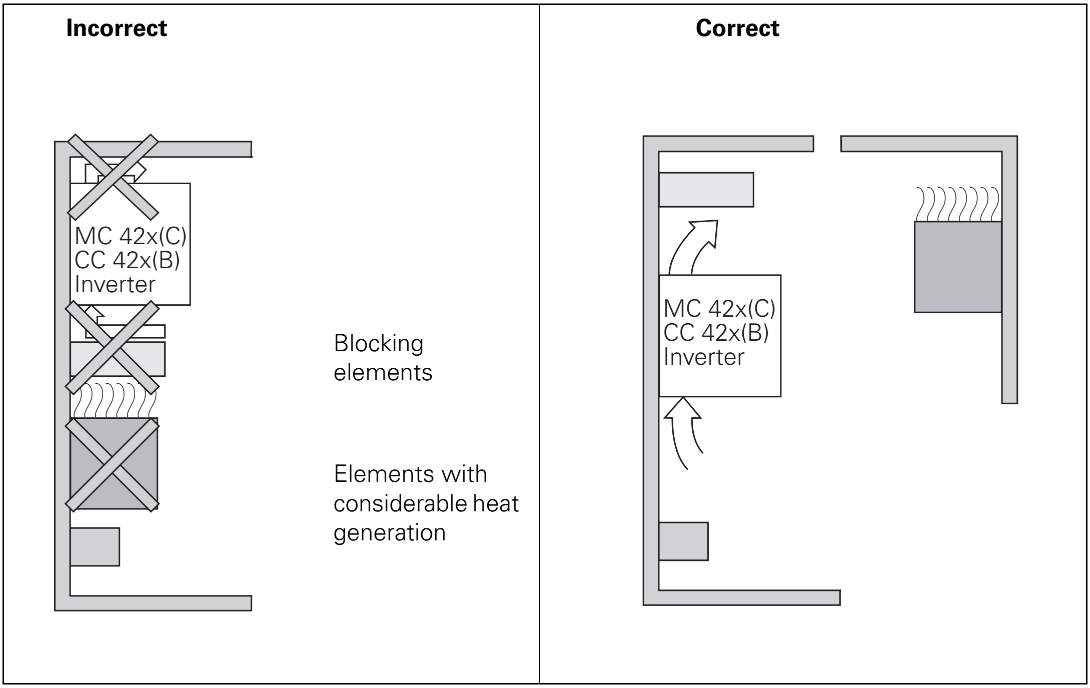
3.3.1 Heat generation and cooling

> **Danger**
>
> The permissible ambient temperature in operation is between 0 °C and 40 °C. Any deviation from this will impair the operating safety of the machine.

A heat exchanger or a cooling unit is preferable for controlling the internal temperature of the electrical cabinet.

If filtered air is blown into the electrical cabinet for cooling purposes, the standard IEC 61800-5-1 applies, which permits contamination level 2.

> **Danger**
>
> Be sure to take the measures required for preventing dust from entering the electrical cabinet.
> Dust depositing inside electrical devices may cause them to fail and impair the safety of the system.



3.3.2 Humidity

Permissible humidity:

- Maximum 75% in continuous operation
- Maximum 95% for not more than 30 days a year (randomly distributed)

In tropical areas it is recommended that the iTNC 530 not be switched off, so as to avoid dew deposition on the circuit boards.

3.3.3 Installation elevation

The maximum installation elevation is 3000 m above sea level.

3.3.4 Mechanical vibration

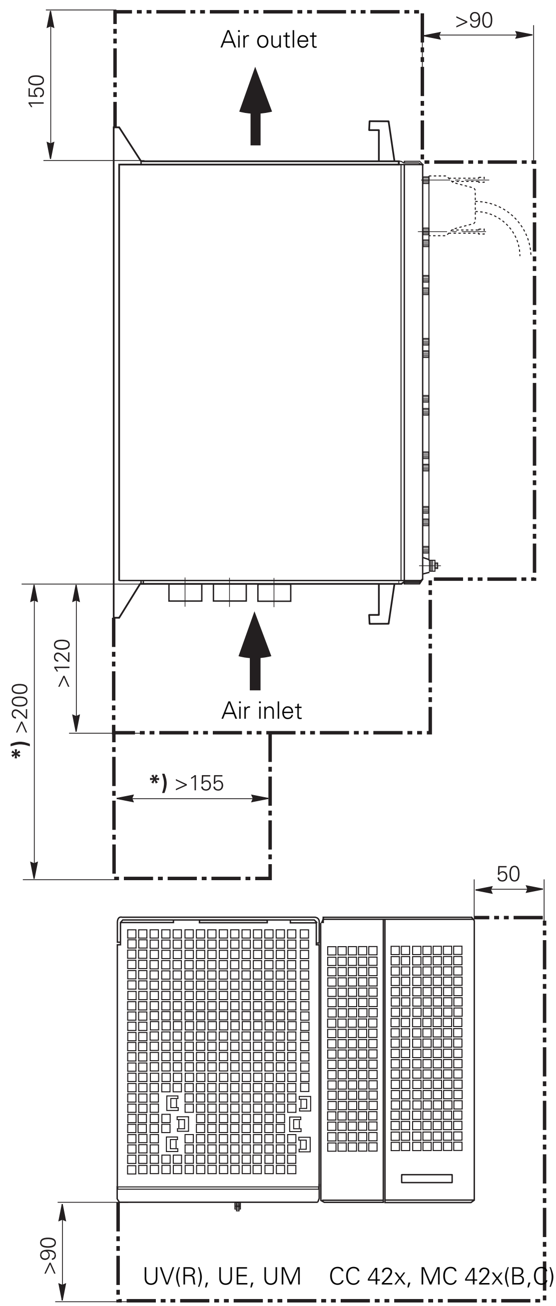
Permissible vibration: ± 0.075 mm, 10 to 41 Hz
5 m/s^2, 41 Hz to 500 Hz

Permissible shock: 50 m/s^2, 11 ms

Permissible shock with shipping brace for hard disk: 300 m/s^2, 11 ms

3.3.5 Mounting attitude of MC 42x(B,C), CC 42x, UV xxx, UM xxx, UE 2xx B

> **Attention**
>
> When mounting, please observe proper minimum clearance, space requirements, length and position of the connecting cables.



Leave space for air circulation! Temperatures of > 150 °C are possible with UE 21xB with integral braking resistor; Do not mount any temperature-sensitive components!

Leave space for servicing!

Leave space for air circulation and servicing!

*) Space for removing the HDR hard disk

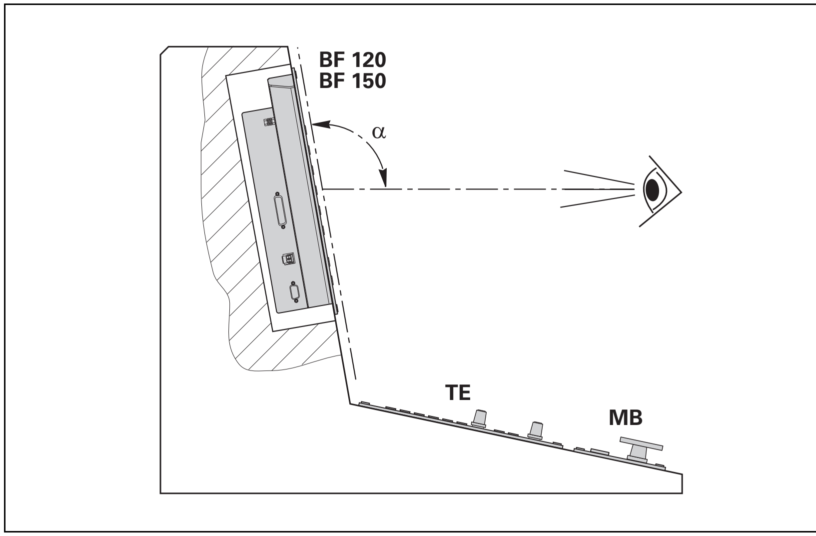
Leave space for servicing and connecting cables!

3.3.6 Mounting position of display

BF 120, BF 150

The BF 120 and BF 150 flat-panel displays must be viewed with a slight backward slant.

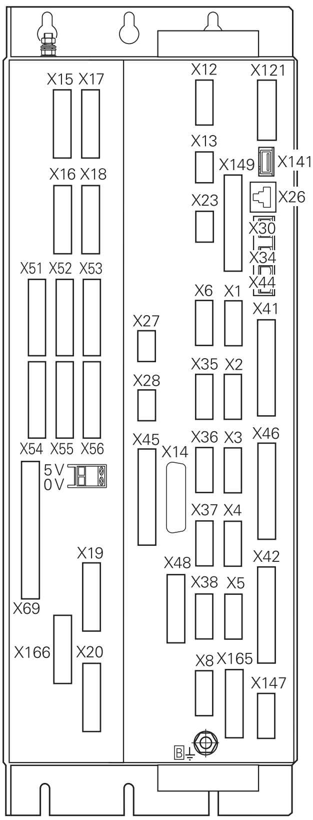
▶ During installation, ensure a viewing angle of 150° > α > 90°.



3.4 Connection Overview for iTNC 530

> **Attention**
>
> Do not engage or disengage any connecting elements while the unit is under power!

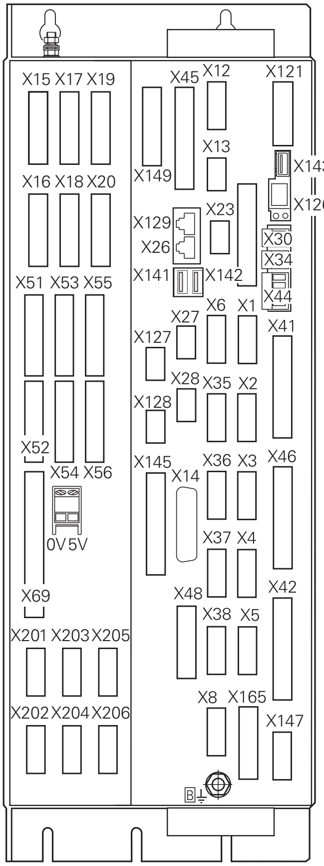
MC 422C / 5 position encoder inputs and CC 422 with 6 control loops



Connection overview	Connector	Function	Page
	X1 to X5	Position encoder	376
	X35 to X38	Not occupied	–
	X15 to X20	Speed encoder	378
	X51 to X60	PWM output	386
	X8	Nominal value output, analog	391
	X12	TS touch-trigger probe	395
	X13	TT tool touch probe	397
	X23	Handwheel	404
	X26	Ethernet data interface	400
	X27	RS-232-C/V.24 data interface	401
	X28	RS-422/V.11 data interface	403
	X141, X142	USB Interface (X142 at bottom of unit)	461
	X30	24 V spindle reference signal	410
	X34	24 V control-is-ready signal output	370
	X41	PLC output	418
	X42	PLC input	412
	X44	24 V PLC supply voltage	366
	X45	Control panel	445
	X46	Machine operating panel	438
	X147	PLC expansion	424
	X48	PLC analog input	388
	X149	BF 150 (BF 120) visual display unit	447
	X69	Power supply	354
	X121	Reserved	–
	X165, X166	Reserved	–
	X74	5 V power supply	355
	X150	Axis-specific drive enabling (at bottom of housing)	373
	B	Signal ground	–
	⏚	Ground lead (YL/GN)	–

MC 422C DP without position encoder inputs and CC 424(B) with 6 control loops

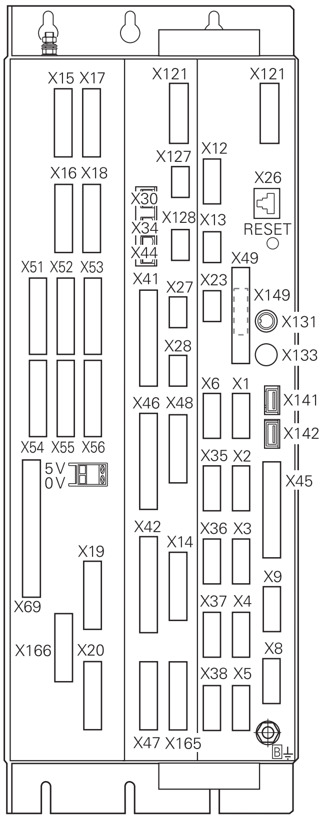
Connection overview



Connector	Function	Page
X1 to X5	Not occupied	376
X35 to X38	Not occupied	–
X15 to X20	Speed encoder	378
X51 to X60	PWM output	386
X8	Nominal value output, analog	391
X12	TS touch-trigger probe	395
X13	TT touch-trigger probe	397
X23	Handwheel	404
X26	Ethernet data interface	400
X27	RS-232-C/V.24 data interface	401
X28	RS-422/V.11 data interface	403
X127	RS-232-C/V.24 data interface (only for Windows 2000)	401
X128	RS-422/V.11 (only for Windows 2000)	403
X141, X142	USB interface	461
X143, X144	USB Interface	461
X30	24 V spindle reference signal	410
X34	24 V for control-is-ready signal output	370
X41	PLC output	418
X42	PLC input	412
X44	24 V PLC supply voltage	366
X45	Control panel	445
X46	Machine operating panel	438
X147	PLC expansion	424
X48	PLC analog input	388
X149	BF 150 visual display unit	447
X126, X129	Reserved for Ethernet	401
X69	Power supply	–
X121	Reserved	–
X165	Reserved	–
X74	5 V power supply	355
X150	Axis-specific drive enabling (at bottom of unit)	373
B	Signal ground	–
⏚	Ground lead (YL/GN)	461

MC 422B / 5 position encoder inputs and CC 422 with 6 control loops

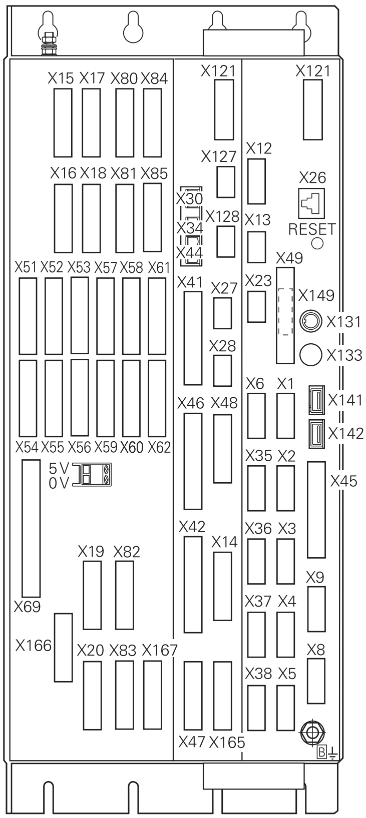
Connection overview



Connector	Function	Page
X1 to X5	Position encoder	376
X35 to X38	Not occupied	–
X15 to X20	Speed encoder	378
X51 to X60	PWM output	386
X8, X9	Nominal value output, analog	391
X12	TS touch-trigger probe	395
X13	TT touch-trigger probe	397
X23	Handwheel	404
X26	Ethernet data interface	400
X27	RS-232-C/V.24 data interface	401
X28	RS-422/V.11 data interface	403
X127	RS-232-C/V.24 data interface (only for Windows 2000)	401
X128	RS-422/V.11 (only for Windows 2000)	403
X141, X142	USB interface	461
X30	24 V spindle reference signal	410
X34	24 V for control-is-ready signal output	370
X41	PLC output	418
X42	PLC input	412
X44	24 V PLC supply voltage	366
X45	Control panel	445
X46	Machine operating panel	438
X47	PLC expansion	424
X48	PLC analog input	388
X149 (X49)	BF 150 (BF 120) visual display unit	447
X131	Reserved	–
X69	Power supply	354
X121, X125	Reserved	–
X165, X166	Reserved	–
X74	5 V power supply	355
X150	Axis-specific drive enabling (at bottom of housing)	373
B	Signal ground	–
⏚	Ground lead (YL/GN)	–

MC 422B / 10 position encoder inputs and CC 422 with 10 or 12 control loops

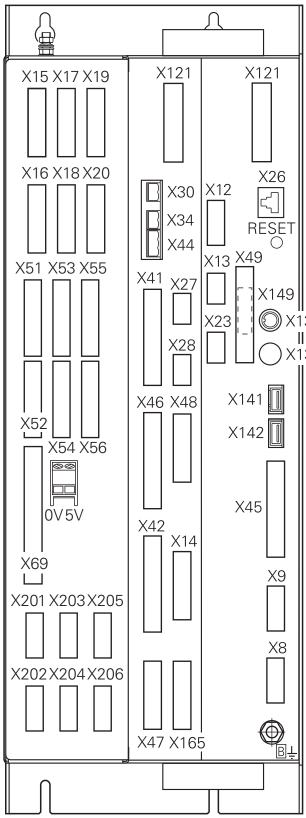
Connection overview



Connector	Function	Page
X1 to X5	Position encoder	376
X35 to X38	Position encoder	376
X15 to X20	Speed encoder	378
X80 to X83	Speed encoder	378
X84, X85	Speed encoder (12 control loops)	378
X51 to X60	PWM output	386
X61, X62	PWM output (12 control loops)	386
X8, X9	Nominal value output, analog	391
X12	TS touch-trigger probe	395
X13	TT touch-trigger probe	397
X23	Handwheel	404
X26	Ethernet data interface	400
X27	RS-232-C/V.24 data interface	401
X28	RS-422/V.11 data interface	403
X127	RS-232-C/V.24 data interface (only for Windows 2000)	401
X128	RS-422/V.11 (only for Windows 2000)	403
X141, X142	USB interface	461
X30	24 V spindle reference signal	410
X34	24 V for control-is-ready signal output	370
X41	PLC output	418
X42	PLC input	412
X44	24 V PLC supply voltage	366
X45	Control panel	445
X46	Machine operating panel	438
X47	PLC expansion	424
X48	PLC analog input	388
X149 (X49)	BF 150 (BF 120) display	447
X131	Reserved	–
X69	Power supply	354
X121, X125	Reserved	–
X165, X166	Reserved	–
X74	5 V power supply	355
X150, X151	Axis-specific drive enabling	373
B	Signal ground	–
⏚	Ground lead (YL/GN)	–

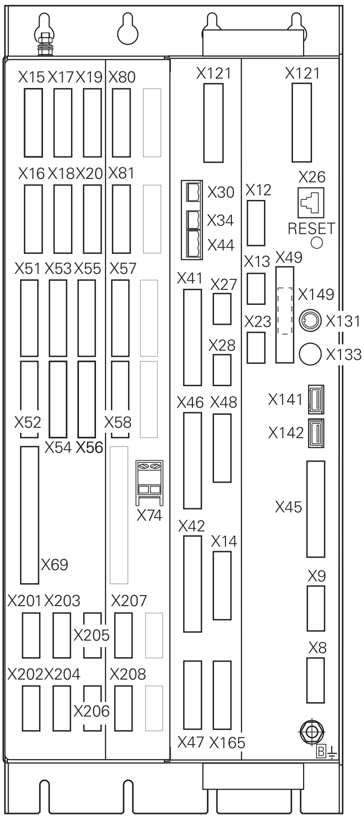
MC 422B and CC 424(B) with 6 control loops

Connection overview



Connector	Function	Page
X201 to X206	Position encoder	376
X15 to X20	Speed encoder	378
X51 to X56	PWM output	386
X8, X9	Nominal value output, analog	391
X12	TS touch-trigger probe	395
X13	TT touch-trigger probe	397
X23	Handwheel	404
X26	Ethernet data interface	400
X27	RS-232-C/V.24 data interface	401
X28	RS-422/V.11 data interface	403
X127	RS-232-C/V.24 data interface (only for Windows 2000)	401
X128	RS-422/V.11 (only for Windows 2000)	403
X141, X142	USB interface	461
X30	24 V spindle reference signal	410
X34	24 V for control-is-ready signal output	370
X41	PLC output	418
X42	PLC input	412
X44	24 V PLC supply voltage	366
X45	Control panel	445
X46	Machine operating panel	438
X47	PLC expansion	424
X48	PLC analog input	388
X149 (X49)	BF 150 (BF 120) display	447
X131	Reserved	–
X69	Power supply	354
X121, X125	Reserved	–
X165	Reserved	–
X74	5 V power supply	355
X150	Axis-specific drive enabling (at bottom of housing)	373
B	Signal ground	–
⏚	Ground lead (YL/GN)	–

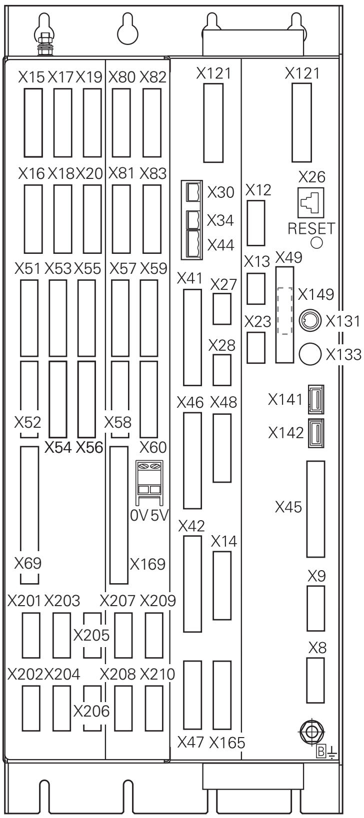
MC 422B and CC 424(B) with 8 control loops



Connection overview	Connector	Function	Page
	X201 to X208	Position encoder	376
	X15 to X20	Speed encoder	378
	X80 to X81	Speed encoder	378
	X51 to X58	PWM output	386
	X8, X9	Nominal value output, analog	391
	X12	TS touch-trigger probe	395
	X13	TT touch-trigger probe	397
	X23	Handwheel	404
	X26	Ethernet data interface	400
	X27	RS-232-C/V.24 data interface	401
	X28	RS-422/V.11 data interface	403
	X127	RS-232-C/V.24 data interface (only for Windows 2000)	401
	X128	RS-422/V.11 (only for Windows 2000)	403
	X141, X142	USB interface	461
	X30	24 V spindle reference signal	410
	X34	24 V for control-is-ready signal output	370
	X41	PLC output	418
	X42	PLC input	412
	X44	24 V PLC supply voltage	366
	X45	Control panel	445
	X46	Machine operating panel	438
	X47	PLC expansion	424
	X48	PLC analog input	388
	X149 (X49)	BF 150 (BF 120) display	447
	X131	Reserved	–
	X69	Power supply	354
	X121, X125	Reserved	–
	X165	Reserved	–
	X74	5 V power supply	355
	X150	Axis-specific drive enabling (at bottom of housing)	373
	B	Signal ground	–
	⏚	Ground lead (YL/GN)	–

MC 422B and CC 424(B) with 10 control loops

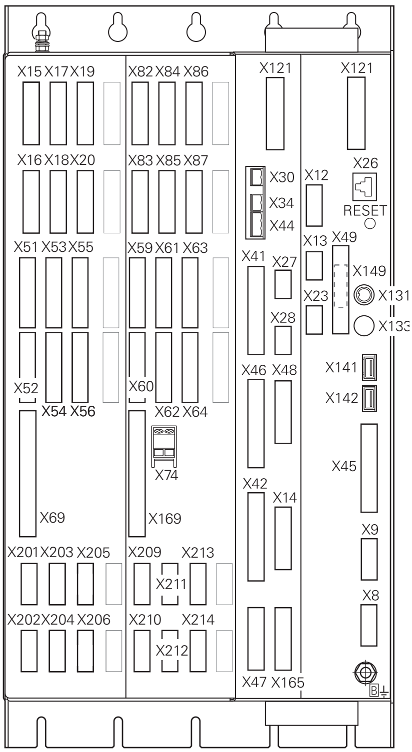
Connection overview



Connector	Function	Page
X201 to X210	Position encoder	376
X15 to X20	Speed encoder	378
X80 to X83	Speed encoder	378
X51 to X60	PWM output	386
X8, X9	Nominal value output, analog	391
X12	TS touch-trigger probe	395
X13	TT touch-trigger probe	397
X23	Handwheel	404
X26	Ethernet data interface	400
X27	RS-232-C/V.24 data interface	401
X28	RS-422/V.11 data interface	403
X127	RS-232-C/V.24 data interface (only for Windows 2000)	401
X128	RS-422/V.11 (only for Windows 2000)	403
X141, X142	USB interface	461
X30	24 V spindle reference signal	410
X34	24 V for control-is-ready signal output	370
X41	PLC output	418
X42	PLC input	412
X44	24 V PLC supply voltage	366
X45	Control panel	445
X46	Machine operating panel	438
X47	PLC expansion	424
X48	PLC analog input	388
X149 (X49)	BF 150 (BF 120) display	447
X131	Reserved	–
X69, X169	Power supply	354
X121, X125	Reserved	–
X165	Reserved	–
X74	5 V power supply	355
X150, X151	Axis-specific drive enabling (at bottom of housing)	373
B	Signal ground	–
⏚	Ground lead (YL/GN)	–

MC 422B and CC 424(B) with 12 control loops

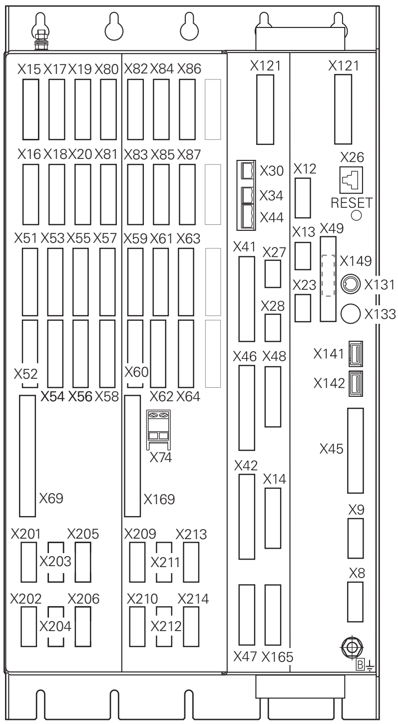
Connection overview



Connector	Function	Page
X201 to X206	Position encoder	376
X209 to X214	Position encoder	376
X15 to X20	Speed encoder	378
X82 to X87	Speed encoder	378
X51 to X56	PWM output	386
X59 to X64	PWM output	386
X8, X9	Nominal value output, analog	391
X12	TS touch-trigger probe	395
X13	TT touch-trigger probe	397
X23	Handwheel	404
X26	Ethernet data interface	400
X27	RS-232-C/V.24 data interface	401
X28	RS-422/V.11 data interface	403
X127	RS-232-C/V.24 data interface (only for Windows 2000)	401
X128	RS-422/V.11 (only for Windows 2000)	403
X141, X142	USB interface	461
X30	24 V spindle reference signal	410
X34	24 V for control-is-ready signal output	370
X41	PLC output	418
X42	PLC input	412
X44	24 V PLC supply voltage	366
X45	Control panel	445
X46	Machine operating panel	438
X47	PLC expansion	424
X48	PLC analog input	388
X149	BF 150 display unit	447
X131	Reserved	–
X69, X169	Power supply	354
X121, X125	Reserved	–
X165	Reserved	–
X74	5 V power supply	355
X150, X151	Axis-specific drive enabling	373
B	Signal ground	–
⏚	Ground lead (YL/GN)	–

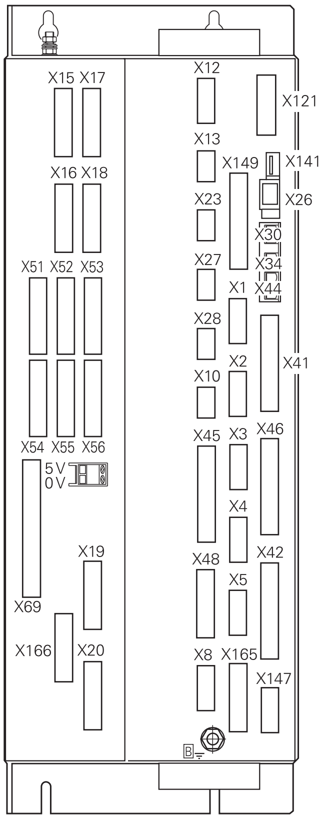
MC 422B and CC 424(B) with 14 control loops

Connection overview



Connection overview	Connector	Function	Page
	X201 to X206	Position encoder	376
	X209 to X214	Position encoder	376
	X15 to X20	Speed encoder	378
	X80 to X87	Speed encoder	378
	X51 to X58	PWM output	386
	X59 to X64	PWM output	386
	X8, X9	Nominal value output, analog	391
	X12	TS touch-trigger probe	395
	X13	TT touch-trigger probe	397
	X23	Handwheel	404
	X26	Ethernet data interface	400
	X27	RS-232-C/V.24 data interface	401
	X28	RS-422/V.11 data interface	403
	X127	RS-232-C/V.24 data interface (only for Windows 2000)	401
	X128	RS-422/V.11 (only for Windows 2000)	403
	X141, X142	USB interface	461
	X30	24 V spindle reference signal	410
	X34	24 V for control-is-ready signal output	370
	X41	PLC output	418
	X42	PLC input	412
	X44	24 V PLC supply voltage	366
	X45	Control panel	445
	X46	Machine operating panel	438
	X47	PLC expansion	424
	X48	PLC analog input	388
	X149	BF 150 display unit	447
	X131	Reserved	–
	X69, X169	Power supply	354
	X121, X125	Reserved	–
	X165	Reserved	–
	X74	5 V power supply	355
	X150, X151	Axis-specific drive enabling	373
	B	Signal ground	–
	⏚	Ground lead (YL/GN)	–

MC 420 and CC 424 with 6 control loops

Connection overview	Connector	Function	Page
	X1 to X5	Position encoder	376
	X15 to X20	Speed encoder	378
	X51 to X56	PWM output	386
	X8	Nominal value output, analog	391
	X12	TS touch-trigger probe	395
	X13	TT touch-trigger probe	397
	X23	Handwheel	404
	X26	Ethernet data interface	400
	X27	RS-232-C/V.24 data interface	401
	X28	RS-422/V.11 data interface	403
	X141	USB Interface	461
	X30	24 V spindle reference signal	410
	X34	24 V for control-is-ready signal output	370
	X41	PLC output	418
	X42	PLC input	412
	X44	24 V PLC supply voltage	366
	X45	Control panel	445
	X46	Machine operating panel	438
	X147	PLC expansion	424
	X48	PLC analog input	388
	X149	BF 150 display unit	447
	X69	Power supply	354
	X10	Reserved	–
	X121	Reserved	–
	X165, X166	Reserved	–
	X74	5 V power supply	355
	X150	Axis-specific drive enabling (at bottom of housing)	373
	X142	USB Interface (at bottom of housing)	461
	B	Signal ground	–
	⏚	Ground lead (YL/GN)	–

3.5 Power Supply for the iTNC 530

The UV 1x0 or UVR 1x0 power supply units provide the iTNC 530 with power. Power is supplied through X69/X169 and possibly in addition through the 5 V terminal on the CC 42x.

The MC 422C (DP) always requires the additional 5 V supply via X74 on the CC 42x.

The control monitors the 5 V supply voltage. If it drops below 4.75 V, the error message 5 V power supply too low appears. If it rises above 5.4 V, 5 V power supply too high is indicated.

For information on the UV 1x0 or UVR 1x0 supply units, such as the ampacity of the +5 V supply voltage, refer to the "Inverter Systems and Motors" Technical Manual.

The ampacity of the supply unit may not be exceeded! If the supply unit does not suffice, then an additional UV 105 must be used. The following table shows the current consumption of the respective devices:

Device	Current consumption of the 5 V supply
MC 420	4.8 A
MC 422C	5.5 A
MC 422C (with 2 processors)	11.0 A
USB components	Max. 2 x 0.5 A[a]
CC 422 / 6 control loops	1.50 A
CC 422 / 10 or 12 control loops	3.00 A
CC 424(B) / 6 control loops	2.50 A
CC 424(B) / 10 control loops	4.60 A
LS, LB	0.15 A
ERN, ROD, RON	0.20 A
Absolute rotary encoders	0.25 A (+0.085 A with line drop compensator)[b]
Absolute angle encoders	0.35 A (+0.085 A with line drop compensator)[b]
LC	0.30 A (+0.085 A with line drop compensator)[b]

a. If USB components require more than 0.5 A, a separate power supply becomes necessary for these components. One possibility is the USB hub from HEIDENHAIN (368 735-01).

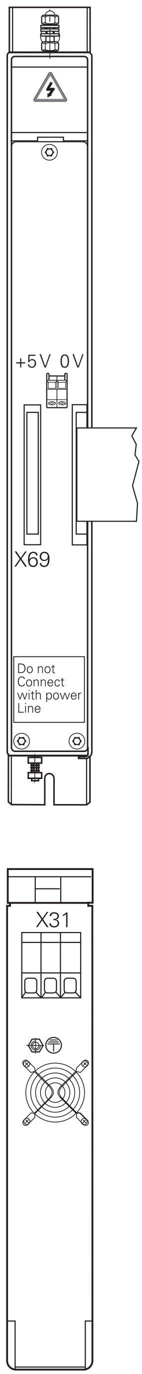
b. For cable lengths > 10 m between the logic unit and the encoders with EnDat interfaces, a line drop compensator is required (efficiency = 75%).

Example:

Device	Current consumption
MC 422C	5.50 A
CC 422 / 6 control loops	1.50 A
3 x LS for X, Y, Z	0.45 A
1 x ROD for C	0.2 A
4 x ERN for X, Y, Z, C, spindle	1.0 A
Total	8.65 A > 8.50 A

If a UV 120 supply unit is used, then for this application a UV 105 is necessary as well.

3.5.1 UV 105 power supply unit

Connection overview	Connector	Function
	Conductor bar	Power supply over U_Z
	X74	5 V power supply for CC 42x
	Free ribbon cable	Supply for CC 42x (to X69 on CC 42x)
	X69	Status signals from UV 1x0
	X31	Power supply

Attention

Do not engage or disengage any connecting elements while the unit is under power!

X69, X169: NC supply voltage and control signals

Note

For the NC to be able to evaluate the status signals of the power supply units, connector X69 of the CC 4xx must be connected by ribbon cable with X69 of the UV 105.
Since non-HEIDENHAIN inverters do not send any status signals, an adapter connector (ID 349 211-01) must be connected to X69 on the UV 105. This connector is delivered with the UV 105.

Pin layout:

50-pin ribbon connector	Assignment	50-pin ribbon connector	Pin layout
1a to 5b	+5 V	16b	GND
6a to 7b	+12 V	17a	RDY.PS
8a	+5 V (low-voltage separation)	17b	GND
8b	0 V (low-voltage separation)	18a	$\overline{\text{ERR.ILEAK}}$
9a	+15 V	18b	GND
9b	–15 V	19a	$\overline{\text{PF.PS.AC}}$ (only UV 120, UV 140, UV 150, UR 2xx)
10a	UZAN	19b	GND
10b	0 V	20a	Do not assign
11a	IZAN	20b	GND
11b	0 V	21a	Do not assign
12a	$\overline{\text{RES.PS}}$	21b	GND
12b	0 V	22a	Do not assign
13a	$\overline{\text{PF.PS.ZK}}$	22b	GND
13b	GND	23a	Reserved (SDA)
14a	$\overline{\text{ERR.UZ.GR}}$	23b	GND
14b	GND	24a	Reserved (SLC)
15a	$\overline{\text{ERR.IZ.GR}}$	24b	GND
15b	GND	25a	RES.LE
16a	$\overline{\text{ERR.TMP}}$	25b	GND

X74: 5 V power supply

Pin layout:

Wire color of 5 V connection	5 V terminal on CC 42x
Black	0 V
Red	+5 V

X31: Supply voltage for UV 105

Supply voltage: 400 V ± 10%

Pin layout:

Connecting terminal	Pin layout
U	U[a]
V	V[a]
⏚	Ground lead (YL/GN), ≥ 10 mm^2

a. Connecting cable 1.5 mm^2

Note

The supply voltage at terminals U and V must:

- be supplied via an isolating transformer (300 VA, basic insulation in accordance with EN 61800-5-1 or VDE 0550) for non-HEIDENHAIN inverters and regenerative HEIDENHAIN inverter systems (UV 120, UV 140, UV 150, UR 2xx).
- There is no need for an isolating transformer if non-regenerative HEIDENHAIN inverter systems are used.

Attention

When using an isolating transformer, do not ground this isolating transformer on the secondary side!

The isolating transformer decouples the dc-link voltage from ground. Grounding the isolating transformer on the secondary side leads to an addition of the dc-link voltage and the supply voltage. This overloads the UV 105, thereby destroying it!

Please keep this in mind in your circuit diagrams.

Power supply of the UV 105 with U_Z

The UV 105 is powered with dc-link voltage U_Z through

- the conductor bars (for HEIDENHAIN inverter systems).
- a cable which is connected instead of the conductor bar (for non-HEIDENHAIN inverter systems).

The dc-link voltage is monitored by the control (See "Monitoring of the Power Supply Unit" on page 906).

3.5.2 UV 105B (non-HEIDENHAIN inverter systems)

General information

The **UV 105B** (ID 532 556-01) was designed solely for the use of HEIDENHAIN controls in connection with non-HEIDENHAIN inverter systems. It is essential for the supply voltages of the HEIDENHAIN control units.

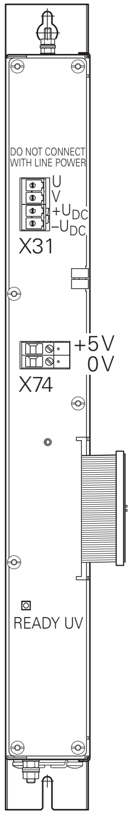
UV 105B power supply unit for the operation of HEIDENHAIN controls with non-HEIDENHAIN inverter systems ID 532 556-01 UV 105B	

Specifications

Specifications	UV 105B
Power supply (at X31)	400 V~ ± 10 % 50 Hz
Protection	6.3 A / gRL
Load capacity (5 V)	20 A
Power consumption	Max. 200 W
Degree of protection	IP 20
Module width	159 mm
Weight	3 kg
ID number	532 556-01

Attention

The **UV 105B** is not compatible with the **UV 105** (ID 344 980-xx), and no HEIDENHAIN inverter components can be operated with this supply voltage.



Connection overview for UV 105B	Connector	Function
	X31, U/V	400 V power supply
	X31 $+U_Z/-U_Z$	Power supply from dc-link
	X74	5 V power supply for CC 42x
	Free ribbon cable	Additional power supply and status signals for CC 42x (to X69 on CC 42x)
	Green LED	Operational status indicator for UV 105B
	⏚	Ground lead (YL/GN)

Attention

Do not engage or disengage any connecting elements while the unit is under power!

Status signals via ribbon cable

> Note
>
> For the control to be able to evaluate the status signals of the power supply unit, the ribbon cable of the UV 105B must be connected with X69 of the control.

Connection:

50-pin ribbon connector	Pin layout	50-pin ribbon connector	Pin layout
1a to 5b	+5 V	16b	GND
6a to 7b	+12 V	17a	RDY.PS
8a	+5 V	17b	GND
8b	+5 V	18a	$\overline{\text{ERR.ILEAK}}$
9a	+15 V	18b	GND
9b	–15 V	19a	$\overline{\text{PF.PS.AC}}$ (only UV 120, UV 140, UV 150, UR 2xx)
10a	UZAN	19b	GND
10b	0 V	20a	Do not assign
11a	IZAN	20b	GND
11b	0 V	21a	Do not assign
12a	$\overline{\text{RES.PS}}$	21b	GND
12b	0 V	22a	Do not assign
13a	$\overline{\text{PF.PS.ZK}}$	22b	GND
13b	GND	23a	Reserved (SDA)
14a	$\overline{\text{ERR.UZ.GR}}$	23b	GND
14b	GND	24a	Reserved (SLC)
15a	$\overline{\text{ERR.IZ.GR}}$	24b	GND
15b	GND	25a	RES.LE
16a	$\overline{\text{ERR.TMP}}$	25b	GND

X74: 5 V connection of the UV 105B

Connection:

Wire color of 5 V connection	5 V terminal on CC 42x
Black	0 V
Red	+5 V

X31: Supply voltage for UV 105B

Supply voltage: 400 V ± 10%

Connection:

Connecting terminal	Pin layout
U	Phase 1 / 400 V~ ±10% / 50 Hz to 60 Hz
V	Phase 2 / 400 V~ ±10% / 50 Hz to 60 Hz
⏚	Ground lead (YL/GY), ≥ 10 mm^2
	Cable: Wire cross section: 1.5 mm^2 (AWG 16) Line fuse: 6.3 A (gRL) Siemens Sitor type
$+U_Z$	Positive dc-link voltage of the non-HEIDENHAIN inverter system
$-U_Z$	Negative or reference potential of the dc-link voltage of the non-HEIDENHAIN inverter system
	Cable: Wire cross section: 1.5 mm^2 (AWG 16) The dc-link connection of the UV 105B is protected by the additional PCB on the non-HEIDENHAIN inverter system (4 A)
Tightening torque: for the connecting terminals 0.6 Nm (6.5 to 7 psi) **Grounding terminal:** ≥ 10 mm^2 (AWG 6) **Strain relief:** Ensure that the connecting cables are not subject to excessive strain	

Note

- If you are using non-HEIDENHAIN inverter systems, you must connect the supply voltage to the terminals U and V via an isolating transformer (300 VA, basic insulation as per EN 50 178 or VDE 0550).

Attention

When using an isolating transformer, do not ground this isolating transformer on the secondary side!

The isolating transformer decouples the line voltage from ground. Grounding the isolating transformer on the secondary side leads to an addition of the dc-link voltage and the supply voltage. This could destroy the UV 105B!

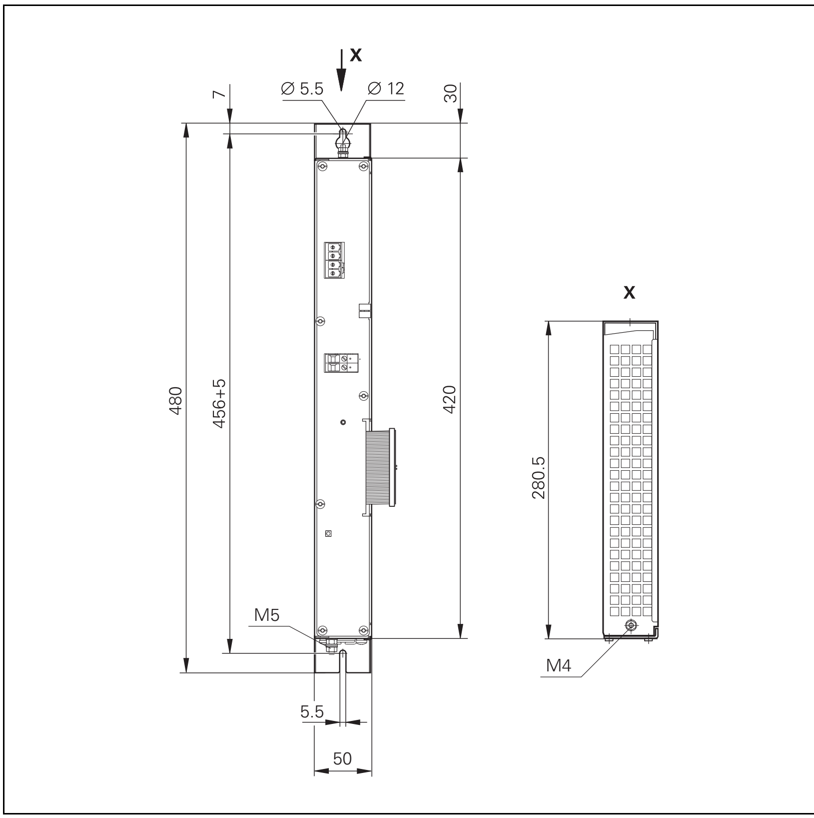
Please keep this in mind in your circuit diagrams.

U_Z: Supply of the UV 105B with U_Z

Since the power to the UV 105B is supplied through the dc-link, the voltage fed into the dc-link by the motors that are still running can be used during line voltage failures. The UV 105B uses this voltage to maintain the power supply to the control until the non-HEIDENHAIN inverter system has been shut down properly by the control.

Connecting terminals	Pin layout
$-U_Z$	DC-link voltage –
$+U_Z$	DC-link voltage +

Mating dimensions of UV 105B



3.5.3 UV106B power supply unit

General information

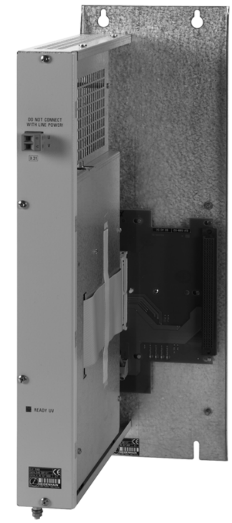
UV 106B power supply unit for analog HEIDENHAIN contouring controls

The **UV 106B** power supply unit was designed so that the iTNC 530 could be used with a compact, coordinated system for analog nominal shaft-speed interfaces (+/–10 V).

It supplies the iTNC 530 with the supply voltages necessary for operation.

The **UV 106B** (ID 546 581-01) is being introduced as a replacement for the **UV 106** (ID 366 572-11).

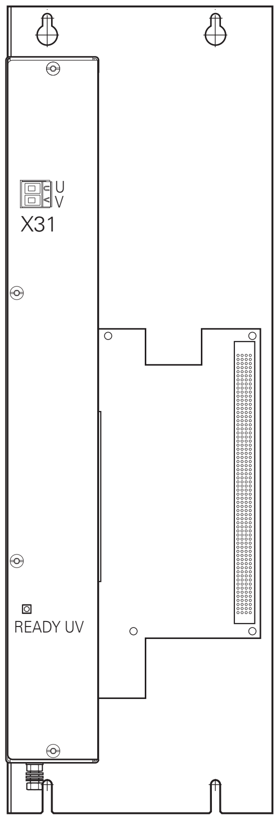
ID 546 581-01 UV 106B



Specifications

Specifications	**UV 106B**
Power supply (at X31)	400 V~ ± 10%[a] 50 Hz
Protection	6.3 A (gR) Siemens Sitor type or 6.3 A (gRL) Siba type
Load capacity (5 V)	20 A
Power consumption	Max. 400 W
Degree of protection	IP 20
Module width	159 mm
Weight	4 kg
ID number	546 581-xx

a. An isolating transformer is not necessary when the UV 106B is connected.

Connection overview for UV 106B	Connector	Function
		
	X31, U/V	400 V power supply
	Green LED	Operational status indicator for UV 105B
	⏚	Ground lead (YL/GN)

Attention

Do not engage or disengage any connecting elements while the unit is under power!

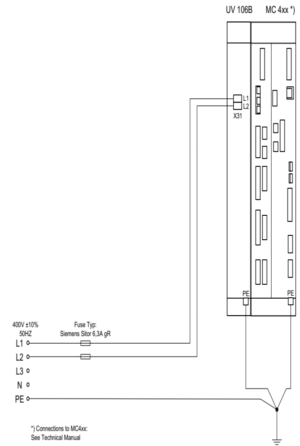
X31: Supply voltage for UV 106B

Supply voltage: 400 V ± 10%

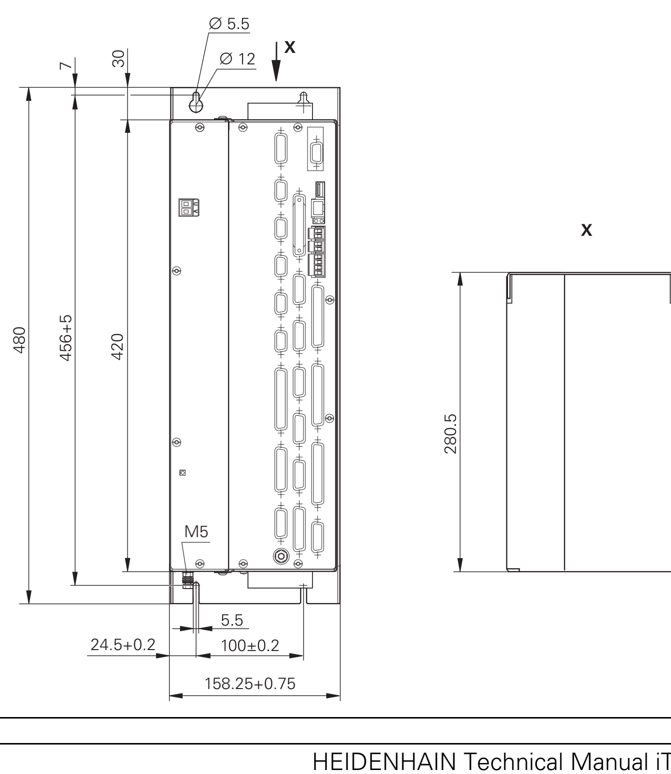
Connection:

<table>
<tr><th>Connecting terminal</th><th>Pin layout</th></tr>
<tr><td>U</td><td>Phase 1 / 400 V~ ±10% / 50 Hz to 60 Hz</td></tr>
<tr><td>V</td><td>Phase 2 / 400 V~ ±10% / 50 Hz to 60 Hz</td></tr>
<tr><td>⏚</td><td>Ground lead (YL/GN),
≥ 10 mm²</td></tr>
<tr><td></td><td>Connecting leads
Wire cross section: 1.5 mm² (AWG 16)</td></tr>
<tr><td colspan="2">Tightening torque:
for the connecting terminals
0.7 Nm (6.5 to 7 psi)
Grounding terminal:
≥ 10 mm² (AWG 6)
Strain relief:
Ensure that the connecting cables are not subject to excessive strain</td></tr>
</table>

Power connection



Mating dimensions of UV 106B



3.6 Power Supply for PLC Outputs

The PLC of the iTNC 530 as well as the PL 410 B/PL 405 B/PL 51x are powered by the 24 V– control voltage of the machine (in accordance with VDE 0551).

The control voltage must be smoothed with a minimum 1000 µF at a rated current capacity of 150 µF/A. At a current load of 15 A, for example, this corresponds to a capacity of 2250 µF.

According to EN 61 131-2:1994, the following is valid for the PL 410 B/ PL 405 B:

- 5% alternating voltage component is permissible
- Minimum absolute value: 20.4 V–
- Maximum absolute value: 28.8 V–

According to EN 61 131-2:1994, the following is valid for the PL 51x:

- 5% alternating voltage component is permissible
- Minimum absolute value: 20.4 V–
- Maximum absolute value: 25.4 V– at 200 W power output
- Maximum absolute value: 28.8 V– at 100 W power output

Danger

Use only original replacement fuses.

Power consumption

If half of the outputs are switched at the same time, the following are the values for power consumption:

MC 42x(B,C):	48 W
PL 410 B:	Approx. 460 W
PL 405 B:	Approx. 235 W
PL 510:	Approx. 485 W

Nominal operating current per output

MC 42x(B,C):	0.150 A
PL 410 B:	2 A (with max. current consumption of 20 A)
PL 405 B:	2 A (with max. current consumption of 20 A)
PLD 16-8:	2 A (only as of version 360 916-11) Simultaneity with a supply voltage of 25.4 V: 4 outputs with 2 A each 8 outputs with 1 A each Total current: Out0 to Out7: ≤ 8 A Out0 to Out3: ≤ 4 A Out4 to Out7: ≤ 4 A

For the PLD 16-8 it must be remembered that a total current of 8 amperes maximum per slot (PLD) must not be exceeded!

X44: PLC supply voltage

Pin layout on the MC 422(B,C):

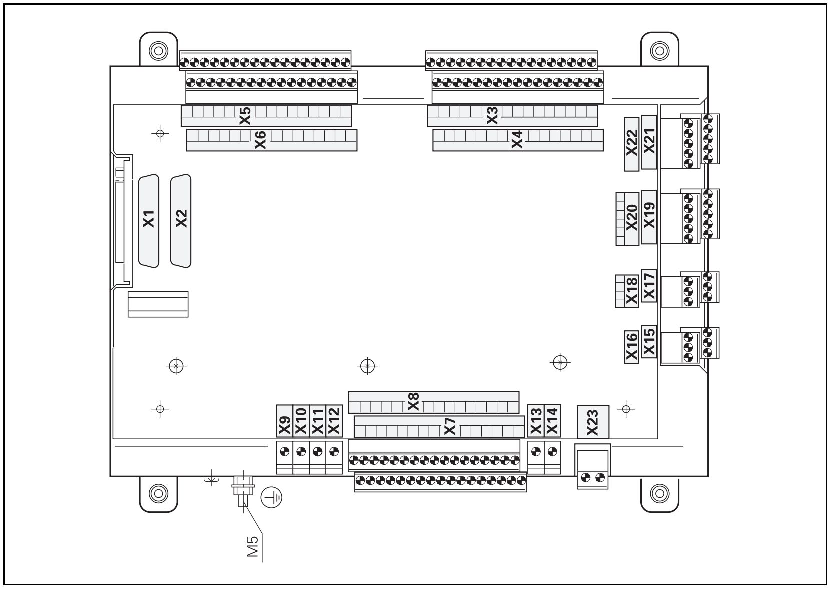
Connection terminal	Assignment	PLC outputs
1	+24 V cannot be switched off with EMGERGENCY STOP	O24 to O30
2	+24 V disconnectable with EMERGENCY STOP	O16 to O23
3		O0 to O15
4	0 V	

Note

If the +24 V power supply (which cannot be shut off via emergency stop) is missing at X44, the error message **`Supply voltage missing at X44`** appears.

3.6.1 Power supply for PL 4xx B

Connection overview



X9 to X14: Supply voltage

Pin layout on the PL 410 B:

Terminal	Pin layout	PL 1	PL 2	PL 3	PL 4
X9	0 V				
X10	+24 V– logic power supply and for control-is-ready signal[a]				
X11	+24 V– power supply for outputs[a]	O32 – O39	O64 – O71	O128 – O135	O160 – O167
X12	+24 V– power supply for outputs[a]	O40 – O47	O72 – O79	O136 – O143	O168 – O175
X13	+24 V– power supply for outputs[a]	O48 – O55	O80 – O87	O144 – O151	O176 – O183
X14	+24 V– power supply for outputs[a]	O56 – O62	O88 – O94	O152 – O158	O184 – O190

a. 20.4 V to 28.8 V

Pin layout on the PL 405 B:

Terminal	Pin layout	PL 1	PL 2	PL 3	PL 4
X9	0 V				
X10	+24 V– logic power supply and for control-is-ready signal[a]				
X13	+24 V– power supply for outputs[a]	O48 – O55	O80 – O87	O144 – O151	O176 – O183
X14	+24 V– power supply for outputs[a]	O56 – O62	O88 – O94	O152 – O158	O184 – O190

a. 20.4 V to 28.8 V

X23: Power supply for the analog inputs on the PL 410 B

The PL 410 B input/output unit is also available with additional analog inputs and inputs for PT 100 thermistors.

Danger

The power supply must correspond to a safety extra-low voltage (SELV) as per EN 61800-5-1.

Terminal	Pin layout
1	+24 V– (safety extra-low voltage as per EN 61800-5-1)
2	+0 V

3.6.2 Power supply for PL 51x

Note

The iTNC 530 cyclically monitors the supply voltage of the PL 51x.

PLB 51x basic module

Pin layout for X3 (power supply for logic circuit):

Terminal	Pin layout
1	+24 V– (20.4 V to 28.8 V)
2	+0 V

PLD 16-8 input/ output module

Pin layout at X6 (power supply for PLC outputs):

Terminal	Pin layout
9	+24 V– (20.4 V to 28.8 V) for group 1
10	+24 V– (20.4 V to 28.8 V) for group 2

3.7 Power Supply for Control-Is-Ready Signal

X34: Power supply for control-is-ready signal

The control-is-ready signal output is powered by 24 V– provided by the UE 2xx B inverter or the UV1xx power supply unit. The voltage is connected with terminal X34.

Pin layout:

Connecting terminal X34	Assignment	Connection when using a HEIDENHAIN inverter
1	+24 V	X72/1
2	0 V	X72/2

3.8 Power Supply for the Display Units

Connecting terminal X1	Assignment
1	+24 V
2	0 V

Power consumption: BF 120: 15 W
BF 150: 45 W

Danger

The power supply must have basic insulation as per EN 61800-5-1.

3.9 Buffer Battery

Note

Make a data backup before changing the buffer battery.

Danger

When exchanging the buffer battery, remember:

- Switch off the machine and the iTNC 530.
- The buffer battery may be exchanged only by trained personnel.

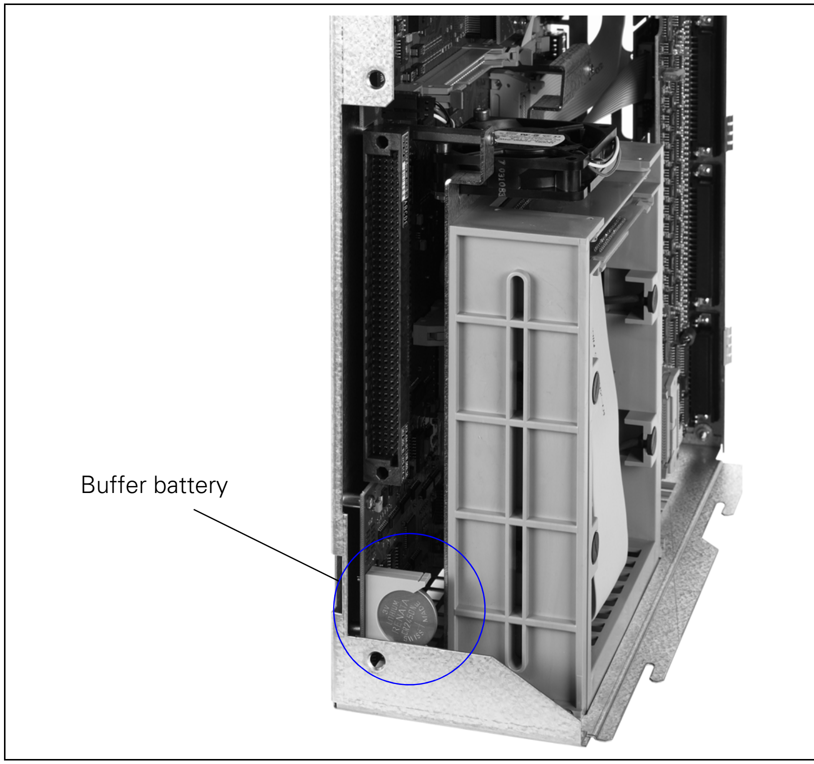
Battery type: 1 lithium battery, type CR 2450N (Renata), ID 315 878-01

The lifetime of the buffer battery is between 2 and 5 years.

If the voltage of the buffer battery falls below 2.6 V the error message **Exchange buffer battery** appears. If the voltage does not exceed 2.6 V, the error message is reactivated after 30 minutes.

Notes on exchanging the buffer battery:

- The buffer battery is located on the rear side of the MC 42x(B,C).
- Exchange the battery; the new battery can be inserted in only one position.



The following information is stored in the battery-buffered memory:

- Non-volatile PLC operands
- Most recent log entries
- Information about the trace function
- Information about program interruption
- Information from absolute encoders with EnDat interface
- Information about the boot process
- Information about errors

3.10 Drive Controller Enable

A drive controller can be enabled by the NC software only if the controller is enabled with 24 V on X150/X151 **and** on X42, pin 33.

X150, X151: Drive controller enabling for axis groups

The connecting terminals X150 and X151 are located on the bottom of the CC 42x.

- X150 controls drive enabling for the axis groups on the first controller board (PWM outputs X51 to X56).
- X151 controls drive enabling for the axis groups on the second controller board (PWM outputs X57 to X60 or X62).

Note

The pin of an axis group must always be wired to the connector on whose PCB the control loop is located.
If an axis group contains control loops located on both PCBs, then the pins of both connectors must be wired.

Pin layout:

Terminal X150/X151	Assignment of X150	Assignment of X151
1	+24 V[a]; drive controller enabling for axis group 1	+24 V[a]; drive controller enabling for axis group 1
2	+24 V[a]; drive controller enabling for axis group 2	+24 V[a]; drive controller enabling for axis group 2
3	+24 V[a]; drive controller enabling for axis group 3	+24 V[a]; drive controller enabling for axis group 3
4	Only CC 424(B): +24 V[a]; drive controller enabling for axis group 4	Only CC 424(B): +24 V[a]; drive controller enabling for axis group 4
5	Only CC 424(B): +24 V[a]; drive controller enabling for axis group 5	Only CC 424(B): +24 V[a]; drive controller enabling for axis group 5
6	Only CC 424(B): +24 V[a]; drive controller enabling for axis group 6	Only CC 424(B): +24 V[a]; drive controller enabling for axis group 6
7	Reserved, do not assign	Reserved, do not assign
8	Reserved, do not assign	Reserved, do not assign
9	0 V	0 V

a. maximum current consumption 10 mA

X42/33: Global drive controller enabling

Pin layout:

D-sub connctn. (male) 37-pin	Assignment
..	..
33	+24 V (drive controller enabling)
..	..

3.11 Encoder Connections

3.11.1 General information

HEIDENHAIN contouring controls are designed for use with incremental or absolute linear and angular encoders as measuring systems. The encoder signals are subdivided 1024-fold.
Encoders with one reference mark or distance-coded reference marks and with EnDat interface are permissible.
HEIDENHAIN recommends the use of absolute encoders with EnDat interface or the use of encoders with distance-coded reference marks because they greatly reduce the traverse distance required to establish the absolute position.

Please use only HEIDENHAIN encoder cables, connectors and couplings. For maximum cable lengths, see "Cable Overview" at the end of this chapter.

	Position encoder	**Speed encoder**
Signal amplitude	EnDat, 1 V_{PP}	EnDat, 1 V_{PP}
Input frequency 1 V_{PP}	■ MC 42x (B): 27 kHz/400 kHz (can be set via MP115.x) ■ CC 424(B): 27 kHz/400 kHz (can be set via MP116.x)	CC 422: 350 kHz CC 424(B): 400 kHz

11 μA_{PP} encoders can be connected through the adapter plug with the ID 317 505-05.

Note

Keep in mind the line count of the speed encoders when choosing the motors:

$$x = \frac{f \cdot 60 \cdot 1000}{n}$$

x:line count of the speed encoder
f:maximum input frequency
n:maximum speed

Example:
f = 350 kHz; n = 10 000 min^{-1}

$$x = \frac{350 \cdot 60 \cdot 1000}{10000} \approx 2048$$

3.11.2 Input of position encoder

X1 to X6, X35 to X38, X201 to X214: Position encoder 1 V_{PP}

Pin layout:

MC 42x(B,C), CC 424(B)		Adapter cable 309 783-xx Adapter cable 310 199-xx			Encoder	
Male	**Assignment**	**Female**	**Color**	**Female**	**Male**	**Color**
1	+5 V (U_P)	1	Brown/Green	12	12	Brown/Green
2	0 V (U_N)	2	White/Green	10	10	White/Green
3	A+	3	Brown	5	5	Brown
4	A–	4	Green	6	6	Green
5	Do not assign	5				
6	B+	6	Gray	8	8	Gray
7	B–	7	Pink	1	1	Pink
8	Do not assign	8				
9	+5 V (sensor)	9	Blue	2	2	Blue
10	R+	10	Red	3	3	Red
11	0 V (sensor)	11	White	11	11	White
12	R–	12	Black	4	4	Black
13	0 V	13				
14	Do not assign	14	Violet	7	7	Violet
15	Do not assign	15				
Hsg.	Ext. shield	Hsg.	Ext. shield	Hsg.	Hsg.	Ext. shield

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

X1 to X6, X35 to X38, X201 to X214: Position encoder with EnDat interface

Pin layout:

MC 42x(B,C), CC 424(B)		Adapter cable 332 115-xx			Connecting cable 323 897-xx				Adapter cable 313 791-xx		
Male	**Assignment**	**Female**	**Color**	**Female**	**Male**	**Color**	**Fem.**		**Male**	**Color**	**Fem.**
1	+5 V (U_P)	1	Brown/ Green	7	7	Brown/ Green	7	Line drop compensator ID 336 697-02, if required	7	Brown/ Green	5b
2	0 V (U_N)	2	White/ Green	10	10	White/ Green	10		10	White/ Green	6a
3	A+	3	Green/ Black	15	15	Green/ Black	15		15	Green/ Black	2a
4	A–	4	Yellow/ Black	16	16	Yellow/ Black	16		16	Yellow/ Black	2b
5	Data	5	Gray	14	14	Gray	14		14	Gray	3b
6	B+	6	Blue/ Black	12	12	Blue/ Black	12		12	Blue/ Black	1a
7	B–	7	Red/ Black	13	13	Red/ Black	13		13	Red/ Black	1b
8	$\overline{\text{Data}}$	8	Pink	17	17	Pink	17		17	Pink	3a
9	+5 V (sensor)	9	Blue	1	1	Blue	1		1	Blue	5a
10	Vacant	10		3	3	Red	3		3		
11	0 V (sensor)	11	White	4	4	White	4		4	White	6b
12	Vacant	12		2	2	Black	2		2		
13	Internal shield	13	Internal shield	11	11	Internal shield	11		11	Internal shield	
14	Clock	14	Violet	8	8	Violet	8		8	Violet	4a
15	$\overline{\text{Clock}}$	15	Yellow	9	9	Yellow	9		9	Yellow	4b
Hsg.	Housing	Hsg.	External shield	Hsg.		External shield			Hsg.	External shield	

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

3.11.3 Input of speed encoder

> **Attention**
>
> If you connect angle or linear encoders from HEIDENHAIN to the speed encoders (such as for torque motors), you must pay attention to the different connector layouts!
> HEIDENHAIN offers special cables and line-drop compensators for such applications. More information is in the Cable Overviews.

X15 to X20, X80 to X87: Speed encoder 1 V_{PP}

Pin layout:

CC 42x		Adapter cable 289 440-xx				Connecting cable 336 847-xx		
Male	**Assignment**	**Female**	**Color**	**Female**		**Male**	**Color**	**Female**
1	+5 V (U_P)	1	Brown/Green	10	Line drop compensator ID 370 226-01, if required	10	Brown/Green	10
2	0 V (U_N)	2	White/Green	7		7	White/Green	7
3	A+	3	Green/Black	1		1	Green/Black	1
4	A–	4	Yellow/Black	2		2	Yellow/Black	2
5	0 V							
6	B+	6	Blue/Black	11		11	Blue/Black	11
7	B–	7	Red/Black	12		12	Red/Black	12
8	0 V	8	Internal shield	17		17	Internal shield	17
9	Do not assign							
10	Do not assign							
11	Do not assign							
12	Do not assign							
13	Temperature +	13	Yellow	8		8	Yellow	8
14	+5 V (sensor)	14	Blue	16		16	Blue	16
15	Do not assign							
16	0 V (sensor)	16	White	15		15	White	15
17	R+	17	Red	3		3	Red	3
18	R–	18	Black	13		13	Black	13
19	C+	19	Green	5		5	Green	5
20	C–	20	Brown	6		6	Brown	6
21	D+	21	Gray	14		14	Gray	14
22	D–	22	Pink	4		4	Pink	4
23	Do not assign							
24	0 V							
25	Temperature –	25	Violet	9		9	Violet	9
Hsg.	Housing	Hsg.	Ext. shield	Hsg.		Hsg.	Ext. shield	Hsg.

> **Note**
>
> The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

X15 to X20, X80 to X85: Speed encoder with EnDat interface

Pin layout:

CC 42x		Adapter cable 336 376-xx				Connecting cable 340 302-xx		
Male	Assignment	Female	Color	Female		Male	Color	Female
1	+5 V (U_P)	1	Brown/Green	10	Line drop compensator ID 370 224-01, if required	10	Brown/Green	10
2	0 V (U_N)	2	White/Green	7		7	White/Green	7
3	A+	3	Green/Black	1		1	Green/Black	1
4	A–	4	Yellow/Black	2		2	Yellow/Black	2
5	0 V							
6	B+	6	Blue/Black	11		11	Blue/Black	11
7	B–	7	Red/Black	12		12	Red/Black	12
8	0 V	8	Internal shield	17		17	Internal shield	17
9	Do not assign							
10	Clock	10	Green	5		5	Green	5
11	Do not assign							
12	$\overline{\text{Clock}}$	12	Brown	14		14	Brown	14
13	Temperature +	13	Yellow	8		8	Yellow	8
14	+5 V (sensor)	14	Blue	16		16	Blue	16
15	Data	15	Red	3		3	Red	3
16	0 V (sensor)	16	White	15		15	White	15
17	Do not assign							
18	Do not assign							
19	Do not assign							
20	Do not assign							
21	Do not assign							
22	Do not assign							
23	$\overline{\text{Data}}$	23	Black	13		13	Black	13
24	0 V							
25	Temperature –	25	Violet	9		9	Violet	9
Hsg.	Housing	Hsg.	Ext. shield	Hsg.		Hsg.	Ext. shield	Hsg.

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Danger

Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

Pin layout (for the LC or RCN):

CC 42x		Adapter cable 336 376-xx				Adapter cable 369 124-xx Adapter cable 369 129-xx	
Male	**Assignment**	**Female**	**Color**	**Female**		**Male**	**Color**
1	+5 V (U_P)	1	Brown/Green	10	Line drop compensator ID 368 210-02, if required	7	Brown/Green
2	0 V (U_N)	2	White/Green	7		10	White/Green
3	A+	3	Green/Black	1		15	Green/Black
4	A–	4	Yellow/Black	2		16	Yellow/Black
5	0 V						
6	B+	6	Blue/Black	11		12	Blue/Black
7	B–	7	Red/Black	12		13	Red/Black
8	0 V	8	Internal shield	17		11	Internal shield
9	Do not assign						
10	Clock	10	Green	5		8	Violet
11	Do not assign						
12	$\overline{\text{Clock}}$	12	Brown	14		9	Yellow
13	Temperature +	13	Yellow	8			
14	+5 V (sensor)	14	Blue	16		1	Blue
15	Data	15	Red	3		14	Gray
16	0 V (sensor)	16	White	15		4	White
17	Do not assign						
18	Do not assign						
19	Do not assign						
20	Do not assign						
21	Do not assign						
22	Do not assign						
23	$\overline{\text{Data}}$	23	Black	13		17	Pink
24	0 V						
25	Temperature –	25	Violet	9			
Hsg.	Housing	Hsg.	Ext. shield	Hsg.		Hsg.	Ext. shield
					1		
					2 Temperature+		
					3 Temperature–		
					4		

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Pin layout (for the LC or RCN):

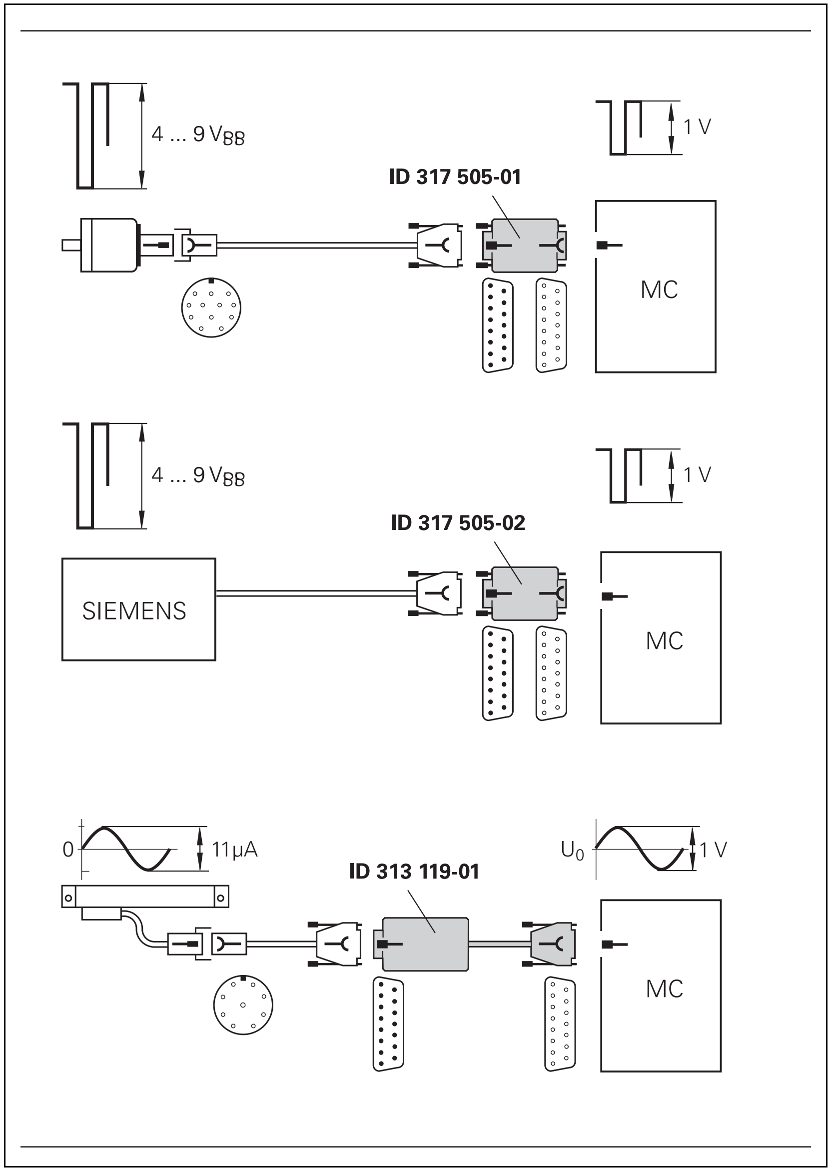
CC 42x		Adapter cable 509 667-xx			Adapter cable 369 124-xx Adapter cable 369 129-xx or RCN	
Male	**Assignment**	**Female**	**Color**	**Female**	**Male**	**Color**
1	+5 V (U_P)	1	Brown/Green	7	7	Brown/Green
2	0 V (U_N)	2	White/Green	10	10	White/Green
3	A+	3	Green/Black	15	15	Green/Black
4	A–	4	Yellow/Black	16	16	Yellow/Black
5	0 V					
6	B+	6	Blue/Black	12	12	Blue/Black
7	B–	7	Red/Black	13	13	Red/Black
8	0 V	8	Internal shield	11	11	Internal shield
9	Do not assign					
10	Clock	10	Green	8	8	Violet
11	Do not assign					
12	$\overline{\text{Clock}}$	12	Brown	9	9	Yellow
13	Temperature +	13	Yellow	5		
14	+5 V (sensor)	14	Blue	1	1	Blue
15	Data	15	Red	14	14	Gray
16	0 V (sensor)	16	White	4	4	White
17	Do not assign					
18	Do not assign					
19	Do not assign					
20	Do not assign					
21	Do not assign					
22	Do not assign					
23	$\overline{\text{Data}}$	23	Black	17	17	Pink
24	0 V					
25	Temperature –	25	Violet	6		
Hsg.	Housing	Hsg.	External shield	Hsg.	Hsg.	External shield

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

3.12 Adapters for Encoder Signals

Encoder signals with 11 µA$_{PP}$ or TTL level can be adapted to the 1 V$_{PP}$ interface with HEIDENHAIN adapter connectors.



Note

Please note:

- The adapters adjust only the levels, not the signal shape.
- The contamination signal of the square-wave encoder cannot be evaluated.
- A square-wave signal can be subdivided no more than 4-fold.
- If encoders with TTL signals are connected to a CC 424(B) via an adapter (ID 317 505-xx), then the phase position of the reference pulse is not correct. In order to deactivate monitoring, please set bit 9 of MP2220.

Adapter connector TTL (HEIDENHAIN)/ 1 V_{PP}

Pin layout of D-sub connector (female) and D-sub connector (male):

D-sub connector (female) 15-pin	Pin layout	D-sub connection (male) 15-pin	Pin layout
1	+5 V (U_P)	1	+5 V (U_P)
2	0 V (U_N)	2	0 V (U_N)
3	A+	3	U_{a1}
4	A–	4	$–U_{a1}$
5	0 V	5	0 V
6	B+	6	U_{a2}
7	B–	7	$–U_{a2}$
8	0 V	8	0 V
9	+5 V	9	+5 V
10	R+	10	U_{a0}
11	0 V	11	0 V
12	R–	12	$–U_{a0}$
13	0 V	13	0 V
14	$–U_{aS}$	14	$–U_{aS}$
15	Not assigned	15	Not assigned

Adapter connector TTL (SIEMENS) / 1 V_{PP}

Pin layout of D-sub connector (female) and D-sub connector (male):

D-sub connector (female) 15-pin	Pin layout	D-sub connection (male) 15-pin	Pin layout
1	Not assigned	1	Not assigned
2	0 V	2	0 V
3	A+	3	U_{a1}
4	A–	4	$–U_{a1}$
5	Not assigned	5	Not assigned
6	B+	6	U_{a2}
7	B–	7	$–U_{a2}$
8	Not assigned	8	Not assigned
9	Not assigned	9	Not assigned
10	R+	10	Not assigned
11	Not assigned	11	Not assigned
12	R–	12	U_{a0}
13	Not assigned	13	$–U_{a0}$
14	Not assigned	14	Not assigned
15	Not assigned	15	Not assigned

Adapter connector 11 µA_{PP} / 1 V_{PP}

Pin layout of D-sub connector (female) and D-sub connector (male):

D-sub connector (female) 15-pin	Pin layout	D-sub connection (male) 15-pin	Pin layout
1	+5 V (U_P)	1	+5 V (U_P)
2	0 V (U_N)	2	0 V (U_N)
3	A+	3	0°+
4	A–	4	0°–
5	0 V	5	0 V
6	B+	6	90°+
7	B–	7	90°–
8	0 V	8	0 V
9	+5 V	9	+5 V
10	R+	10	R+
11	0 V	11	0 V
12	R–	12	R–
13	0 V	13	0 V
14	Not assigned	14	Not assigned
15	Not assigned	15	Not assigned

3.13 Motor Power Stage Connection

The iTNC 530 is connected with HEIDENHAIN or non-HEIDENHAIN inverters through a PWM interface.
For a description of the HEIDENHAIN inverter systems, refer to the Technical Manual "Inverter Systems and Motors". The components required for operation of the iTNC 530 with non-HEIDENHAIN inverter systems are described in the manual "Technical Information for the Operation of SIMODRIVE and POWER DRIVE Inverter Systems".

The individual PWM outputs of the CC 422 are assigned to different controller groups (See "Maximum spindle speed" on page 656 and "PWM frequencies of the CC 422" on page 1012).

The following applies to the output signals to the power stage:

Logic level:5 V
Analog signals I_{ACT}: ±7.5 V
PWM frequency:MP2180.x can be used to set it at
3333 Hz, 4166 Hz, 5000 Hz, 6666 Hz,
8166 Hz and 10000 Hz

X51 to X64: PWM output

Pin layout:

Ribbon connector, 20-pin	Assignment
1a	PWM U1
1b	0 V U1
2a	PWM U2
2b	0 V U2
3a	PWM U3
3b	0 V U3
4a	$\overline{\text{SH2}}$
4b	0 V ($\overline{\text{SH2}}$)
5a	$\overline{\text{SH1}}$
5b	0 V ($\overline{\text{SH1}}$)
6a	+IIST 1
6b	–IIST 1
7a	0 V (analog)
7b	+IIST 2
8a	–IIST 2
8b	0 V (analog)
9a	$\overline{\text{BRK}}$
9b	Do not assign
10a	$\overline{\text{ERR}}$
10b	RDY

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

3.14 Analog Input

The MC 42x(B,C), the PL 410B PLC I/O unit, and the PLA 4-4 (for PL 51x) have analog inputs and inputs for Pt 100 thermistors.

The PL 410B is available with and without analog inputs.

	Analog inputs (±10 V)	Inputs for PT 100 thermistors
MC 42x(B,C), X48	3	3
PL 405 B	–	–
PL 410 B (263 371-02)	4	4
PLA 4-4 (PLB 51x)	4	4

Analog inputs

Voltage range:–10 V to +10 V
Input resistance: > 250 kΩ
Resolution (W480, W482, W484):100 mV
Resolution (Module 9003, 9138):10 mV (MC 42x(B,C))
100 mV (PL 410B)
10 mV (PLA 4-4)
Internal value range:–100 to +100, resolution approx. 100 mV
–1000 to +1000, resolution approx. 10 mV

Inputs for PT 100 thermistors

Constant current:5 mA
Temperature range: 0 °C to 100 °C
Resolution (W486, W488, W490):0.5 °C
Resolution (Module 9003, 9138): 0.1 °C (MC 42x(B,C))
0.5 °C (PL 410B)
0.01 °C (PLA 4-4), increment: 0.03°C
Internal value range:0 to 200, at a resolution of 0.5 °C
0 to 1000, at a resolution of 0.1 °C
0 to 10000, at a resolution of 0.01 °C

X48: Analog input (PLC) on the MC 42x(B,C)

Pin layout:

Attention

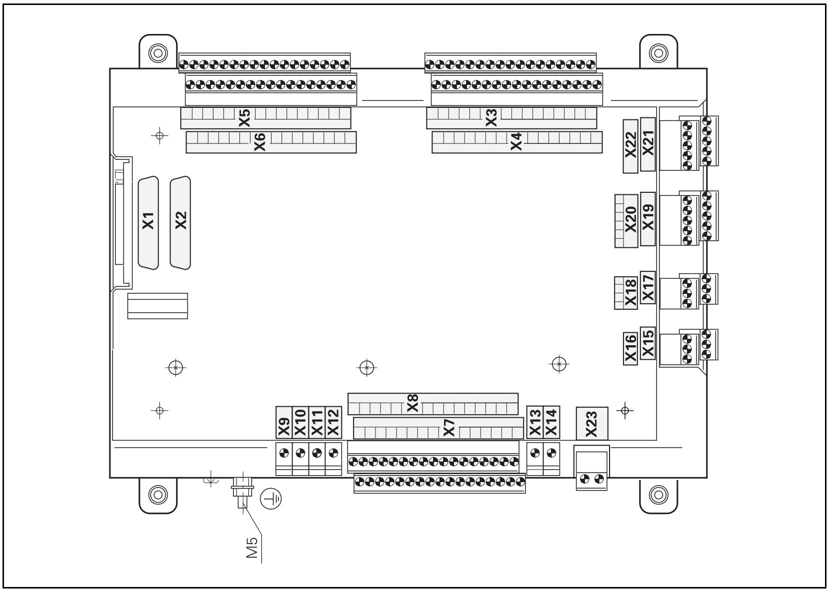
Remember to connect the analog inputs with the correct polarity!

D-sub connection (female) 25-pin	Assignment
1	I_1+ Constant current for Pt 100
2	I_1– Constant current for Pt 100
3	U_1+ Measuring input for Pt 100
4	U_1– Measuring input for Pt 100
5	I_2+ Constant current for Pt 100
6	I_2– Constant current for Pt 100
7	U_2+ Measuring input for Pt 100
8	U_2– Measuring input for Pt 100
9	I_3+ Constant current for Pt 100
10	I_3– Constant current for Pt 100
11	U_3+ Measuring input for Pt 100
12	U_3– Measuring input for Pt 100
13	Do not assign
14	Analog input 1: –10 V to +10 V
15	Analog input 1: 0 V (reference potential)
16	Analog input 2: –10 V to +10 V
17	Analog input 2: 0 V (reference potential)
18	Analog input 3: –10 V to +10 V
19	Analog input 3: 0 V (reference potential)
20 to 25	Do not assign
Housing	Ext. shield

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Connection overview for the PL 410B



X15 to X18: Analog input on the PL 410 B

Pin layout

Connecting terminals	Pin layout
1	–10 V to +10 V
2	0 V (reference potential)
3	Shield

Note

The interfaces comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

X19 to X22: Connection for Pt 100 on the PL 410 B

Pin layout:

Connecting terminals	Pin layout
1	I+ Constant current for PT 100
2	U+ Measuring input for PT 100
3	U– Measuring input for PT 100
4	I– Constant current for PT 100
5	Shield

Note

The interfaces comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

X15 to X18: Analog input on the PLA 4-4 analog module

Pin layout

Connecting terminals	Pin layout
1	–10 V to +10 V
2	0 V (reference potential)
3	Shield

Note

The interfaces comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

X19 to X22: Connection for Pt 100 on the PLA 4-4 analog module

Pin layout:

Connecting terminals	Pin layout
1	I+ Constant current for PT 100
2	U+ Measuring input for PT 100
3	U– Measuring input for Pt 100
4	I– Constant current for Pt 100
5	Shield

Note

The interfaces comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

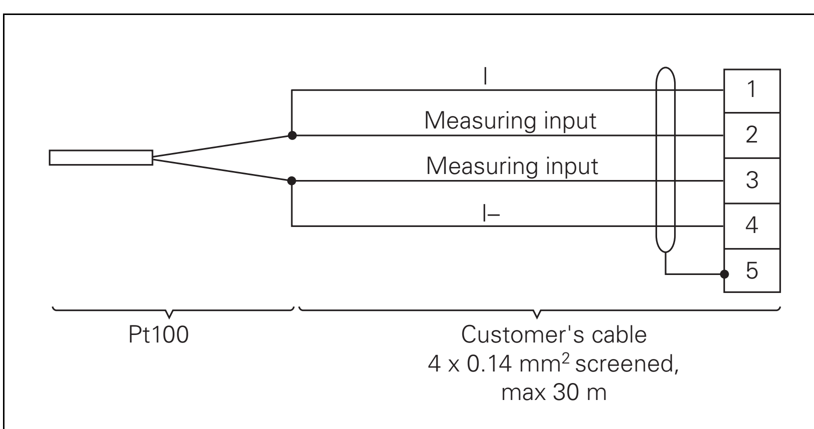
Connection of analog voltage

Characteristics of the connecting cable:

- Shielding
- 2 conductors with 0.14 mm²
- Maximum length: 50 m

Connection of the Pt100 thermistors

▶ Configure the thermistor connection as a "four-conductor circuit":



3.15 Analog Nominal Value Output

Output:±10 V
Maximum load of outputs:2 mA
Maximum capacitance:2 nF

13 analog outputs are available:

- Connection X8: Analog outputs 1 to 6
- Connection X9: Analog outputs 7 to 13

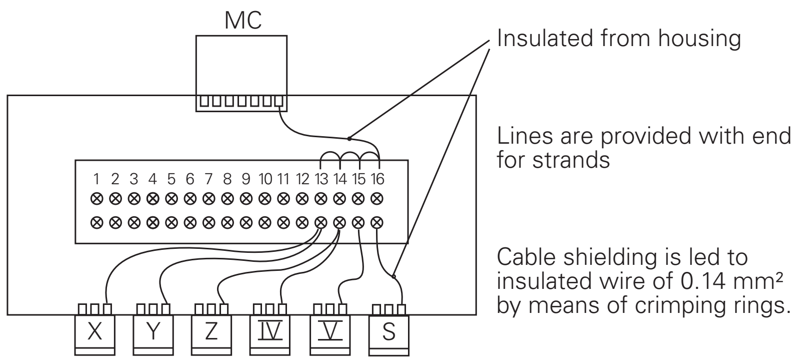
PLC analog output The PLC analog outputs can be controlled through Module 9130.

3.15.1 Nominal value output

Please note:

- For analog axes and analog spindle, use MP120.x and MP121.x to assign the corresponding analog outputs on terminal X8 or X9 to the nominal speed outputs.
- The connecting cables to the nominal value outputs must not have more than one intermediate terminal.
- If you must branch to physically separate servo inputs, the connection must be made in a grounded terminal box, e.g. ID 251 249-01 from HEIDENHAIN. The housing of the terminal box must be electrically connected with the frame of the machine.
- The 0 V connections of the nominal-value-difference inputs must be connected with the signal ground. Cross section ≥ 6 mm^2
- Use only original HEIDENHAIN cables and connecting elements.

The following wiring plan is suggested for shielding the terminal box:



Example for pin assignment in the terminal box:

Connecting terminals	Axis/Spindle	Assignment
1	Nominal value in X axis	±10 V
2		0 V
3	Nominal value in Y axis	±10 V
4		0 V
5	Nominal value in Z axis	±10 V
6		0 V
7	Nominal value in axis 4	±10 V
8		0 V
9	Nominal value in axis 5	±10 V
10		0 V
11	Spindle nominal value	±10 V
12		0 V
13 to 16	Shield connection	

X8: Analog outputs 1 to 6

For connecting cables, see "Cable Overview" at the end of this chapter.

Pin layout on the MC 42x(B,C) and connecting cables:

MC 42x(B,C)		Connecting cables	
D-sub connctn. (female) 15-pin	Pin layout	D-sub connctr. (male) 15-pin	Color
1	Analog output 1: ±10 V	1	Brown
2	Do not assign	2	Brown/Green
3	Analog output 2: ±10 V	3	Yellow
4	Analog output 5: ±10 V	4	Red/Blue
5	Analog output 3: ±10 V	5	Pink
6	Analog output 5: 0 V	6	Gray/Pink
7	Analog output 4: ±10 V	7	Red
8	Analog output 6: ±10 V	8	Violet
9	Analog output 1: 0 V	9	White
10	Do not assign	10	White/Gray
11	Analog output 2: 0 V	11	Green
12	Do not assign	12	
13	Analog output 3: 0 V	13	Gray
14	Analog output 4: 0 V	14	Blue
15	Analog output 6: 0 V	15	Black
Housing	Ext. shield	Housing	Ext. shield

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Danger

Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

X9: Analog outputs 7 to 13

For connecting cables, see "Cable Overview" at the end of this chapter.

There is no connector X9 on the MC 422C.

Pin layout on the MC 422(B) and connecting cables:

MC 42x(B)		Connecting cables	
D-sub connctn. (female) 15-pin	**Pin layout**	**D-sub connctr. (male) 15-pin**	**Color**
1	Analog output 7: ±10 V	1	Brown
2	Analog output 13 [a]: ±10 V	2	Brown/Green
3	Analog output 8: ±10 V	3	Yellow
4	Analog output 11: ±10 V	4	Red/Blue
5	Analog output 9: ±10 V	5	Pink
6	Analog output 11: 0 V	6	Gray/Pink
7	Analog output 10: ±10 V	7	Red
8	Analog output 12: ±10 V	8	Violet
9	Analog output 7: 0 V	9	White
10	Analog output 13[a]: 0 V	10	White/Gray
11	Analog output 8: 0 V	11	Green
12	Do not assign	12	
13	Analog output 9: 0 V	13	Gray
14	Analog output 10: 0 V	14	Blue
15	Analog output 12: 0 V	15	Black
Housing	Ext. shield	Housing	Ext. shield

a. Only for MC 422B, not for MC 422

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Danger

Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

3.16 Touch Probe Systems

The following touch probes can be connected to the iTNC 530:

- TS 220, touch-trigger probe with cable connection for workpiece setup and measurement during machining
- TS 440, TS 640, touch-trigger probes with infrared transmission for workpiece setup and measurement during machining
- TT 130, a touch probe for workpiece measurement

For suitable connecting cables, see "Cable Overview" at end of chapter.

3.16.1 Triggering touch probe for workpiece measurement

X12: Touch probe connection

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Pin layout for TS 220:

MC 42x(B,C)		Adapter cable 274 543-xx			TS 220	
Female	**Assignment**	**Male**	**Color**	**Pin**	**Pin**	**Color**
1	0 V (internal shield)	1				
2	Do not assign	2				
3	Readiness	3	Pink	4	4	
4	Start	4				
5	+15 V ± 10% (U_P), max. 100 mA	5	Gray	3	3	
6	+5 V ± 5% (U_P), max. 100 mA	6	Brown/Green	2	2	Brown
7	$\overline{\text{Battery warning}}$	7	Gray			
8	0 V (U_N)	8	White/Green	1	1	White
9	Trigger signal	9	Green	5	5	Green
10	$\overline{\text{Trigger signal}}$[a]	10	Yellow	6	6	Yellow
11 to 15	Do not assign	11 to 15				
Hsg.	External shield	Hsg.	External shield	Hsg.		

a. Stylus at rest means logic level HIGH.

Pin layout for TS 440, TS 640 with SE 640:

MC 42x(B,C)		Adapter cable 310 197-xx			SE 640		TS 440, TS 640
Female	**Assignment**	**Male**	**Color**	**Female**	**Male**	**Color**	
1	0 V (internal shield)	1	White/ Brown	7			
2	Do not assign						
3	Readiness	3	Gray	5	5	Gray	
4	Start	4	Yellow	3	3		
5	+15 V ± 10% (U_P), max. 100 mA	5	Brown	2	2	Brown	
6	+5 V ± 5% (U_P), max. 100 mA						
7	$\overline{\text{Battery warning}}$	7	Blue	6	6	Blue	
8	0 V (U_N)	8	White	1	1	White	
9	Trigger signal						
10	$\overline{\text{Trigger signal}}$[a]	10	Green	4	4	Green	
11 to 15	Do not assign						
Hsg.	Ext. shield	Hsg.	External shield	Hsg.	Hsg.		

a. Stylus at rest means logic level HIGH.

Pin layout for TS 440, TS 640 with SE 540:

MC 42x(B,C)		Adapter cable 310 197-xx			Adapter cable 517 375-xx			SE 540
Female	**Assignment**	**Male**	**Color**	**Female**	**Male**	**Color**	**Female**	
1	0 V (internal shield)	1	White/ Brown	7	7	Internal shield	7	**TS 440, TS 640**
2	Do not assign							
3	Readiness	3	Gray	5	5	Gray	5	
4	Start	4	Yellow	3	3	Yellow	3	
5	+15 V ± 10% (U_P), max. 100 mA	5	Brown	2	2	Brown	2	
6	+5 V ± 5% (U_P), max. 100 mA							
7	$\overline{\text{Battery warning}}$	7	Blue	6	6	Blue	6	
8	0 V (U_N)	8	White	1	1	White	1	
9	Trigger signal							
10	$\overline{\text{Trigger signal}}$[a]	10	Green	4	4	Green	4	
11 to 15	Do not assign							
Hsg.	Ext. shield	Hsg.	External shield	Hsg.	Hsg.	External shield	Hsg.	

a. Stylus at rest means logic level HIGH.

3.16.2 Triggering touch probe for tool measurement

X13: Connection of the touch probe

Pin layout on the MC 422(B,C):

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Pin layout on adapter cable and touch probe:

MC 42x(B,C)		Adapter cable 335 332-xx			TT130 296 537-xx	
Female	Assignment	Male	Color	Female	Male	Color
1	Readiness	1	Pink	6	6	
2	0 V (U_N)	2	White/Green	1	1	White
3	Do not assign	3				
4	+15 V ±5% (U_P)	4	Brown/Green	2	2	Brown
5	Do not assign	5		5	5	
6	Do not assign	6				
7	+5 V ±5% (U_P)	7				
8	Trigger signal	8	Brown	3	3	Green
9	$\overline{\text{Trigger signal}}$[a]	9	Green	4	4	Yellow
–	–	–	–	7	7	
Hsg.	Ext. shield	Hsg.	Ext. shield	Hsg.	Hsg.	

a. Stylus at rest means logic level HIGH.

3.17 Data Interfaces

Please note:

- Maximum cable length with Ethernet: 400 m (shielded), 100 m (unshielded)
- Maximum cable length with RS-232-C/V.24 is 20 meters.
- Maximum cable length with RS-422/V.11 is 1000 meters.

For connecting cables, see "Cable Overview" at the end of this chapter.

General information

Keep the following information in mind when connecting external peripheral devices to the data interfaces of the control:

- **Wiring:**
  The correct routing of the data cable is very important. The cable should not be located in the vicinity of power cables, except if appropriate protective measures are taken.
- **Software:**
  The use of non-HEIDENHAIN software may cause problems. The software supplier is responsible for the correct configuration of the control’s data interfaces.
- **Switch box:**
  In many cases, a switch box is used to connect several control interfaces to a PC. The difference in potential of the machines may lead to fault currents causing damage to the interfaces. If you intend to install such a switch box, you should connect the interface of the control to the switch box via a unit with electrical isolation.

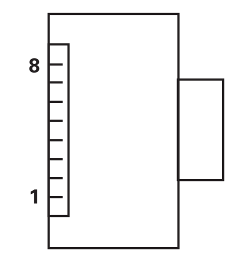
X26: Ethernet interface RJ45-port

Maximum data transfer rate:
Approx. 2 to 5 Mbps (depending on file type and network utilization)

Maximum cable length if shielded: 100 m

RJ45 connection (female) 8-pin	Assignment
1	TX+
2	TX-
3	REC+
4	Do not assign
5	Do not assign
6	REC-
7	Do not assign
8	Do not assign
Housing	Ext. shield

Face of the connector:



Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Meanings of the LEDs on the Ethernet data interface X26:

LED	Status	Meaning
Green	Blinking	Interface active
	Off	Interface inactive
Yellow	On	100 Mb network
	Off	10 Mb network

X126, X129: Ethernet interface RJ45 port

Reserved for the connection of Windows computers and real-time computers to the MC 422C DP.

Note

On the MC 422C DP, X129 and X126 must be connected with each other via an RJ 45 cable (Ethernet cable).
If this connection is missing or was removed during operation, the main computer must be shut down and switched off with the main switch to ensure that it is not under power. The HeROS computer cannot be rebooted until the connection between X129 and X126 has been restored.

X27, X127: RS-232-C/V.24 data interface

Pin layout:

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

25-pin adapter block:

MC 42x(B,C)		**Connecting cable 365 725-xx**			**Adapter block 310 085-01**		**Connecting cable 274 545-xx**		
Male	**Assignment**	**Female**	**Color**	**Female**	**Male**	**Female**	**Male**	**Color**	**Female**
1	Do not assign	1		1	1	1	1	White/ Brown	1
2	RXD	2	Yellow	3	3	3	3	Yellow	2
3	TXD	3	Green	2	2	2	2	Green	3
4	DTR	4	Brown	20	20	20	20	Brown	6
									8
5	Signal GND	5	Red	7	7	7	7	Red	7
6	DSR	6	Blue	6	6	6	6	Violet	20
							8		
7	RTS	7	Gray	4	4	4	4	Gray	5
8	CTS	8	Pink	5	5	5	5	Pink	4
9	Do not assign	9							
Hsg.	Ext. shield	Hsg.	Ext. shield	Hsg.	Hsg.	Hsg.	Hsg.	Ext. shield	Hsg.

9-pin adapter block:

MC 42x(B,C)		Connecting cable 355 484-xx			Adapter block 363 987-02		Connecting cable 366 964-xx		
Male	**Pin layout**	**Female**	**Color**	**Male**	**Female**	**Male**	**Female**	**Color**	**Female**
1	Do not assign	1	Red	1	1	1	1	Red	1
2	RXD	2	Yellow	2	2	2	2	Yellow	3
3	TXD	3	White	3	3	3	3	White	2
4	DTR	4	Brown	4	4	4	4	Brown	6
5	Signal GND	5	Black	5	5	5	5	Black	5
6	DSR	6	Violet	6	6	6	6	Violet	4
7	RTS	7	Gray	7	7	7	7	Gray	8
8	CTS	8	White/ Green	8	8	8	8	White/ Green	7
9	Do not assign	9	Green	9	9	9	9	Green	9
Hsg.	Ext. shield	Hsg.	Ext. shield	Hsg.	Hsg.	Hsg.	Hsg.	Ext. shield	Hsg.

X28, X128: RS-422/V.11 Data interface

Pin layout:

MC 42x(B,C)		Connecting cable 355-484 xx			Adapter block 363 987-01	
Female	**Pin layout**	**Male**	**Color**	**Female**	**Male**	**Female**
1	RTS	1	Red	1	1	1
2	DTR	2	Yellow	2	2	2
3	$\overline{\text{RXD}}$	3	White	3	3	3
4	$\overline{\text{TxD}}$	4	Brown	4	4	4
5	0 V	5	Black	5	5	5
6	CTS	6	Violet	6	6	6
7	DSR	7	Gray	7	7	7
8	RxD	8	White/ Green	8	8	8
9	TxD	9	Green	9	9	9
Hsg.	Ext. shield	Hsg.	Ext. shield	Hsg.	Hsg.	Hsg.

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

3.18 Handwheel Input

The following handwheels can be used with HEIDENHAIN contouring controls:

- HR 130 panel-mounted handwheel
- HR 150 panel-mounted handwheels via HRA 110 handwheel adapter
- HR 410 or HR 420 portable handwheel
- HR 520 portable handwheel

X23: Handwheel input

Pin layout:

D-sub connection (female) 9-pin	Pin layout
1	CTS
2	0 V
3	RTS
4	+12 V
5	Do not assign
6	DTR
7	TxD
8	RxD
9	DSR
Housing	Ext. shield

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

3.18.1 HR 4xx or HR 5xx portable handwheel

The HR 4xx and HR 5xx are portable electronic handwheels.

For the assignment of the keys of the HR 410 to the PLC inputs and outputs, See "HR 410 portable handwheel" on page 1453.

Pin layout

Pin layout for the various extension cables, adapter cables, connecting cables, and the handwheel:

Extension cable ID 281 429-xx			**Adapter cable ID 296 466-xx**			**Connecting cable ID: see "Introduction" chapter**			**HR 4xx HR 5xx**	
D-sub connector (male) 9-pin		**D-sub cnnctr. (female) 9-pin**	**D-sub connector (male) 9-pin**		**Coupling on mounting base (female) (5+7)-pin**	**Cnnctr. (male) (5+7)-pin**		**Cnnctr. (female) (5+7)-pin**	**Connctr. (male) (5+7)-pin**	
Hsg.	Shield	Housing	Hsg.	Shield	Housing	Hsg.	Shield	Housing	Hsg.	Shield
2	White	2	2	White	E	E	White	E	E	
4	Brown	4	4	Brown	D	D	Brown	D	D	
6	Yellow	6	6	Yellow	B	B	Yellow	B	B	
7	Gray	7	7	Gray	A	A	Gray	A	A	
8	Green	8	8	Green	C	C	Green	C	C	
					6	6	BK	6	6	
					7	7	Red/ Blue	7	7	
					5	5	Red	5	5	
					4	4	Blue	4	4	
					2	2	WH/GN	2	2	
					3	3	BN/GN	3	3	
					1	1	GY/PK	1	1	
				WH/BN	3	Contacts 1 + 2				
				WH/YL	2	Contact 2 (left) permissive button				
				WH/GN	1	Contact 1 (right)				
				WH/BL	1	Contact 1				
				WH/RD	2	Contact 1 EMERGENCY STOP				
				YL/BK	3	Contact 2				
				WH/BK	4	Contact 2				

Note

The interfaces comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

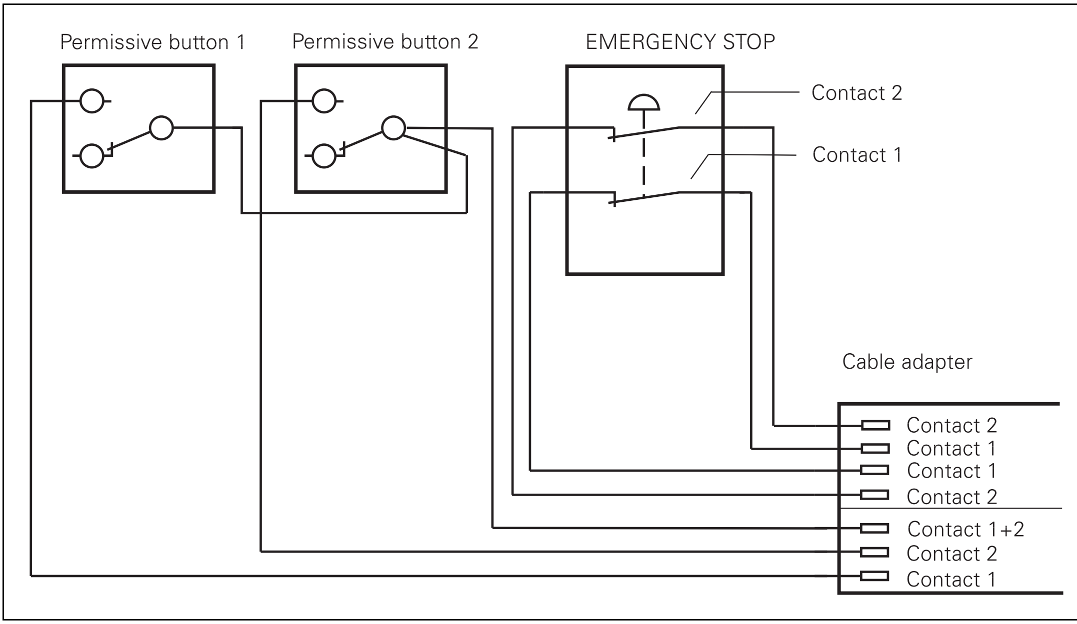
Danger

Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

The adapter includes plug-in terminal strips for the contacts of the EMERGENCY STOP button and permissive button (max. load 24 V–, 1.2 A).

The plug-in terminal strips are supplied together with the adapter cable. If you have an immediate need for these terminal strips, they can be ordered in advance. See the "Additional components" table below.

Internal wiring of the contacts for the EMERGENCY STOP and permissive buttons:



Note

HEIDENHAIN would like to point out that the permissive buttons on the HR 520, HR 420 and HR 410 handwheels are not cross-circuit proof. If your machine requires a handwheel with cross-circuit proof permissive buttons, you must use a HR 410S or HR 520S handwheel.

You must calculate the probability of failure according to EN 13849-1 for the safety circuit with permissive information. This value is used to determine a maximum testing time for this safety circuit, which must not be exceeded. The control must be turned off and on again within this testing time.

Additional components		**ID**
Dummy plug for EMERGENCY STOP circuit		271 958-03
Connecting cables		
	Spiral cable	312 879-01
	Normal cable	296 467-xx
	Metal armor tubing	296 687-xx
Plug-in terminal strips for advance ordering		
	3-pin terminal block	266 364-06
	4-pin terminal block	266 364-12

3.18.2 HR 130 panel-mounted handwheel

Standard cable length for the HR 130 is 1 meter.

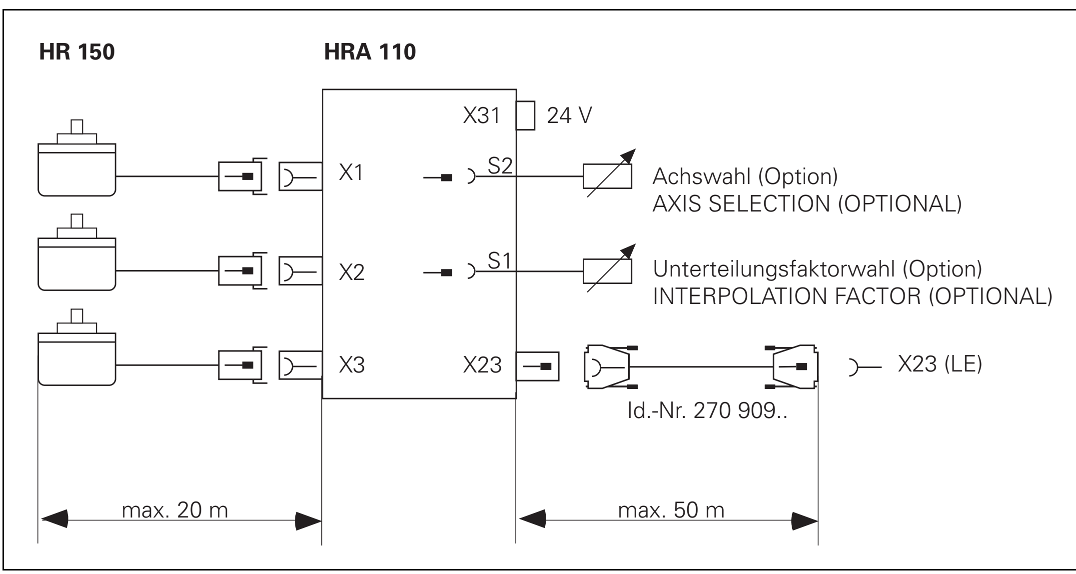
Pin layout for extension cable and handwheel:

Extension cable ID 281 429-xx			HR 130 ID 254 040-xx	
D-sub connector (male) 9-pin		**D-sub cnnctr. (female) 9-pin**	**D-sub connector (male) 9-pin**	
Housing	Shield	Housing	Housing	Shield
2	White	2	2	White
4	Brown	4	4	Brown
6	Yellow	6	6	Yellow
8	Green	8	8	Green
7	Gray	7		

3.18.3 HRA 110 handwheel adapter

With the handwheel adapter you can connect two or three HR 150 panel-mounted handwheels to the TNC.

The first and second handwheels are assigned to the X and Y axes. The third handwheel can be assigned either through a selection switch (option) or with MP7645.



An additional switch enables you to select, for example, the subdivision factor for the handwheel. In the PLC you must evaluate the current position of the handwheel selection switch and activate the corresponding subdivision factor with Module 9036.

X1 to X3: Inputs on the HRA 110 for the HR 150

Pin layout:

HRA 110	
Connection (female) 9-pin	**Assignment**
1	I_1 +
2	I_1 –
5	I_2 +
6	I_2 –
7	I_0 –
8	I_0 +
3	+ 5 V
4	0 V
9	Internal shield
Housing	Ext. shield

X23: Connection to MC 422x

Pin layout on the HRA 110:

HRA 110	
D-sub connection (female) 9-pin	**Pin layout**
1	RTS
2	0 V
3	CTS
4	+12 V +0.6 V (U_V)
5	Do not assign
6	DSR
7	RxD
8	TxD
9	DTR
Housing	Ext. shield

X31: HRA 110 supply voltage

Pin layout on the HRA 110:

Attention

The power supply of the PLC must not be used simultaneously for the HRA 110, otherwise the metallic isolation of the PLC inputs/outputs would be bridged.

HRA 110	
Connecting terminal	**Pin layout**
1	+24 V– as per IEC 742 (VDE 551)
2	0 V

Maximum current consumption 200 mA.

3.19 Input for Spindle Reference Signal

If you mount a HEIDENHAIN encoder directly onto the spindle—without a transmission—you must not wire this input.

If you use the X30 input for evaluation of the reference signal, then adjust this function with MP3143.

X30: Reference signal for spindle

Pin layout:

Connecting terminal	Pin layout
1	+24 V
2	0 V

3.20 Switching Inputs 24 V– (PLC)

3.20.1 Input signals and addresses

Input signals of the switching inputs on the MC 42x(B,C), PL 4xxB, and PLD 16-8:

Voltage range	MC 42x(B,C), PL 4xxB	PLD 16-8
"1" signal: U_i	13 V to 30.2 V	13 V to 28.8 V
"0" signal: U_i	–20 V to 3.2 V	–3 V to 2.5 V

Current ranges	MC 42x(B,C)	PL 4xxB	PLD 16-8
"1" signal: I_i	3.8 mA to 8.9 mA	2.5 mA to 6 mA	2.5 mA to 5.8 mA
"0" signal: I_i when U_i = 3.2 V	1.0 mA	0.65 mA	0.3 mA

Addresses of the switching inputs at:

Address	Number	Device
I0 to I31	31 + Control-is-ready signal	MC 42x(B,C), X42 (PLC input)
I128 to I152	25	MC 42x(B,C), X46 (machine operating panel)
I64 to I127 I64 to I95	64 32	First PL 410B, PL 51x (PLD 16-8) First PL 405B
I192 to I255 I192 to I223	64 32	Second PL 410B, PL 51x (PLD 16-8) Second PL 405B
I256 to I319 I256 to I287	64 32	Third PL 410B, PL 51x (PLD 16-8) Third PL 405B
I320 to I383 I320 to I351	64 32	Fourth PL 410B, PL 51x (PLD 16-8) Fourth PL 405B

X42: PLC inputs on the MC 422x

Pin layout:

MC 42x(B,C)		Connecting cable ID 635 876-xx, ID 635 877-xx	
D-sub connection (female) 37-pin	**Assignment**	**D-sub connection (male) 37-pin**	
1	I0	1	Gray/Red
2	I1	2	Brown/Black
3	I2	3	White/Black
4	I3 Control-is-ready signal acknowledgement	4	Green/Black
5	I4	5	Brown/Red
6	I5	6	White/Red
7	I6	7	White/Green
8	I7	8	Red/Blue
9	I8	9	Yellow/Red
10	I9	10	Gray/Pink
11	I10	11	Black
12	I11	12	Pink/Brown
13	I12	13	Yellow/Blue
14	I13	14	Green/Blue
15	I14	15	Yellow
16	I15	16	Red
17	I16	17	Gray
18	I17	18	Blue
19	I18	19	Pink
20	I19	20	White/Gray
21	I20	21	Yellow/Gray
22	I21	22	Green/Red
23	I22	23	White/Pink
24	I23	24	Gray/Green
25	I24	25	Yellow/Brown
26	I25	26	Gray/Brown
27	I26	27	Yellow/Black
28	I27	28	White/Yellow
29	I28	29	Gray/Blue
30	I29	30	Pink/Blue
31	I30	31	Pink/Red
32	I31	32	Brown/Blue
33	I32 Drive enable	33	Pink/Green
34	Do not assign	34	Brown
35	0 V (PLC) test output; do not assign	35	Yellow/Pink
36	0 V (PLC) test output; do not assign	36	Violet
37	0 V (PLC) test output; do not assign	37	White
Housing	Ext. shield	Housing	Ext. shield

3.20.2 PLC inputs on the PL 410B

X3 to X6: PLC inputs

Pin layout on the PL:

X3				
Terminal	**Pin layout**			
	PL 1	**PL 2**	**PL 3**	**PL 4**
1	I64	I192	I256	I320
2	I65	I193	I257	I321
3	I66	I194	I258	I322
4	I67	I195	I259	I323
5	I68	I196	I260	I324
6	I69	I197	I261	I325
7	I70	I198	I262	I326
8	I71	I199	I263	I327
9	I72	I200	I264	I328
10	I73	I201	I265	I329
11	I74	I202	I266	I330
12	I75	I203	I267	I331
13	I76	I204	I268	I332
14	I77	I205	I269	I333
15	I78	I206	I270	I334
16	I79	I207	I271	I335

X4				
Terminal	**Pin layout**			
	PL 1	**PL 2**	**PL 3**	**PL 4**
1	I80	I208	I272	I336
2	I81	I209	I273	I337
3	I82	I210	I274	I338
4	I83	I211	I275	I339
5	I84	I212	I276	I340
6	I85	I213	I277	I341
7	I86	I214	I278	I342
8	I87	I215	I279	I343
9	I88	I216	I280	I344
10	I89	I217	I281	I345
11	I90	I218	I282	I346
12	I91	I219	I283	I347
13	I92	I220	I284	I348
14	I93	I221	I285	I349
15	I94	I222	I286	I350
16	I95	I223	I287	I351

X5				
Terminal	**Pin layout**			
	PL 1	**PL 2**	**PL 3**	**PL 4**
1	I96	I224	I288	I352
2	I97	I225	I289	I353
3	I98	I226	I290	I354
4	I99	I227	I291	I355
5	I100	I228	I292	I356
6	I101	I229	I293	I357
7	I102	I230	I294	I358
8	I103	I231	I295	I359
9	I104	I232	I296	I360
10	I105	I233	I297	I361
11	I106	I234	I298	I362
12	I107	I235	I299	I363
13	I108	I236	I300	I364
14	I109	I237	I301	I365
15	I110	I238	I302	I366
16	I111	I239	I303	I367

X6				
Terminal	**Pin layout**			
	PL 1	**PL 2**	**PL 3**	**PL 4**
1	I112	I240	I304	I368
2	I113	I241	I305	I369
3	I114	I242	I306	I370
4	I115	I243	I307	I371
5	I116	I244	I308	I372
6	I117	I245	I309	I373
7	I118	I246	I310	I374
8	I119	I247	I311	I375
9	I120	I248	I312	I376
10	I121	I249	I313	I377
11	I122	I250	I314	I378
12	I123	I251	I315	
13	I124	I252	I316	I380
14	I125	I253	I317	I381
15	I126	I254	I318	I382
16	I127	I255	I319	I383

3.20.3 PLC inputs on the PL 405B

X3, X4: PLC inputs

Pin layout on the PL:

X3				
Terminal	**Pin layout**			
	PL 1	**PL 2**	**PL 3**	**PL 4**
1	I64	I192	I256	I320
2	I65	I193	I257	I321
3	I66	I194	I258	I322
4	I67	I195	I259	I323
5	I68	I196	I260	I324
6	I69	I197	I261	I325
7	I70	I198	I262	I326
8	I71	I199	I263	I327
9	I72	I200	I264	I328
10	I73	I201	I265	I329
11	I74	I202	I266	I330
12	I75	I203	I267	I331
13	I76	I204	I268	I332
14	I77	I205	I269	I333
15	I78	I206	I270	I334
16	I79	I207	I271	I335

X4				
Terminal	**Pin layout**			
	PL 1	**PL 2**	**PL 3**	**PL 4**
1	I80	I208	I272	I336
2	I81	I209	I273	I337
3	I82	I210	I274	I338
4	I83	I211	I275	I339
5	I84	I212	I276	I340
6	I85	I213	I277	I341
7	I86	I214	I278	I342
8	I87	I215	I279	I343
9	I88	I216	I280	I344
10	I89	I217	I281	I345
11	I90	I218	I282	I346
12	I91	I219	I283	I347
13	I92	I220	I284	I348
14	I93	I221	I285	I349
15	I94	I222	I286	I350
16	I95	I223	I287	I351

3.20.4 PLC inputs on the PL 51x

X4 to X5: PLC inputs

Pin layout on the PLD 16-8 input/output module:

When using a PLB 511 or PLB 512, please note the additional conditions starting on page 431.

Note

The 0 V terminals of X4 and X5 of the PLD 16-8 are connected internally. These connections are used for connecting the potential of the electronics and for operating the LEDs. Since only a low current is required (max. 50 mA), it is sufficient to establish only one 0 V connection (preferably at X4).

X4											
Assignment		**Terminal**									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
First PL 51x	Slot 1	0 V	0 V	I64	I65	I66	I67	I68	I69	I70	I71
	Slot 2	0 V	0 V	I80	I81	I82	I83	I84	I85	I86	I87
	Slot 3	0 V	0 V	I96	I97	I98	I99	I100	I101	I102	I103
	Slot 4	0 V	0 V	I112	I113	I114	I115	I116	I117	I118	I119
Second PL 51x	Slot 1	0 V	0 V	I192	I193	I194	I195	I196	I197	I198	I199
	Slot 2	0 V	0 V	I208	I209	I210	I211	I212	I213	I214	I215
	Slot 3	0 V	0 V	I224	I225	I226	I227	I228	I229	I230	I231
	Slot 4	0 V	0 V	I240	I241	I242	I243	I244	I245	I246	I247
Third PL 51x	Slot 1	0 V	0 V	I256	I257	I258	I259	I260	I261	I262	I263
	Slot 2	0 V	0 V	I272	I273	I274	I275	I276	I277	I278	I279
	Slot 3	0 V	0 V	I288	I289	I290	I291	I292	I293	I294	I295
	Slot 4	0 V	0 V	I304	I305	I306	I307	I308	I309	I310	I311
Fourth PL 51x	Slot 1	0 V	0 V	I320	I321	I322	I323	I324	I325	I326	I327
	Slot 2	0 V	0 V	I336	I337	I338	I339	I340	I341	I342	I343
	Slot 3	0 V	0 V	I352	I353	I354	I355	I356	I357	I358	I359
	Slot 4	0 V	0 V	I368	I369	I370	I371	I372	I373	I374	I375

Output signals and addresses

The switching outputs are transistor outputs with current limitation.

Please note:

- Permissible load: Resistive load—inductive load only with quenching diode parallel to inductance
- MC 42x(B,C), PL 4xxB: Short circuiting of **one** output is **permissible. No more than one** output may be short-circuited **at one time.**
- PLD 16-8: The outputs are short-circuit proof.

Output signals:

	MC 42x(B,C), PL 4xxB, PLD 16-8
Min. output voltage for "1" signal	3 V below supply voltage

Note

The switching outputs need a minimum load of 5 mA.
They conform to EN 61131-2.

Attention

PLC outputs must neither be connected to a 24 V supply, nor to other PLC outputs with a difference in potential. Otherwise, the voltage present at the PLC outputs is transmitted to the power supply. As a result, the PLC outputs that can be switched off may nevertheless be supplied with this voltage.

Addresses:

Address	Number	Device
O0 to O30	31	MC 42x(B,C), X41 (PLC output)
O0 to O7	8	MC 42x(B,C), X46 (machine operating panel)
O32 to O62	31	First PLC input/output unit
O64 to O94	31	Second PLC input/output unit
O128 to O158	31	Third PLC input/output unit
O160 to O190	31	Fourth PLC input/output unit

The "control-is-ready" output at X41 can have the same load as a normal PLC output. If a higher current is required for switching a relay, the control-is-ready outputs of the PLs can be used in addition. A separate power supply for the PLs is necessary for this.

X41: PLC outputs on the MC 42x(B,C)

Pin layout:

MC 42x(B,C)		Connecting cable ID 635 876-xx, ID 635 877-xx	
D-sub connctn. (female) 37-pin	**Pin layout**	**D-sub cnnctr. (male) 37-pin**	
Supply via X44, pin 3; can be switched off with EMERGENCY STOP			
1	O0[a]	1	Gray/Red
2	O1[a]	2	Brown/Black
3	O2 [a]	3	White/Black
4	O3[a]	4	Green/Black
5	O4[a]	5	Brown/Red
6	O5[a]	6	White/Red
7	O6[a]	7	White/Green
8	O7[a]	8	Red/Blue
9	O8	9	Yellow/Red
10	O9	10	Gray/Pink
11	O10	11	Black
12	O11	12	Pink/Brown
13	O12	13	Yellow/Blue
14	O13	14	Green/Blue
15	O14	15	Yellow
16	O15	16	Red
Supply via X44, pin 2; can be switched off with EMERGENCY STOP			
17	O16	17	Gray
18	O17	18	Blue
19	O18	19	Pink
20	O19	20	White/Gray
21	O20	21	Yellow/Gray
22	O21	22	Green/Red
23	O22	23	White/Pink
24	O23	24	Gray/Green
Supply via X44, pin 1; **cannot** be switched off with EMERGENCY STOP			
25	O24	25	Yellow/Brown
26	O25	26	Gray/Brown
27	O26	27	Yellow/Black
28	O27	28	White/Yellow
29	O28	29	Gray/Blue

MC 42x(B,C)		Connecting cable ID 635 876-xx, ID 635 877-xx	
D-sub connctn. (female) 37-pin	**Pin layout**	**D-sub cnnctr. (male) 37-pin**	
30	O29	30	Pink/Blue
31	O30	31	Pink/Red
32, 33	Do not assign	32	Brown/Blue; Pink/ Green
34	Control-is-ready signal	34	Brown
35, 36, 37	Do not assign	35	Yellow/Pink; Violet; White
Housing	Ext. shield	Housing	Ext. shield

a. Also via X46 (PLC inputs/outputs)

X7, X8: PLC outputs on the PL 410 B

Pin layout on the PL:

X7				
Terminal	**Pin layout**			
	PL 1	**PL 2**	**PL 3**	**PL 4**
1	O32	O64	O128	O160
2	O33	O65	O129	O161
3	O34	O66	O130	O162
4	O35	O67	O131	O163
5	O36	O68	O132	O164
6	O37	O69	O133	O165
7	O38	O70	O134	O166
8	O39	O71	O135	O167
9	O40	O72	O136	O168
10	O41	O73	O137	O169
11	O42	O74	O138	O170
12	O43	O75	O139	O171
13	O44	O76	O140	O172
14	O45	O77	O141	O173
15	O46	O78	O142	O174
16	O47	O79	O143	O175

X8				
Terminal	**Pin layout**			
	PL 1	**PL 2**	**PL 3**	**PL 4**
1	O48	O80	O144	O176
2	O49	O81	O145	O177
3	O50	O82	O146	O178
4	O51	O83	O147	O179
5	O52	O84	O148	O180
6	O53	O85	O149	O181
7	O54	O86	O150	O182
8	O55	O87	O151	O183
9	O56	O88	O152	O184
10	O57	O89	O153	O185
11	O58	O90	O154	O186
12	O59	O91	O155	O187
13	O60	O92	O156	O188
14	O61	O93	O157	O189
15	O62	O94	O158	O190
16	Control-is-ready signal			

X8: PLC outputs on the PL 405 B

Pin layout on the PL:

X8				
Terminal	**Pin layout**			
	PL 1	**PL 2**	**PL 3**	**PL 4**
1	O48	O80	O144	O176
2	O49	O81	O145	O177
3	O50	O82	O146	O178
4	O51	O83	O147	O179
5	O52	O84	O148	O180
6	O53	O85	O149	O181
7	O54	O86	O150	O182
8	O55	O87	O151	O183
9	O56	O88	O152	O184
10	O57	O89	O153	O185
11	O58	O90	O154	O186
12	O59	O91	O155	O187
13	O60	O92	O156	O188
14	O61	O93	O157	O189
15	O62	O94	O158	O190
16	Control-is-ready signal			

X6: PLC outputs on the PL 51x

Pin layout on the PLD 16-8 input/output module:

When using a PLB 511 or PLB 512, please note the additional conditions starting on page 431.

X6											
Pin layout		**Terminal**									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
First PL 510	Slot 1	O32	O33	O34	O35	O36	O37	O38	O39[a]	+24 V[b]	+24 V[c]
	Slot 2	O40	O41	O42	O43	O44	O45	O46	O47[a]	+24 V[b]	+24 V[c]
	Slot 3	O48	O49	O50	O51	O52	O53	O54	O55[a]	+24 V[b]	+24 V[c]
	Slot 4	O56	O57	O58	O59	O60	O61	O62	O63[a]	+24 V[b]	+24 V[c]
Second PL 510	Slot 1	O64	O65	O66	O67	O68	O69	O70	O71[a]	+24 V[b]	+24 V[c]
	Slot 2	O72	O73	O74	O75	O76	O77	O78	O79[a]	+24 V[b]	+24 V[c]
	Slot 3	O80	O81	O82	O83	O84	O85	O86	O87[a]	+24 V[b]	+24 V[c]
	Slot 4	O88	O89	O90	O91	O92	O93	O94	O95[a]	+24 V[b]	+24 V[c]
Third PL 510	Slot 1	O128	O129	O130	O131	O132	O133	O134	O135[a]	+24 V[b]	+24 V[c]
	Slot 2	O136	O137	O138	O139	O140	O141	O142	O143[a]	+24 V[b]	+24 V[c]
	Slot 3	O144	O145	O146	O147	O148	O149	O150	O151[a]	+24 V[b]	+24 V[c]
	Slot 4	O152	O153	O154	O155	O156	O157	O158	O159[a]	+24 V[b]	+24 V[c]
Fourth PL 510	Slot 1	O160	O161	O162	O163	O164	O165	O166	O167[a]	+24 V[b]	+24 V[c]
	Slot 2	O168	O169	O170	O171	O172	O173	O174	O175[a]	+24 V[b]	+24 V[c]
	Slot 3	O176	O177	O178	O179	O180	O181	O182	O183[a]	+24 V[b]	+24 V[c]
	Slot 4	O184	O185	O186	O187	O188	O189	O190	O191[a]	+24 V[b]	+24 V[c]

a. The function of this terminal can be set with a sliding switch on the rear side of the PLD 16-8 I/O module:
Setting 1: Control-is-ready signal
Setting 2: PLC output

b. Group 1 (terminals 1 to 4)

c. Group 2 (terminals 5 to 8)

Note

If you use only the outputs at X6 for a PLD 16-8 I/O unit (and no inputs), the 0 V connection for supplying the electronics and for operating the LEDs must be established at X4 or X5 (See "X4 to X5: PLC inputs" on page 416).

Note

The iTNC 530 cyclically monitors the PLC outputs of the PL 510 for a short-circuit.

3.22 PLC Input/Output Units

3.22.1 PL 4xx B

Up to four PL 4xxB can be connected to the MC 422(B,C).

The PL 410 B is available with and without analog inputs.

Device	ID	Switching inputs 24 V–	Switching outputs 24 V–	Analog inputs (± 10 V)	Inputs for Pt 100 thermistors
PL 410 B	263 371-12	64	31	–	–
PL 410 B	263 371-02	64	31	4	4
PL 405 B	263 371-22	32	15	–	–

No more than one PL 405 B may be used. If connecting through a PL 410 B, the PL 405 B must be connected last.

X47: PLC expansion on the MC 422(B,C)

Pin layout:

MC 42x(B,C)		Connecting cable ID 635 879-xx / ID 635 880-xx			1. PL 410B/PL405B	
D-sub cnnctn. (male) 25-pin	**Pin layout**	**D-sub cnnctr. (female) 25-pin**		**D-sub cnnctr. (male) 25-pin**	**X1 D-sub cnnctn. (female) 25-pin**	**Pin layout**
1	0 V	1	Brown, Yellow, Pink, Red, Violet	1	1	0 V
2	0 V	2	RD/BL, BN/GN, YL/BN, GY/BN, PK/BN	2	2	0 V
3	0 V	3	BN/BL, BN/RD, BN/BK, YL/GY, YL/PK	3	3	0 V
4	Do not assign	4	Gray/Green	4	4	Serial IN 2
5	$\overline{\text{Address 6}}$	5	White/Green	5	5	$\overline{\text{Address 6}}$
6	INTERRUPT	6	Pink/Green	6	6	INTERRUPT
7	$\overline{\text{RESET}}$	7	Green/Blue	7	7	$\overline{\text{RESET}}$
8	$\overline{\text{WRITE EXTERN}}$	8	White/Blue	8	8	$\overline{\text{WRITE EXTERN}}$
9	WRITE EXTERN	9	White/Red	9	9	WRITE EXTERN
10	$\overline{\text{Address 5}}$	10	Gray/Pink	10	10	$\overline{\text{Address 5}}$
11	$\overline{\text{Address 3}}$	11	Blue	11	11	$\overline{\text{Address 3}}$
12	$\overline{\text{Address 1}}$	12	Green	12	12	$\overline{\text{Address 1}}$
13	Do not assign	13		13	13	Do not assign
14	PCB identifier 3	14	Yellow/Blue, Pink/Blue, Yellow/Black	14	14	+ 12 V
15	PCB identifier 4	15	Yellow/Red, Gray/Red, Pink/Red	15	15	+ 12 V
16	Do not assign	16	Gray/Blue	16	16	PCB identifier 2
17	Do not assign	17	Green/Black	17	17	PCB identifier 1
18	$\overline{\text{Address 7}}$	18	White/Yellow	18	18	$\overline{\text{Address 7}}$
19	Serial IN 1	19	White/Black	19	19	Serial IN 1
20	EM. STOP	20	Green/Red	20	20	EM. STOP
21	$\overline{\text{Serial OUT}}$	21	White/Gray	21	21	$\overline{\text{Serial OUT}}$
22	Serial OUT	22	White/Pink	22	22	Serial OUT
23	$\overline{\text{Address 4}}$	23	Black	23	23	$\overline{\text{Address 4}}$
24	$\overline{\text{Address 2}}$	24	Gray	24	24	$\overline{\text{Address 2}}$
25	$\overline{\text{Address 0}}$	25	White	25	25	$\overline{\text{Address 0}}$
Housing	Ext. shield	Housing	Ext. shield	Housing	Housing	Ext. shield

X2: PLC expansion PL 4xx B on the PL 410B

Pin layout:

PL 410 B		Connecting cable ID 635 879-xx / ID 635 880-xx			PL 410B PL 405B on the PL 410B	
D-sub cnnctn. (male) 25-pin	**Pin layout**	**D-sub cnnctr. (female) 25-pin**		**D-sub cnnctr. (male) 25-pin**	**X1 D-sub cnnctn. (female) 25-pin**	**Pin layout**
1	0 V	1	Brown, Yellow, Pink, Red, Violet	1	1	0 V
2	0 V	2	RD/BL, BN/GN, YL/BN, GY/BN, PK/BN	2	2	0 V
3	0 V	3	BN/BL, BN/RD, BN/BK, YL/GY, YL/PK	3	3	0 V
4	Do not assign	4	Gray/Green	4	4	Serial IN 2
5	$\overline{\text{Address 6}}$	5	White/Green	5	5	$\overline{\text{Address 6}}$
6	INTERRUPT	6	Pink/Green	6	6	INTERRUPT
7	$\overline{\text{RESET}}$	7	Green/Blue	7	7	$\overline{\text{RESET}}$
8	$\overline{\text{WRITE EXTERN}}$	8	White/Blue	8	8	$\overline{\text{WRITE EXTERN}}$
9	WRITE EXTERN	9	White/Red	9	9	WRITE EXTERN
10	$\overline{\text{Address 5}}$	10	Gray/Pink	10	10	$\overline{\text{Address 5}}$
11	$\overline{\text{Address 3}}$	11	Blue	11	11	$\overline{\text{Address 3}}$
12	$\overline{\text{Address 1}}$	12	Green	12	12	$\overline{\text{Address 1}}$
13	Do not assign	13		13	13	Do not assign
14	PCB identifier 4	14	Yellow/Blue, Pink/Blue, Yellow/Black	14	14	+12 V
15	PCB identifier 3	15	Yellow/Red, Gray/Red, Pink/Red	15	15	+12 V
16	PCB identifier 2	16	Gray/Blue	16	16	PCB identifier 2
17	PCB identifier 1	17	Green/Black	17	17	PCB identifier 1
18	$\overline{\text{Address 7}}$	18	White/Yellow	18	18	$\overline{\text{Address 7}}$
19	Serial IN 1	19	White/Black	19	19	Serial IN 1
20	EM. STOP	20	Green/Red	20	20	EM. STOP
21	$\overline{\text{Serial OUT}}$	21	White/Gray	21	21	$\overline{\text{Serial OUT}}$
22	Serial OUT	22	White/Pink	22	22	Serial OUT
23	$\overline{\text{Address 4}}$	23	Black	23	23	$\overline{\text{Address 4}}$
24	$\overline{\text{Address 2}}$	24	Gray	24	24	$\overline{\text{Address 2}}$
25	$\overline{\text{Address 0}}$	25	White	25	25	$\overline{\text{Address 0}}$
Housing	Ext. shield	Housing	Ext. shield	Housing	Housing	Ext. shield

3.22.2 PL 510

Up to four PL 510 can be connected to the MC 42x(B,C).

The PLB 510 basic modules can be fitted with any combination of PLD 16-8 input/output modules and PLA 4-4 analog modules. It is also possible to leave gaps, since not all slots on the PLB 510 must be used.

Meaning of the LEDs on the PLD 16-8

LED	Meaning
Red LED at X4, pin 1	Short circuit of the outputs[a]
Yellow LEDs at X4, X5 and X6	Status of the inputs and outputs
Green LEDs at X6, pin 9 and pin 10	24 V power supply of the outputs

a. An output is reset when a short circuit occurs. Short-circuit monitoring remains in place, and must therefore be reset with Module 9139.
In order to recognize a short circuit, a current of 20 A must be able to flow for approximately 3 ms. If this is not the case (e.g. the 24 V supply limits the current sooner), the short-circuit monitoring might not become effective.

X47: PLC expansion on the MC 422 (B,C)

Pin layout:

MC 42x(B,C)		Connecting cable ID 371 046-xx			First PL 510	
D-sub cnnctn. (male) 25-pin	**Pin layout**	**D-sub cnnctr. (female) 25-pin**		**D-sub cnnctr. (male) 26-pin**	**X1 D-sub cnnctn. (female) 26-pin**	**Pin layout**
1	0 V	1	Black	1	1	0 V
2	0 V	2	Violet	2	2	0 V
3	0 V	3		3	3	0 V
4	Do not assign	4		4	4	Do not assign
5	Address 6	5	Yellow	5	5	Address 6
6	INTERRUPT	6	Blue	6	6	INTERRUPT
7	RESET	7	Red	7	7	RESET
8	WRITE EXTERN	8	Gray	8	8	WRITE EXTERN
9	WRITE EXTERN	9	Pink	9	9	WRITE EXTERN
10	Address 5	10	Green	10	10	Address 5
11	Address 3	11	White	11	11	Address 3
12	Address 1	12	Brown	12	12	Address 1
13	Do not assign	13		13	13	Do not assign
14	+5 V (output)	14	White/Blue	14	14	
15	+5 V (feedback)	15	Brown/Blue	15	15	
16	Do not assign	16	White/Pink	16	16	PCB identifier 2
17	Do not assign	17	Pink/Brown	17	17	PCB identifier 1
18	Address 7	18	Brown/Green	18	18	Address 7
19	Serial IN 1	19	White/Gray	19	19	Serial IN 1
20	EM. STOP	20	Gray/Brown	20	20	EM. STOP
21	Serial OUT	21	White/Yellow	21	21	Serial OUT
22	Serial OUT	22	Yellow/Brown	22	22	Serial OUT
23	Address 4	23	White/Green	23	23	Address 4
24	Address 2	24	Gray/Pink	24	24	Address 2
25	Address 0	25	Red/Blue	25	25	Address 0
				26	26	
Housing	Ext. shield	Housing	Ext. shield	Housing	Housing	Ext. shield

X147: PLC expansion on MC 420

Pin layout:

PL 420		Connecting cable ID 371 046-xx			First PL 510	
D-sub cnnctn. (male) 26-pin	**Pin layout**	**D-sub cnnctr. (female) 26-pin**		**D-sub cnnctr. (male) 26-pin**	**X1 D-sub cnnctn. (female) 26-pin**	**Pin layout**
1	0 V	1	Black	1	1	0 V
2	0 V	2	Violet	2	2	0 V
3	0 V	3		3	3	0 V
4	Do not assign	4		4	4	Do not assign
5	$\overline{\text{Address 6}}$	5	Yellow	5	5	$\overline{\text{Address 6}}$
6	INTERRUPT	6	Blue	6	6	INTERRUPT
7	$\overline{\text{RESET}}$	7	Red	7	7	$\overline{\text{RESET}}$
8	$\overline{\text{WRITE}}$ $\overline{\text{EXTERN}}$	8	Gray	8	8	$\overline{\text{WRITE}}$ $\overline{\text{EXTERN}}$
9	WRITE EXTERN	9	Pink	9	9	WRITE EXTERN
10	$\overline{\text{Address 5}}$	10	Green	10	10	$\overline{\text{Address 5}}$
11	$\overline{\text{Address 3}}$	11	White	11	11	$\overline{\text{Address 3}}$
12	$\overline{\text{Address 1}}$	12	Brown	12	12	$\overline{\text{Address 1}}$
13	Do not assign	13		13	13	Do not assign
14	+5 V (output)	14	White/Blue	14	14	
15	+5 V (feedback)	15	Brown/Blue	15	15	
16	PCB identifier 2	16	White/Pink	16	16	PCB identifier 2
17	PCB identifier 1	17	Pink/Brown	17	17	PCB identifier 1
18	$\overline{\text{Address 7}}$	18	Brown/Green	18	18	$\overline{\text{Address 7}}$
19	Serial IN 1	19	White/Gray	19	19	Serial IN
20	EM. STOP	20	Gray/Brown	20	20	EM. STOP
21	$\overline{\text{Serial OUT}}$	21	White/Yellow	21	21	$\overline{\text{Serial OUT}}$
22	Serial OUT	22	Yellow/Brown	22	22	Serial OUT
23	$\overline{\text{Address 4}}$	23	White/Green	23	23	$\overline{\text{Address 4}}$
24	$\overline{\text{Address 2}}$	24	Gray/Pink	24	24	$\overline{\text{Address 2}}$
25	$\overline{\text{Address 0}}$	25	Red/Blue	25	25	$\overline{\text{Address 0}}$
26		26		26	26	
Housing	Ext. shield	Housing	Ext. shield	Housing	Housing	Ext. shield

X2: PLC expansion PL 510 on the PL 510

Pin layout:

PL 510		Connecting cable ID 371 046-xx			PL 510 on PL 510	
D-sub cnnctn. (male) 26-pin	Pin layout	D-sub cnnctr. (female) 26-pin		D-sub cnnctr. (male) 26-pin	X1 D-sub cnnctn. (female) 26-pin	Pin layout
1	0 V	1	Black	1	1	0 V
2	0 V	2	Violet	2	2	0 V
3	0 V	3		3	3	0 V
4	Do not assign	4		4	4	Do not assign
5	$\overline{\text{Address 6}}$	5	Yellow	5	5	$\overline{\text{Address 6}}$
6	INTERRUPT	6	Blue	6	6	INTERRUPT
7	$\overline{\text{RESET}}$	7	Red	7	7	$\overline{\text{RESET}}$
8	$\overline{\text{WRITE EXTERN}}$	8	Gray	8	8	$\overline{\text{WRITE EXTERN}}$
9	WRITE EXTERN	9	Pink	9	9	WRITE EXTERN
10	$\overline{\text{Address 5}}$	10	Green	10	10	$\overline{\text{Address 5}}$
11	$\overline{\text{Address 3}}$	11	White	11	11	$\overline{\text{Address 3}}$
12	$\overline{\text{Address 1}}$	12	Brown	12	12	$\overline{\text{Address 1}}$
13	Do not assign	13		13	13	Do not assign
14	+5 V (output)	14	White/Blue	14	14	
15	+5 V (feedback)	15	Brown/Blue	15	15	
16	PCB identifier 2	16	White/Pink	16	16	PCB identifier 2
17	PCB identifier 1	17	Pink/Brown	17	17	PCB identifier 1
18	$\overline{\text{Address 7}}$	18	Brown/Green	18	18	$\overline{\text{Address 7}}$
19	Serial IN 1	19	White/Gray	19	19	Serial IN
20	EM. STOP	20	Gray/Brown	20	20	EM. STOP
21	$\overline{\text{Serial OUT}}$	21	White/Yellow	21	21	$\overline{\text{Serial OUT}}$
22	Serial OUT	22	Yellow/Brown	22	22	Serial OUT
23	$\overline{\text{Address 4}}$	23	White/Green	23	23	$\overline{\text{Address 4}}$
24	$\overline{\text{Address 2}}$	24	Gray/Pink	24	24	$\overline{\text{Address 2}}$
25	$\overline{\text{Address 0}}$	25	Red/Blue	25	25	$\overline{\text{Address 0}}$
26		26		26	26	
Housing	Ext. shield	Housing	Ext. shield	Housing	Housing	Ext. shield

3.23 PLB 511/PLB 512

Commissioning and configuration

The basic modules PLB 511 and PLB 512 can also be used in combination with a PLB 510. A mixed configuration of the basic modules is possible. The same cable is used for the connection as for the PLB 510. The configuration and interrogation within the PLC is performed as with the PLB 510. The PLD 16-8 and PLA 4-4 PLC modules can be used as input and output modules.

The following basic conditions must be taken into consideration:

- The control always and only detects PLB basic modules with 4 slots. A PLB 511 and a PLB 512 are each always interpreted as two PLBs with 4 slots each.
- As before, a maximum of 16 PLC I/O modules (slots), distributed over all PLBs used, can be addressed.
- Due to the fixed addressing in blocks of 4 slots, the non-existent slots 7 and 8 on a PLB 511 with 6 slots are wasted (see the example for two PLBs with 6 slots).
- In compatible mode, only the digital inputs/outputs of slots 1 to 4 function on a PLB 512. The digital inputs/outputs of slots 5 to 8, analog inputs and status information cannot be used.
- The sum of all logically existing PLs must be less than or equal to four PLs.

In addition, the following must be considered for controlling from a PLC program:

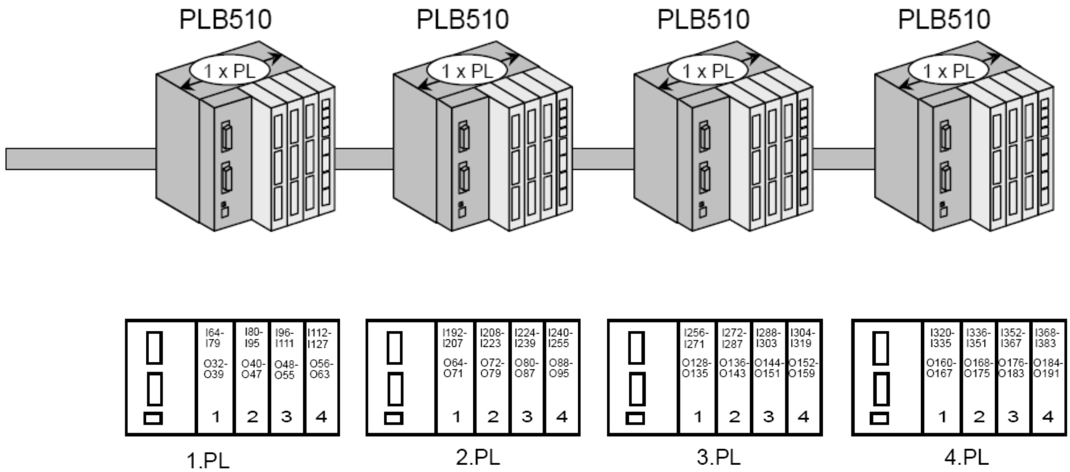
- In order to use PLB 511 or PLB 512 completely, the corresponding PLC modules must be called for two PLs in each case.
- Slots 5 to 8 cannot be addressed. Here a recalculation to another basic module and slots 1 to 4 must occur.
- A PLB 511 presents the non-existing slots 7 and 8 as "empty."

Using PLC Modules

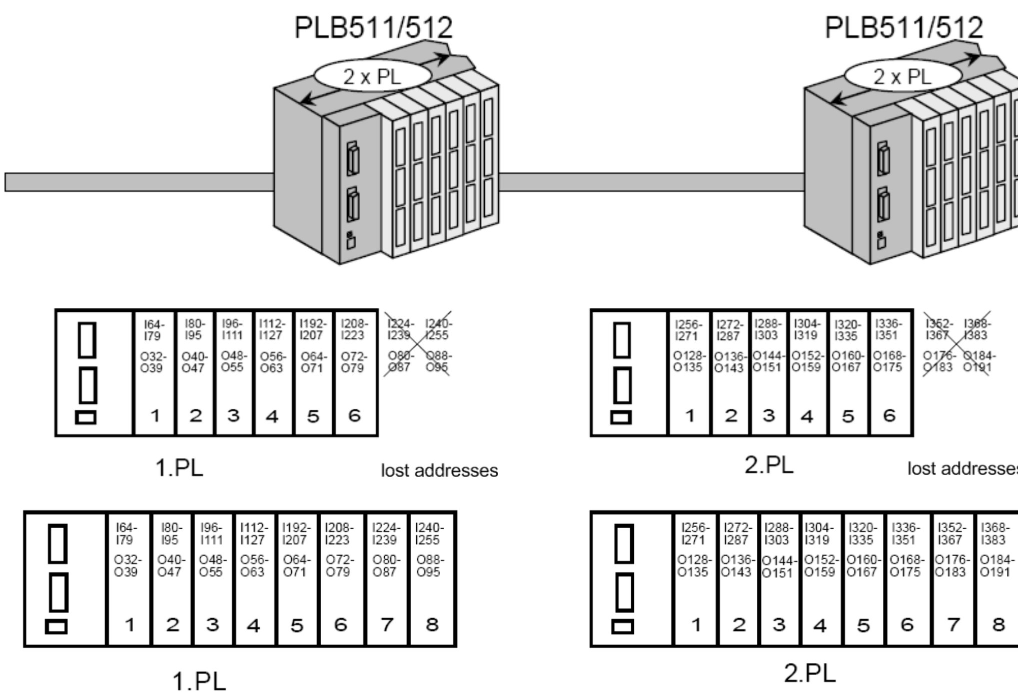
- **Module 9007 – Diagnostic information**
  Diagnostic information 3 (number of connected PLs) supplies two modules per PLB 511 or PLB 512.
- **Module 9002, 9005, 9008, 9009 – Read PL inputs/Set PL outputs**
  These modules must each be called twice for each PLB 511 or PLB 512.
- **Module 9137, 9138 – Read diagnosis/Read analog inputs**
  These modules address the required information in the following format: "Basic-module number/slot/number of the information." Here the slots 5 to 6 (PLB 511) or slots 5 to 8 (PLB 512) must be recalculated to another basic module and to slots 1 to 2 or 1 to 4 respectively.

Using address ranges

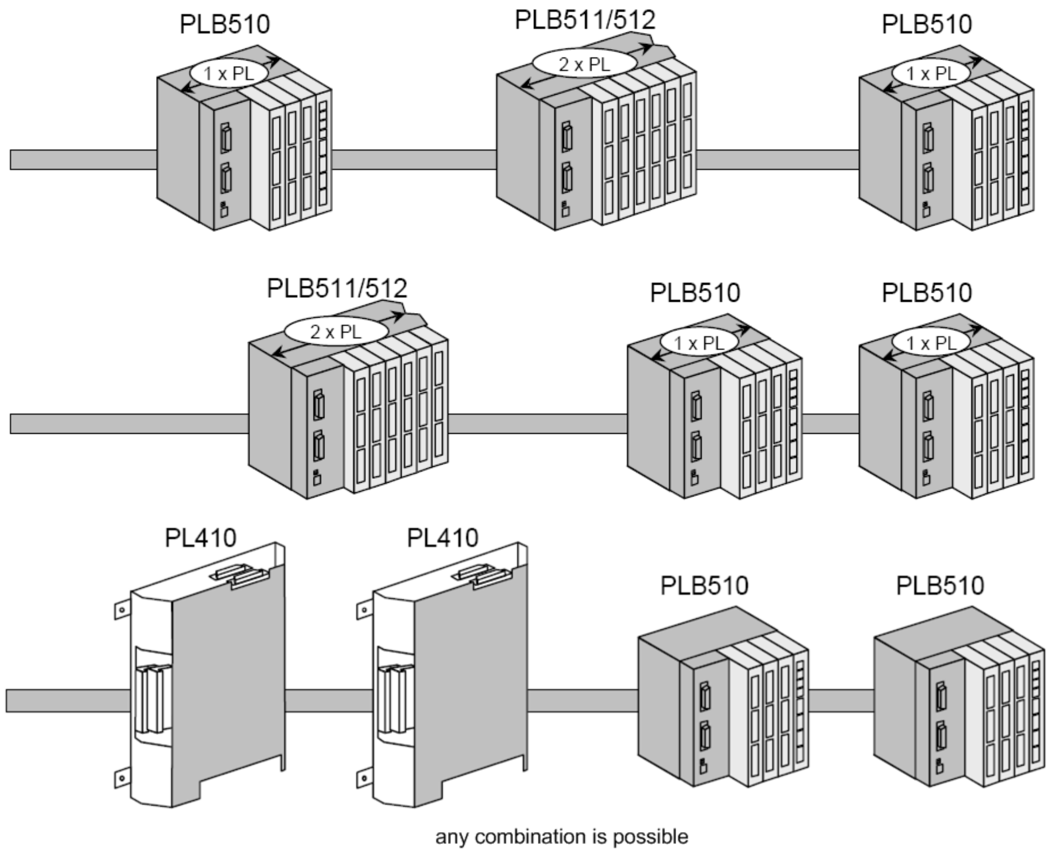
Address ranges when using four PLB 510 modules with 4 slots each:



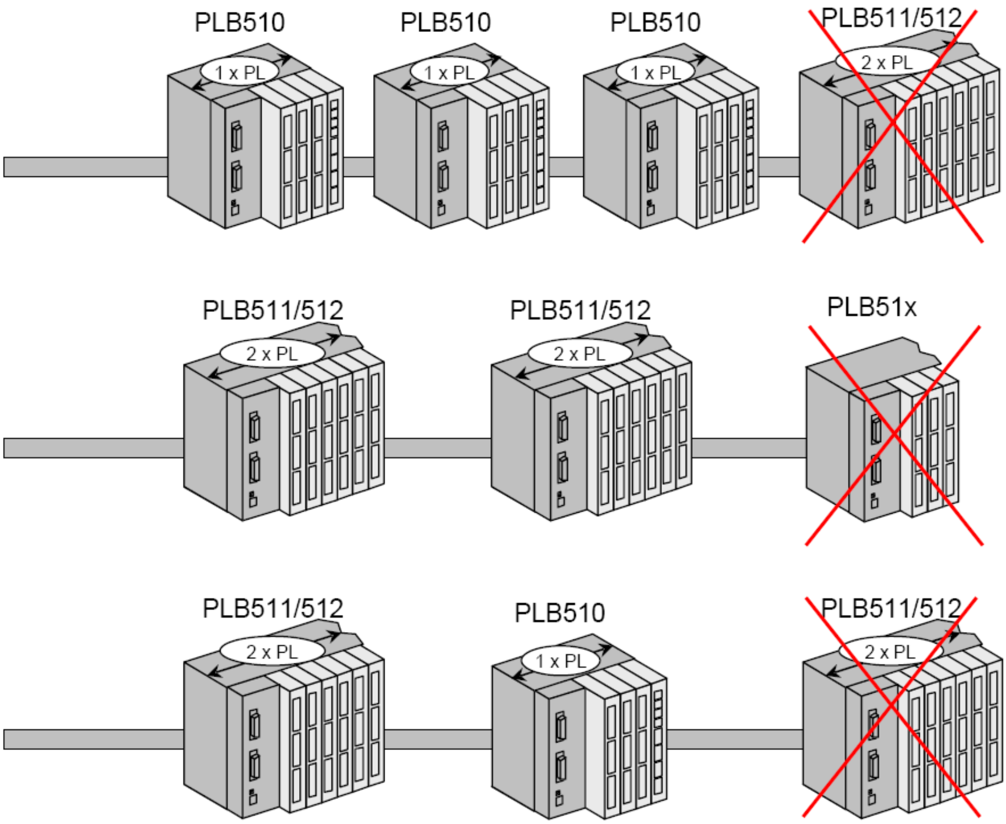
Address ranges when using two PLB 511 or PLB 512 modules:



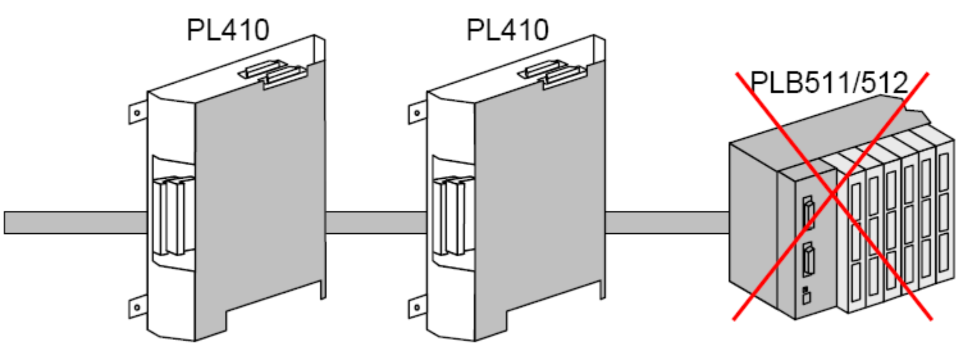
Examples of possible combinations:



Examples of combinations that are not possible:

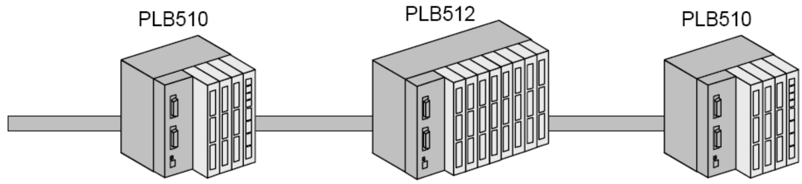


A combination of PL 410 and PLB 511 or PLB 512 is not possible:

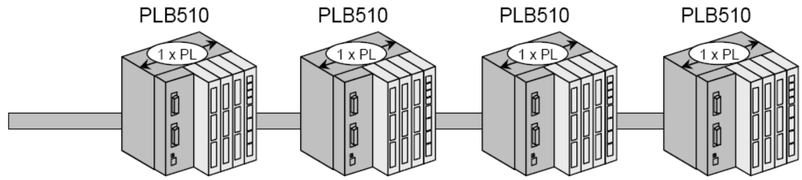


Example

This example uses the following system. It is assumed that the number of the logical PL corresponds to its physical position.



- The following picture shows what the control "sees":



- PLC Module 9007 supplies the value 4 for the "Number of connected PLs."

- The digital inputs are read as follows from the PLC program:

```
PS      K+0             * First PLB510, I64 to I127
CM      9002

PS      K+1             * PLB512 (1st to 4th slot), I192 to I255
CM      9002

PS      K+2             * PLB512 (5th to 8th slot), I256 to I319
CM      9002

PS      K+3             * Second PLB510, I320 to I383
CM      9002
```

- Or realization via loop programming (principle depiction):

```
PS      K+0
PS      K+3             * Number of connected PLs
CM      9007
PL      DG_blockNumber

WHILET
        PS  block
        CM 9002
        INC block
ENDW
```

- The analog inputs can be read out from the PLC program as follows (assuming that both PLB 510s have analog modules in their first slots, and the PLB 512 has them in the 4th and 8th slots)
  The third analog input is read in each case.

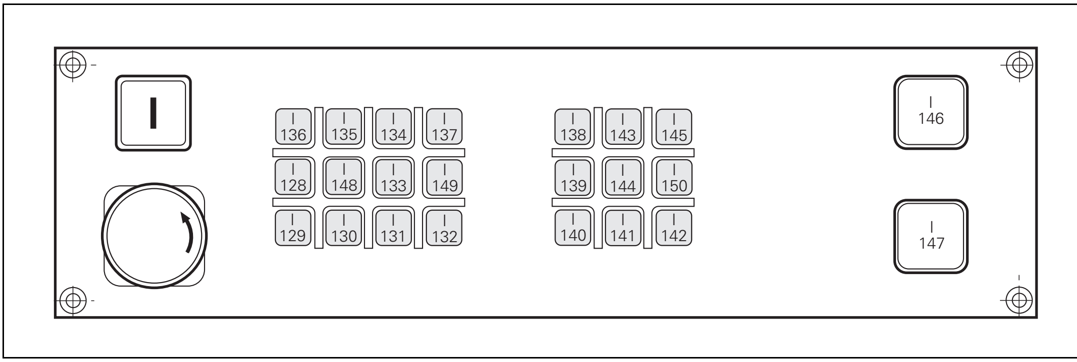
```
    PS   K+0                          * Number of the central
                                        module
    PS   K+3                          * Number of slots 0 to 3
    PS   K+2                          * Analog inputs from 0 to 7
    CM   9138
    PL   DG_Analog_PL510_1_Slot4_AE3  * Analog value

    PS   K+1                          * Number of the central
                                        module
    PS   K+3                          * Number of slots 0 to 3
    PS   K+2                          * Analog inputs from 0 to 7
    CM   9138
    PL   DG_Analog_PL512_Slot4_AE3    * Analog value

*   The following program section would be incorrect since slot 7 does not
*   exist for the control. Slots 4 to 8 are seen as a separate PL. The
*   addressing must be adapted to the PLB 510.
*   PS   K+1                          * Number of the central
                                        module
*   PS   K+7                          * Number of slots 0 to 3
*   PS   K+2                          * Analog inputs from 0 to 7
*   CM   9138
*   PL   DG_Analog_PL512_Slot4_AE3    * Analog value

*   Slots 4 to 8 of the PLB 512 can be accessed with this program section.
*
    PS   K+2                          * Number of the central
                                        module
    PS   K+3                          * Number of slots 0 to 3
    PS   K+2                          * Analog inputs from 0 to 7
    CM   9138
    PL   DG_Analog_PL512_Slot4_AE3    * Analog value

    PS   K+3                          * Number of the central
                                        module
    PS   K+3                          * Number of slots 0 to 3
    PS   K+2                          * Analog inputs from 0 to 7
    CM   9138
    PL   DG_Analog_PL510_4_Slot4_AE3  * Analog value
```

3.24 Machine Operating Panel

MB 420

For machines with up to four axes, HEIDENHAIN offers the MB 420 machine operating panel. It is installed below the TNC operating panel. There is a version of the MB 420 available with a standard set of keys (see connector layout for X46). On the underside of the machine operating panel are two terminal strips bearing the PLC inputs I151 and I152 as well as the PLC outputs O0 to O7.

Assignment of PLC inputs to the keys of the MB 420:



X3: PLC inputs

Pin layout:

Terminal	Pin layout
1	I151
2	I152
3	+24 V

X4: PLC outputs

Pin layout:

Terminal	Pin layout
1	O0
2	O1
3	O2
4	O3
5	O4
6	O5
7	O6
8	O7
9	0 V

X46: PLC inputs and outputs

PLC inputs I128 to I152 and PLC outputs O0 to O7 are on connection X46 of the machine operating panel. The reference potential (PLC) for outputs O0 to O7 is connected to pins 34 and 35.

Pin layout on the MC 42x(B,C), connecting cables and machine operating panel:

> Attention
>
> PLC inputs I128 to I152 must be driven only with the power supply from pins 36 and 37, since this power supply is internally protected (PLC power supply from X44 connection 2).

MC 42x(B,C)		Connecting cable ID 635 877-xx			MB 420	
D-sub Connection (female) 37-pin	Pin layout	D-sub cnnctr. (male) 37-pin		D-sub cnnctr. (female) 37-pin	D-sub connctn. (male) 37-pin	Key
1	I128	1	Gray/Red	1	1	X–
2	I129	2	Brown/Black	2	2	Y–
3	I130	3	White/Black	3	3	Z–
4	I131	4	Green/Black	4	4	IV–
5	I132	5	Brown/Red	5	5	V–
6	I133	6	White/Red	6	6	X +
7	I134	7	White/Green	7	7	Y +
8	I135	8	Red/Blue	8	8	Z+
9	I136	9	Yellow/Red	9	9	IV +
10	I137	10	Gray/Pink	10	10	V+
11	I138	11	Black	11	11	Tool change
12	I139	12	Pink/Brown	12	12	Unlock tool
13	I140	13	Yellow/Blue	13	13	Menu selection
14	I141	14	Green/Blue	14	14	Unlock door
15	I142	15	Yellow	15	15	Chip removal
16	I143	16	Red	16	16	Spindle on
17	I144	17	Gray	17	17	Spindle off
18	I145	18	Blue	18	18	Coolant
19	I146	19	Pink	19	19	NC start
20	I147	20	White/Gray	20	20	NC stop
21	I148	21	Yellow/Gray	21	21	Rapid traverse
22	I149	22	Green/Red	22	22	Retract axis
23	I150	23	White/Pink	23	23	Rinse water jet
24	I151	24	Gray/Green	24	24	Via X3
25	I152	25	Yellow/Brown	25	25	Via X3

MC 42x(B,C)		Connecting cable ID 635 877-xx			MB 420	
D-sub Connection (female) 37-pin	**Pin layout**	**D-sub cnnctr. (male) 37-pin**		**D-sub cnnctr. (female) 37-pin**	**D-sub connctn. (male) 37-pin**	**Key**
26	O0[a]	26	Gray/Brown	26	26	Via X4
27	O1[a]	26	Yellow/Black	27	27	Via X4
28	O2 [a]	28	White/Yellow	28	28	Via X4
29	O3[a]	29	Gray/Blue	29	29	Via X4
30	O4[a]	30	Pink/Blue	30	30	Via X4
31	O5[a]	31	Pink/Red	31	31	Via X4
32	O6[a]	32	Brown/Blue	32	32	Via X4
33	O7[a]	33	Pink/Green	33	33	Via X4
34, 35	0 V (PLC)	34, 35	Brown, Yellow/ Pink	34, 35	34, 35	
36, 37	+24 V (PLC)	36	Violet, White	36, 37	36, 37	
Housing	Ext. shield	Housing	Ext. shield	Housing	Housing	

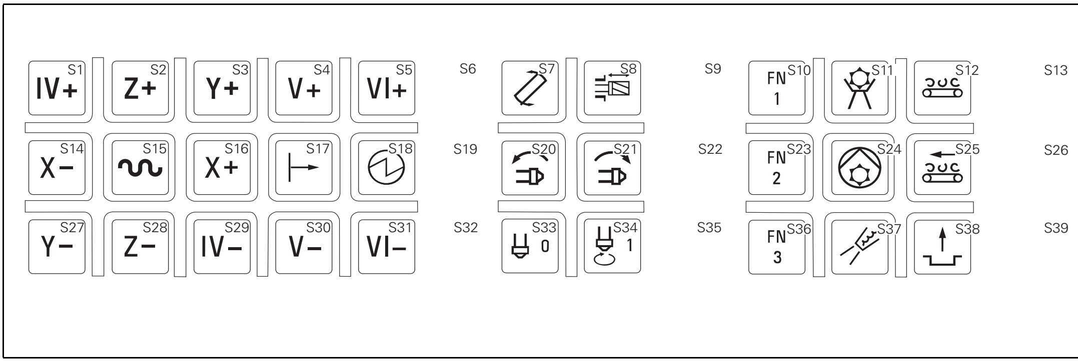
a. Also via X41 (PLC outputs on the MC 42x(B,C))

MB 520

For machines with up to five axes, HEIDENHAIN offers the MB 520 machine operating panel. It is generally installed below the TNC operating panel. On the underside of the machine operating panel there are two terminal strips (X11 and X12) bearing the PLC inputs as well as the PLC outputs.

The MB 520 is an integral component of the TE 535Q keyboard unit.

Machine operating panel: Key assignment



Connecting cable ID 263 954-xx:

X3 D-sub connctn. (male) 37-pin	PLC input	Meaning of the keys	SN/Signal
1	I128	X minus	S14
2	I129	Y minus	S27
3	I130	Z minus	S28
4	I131	IV minus	S29
5	I132	V minus	S30
6	I133	X plus	S16
7	I134	Y plus	S3
8	I135	Z plus	S2
9	I136	IV plus	S1
10	I137	V plus	S4
11	I138	Select tool change	S7
12	I139	Unclamp tool	S8
13	I140	Chip conveyor back	S25
14	I141	Unlock working space	S38
15	I142	Chip conveyor	S12

X3 D-sub connctn. (male) 37-pin	PLC input	Meaning of the keys	SN/Signal
16	I143	Spindle start	S34
17	I144	Spindle stop	S33
18	I145	Coolant, external (M08)	S11
19	I146	NC start	STRT from terminal strip X15
20	I147	NC stop	STP from terminal strip X14
21	I148	Rapid traverse	S15
22	I149	Retract axis	S17
23	I150	Coolant, internal (M07)	S24
24	I151	Machine control voltage ON	STSP from terminal strip X13
25	I152	Additional coolant	S37
26	O0	Machine control voltage ON NC start	From terminal strip X15
27	O1	Machine control voltage ON lamp	From terminal strip X14
28	O2	Machine control voltage ON lamp	From terminal strip X13
29	O3	Vacant	
30	O4	Vacant	
31	O5	Vacant	
32	O6	Vacant	
33	O7	Vacant	
34	0 V		
35	0 V		
36	+24 V		
37	+24 V		

X10: to transfer unit Connecting cable ID 629 663-xx:

X10 D-sub connctn. (male) 37-pin	PLC input	Meaning of the keys	SN/Signal	Color
1	Vacant	VI plus	S5	White
2	Vacant	VI minus	S31	Brown
3	Vacant	Jog spindle to left	S20	Green
4	Vacant	Jog spindle to right	S21	Yellow
5	Vacant	Permissive mode	S18	Violet
6	Vacant	FN1	S10	Brown/ Green
7	Vacant	FN2	S23	Gray
8	Vacant	FN3	S36	Pink
9	Vacant	Not assigned	Vacant input X11/1	Blue
10	Vacant	Not assigned	Vacant input X11/2	Red
11	Vacant	Not assigned	Vacant input X11/3	Black
12	Vacant	Not assigned	Vacant input X11/4	White/ Green
13	Vacant	Not assigned	Vacant input X11/5	Red/Black
14	Vacant	Not assigned	Vacant input X11/6	Yellow/ Black
15	Vacant	Not assigned	Vacant input X11/7	Blue/Black

X11: Vacant inputs

X11	PLC operand	Meaning	Signal
1	Ixxx	Vacant	Vacant input to X10/9
2	Ixxx	Vacant	Vacant input to X10/10
3	Ixxx	Vacant	Vacant input to X10/11
4	Ixxx	Vacant	Vacant input to X10/12
5	Ixxx	Vacant	Vacant input to X10/13
6	Ixxx	Vacant	Vacant input to X10/14
7	Ixxx	Vacant	Vacant input to X10/15
8		+24 V	

X12: Vacant outputs

X12	PLC operand	Meaning	Signal
1	O0	Machine control voltage ON NC start	From terminal strip X15
2	O1	Machine control voltage ON lamp	From terminal strip X14
3	O2	Machine control voltage ON lamp	From terminal strip X13
4	O3	Vacant	Vacant output X3
5	O4	Vacant	Vacant output X3
6	O5	Vacant	Vacant output X3
7	O6	Vacant	Vacant output X3
8	O7	Vacant	Vacant output X3
9	0 V		
10	+24 V		

X13, X14, X15

X13, X14 and X15 are terminal strips that were already wired by HEIDENHAIN before shipping, and do not need to be adapted:

- X13: Terminal strip for the 'machine control voltage ON' key
- X14: Terminal strip for the 'NC stop' key
- X14: Terminal strip for the 'NC start' key

3.25 iTNC Control Panel

X1: Connection of soft keys on the visual display unit with the iTNC control panel

Pin layout:

Connecting element (male) 9-pin	Assignment
1	SL0
2	SL1
3	SL2
4	SL3
5	Do not assign
6	RL15
7	RL14
8	RL13
9	RL12

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Danger

Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

X45/X2: iTNC control panel

Pin layout:

MC 42x(B,C) – X45		Connecting cable ID 635 877-xx			TE – X2
D-sub connctn. (female) 37-pin	Assignment	D-sub connctr. (male) 37-pin		D-sub cnnctr. (female) 37-pin	D-sub connctn. (male) 37-pin
1	RL0	1	Gray/Red	1	1
2	RL1	2	Brown/Black	2	2
3	RL2	3	White/Black	3	3
4	RL3	4	Green/Black	4	4
5	RL4	5	Brown/Red	5	5
6	RL5	6	White/Red	6	6
7	RL6	7	White/Green	7	7
8	RL7	8	Red/Blue	8	8
9	RL8	9	Yellow/Red	9	9
10	RL9	10	Gray/Pink	10	10
11	RL10	11	Black	11	11
12	RL11	12	Pink/Brown	12	12
13	RL12	13	Yellow/Blue	13	13
14	RL13	14	Green/Blue	14	14

MC 42x(B,C) – X45		Connecting cable ID 635 877-xx			TE – X2
D-sub connctn. (female) 37-pin	Assignment	D-sub connctr. (male) 37-pin		D-sub cnnctr. (female) 37-pin	D-sub connctn. (male) 37-pin
15	RL14	15	Yellow	15	15
16	RL15	16	Red	16	16
17	RL16	17	Gray	17	17
18	RL17	18	Blue	18	18
19	RL18	19	Pink	19	19
20	SL0	20	White/Gray	20	20
21	SL1	21	Yellow/Gray	21	21
22	SL2	22	Green/Red	22	22
23	SL3	23	White/Pink	23	23
24	SL4	24	Gray/Green	24	24
25	SL5	25	Yellow/ Brown	25	25
26	SL6	26	Gray/Brown	26	26
27	SL7	26	Yellow/Black	27	27
28	RL19	28	White/Yellow	28	28
29	RL20	29	Gray/Blue	29	29
30	Do not assign	30	Pink/Blue	30	30
31	RL21	31	Pink/Red	31	31
32	RL22	32	Brown/Blue	32	32
33	RL23	33	Pink/Green	33	33
34	Spindle override (wiper)	34	Brown	34	34
35	Feed rate override (wiper)	35	Yellow/Pink	35	35
36	+5 V override potentiometer	36	Violet	36	36
37	0 V override potentiometer	37	White	37	37
Housing	Ext. shield	Housing	Ext. shield	Housing	Housing

X9: USB connection for touchpad

Pin layout

USB connection (female) 4-pin	Assignment
1	+5 V
2	USBP–
3	USBP+
4	GND

3.26 Flat-Panel Display

X3: Connection of screen soft keys

See "iTNC Control Panel" on page 445

X49: BF 120 flat-panel display

Pin layout:

MC 42x(B,C), X49		Connecting cable ID 340 300-xx			BF 120, X2
D-sub cnnctr. (female) 62-pin	**Assign-ment**	**D-sub cnnctr. (male) 62-pin**		**D-sub cnnctr. (female) 62-pin**	**D-sub conntn. (male) 62-pin**
1	0 V	1	Gray/Black	1	1
2	CLK.P	2	Brown/Black	2	2
3	HSYNC	3	Green/Black	3	3
4	$\overline{\text{BLANK}}$	4	Orange/Black	4	4
5	VSYNC	5	Blue/Black	5	5
6	0 V	6	Green/White	6	6
7	R0	7	Orange/White	7	7
8	R1	8	Brown/White	8	8
9	R2	9	Gray/White	9	9
10	R3	10	Blue/White	10	10
11	0 V	11	Violet/White	11	11
12	G0	12	Violet/Brown	12	12
13	G1	13	Violet/Green	13	13
14	G2	14	Violet/Orange	14	14
15	G3	15	Violet/Blue	15	15
16	0 V	16	Red/Gray	16	16
17	B0	17	Red/Brown	17	17
18	B1	18	Yellow/Gray	18	18
19	B2	19	Yellow/Brown	19	19
20	B3	20	Yellow/Green	20	20
21	0 V	21	Vacant	21	21
22	0 V	22	Black/Gray	22	22
23	$\overline{\text{CLP.P}}$	23	Black/Brown	23	23
24	$\overline{\text{HSYNC}}$	24	Black/Green	24	24
25	BLANK	25	Black/Orange	25	25
26	$\overline{\text{VSYNC}}$	26	Black/Blue	26	26
27	0 V	27	White/Green	27	27
28	$\overline{\text{R0}}$	28	White/Orange	28	28
29	$\overline{\text{R1}}$	29	White/Brown	29	29

MC 42x(B,C), X49		Connecting cable ID 340 300-xx			BF 120, X2
D-sub cnnctr. (female) 62-pin	**Assign-ment**	**D-sub cnnctr. (male) 62-pin**		**D-sub cnnctr. (female) 62-pin**	**D-sub conntn. (male) 62-pin**
30	$\overline{R2}$	30	White/Gray	30	30
31	$\overline{R3}$	31	White/Blue	31	31
32	0 V	32	Gray/Violet	32	32
33	$\overline{G0}$	33	Brown/Violet	33	33
34	$\overline{G1}$	34	Green/Violet	34	34
35	$\overline{G2}$	35	Orange/Violet	35	35
36	$\overline{G3}$	36	Blue/Violet	36	36
37	0 V	37	Gray/Red	37	37
38	$\overline{B0}$	38	Brown/Red	38	38
39	$\overline{B1}$	39	Gray/Yellow	39	39
40	$\overline{B2}$	40	Brown/Yellow	40	40
41	$\overline{B3}$	41	Green/Yellow	41	41
42	0 V	42	Vacant	42	42
43	$\overline{\text{DISP. LOW}}$	43	Red/Blue	43	43
44	DISP. LOW	44	Blue/Red	44	44
45	$\overline{\text{DISP.ON}}$	45	Red/Orange	45	45
46	DISP.ON	46	Orange/Red	46	46
47	C0	47	Green/Red	47	47
48	C1	48	Red/Green	48	48
49	C2	49	Orange/Yellow	49	49
50	C3	50	Yellow/Orange	50	50
51	C4	51	Yellow/Blue	51	51
52	C5	52	Blue/Yellow	52	52
53 to 56	Do not assign	53 to 56	Vacant	53 to 56	53 to 56
57 to 62	0 V	57 to 62	Vacant	57 to 62	57 to 62
Housing		Housing		Housing	Housing

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Danger

Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

X149: BF 150 flat-panel display

Pin layout:

MC 42x(B,C), X149		Connecting cable ID 353 545-xx			BF 150, X2
D-sub cnnctr. (female) 44-pin	Assignment	D-sub cnnctr. (male) 44-pin		D-sub cnnctr. (female) 44-pin	D-sub cnnctn. (male) 44-pin
1	A7M	1		1	1
2	A6M	2	White/ Brown	2	2
3	A5M	3	White/ Green	3	3
4	A4M	4	Red/Gray	4	4
5	A3M			5	5
6	CLKM	6	Red/Blue	6	6
7	A2M	7	White/ Orange	7	7
8	A1M	8	Red/Brown	8	8
9	A0M	9	Red/Green	9	9
10	LVDSGND	10	Red/Orange	10	10
11	HWK_GND	11	Orange/Red	11	11
12	HWK0	12	White/Blue	12	12
13	HWK1	13	Blue/White	13	13
14	HWK2	14	White/Gray	14	14
15	HWK3	15	Gray/White	15	15
16	A7P	16		16	16
17	A6P	17	Brown/ White	17	17
18	A5P	18	Green/ White	18	18
19	A4P	19	Gray/Red	19	19
20	A3P			20	20
21	CLKP	21	Blue/Red	21	21
22	A2P	22	Orange/ White	22	22
23	A1P	23	Brown/Red	23	23
24	A0P	24	Green/Red	24	24

MC 42x(B,C), X149		Connecting cable ID 353 545-xx			BF 150, X2
D-sub cnnctr. (female) 44-pin	**Assignment**	**D-sub cnnctr. (male) 44-pin**		**D-sub cnnctr. (female) 44-pin**	**D-sub cnnctn. (male) 44-pin**
25	Not assigned			25	25
to				to	to
30	Not assigned			30	30
31	LVDSGND			31	31
to				to	to
39	LVDSGND			39	39
40	Not assigned			40	40
to				to	to
44	Not assigned			44	44
Housing		Housing		Housing	Housing

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

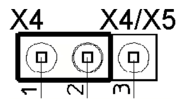
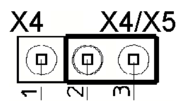
Danger

Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

3.27 BTS 1x0 Monitor/Keyboard Switch Unit

Two monitors (BTS 110: 2 x BF 120, BTS 150: 2 x BF 150) and two TE keyboards can be connected to an MC 42x(B,C) with the BTS 1x0.

The two monitors are always active. Switchover between the two keyboard units is realized by a 24 V switching input on the BTS 1x0.
A jumper on the PCB is used to determine which potentiometer should be active. The jumper is on the upper PCB next to the ID plate.

Jumper setting	Active potentiometers
	Always keyboard 1 (at X4)
	Currently active keyboard

Note

You cannot switch between the two touchpads on the TE 530 with the BTS 1x0. You must connect both touchpads to the MC 42x(B,C) (possibly via the USB hub).

X1, X2, X4, X5 to X7: Monitor and keyboard connections

Refer to the sections "TNC Operating Panel" and "Flat-Panel Display" for the pin layouts of the individual connections.

Connection designation	Monitor/Keyboard
X1	BF 120 or BF 150 input
X2	TE input
X4	First TE output
X5	Second TE output
X6	First BF 120 or BF 150 output
X7	Second BF 120 or BF 150 output

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

X3: Switching between keyboards

Depending on the signal at X3, one of the keyboards at X4 or X5 is activated:

Signal at X3		Active keyboard
Terminal 1	**Terminal 2**	
0 V	0 V	At X4
+24 V	0 V	At X5

X8: Power supply for BTS 1x0

Power supply with double insulation in accordance with EN 61800-5-1.

Pin layout:

Connecting terminal	Assignment
1	+24 V
2	0 V

Current consumption: Max. 100 mA.

3.28 MS110 / MS111 Installation Kit

Sometimes limited space prevents the control and inverter system from being mounted in the same row in a machine's electrical cabinet, meaning that they must be mounted in two separate rows. In other cases the design calls for a second electrical cabinet to house the inverter system. This means that the distribution and arrangement of the components can be very different from case to case.

In order to establish an electrical connection (immune to noise) between the components of the inverter system, the MS 1xx installation kits are needed. The ribbon cables (unit bus, PWM lines, supply bus) from the other components are connected to the MS mounting cases, and shielded round cables of the appropriate lengths connect the MS mounting cases with each other.

In some cases, in order to ensure that the power supply for the fans of the inverters is maintained under all circumstances, it is also necessary to feed 24 V from an external power supply unit to the unit bus (since this is handled by the unit bus).

In most cases the additional 24 V are not needed, since the UVR 1xxD provides enough current for the fans.

If this is the case, and a double-row configuration is used, then two MS 110 mounting cases are necessary (see basic circuit diagram). For the current consumption of the fans, refer to the "Inverter Systems and Motors" Technical Manual. Based on these values you can calculate whether feeding in the additional 24 V is necessary.

For inverter systems with many powerful UM 1xxD power modules, it might be the case that the current provided by the UV(R)1xx power supply unit for the fans of the UMs does not suffice to guarantee safe and reliable operation of the fans. The sum of the currents must not exceed the maximum current provided by the UV(R).

If it is exceeded, then an MS111 must be used in the inverter row where the current consumption is very high.

With the MS 111, the 24 V from an external 24 V power supply unit are fed to the unit bus X79 in order to ensure reliable operation of the fans, and therefore the reliable cooling of the inverters. The 24 V supply that is routed via X79C is interrupted internally at the MS 111.

Note

With the MS 111, an additional power supply unit must be used for the 24 V power supply of the fans.

Double-row configuration

Components and cables for double-row configuration:

Component/Cable	ID
Unit bus cable (shielded, round) with 37-pin D-sub at both ends; max. length: 3 m	ID 664 023-xx
PWM cable (round) with ribbon connector at both ends; max. length: 5 m	ID 664 332-xx
Supply bus cable (round) with ribbon connector at both ends; max. length: 5 m (only necessary if the UV(R) 1xxD is not in the same row as the CC/MC)	ID 361 508-xx
Wire for dc-link (16 mm2, shielded, color: red); max. length: 3 m	ID 655 440-xx
Wire for dc-link (16 mm2, shielded, color: blue); max. length: 3 m	ID 655 438-xx

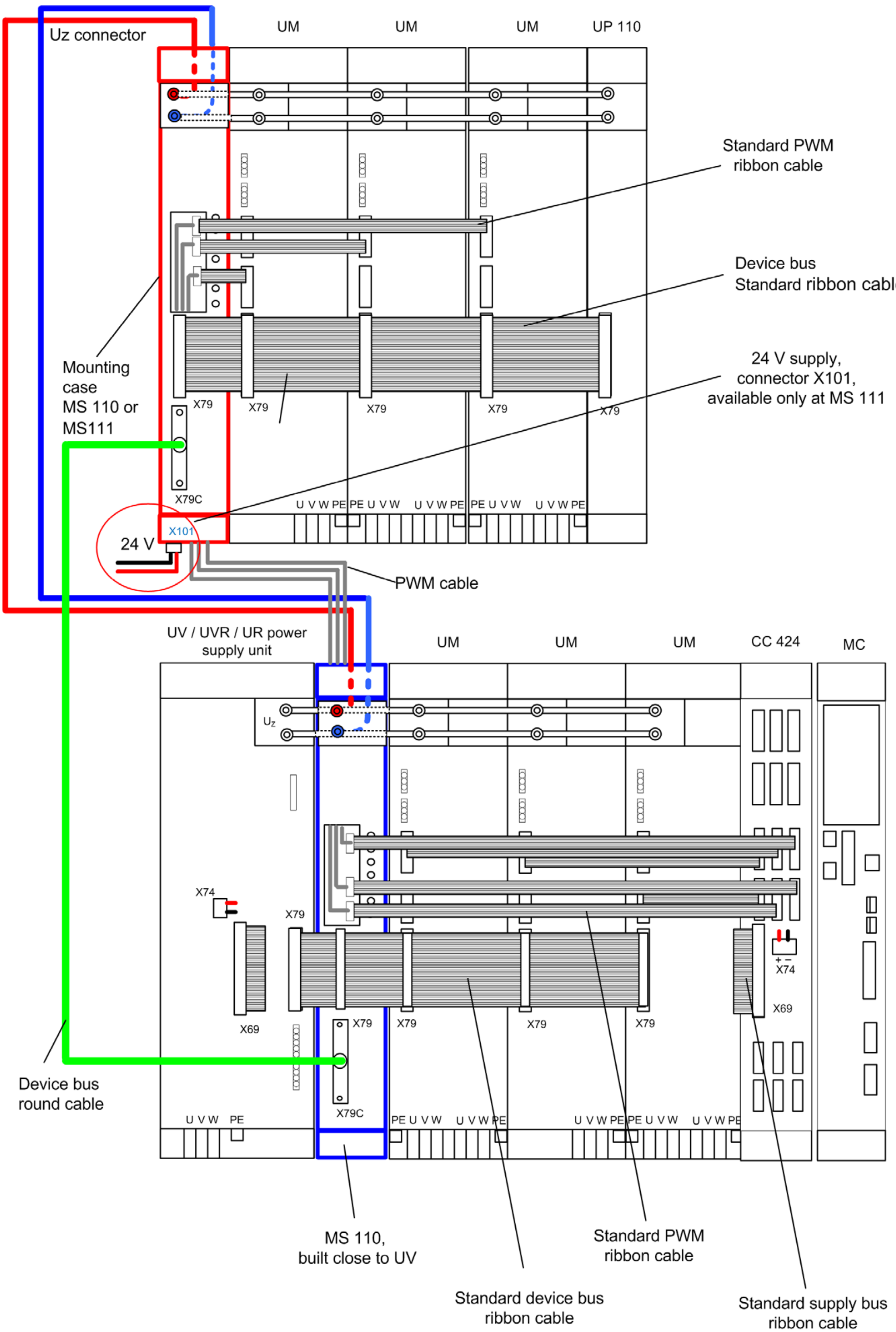
When using a double-row configuration, please keep the following in mind:

- The litz wires used for the DC-link connection of the power modules in the "second row" must not be longer than 3 m.
- Litz wires with 16 mm^2 cross section make a DC-link current of approx. 67 A possible. In a regenerative system, this results in approx. 35 kW of continuous power for the system connected by these wires.
- In a nonregenerative system the resulting maximum power is approx. 25 kW.
- Use fast-acting semiconductor fuses for protection of the UV(R) 1xxD on the primary side.
- The length of the unit bus ribbon cable must not exceed 1 m!
- If necessary, place the MS 110 or MS 111 in the "second row" in the center of the UMs.
- When calculating the length of the ribbon cables, make sure to include the module width of the MS110 or MS111.

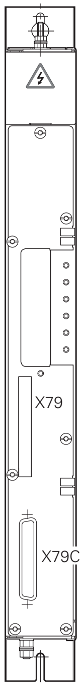
Note

Please also refer to the "Inverter Systems and Motors" Technical Manual.

Basic circuit diagram for double-row configuration:



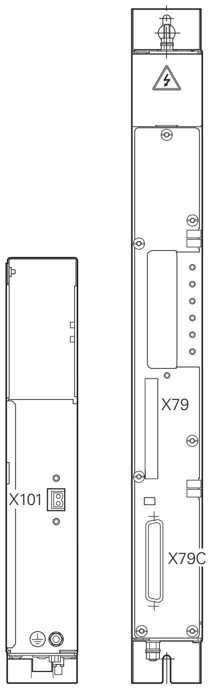
Connection overview for MS 110

MS 110 pin layout	Connector	Function	Page
	DC-link conductor bar	400 V– to 750 V–	
	Screw fastening	For fastening the PWM round cable shields	
	X79	Unit bus (connection for ribbon cable)	459
	X79C	Unit bus (connection for round cable)	460
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

Connection overview for MS 111

MS 111 pin layout	Connector	Function	Page
	DC-link conductor bar	400 V– to 750 V–	
	Screw fastening	For fastening the PWM round cable shields	
	X79	Unit bus (connection for ribbon cable)	459
	X79C	Unit bus (connection for round cable)	460
	LED	+24 V supply for fans is available	
	X101	Connection for +24 V supply for fans	460
	⏚	Protective ground	

Attention

Do not engage or disengage any connecting elements while the unit is under power!

X79: Unit bus

Connection:

<table>
<tr><th>40-pin ribbon connector</th><th>Assignment</th><th></th></tr>
<tr><td>1a to 3b</td><td>0 V *1</td><td rowspan="7">These voltages must
not be linked with other voltages
(only basic insulation)!</td></tr>
<tr><td>4a</td><td>+24 V *1</td></tr>
<tr><td>4b</td><td>+24 V *1</td></tr>
<tr><td>5a</td><td>+15 V *1</td></tr>
<tr><td>5b</td><td>+24 V *1</td></tr>
<tr><td>6a</td><td>+15 V *1</td></tr>
<tr><td>6b</td><td>+15 V *1</td></tr>
<tr><td>7a to 8b</td><td>Do not assign</td><td></td></tr>
<tr><td>9a</td><td>Reserved (SDA)</td><td></td></tr>
<tr><td>9b</td><td>Do not assign</td><td></td></tr>
<tr><td>10a</td><td>Reserved (SCL)</td><td></td></tr>
<tr><td>10b</td><td>$\overline{\text{ERR.TEMP}}$</td><td></td></tr>
<tr><td>11a</td><td>$\overline{\text{PF.PS}}$</td><td></td></tr>
<tr><td>11b</td><td>0 V</td><td></td></tr>
<tr><td>12a</td><td>$\overline{\text{RES.PS}}$</td><td></td></tr>
<tr><td>12b</td><td>0 V</td><td></td></tr>
<tr><td>13a</td><td>$\overline{\text{PWR.OFF}}$</td><td></td></tr>
<tr><td>13b</td><td>0 V</td><td></td></tr>
<tr><td>14a</td><td>5 V FS (spindle enable)</td><td></td></tr>
<tr><td>14b</td><td>0 V</td><td></td></tr>
<tr><td>15a</td><td>5 V FA (axis enable)</td><td></td></tr>
<tr><td>15b to 16b</td><td>0 V</td><td></td></tr>
<tr><td>17a and 17b</td><td>–15 V</td><td></td></tr>
<tr><td>18a and 18b</td><td>+15 V</td><td></td></tr>
<tr><td>19a to 20b</td><td>+5 V</td><td></td></tr>
</table>

Danger

The interface complies with the requirements of EN 61800-5-1 for low voltage electrical separation (except for 1a to 6b).

X79C: Unit bus

Connection:

Round cable connector 37-pin	Assignment	
17 -19, 35 -37	0 V *1	These voltages must not be linked with other voltages (only basic insulation)!
16	+24 V *1	
34	+24 V *1	
15	+15 V *1	
33	+24 V *1	
14	+15 V *1	
32	+15 V *1	
12, 13	Do not assign	
11	Reserved (SDA)	
30, 31	Do not assign	
29	Reserved (SCL)	
10	$\overline{\text{ERR.TEMP}}$	
28	$\overline{\text{PF.PS}}$	
9	0 V	
27	$\overline{\text{RES.PS}}$	
0	0 V	
26	$\overline{\text{PWR.OFF}}$	
25	0 V	
7	5 V FS (spindle enable)	
6	5 V FA (axis enable)	
5, 23, 24	0 V	
4, 22	–15 V	
3, 21	+15 V	
1, 2, 20	+5 V	

Danger

The interface complies with the requirements of EN 61800-5-1 for low voltage electrical separation (except for 1a to 6b).

X101: Power supply

Pin layout:

Connecting terminals	Assignment
+	+24 V NC
–	0 V NC

3.29 USB Interface

X141, X142

Pin layout:

USB connection (female) 4-pin	Assignment
1	+5 V
2	USBP–
3	USBP+
4	GND

As of software version 340 49x-04 it is possible to connect a mouse or touchpad (USB mouse device) to each of the two USB ports (X141, X142), and to operate them simultaneously.

If you want to use this possibility so as to operate the control with two USB mouse devices from two separate locations, the PLC must ensure that only one of the two devices is active at any one time. This could be realized by connecting USB hubs with switchable supply voltage, for example.
Also, a USB mouse can be disconnected and reconnected again during operation of the control. The USB mouse is automatically recognized without needing to reboot the control.

Note

The interface complies with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Danger

Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

Particular to the MC 422C DP:

- A second USB connection (X144) is located on the bottom of the MC 422C DP's housing.
- X143, X144 are only displayed on the HeROS operating-system level and can only be used on the HeROS operating-system level. They have the same properties as the USB ports of an MC 422C single-processor version.
- X141, X142 are available in Windows and HeROS. The properties of these two USB ports cannot be influenced with a settings file of the HeROS system.

USB hub

Connections on the USB hub (368 735-01):

Connection designation	Function
X1	24 V power supply
X32	5 V output
X140	USB input (to MC 42x(B,C))
X141	USB output 1
X142	USB output 2
X143	USB output 3
X144	USB output 4

Note

The interfaces comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)."

Danger

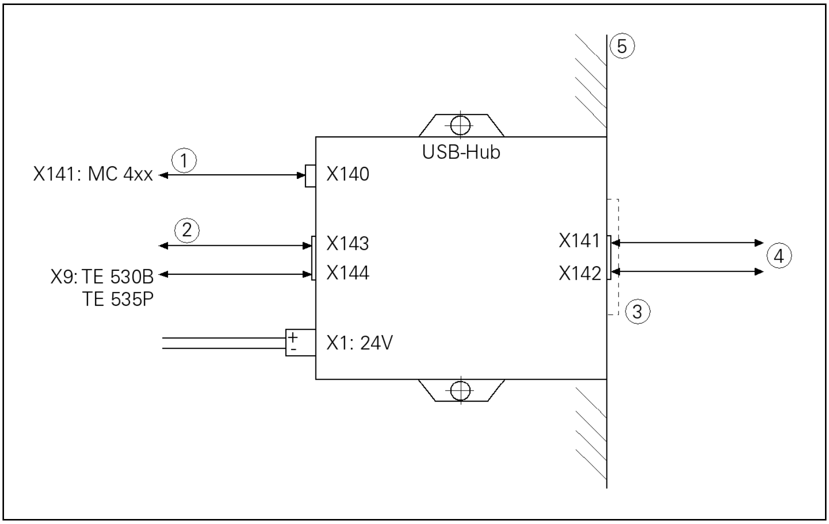
Only units that comply with the requirements of EN 61800-5-1 for "protective extra-low voltage (PELV)" may be connected.

USB hub for operating panel

A USB hub for integration in the control operating panel has been available since October 2006. The touchpad of the TE 530B or TE 535Q and other USB units can be connected to this hub. A 24 V power supply is needed for the USB hub.

The hub can be installed in the operating panel in such a way that the USB sockets X141 and X142 can be accessed from outside the operating panel. You can protect these sockets against splash water with the optional cover cap (ID 508 921-01). The cover is secured with a double-sided adhesive tape (shaped part that is included) and adjusted using adjustment pins. The positions of the holes necessary for the adjustment pins and the cutout for X141/X142 in the operating panel can be seen in the dimension drawing.

Connection overview:



- 1: USB connecting cable 354 770-xx. For lengths starting from 6 m, a USB connecting cable with amplifiers (624 775-xx) must be used.
- 2: USB connecting cable 354 770-xx. Connection to X143 or X144. The other connection remains vacant. For lengths starting from 6 m, a USB connecting cable with amplifiers (624 775-xx) must be used.
- 3: Optional cover cap as protection against splash water 508 921-01
- 4: USB memory stick, USB cable, etc. to external devices (for any uses).
- 5: Operating panel

3.30 CML 110 Capacitor Module

> **Danger**
>
> Before service or maintenance work, you must ensure that the CML 110 has been completely discharged.

The CML 110 (Capacitor Module Low Voltage) for realizing the LIFTOFF function in case of a powerfail has been available for the iTNC 530 with CC 424(B) since April 2006.

The LIFTOFF function can protect workpieces and tools from damage, see page 1092. When a power failure occurs and the LIFT OFF function is active, the iTNC 530 tries to lift the tool off of the contour, using the residual energy of the dc-link. In this case, the various enablings for operating the control system must be maintained during the LIFTOFF. The CML 110 ensures the 24 V supply for this.

Connection

The CML 110 capacitor module is connected via X1 parallel to the 24 V power supply (++/- -).

The two ++ and - - terminals of the CML 110 are each connected to each other internally. This makes it possible to connect several CMLs 110 in parallel without needing to use additional external terminals.

Utilizability

Calculating the utilizability of the CML 110:

A successful LIFTOFF mainly depends on sufficient energy being available in the dc-link of the inverter system. Generally it suffices if the energy in the dc-link is available for the duration of one second. The 24 V supply must also be ensured for precisely this time. The following formula can be used to check this:

$$t = R_L \times C \times \ln (U_O/U_C)$$

where:

t = time until U_C is reached
R_L = ohmic load of the consumers
C = capacitance of the CML (for CML 110 = 8.3 F)
ln = natural logarithm
U_O = output load of the power supply unit with which the CML is operated
U_C = lowest voltage at which the consumers still fulfill their function

Example:
During operation at 24 V, a total current of 10 A is required for switching the control components on. This corresponds to an ohmic load of 2.4 ohms. In addition, the voltage for the 24 V components may not sink below 18 V (e.g. switching voltage of the contactors), for example.
This means:
t = 2.4 Ohm x 8.3 F x ln(24 V/18 V)
t = **5.73 s**
If the line voltage fails, then in the best case the voltage will not fall below 18 V until 5.73 seconds have passed. This is significantly longer than 1 second, and so the CML 110 is suitable for LIFTOFF here.

If the capacitance of the CML 110 should not suffice, then you can also switch more than one CML 110 in parallel. However, here you must note that a maximum charging current of 2.4 A per CML 110 is to be expected at switch-on. The full power of the 24 V power supply unit can only be used once all CMLs have finished charging.

3.31 Cable Specifications

Device	ID	Max. bend radius (rigid configuration)	Max. bend radius (frequent flexing)	Cable diameter
Position 1 V_{PP}	298 429-xx, 298 430-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
Position 1 V_{PP}	310 199-xx, 309 783-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Position EnDat	332 115-xx, 323 897-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Position EnDat	313 791-xx, 332 790-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
Speed 1 V_{PP}	289 440-xx, 336 376-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Speed EnDat	336 376-xx, 340 302-xx, 369 502-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Analog outputs	290 110-xx, 290 109-xx	≥ 40 mm	[a]	ø 7.3 mm
TS 220	274 543-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
SE 640, SE 540	310 197-xx, 517 518-xx	≥ 40 mm, ≥ 10 mm	≥ 100 mm, ≥ 50 mm	ø 8 mm ø 4.5 mm
TT 130	335 332-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
HRA 110	270 909-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
HR 130, HR 410 (extension)	281 429-xx	≥ 20 mm	[a]	ø 5.6 mm
HR 4xx (extension)	296 466-xx	≥ 20 mm	[a]	ø 5.6 mm
PLC inputs/outputs	635 877-xx, 635 876-xx	≥ 40 mm	≥ 150 mm	ø 10 mm
PL 4xx	635 879-xx, 635 880-xx	≥ 40 mm	≥ 150 mm	ø 10 mm
PL 51x	371 045-xx, 371 046-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
BF120	340 300-xx	≥ 40 mm	≥ 120 mm	ø 9.7 mm
BF 120 (extension)	312 876-xx	≥ 70 mm	≥ 200 mm	ø 14.3 mm
BF150	353 545-xx	≥ 65 mm	≥ 165 mm	ø 10.9 mm
TE 420, TE 530, MB 420	635 877-xx, 635 878-xx	≥ 40 mm	≥ 150 mm	ø 10 mm
RS-232, 9-pin	355 484-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
RS-232, 9-pin (extension)	366 964-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
RS-232, 25-pin	365 725-xx	≥ 40 mm	≥ 100 mm	ø 7.1 mm
RS-232, 25-pin (extension)	274 545-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
RS-422	355 484-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
USB up to 6 m	354 770-xx	≥ 20 mm	≥ 70 mm	ø 4.5 mm
USB from 6 m (with hub)	624 775-xx	Cable like 354 770-xx, hub: ø ~ 20 mm, length ~ 115 mm		

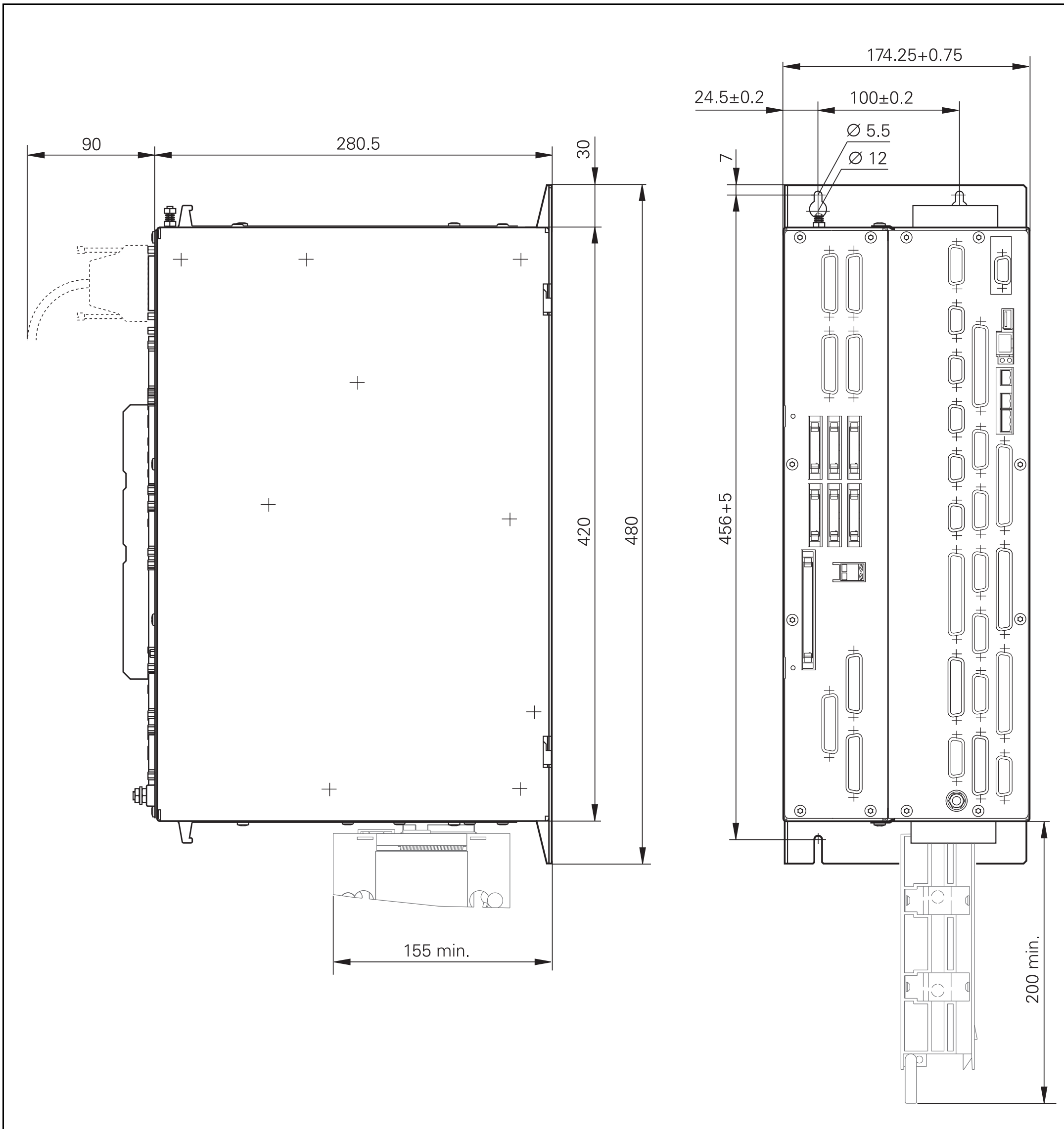
a. Conditionally resistant to frequent flexing and torsion

3.32 Dimensions

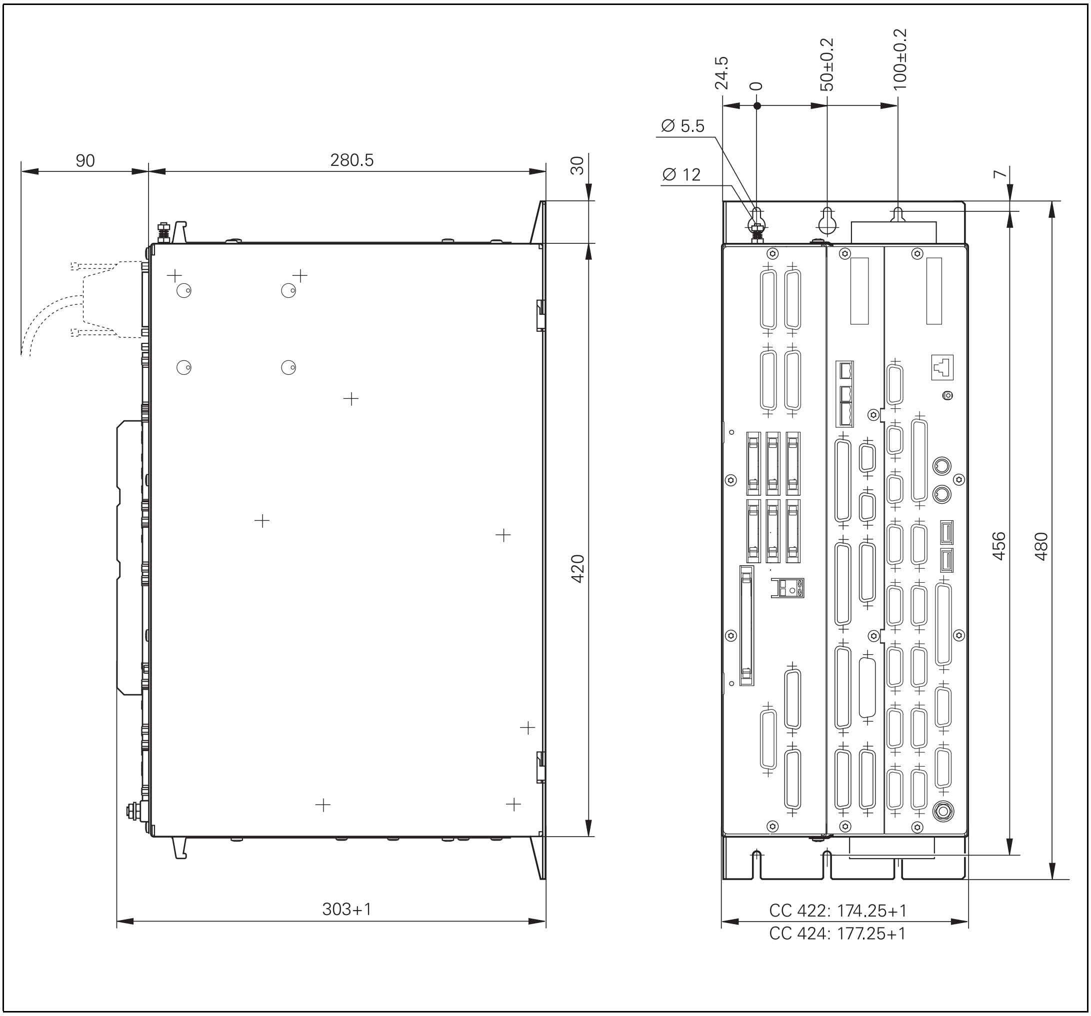
Note

All dimensions are in millimeters [mm].

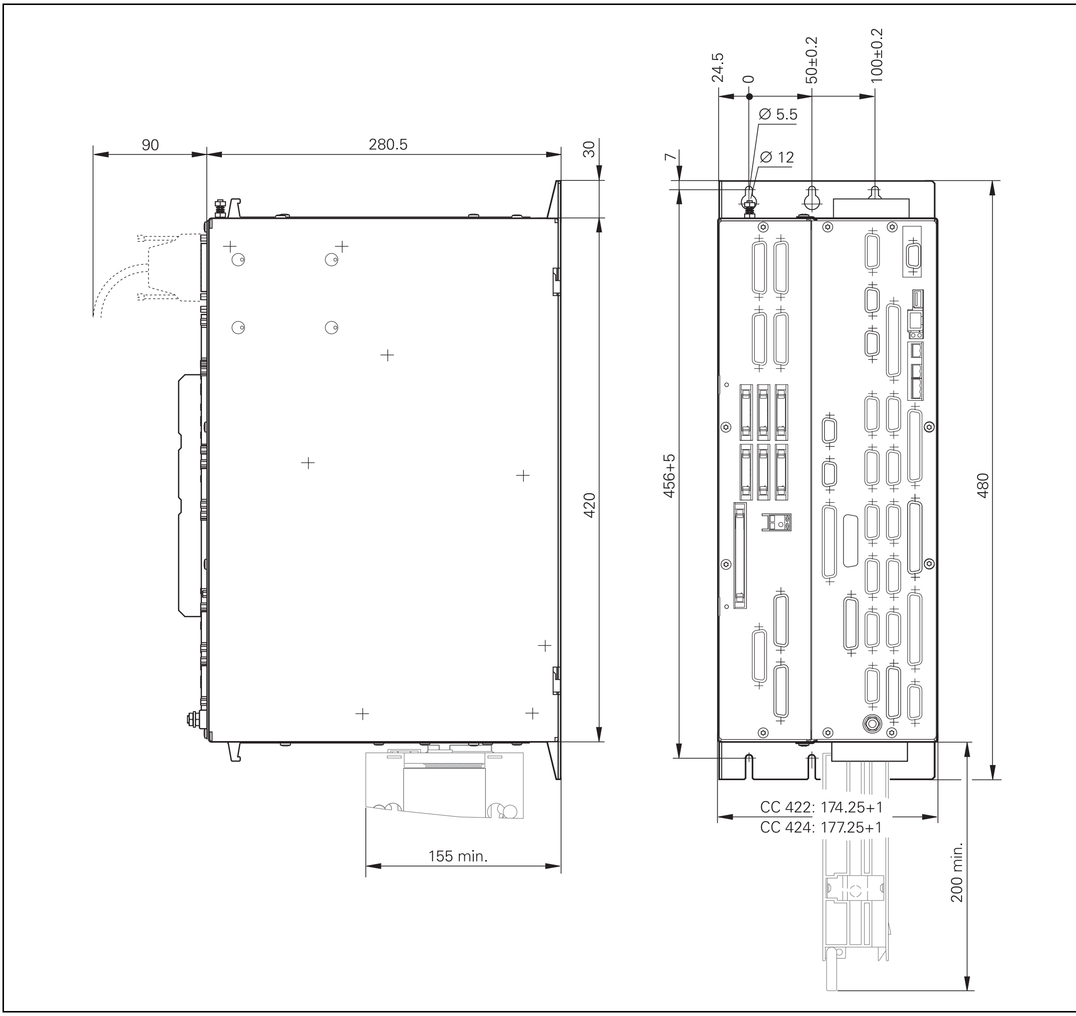
3.32.1 MC 420 and CC 422 with 6 Control Loops



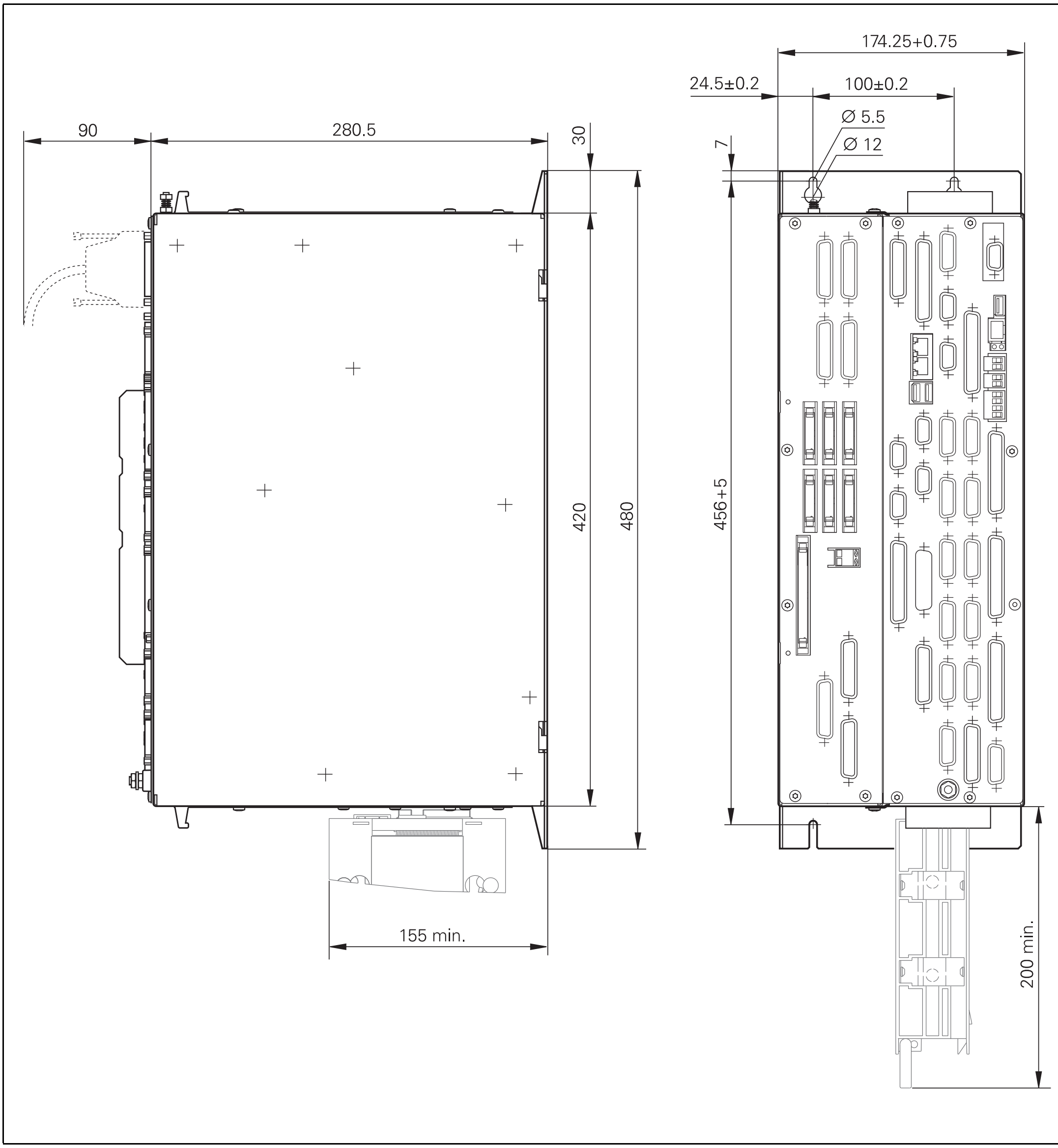
3.32.2 MC 422B / 5 position encoder inputs and CC 422 with 6 control loops



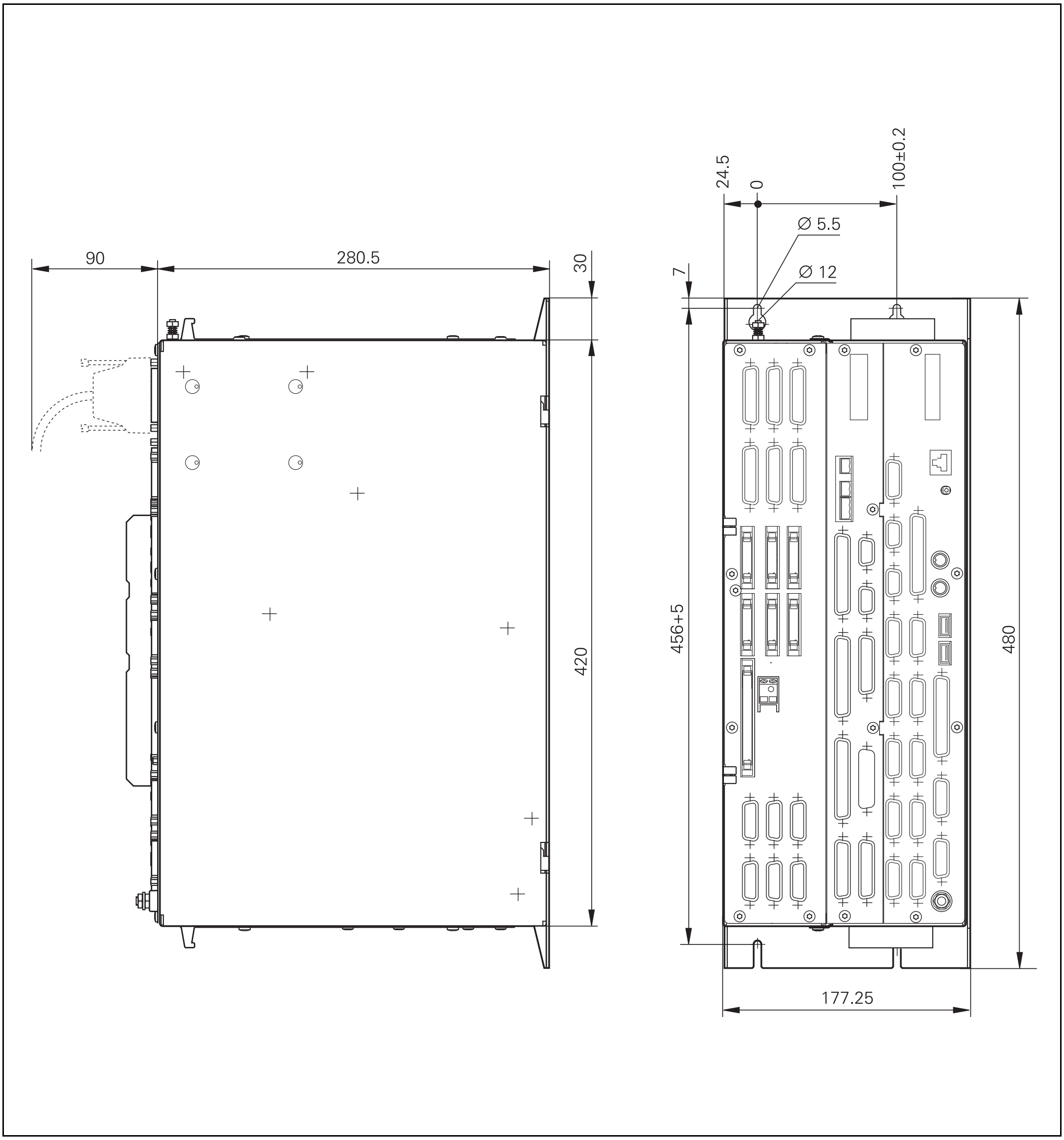
3.32.3 MC 422C / 5 position encoder inputs and CC 422 with 6 control loops



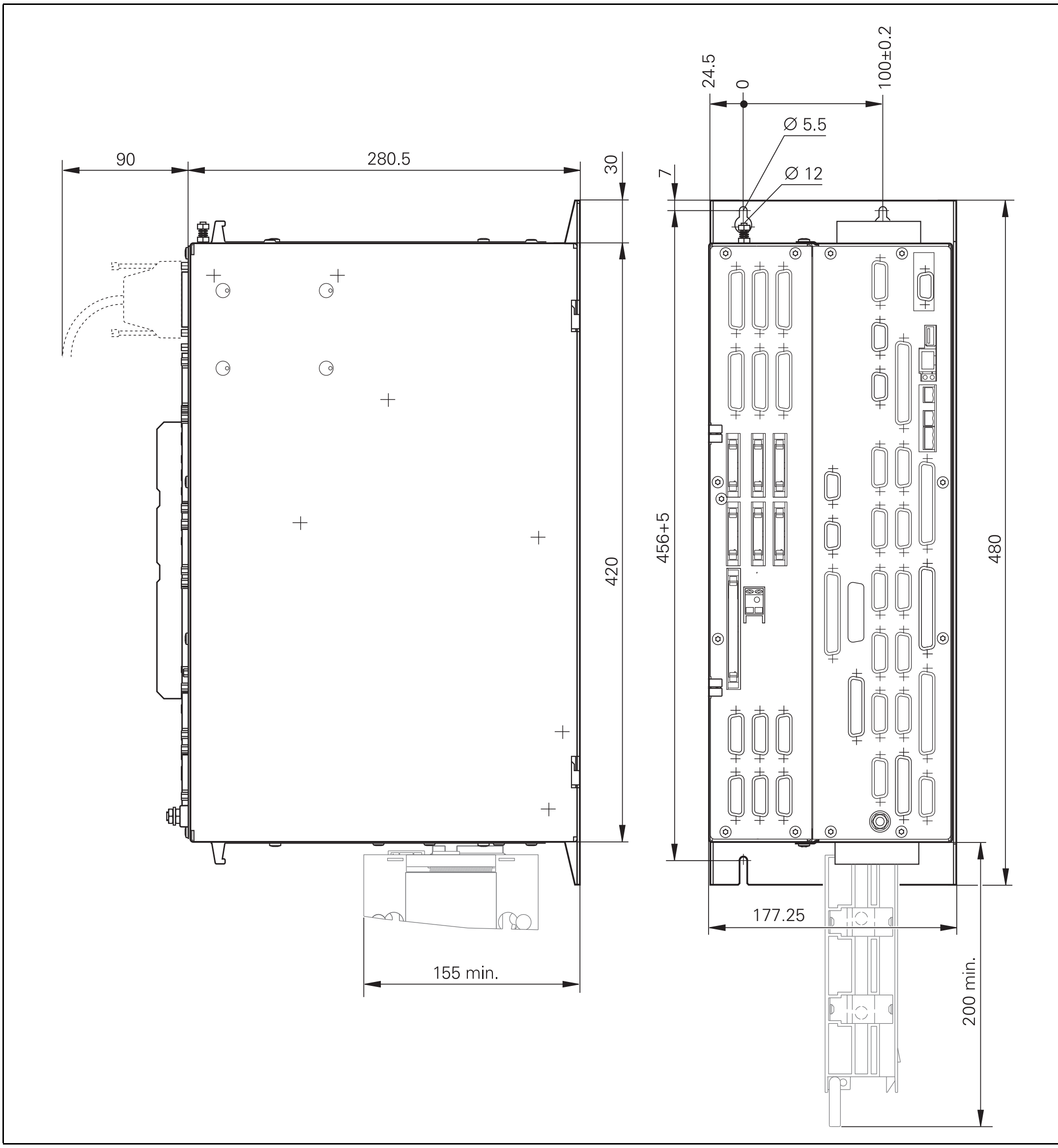
3.32.4 MC 422C DP/CC 422 with 6 control loops



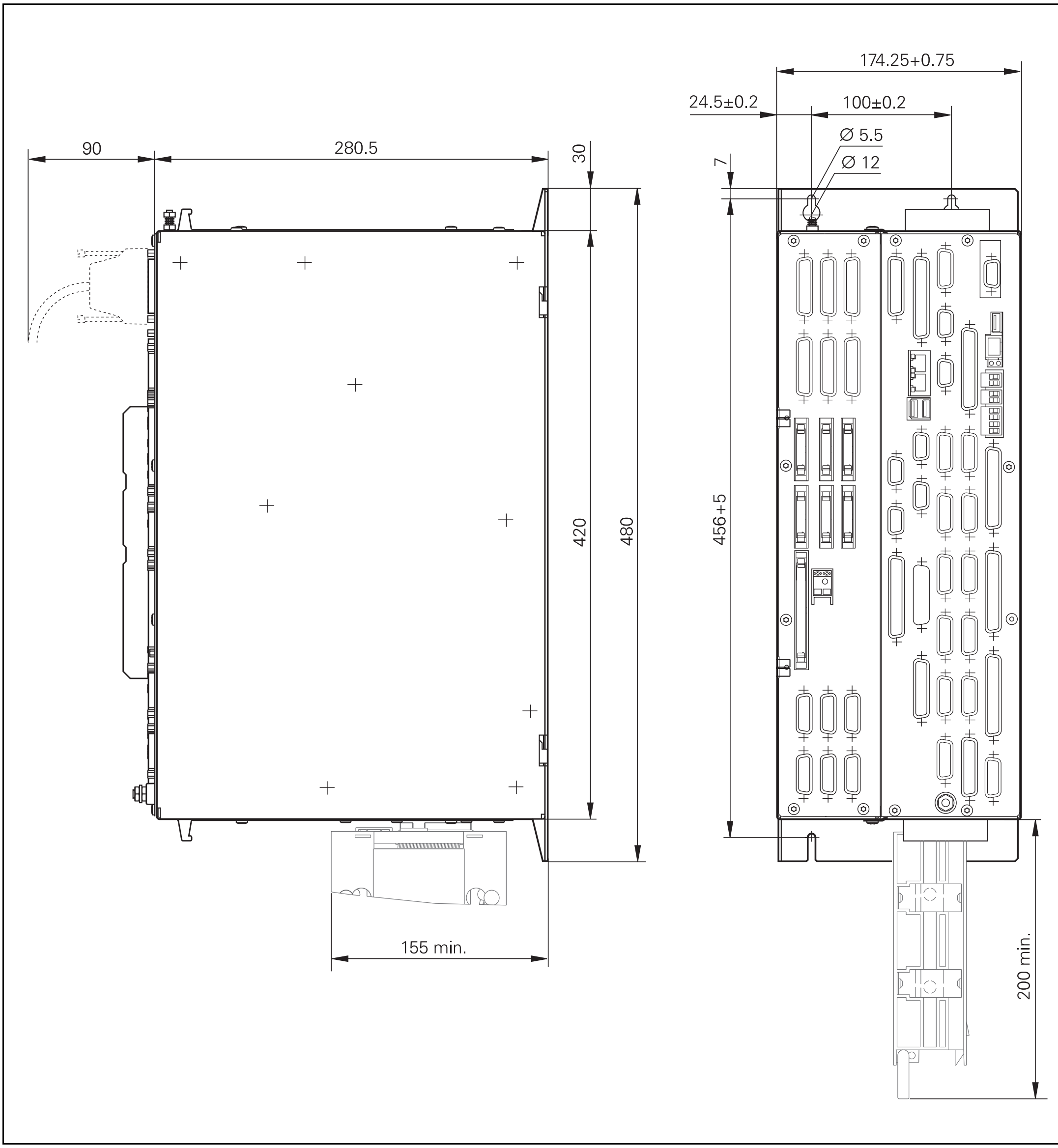
3.32.5 MC 422B/CC 424(B) with 6 control loops



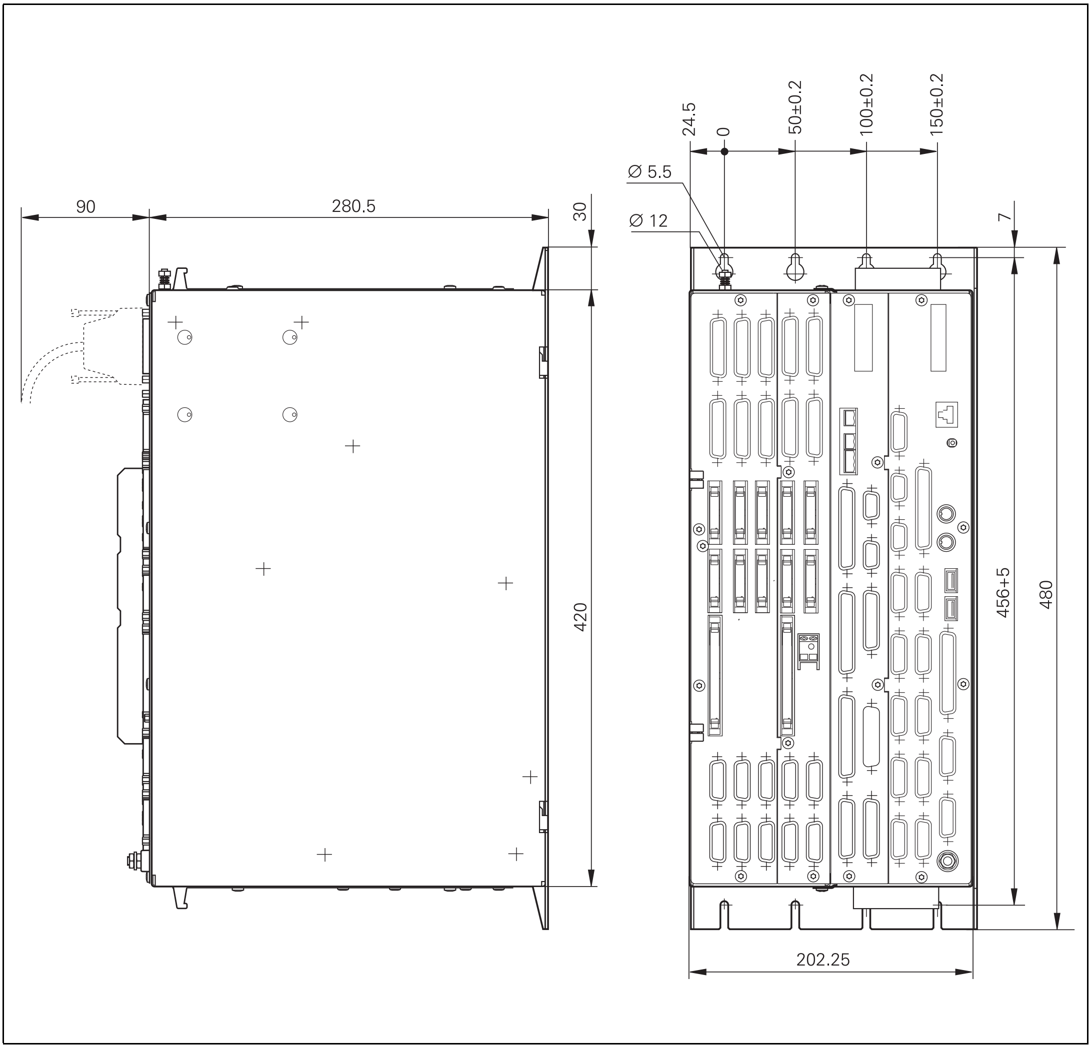
3.32.6 MC 422C/CC 424(B) with 6 control loops



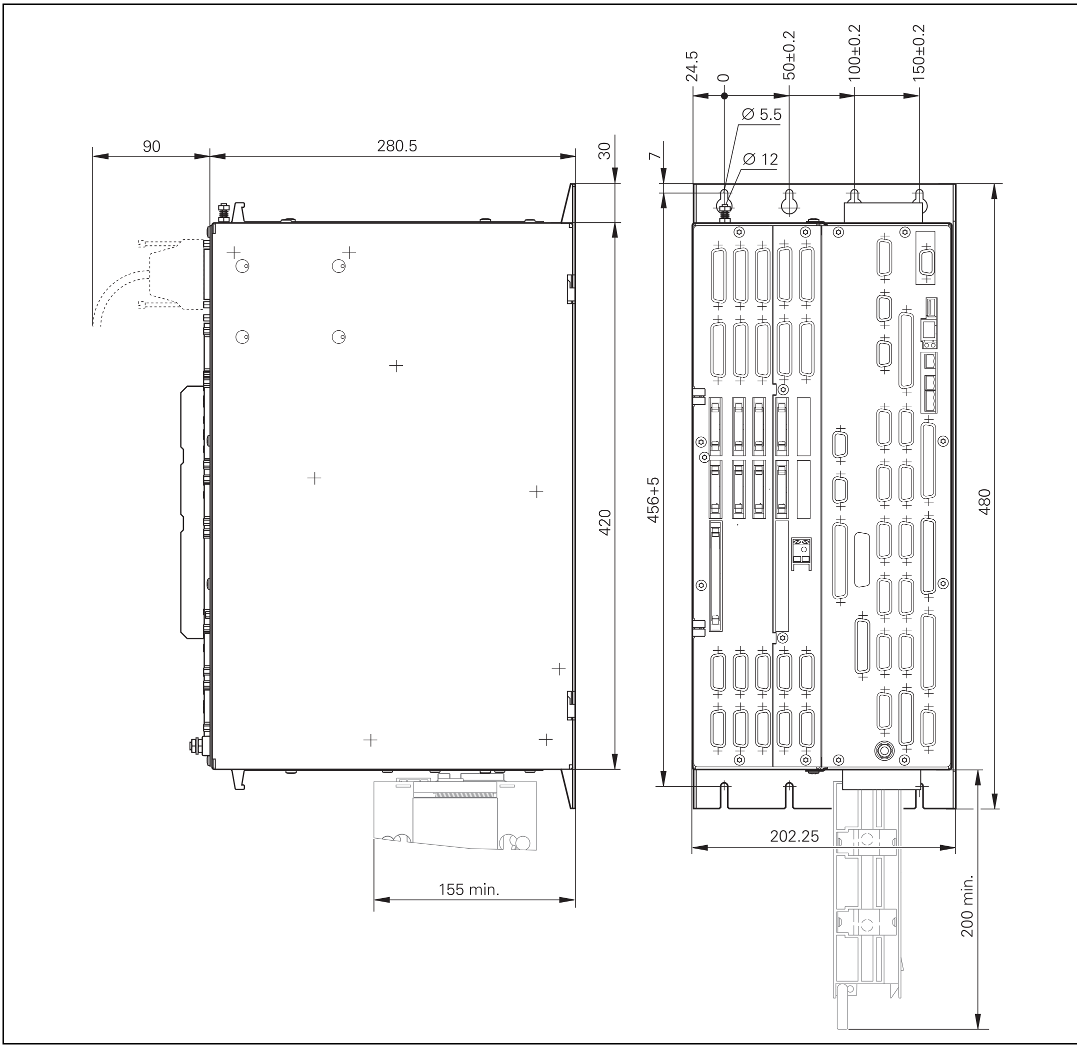
3.32.7 MC 422C DP/CC 424(B) with 6 control loops



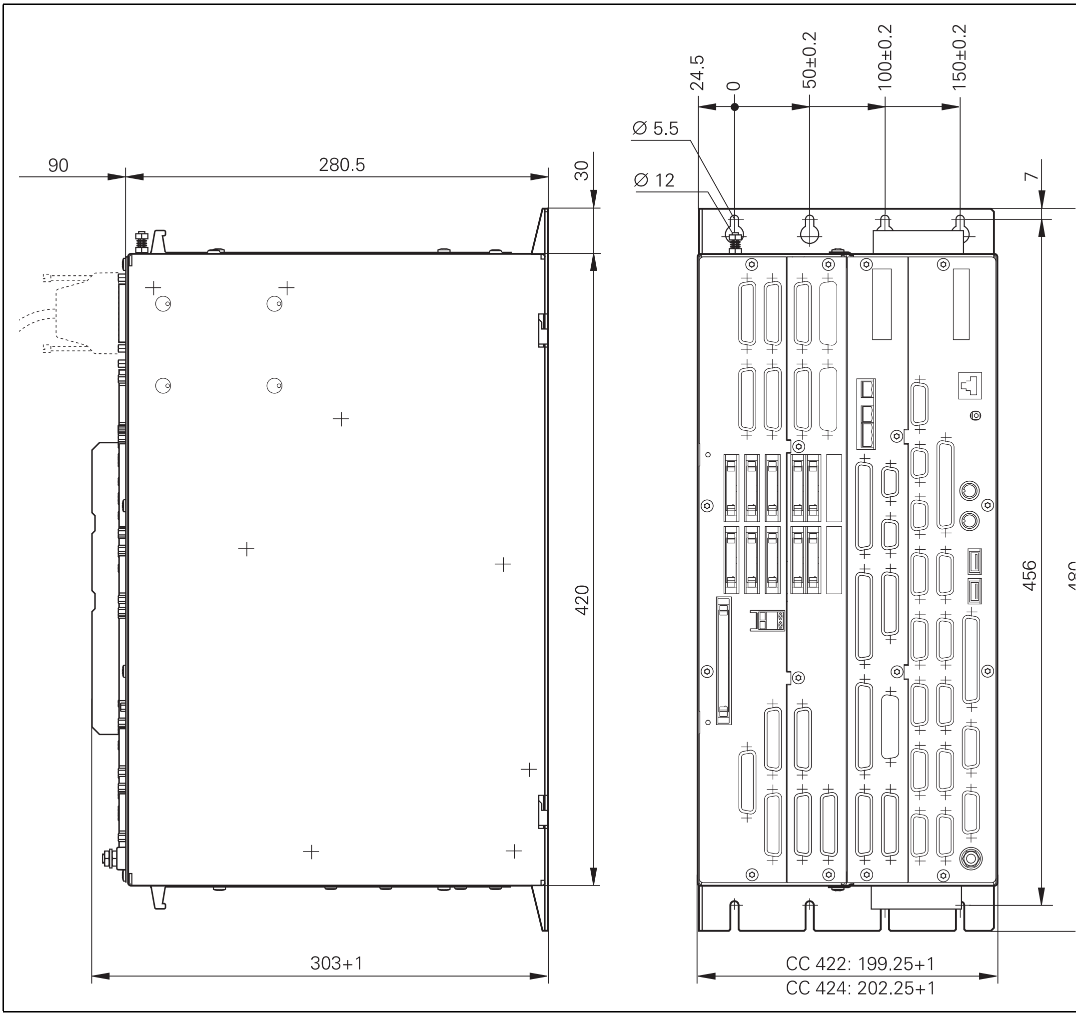
3.32.8 MC 422B/CC 424(B) with 8 control loops



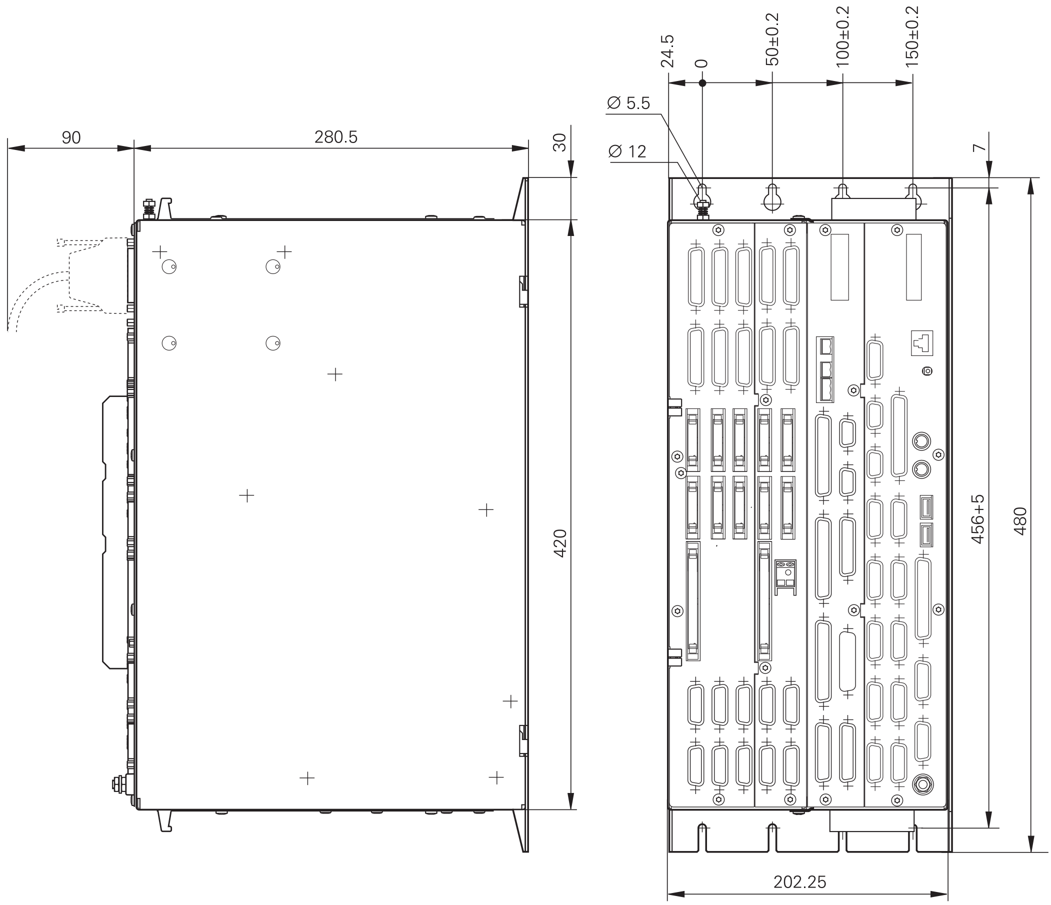
3.32.9 MC 422C/CC 424(B) with 8 control loops



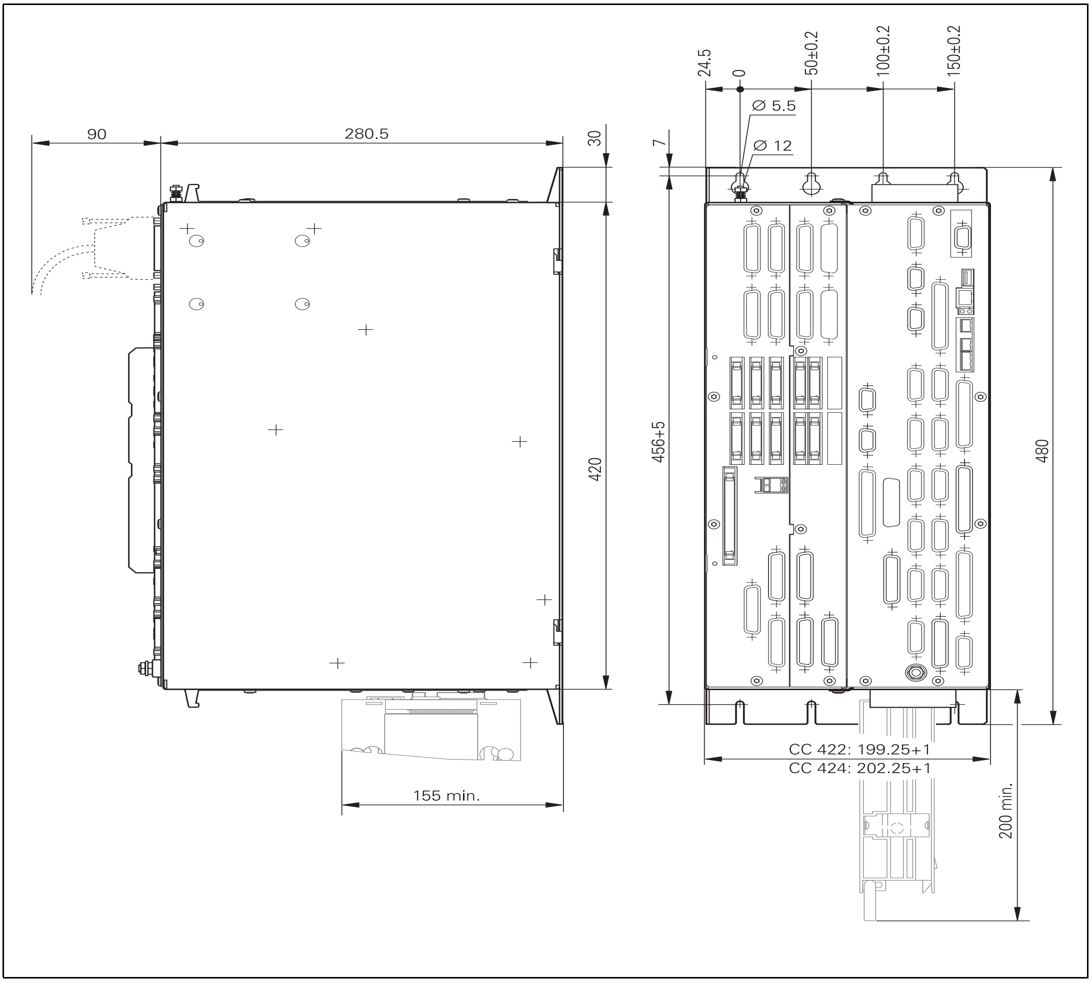
3.32.10 MC 422B / 10 position encoder inputs and CC422 with 10 control loops



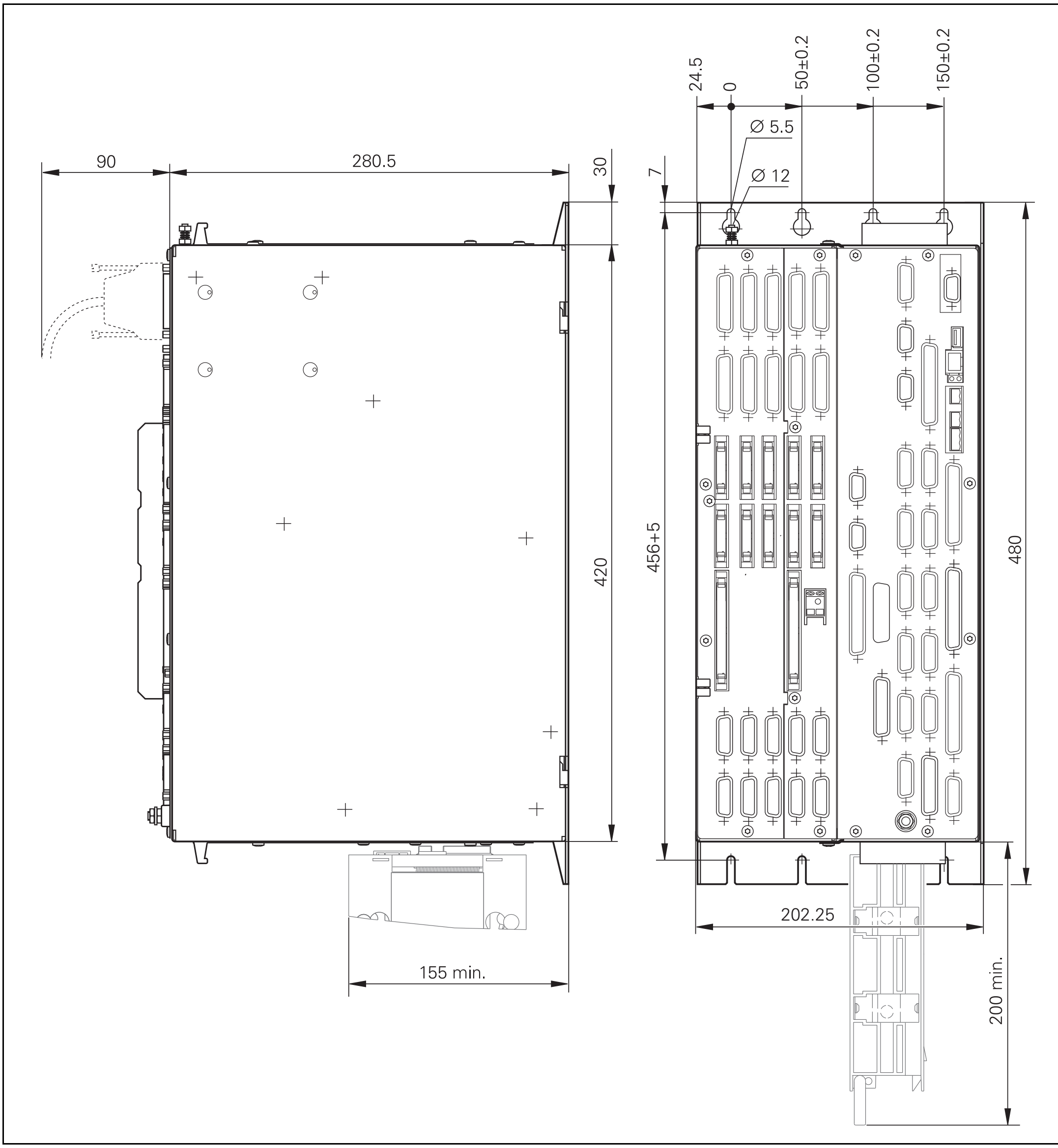
3.32.11 MC 422B/CC 424(B) with 10 control loops



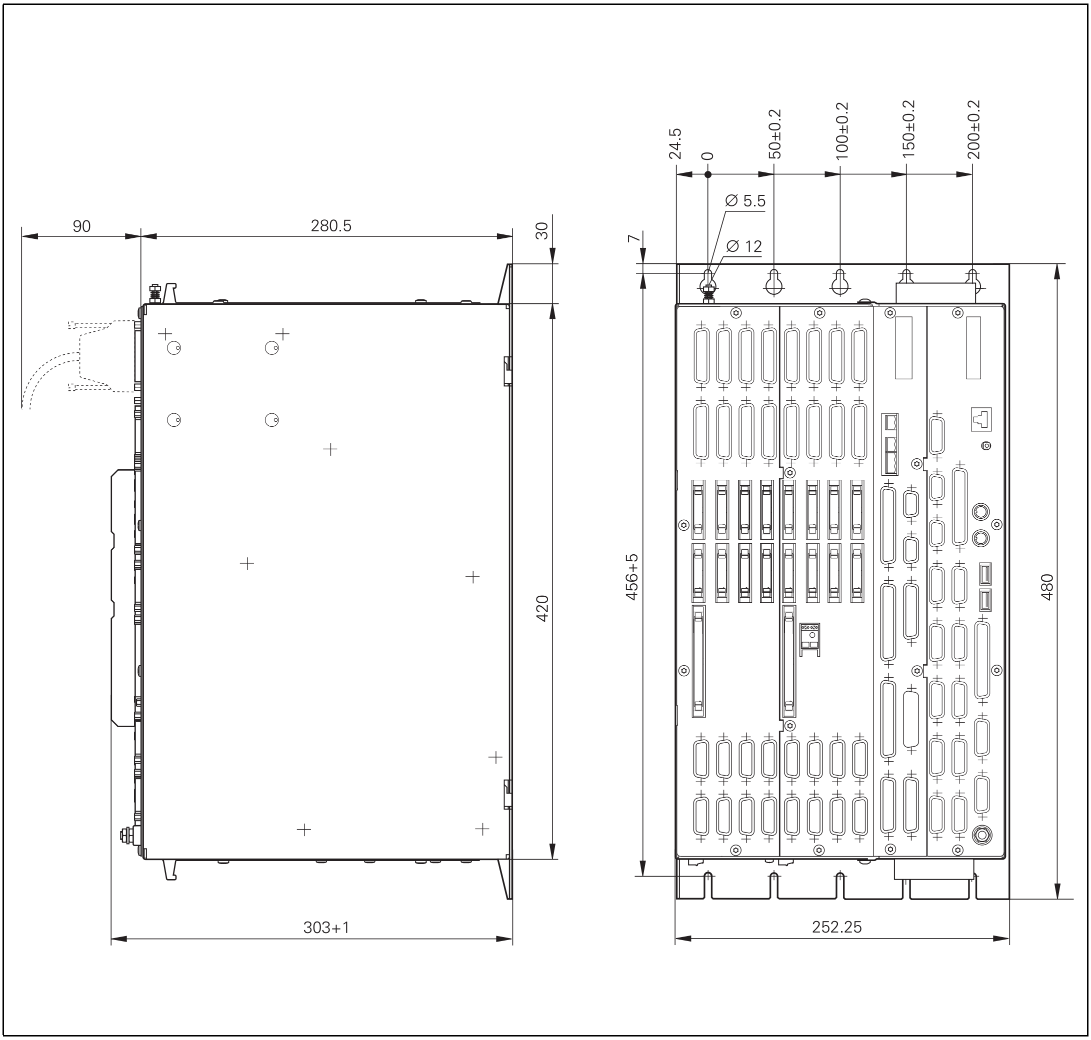
3.32.12 MC 422B / 10 position encoder inputs and CC 422 with 10 or 12 control loops



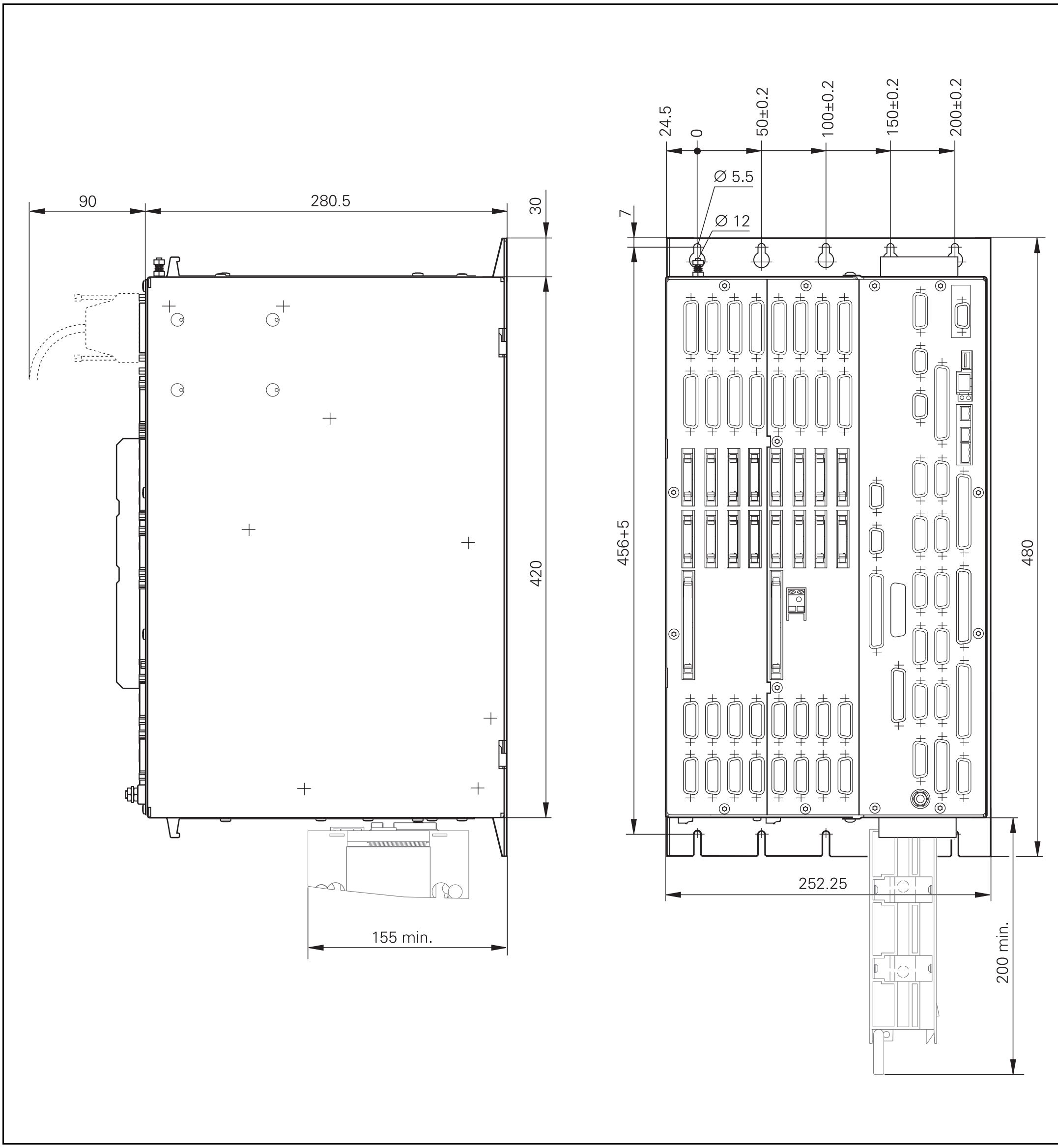
3.32.13 MC 422C/CC 424(B) with 10 control loops



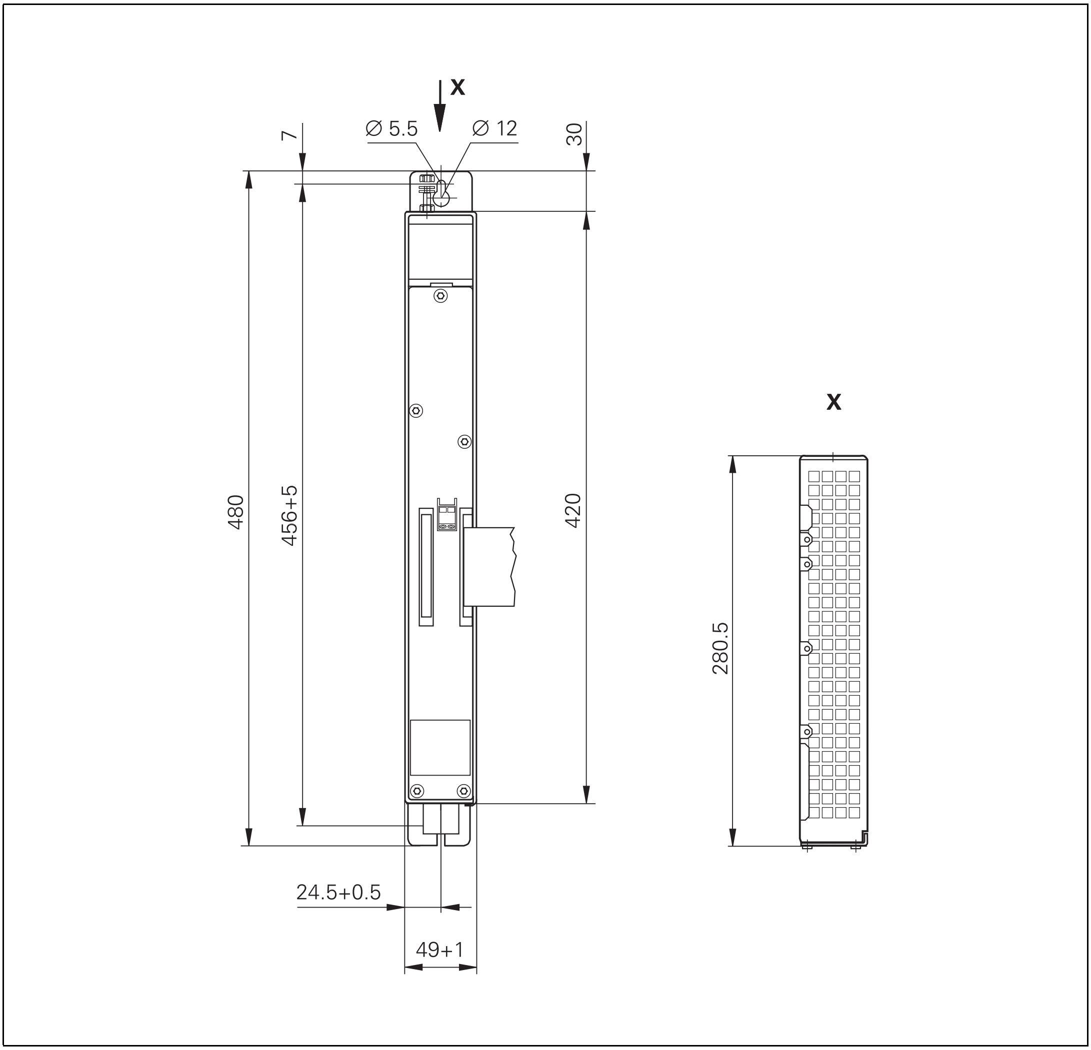
3.32.14 MC 422B/CC 424(B) with 12 and with 14 control loops



3.32.15 MC 422C/CC 424(B) with 12 and with 14 control loops

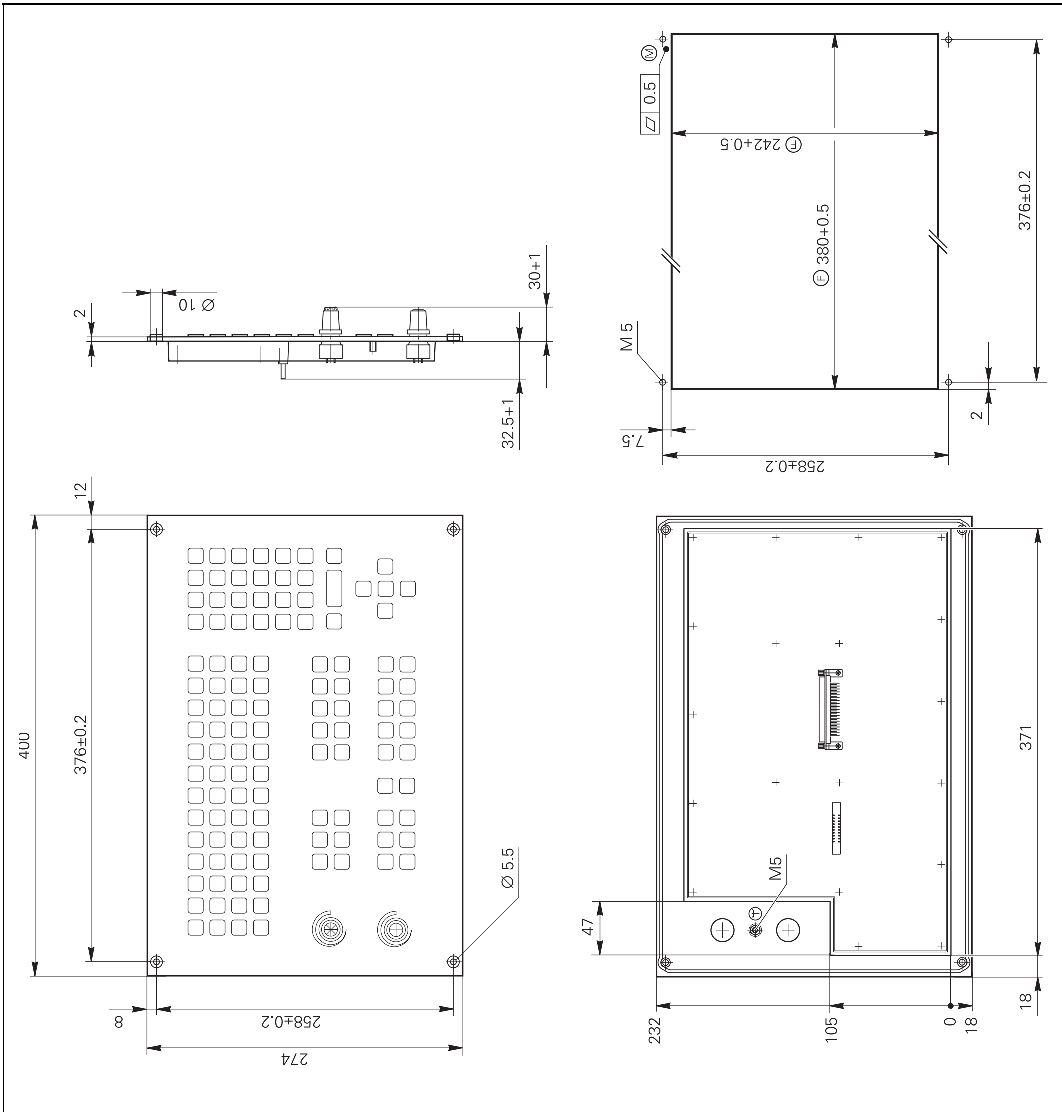


3.32.16 UV 105



3.32.17 TE 420

Weight: 2.4 kg

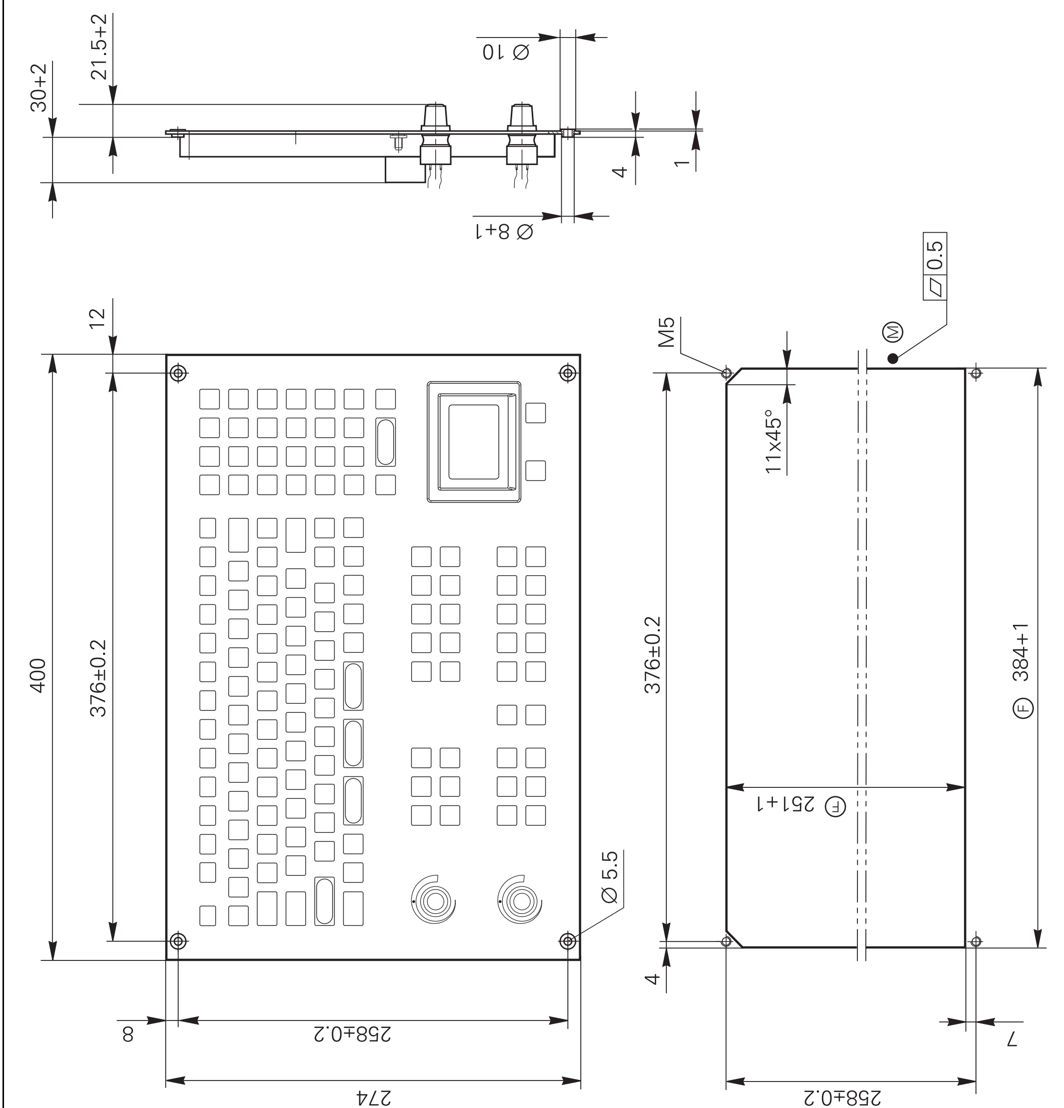


M: Mounting surface

F: Front panel opening

3.32.18 TE 520B / TE 530 / TE 530B

Weight: Approx. 2.4 kg

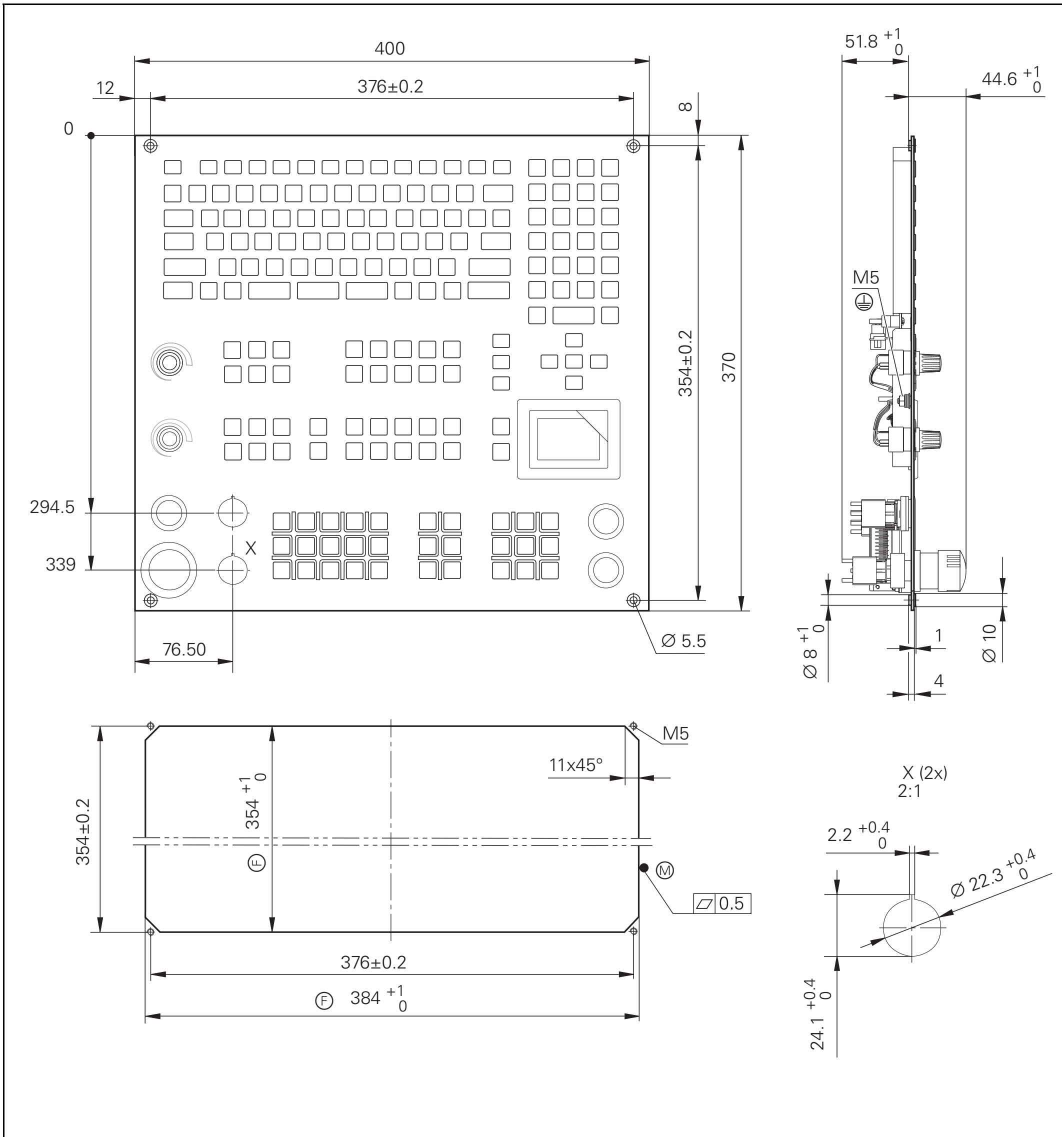


M: Mounting surface

F: Front panel opening

3.32.19 TE 535Q

Weight: Approx. 3.2 kg

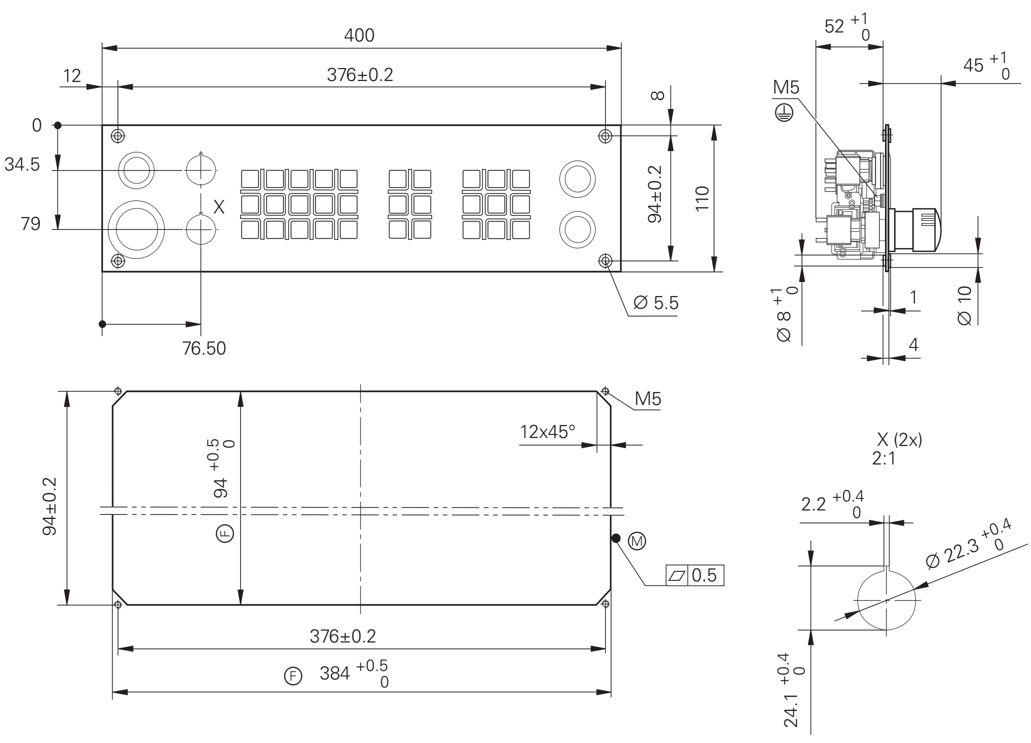


M: Mounting surface

F: Front panel opening

3.32.20 MB 520

Weight: 0.9 kg

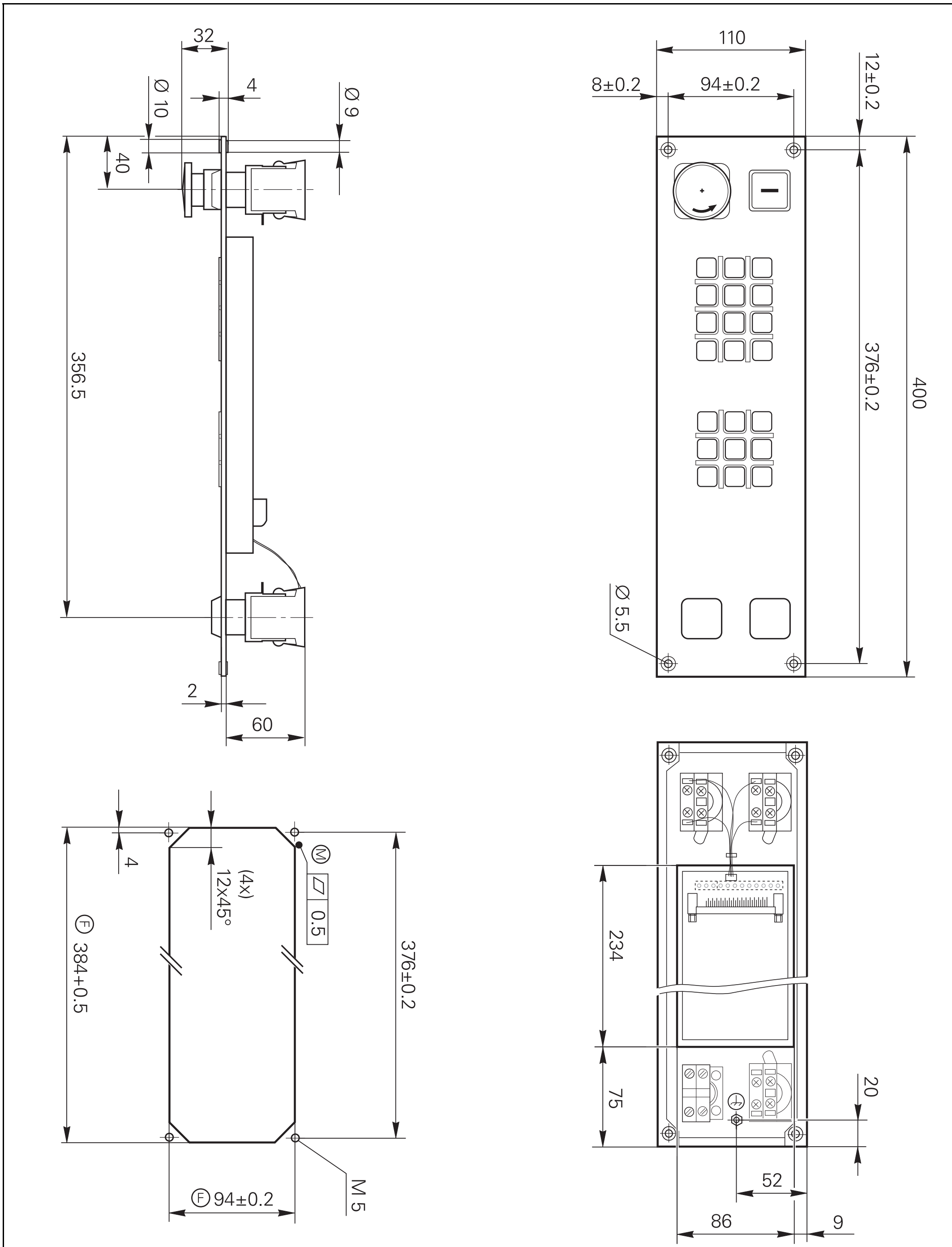


M: Mounting surface

F: Front panel opening

3.32.21 MB 420

Weight: 0.9 kg

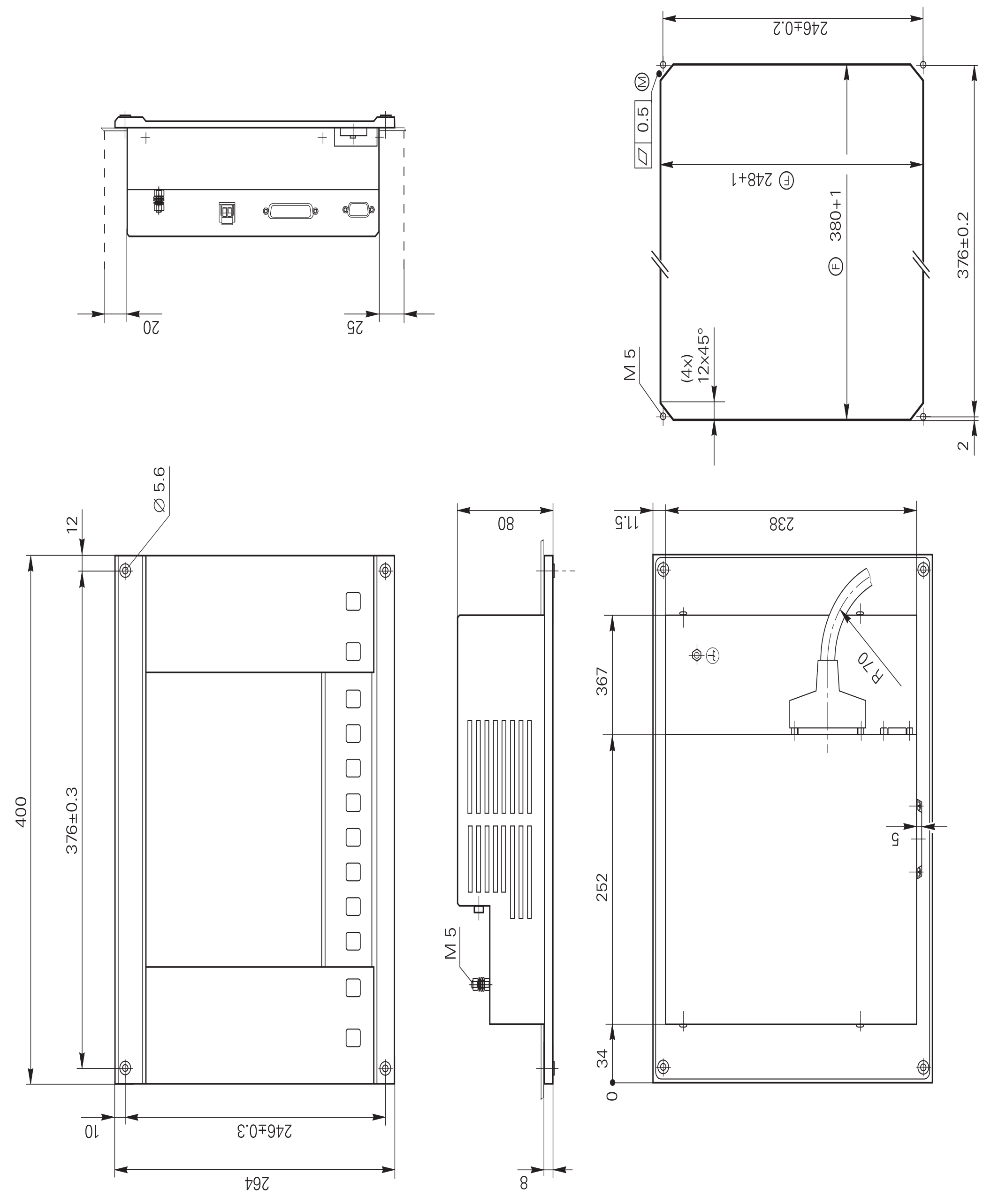


M: Mounting surface

F: Front panel opening

3.32.22 BF 120

Weight: 3 kg

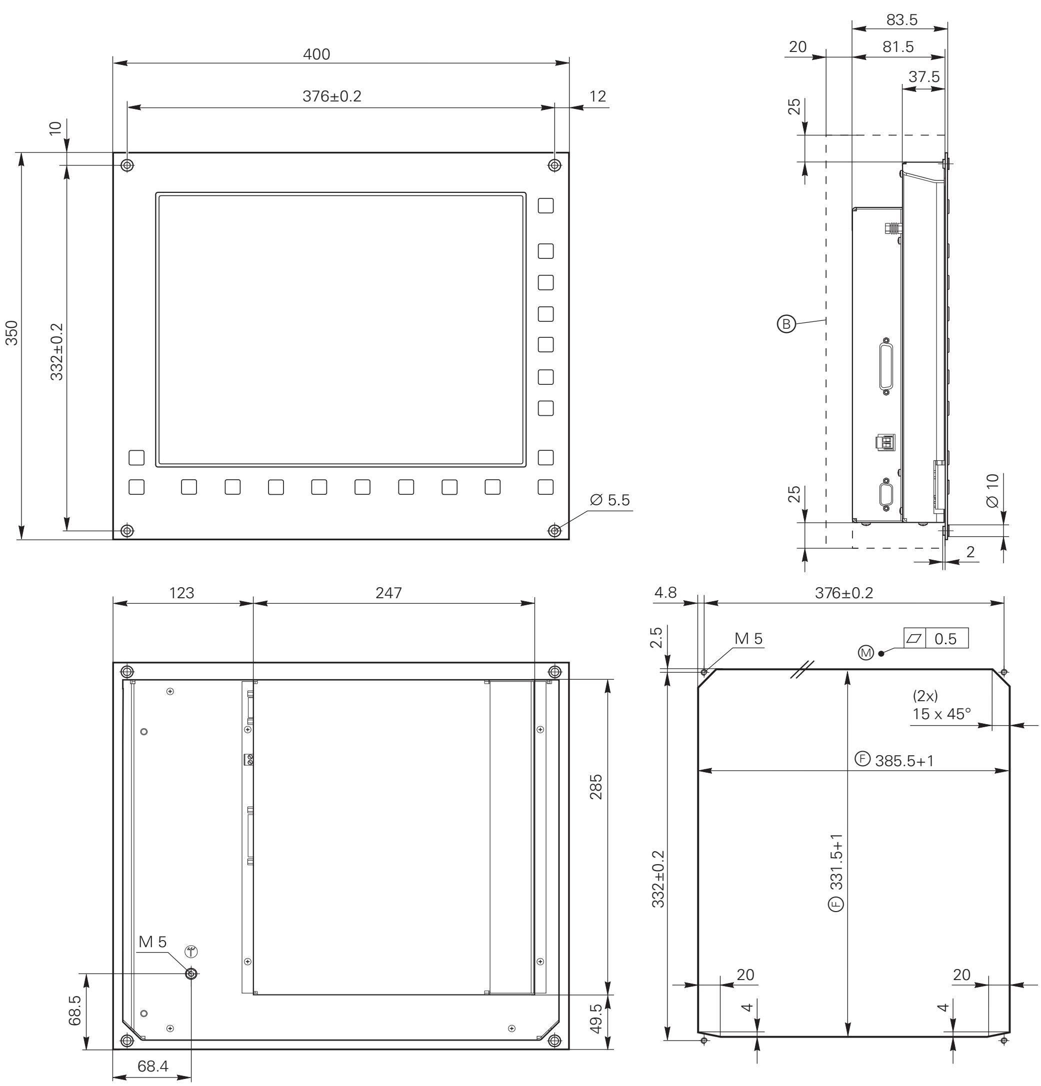


M: Mounting surface

F: Front panel opening

3.32.23 BF 150

Weight: 3 kg

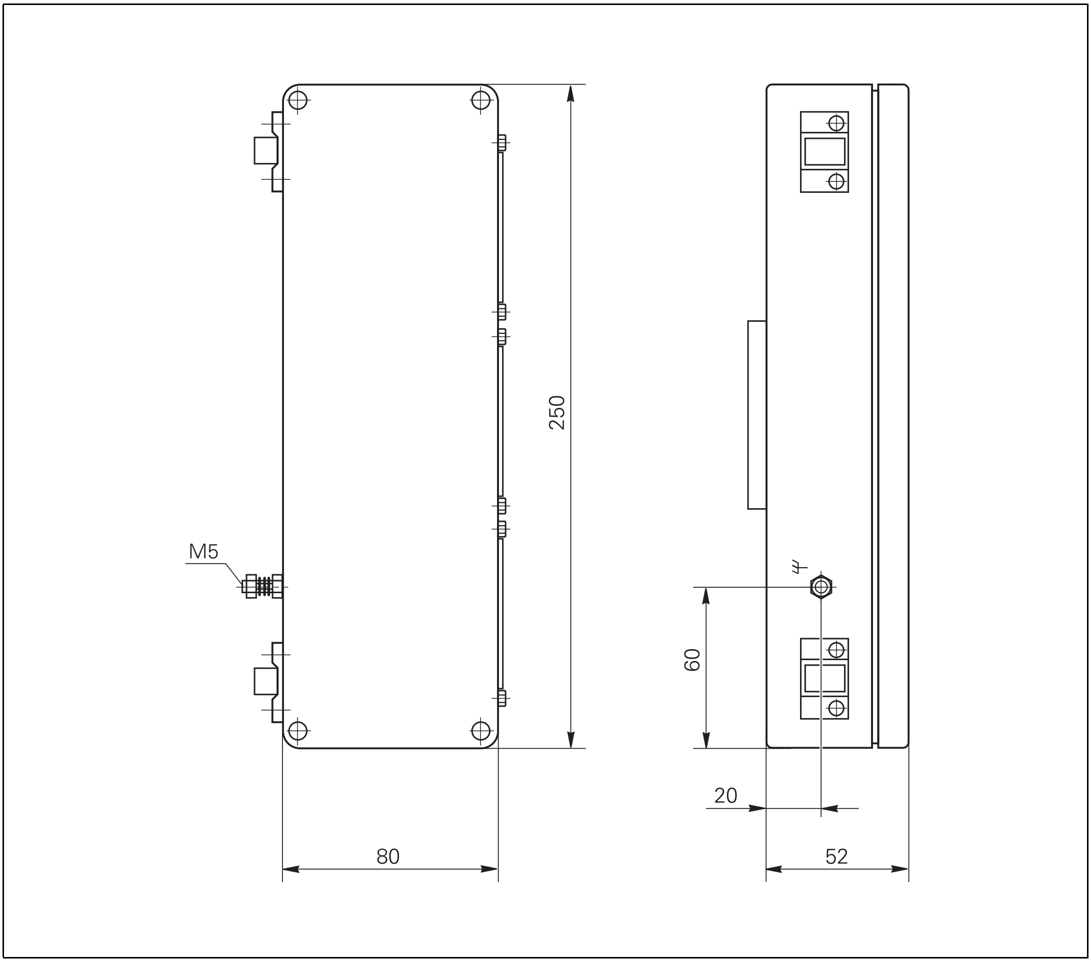


M: Mounting surface

F: Front panel opening

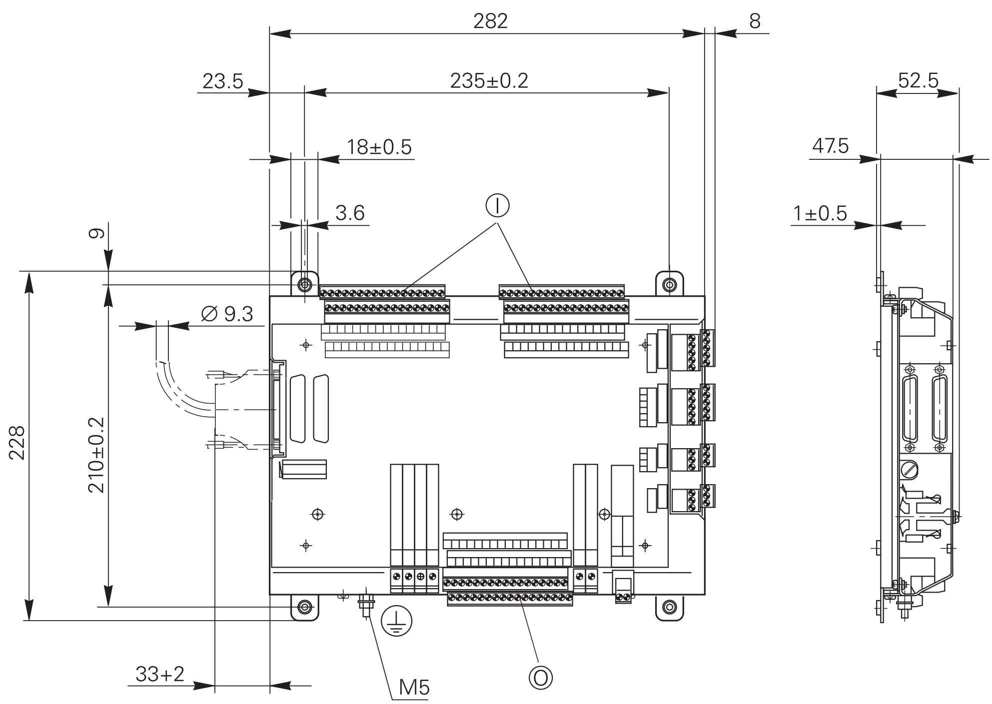
Dashed line: Space for air circulation

3.32.24 BTS 120/BTS 150



3.32.25 PL 4xx B

Weight: 1.5 kg

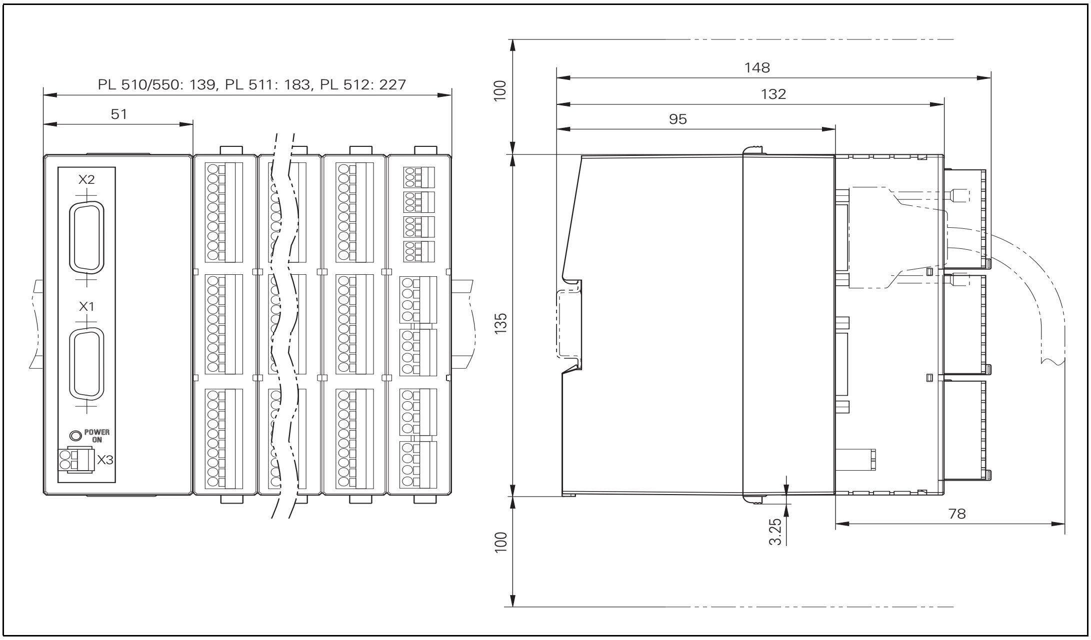


I: PLC inputs

O: PLC outputs

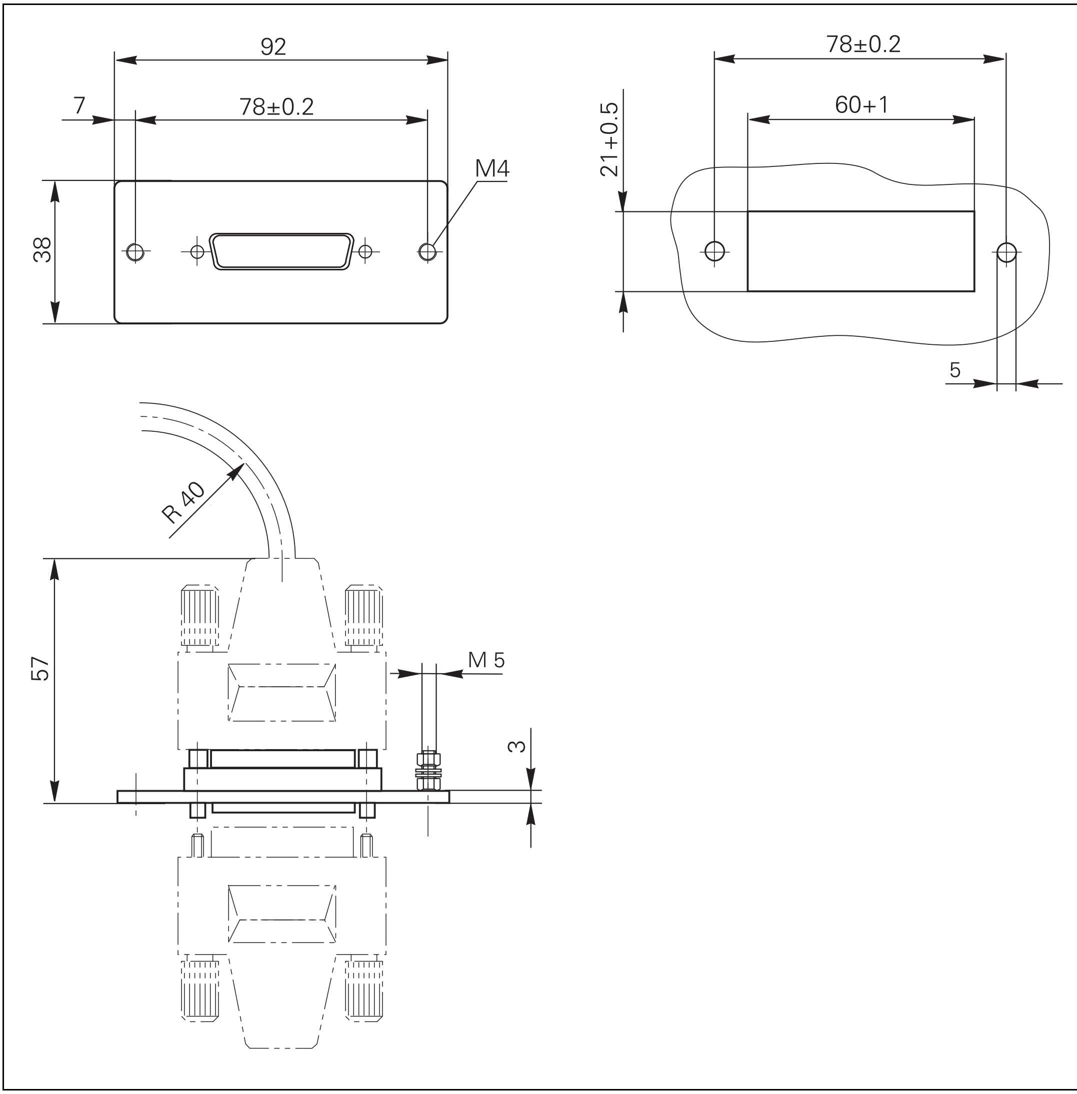
3.32.26 PL 51x

Weight: Approx. 1.0 kg

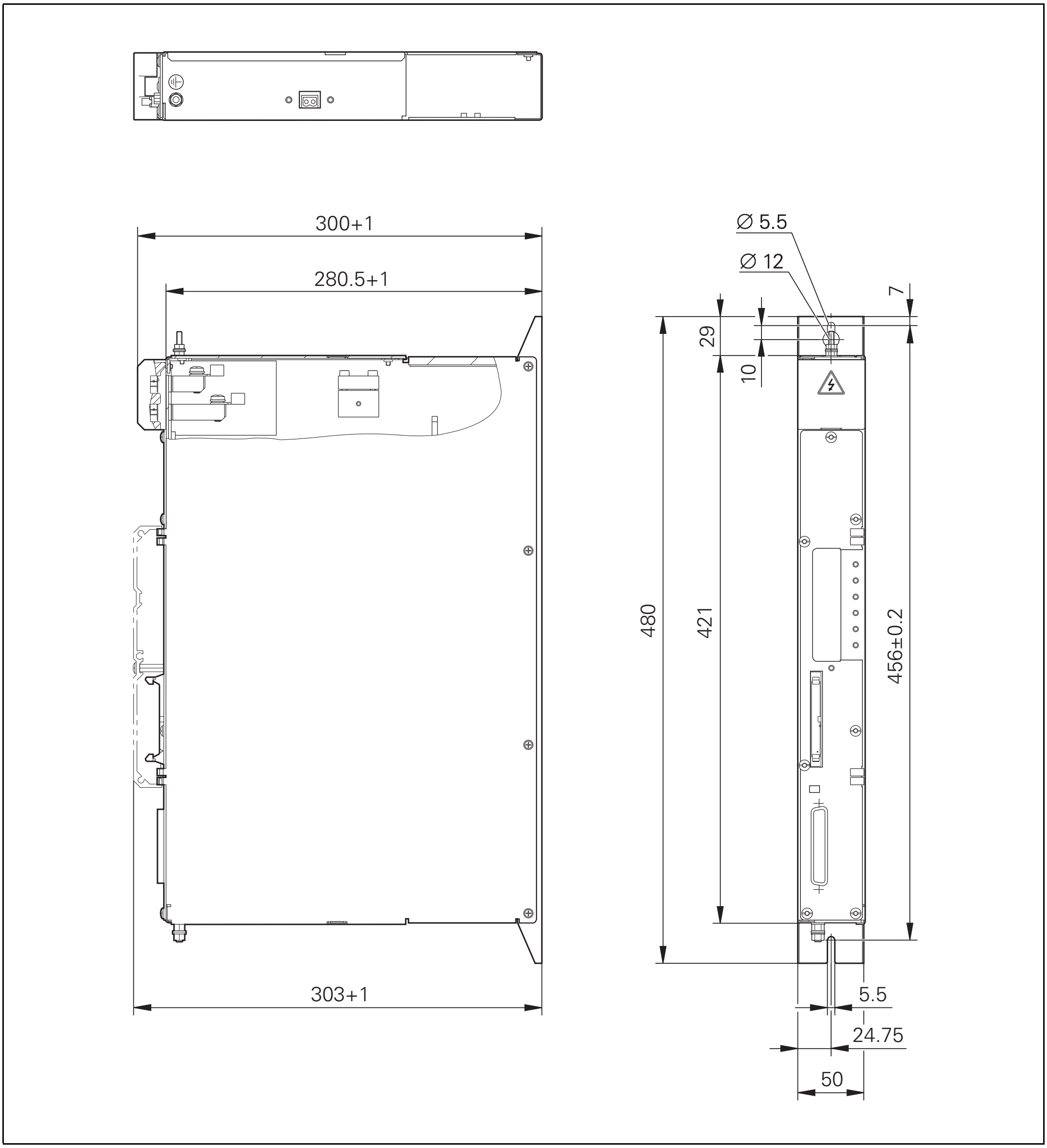


3.32.27 Adapter block for the data interface

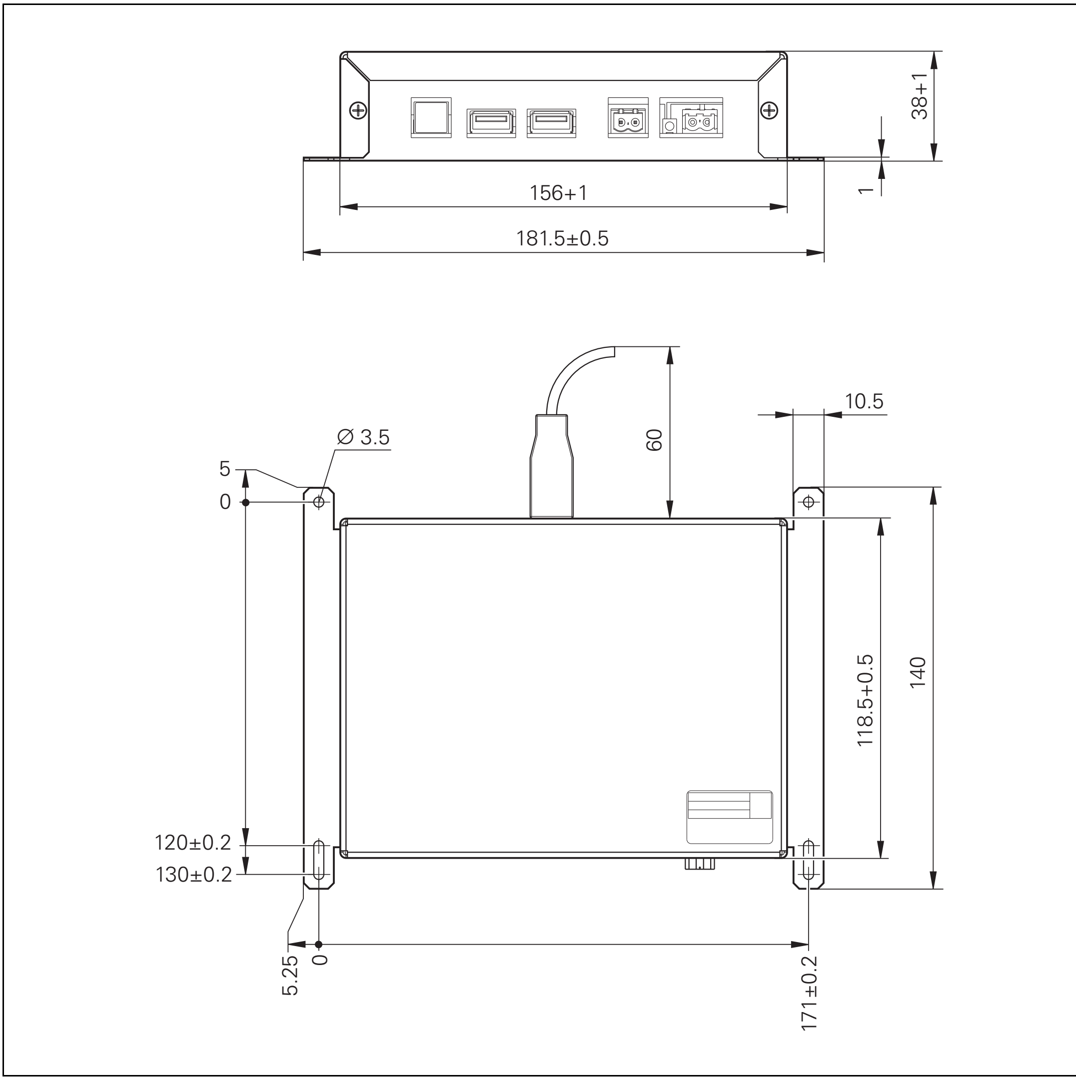
RS-232-C/V.24 adapter block and
RS-422/V.11 adapter block



3.32.28 MS 11x

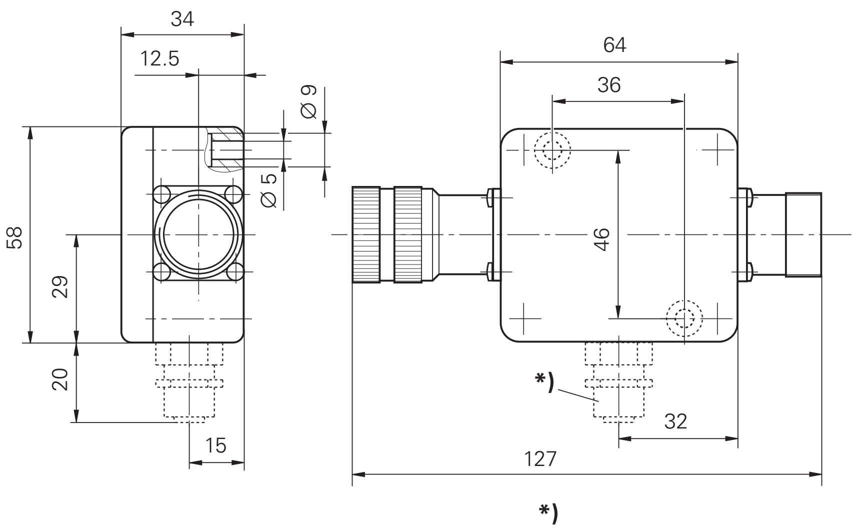


3.32.29 USB hub



3.32.30 Line-drop compensator

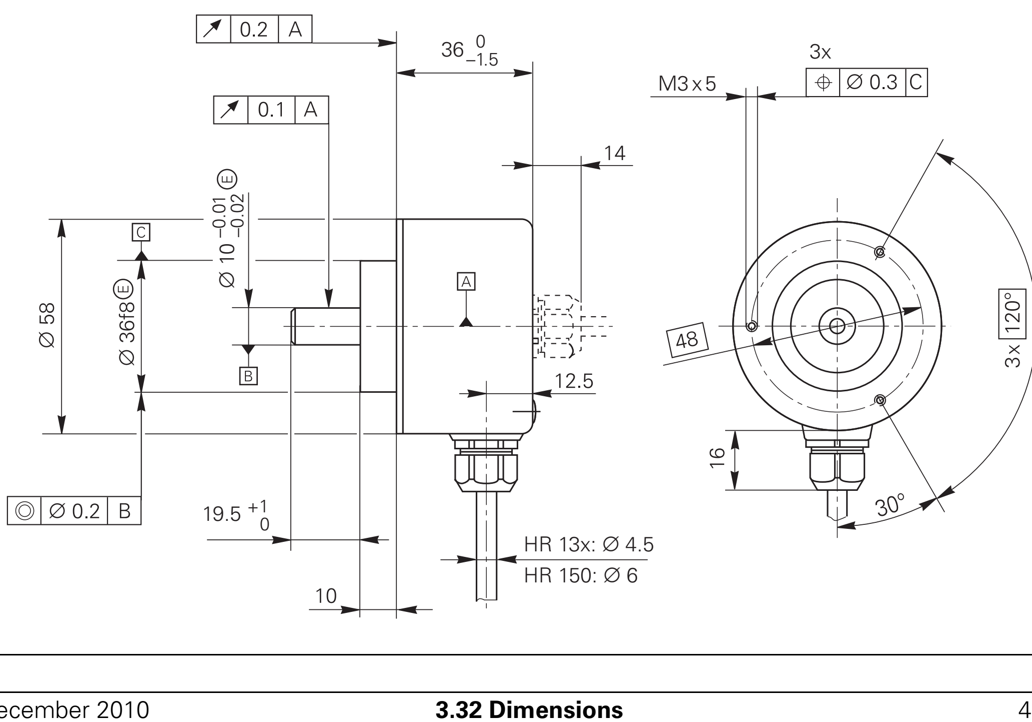
Line drop compensator for encoders with EnDat interface



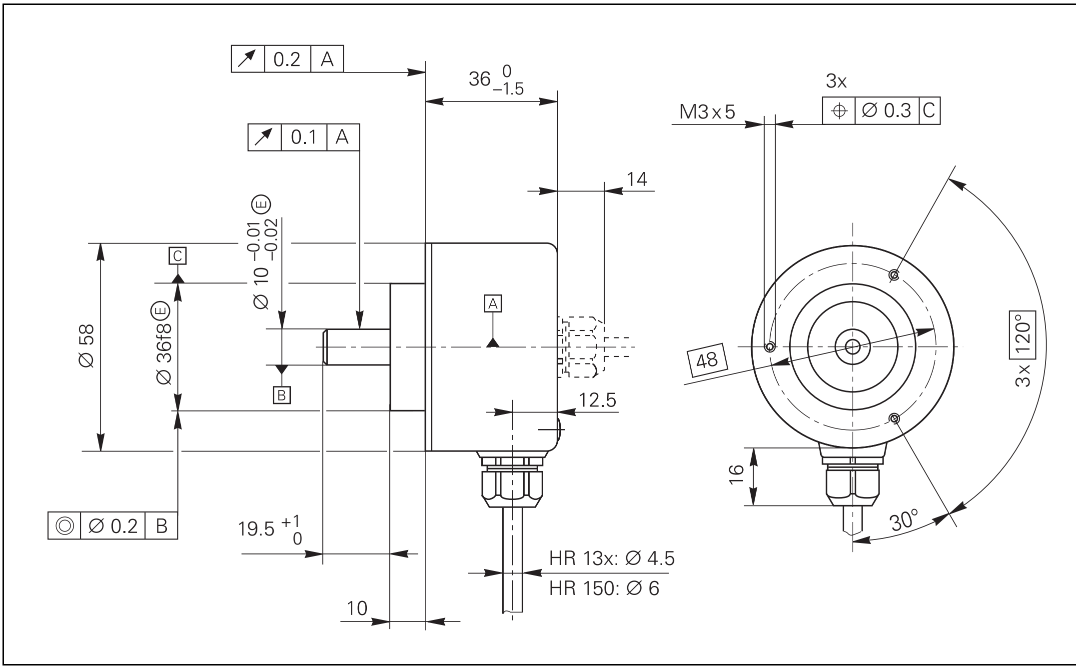
*) KTY

3.32.31 Handwheels

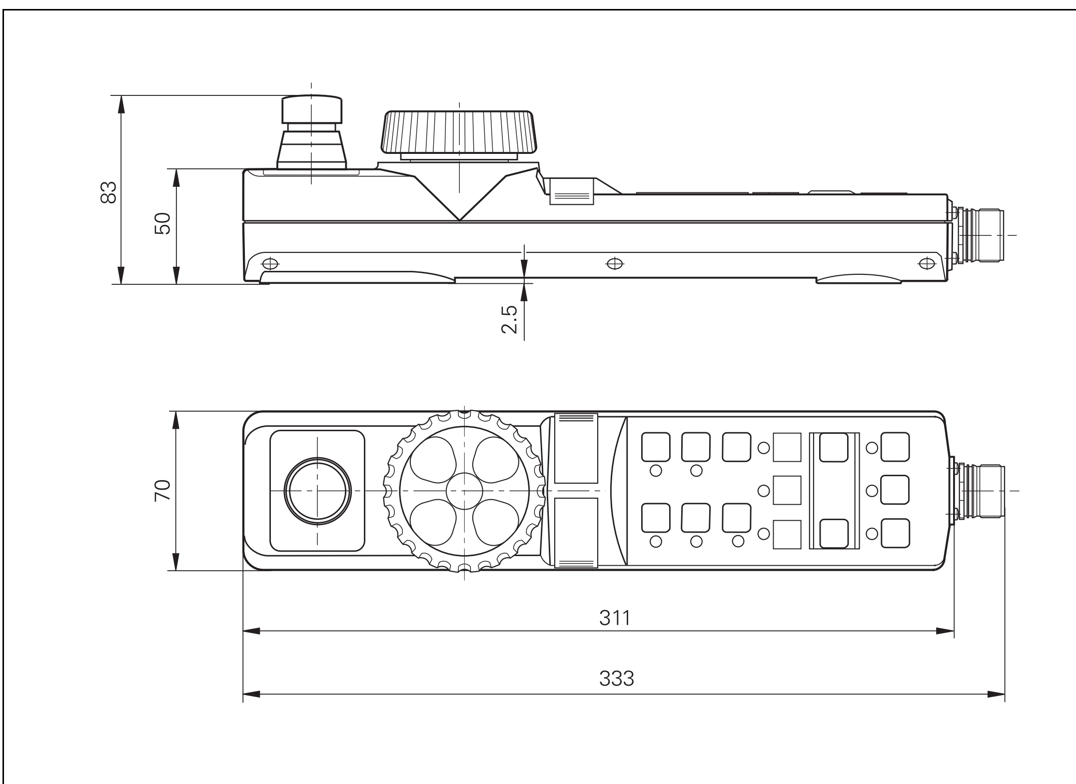
HR 130 Weight: Approx. 0.7 kg



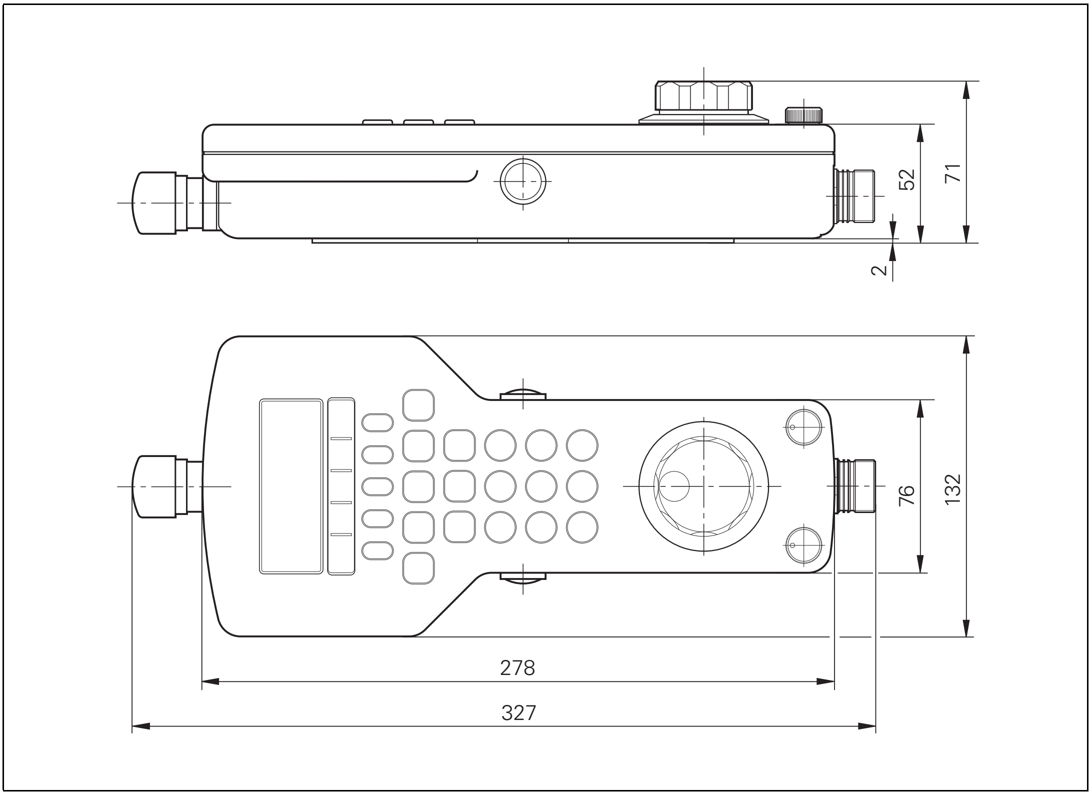
HR 150 Weight: Approx. 0.7 kg



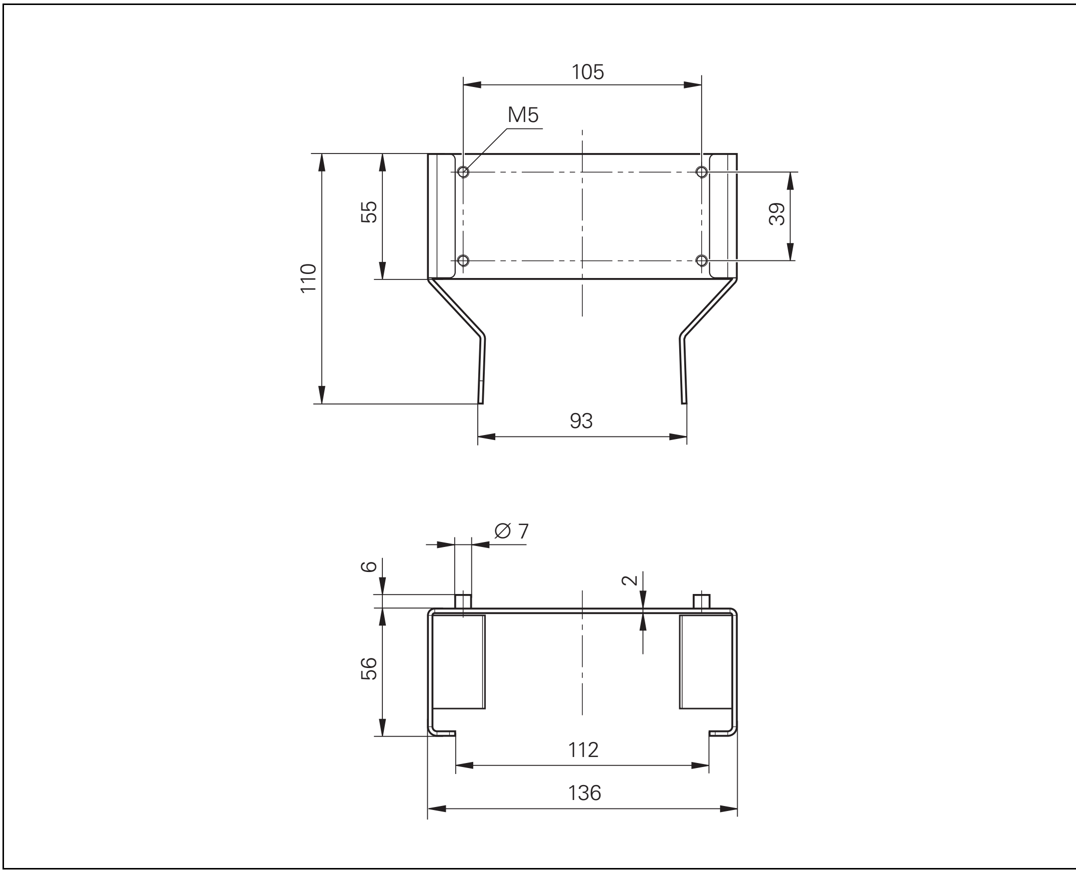
HR 410



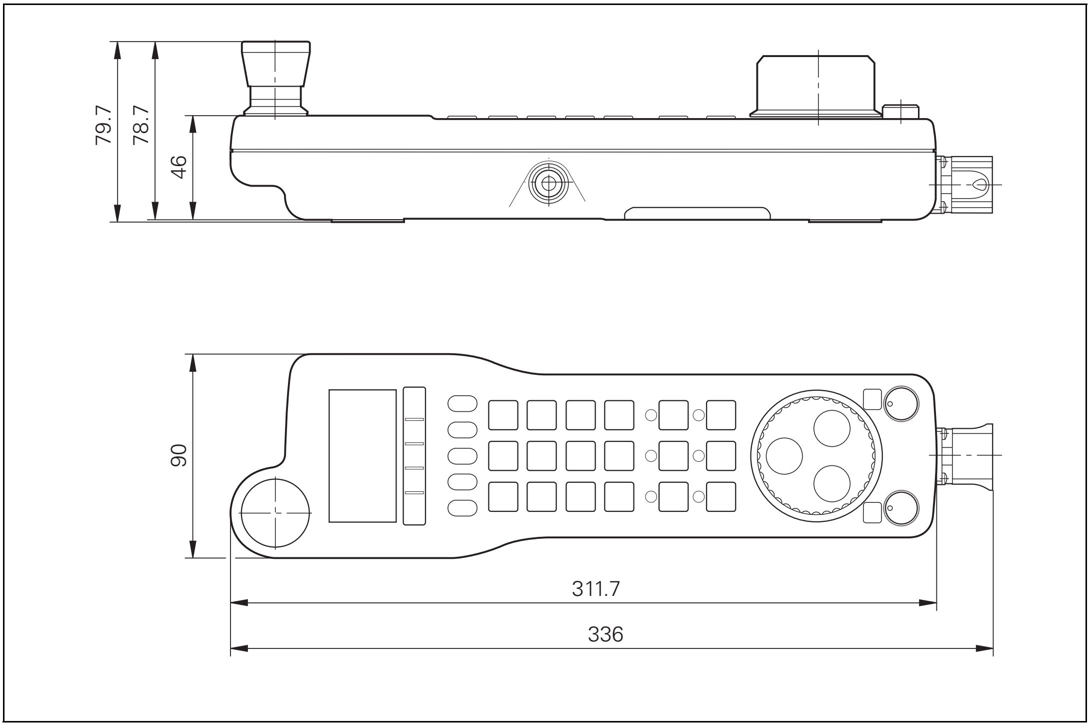
HR 420



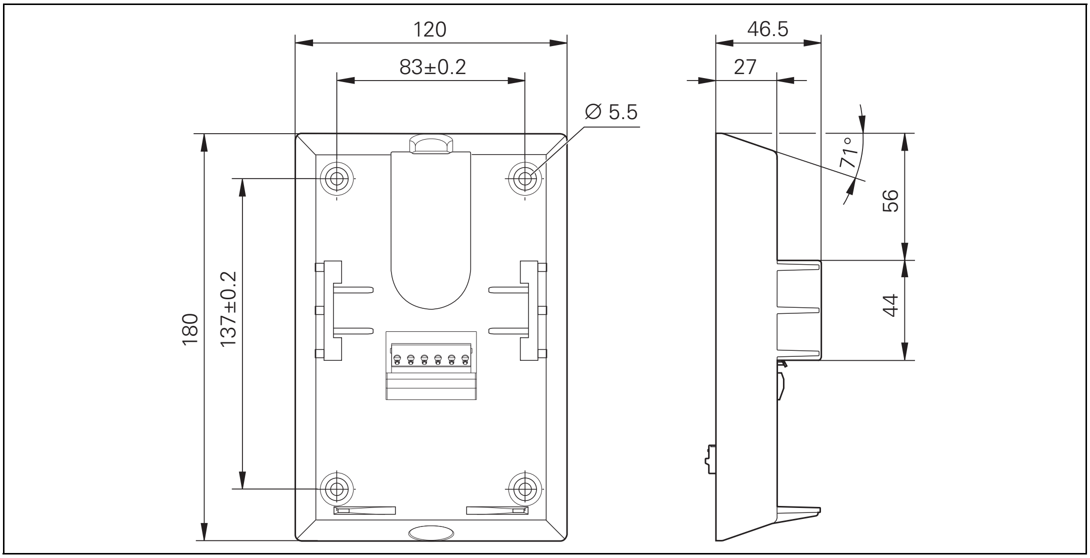
Mount for HR 420:



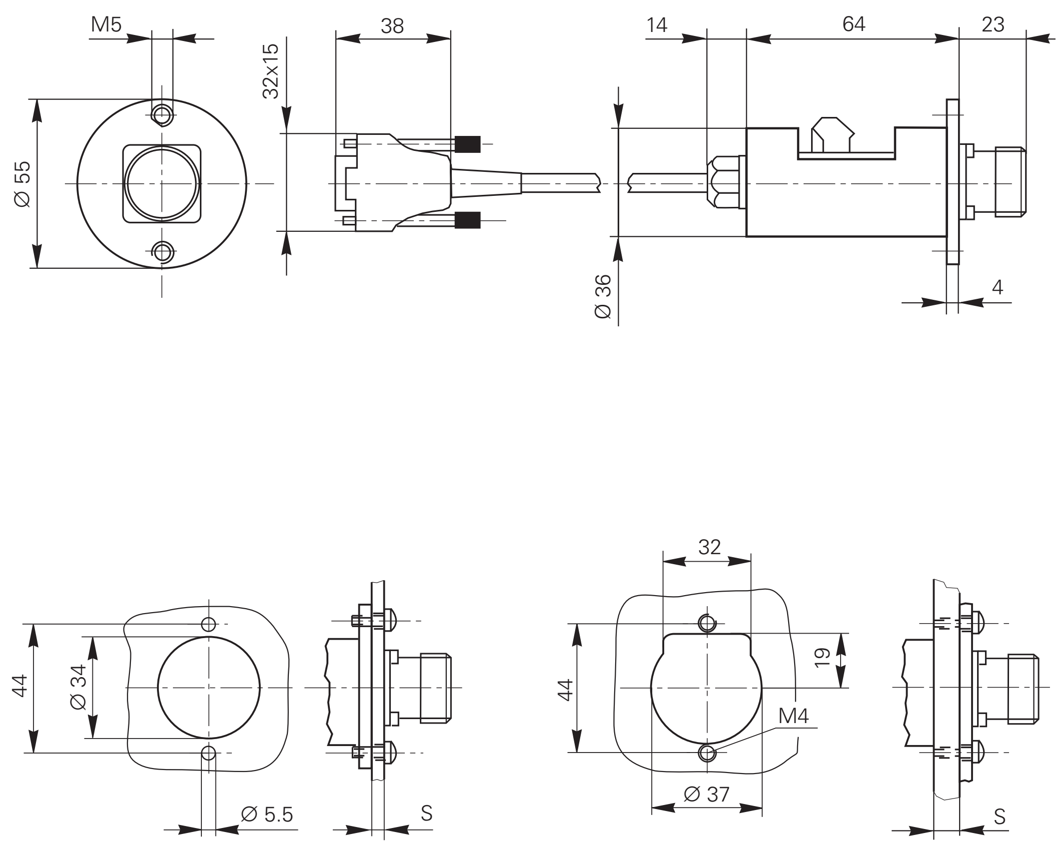
HR 520



Mount for HR 520



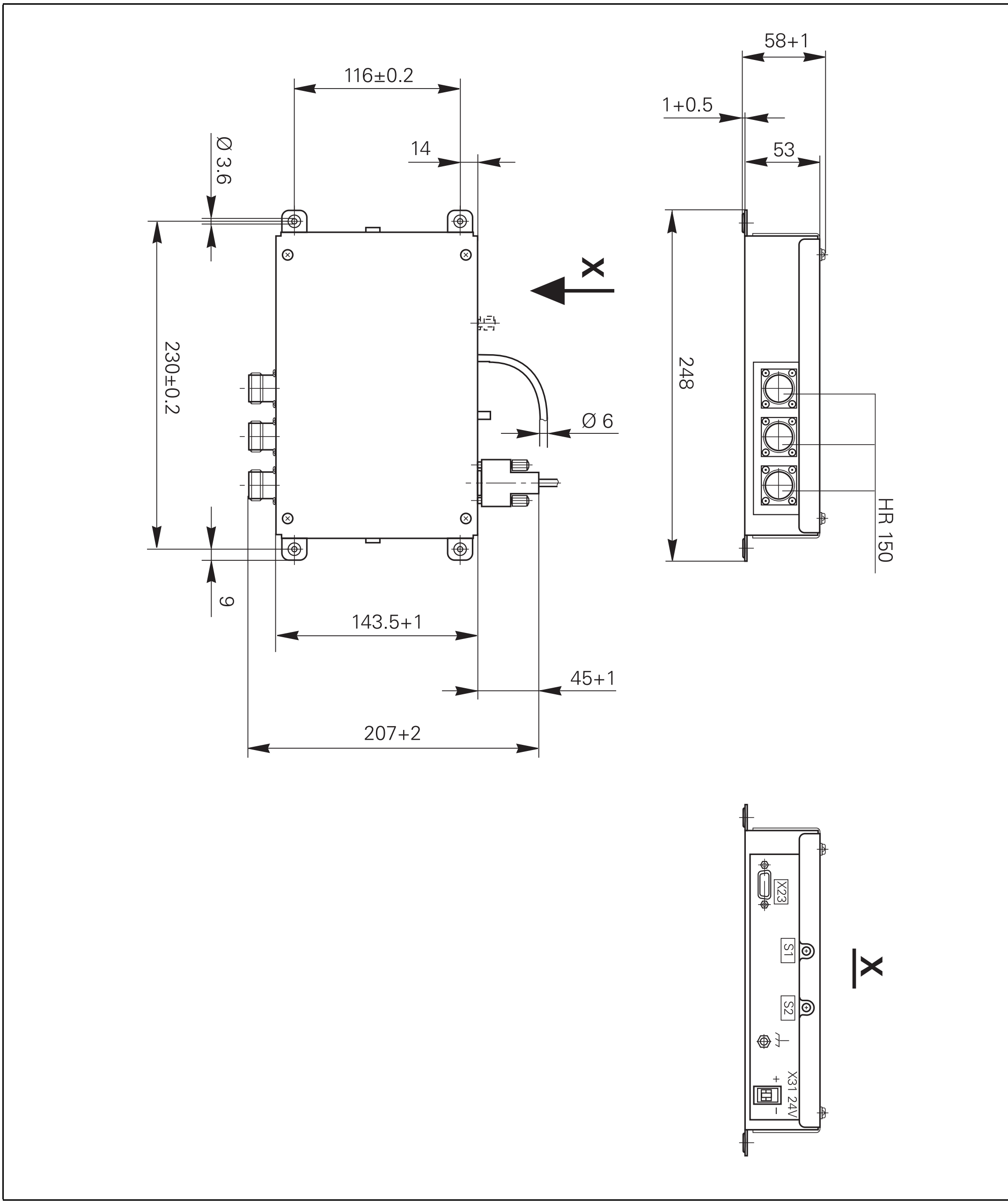
Adapter cables



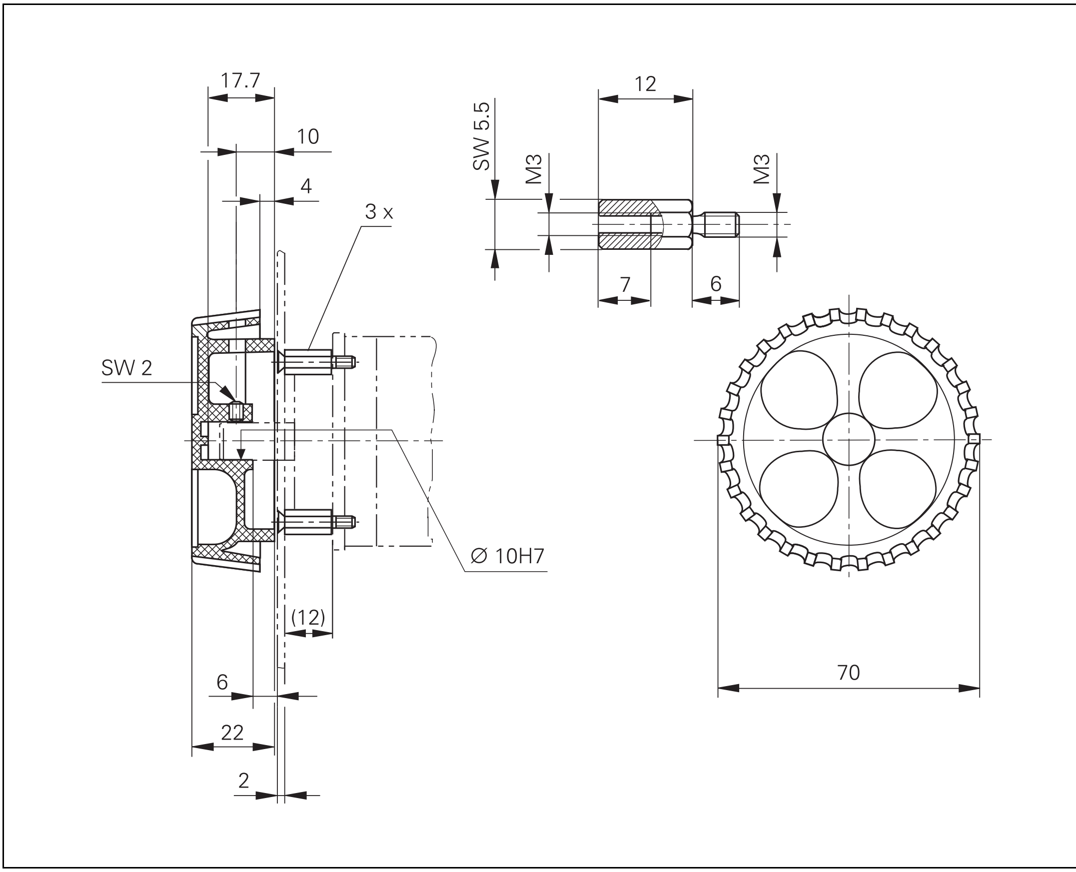
Mounting opening for wall thickness S < 4

Mounting opening for wall thickness S > 4

HRA 110 Weight: Approx. 1.5 kg

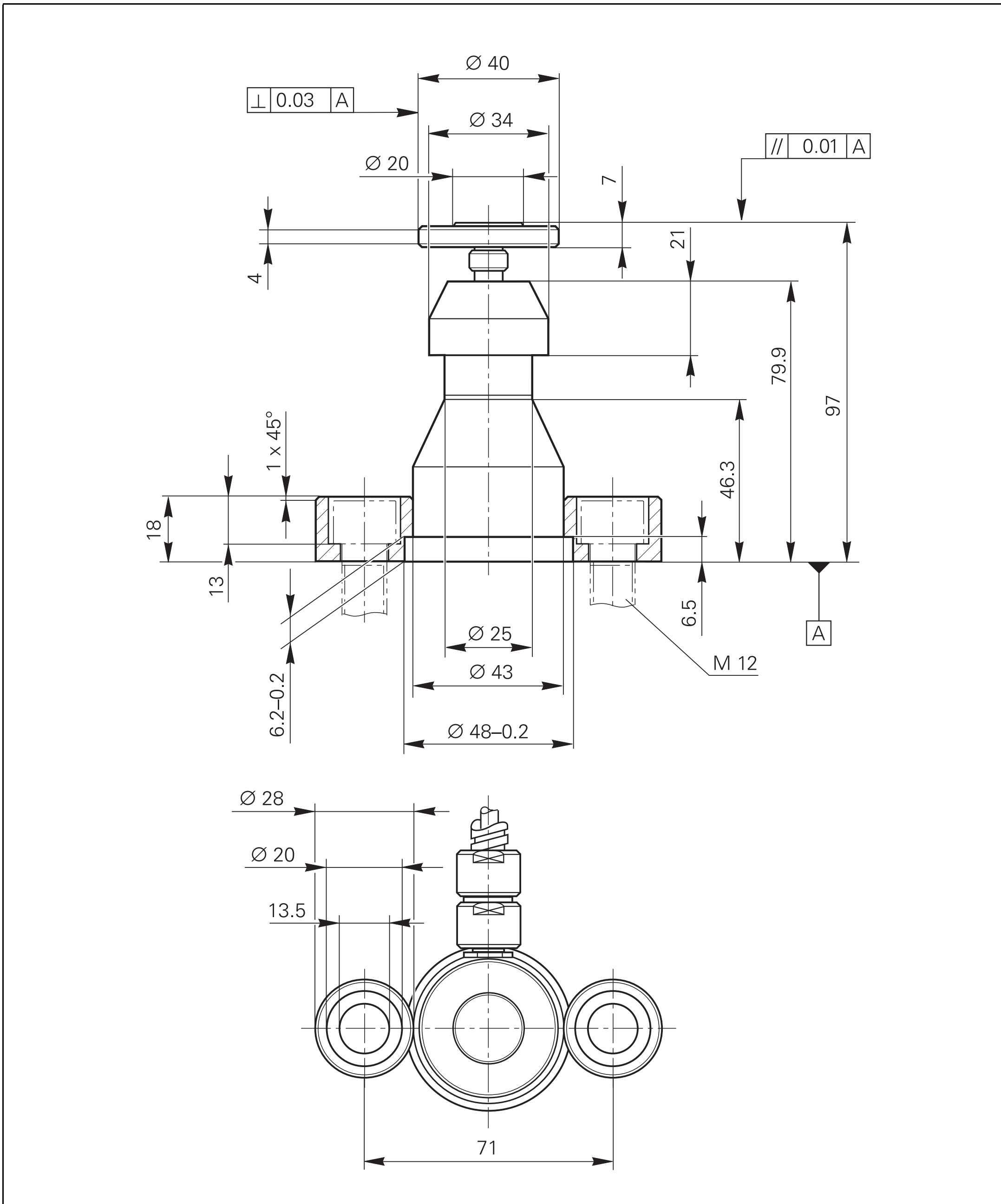


Control knob for HR 130 and HR 150



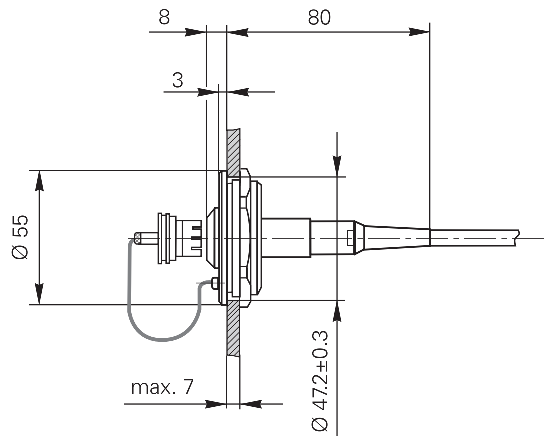
3.32.32 Touch probe systems

TT 130

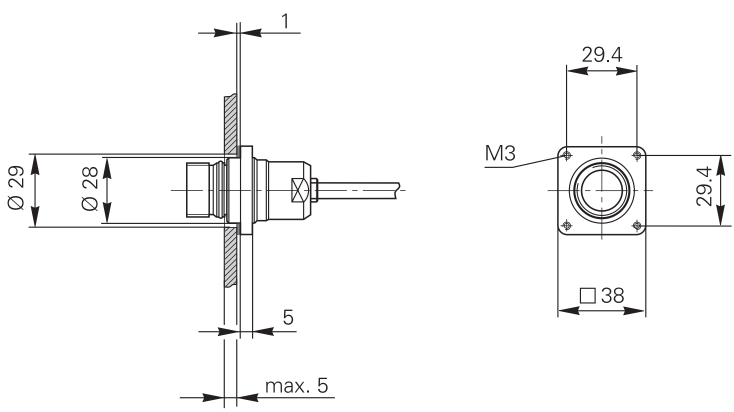


Adapter cable for TT and TS

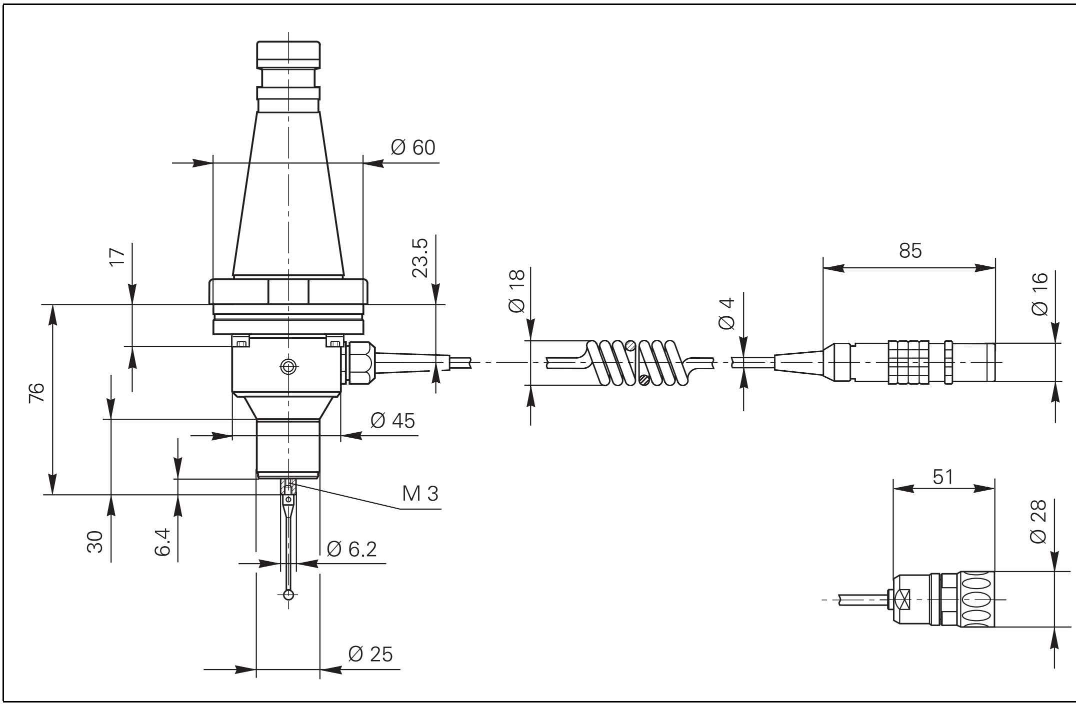
Mounting coupling for quick connection



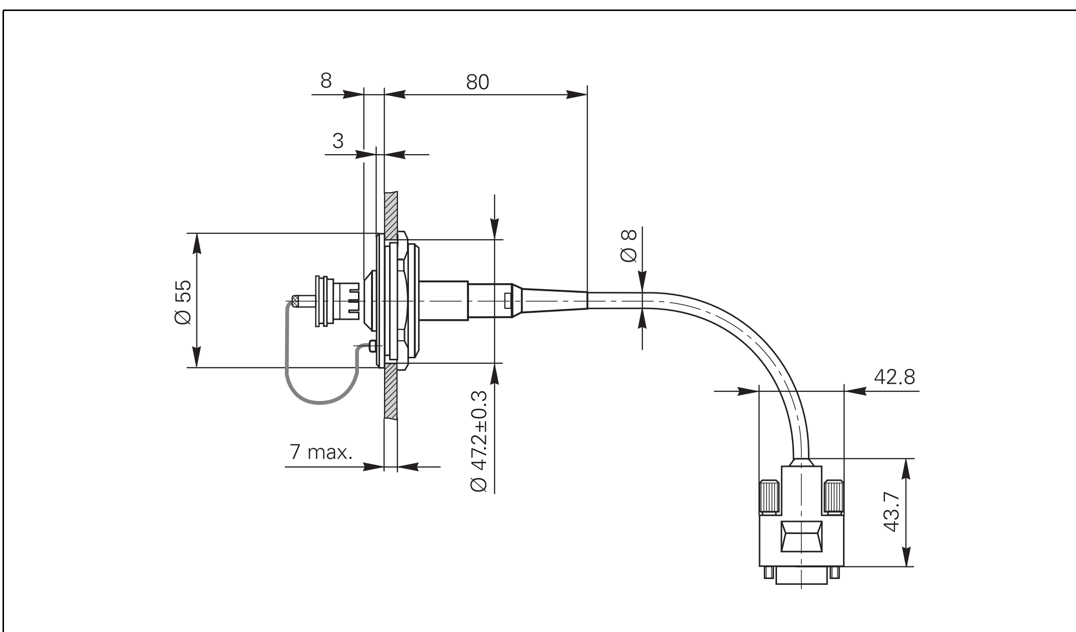
Mounting coupling for HEIDENHAIN standard connector



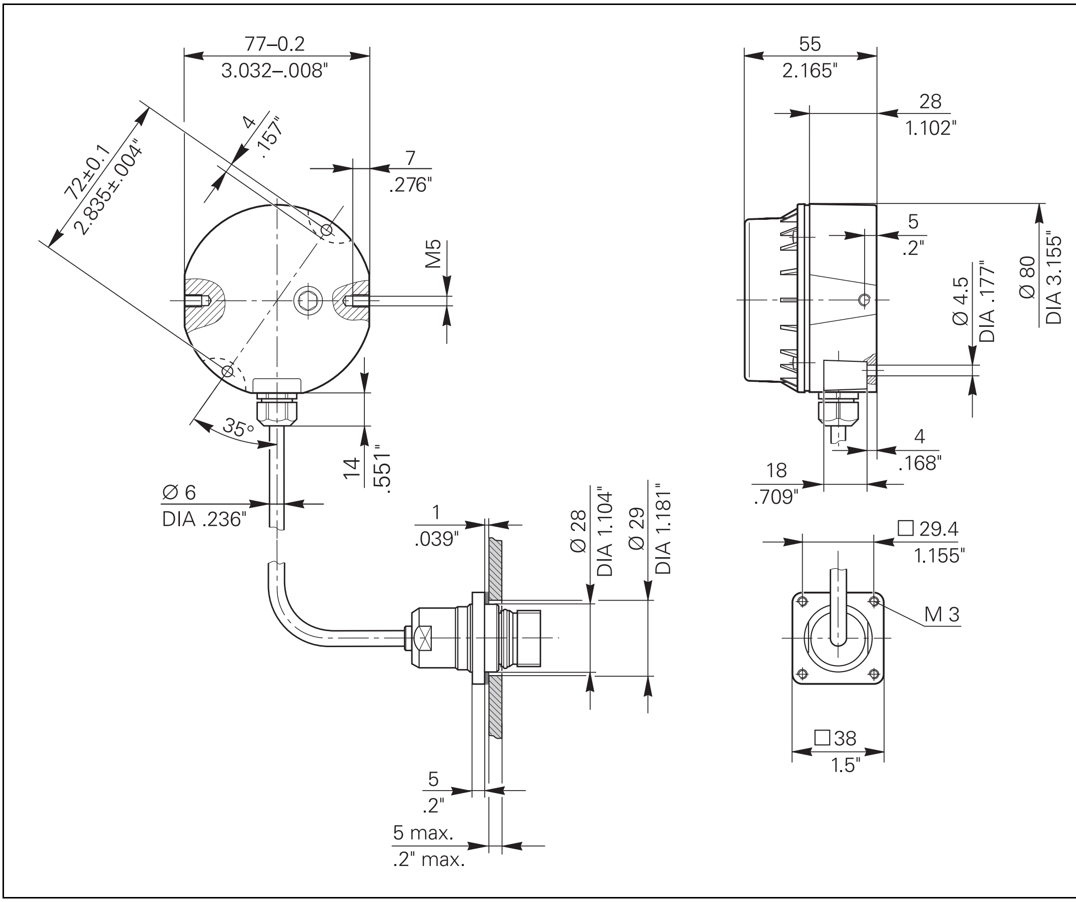
TS 220



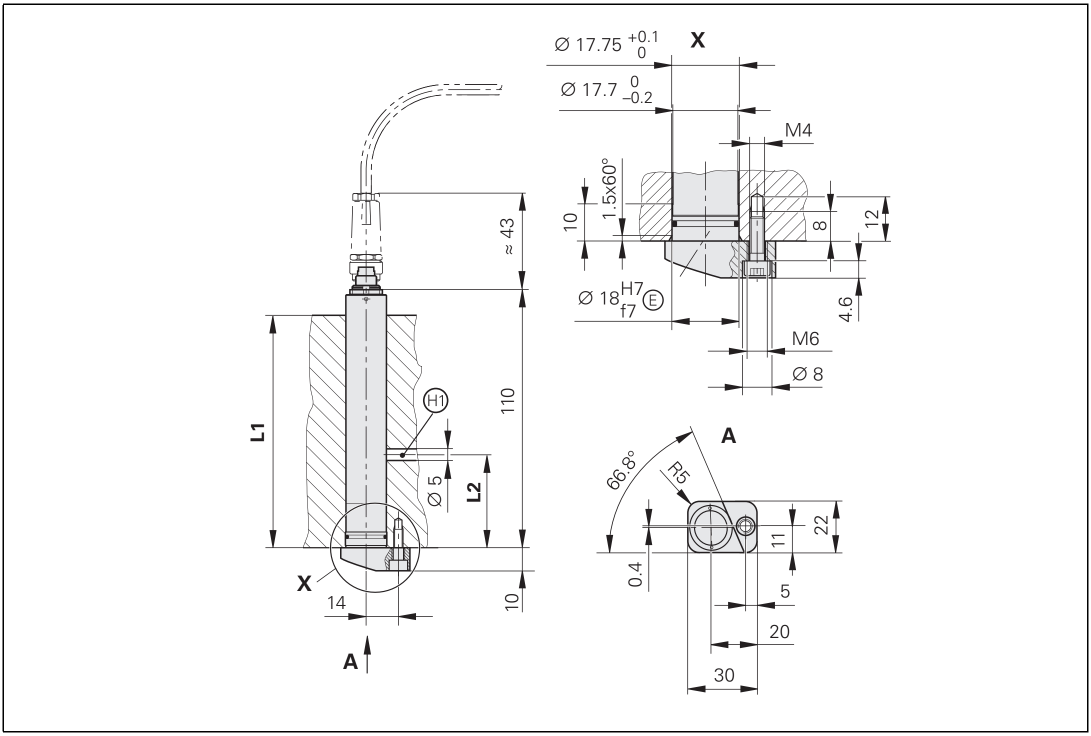
Adapter cable for TS 120/TS 220



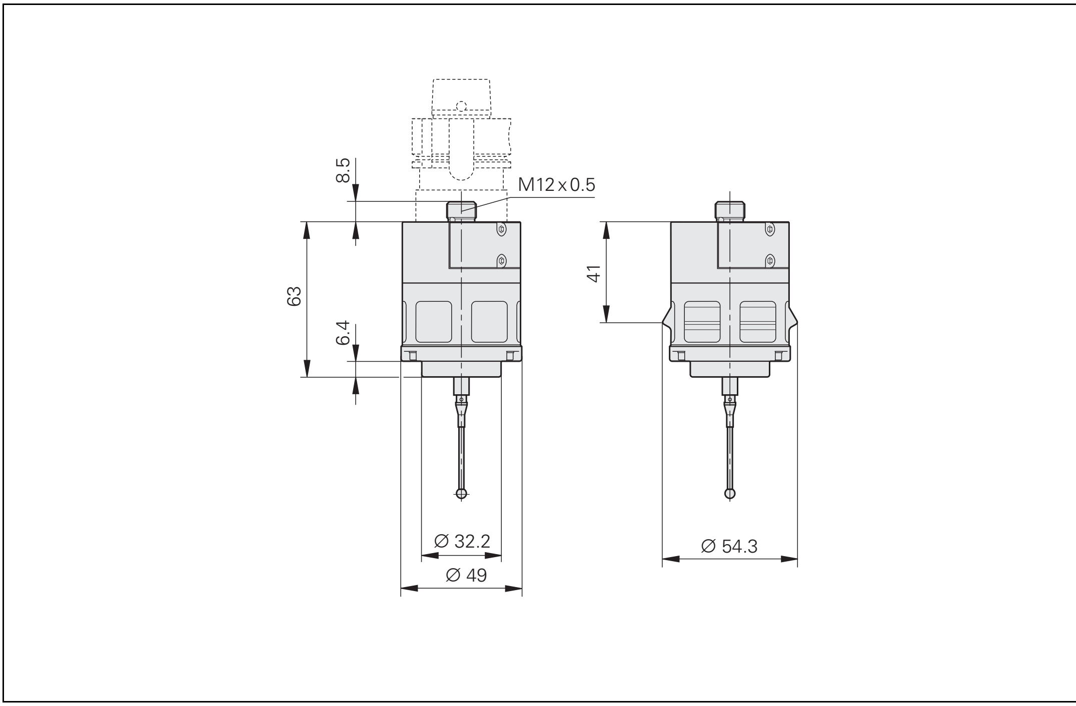
SE 640 transmitter-receiver unit



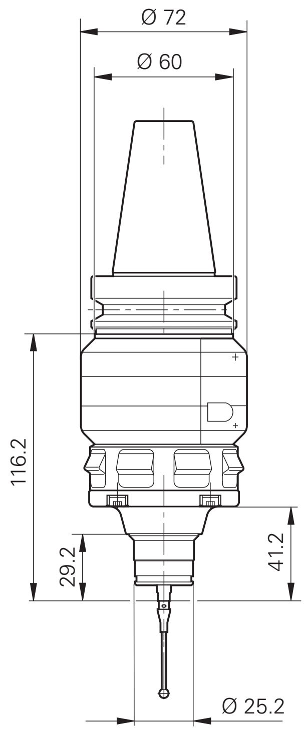
SE 540 transmitter-receiver unit



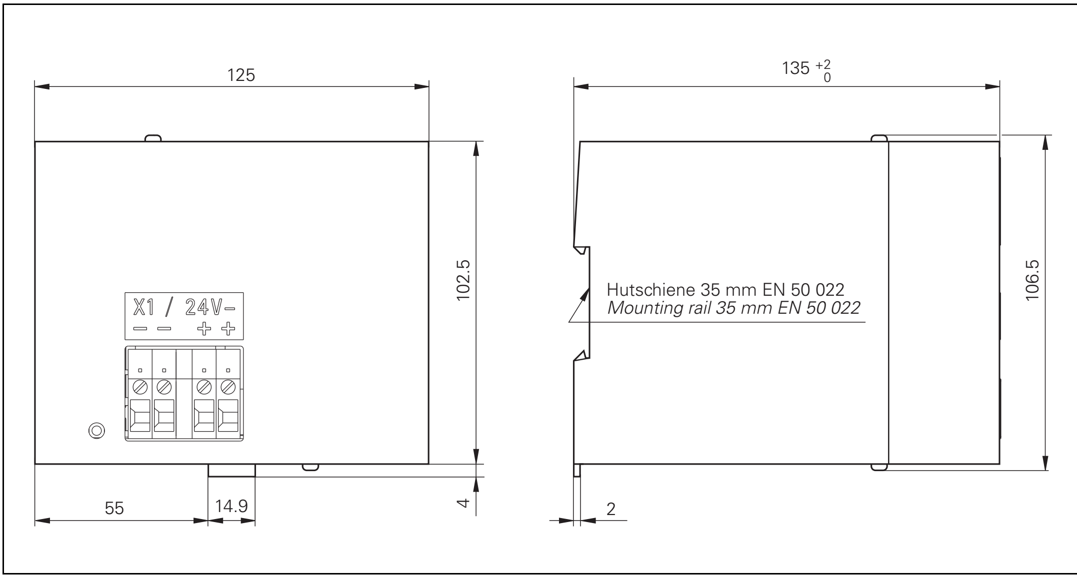
TS 440



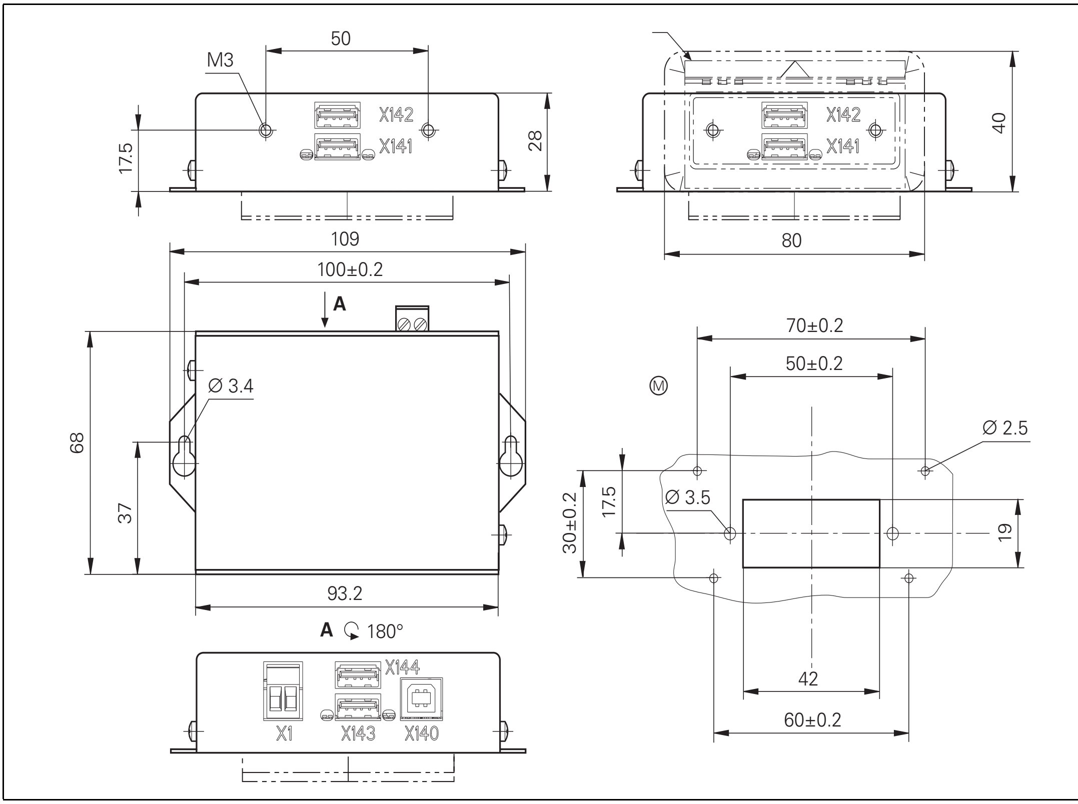
TS 640

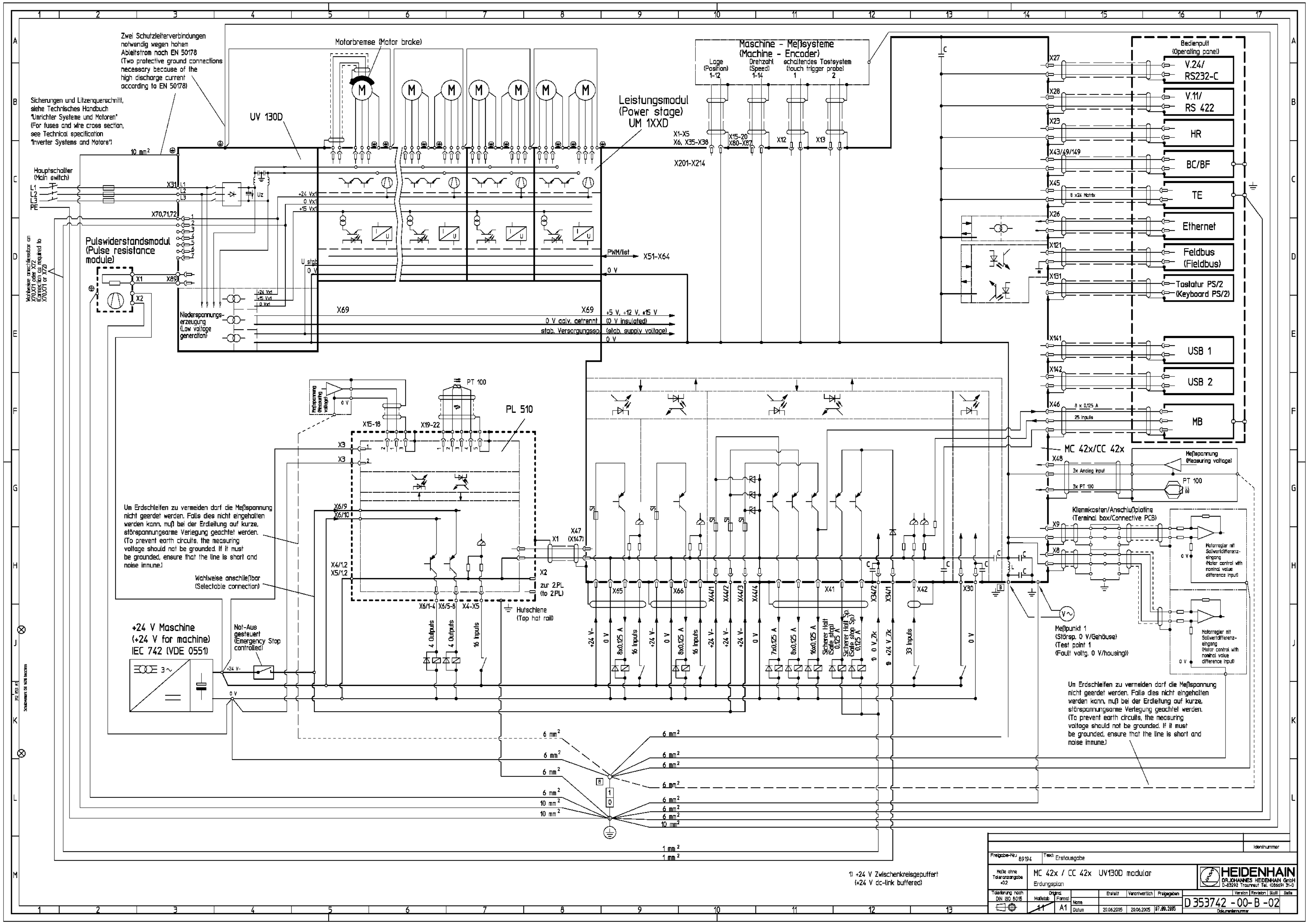


3.32.33 CML 110

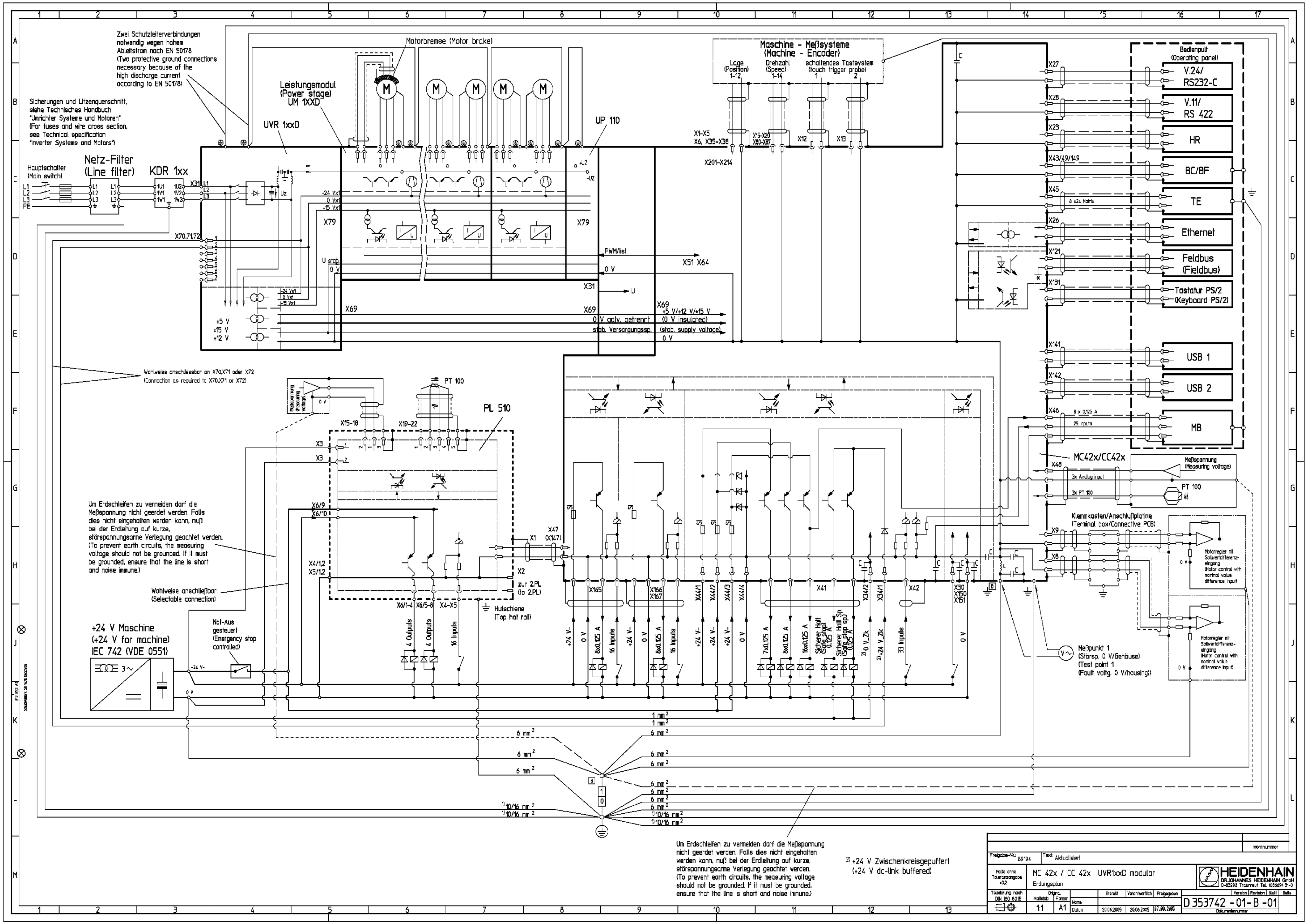


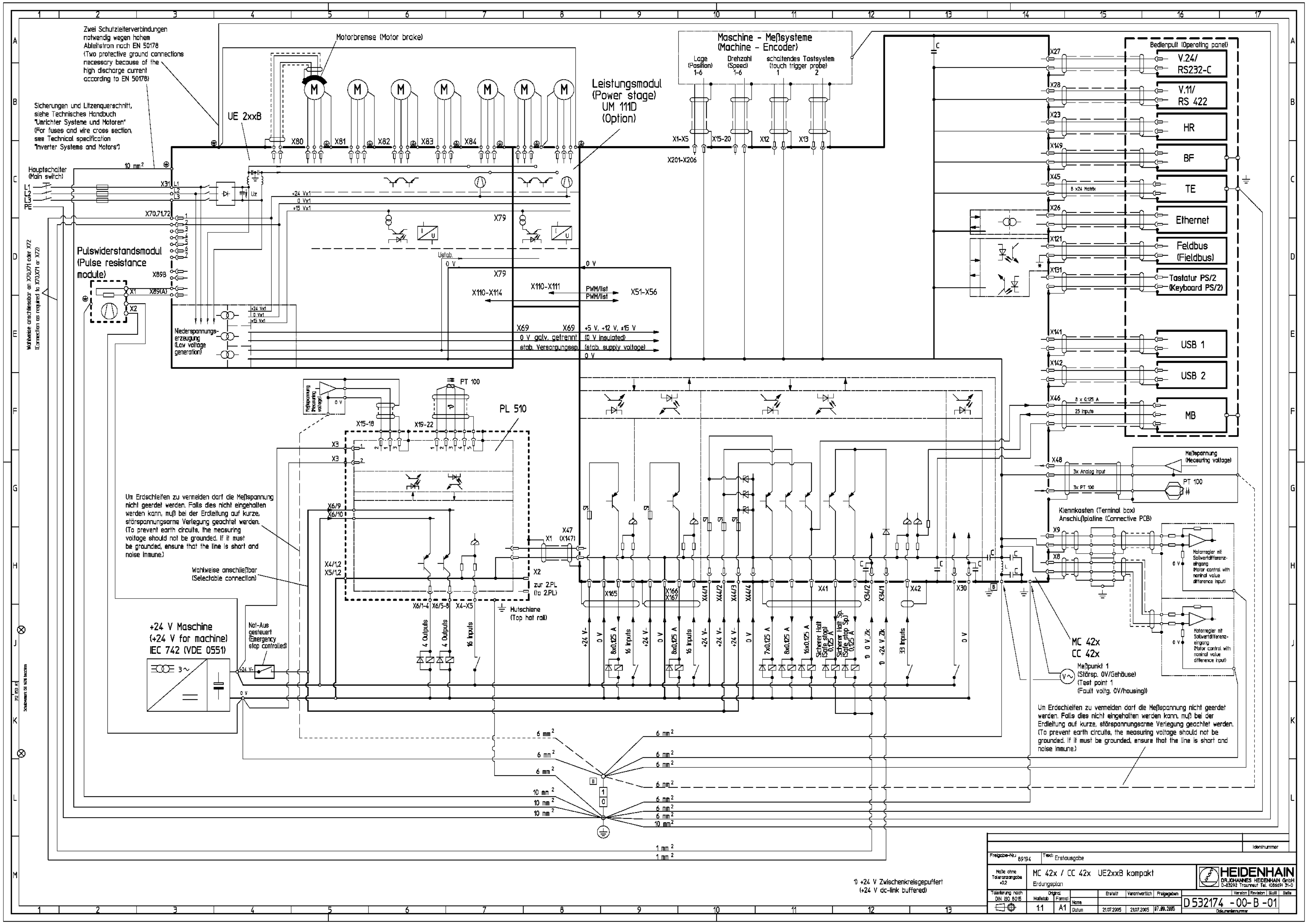
3.32.34 USB hub for operating panel



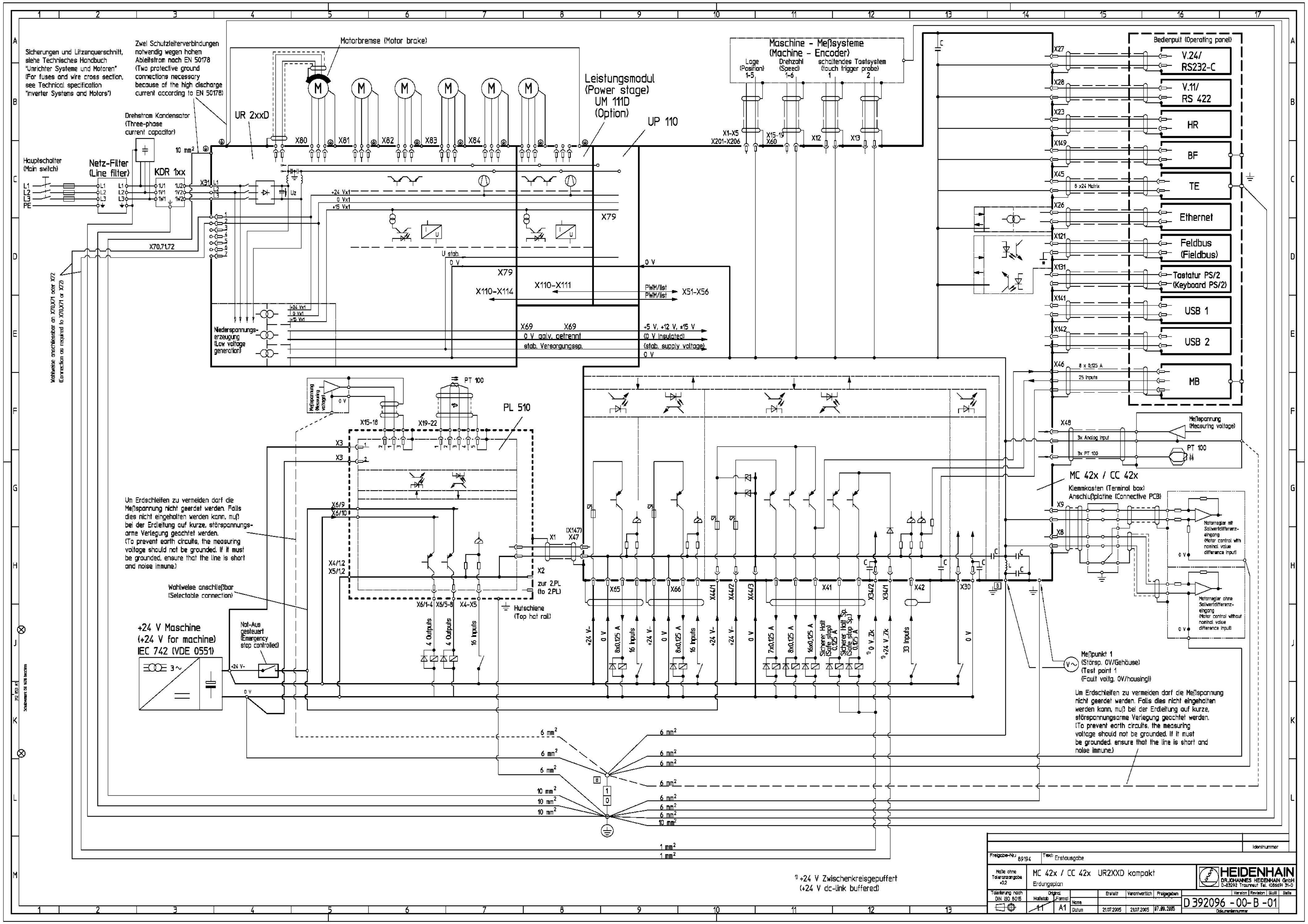


3.34 Grounding Diagram: Modular Regenerative Inverter System

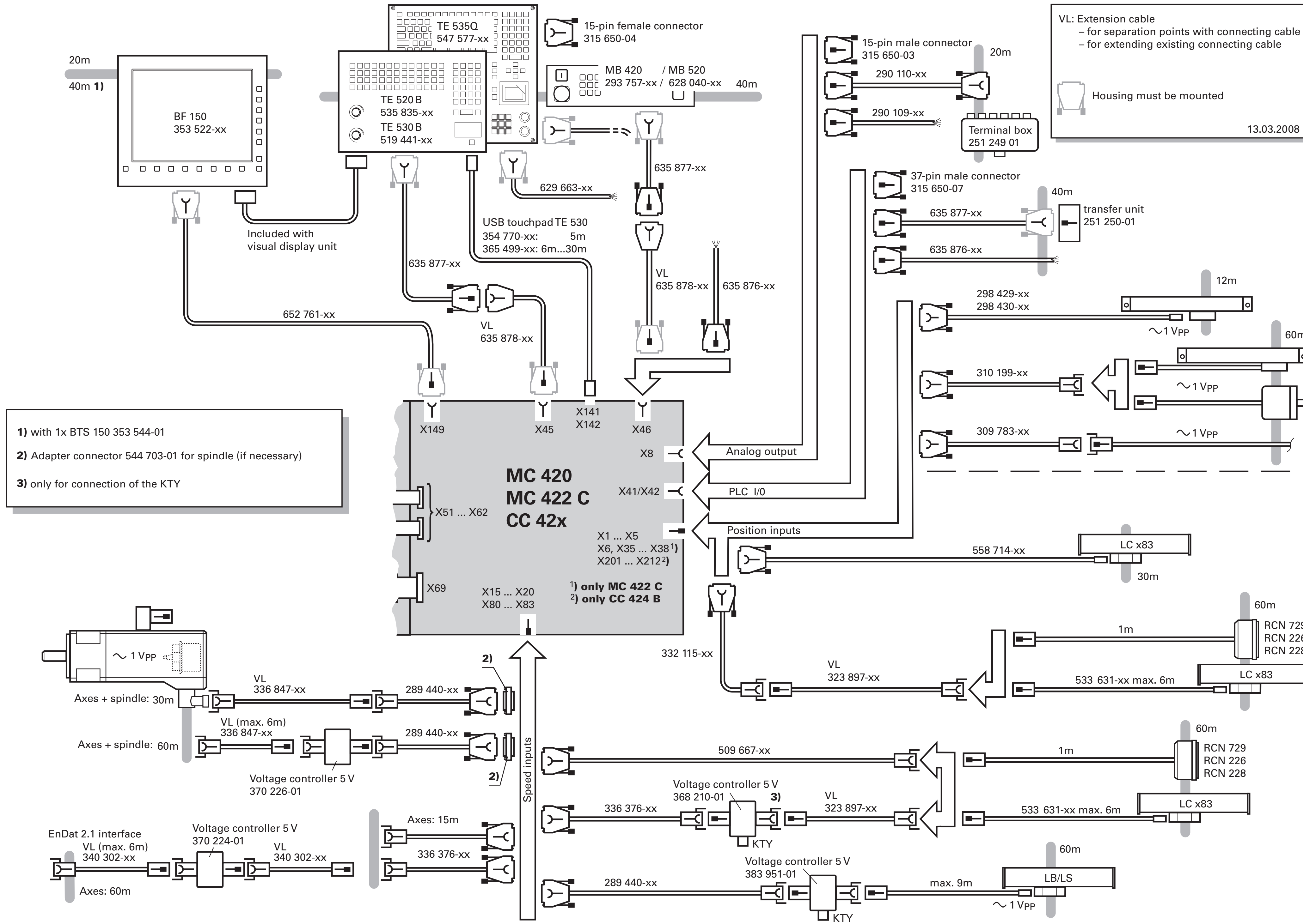



Motorbremse (Motor brake)
Leistungsmodul (Power stage) UM 111D (Option)
UE 2xxB
Maschine - Meßsysteme (Machine - Encoder)
Bedienpult (Operating panel)
V.24/ RS232-C
V.11/ RS 422
HR
BF
TE
Ethernet
Feldbus (Fieldbus)
Tastatur PS/2 (Keyboard PS/2)
USB 1
USB 2
MB
Hauptschalter (Main switch)
Pulswiderstandsmodul (Pulse resistance module)
Niederspannungserzeugung (Low voltage generation)
PL 510
+24 V Maschine (+24 V for machine) IEC 742 (VDE 0551)
Not-Aus gesteuert (Emergency stop controlled)
Hutschiene (Top hat rail)
MC 42x CC 42x
Klemmkasten (Terminal box) Anschlußplatine (Connective PCB)
Meßpunkt 1 (Störsp. 0V/Gehäuse) (Test point 1 (Fault voltg. 0V/housing))
Um Erdschleifen zu vermeiden darf die Meßspannung nicht geerdet werden. Falls dies nicht eingehalten werden kann, muß bei der Erdleitung auf kurze, störspannungsarme Verlegung geachtet werden. (To prevent earth circuits, the measuring voltage should not be grounded. If it must be grounded, ensure that the line is short and noise immune.)
1) +24 V Zwischenkreisgepuffert (+24 V dc-link buffered)
MC 42x / CC 42x UE2xxB kompakt
Erdungsplan
HEIDENHAIN
D532174 -00-B -01

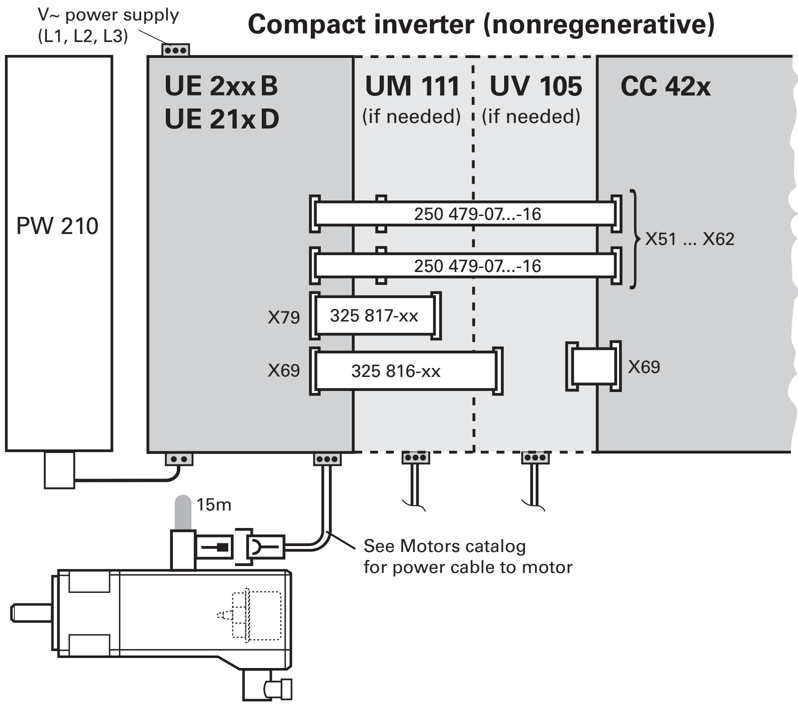
3.36 Grounding Diagram for iTNC 530 with UR 2xx Compact Inverter

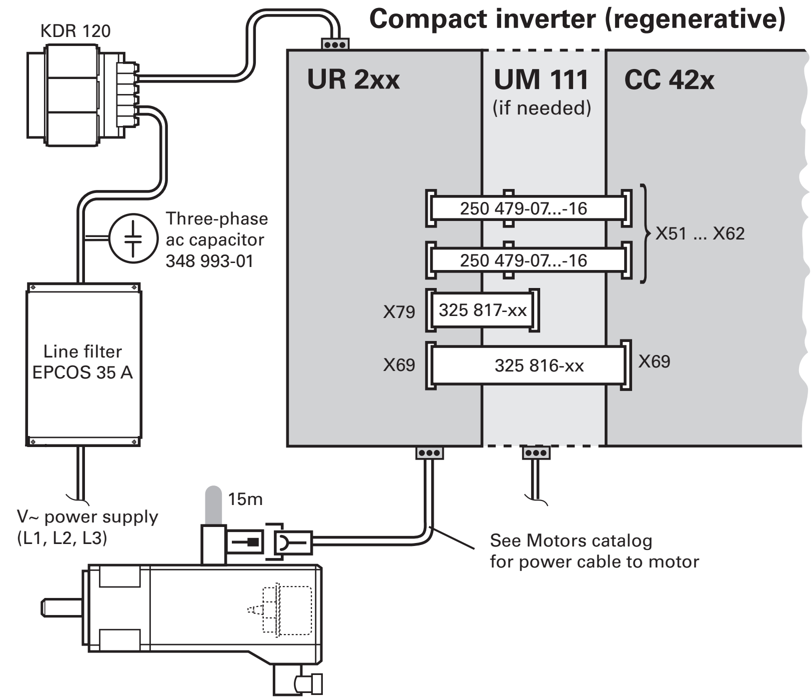


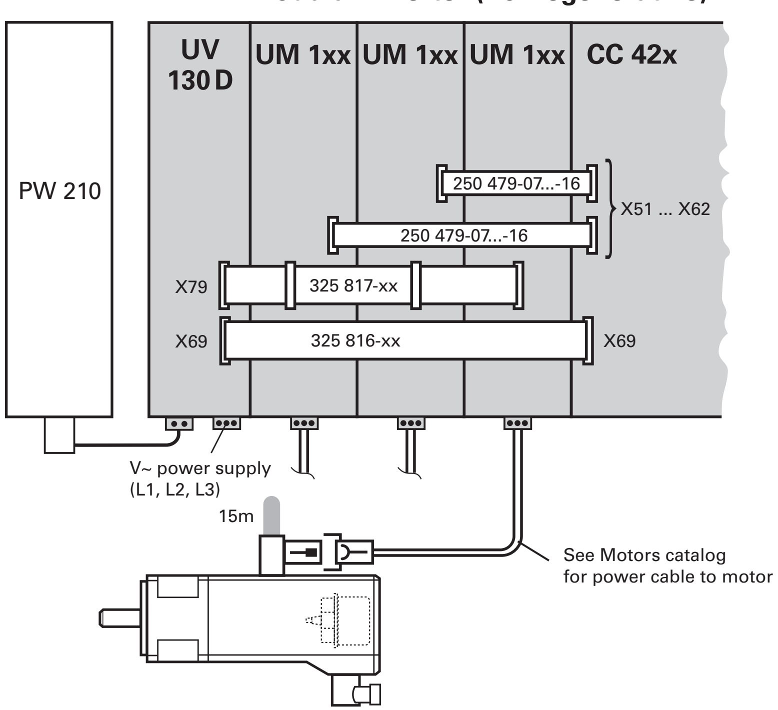
3.37 Cable Overview for iTNC 530 – Basic Configuration

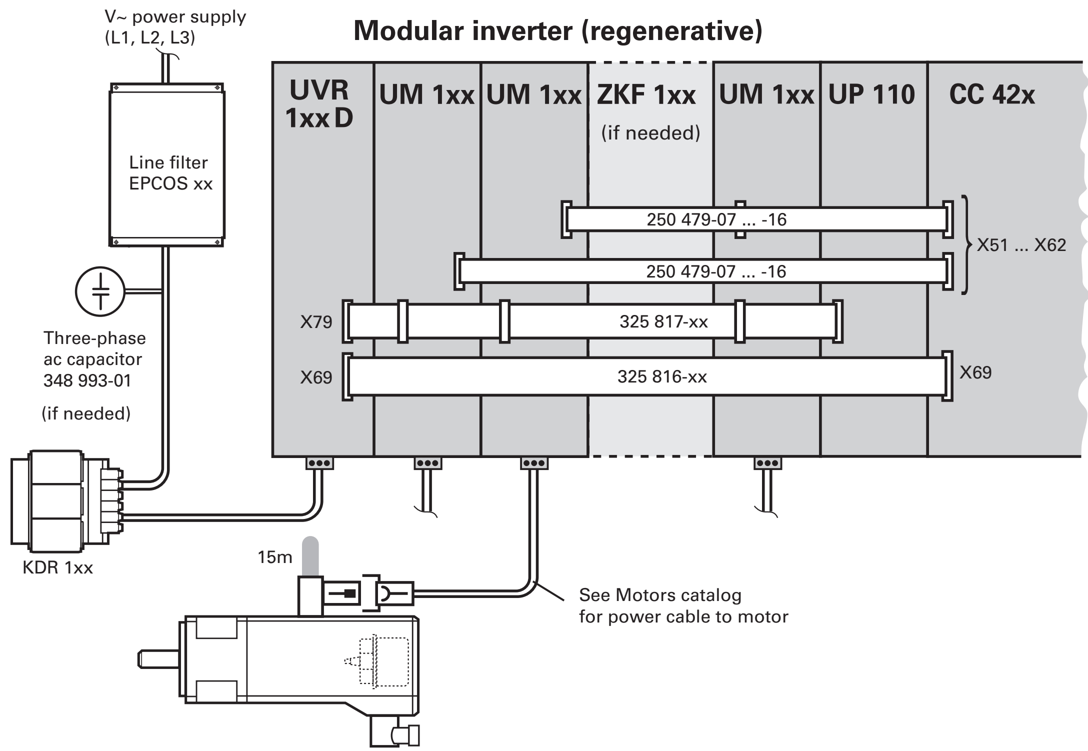


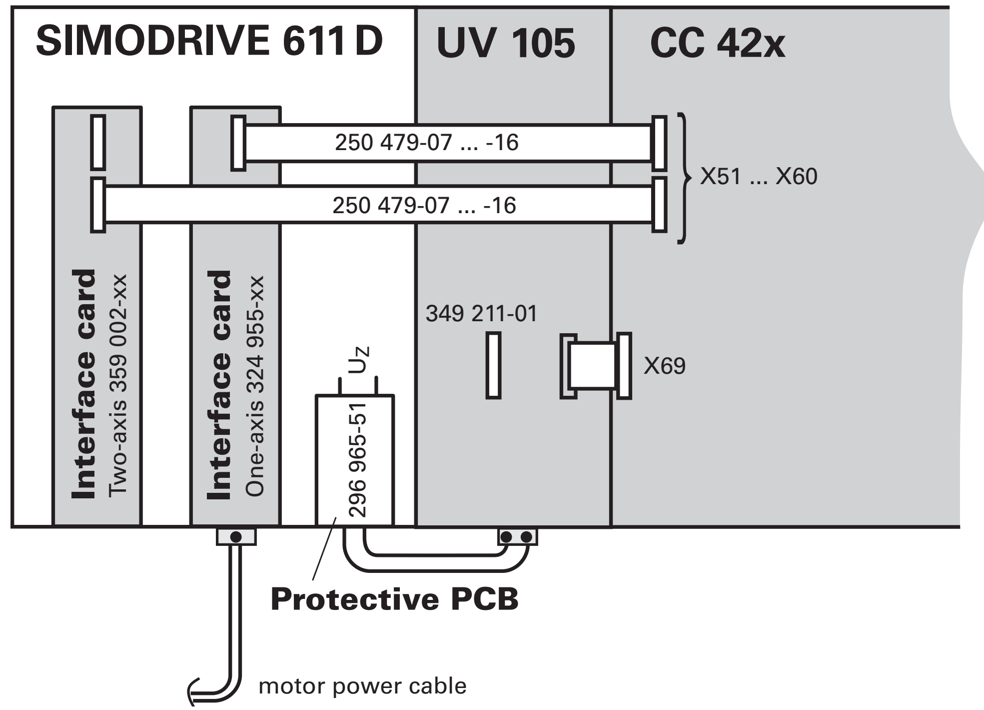
3.38 Cable Overview for iTNC 530 with HEIDENHAIN Inverter Systems

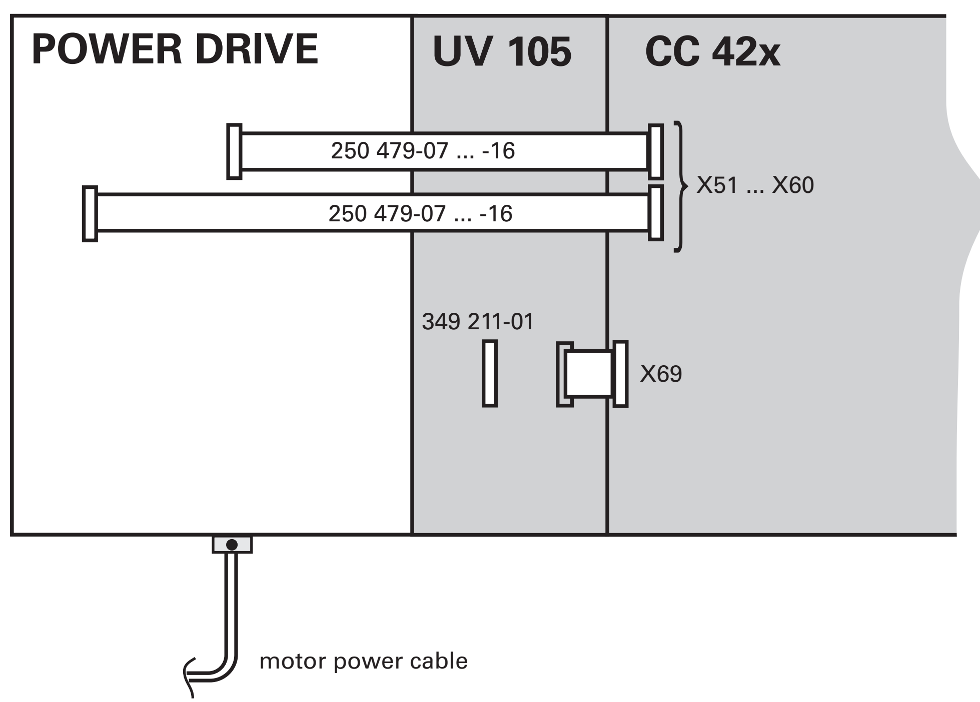




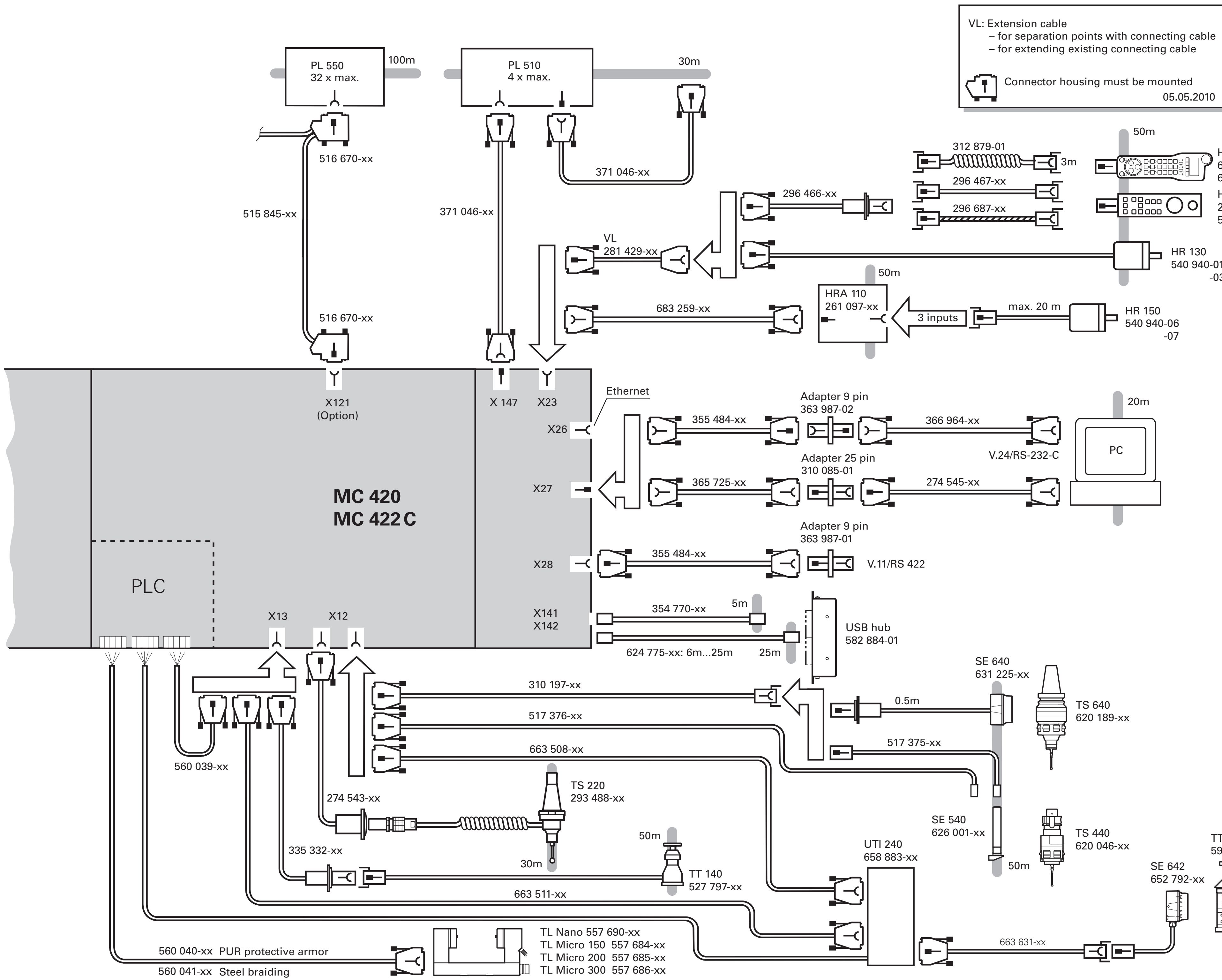





SIMODRIVE 611 D
UV 105
CC 42x
250 479-07 ... -16
250 479-07 ... -16
X51 ... X60
Interface card
Two-axis 359 002-xx
Interface card
One-axis 324 955-xx
349 211-01
U_Z
X69
296 965-51
Protective PCB
motor power cable


POWER DRIVE
UV 105
CC 42x
250 479-07 ... -16
250 479-07 ... -16
X51 ... X60
349 211-01
X69
motor power cable

3.40 Cable Overview for iTNC 530 – Accessories



4 Machine Parameters

4.1 What is a Machine Parameter?

A contouring control must have access to specific data (e.g., traverse distances, acceleration) before it can execute its programmed instructions. You define these data in machine parameters.

This list of machine parameters is divided into groups according to topic.

Machine parameters	Topics
10 to 999	Encoders and machines
1000 to 1399	Positioning
1400 to 1699	Operation with velocity feedforward control
1700 to 1999	Operation with following error (servo lag)
2000 to 2999	Integrated speed and current control
3000 to 3999	Spindle
4000 to 4999	Integral PLC
5000 to 5999	Data interface
6000 to 6199	3-D touch probe
6500 to 6599	Tool measurement with triggering touch probe
7100 to 7199	Tapping
7200 to 7349	Display and programming
7350 to 7399	Colors
7400 to 7599	Machining and Program Run
7600 to 7699	Hardware

If there is more than one input value for a single function (e.g., a separate input for each axis), the parameter number is extended by indices. Index zero is always axis 1, index one is axis 2, etc.

Example:

MP1010.0-8	**Rapid traverse**
MP1010.0	Rapid traverse for axis 1
MP1010.1	Rapid traverse for axis 2
MP1010.2	Rapid traverse for axis 3
MP1010.3	Rapid traverse for axis 4
MP1010.4	Rapid traverse for axis 5
MP1010.5	Rapid traverse for axis 6
MP1010.6	Rapid traverse for axis 7
MP1010.7	Rapid traverse for axis 8
MP1010.8	Rapid traverse for axis 9

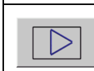
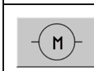
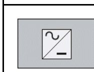
Enter into OEM.SYS, using the code word **`AXISNUMBER =`**, the number of axes being used, so that only the necessary index parameters are displayed.

With other machine parameters you can activate specific functions. In this case, the parameters serve as on/off switches for these functions. These parameters are bit-encoded. Each bit is assigned either to an axis or a function.

4.2 The "Machine Parameter Programming" Mode of Operation

- Enter the code number 95148 to access the **`Machine Parameter Programming`** mode of operation

Meaning of the soft keys in the **`Machine Parameter Programming`** mode of operation:

Meaning of the soft keys:	
INSERT / OVERWRITE	Switch between insertion and overwrite modes
MOVE WORD >>	Jump to the beginning of the next word in the line
MOVE WORD <<	Jump to the beginning of the previous word in the line
PAGE ↑	Go back one page in the machine parameter file
PAGE ↓	Go forward one page in the machine parameter file
BEGIN ↑	Jump to the beginning of the machine parameter file
END ↓	Jump to the end of the machine parameter file
FIND	Search the machine parameter file for a text string
DELETE CHAR	Delete the character covered by the cursor
DELETE WORD	Delete the word that the cursor is in
DELETE LINE	Delete the line that the cursor is in
INSERT LINE / WORD	Reinsert last deleted word or line
	Open the selection list for power modules
	Open the selection list for motors
	Open the selection list for power supply modules

4.3 Input and Output of Machine Parameters

If the machine parameters have not yet been entered in a HEIDENHAIN contouring control (e.g., before commissioning), the iTNC presents the list of machine parameters after the memory test:

- Enter the values for the machine parameters either by hand on the keyboard or download them through the data interface.

4.3.1 Input Format

You can enter the input values in decimal, binary (%) or hexadecimal ($) format.

- Enter a number for each machine parameter.

The value represents, for example, the acceleration in mm/s^2 or the analog voltage in V. You can add a comment to your entry by preceding it with a semicolon (;). Binary input (%) is the best format for machine parameters that activate individual functions bit-encoded.

Example: Disabling soft keys for file types with MP7224.0

Bit 0	HEIDENHAIN programs	.H
Bit 1	DIN/ISO programs	.I
Bit 2	Tool tables	.T
Bit 3	Datum tables	.D
Bit 4	Pallet tables	.P
Bit 5	Text files	.A
Bit 6	Reserved	
Bit 7	Point tables	.PNT

The soft keys for datum tables and text files are to be disabled:

0: Do not disable
1: Disable

Input value for MP7224.0 =	Binary	%00101000
	Hexadecimal	$28
	Decimal	40 (32+8)

Special case: Entering a formula

Currently only for MP1054.x (linear distance of one motor revolution) and for MP7530 within description tables for the kinematics).

You can enter a formula instead of a fixed value. When entering the formula, you must pay attention to the case of the letters (whether they are small or capital). Functions are written small, variables are written in capitals.

Functions:

+	Addition	sin	Sine
–	Subtraction	cos	Cosine
·	Multiplication	tan	Tangent
/	Division	asin	Arc sine
log	Logarithm	acos	Arc cosine
log10	Logarithm to the base of 10	atan	Arc tangent
exp	Exponent	sqrt	Square root
()	Expressions in parentheses are solved	sqr	Square
^	Exponential calculation		

Variable:

"REF" or "x"	Current position of the axis relative to the machine datum (resolution 0.0001 mm or °) ‘x’ was simply introduced as a short form

Example:

`MP1054.0:` **`x*0.1^6+15`** (X in 0.0001 mm)

Entering a formula in MP1054.x is only useful if a position encoder is connected to the axis.

For machine parameters with formula input (e.g. MP1054), the formula can be indicated in an ASCII file. For identification, the character "@", followed by the absolute path name, is entered in the machine parameter.

Example:

`@PLC:\MP\1054-X.A`

The file must contain only the formula. Comments are not allowed. The formula in the file may consist of several lines and contain blank lines. The first 1024 characters of the file are evaluated.

4.3.2 Activating the Machine Parameter List

After you have entered all the values for the machine parameters:

- Exit the machine parameter list by pressing the END key.

Missing or incorrect entries result in error messages from the control that prompt you to correct your entry. The following errors are displayed:

Input error	Meaning
0	No MP number found
1	Invalid MP number
2	No separator (:) found
3	Input value incorrect
4	MP doubly defined
6	MP cannot be stored

If the control does not recognize any errors, it automatically exits the machine parameter editor and is ready for operation.

If you do not make any entries in the machine parameter list during initial commissioning and exit the editor with the END key, the iTNC generates a standard machine parameter list (MP NAME). In this list the iTNC is defined as a programming station with the HEIDENHAIN standard colors. In all other machine parameters a default value is entered.

You can enter more than one machine parameter list in the iTNC:

- Select the lists with the PGM MGT key and the **SELECT** soft key. The last selected machine parameter list becomes active when you exit the machine parameter editor.

4.3.3 Changing the Input Values

A machine parameter list can be changed either with the machine parameter editor or directly through the PLC. The "List of Machine Parameters" includes the following symbols:

Indicator	Change by / Reaction
CN123	The MP is also accessible through the code number 123.
PLC	The MP can be changed via the PLC; it can also be changed in a running NC program during a strobe output.
RUN	The MP can also be changed while an NC program is running.
RESET	Changing the MP results in a reset.
REF	The axis must be moved over the reference mark again.

Manual input

- Call the machine parameter editor through the MOD function "code number":
  - By entering the code number **95148** you gain access to the complete list of machine parameters.
  - By entering the code number **123** you gain access to a subset of machine parameters. This subset can be changed by the user (see the User's Manual). Machine parameters that can be accessed through the code number 123 are indicated in the list with the symbol CN123.
- To exit the machine parameter editor, press the END key.

User parameters

You can access some machine parameters without first entering a code number.

- In MP7330.x, define up to 16 machine parameters and define the associated dialog in MP7340.x. The dialog is shown whenever the **USER PARAMETERS** soft key is pressed (up to 37 characters).
- Select the MOD function **USER PARAMETER.**

Protecting the machine parameter list

To protect the current machine parameter list from being edited through the code number 95148:

- In the OEM.SYS file, define a new code number in the entry **MPPASSWORD =** for editing the machine parameter list. Then it is **no longer possible** to edit through the code number 95148.

To protect individual machine parameters against editing:

- In the **MPLOCKFILE =** entry in the OEM.SYS file, enter the path of a machine-parameter subfile. Then it is only possible to edit those machine parameters that have no value assigned in this file. If there is a difference between the current MP value and the MP value in this subfile, the control displays an error message and a window offering the value from the subfile for your acceptance.

The following code numbers control access rights to MPs and protected MPs in the iTNC 530:

- Entry of code number 95148
  Reading and editing of all machine parameters that are not listed in **MPLOCKFILE**. Protected MPs can be read but not edited. This code number cannot be changed, and is therefore always valid.
- Entry of the code number for protected MPs
  Reading and editing of all machine parameters and protected machine parameters. The code number for protected MPs can be changed via the keyword **MPPASSWORD =** in the OEM.SYS file, in order to prevent protected MPs from unauthorized changes. The old code number thereby becomes invalid.
- PLC password
  Access to the OEM.SYS file is only possible upon entry of the PLC password. The OEM can change the PLC password via the **PLCPASSWORD =** entry in order to protect the OEM.SYS file from unauthorized changes.

Overwriting machine parameters

Machine parameters can be overwritten by the PLC or from an NC macro. In the **Program run, full sequence, Program run, single block** and **Positioning with manual data input** operating modes, machine parameters can be overwritten only when the drives are stationary, and not during a movement.

> Attention
>
> In the **Manual** and **Electronic Handwheel** operating modes machine parameters should not be overwritten during a movement, since this might result in critical conditions.

Changing the input values via PLC

You can also change the machine parameters through the PLC. The following modules are available for this purpose

- Module 9031 Overwrite machine parameter
- Module 9032 Read machine parameter
- Module 9310 Read the machine parameter from the run-time memory
- Module 9033 Select machine parameter file
- Module 9034 Load machine parameter subfile
- Module 9312 Overwrite machine parameters in the active MP file or in the run-time memory
- Module 9313 Read machine parameters from the active MP file or from the run-time memory

The machine parameters that you can change with Module 9031, Module 9034 or Module 9312 are indicated with PLC in the overview.

Module 9031 Overwrite machine parameter

With this module you can overwrite the value of the given machine parameter with a new value. The input value must be a natural number with the decimal point shifted by the number of possible decimal places.

Example:

MP910.0 = 100.12 [mm] Transfer value: 1001200 (4 decimal places)

The value in the run-time memory is changed. The value from the editable machine parameter file does not change. The old value becomes valid again after the machine parameter file is edited and exited.

For non-indexed machine parameters, zero must be transferred as the index. Once the NC program has started, the module operates only during the output of an M/S/T/Q strobe.

Call only in a submit job.

Call:

```
PS   B/W/D/K   <MP number>
PS   B/W/D/K   <MP index>
PS   B/W/D     <MP value>
CM   9031
PL   B/W/D     <Error code>
                 0: No error
                 1: MP does not exist / is not changeable / is not changeable
                 during a running program
                 2: MP value out of range
                 3: Error while saving (fatal error)
                 4: Call was not in a submit or spawn job
                 5: Call during running program without strobe.
```

Error recognition:

Marker	Value	Meaning
M4203	0	MP was overwritten
	1	MP could not be overwritten

Module 9032 Read machine parameter

With this module you can read the value of the given machine parameter from the active machine parameter file. The input value is transferred as a natural number with the decimal point shifted by the number of possible decimal places.

Only the value from the editable machine parameter file is read, not any value modified in the run-time memory by PLC Module 9031.

For non-indexed machine parameters, zero must be transferred as the index.

Call only in a submit job.

Call:

```
PS    B/W/D/K   <MP number>
PS    B/W/D/K   <MP index>
CM    9032
PL    B/W/D     <MP value / Error code>
                1: MP number does not exist
                2: No separator (:)
                3: MP value out of range
                4: MP not found in file
                5: No MP file found
                6: Call was not in a submit or spawn job
                7: MP is of the "string" type
                8: No system memory
```

Error recognition:

Marker	Value	Meaning
M4203	0	MP was read
	1	MP could not be read from the table

Module 9310 Read the machine parameter from the run-time memory

Use this module to read the value of the given machine parameter from the run-time memory. The input value is transferred as a natural number with the decimal point shifted by the number of possible decimal places. Machine parameters whose contents exceed the 32-bit limit cannot be read.

A value is read from the run-time memory.

For non-indexed machine parameters, zero must be transferred as the index.

Call:
```
PS    B/W/D/K   <MP number>
PS    B/W/D/K   <MP index>
CM    9310
PL    B/W/D     <MP value/error code>
                1: MP number does not exist
                3: MP outside value range
                6: Call was not in a submit or spawn job
                7: MP is of the "string" type
                8: No system memory
```

Error recognition:

Marker	Value	Meaning
M4203	0	MP was read
	1	Error code in W1022
W1022	20	Module was not called in a spawn job or submit job

Module 9312 Change machine parameters in the current machine parameter file

Module 9312 enables you to dynamically overwrite the values of the machine parameters in the active machine parameter file and the DSP process memory, or only in the DSP process memory. For machine parameters defined as numerical values, the new value can also be programmed as a string.

Conditions:

- For numbers the values must be returned as an integer. The decimal point is shifted by the number of possible decimal places. For example, if MP910.0 is to be set to 100.12 mm, the transferred <MP value > must be equal to 1001200. (4 possible decimal places lead to multiplication by 10 000).
- For non-indexed machine parameters, zero must be programmed as the index.
- Once the NC program has started, the module operates only during the output of M/G/S/T/T2/Q (strobe) change signals.
- Depending on the changed machine parameter’s type, the geometry is re-initialized.
- Not every MP can be changed by the PLC.
- If a new value for a machine parameter defined as a numerical value is programmed as a string, the value is converted to the decimal system (even if the new value is programmed in the "$0004" or "%0000011111" format, for example).
- If the new values are immediately transferred after the modification of the active machine-parameter file and the process memory (bits 1 and 2 not set in the parameter "Mode"), the marker 4174 (first run after MP was changed) is set.
- If the new machine parameters are not transferred immediately, the new values will become effective when the module is called again without bit 2 being set.

Call:
PS B/W/D/K <Mode>
Bit 0:
0 = Modify MP numerical value (MP is a number)
1 = Modify MP string (MP is a string)
Bit 1:
0 = Modify parameter in the file and in the process memory (Bit 2 is then also relevant)
1 = Modify parameter in run-time memory (Behavior like Module 9031)
Bit 2:
0 = Change the file and the run-time memory and immediately transfer value(s) into the NC
1 = Change the file and the run-time memory, do not immediately assume new value(s) in NC Will be assumed at next call with bit 2 = 0
PS B/W/D/K/S<File name>
not supported at present
(a value must be transferred (e.g. S""), but the value is not evaluated yet)
PS B/W/D/K <MP number>
PS B/W/D/K <MP index>
PS B/W/D/K/S<MP value>
Depending on the mode, the MP value is interpreted as a PLC string number, PLC constant string or number
CM 9312
PL B/W/D <Error code>
0: No error
1: Parameter does not exist or cannot be changed
2: New value for parameter is invalid
3: Error while saving
4: Call was not in a submit or spawn job
5: Call during running NC program without change signal
6: Invalid PLC string for file name or MP value
7: Parameter file does not exist

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	See error code

Module 9313 Read machine parameters

Module 9313 reads the contents of machine parameters from the process memory or from the current machine-parameter file. For machine parameters defined as numerical values, the content can also be read as a string. Machine parameters defined as strings cannot be read as numbers.

Machine parameters that have no function in the currently active machine configuration (highlighted in the MP editor) previously reported the value 0 when they were read by PLC module 9313. As of software 340 49x-05, the value entered in inactive machine parameters will also be reported. For MPs that are not in the MP file, the default value is read.

Call:
PS B/W/D/K <Mode>
Bit 0:
0 = Read MP as numerical value (MP must be a number)
1 = Read MP as string (MP can be a number or text)
Bit 1:
0 = Read MP from file
1 = Read MP from memory
PS B/W/D/K/S<File name>
not supported at present
(a value must be transferred, but the value entered is not evaluated yet)
(only the active MP file can be read from)
PS B/W/D/K <MP number>
PS B/W/D/K <MP index>
PS B/W/D/K/S<Target word or target string>
Mode bit 0 = 0: Interpretation as a double-word address
Mode bit 0 = 1: Interpretation as a PLC string number
CM 9313
PL B/W/D <Error>
0: No error
1: MP number, PLC string or PLC double-word address invalid
2: No ":" separator in MP file, e.g. (MP 10: %0111)
3: MP value outside permissible value range
4: MP not found in file
5: No MP file found
6: Call was not in a submit or spawn job
7: MP (defined as string) cannot be read as a number
8: No system memory

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	See error code

Module 9314 Activate/Deactivate machine-parameters

PLC module 9314 is used to axis-specifically activate/deactivate other values from machine parameters during run time.

Condition:

- The module transmits the information at the end of a PLC cycle.
- After the PLC program has been restarted, the module will no longer be effective and the original configuration will become active again.
- HEIDENHAIN recommends using power limiting (mode 0) via MP2393.x only for spindles. Power limiting for axes might cause positioning errors.

Call:
PS K/B/W/D <Bit mask for axes/spindles>
Bit for axis/spindle = 1: Value from MPxxxx.y is active
Bit for axis/spindle = 0: Value from MPxxxx.y is not active
PS K/B/W/D <Mode>
0: Activation/Deactivation of power limiting (MP2393.x)
CM 9314

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Invalid value for this mode is active
	24	Call was not from a cyclic program

Module 9033 Select machine parameter file

With this module you can select a new machine parameter file. If machine parameter files that set off a reset were changed, the control system will restart.

> Attention
>
> The module does not take any existing safety problems into account when setting off a control reset (e.g., axes and spindle coasting to a stop).

The file to be selected is checked; a faulty file is not selected. If file selection is successful, there is no return to the calling PLC program.

The file name is transferred in a string that must contain the complete path, name and file extension. Further characters, even space characters, are not permitted.

If the PLC program is created externally, ensure that lower-case letters are not used for the file name!

The path for the MP (sub)file can only be entered as absolute path (e.g. PLC:\BASIS\PROGRAM\MP\subfile1.A) or relative to the directory of the PLC main program (e.g. \MP\subfile1.A). If a relative path is entered, the MP (sub)file must be located in the directory of the PLC main program or an associated subdirectory. The possibility of entering a relative path eliminates the need for adapting the source code of the PLC program if the PLC program is moved to the encrypted PLCE partition.

Once the NC program has started, the module operates only during the output of an M/S/T/Q strobe.

Call only in a submit job.

Call:

```
PS    B/W/D/K    <String number>
CM    9033
         Note: If a new file is selected, program execution ends here.
PL    B/W/D      <Error code>
         0: No error. File was already selected.
         1: String does not contain a valid file name.
         2: File not found
         3: File is faulty
         4: Incorrect string number transferred
         5: Call was not in a submit job.
         6: Call during running program without strobe.
```

Machine parameter subfile

A machine parameter subfile can be activated via Module 9034, or from the NC program via **FN17: SYSWRITE** (also see page 1638).

Module 9034 Load a machine parameter subfile

With this module you load the contents of the given machine parameter into the main memory. All MPs not listed in this file remain unchanged.

The MP file to be selected is checked. A faulty file is not loaded. If the MP file contains parameters that require a system reset, the file is not loaded.

The file name is transferred in a string that must contain the complete path, name and file extension. Further characters, even space characters, are not permitted.

If the PLC program is created externally, ensure that lower-case letters are not used for the file name!

The path for the MP (sub)file can only be entered as absolute path (e.g. PLC:\BASIS\PROGRAM\MP\subfile1.A) or relative to the directory of the PLC main program (e.g. \MP\subfile1.A). If a relative path is entered, the MP (sub)file must be located in the directory of the PLC main program or an associated subdirectory. The possibility of entering a relative path eliminates the need for adapting the source code of the PLC program if the PLC program is moved to the encrypted PLCE partition.

Once the NC program has started, the module operates only during the output of an M/S/T/Q strobe.

Call only in a submit job.

Call:

```
PS   B/W/D/K     <String number>
        0 to 99
CM   9034
PL   B/W/D       <Error code>
        0: No error
        1: String does not contain a valid file name,
        or the name (including the path) is too long.
        2: File not found
        3: File is faulty / contains reset parameters
        4: Incorrect string number was transferred (0 to 3).
        5: Call was not in a submit job.
        6: Call during running program without strobe.
```

4.4 List of Machine Parameters

4.4.1 Encoders and Machines

MP	Function and input	Software version & behavior	Page
MP10	Active axes Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 correspond to axes 1 to 14 0: Axis not active 1: Axis active	PLC RUN	645
MP12	Axis-specific demo operation for NC axes Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Demo operation not active 1: Demo operation active	PLC RUN	
MP20 MP20.0 MP20.1 MP20.2	Monitoring functions for the axes Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Monitoring not active 1: Monitoring active Absolute position of the distance-coded reference marks Amplitude of encoder signals Edge separation of encoder signals	PLC RUN	652
MP21 MP21.0 MP21.1 MP21.2	Monitoring functions for the spindle Format: %xx Input: Bit 0 – Spindle 1 0: Monitoring not active 1: Monitoring active Bit 1 – Spindle 2 0: Monitoring not active 1: Monitoring active Absolute position of the distance-coded reference marks Amplitude of encoder signals Edge separation of encoder signals	PLC RUN	652
MP100 MP100.0 MP100.1 MP100.2	Designation of axes Format: -wvucbazyxWVUCBAZYX Input: Characters 1 to 9 from the right represent axes 1 to 9 Traverse range 1 Traverse range 2 Traverse range 3	PLC RUN	645, 669

MP	Function and input	Software version & behavior	Page
MP110.x	Assignment of position encoder inputs to the axes Input: 0: No position encoder input 1 to 6: Position encoder inputs X1 to X6 35 to 38: Position encoder inputs X35 to X38 201 to 214: Position encoder inputs X201 to X214	RESET	655
MP111.x	Position encoder input for the spindle/spindles Input: 0: No position encoder input 1 to 6: Position encoder inputs X1 to X6 35 to 38: Position encoder inputs X35 to X38 201 to 214: Position encoder inputs X201 to X214	REF	656, 944
MP111.0	Position encoder input for the first spindle		
MP111.1	Position encoder input for the second spindle		
MP112.x	Assignment of speed encoder inputs to the axes Input: 0: No speed encoder input 15 to 20: Speed encoder inputs X15 to X20 80 to 85: Speed encoder inputs X80 to X85	RESET	655
MP113.x	Speed encoder for the spindle/spindles Input: 0: No speed encoder input 15 to 20: Speed encoder inputs X15 to X20 80 to 85: Speed encoder inputs X80 to X85	REF	657, 947
MP113.0	Speed encoder for the first spindle		
MP113.1	Speed encoder for the second spindle		
MP115.0	Position encoder input 1 V_{PP} or 11 μA_{PP} Format: %xxxxxxxxxxx Input: Bit 0 to bit 5: Position encoder inputs X1 to X6 Bit 6 to bit 9: Position encoder inputs X35 to X38 Bit 10: No function 0: 1 V_{PP} 1: 11 μA_{PP}	RESET	650
MP115.1	Reserved Format: %xxxxxxxxxxx Input: Enter %00000000000		
MP115.2	Input frequency of position encoder inputs Format: %xxxxxxxxxxx Input: Bit 0 to bit 5: Position encoder inputs X1 to X6 Bit 6 to bit 9: Position encoder inputs X35 to X38 Bit 10: No function With 1 V_{PP}: 0: 27 kHz 1: 400 kHz With 11 μA_{PP}: 0: 27 kHz 1: 140 kHz		

MP	Function and input	Software version & behavior	Page
MP116.0	Only CC 424(B): Position encoder input 1 V_{PP} or 11 μA_{PP} Format: %xxxxxxxxxx Input: Bit 0 to bit 9: Linear encoder inputs X201 to X210 Bit 10: No function 0: 1 V_{PP} 1: 11 μA_{PP}	RESET 340 420-08, 340 422-02, 340 480-02	1071
MP116.1	Only CC 424(B): Reserved Format: %xxxxxxxxxxx Input: Enter %00000000000		
MP116.2	Only CC 424(B): Input frequency of the position encoder inputs Format: %xxxxxxxxxxx Input: Bit 0 to bit 9: Linear encoder inputs X201 to X210 Bit 10: No function With 1 V_{PP}: 0: 27 kHz 1: 400 kHz With 11 μA_{PP}: 0: 27 kHz 1: 140 kHz		

MP	Function and input	Software version & behavior	Page
MP120.x	Nominal speed command outputs of the axes Input: 0: No servo-controlled axis 1 to 6: Analog outputs 1 to 6 at terminal X8 7 to 12: Analog outputs 7 to 12 at terminal X9 51 to 62: Digital output X51 to X62	RESET	655
MP121.0	Nominal speed command output of the first spindle Input: 0: No servo-controlled axis 1 to 6: Analog outputs 1 to 6 at terminal X8 7 to 12: Analog outputs 7 to 13 at terminal X9 51 to 62: Digital output X51 to X62	RESET	657
MP121.1	Nominal speed command output of the second spindle Input: 0: No servo-controlled axis 1 to 6: Analog outputs 1 to 6 at terminal X8 7 to 12: Analog outputs 7 to 13 at terminal X9 51 to 62: Digital output X51 to X62	RESET	
MP130.x	Y index of the machine parameters MP2xxx.y for the axes Input: 0 to 12	PLC RUN	655
MP131.x	Y index of the machine parameters MP2xxx.y for the spindle(s) in operating mode 0 Input: 0 to 12	PLC RUN	657
MP131.0	Index for the first spindle		
MP131.1	Index for the second spindle		
MP132.x	Y index of the machine parameters MP2xxx.y for the spindle(s) in operating mode 1 Input: 0 to 12	PLC RUN	657
MP132.0	Index for the first spindle		
MP132.1	Index for the second spindle		
MP210	Counting direction of position encoder output signals Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Positive 1: Negative	REF	651

MP	Function and input	Software version & behavior	Page
MP331.x	Distance for the number of signal periods in MP332 Input: 0.0001 to +1.797693135E+308 [mm] or [°]	PLC RUN REF	649
MP332.x	Number of signal periods for the distance in MP331 Input: 1 to +1.797693135E+308	PLC RUN REF	649
MP334.x	Nominal increment between two fixed reference marks on encoders with distance-coded reference marks Input: 1 to 65 535 0: 1 000	PLC RUN REF	649
MP340.x	Interpolation factor for external interpolation Input: 0 to 99 0 = 1: No external interpolation	RESET	649
MP410 MP410.3 MP410.4 MP410.5	Assignment of axis keys IV, V and VI Input: Axis labels A/B/C/U/V/W/T IV axis key V axis key VI axis key (only HR 5xx)	PLC RUN 340 490-05	646
MP420.x	Hirth coupling Input: 0: No Hirth coupling 1: Hirth coupling	PLC RUN	1478
MP430.x	Prescribed increment for Hirth coupling Input: 0.0000 to 30.0000 [°]	PLC RUN	1478
MP708.x	Only CC 424(B): Path for acceleration-dependent backlash compensation Input: 0.0001 to 1.000 [mm] 0: Function inactive	340 490-06 PLC RUN	688
MP709.x	Time constant for backlash compensation Input: 1 to 1000 [ms]	PLC RUN	688
MP710.x	Backlash compensation Input: –9.9999 to +9.9999 [mm] or [°]	PLC RUN 340 490-05	688
MP711.x	Height of peaks during circular movement (analog only) Input: –1.0000 to +1.0000 [mm] (digital: 0)	PLC RUN	700
MP712.x	Compensation value per control loop cycle time Input: 0.000000 to 99.999999 [mm] (digital: 0)	PLC RUN	700
MP715.x	Height of peaks during circular movement (analog only) with M105 Input: -1.0000 to +1.0000 [mm] (digital: 0)	PLC RUN	700

MP	Function and input	Software version & behavior	Page
MP716.x	Compensation value per control loop cycle time with M105 Input: 0.000000 to 99.999999 [mm] (digital: 0)	PLC RUN	700
MP720.x	Linear axis error compensation Input: –1.000 to +1.000 [mm/m]	PLC RUN	690
MP730	Selection of linear/nonlinear axis error compensation Format: %xxxxxxxxxxxxxx Input: Bits 0 to 3 represent axes 1 to 14: 0: Linear axis error compensation 1: Nonlinear axis error compensation	PLC RUN	690, 696
MP732	Nonlinear axis-error compensation for rotary axes Format: %xxxxxxxxxxxxxx Input: Bits 0 to 3 represent axes 1 to 14: 0: Not active (usual compensation) 1: Active (mapped to traverse range)	340 490-03	696
MP750.x	Reversal error (backlash compensation) Input: -9.9999 to +9.9999 [mm] or [°]	PLC RUN 340 490-04	689
MP752.x	Compensation time for reversal error Input: 0 to 1000 [ms]	PLC RUN	689
MP810.x	Display mode for rotary axes and PLC auxiliary axes Input: 0.0000 to 99 999.9999 [°] 0: Display +/–99 999.9999 1: Modulo value for display	PLC RUN REF	1141
MP812	Activate software limit switches for tilting axes with modulo display, M94 and encoders with EnDat interface Format: %xxxxxxxxxxxxxx Input: Bits 0 to 3 represent axes 1 to 14: 0: Software limit switch not active 1: Software limit switch active	RESET	1141
MP850.x	Synchronized axes Input: 0: Master axis 1: Slave axis to axis 1 2: Slave axis to axis 2 3: Slave axis to axis 3 4: Slave axis to axis 4 5: Slave axis to axis 5 6: Slave axis to axis 6 7: Slave axis to axis 7 8: Slave axis to axis 8 9: Slave axis to axis 9	PLC RUN	787
MP855.x	Synchronization monitoring Input: 0 to 100.0000 [mm] 0: Monitoring not active	PLC RUN	789

MP	Function and input	Software version & behavior	Page
MP860.x	Datum for synchronous control Format: %xxx Input: Bit 0 – Datum position for synchronous control 0: Datum at position after switch-on 1: Datum at reference marks (machine datum) Bit 1 – Master-slave torque control 0: Not a torque slave axis 1: Axis is torque slave axis Bit 2 – Brake test of slave axis 0: No automatic brake test of the slave axis 1: Brake test of the slave axis is automatically run together with master axis.	PLC RUN 340 490-06	789, 794
MP910.x	Positive software limit switches, traverse range 1 (default setting after power on) Input: –99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	666
MP911.x	Positive software limit switches, traverse range 2 Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	666
MP912.x	Positive software limit switches, traverse range 3 Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	666
MP920.x	Negative software limit switches, traverse range 1 (default setting after power on) Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	666
MP921.x	Negative software limit switches, traverse range 2 Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	666
MP922.x	Negative software limit switches, traverse range 3 Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	666
MP950.x	Datum for positioning blocks with M92 for axes 1 to 9 Input: –99 999.9999 to +99 999.9999 [mm] or [°] Values with respect to the machine datum	PLC RUN	1194
MP951.x	Simulated tool-change position for TOOL CALL during mid-program startup (block scan) Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	1227
MP960.x	Machine datum Input: –1.79769313486E+308 to +1.79769313486E+308 [mm] or [°] Values with respect to the scale reference point	PLC RUN	805, 1194

4.4.2 Positioning

MP	Function and input	Software version & behavior	Page
MP1010.x	Rapid traverse Input: 10 to 1 000 000 [mm/min or °/min]	PLC RUN	853
MP1011	Limit of rapid traverse on the path Input: 10 to 1 000 000 [mm/min or °/min]	340 420-05 PLC RUN	853
MP1012.x	Second axis-specific rapid traverse Input: 10 to 1 000 000 [mm/min or °/min]	340 490-03	853
MP1020.x	Manual feed Input: 10 to 300 000 [mm/min]	PLC RUN	853
MP1030.x	Positioning window Input: 0.0001 to 2.0000 [mm]	PLC RUN	904
MP1040	Analog axes: Polarity of nominal value voltage Digital axes: Algebraic sign of the nominal speed value Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Positive 1: Negative		651
MP1050.x	Analog axes: Analog voltage at rapid traverse Input: 1.000 to 9.000 [V] Digital axes: without function Input: 1	PLC RUN	853
MP1054.x	Distance of a motor revolution (mm °) Input: Analog axes: Without function Digital axes: Entry of a formula possible, *** "Special case: Entering a formula" on page 525 ***	PLC RUN	903
MP1060.x	Acceleration Input: 0.001 to 500 [m/s^2]	PLC RUN 340 490-04	820
MP1061	Limitation of the path acceleration Input: 0.001 to 500 [m/s^2]	340 490-04 PLC RUN	820
MP1070	Radial acceleration Input: 0.001 to 500 [m/s^2]	PLC RUN 340 490-04	894
MP1080.x	Analog axes: Integral factor for offset adjustment Input: Enter 0 to 65 535 Digital axes: No function Input: 0	PLC RUN	893

MP	Function and input	Software version & behavior	Page
MP1085.x	Maximum permissible axis-specific jerk for path movements in the operating modes Program Run Full Sequence, Program Run Single Block, and Positioning with Manual Data Input Input: 0.0 to 9999.9 [m/s^3 or °/s^3]	340 490-04	817
MP1086.x	Maximum permissible axis-specific jerk for rapid traverse movements in the operating modes Program Run Full Sequence, Program Run Single Block, and Positioning with Manual Data Input Input: 0: Function inactive 0.0 to 9999.9 [m/s^3 or °/s^3]	340 490-04 PLC RUN	817
MP1087.x	Maximum permissible axis-specific jerk for Manual mode Input: 0.1 to 1000.0 [m/s^3 or 1000° /s^3]	PLC RUN	821
MP1088.x	Axis-specific jerk limiting for unfiltered positioning movements Input: 0.1 to 9999.9 [m/s^3]	340 490-05	821
MP1089.x	Max. permissible axis-specific jerk for Pass Over Reference Point mode Input: 0.1 to 1000.0 [m/s^3 or 1000°/s^3]	PLC RUN	821
MP1090	Maximum permissible jerk on the tool path Input: 0: Not active 0.0 to 9999.9 [m/s^3 or °/s^3]	PLC RUN 340 490-04	821
MP1090.0	For movements not at rapid traverse or feed rate < MP1092		
MP1090.1	For movements at rapid traverse or feed rate > MP1092		
MP1092	Feed rate threshold for MP1085.x and MP1086.x Input: 1 to 300 000 [mm/min] (previous behavior) 0: Not active	PLC RUN 340 490-03	821
MP1094	HSC filters Input: 0: HSC filter inactive 0.1 to 166.0: Cutoff frequency for HSC filter	As of 340 490-01, 340 492-01 only via **MPMODE = 340422** in OEM.SYS	840
MP1095	Nominal position value filter Input: 0: Single filter 1: Double filter	PLC RUN As of 340 490-01, 340 492-01 only via **MPMODE = 340422** in OEM.SYS	840
MP1095.0	In the Program Run Full Sequence, Program Run Single Block, and Positioning With Manual Data Input operating modes		
MP1095.1	In the Manual, Handwheel, Jog Increment and Pass Over Reference Point operating modes		

MP	Function and input	Software version & behavior	Page
MP1096	Tolerance for contour transitions at corners Input: 0: No nominal position value filter 0.001 to 3.000 [mm]	PLC RUN As of 340 490-01, 340 492-01 only via **MPMODE = 340422** in OEM.SYS	840, 895
MP1096.0	With machining feed rate		
MP1096.1	With rapid traverse		
MP1097.x	Maximum permissible axis-specific jerk (single/HSC filter) Input: 0.1 to 1000.0 [m/s^3 or $1000°/s^3$]	PLC RUN As of 340 490-01, 340 492-01 only via **MPMODE = 340422** in OEM.SYS	840
MP1098.x	Maximum permissible axis-specific jerk (double/HSC filter) Input: 0.1 to 1000.0 [m/s^3 or $1000°/s^3$]	PLC RUN As of 340 490-01, 340 492-01 only via **MPMODE = 340422** in OEM.SYS	840
MP1099	Minimum filter order Input: 0 to 20	PLC RUN As of 340 490-01, 340 492-01 only via **MPMODE = 340422** in OEM.SYS	840
MP1099.0	Minimum filter configuration for single filter (MP1095 = 0)		
MP1099.1	Minimum filter configuration for double filter (MP1095 = 1)		
MP1110.x	Standstill monitoring Input: 0.0010 to 30.0000 [mm]	PLC RUN	904
MP1120.x	Standstill monitoring when determining the field angle Input: 0.0000 to 300.0000 [mm] or [°]	340 422-03, 340 480-03 PLC RUN	1017

MP	Function and input	Software version & behavior	Page
MP1140.x	Threshold above which the movement monitoring functions Input: Analog axes: 0.030 to 10.000 [V] Digital axes: 0.030 to 10.000 [1000 min] Recommended: 0.030 [1000 min]	PLC RUN	903
MP1144.x	Motion monitor for position and speed Input: Analog axes: Without function Digital axes: 0 to 99 999.999 [mm] 0: No monitoring	PLC RUN	903

MP	Function and input	Software version & behavior	Page
MP1146.x	Difference between the position at shutdown and the position read in via the EnDat interface Input: 0.0000 to 300.0000 [mm] or [°] 0: No difference permitted	340 420-05 PLC RUN	901
MP1150.0	Delay time for deleting the nominal velocity value with the erasable error message **EXCESSIVE SERVO LAG IN <AXIS>** Input: 0 to 65.535 [s] Recommended: 0	PLC RUN	855, 898, 900
MP1150.1	Time period for which the monitoring function is to remain off after the fast PLC input defined in MP4130.0 is set. Input: 0 to 65.535 [s] 0: Monitoring functions on Recommended: 0.2 to 0.5		
MP1150.2	Minimum time period for which the monitoring functions are to remain effective after expiration of the time from MP1150.1. Input: 0 to 65.535 [s]		
MP1160	As of CC 424(B): LIFTOFF at powerfail Input: 0 to 30.0000 [mm] Default: 0.1 [mm]	340 490-04 PLC RUN	1092
MP1200	Selection of the nominal position value filter used Input: 0: Single filter 1: Double filter 2: HSC filter 3: Advanced HSC filter	340 490-01 PLC RUN	823
MP1201	Nominal position value filter in manual operation Input: 0: Single filter 1: Double filter	340 490-01 PLC RUN	823
MP1202	Predefined tolerance for Cycle 32 Input: 0.0000 to 3.0000 [mm]	340 490-01 PLC RUN	823
MP1202.0	Tolerance at corners for movements at machining feed rate		
MP1202.1	Tolerance at corners for movements at rapid traverse		
MP1205	Reduction of the contouring feed rate at the beginning of a contour element Input: 0: Not active (fast, possibly less precise) 1: Active (slow but likely more precise)	340 490-03	829
MP1210	Limit frequency for single filter Input: 0: Filter is switched off 0.0 to 166.0 [Hz]	340 490-01 PLC	823
MP1211	Limit frequency for double filter Input: 0: Filter is switched off 0.0 to 166.0 [Hz]	340 490-01 PLC	823

MP	Function and input	Software version & behavior	Page
MP1212	Limit frequency for HSC filter Input: 0: Filter is switched off 0.0 to 166.0 [Hz]	340 490-01 PLC	823
MP1213	Limit frequency for advanced HSC filter Input: 0: Filter is switched off 0.0 to 166.0 [Hz]	340 490-01 PLC	823
MP1222	Tolerance for curvature changes with HSC filter (only effective if MP7684 bit 4 = 0) Entry: 0: Do not include the tolerance 1: Include the tolerance	340 490-01 PLC RUN	823
MP1223	Tolerance for curvature changes with advanced HSC filter (only effective if MP7684 bit 4 = 0) Entry: 0: Do not include the tolerance 1: Include the tolerance	340 490-01 PLC RUN	823
MP1230.x	Max. permissible axis-specific jerk at corners for single filter Entry: 0.1 to 1000.0 [m/s^3]	340 490-01 PLC RUN	823
MP1231.x	Max. permissible axis-specific jerk at corners for double filter Entry: 0.1 to 1000.0 [m/s^3]	340 490-01 PLC RUN	823
MP1232.x	Max. permissible axis-specific jerk at corners for HSC filter Entry: 0.1 to 1000.0 [m/s^3]	340 490-01 PLC RUN	823
MP1233.x	Max. permissible axis-specific jerk at corners for advanced HSC filter Entry: 0.1 to 1000.0 [m/s^3]	340 490-01 PLC RUN	823
MP1240.x	Max. permissible axis-specific jerk at curvature changes for single filter Entry: 0.1 to 1000.0 [m/s^3]	340 490-01 PLC RUN	823
MP1241.x	Max. permissible axis-specific jerk at curvature changes for double filter Entry: 0.1 to 1000.0 [m/s^3]	340 490-01 PLC RUN	823
MP1242.x	Max. permissible axis-specific jerk at curvature changes for HSC filter Entry: 0.1 to 1000.0 [m/s^3]	340 490-01 PLC RUN	823
MP1243.x	Max. permissible axis-specific jerk at curvature changes for advanced HSC filter Entry: 0.1 to 1000.0 [m/s^3]	340 490-01 PLC RUN	823

MP	Function and input	Software version & behavior	Page
MP1250.x	Factor for axis-specific jerk at corners at rapid traverse (from value in MP123x.x) Input: 0.0000 to 30.0000 1: No change at rapid traverse	340 490-02 PLC RUN	823
MP1262	Only CC 424(B): Filter order used for HSC filter Input: 0 to 31 [filter order] 31: Default	340 490-02 PLC RUN	1092
MP1263	Only CC 424(B): Filter order used for advanced HSC filter Input: 0 to 31 [filter order] 31: Default	340 490-02 PLC RUN	1092
MP1290	Only with option #40: Maximum angle tolerance for DCM (Dynamic Collision Monitoring) Input: 0.0000° to 3.0000° 3: Default	340 490-02 PLC RUN	1379
MP1292	Only with option #40: Manual oversize for DCM (Dynamic Collision Monitoring) Input: 0 to 1000 [mm] 0: Default	340 490-02 PLC RUN	1379
MP1294	Only with option #40: Higher traversing speed for Dynamic Collision Monitoring (DCM) through movement of only a single axis Input: 0: Function inactive 1: Function active	340 490-06 PLC RUN	1375
MP1320	Direction for traversing the reference marks Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Positive 1: Negative	PLC RUN	805
MP1330.x	Velocity for traversing the reference marks Input: 80 to 1 000 000 [mm/min]	PLC RUN	805
MP1331.x	Velocity for leaving the reference mark end position for axes 1 to 9 (only for rotary encoders MP1350 = 2) Input: 10 to 1 000 000 [mm/min]	PLC RUN	805
MP1340.x	Sequence for traversing the reference marks Input: 0: No evaluation of reference marks 1 to 14: Axes 1 to 14	PLC RUN REF	805

MP	Function and input	Software version & behavior	Page
MP1350.x	Sequence for finding the reference mark Input: 0: Linear encoder with distance-coded reference marks (old routine) 1: Position encoder with one reference mark 2: Special type (length measurement with ROD) 3: Linear encoder with distance-coded reference marks (new routine) 4: Same as 3 except that two reference marks are evaluated 5: Encoder with EnDat interface 6: Reference pulse via fast PLC input	PLC RUN REF	806
MP1352	Activate the software limit switches before traversing the reference marks Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Software limit switch not active 1: Software limit switch active	340 490-04	803
MP1355	Double reference run Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Reference run as defined in MP1350.x 1: Double reference run	340 420-05 PLC RUN REF	806
MP1356.x	Distance between speed and position encoder for double reference run. Input: –99 999.999 to +99 999.999 [mm] or [°]	340 420-05 PLC RUN REF	806
MP1357.x	W1032 for double reference run Input: 0: Reset W1032 if the reference run has been over the EnDat interface of the speed encoder 1: Reset W1032 if the reference mark was traversed with the position encoder	340 422-05, 340 480-05 PLC RUN	806
MP1360.x	Fast PLC input for reference pulse Input: 0: No fast PLC input for reference pulse 1 to 5: Fast PLC input 1 to 5 (MP4130.x)	PLC RUN REF	806

MP	Function and input	Software version & behavior	Page
MP1391 MP1391.0 MP1391.1	Velocity and acceleration feedforward control in the MANUAL and HANDWHEEL operating modes Format: %xxxxxxxxxxxxxx Velocity feedforward control Input: Bits 0 to 13 represent axes 1 to 14 0: Inactive 1: Active Acceleration feedforward Input: Bits 0 to 13 represent axes 1 to 14 0: Inactive 1: Active	340 490-01 PLC RUN	701, 846
MP1392	Velocity feedforward in the operating modes Program Run Single Block, Program Run Full Sequence and Positioning with Manual Data Input Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Operation with following error (lag) 1: Operation with velocity feedforward control	PLC RUN	843
MP1396.x	Feedback control with velocity semifeedforward Input: 0.001 to 0.999 1: Velocity feedforward control	PLC RUN	849

4.4.3 Operation with Velocity Feedforward Control

MP	Function and input	Software version & behavior	Page
MP1410.x	Position monitoring for operation with velocity feedforward control (erasable) Input: 0.0010 to 30.0000 [mm] Recommended: 0.5 mm	PLC RUN	900
MP1420.x	Position monitoring for operation with velocity feedforward control (EMERGENCY STOP) Input: 0.0010 to 30.0000 [mm] Recommended: 2 mm	PLC RUN	900
MP1510.x	k_V factor for velocity feedforward control Input: 0.100 to 1 000.000 [(m/min)/mm]	PLC RUN	847
MP1511.x	Factor for static friction compensation Input: 0 to 16 777 215 [s]	PLC RUN	702
MP1512.x	Limitation of the amount of the static friction compensation Input: 0 to 16 777 215 [counting steps]	PLC RUN	702
MP1513.x	Feed-rate limitation for static friction compensation Input: 0 to 300 000 [mm/min]	PLC RUN	702
MP1515.x	k_V factor for velocity feedforward control effective after M105 Input: 0.100 to 1000.000 [m/(min*mm)]	PLC RUN 340 490-04	847
MP1516.x	k_V factor for velocity semifeedforward control Input: 0.100 to 20.000 [(m/min)/mm]	PLC RUN	849
MP1521	Transient response during acceleration and deceleration Input: 1 to 255 [ms] 0: Function inactive	PLC RUN	821
MP1522	Feed-rate smoothing Input: 0 to 60 [ms] 0: Function inactive	340 422-10, 340 480-10 PLC RUN	822

4.4.4 Operation with Following Error (Servo Lag)

MP	Function and input	Software version & behavior	Page
MP1710.x	Position monitoring for operation with following error (erasable) Input: 0.0000 to 300.0000 [mm] Recommended: 1.2 · following error	PLC RUN	900
MP1720.x	Position monitoring for operation with following error (EMERGENCY STOP) Input: 0.0000 to 300.0000 [mm] Recommended: 1.4 · following error	PLC RUN	900
MP1810.x	k_V factor for control with following error Input: 0.100 to 20.000 [(m/min)/mm]	PLC RUN	845
MP1815.x	k_V factor for control with following error effective after M105 Input: 0.100 to 20.000 [(m/min)/mm]	PLC RUN	845
MP1820.x	Multiplier for the k_V factor (as of 340 49x-03 also for CC424(B)) Input: 0.001 to 1.00000	PLC RUN	854
MP1830.x	Characteristic curve kink point (as of 340 49x-03 also for CC424(B)) Input: 0.000 to 100.000 [%]	PLC RUN	854

4.4.5 Integrated Speed and Current Control

MP	Function and input	Software version & behavior	Page
MP2040	Axis groups (for drive enabling through X150/X151) Format: %xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx (Bit #31 = spindle) Input: 0: Axis/spindle not assigned (disabling only through I32) 1: Axis/spindle assigned	PLC RUN	874
MP2040.0-2	Axis group 1 to 3		
MP2040.3-7	Reserved, enter %00000000000000		
MP2050	Functionality of drive enabling I32 (X42/33) MP2050 can also be overwritten by the PLC and the LSV2 protocol. Input: 0: Emergency stop for all axes, Module 9169 not effective 1: Emergency stop for all axes that are not excepted with Module 9169 2: I32 and Module 9169 have no function	340 490-03	874
MP2100.x	Type of axis power modules (change possible without automatic restart) Input: Name from file <Motor.amp>	PLC RUN 340 490-03	1001
MP2150	Signal for powerfail MP2150 can also be overwritten by the PLC and the LSV2 protocol. Input: 0: AC fail 1: Powerfail and AC fail 2: Reserved 3: Powerfail	340 490-03	906
MP2160.x	Field weakening with synchronous motors Input: 0: No voltage-protection module 1: Voltage-protection module present 2: Limited field weakening without voltage-protection module for EcoDyn motors		891
MP2170	Waiting time between the switch-on of the drive and the drive's standby signal Input: 0.001 to 4.999 [s] 0: 2 [s]		874
MP2172	Delay of the SH1Bsignal (inverter enable) at internal emergency stop (e.g. standstill monitoring, PLC via error table...) Input: 0 to 6 [s] as an integer 0: 3 [s] Default	340 490-02	932

MP	Function and input	Software version & behavior	Page
MP2173.x	Pulse switch-on of the power stage Input: 0.2 to 100.000 [s] 0: 3 [s] default	340 490-05	933
MP2180.x	PWM frequency Input: 0: f_{PWM} = 5000 Hz 3200 to 3999: f_{PWM} = 3333 Hz 4000 to 4999: f_{PWM} = 4166 Hz (CC 424(B): 4000 Hz) 5000 to 5999: f_{PWM} = 5000 Hz 6000 to 7999: f_{PWM} = 6666 Hz 8000 to 9999: f_{PWM} = 8333 Hz (CC 424(B): 8000 Hz) 10000: f_{PWM} = 10000 Hz	CC 422: RESET CC 424(B): PLC, RUN	1012
MP2182.x	Only CC 424(B): Cycle time of current controller at double the fundamental PWM frequency Input: 0: Standard case: MP2180 = [3333 to 5000 Hz] with single-speed axes MP2180 = [3333 to 10000 Hz] with double-speed axes (current controller cycle time = 1 / (2 * MP2180)) 1: Special case 1, only CC 424, not CC 61xx: MP2180 = [6666 to 10000 Hz] for high PWM frequencies with single-speed axes (current controller cycle time = 1 / MP2180) 2: Special case 2, only CC424, CC61xx: MP2180 = [3333 to 5000 Hz] with speed-dependent doubling (MP2186, MP2188) of the fundamental PWM frequency from MP2180 with double-speed axes (current controller cycle time = 1 / (4 * MP2180))	340 490-03 PLC RUN	1079
MP2184.x	Only CC424(B) (not CC61xx): Reserved Input: 0	340 490-03	
MP2186.x	As of CC424(B): Specifies the shaft speed at which the PWM frequency is switched to twice the PWM frequency Input: 0 to 100 000 [U/min]	340 490-03	1082
MP2188.x	As of CC424(B): Specifies the shaft speed at which the factor 2 PWM frequency is switched to a factor 1 Input: 0 to 100 000 [U/min]	340 490-03	1082
MP2190.x	As of CC424(B): dc-link voltage U_Z of the power supply module Input: 0 to 3000 [V] *: Entry from the power supply module table HEIDENHAIN inverters: Non-regenerative: 565 V Regenerative: 650 V	340 490-05	1020

MP	Function and input	Software version & behavior	Page
MP2192	As of CC424(B): Threshold sensitivity for LIFTOFF Input: 0 to 100 [%]	340 490-03	1094
MP2194	As of CC 424(B): DC-link voltage as of which the spindle is braked in a powerfail Input: 0 to 3000 [V]	340 490-03	1094
MP2195	Handling of status signals from HEIDENHAIN power supply units. MP2195 can also be overwritten by the PLC and the LSV2 protocol. Input: Bit 0 – Status signals that are already active during control power-up. 0: Missing signals are ignored 1: Missing signals are evaluated Bit 1– $\overline{\text{ERR.UZ.GR}}$ signal 0: Error message is not suppressed 1: Error message is suppressed Bit 2 – $\overline{\text{ERR.TMP}}$ signal 0: Error message is not suppressed 1: Error message is suppressed Bit 3 – Reserved Bit 4 – $\overline{\text{ERR.IZ.GR}}$ signal 0: Error message is not suppressed 1: Error message is suppressed Bit 5 – RDY.PS signal 0: Error message is not suppressed 1: Error message is suppressed Bit 6 – $\overline{\text{ERR.ILEAK}}$ signal 0: Error message is not suppressed 1: Error message is suppressed Bit 7 - PF.PS.AC 0: Error message is not suppressed 1: Error message is suppressed Bit 8 - PF.PS.DC 0: Error message is not suppressed 1: Error message is suppressed	340 490-04	925
MP2198.x	As of CC 424(B): Type of the power supply module Input: Name from file SUPPLY.SPY Default setting: Empty string	340 490-04	1001
MP2199.x	As of CC 424(B): Assignment of the drive to the power supply module Input: 0: The axis/spindle is assigned to the UV in MP2198.0 1: The axis/spindle is assigned to the UV in MP2198.1	340 490-04	1001
MP2200.x	Motor model Input: Name of the selected motor (is entered by the iTNC)	PLC RUN	1001

MP	Function and input	Software version & behavior	Page
MP2202.x	Overwrite "Line count" from the motor table Input: *: Input from the motor table active 0: No speed encoder (volts-per-hertz control mode) 1 to 999 999	340 420-05 PLC RUN	1001
MP2204.x	Overwrite "Counting direction" from the motor table Input: *: Input from the motor table active +: Positive counting direction –: Negative counting direction	340 420-05 RESET	1001
MP2206.x	Overwrite "Type of encoder" from the motor table Input: *: Input from the motor table active 0: No speed encoder (volts-per-hertz control mode) 1: Incremental rotary encoder with Z1 track 2: Absolute rotary encoder with EnDat interface (aligned) 3: Absolute linear encoder with EnDat interface 4: Linear motor with one reference mark (CC424(B)) 5: Absolute rotary encoder with EnDat interface (not aligned) 6: Incremental rotary encoder without Z1 track 7: Incremental rotary encoder with distance-coded reference marks (nonaligned) 8: Incremental linear encoder with distance-coded reference marks (not aligned)	340 420-05 RESET 340 490-03	1001
MP2208.x	Inductance of the series reactor Input: * = Input from the motor table active Value of the series reactor in [µH]	340 490-03	1008
MP2209.x	Mass moment of inertia of a drive motor Input: * = Input from the motor table active Value of the mass moment of inertia in [kgm^2]	340 490-03	1138
MP2210.x	Only CC 424(B): Reduction of the nominal voltage (and, as a result, the nominal magnetizing current) at the rpm for field weakening during idle running. Input: 0 to 60 [%] 0 = Function inactive	340 490-01	1097

MP	Function and input	Software version & behavior	Page
MP2220.x	Monitoring functions Format: %xxxxxxxxxxxxxxxx Input: Bit 0 – Monitoring the reference mark 0: Monitoring active 1: Monitoring inactive Bit 1 – Monitoring the direction of rotation 0: Monitoring active 1: Monitoring inactive Bit 2 – Power limit of spindle with $\overline{\text{ERR.IZ.GR}}$ (only for HEIDENHAIN inverters, except UE 2xx) 0: Power limit active 1: Power limit inactive (All HEIDENHAIN inverters except UE 2xx) Bit 3 – Switching off the controller when the motor brakes are activated 0: Suppress oscillations 1: Vibrations are allowed CC 422: Bit 4 to bit 8 reserved Bit 4 – Only CC 424(B): Monitoring for excessive temperature 0: Active 1: Inactive Bit 5 – Only CC 424(B): Monitoring for insufficient temperature 0: Active 1: Inactive Bit 6 – Reserved Bit 7– Only CC 424(B): Monitoring of encoder input frequency 0: Active 1: Inactive Bit 8 – Only CC 424(B): Adjust mechanical offset by gradually increasing the k_V factor 0: Active 1: Inactive Bits 9 to 15: Reserved	PLC RUN	885; 926, 947, 1084
MP2221.x	Bit 7 – Switch-on time of the drive Input: 0: Reduction of the switch-on time is active 1: Reduction of the switch-on time is not active	340 490-04	874
MP2222.x	Reserved	340 490-04	-
MP2223.x	Reserved	340 490-04	-
MP2230.x	Factor for motor standstill current during test of motor brake Input: 0.1 to 30.0 [· motor standstill current] 0: No test of motor brakes, or motor without brake	340 420-08	928
MP2232.x	Maximum permissible path during test of motor brakes Input: 0 to 10.0000 [mm] or [°]	340 420-08	928

MP	Function and input	Software version & behavior	Page
MP2234.x	Internal triggering of the motor brakes via the PWM interface Format: %xx Input: Bit 0 – 0: Signal is transmitted 1: Signal is not transmitted Bit 1– reserved	340 422-06, 340 480-06 PLC RUN	926
MP2250.x	As of CC 424(B): Determining the field angle without motor motion Input: 0: Same as input value 2 1: Reserved 2: Method 2 (brakes applied) 3: Method 3 (same as Method 2, but motor brake is not applied) 4: Method 4 (if there is a lot of noise in the encoder signals)	340 490-04 PLC RUN	1107
MP2252.x	Only CC 424(B): Reserved Input: Enter 0	340 422-03, 340 480-03 PLC RUN	1107
MP2254.x	Determining the field angle Input: 0: Field angle is determined during operation; soft key has no function (without plausibility test) 1: Only CC 422: Field angle is determined via soft key; motor motion is permitted 2: Only CC 424(B): Field angle is determined via soft key; motor motion is permitted (with plausibility test) 3: Only CC 424(B): Same as 2, but the drive must no longer be switched on via the PLC. The drive is moved immediately!	340 420-09 PLC RUN 340 490-01	1017, 1103
MP2256.x	Determined field angle Input: 0: Field angle does not need to be determined, or has not been determined	340 422-03, 340 480-03 PLC RUN	1019, 1113
MP2257.x	Control or encoder identification for the field angle from MP2256.x Input: 0: Field angle does not need to be determined, or has not been determined	340 422-03, 340 480-03 PLC RUN	1019, 1113
MP2260.x	Only CC 424(B): "TRC – Torque Ripple Compensation" File name for the torque-ripple-compensation file Input: xx_<MotorNamefromMotorTable>.TRC (generated in TNCopt) No entry: No compensation	340 490-02 PLC RUN	1095

MP	Function and input	Software version & behavior	Page
MP2261.x	As of CC424(B): Deactivate compensation Bit 0: Torque ripple compensation Bit 1: Gear error compensation Input: %0000000000000000 1: Compensation not active	340 490-05	1096
MP2302.x	Reference value for I^2t monitoring of motor Input: 0 to 1000.000 [· rated current of motor] 0: I^2t monitoring of motor switched off 1: Rated current of motor as reference value	PLC	916
MP2304.x	Reference value for I^2t monitoring of power module Input: 0 to 1000.000 [· rated current of power module] 0: I^2t monitoring of power module switched off 1: Rated current of power module as reference value	340 420-06 PLC	916
MP2308.x	Time between output of the braking signal $\overline{\text{BRK}}$ and switching off of the controller (overlap time) Input: 0.001 to 5.000 [s] 0: 0.200 s	340 420-06	926
MP2309.x	Controller parameters adjusted to closed brake Input: 0: Not active 0.001 to 5.000 [s]	340 490-04	926
MP2312.x	Factor for utilization of motors Input: 0 to 1 000.000 0: Factor = 1		922
MP2390.x	Maximum braking power Input: 0.1 to 3 000.000 [kW] 0: Braking power is not limited		882
MP2392.x	Power limit Input: 0: No power limit 0.1 to 3000.000 [kW]		885
MP2393.x	Only CC424(B): Power limiting as per PLC request Input: 0: No power limit 0.001 to 3000.000 [kW]	340 490-06	885
MP2394.x	Max. braking performance at power failure Input: 0.1 to 3 000.000 [kW] 0: Braking power is not limited		882
MP2396.x	Maximum torque Input: 0.1 to 30 000.0 [Nm] 0: Torque is not limited	PLC	885
MP2420.x	Proportional factor of the current controller Input: 0.00 to 9999.99 [V/A] * = automatic calculation of the P factor	PLC 340 490-03	879

MP	Function and input	Software version & behavior	Page
MP2430.x	Integral factor of the current controller Input: 0.00 to 9 999 999 [Vs/A] * = automatic calculation of the I factor	PLC 340 490-03	879
MP2500.x	Proportional factor of the speed controller Input: 0 to 1 000 000.000 [As]	PLC RUN	859
MP2510.x	Integral factor of the speed controller Input: 0 to 100 000 000 [A]	PLC RUN	859
MP2512.x	Limiting the integral-action component of the speed controller Input: 0.000 to 30.000 [s] (realistic values: 0.1 to 2.0)	PLC RUN	702, 865
MP2520.x	Differential factor of the speed controller Input: 0 to 1.0000 [As]	PLC RUN	861
MP2530.x	PT_2 element of the shaft speed controller (2nd-order delay) Input: 0 to 1.0000 [s]	PLC RUN	862
MP2540.x	Only CC 422: Band-rejection filter damping Input: 0.0 to 18.0 [dB]	PLC RUN	862
MP2542.x	Only CC 424(B): Damping/phase increase for filter 1 Input: 0 to 99.0 [dB]	PLC RUN	1088
MP2543.x	Only CC 424(B): Damping/phase increase for filter 2 Input: 0 to 99.0 [dB]	PLC RUN	1088
MP2544.x	Only CC 424(B): Damping/phase increase for filter 3 Input: 0 to 99.0 [dB]	PLC RUN	1088
MP2545.x	Only CC 424(B): Damping/phase increase for filter 4 Input: 0 to 99.0 [dB]	PLC RUN	1088
MP2546.x	Only CC 424(B): Damping/phase increase for filter 5 Input: 0 to 99.0 [dB]	PLC RUN	1088
MP2550.x	Only CC 422: Band-rejection filter center frequency Input: 0.0 to 999.9 [Hz]	PLC RUN	862
MP2552.x	Only CC 424(B): Center/corner frequency for filter 1 Input: 0 to 30 000.0 [Hz]	PLC RUN	1088
MP2553.x	Only CC 424(B): Center/corner frequency for filter 2 Input: 0 to 30 000.0 [Hz]	PLC RUN	1088
MP2554.x	Only CC 424(B): Center/corner frequency for filter 3 Input: 0 to 30 000.0 [Hz]	PLC RUN	1088
MP2555.x	Only CC 424(B): Center/corner frequency for filter 4 Input: 0 to 30 000.0 [Hz]	PLC RUN	1088
MP2556.x	Only CC 424(B): Center/corner frequency for filter 5 Input: 0 to 30 000.0 [Hz]	PLC RUN	1088

MP	Function and input	Software version & behavior	Page
MP2560.x	Low-pass filter Input: 0: No low-pass filter 1: 1st-order low-pass filter 2: 2nd-order low-pass filter	PLC RUN	861
MP2560.x	Only CC 424(B): Filter order of the low-pass filter Input: 0 to 20	340 420-09 PLC RUN	1090

MP	Function and input	Software version & behavior	Page
MP2562.x	Only CC 424(B): Filter type for filter 1 Input: 0: No filter 1: PT2 low-pass filter (speed controller) 2: Band-rejection filter (speed controller) 3: Phase increase (speed controller) 11: PT2 low-pass filter (position controller) 12: Band-rejection filter (position controller) 13: Phase increase (position controller)	PLC RUN	1088
MP2563.x	Only CC 424(B): Filter type for filter 2 Input: 0: No filter 1: PT2 low-pass filter (speed controller) 2: Band-rejection filter (speed controller) 3: Phase increase (speed controller) 11: PT2 low-pass filter (position controller) 12: Band-rejection filter (position controller) 13: Phase increase (position controller)	PLC RUN	1089
MP2564.x	Only CC 424(B): Filter type for filter 3 Input: 0: No filter 1: PT2 low-pass filter (speed controller) 2: Band-rejection filter (speed controller) 3: Phase increase (speed controller) 11: PT2 low-pass filter (position controller) 12: Band-rejection filter (position controller) 13: Phase increase (position controller)	PLC RUN	1089
MP2565.x	Only CC 424(B): Filter type for filter 4 Input: 0: No filter 1: PT2 low-pass filter (speed controller) 2: Band-rejection filter (speed controller) 3: Phase increase (speed controller) 11: PT2 low-pass filter (position controller) 12: Band-rejection filter (position controller) 13: Phase increase (position controller)	PLC RUN	1089
MP2566.x	Only CC 424(B): Filter type for filter 5 Input: 0: No filter 1: PT2 low-pass filter (speed controller) 2: Band-rejection filter (speed controller) 3: Phase increase (speed controller) 11: PT2 low-pass filter (position controller) 12: Band-rejection filter (position controller) 13: Phase increase (position controller)	PLC RUN	1089
MP2572.x	Only CC 424(B): Bandwidth for filter 1 Input: 0 to 30 000.0 [Hz]	PLC RUN	1089
MP2573.x	Only CC 424(B): Bandwidth for filter 2 Input: 0 to 30 000.0 [Hz]	PLC RUN	1089

MP	Function and input	Software version & behavior	Page
MP2574.x	Only CC 424(B): Bandwidth for filter 3 Input: 0 to 30 000.0 [Hz]	PLC RUN	1089
MP2575.x	Only CC 424(B): Bandwidth for filter 4 Input: 0 to 30 000.0 [Hz]	PLC RUN	1089
MP2576.x	Only CC 424(B): Bandwidth for filter 5 Input: 0 to 30 000.0 [Hz]	PLC RUN	1089
MP2590.x	Braking ramp in an emergency stop Input: 0.1 to 999.9 [min^{-1}/ms] 0: Function inactive	PLC RUN	881
MP2600.x	Acceleration feedforward control Input: 0 to 100.0000 [A/(rev/s^2)]	PLC	865
MP2602.x	IPC time constant T_1 Input: 0.0001 to 1.0000 [s] 0: IPC inactive	PLC RUN	866
MP2604.x	IPC time constant T_2 Input: 0.0001 to 1.0000 [s] 0: IPC inactive	PLC RUN	866
MP2606.x	Jerk feedforward control. Minimizing the following error (due to mechanical deformation) during the jerk phase Input: 0.000 to 10.000	PLC RUN	867
MP2607.x	Damping factor for active damping Input: 0 to 30.000 0: No damping 1.5: Typical damping factor	340 422-03, 340 480-03 PLC RUN	863
MP2608.x	Damping time constant for active damping Input: 0.000 to 0.9999 [s] 0: No damping 0.005 to 0.02: Typical damping time constant	340 422-03, 340 480-03 PLC RUN	863
MP2610.x	Friction compensation at low speeds (effective only with velocity feedforward control) Input: 0 to 100.0000 [A] 0: No friction compensation (or axis is analog)	PLC RUN	703
MP2610.x	Only CC 424(B): Low-speed friction compensation Input: 0 to 30.0000 [A] (effective value) 0: No friction compensation	PLC RUN	1099

MP	Function and input	Software version & behavior	Page
MP2612.x	Delay of the friction compensation (effective only with velocity feedforward control) Input: 0.0000 to 1.0000 [s] (typically: 0.015 s) 0: No friction compensation (or axis is analog)	PLC RUN	703
MP2612.x	Input: 0.000 to 1.000 [mm] or [°] 0: No friction compensation 0.1: Typical input value Only CC 424(B): Distance before the reversal point from which a reduction of the current from MP2610.x is to go into effect.	PLC RUN	1099
MP2614.x	Only CC 424(B): Distance after the reversal point from which a reduction of the current from MP2610.x is to go into effect. Input: 0.000 to 1.000 [mm] or [°] 0: Friction compensation same as CC 424 0.1: Typical input value	PLC RUN	1099
MP2620.x	Friction compensation Input: 0 to 100.000 [A] 0: No friction compensation (or axis is analog)	PLC RUN	703
MP2630.x	Holding current Input: –100.000 to +100.000 [A]	PLC RUN 340 490-03	868
MP2640.x	As of CC424(B): Torsion compensation between position encoder and speed encoder Input: 0.001 to 100.000 [µm/A] 0: Not active	340 490-03	704
MP2900.x	Tensioning torque between master and slave for master-slave torque control (entry for the slave axis) Input: –100.00 to +100.00 [Nm]	PLC	798
MP2910.x	P factor of the torque controller for master-slave torque control (entry for the slave axis) Input: 0.00 to 999.99 [1/(Nm · min)]	PLC	798
MP2920.x	Factor for variable torque distribution for master-slave torque control (entry for the slave axis) Input: 0.000 to 100.000 1: Master and slave axes have identical motors	PLC	798
MP2930.x	Speed compensation ratio for master-slave torque control (entry for the slave axis) Input: –100.00 to +100.00 [%]	PLC	798

4.4.6 Spindle

MP	Function and input	Software version & behavior	Page
MP3010	Output of speed, gear range Input: 0: No output of spindle speed 1: Speed code if the speed changes 2: Speed code at every TOOL CALL 3: Nominal speed value always, G code if the gear range shifts 4: Nominal speed value always, G code at every TOOL CALL 5: Nominal speed value always, no G code 6: Same as 3, but with controlled spindle for orientation 7: Same as 4, but with controlled spindle for orientation 8: Same as 5, but with controlled spindle for orientation	PLC RUN	941
MP3011	Function of analog output S, if MP3010 < 3 Input: 0: No special function 1: Voltage is proportional to the current contouring feed rate, depending on MP3012 2: Voltage is defined as through Module 9130 3: Voltage is defined through M functions (M200 to M204)		1509
MP3012	Feed rate from output of an analog voltage of 10 V, MP3011 = 1 Input: 0 to 300 000 [mm/min]		1509
MP3013.x	Characteristic curve kink points (velocity) for output of the analog voltage with M202 Input: 10 to 300 000 [mm/min]	PLC RUN	1510
MP3014.x	Characteristic curve kink points (voltage) for output of the analog voltage with M202 Input: 0.000 to 9.999 [V]	PLC RUN	1510
MP3020	Speed range for S code output Format: xxyyz xx: S code for minimum speed yy: S code for maximum speed z: Speed increment Input: 0 to 99 999	PLC RUN	961

MP	Function and input	Software version & behavior	Page
MP3030	Behavior of the spindle Input:Bit 0 – 0: Axis stop for TOOL CALL S 1: No axis stop for TOOL CALL S Bit 1: Zero spindle speed when switching to another gear range 0: Reduce speed to 0 1: Do not reduce speed to 0	PLC RUN	957, 1512
MP3120	Zero speed permitted Input: 0: S = 0 allowed 1: S = 0 not allowed	PLC RUN	955
MP3130	Polarity of the nominal spindle speed Input: 0: M03 positive, M04 negative 1: M03 negative, M04 positive 2: M03 and M04 positive 4: M03 and M04 negative	PLC RUN	953
MP3140	Counting direction of spindle position encoder output signals Input: 0: Positive counting direction with M03 1: Negative counting direction with M03	PLC RUN	953
MP3142	Line count of the rotary encoder on the spindle Input: 100 to 100 000 [lines]	PLC RUN 340 490-03	944
MP3143	Mounting configuration of the spindle position encoder Input: 0: Position encoder directly on the first spindle 1: Position encoder via transmission (ratio in MP3450.x and MP3451.x); X30 pin 1: reference pulse 2: Position encoder via transmission (ratio in MP3450 and MP3451); X30 pin 1: reference pulse release 3: Same as input value 1, except that the second reference pulse is evaluated. 4: Reference-mark evaluation of the spindle via EnDat. The encoder must be mounted directly (without transmission). No reference pulse is necessary. A new reference-mark evaluation via marker M4015 may only be performed at standstill. 5: The position encoder is mounted directly (same as input value 0) and the transmission ratio is also evaluated. Entering the value 5 results in the same behavior as that of software version 340 49x-05 (rigid tapping will also be possible without a position encoder on the spindle)!	PLC RUN	944

MP	Function and input	Software version & behavior	Page
MP3210.0-7	Analog nominal spindle voltage at rated speed for the gear ranges 1 to 8 Input: 0 to 100.000 [V] Digital spindle motor revolutions at rated speed for the gear ranges 1 to 8 Input: 0 to 100.000 [1000 min^{-1}]	PLC RUN	955
MP3240.1	Analog spindle: Minimum nominal value voltage Input: 0 to 9.999 [V] Digital spindle: Minimum motor speed Input: 0 to 9.999 [1000 min^{-1}]	PLC RUN	955, 955
MP3240.2	Analog spindle: Spindle jog voltage for gear shifting (M4009/M4010) Input: 0 to 9.999 [V] Digital spindle: Motor speed for gear shifting (M4009/M4010) Input: 0 to 9.999 [1000 min^{-1}]		
MP3310	Limitation for spindle speed override Input: 0 to 150 [%]	PLC RUN	959
MP3310.0	Upper limit		
MP3310.1	Lower limit		
MP3350	Maximum permissible overshoot of the spindle speed in percent of the nominal spindle speed Input: 0 to 100 [%]	340 490-03	951
MP3351	Entry of an absolute value for the permissible overshoot of the spindle speed Input: 0.001 to 100 000.000 [U/min] 0 = Monitoring off	340 490-03	951
MP3411.0-7	Ramp gradient of the spindle with M03 and M04 for gear ranges 1 to 8 Input: Analog axes: 0 to 1.999 [V/ms] Digital axes: 0 to 1.999 [1000 min^{-1}/ms]	PLC RUN	950
MP3412	Multiplication factor for MP3411.x Input: 0.000 to 1.999	PLC RUN	950, 966, 971, 975
MP3412.0	With M05		
MP3412.1	With oriented spindle stop		
MP3412.2	With tapping with floating tap holder		
MP3412.3	With rigid tapping		
MP3415	Overshoot behavior of the spindle with M03, M04 and M05 Input: 0 to 1000 [ms]	PLC RUN	955, 955, 955, 955
MP3415.0	With M03, M04 and M05		
MP3415.1	For oriented spindle stop		
MP3415.2	With tapping		
MP3415.3	With rigid tapping		

MP	Function and input	Software version & behavior	Page
MP3420	Spindle positioning window Input: 0 to 360.0000 [°]	PLC RUN	966
MP3430	Deviation of the reference mark from the desired position (spindle preset) Input: 0 to 360 [°]	PLC RUN	966
MP3440.0-7	k_V factor for spindle orientation for gear ranges 1 to 8 Input: 0.1 to 10 [(1000°/ min) /°]	PLC RUN	966
MP3450.0-7	Number of spindle position-encoder revolutions for gear ranges 1 to 8 Input: 0 to 65 535 0: No transmission	PLC RUN 340 490-05	944
MP3451.0-7	Number of spindle revolutions for gear ranges 1 to 8 Input: 0 to 65 535 0: No transmission	PLC RUN 340 490-05	944
MP3510.0-7	Rated speed for the gear ranges 1 to 8 Input: 0 to 99 999.999 [min^{-1}]	PLC RUN	955
MP3515.0-7	Maximum spindle speed for gear ranges 1 to 8 Input: 0 to 99 999.999 [min^{-1}]	PLC RUN	959
MP3520.0 MP3520.1	Speed activation through marker M4011 Input: 0 to 99 999.999 [min^{-1}] Spindle speed for oriented stop Input: 0 to 99 999.999 [min^{-1}]	PLC RUN	969, 966
MP3530	Increased spindle power for roughing Input: 0 = Not active 1 = Increased spindle power for roughing	340 490-03	953
MP3540	As of CC424(B): Permissible lower spindle speed limit Input: 0.001 to 0.999 0: Monitoring not active	340 490-05	952
MP3542	As of CC 424(B): Minimum spindle speed as of which the monitoring in MP3540 becomes active Input: 0.001 to 0.999 0: Monitoring not active	340 490-05	952
MP3550	As of CC424(B): Delay of EMERGENCY STOP reaction of spindles Input: 0.001 to 0.100 [s] 0: Delay not active	340 490-05	882

4.4.7 Integral PLC

MP	Function and input	Software version & behavior	Page
MP4000.0-63	Options for the conditional compilation of the PLC program		1624
MP4020	PLC functions Format: %xxxxxxxxxxxxxx Input: Bit 0 to bit 4: Reserved Bit 5: Single or double spindle operation 0: Single-spindle operation 1: Double-spindle operation Bit 6 – Reserved Bit 7 – Transferring the values of the Pt 100 inputs 0: Accept values at a change rate of 1 K/s 1: Accept results immediately Bit 8 – Behavior after an ext. emergency stop 0: "Approach position" is not automatically activated 1: "Approach position" is automatically activated Bit 9 – Behavior of a simulated key 0: Simulated key is transferred immediately to the NC 1: Simulated key is processed first by an active PLC window before being transferred to the NC Bit 10 – Behavior of a locked key 0: Locked key only works on the active PLC window 1: Locked key works on neither the active PLC window nor on the NC Bit 11 – PLC counter in MP4120.x 0: Input in PLC cycles 1: Input in seconds Bit 12 – Font size in PLC window 0: Automatic adaptation of font size to screen 1: Font size for BF 120 Bit 13 – Monitoring the housing fan 0: Monitoring active 1: Inactive Bit 14 – Reserved	RESET 340 490-05	855, 981, 1340, 1474
MP4030	Assignment of physical to logical PL Input: 0: First logical PL 1: Second logical PL 2: Third logical PL 3: Fourth logical PL	PLC RUN	1465
MP4030.0	First physical PL		
MP4030.1	Second physical PL		
MP4030.2	Third physical PL		
MP4030.3	Fourth physical PL		

MP	Function and input	Software version & behavior	Page
MP4031	Monitoring of number of PLs Input: -1: Monitoring not active 0 to 4: Number of PLs being monitored	340 490-04	1465
MP4040	Set PLC output after shutdown	340 420-03 PLC RUN	1243
MP4041	Time after shutdown until setting of the PLC output from MP4042 Input: 0 to 1000 [s]	340 420-03 PLC RUN	1243
MP4042	PLC output to be set after shutdown Input: 0 to 31	340 420-03 PLC RUN	1243
MP4043	Delay during shutdown for the PLC to execute final actions Input: 1 to 60 [s] 0: No delay	340 490-04 PLC RUN	1469
MP4044	Switch off outputs that cannot be switched off by emergency stop after 250-ms delay Input: %xxxxxxxx Bits 0 to 7 correspond to O16 to O23 0: Do not switch off output with delay 1: Switch off output with delay	340 422-07, 340 480-07 Only until 340 422-09, 340 480-09 PLC RUN	1469
MP4045	Switch off outputs that cannot be switched off by emergency stop after 250-ms delay Input: % xxxxxxx Bits 0 to 6 correspond to O24 to O30 0: Do not switch off output with delay 1: Switch off output with delay	340 420-08 Only until 340 422-09, 340 480-09 PLC RUN	1469
MP4050.0-8	Traverse distance for lubrication of axes 1 to 9 Input: 0 to 99 999.999 [m or 1000°]	PLC RUN	667
MP4060.0-3	Outputs that are to be switched off with the delay from MP4061.x when all outputs are switched off Input: 0 to 30 [no. of the output] –1: Do not switch off any outputs with delay	340 422-09, 340 480-09 PLC	–
MP4061.0-3	Delay time for switching off the outputs in MP4060.x Input: 0 to 5.000 [s]	340 422-09, 340 480-09 PLC	–
MP4070	Compensation amount per PLC cycle for lagged-tracking axis error compensation Input: 0.0001 to 0.5000 [mm]	PLC RUN	697

MP	Function and input	Software version & behavior	Page
MP4110.0-47	Run time PLC timer T0 to T47 Input: 0 to 1 000 000.000 [s]	PLC RUN	1678
MP4111.96-x	Run time PLC timer T96 to x (defined in OEM.SYS) Input: 0 to 1 000 000.000 [s]	PLC RUN	1678
MP4120.0-47	PLC counter preset value Input: 0 to 1 000 000.000 [s or PLC cycles, depending on MP4020, bit 11]	PLC RUN	1681
MP4130.0 MP4130.1 MP4130.2-5	Number of the high-speed PLC input for switching off the monitoring functions Reserved Numerical designation for fast PLC inputs Input: 0 to 20 000 [no. of the PLC input] -1: Function inactive		898, 1682
MP4131.0 MP4131.1 MP4131.2-5	Activation criterion for fast PLC input for switching off the monitoring functions Reserved Activation criterion for fast PLC inputs Input: 0: Activation at low level 1: Activation at high level		898, 1682
MP4210.0-47	Setting a number in the PLC (D768 to D956) Input: –99 999.9999 to +99 999.9999		969, 1673
MP4220.0-4	Setting a number in the PLC (W960 to W968) Input: 10 to 30 000		1673
MP4230.0-31	Setting a number in the PLC (Module 9032) The number of indexes can be increased via an entry in OEM.SYS. Input: -99 999.9999 to +99 999.9999		1673
MP4231.0-31	Setting a number in the PLC (Module 9032) Input: -99 999.9999 to +99 999.9999		1673
MP4310.0-9	General parameters in the PLC (W976 to W994, M4300 to M4459) Format: Number, $xxxx [Hex], %xxxxxxxxxxxxxxxx [Bin] Input:0 to 65535	340 490-03	1673

4.4.8 Configuration of the Data Interface

MP	Function and input	Software version & behavior	Page
MP5000	Disable data interfaces Input: 0: No interface disabled 1: RS-232-C/V.24 interface disabled 2: RS-422/V.11 interface disabled 3: RS-232-C/V.24 and RS-422/V.11 interfaces disabled	PLC RUN	1816
MP5020 MP5020.0 MP5020.1 MP5020.2 MP5020.3	Configuration of the data interface Format: %xxxxxxxx Input: Bit 0 – 0: 7 data bits 1: 8 data bits Bit 1 – 0 : Any BCC character 1 : BCC not control character Bit 2 – 0: Transmission stop by RTS not active 1: Active Bit 3 – 0: Transmission stop by DC3 not active 1: Active Bit 4 – 0: Character parity even 1: Odd Bit 5 – 0: Character parity not desired 1: Desired Bit 6 = 0, Bit 7 = 0: 1.5 stop bits Bit 6 = 1, Bit 7 = 0: 2 stop bits Bit 6 = 0, Bit 7 = 1: 1 stop bit Bit 6 = 1, Bit 7 = 1: 1 stop bit Bits 8 and 9 – Reserved Bit 10 – Check for missing connection 0: Active 1: Not active Operating mode EXT1 Operating mode EXT2 Operating mode EXT3 (PLC) Operating mode EXT4 (PLC)	PLC RUN CN123 340 490-05	1819

MP	Function and input	Software version & behavior	Page
MP5030	Communications protocol Input: 0 = Standard data transfer protocol 1 = Blockwise transfer 2 = Without protocol (only for MP5030.2)	PLC RUN CN123	1820
MP5030.0	Operating mode EXT1		
MP5030.1	Operating mode EXT2		
MP5030.2	Operating mode EXT3 (PLC)		
MP5030.3	Operating mode EXT4 (PLC)		
MP5040	Data transfer rate in operating mode EXT3 or EXT4 (data transfer through PLC) Input: 0: 110 bps 1: 150 bps 2: 300 bps 3: 600 bps 4: 1200 bps 5: 2400 bps 6: 4800 bps 7: 9600 bps 8: 19200 bps 9: 38400 bps 10: 57600 bps 11: 115200 bps	PLC RUN	1830
MP5040.0	Operating mode EXT3 (PLC)		
MP5040.1	Operating mode EXT4 (PLC)		

4.4.9 3-D Touch Probe

MP	Function and input	Software version & behavior	Page
MP6010	Selection of the touch probe Input: 0: Touch probe with cable transmission (TS 120, TS 220) 1: Touch probe with infrared transmission (TS 632) 2: Touch probe with infrared transmission (TS 440, TS 640) 3: Battery-free TS 444 touch probe	PLC CN123 340 490-04	1481
MP6120	Probing feed rate (triggering touch probe) Input: 1 to 10 000 [mm/min]	PLC RUN CN123 340 490-05	1486
MP6130	Maximum measuring range Input: 0.001 to 99 999.9999 [mm]	PLC RUN CN123	1486
MP6140	Setup clearance above measuring point Input: 0.001 to 99 999.9999 [mm]	PLC RUN CN123	1486
MP6150	Rapid traverse in probing cycle Input: 10 to 20 000 [mm/min]	PLC RUN CN123	1486
MP6151	Pre-positioning in probing cycle with rapid traverse Input: 0: Pre-position with speed from MP6150 1: Pre-positioning at rapid traverse	340 490-02 PLC RUN CN123	1485
MP6160	M function for probing from opposite directions Input: –1: Spindle orientation directly by NC 0: Function inactive 1 to 999: Number of the M function for spindle orientation through PLC	PLC RUN CN123	1490
MP6161	M function for orienting the touch probe before every measuring process Input: –1: Spindle orientation directly by the NC 0: Function inactive 1 to 999: Number of the M function	PLC RUN CN123	1488

MP	Function and input	Software version & behavior	Page
MP6162	Orientation angle Input: 0 to 359.9999 [°]	PLC RUN CN123	1488
MP6163	Minimum difference between the current spindle angle and MP6162 before executing an oriented spindle stop Input: 0 to 3.0000 [°]	PLC RUN CN123	1488
MP6165	Orient the probe before approaching with Cycle 0 or 1, or with manual probing Input: 0: Probe is not oriented before each probing 1: Probe is oriented and always deflected in the same direction	PLC RUN CN123	1486
MP6166	Probing direction of the touch probe with consideration of an active basic rotation (only manual measuring cycles) Input: 0: Inactive 1: Active	PLC RUN CN123	1486
MP6170	Number of measurements in a programmed measurement (touch probe block) Input: 1 to 3	PLC RUN CN123	1491
MP6171	Confidence range for programmed measurement (MP6170 > 1) Input: 0.002 to 0.999 [mm]	PLC RUN CN123	1491
MP6180	Coordinates of the ring gauge center for automatic calibration (Probing Cycle 2) with respect to the machine datum (traverse range 1) Input: 0 to +99 999.9999 [mm]	PLC CN123	1490
MP6180.0	X coordinate		
MP6180.1	Y coordinate		
MP6180.2	Z coordinate		
MP6181	Coordinates of the ring gauge center for automatic calibration (Probing Cycle 2) with respect to the machine datum (traverse range 2) Input: 0 to +99 999.9999 [mm]	PLC CN123	1490
MP6181.0	X coordinate		
MP6181.1	Y coordinate		
MP6181.2	Z coordinate		

MP	Function and input	Software version & behavior	Page
MP6182	Coordinate of the ring gauge center for Probing Cycle 2 with respect to the machine datum (traverse range 3) Input: 0 to +99 999.9999 [mm]	PLC CN123	1491
MP6182.0	X coordinate		
MP6182.1	Y coordinate		
MP6182.2	Z coordinate		
MP6185	Distance of probing point below ring top surface during calibration Input: +0.001 to +99 999.9999 [mm]	PLC CN123	1491

4.4.10 Tool Measurement with TT

MP	Function and input	Software version & behavior	Page
MP6500	Tool measurement with TT 130 Format: %xxxxxxxxxxxxxxx Input: Bit 0 – Cycles for tool measurement 0: Locked 1: Not locked Bit 1 – 0: Tool radius measurement allowed. Tool length measurement with rotating spindle 1: Tool radius measurement and individual tooth measurement disabled Bit 2 – 0: Tool length measurement with rotating spindle (bit 1=1) 1: Tool length measurement with rotating spindle, only if a tool radius offset (**TT: R-OFFS**) has been entered in the tool table Bit 3 – 0: Tool measurement with spindle orientation 1: Tool measurement without spindle orientation. Individual tooth measurement not possible. Tool radius measurement possibly faulty. Bit 4 – 0: Automatically determine speed 1: Always use minimum spindle speed Bit 5 – NC stop during **Tool checking** 0: The NC program is not stopped when the breakage tolerance is exceeded. 1: If the breakage tolerance is exceeded, the NC program is stopped and the error message **Tool broken** is displayed. Bit 6 – NC stop during **tool measurement** 0: The NC program is not stopped when the breakage tolerance is exceeded. 1: If the breakage tolerance is exceeded, the NC program is stopped and the error message **Touch point inaccessible** is displayed.	PLC RUN	1498, 1499, 1501, 1504, 1506, 1506

MP	Function and input	Software version & behavior	Page
MP6500	Tool measurement with TT 130 Format: %xxxxxxxxxxxxxxx Input: Bit 7 – Reserved Bit 8 – Probing routine 0: Probe contact is probed from several directions 1: Probe contact is probed from one direction Bit 9 – Automatic measurement of the direction of the probe contact basic rotation (bit 8 = 1) 0: Basic rotation is not measured 1: Basic rotation of the probe element is automatically measured Bit 10 – Probing routine (bit 8 = 1) 0: Pre-positioning to starting point in all three principal axes 1: Pre-positioning to starting point in the tool axis and in the axis of the probing direction (MP6505) (bit 9 = 0) Bit 11 – **Tool checking** and changing in the tool table 0: After **Tool checking** the tool table is changed 1: After **Tool checking** the tool table is not changed Bit 12 – PLC datum shift 0: Do not include 1: Include Bit 13 – 0: Tool is measured in the tilt position in which the tool touch probe was also calibrated 1: Tool is measured in another tilt position Bit 14 – Tool measurement with number of teeth = 0 0: Tool measurement with rotating spindle 1: Tool measurement with stationary spindle	PLC RUN	
MP6505	Probing direction for tool radius measurement for 3 traverse ranges Input: 0: Positive probing direction of the angle reference axis (0° axis) 1: Positive probing direction in the +90° axis 2: Negative probing direction of the angle reference axis (0° axis) 3: Negative probing direction in the +90° axis	PLC RUN CN123	1500
MP6505.0	Traverse range 1		
MP6505.1	Traverse range 2		
MP6505.2	Traverse range 3		

MP	Function and input	Software version & behavior	Page
MP6507	Calculation of the probing feed rate Input: 0: Calculation of the probing feed rate with constant tolerance 1: Calculation of the probing feed rate with variable tolerance 2: Constant probing feed rate	PLC RUN CN123	1504
MP6510	Permissible measuring error for tool measurement with rotating tool Input: 0.002 to 0.999 [mm]	PLC RUN CN123	1505
MP6510.0	First measurement error		
MP6510.1	Second measurement error		
MP6520	Probing feed rate for tool measurement with non-rotating tool Input: 1 to 10 000 [mm/min]	PLC RUN CN123 340 490-05	1504
MP6530	Distance from the tool end to the top of the probe contact during tool radius measurement for 3 traverse ranges Input: 0.001 to 99.9999 [mm]	PLC RUN CN123	1500
MP6530.0	Traverse range 1		
MP6530.1	Traverse range 2		
MP6530.2	Traverse range 3		
MP6531	Diameter or edge length of the TT 130 probe contact for 3 traverse ranges Input: 0.001 to 99.9999 [mm]	PLC RUN	1502
MP6531.0	Traverse range 1		
MP6531.1	Traverse range 2		
MP6531.2	Traverse range 3		
MP6540	Safety zone around the probe contact of the TT 130 for pre-positioning Input: 0.001 to 99 999.9999 [mm]	PLC RUN CN123	1500
MP6540.0	Safety clearance in tool axis direction		
MP6540.1	Safety clearance in the plane perpendicular to the tool axis		
MP6550	Rapid traverse in probing cycle for TT 130 Input: 10 to 1 000 000 [mm/min]	PLC RUN CN123	1500
MP6560	M function for spindle orientation during individual tooth measurement Input: –1: Spindle orientation directly by NC 0: Function inactive 1 to 999: Number of the M function for spindle orientation by the PLC	PLC RUN CN123	1499

MP	Function and input	Software version & behavior	Page
MP6562	M function before and after tool measurement cycle (TT cycle) Input: -1: Function inactive 0 to 999: Number of the M function	349 490-06 PLC RUN CN123	1507
MP6562.0	M function before the cycle start		
MP6562.1	M function after cycle end		
MP6570	Max. permissible surface cutting speed at the tooth edge Input: 1.0000 to 129.0000 [m/min]	PLC RUN CN123	1504
MP6572	Maximum permissible speed during tool measurement Input: 1 to 1000 [min^{-1}] 0: 1000 [min^{-1}]	PLC RUN CN123	1504
MP6580.0-2	Coordinates of the TT 130 probe contact center with respect to the machine datum (traverse range 1) Input: –99 999.9999 to +99 999.9999 [mm]	PLC RUN CN123	1502
MP6581.0-2	Coordinates of the TT 130 probe contact center with respect to the machine datum (traverse range 2) Input: –99 999.9999 to +99 999.9999 [mm]	PLC RUN CN123	1502
MP6582.0-2	Coordinates of the TT 130 probe contact center with respect to the machine datum (traverse range 3) Input: –99 999.9999 to +99 999.9999 [mm]	PLC RUN CN123	1502
MP6585	Monitoring the position of the rotary and additional linear axes during the tool measurement cycles Format: %xxxxxx Input: 0: Axis is not monitored 1: Axis is monitored Bit 0 – A axis Bit 1 – B axis Bit 2 – C axis Bit 3 – U axis Bit 4 – V axis Bit 5 – W axis	PLC RUN CN123	1505
MP6586	Ref. coordinate for monitoring the position of the rotary and additional linear axes during the tool measurement cycles Input: –99 999.9999 to +99 999.9999 [mm or °]	PLC RUN CN123	1505
MP6586.0	A axis		
MP6586.1	B axis		
MP6586.2	C axis		
MP6586.3	U axis		
MP6586.4	V axis		
MP6586.5	W axis		

MP	Function and input	Software version & behavior	Page
MP6600	KinematicsOpt: Maximum permitted change value Input: 0.010 to 1.000 [mm]	340 490-04	760
MP6601	KinematicsOpt: Radius deviation of the calibration sphere Input: 0.010 to 0.100 [mm]	340 490-04	760
MP6602	KinematicsOpt: M-function macro for positioning the rotary axes Input: 0 to 999 -1: Function inactive	340 490-06	761

4.4.11 Tapping

MP	Function and input	Software version & behavior	Page
MP7110.0	Minimum for feed-rate override during tapping Input: 0 to 150 [%]	PLC RUN	971
MP7110.1	Maximum for feed-rate override during tapping Input: 0 to 150 [%]		
MP7120.0	Dwell time for reversal of spindle rotational direction Input: 0 to 65.535 [s]	PLC RUN	971, 971
MP7120.1	Advanced switching time of the spindle during tapping with coded spindle-speed output Input: 0 to 65.535 [s]		
MP7120.2	Spindle slow-down time after reaching the hole depth Input: 0 to 65.535 [s]		
MP7130	Run-in behavior of the spindle during rigid tapping Input: 0.001 to 10 [°/min]	PLC RUN	975
MP7150	Positioning window of the tool axis during rigid tapping Input: 0.0001 to 2 [mm]	PLC RUN	975
MP7160	Spindle response during Cycles 17, 207 and 18 Format: %xxxxx Input: Bit 0 – Oriented spindle stop with Cycles 17 and 207 0: Oriented spindle stop before execution of the cycle 1: No oriented spindle stop before execution of the cycle Bit 1 – Spindle speed 0: Spindle speed is not limited 1: Spindle speed is limited so that it runs with constant speed approx. 1/3 of the time Bit 2 – Spindle in position feedback control 0: Spindle operated without position feedback control 1: Spindle operated with position feedback control Bit 3 – Acceleration feedforward control 0: Active 1: Not active Bit 4 – 0: Tool axis tracks the spindle 1: Tool axis and spindle interpolated	PLC RUN CN123	975

4.4.12 Display and Operation

MP	Function and input	Software version & behavior	Page
MP7210	Programming station Input: 0: Controlling and programming 1: Programming station with PLC active 2: Programming station with PLC inactive 3: Programming station with PLC and emergency stop active	CN123 340 49x-02	1248
MP7212	Power interrupted message Input: 0: Acknowledge message **Power interrupted** with CE key 1: **Power Interrupted** message does not appear	PLC RUN CN123	1244
MP7220	Block number increment for DIN/ISO programs Input: 0 to 250	PLC RUN CN123	1197
MP7224	Disable file types Input: 0: Do not disable 1: Disable Bit 0 – HEIDENHAIN programs *.H Bit 1 – DIN/ISO programs *.I Bit 2 – Tool tables *.T Bit 3 – Datum tables *.D Bit 4 – Pallet tables *.P Bit 5 – Text files *.A Bit 6 – HELP files *.HLP Bit 7 – Point tables *.PNT	PLC RUN CN123	1355
MP7224.0	Disable soft keys for file types		
MP7224.1	Protecting file types		
MP7224.2	Disable the EDIT ON/OFF soft key	340 422-07, 340 480-07	
MP7225	Disable Windows drives in the TNC file manager Format: ABCDEFGHIJKLMNOPQRSTUVWXYZ Input: If there are more than one drive, they are entered without spaces, e.g. MP7225 = CDE	340 480-06 PLC RUN	1847
MP7226.0	Reserved	PLC	
MP7226.1	Size of the datum table Input: 0 to 255 [lines]	RUN CN123	1358

MP	Function and input	Software version & behavior	Page
MP7229	Properties of the NC program	PLC	1199
MP7229.0	Line number for program testing	RUN	
	Input: 100 to 9999	CN123	
MP7229.1	Program length up to which FK blocks are permitted		
	Input: 100 to 9999		
MP7230.x	Switching the conversational language	PLC	1263
	Input: 0: English 1: German 2: Czech 3: French 4: Italian 5: Spanish 6: Portuguese 7: Swedish 8: Danish 9: Finnish 10: Dutch 11: Polish 12: Hungarian 13: Reserved 14: Russian (Cyrillic characters) 15: Chinese (simplified) 16: Chinese (traditional) 17: Slovenian (option #41) 18: Norwegian (option #41) 19: Slovak (option #41) 20: Latvian (option #41) 21: Korean (option #41) 22: Estonian (option #41) 23: Turkish (option #41) 24: Romanian (option #41) 14, 15, 16 and 17 only in connection with BF 150	RUN CN123 340 490-04	
MP7230.0	NC conversational language, soft keys for OEM cycles		
MP7230.1	PLC conversational language (user parameters)		
MP7230.2	PLC error messages		
MP7230.3	Help files		
MP7235	Time difference to Universal Time (Greenwich Mean Time) Input: –23 to +23 [hours] MP eliminated as of 340 490-03	No longer used as of 340 480-03, 340 49x-03 CN123 RESET	1163

MP	Function and input	Software version & behavior	Page
MP7237 MP7237.0 MP7237.1 MP7237.2	Display and reset the operating times Display PLC operating times Input: Bits 0 to 12 correspond to PLC operating times 1 to 13 0: Do not display 1: Display Reset PLC operating times with code number 857282 Input: Bits 0 to 12 represent PLC operating times 1 to 13 0: Do not reset 1: Reset Reset NC operating times with code number 857282 Input: Bit 0 – No function Bit 1 – "Machine on" operating time Bit 2 – "Program run" operating time 0: Do not reset 1: Reset	PLC RUN	1160
MP7238.0-12	Dialog messages for PLC operating times 1 to 13 Input: 0 to 4095 Dialog no. from the file (OEM.SYS)	PLC RUN	1160
MP7245	Disable auxiliary cycles Input: 0: Auxiliary cycles disabled 1: Auxiliary cycles permitted	PLC RUN	1240
MP7246	Machine parameter with multiple function Input: %xxxx Bit 0 – Paraxial positioning blocks 0: Permitted 1: Disabled Bit 1 – Clear with DEL key 0: Does not need confirmation 1: Must confirm via soft key Bit 2 – Tool usage file 0: Do not generate 1: Generate Bit 3 – Settings file for AFC 0: Do not generate 1: Generate Bit 4 – ASCII file for machining time per NC block 0: Do not create ASCII file for machining time per NC block 1: Create ASCII file for machining time per NC block	PLC RUN 340 490-04	1199, 1246, 1543
MP7251	Number of global Q parameters starting from Q99 (up to Q60) that are transferred from the OEM cycle to the calling program. Input: 0 to 40	PLC RUN	1649

MP	Function and input	Software version & behavior	Page
MP7260	Number of tools in the tool table MP7260 can also be overwritten by the PLC and the LSV2 protocol. Input: 0 to 30 000	CN123 340 490-03	1515
MP7261.0-7	Number of pockets in the tool magazine 1 to 8 MP7261 can also be overwritten by the PLC and the LSV2 protocol. Input: 0 to 9999	CN123 340 490-05	1515
MP7262	Maximum tool index number for indexed tools MP7262 can also be overwritten by the PLC and the LSV2 protocol. Input: 0 to 9	SZ123 340 490-03	1542
MP7263	Pocket table MP7263 can also be overwritten by the PLC and the LSV2 protocol. Format: %xxxxxxx Input: Bit 0 – 0: Show POCKET TABLE soft key 1: Hide POCKET TABLE soft key Bit 1 – Output of the columns for file functions 0: Output only the displayed columns 1: Output all columns Bit 2 – Show the "Edit ON/OFF" soft key in the pocket table 0: Display soft key 1: Do not display soft key Bit 3 – Soft keys "Reset pocket table" and "Reset column T" 0: Display soft keys 1: Do not display soft keys Bit 4 - Deletion possible for a tool that is in the pocket table. Deletion must be confirmed. 0: Deletion impossible 1: Deletion possible (with confirmation) Bit 5 – Deletion of a tool possible even without confirmation (if bit #4 = 1) 0: Deletion not possible without confirmation 1: Deletion possible without confirmation Bit 6 – Deletion of index entries of a tool behaves like deletion of a tool. The settings of bit#4 and #5 also apply to the index entries if bit#6 is set. 0: Deletion always impossible 1: Deletion possible depending on settings in bits #4 and #5	CN123 340 490-05 340 490-06	1515

MP	Function and input	Software version & behavior	Page
MP7266	Elements of the tool table MP7266 can also be overwritten by the PLC and the LSV2 protocol. Input: 0: No display 1 to 99: Position in the tool table	CN123 340 490-03	1515
MP7266.0	16-character alphanumeric tool name (NAME)		
MP7266.1	Tool length (L)		
MP7266.2	Tool radius (R)		
MP7266.3	Tool radius 2 for toroidal cutter (R2)		
MP7266.4	Oversize in tool length (DL)		
MP7266.5	Oversize in tool radius (DR)		
MP7266.6	Oversize in tool radius 2 (DR2)		
MP7266.7	Locked tool? (TL)		
MP7266.8	Replacement tool (RT)		
MP7266.9	Maximum tool age, M4543 (TIME1)		
MP7266.10	Maximum tool age TOOL CALL (TIME2)		
MP7266.11	Current tool age (CUR.TIME)		
MP7266.12	Comment on the tool (DOC)		
MP7266.13	Number of tool teeth (CUT)		
MP7266.14	Wear tolerance for tool length (LTOL)		
MP7266.15	Wear tolerance for tool radius (RTOL)		
MP7266.16	Cutting direction of the tool (DIRECT)		
MP7266.17	Additional information for PLC, module 9093 (PLC)		
MP7266.18	Tool offset for tool length (TT:LOFFS)		
MP7266.19	Tool offset for tool radius (TT: ROFFS)		
MP7266.20	Breakage tolerance for tool length (LBREAK)		
MP7266.21	Breakage tolerance for tool radius (RBREAK)		
MP7266.22	Tooth length (LCUTS)		
MP7266.23	Plunge angle (ANGLE)		
MP7266.24	Tool type (TYP)		
MP7266.25	Tool material (TMA)		
MP7266.26	Cutting-data table (CDT)		
MP7266.27	PLC value (PLC-VAL)		
MP7266.28	Probe center offset in reference axis (CAL-OF1)		
MP7266.29	Probe center offset in minor axis (CAL-OF2)		
MP7266.30	Spindle angle during calibration (CAL-ANG)		
MP7266.31	Tool type for pocket table (PTYP)	340 420-02	
MP7266.32	Maximum shaft speed [1/min] (NMAX)	340 422-03, 340 480-03	
MP7266.33	Retract tool (LIFTOFF)	340 422-06, 340 480-06	

MP	Function and input	Software version & behavior	Page
MP7266.34	PLC value (P1)	340 490-01	
MP7266.35	PLC value (P2)	340 490-01	
MP7266.36	PLC value (P3)	340 490-01	
MP7266.37	Additional kinematics description (KINEMATIC)	340 490-01	
MP7266.38	Point angle for DRILL and CSINK (T-ANGLE)	340 490-01	
MP7266.39	Thread pitch for TAP (PITCH)	340 490-01	
MP7266.40	Control strategy name for AFC (Adaptive Feed Control)	340 490-03	
MP7266.41	Tool value or tool radius R2 (R2TOL)	340 490-05	
MP7266.42	Compensation value table for 3DToolComp (DR2TABLE)	340 490-06	
MP7266.43	Time stamp during tool changing (LAST_USE)	340 490-06	
MP7267	Elements of the pocket table MP7267 can also be overwritten by the PLC and the LSV2 protocol. Input: 0: No display 1 to 99: Position in the pocket table	CN123 340 490-03	1519
MP7267.0	Tool number (T)		
MP7267.1	Special tool (ST)		
MP7267.2	Fixed pocket (F)		
MP7267.3	Locked pocket (L)		
MP7267.4	PLC status (PLC)		
MP7267.5	Tool name (TNAME)		
MP7267.6	Comment on the tool (DOC)		
MP7267.7	Tool type for pocket table (PTYP)	340 420-02	
MP7267.8	Value 1 (P1)		
MP7267.9	Value 2 (P2)		
MP7267.10	Value 3 (P3)		
MP7267.11	Value 4 (P4)		
MP7267.12	Value 5 (P5)		
MP7267.13	Reserve pocket (RSV)		
MP7267.14	Pocket above locked (LOCKED_ABOVE)		
MP7267.15	Pocket below locked (LOCKED_BELOW)		
MP7267.16	Pocket at left locked (LOCKED_LEFT)		
MP7267.17	Pocket at right locked (LOCKED_RIGHT)		
MP7267.18	S1 value (P6)	340 490-05	
MP7267.19	S2 value (P7)	340 490-05	

MP	Function and input	Software version & behavior	Page
MP7270	Feed rate display in the operating modes MANUAL OPERATION and ELECTRONIC HANDWHEEL Input: 0: Display of axis feed rate through pressing an axis direction key (axis-specific feed rate from MP1020) 1: Display of axis feed rate also before an axis direction key is pressed (smallest value from MP1020 for all axes)	PLC RUN CN123	1147
MP7280	Decimal character Input: 0: Decimal comma 1: Decimal point	PLC RUN CN123	1264
MP7281	Depiction of the NC program Input: 0: All blocks completely 1: Current block completely, others line by line 2: All blocks line by line; complete block when editing	PLC RUN CN123	1199
MP7285	Tool length offset in the tool-axis position display Input: 0: Tool length is not offset 1: Tool length is offset	PLC RUN CN123	1140
MP7289	Position display step for the spindle Input: 0: 0.1° 1: 0.05° 2: 0.01° 3: 0.005° 4: 0.001° 5: 0.0005° 6: 0.0001°	PLC RUN CN123	1140
MP7290.0-8	Position display step for axes 1 to 9 Input 0: 0.1 mm or 0.1° 1: 0.05 mm or 0.05° 2: 0.01 mm or 0.01° 3: 0.005 mm or 0.005° 4: 0.001 mm or 0.001° 5: 0.0005 mm or 0.0005° 6: 0.0001 mm or 0.0001°	PLC RUN CN123	1140
MP7291	Display of axes on the screen Format: SXYZABCUVWxyzabcuvw- Input: Characters 1 to 9 from the right represent lines 1 to 9 Character 10 is spindle S which is always output in line 10.	PLC RUN	645
MP7291.0	Display in traverse range 1		
MP7291.1	Display in traverse range 2		
MP7291.2	Display in traverse range 3		

MP	Function and input	Software version & behavior	Page
MP7294	Disable axis-specific "Datum setting" in the preset table Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Not disabled 1: Disabled	340 422-01, 340 480-02 PLC RUN CN123	1196
MP7295	Disable "Datum setting" Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Not disabled 1: Disabled	PLC RUN CN123	1195
MP7296	"Datum setting" through axis keys Input: 0: Datum can be set by axis keys and soft key 1: Datum can be set only by soft key	PLC RUN CN123	1195
MP7300	Erasing the status information, tool data and Q parameters Input: 0: Erase the status information, Q parameters and tool data when a program is selected. 1: Erase the status information, Q parameters and tool data if a program is selected and in the event of M02, M30, and END PGM. 2: Erase the status information and tool data if a program is selected. 3: Erase the status information and tool data if a program is selected and in the event of M02, M30, END PGM. 4: Erase the status information and Q parameters if a program is selected. 5: Erase the status information and Q parameters if a program is selected and in the event of M02, M30, END PGM. 6: Erase the status information if a program is selected and in the event of M02, M30, END PGM. 7: Erase the status information if a program is selected and in the event of M02, M30, END PGM.	PLC RUN CN123	1151

MP	Function and input	Software version & behavior	Page
MP7310	Graphic display mode Format: %xxxxxxxx Input: Bit 0 – Projection in three planes: 0: German-preferred projection 1: US-preferred projection Bit 1 – Rotating the coordinate system in the working plane by 90°: 0: No rotation 1: Rotation by +90° Bit 2 – BLK form after datum shift: 0: Shifted 1: Not shifted Bit 3 – Display of the cursor position: 0: Not displayed 1: Displayed Bit 4 – Reserved Bit 5 – 3-D graphics during program test 0: 2.5-D and 3-D (only with MC 420 or MC 422B and higher) 1: 2.5 D Bit 6 – Stock removal with an inclined tool 0: Not active 1: Active Bit 7 – Exact evaluation of the column LCUTS (cutting length) from the TOOL.T table in order to display special tools (e.g. saw blade). 0: Free evaluation 1: Exact evaluation for special tools	PLC RUN CN123 340 490-06	1255
MP7312	Limitation of the tool tooth length LCUTS if no value was given for the tooth length in the tool table Input: = 0: No limitation, infinitely long tooth length > 0: Tooth length = 2 * tool radius * MP7312	PLC RUN	1511
MP7315	Tool radius for graphic simulation without TOOL CALL Input: 0.0000 to 99 999.9999 [mm]	PLC RUN CN123	1511
MP7316	Penetration depth of the tool Input: 0.0000 to 99 999.9999 [mm]	PLC RUN CN123	1511

MP	Function and input	Software version & behavior	Page
MP7317 MP7317.0 MP7317.1	M function for graphic simulation Beginning of graphic simulation Input: 0 to 88 Interruption of the graphic simulation Input: 0 to 88	PLC RUN CN123	1511
MP7330.0-15	Specification of user parameters 1 to 16 Input: 0 to 9999.00 (no. of the user parameter)	PLC RUN	1246
MP7340.0-15	Dialog messages for user parameters 1 to 16 Input: 0 to 4095 (line number of the PLC dialog message file)	PLC RUN	1246

4.4.13 Colors

MP	Function and input	Software version & behavior	Page
MP7350	Window frames	PLC RUN	1249
MP7351 MP7351.0 MP7351.1 MP7351.2	Error messages Priority 0 (error) Priority 1 (warning) Priority 2 (information)	PLC RUN 340 422-06, 340 480-06	1249
MP7352 MP7352.0 MP7352.1 MP7352.2	"Machine" operating mode display Background Text for operating mode Dialog	PLC RUN	1249
MP7353 MP7353.0 MP7353.1 MP7353.2	"Programming" operating mode display Background Text for operating mode Dialog	PLC RUN	1249
MP7354 MP7354.0 MP7354.1 MP7354.2 MP7354.3 MP7354.4	"Machine" program text display Background General program text Active block Color of the comments and unused machine parameters in the machine parameter file Background of inactive window	PLC RUN 340490-05	1249
MP7355 MP7355.0 MP7355.1 MP7355.2 MP7355.3 MP7355.4	"Programming" program text display Background General program text Active block Color of the comments and unused machine parameters in the machine parameter file Background of inactive window	PLC RUN 340490-05	1249
MP7356 MP7356.0 MP7356.1 MP7356.2	Status window and PLC window Background Axis positions in the status display Status display other than axis positions	PLC RUN	1250
MP7357 MP7357.0 MP7357.1 MP7357.2 MP7357.3	"Machine" soft-key display Background Text color Inactive soft-key row Active soft-key row	PLC RUN	1250

MP	Function and input	Software version & behavior	Page
MP7358	"Programming" soft-key display	PLC	1250
MP7358.0	Background	RUN	
MP7358.1	Text color		
MP7358.2	Inactive soft-key row		
MP7358.3	Active soft-key row		
MP7360	Graphics: 3-D view and plan view	PLC	1250
MP7360.0	Background	RUN	
MP7360.1	Surface		
MP7360.2	3-D: Front face		
MP7360.3	Text display in the graphics window		
MP7360.4	3-D: Lateral face		
MP7360.5	Lowest point of blank form		
MP7360.6	Highest point of blank form (below surface)		
MP7361	Graphics: Projection in three planes	PLC	1250
MP7361.0	Background	RUN	
MP7361.1	Top view		
MP7361.2	Front and side view		
MP7361.3	Axis cross and text in the graphic display		
MP7361.4	Cursor		
MP7362	Additional status display in the graphics window	PLC	1250
MP7362.0	Background of graphic window	RUN	
MP7362.1	Background of status display	340 490-04	
MP7362.2	Status symbols		
MP7362.3	Status values		
MP7362.4	Color of the unselected tabs in the graphics window		
MP7362.5	AFC tab – Background color		
MP7362.6	AFC tab – Color of actual override factor		
MP7362.7	AFC tab – Color of actual spindle factor		
MP7363	Programming graphics	PLC	1250
MP7363.0	Background	RUN	
MP7363.1	Resolved contour		
MP7363.2	Subprograms and frame for zooming		
MP7363.3	Alternative solutions		
MP7363.4	Unresolved contour		
MP7363.5	Rapid traverse movements		
MP7364	Color of the help illustrations for cycles	PLC	1251
MP7364.0-6	Colors 1 to 7 of the graphic program used	RUN	
MP7364.7	Line color (color 8 of the graphic program)		
MP7364.8	Color for highlighted graphic elements if defined in the help illustration		
MP7364.9	Background		

MP	**Function and input**	**Software version & behavior**	**Page**
MP7365 MP7365.0 MP7365.1 MP7365.2 MP7365.3 MP7365.4-9	Oscilloscope Background Grid Cursor and text Selected channel Channel 1 to 6	PLC RUN 340 420-02	1251
MP7366 MP7366.0 MP7366.1 MP7366.2 MP7366.3 MP7366.4 MP7366.5 MP7366.6-14	Pop-up window (HELP key, pop-up menus etc.) Background Text or foreground Active line Title bar Scroll-bar field Scroll bar Reserved	PLC RUN	1251
MP7367 MP7367.0 MP7367.1-7 MP7367.8-14	Large PLC window Background Colors 1 to 7 (Color 8: MP7350) Colors 9 to 15	PLC RUN	1251
MP7368 MP7368.0 MP7368.1 MP7368.2 MP7368.3	Calculator Background Background of displays and keys Key texts ("os" in "cos") Key symbols	PLC RUN	1251
MP7369 MP7369.0 MP7369.1 MP7369.2 MP7369.3 MP7369.4 MP7369.5 MP7369.6	Directory tree in PGM MGT Text background Text Text background of the active folder Line color of the tree structure Folders Drives Text background of the heading in the browser window	PLC RUN	1251
MP7370 MP7370.0 MP7370.1-15	Small PLC Window Background Colors 1 to 15	340 420-05 PLC RUN	1251

MP	Function and input	Software version & behavior	Page
MP7392	Settings for screensaver	PLC	1251
MP7392.0	Time after which the screensaver is activated Input: 1 to 99 [min] 0: No screensaver	RUN CN123	
MP7392.1	Type of screensaver Input: 0: No screensaver 1: Default screensaver of the X server 2: 3-D line graphics	340 490-03	

4.4.14 Machining and Program Run

MP	Function and input	Software version & behavior	Page
MP7400	Look-ahead – Number of NC blocks for advance calculation of the path Input: 0: 256 [blocks] (default) 1: 512 [blocks] 2: 1024 [blocks]	340 490-02 PLC RUN	1197
MP7410	Scaling cycle in two or three axes Input: 0: Scaling cycle is effective in all three principal axes 1: Scaling cycle is effective only in the working plane	PLC RUN CN123	1224
MP7411	Tool data in the touch probe block Input: Bit 0 – 0: Use the calibrated data of the touch probe 1: Use the current tool data from the last TOOL CALL Bit 1 – 0: Only one set of touch probe calibration data 1: Use the tool table to manage more than one set of touch probe calibration data; display the tool name and tool number	PLC RUN CN123	1487
MP7420	Cycles for milling pockets with combined contours Format: %xxxxx Input: Bit 0 – Milling direction for channel milling: 0: Counterclockwise for pockets, clockwise for islands 1: Clockwise for pockets, counterclockwise for islands Bit 1 – Sequence for rough-out and channel milling (only for SL 1): 0: First channel milling, then pocket rough-out 1: First pocket rough-out, then channel milling Bit 2 – Merging of listed contours: 0: Contours are merged only if the tool-center paths intersect 1: Contours are merged if the programmed contours intersect Bit 3 – Rough-out and channel milling to pocket depth or for every infeed 0: Each process uninterrupted to pocket depth 1: Both processes for each pecking depth before proceeding to the next depth Bit 4 – Position after completion of the cycle: 0: Tool moves to the same position as before the cycle was called 1: Tool only moves in the tool axis to the "clearance height"	PLC RUN CN123	1224

MP	Function and input	Software version & behavior	Page
MP7430	Overlap factor for pocket milling Input: 0.001 to 1.414	PLC RUN CN123	1222
MP7431	Arc end-point tolerance Input: 0.0001 to 0.016 [mm]	PLC RUN CN123	1245
MP7432	Limit-switch tolerance for M140 / M150 Input: 0.0001 to 1.0000 [mm] 0: Limit-switch tolerance not active	340 490-03	1245
MP7440	Output of M functions Format: %xxxxxxx Input: Bit 0 – Program stop with M06 0: Program stop with M06 1: No program stop with M06 Bit 1 – Modal cycle call M89 0: Normal code transfer of M89 at beginning of block 1: Modal cycle call M89 at end of block Bit 2 – Program stop with M functions: 0: Program stop until acknowledgment of the M function 1: No program stop: No waiting for acknowledgment. Bit 3 – Switching of k_V factors with M105/M106: 0: Function is not in effect 1: Function is effective Bit 4 – Reduced feed rate in the tool axis with M103: 0: Function is not in effect 1: Function is effective Bit 5 – Reserved Bit 6 – Automatic activation of M134 0: M134 must be activated in the NC program 1: M134 is automatically activated when an NC program is selected	PLC RUN CN123	700, 781, 845, 847, 1239, 1512
MP7441	Error message during cycle call Format: %xxx Input: Bit 0 – 0: Error message **Spindle ?** is not suppressed 1: Error message **Spindle ?** is suppressed Bit 1: Reserved, enter 0 Bit 2 – 0: Error message **Enter depth as negative** is suppressed 1: Error message **Enter depth as negative** is not suppressed	PLC RUN CN123	1239

MP	Function and input	Software version & behavior	Page
MP7442	Number of the M function for spindle orientation in the cycles Input: 1 to 999: Number of the M function 0: No oriented spindle stop –1: Oriented spindle stop by the NC	PLC RUN CN123	964
MP7444	Delay time for the change signals (M/S/T) Input: 0, 1, 2 0: Change signal after complete filter run time (previous behavior) 1 = Change signal if the nominal feed rate reached the value 0 before the actual value (formed by the filters) did. 2 = Change signal if the nominal feed rate reached the value 0 before the actual value (formed by the filters) did, and an additional delay until the actual value reaches the time window the first time.	340 490-05	1230
MP7450	Offsetting the tool change position from MP951.x in block scan Format: %xxxxxxxxxxxxxx Input: Bits 0 to 3 represent axes 1 to 14: 0: Do not offset 1: Offset	PLC RUN	1227
MP7451.0-8	Feed rate for returning to the contour for axes 1 to 9 Input: 10 to 1 000 000 [mm/min]	PLC RUN	1227
MP7460.x	Reserved	340 422-10, 340 480-10 PLC RUN CN123	–
MP7461.x	Reserved	340 422-10, 340 480-10 PLC RUN CN123	–
MP7470	Maximum contouring tool feed rate at 100% override Input: 0 to 300 000 [mm/min] 0: No limitation	PLC RUN CN123	–
MP7471	Maximum velocity of the principal axes during compensating movements through M128 or TCPM Input: 0 to 300 000 [mm/min]	PLC RUN CN123	781

MP	Function and input	Software version & behavior	Page
MP7475	Reference for datum table Input: 0: Reference is workpiece datum 1: Reference is machine datum (MP960.x)	PLC RUN CN123	1358
MP7480 MP7480.0	Output of the tool or pocket number With TOOL CALL block Input: 0: No output 1: Tool number output only when tool number changes 2: Output of tool number for every TOOL CALL block 3: Output of the pocket number and tool number only when tool number changes 4: Output of pocket number and tool number for every TOOL CALL block 5: Output of the pocket number and tool number only when tool number changes. Pocket table is not changed. 6: Output of pocket number and tool number for every TOOL CALL block. Pocket table is not changed.	PLC RUN	1555
MP7480.1	With TOOL DEF block Input: 0: No output 1: Tool number output only when tool number changes 2: Output of tool number for every TOOL DEF block 3: Pocket number and tool number output only when tool number changes 4: Output of pocket number and tool number for every TOOL DEF block		

MP	Function and input	Software version & behavior	Page
MP7481.x	Sequence for new and returned tool when changing tools Format: %xxxxxxxx 0: First, output the pocket of the tool to be returned 1: First, output the pocket of the new tool Input:Bit 0: New tool from magazine 1 Bit 1: New tool from magazine 2 Bit 2: New tool from magazine 3 Bit 3: New tool from magazine 4 Bit 4: New tool from magazine 5 Bit 5: New tool from magazine 6 Bit 6: New tool from magazine 7 Bit 7: New tool from magazine 8	340 490-05 PLC RUN	1558
MP7481.0	Tool from magazine 1 to be returned		
MP7481.1	Tool from magazine 2 to be returned		
MP7481.2	Tool from magazine 3 to be returned		
MP7481.3	Tool from magazine 4 to be returned		
MP7481.4	Tool from magazine 5 to be returned		
MP7481.5	Tool from magazine 6 to be returned		
MP7481.6	Tool from magazine 7 to be returned		
MP7481.7	Tool from magazine 8 to be returned		
MP7482	Pocket coding of the tool magazine Format: %xxxx 0: Variable pocket coding 1: Fixed pocket coding Input: Bit 0: Magazine 1 Bit 1: Magazine 2 Bit 2: Magazine 3 Bit 3: Magazine 4	340 420-06 PLC RUN	1549
MP7483	Tool name/number for TOOL CALL / TOOL DEF Input: 0: Names and numbers are permitted (as before) 1: Only names are permitted 2: Only numbers are permitted	340 490-03	1556
MP7484	Search sequence in tool magazines Input: 0 to 7 [index from MP7261] –1: Cancel	340 490-05	1537
MP7485	Add usage time for tool selection Input: 0 to 100 [%] Default setting: 10	340 490-05	1543

MP	Function and input	Software version & behavior	Page
MP7490	Functions for traverse ranges Format: %xxxx Input: Bit 0 – 0: Display one traverse range with MOD 1: Display three traverse ranges with MOD Bit 1 – 0: Each traverse range has its own datum (and 3 memories for the positions of the swivel head) 1: One datum for all traverse ranges Bit 2 – Calibration data: touch probe for workpiece measurement: 0: One set of calibration data for all traverse ranges 1: Every traverse range has its own set of calibration data Bit 3 – Calibration data: touch probe for tool measurement: 0: One set of calibration data for all traverse ranges 1: Every traverse range has its own set of calibration data	PLC RUN	666, 1487, 1498
MP7492.x MP7492.0 MP7492.13	Number of axis in which the same datum is to be set during Datum Setting (with active preset table) Input: 0 to 17 –1: Do not set a datum Datum set in the first axis to Datum set in the 14th axis	340 422-03, 340 480-03 PLC RUN	1195
MP7493	Maximum deviation of the current tool orientation relative to the tool axis when setting a reference point with M114 Input: 0.0000 to 30.0000 [degrees] Default: 0.005	340 490-02 PLC RUN	774
MP7494	Axes for which an exact stop is to occur after positioning Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: No exact stop 1: Exact stop	340 422-06, 340 480-06 PLC RUN	–

MP	Function and input	Software version & behavior	Page
MP7500 (is set via the kinematics table)	Tilt working plane (inactive preset table) Format: %xxxxxxxxx Input:Bit 0 – "Tilted working plane" 0: Off 1: On Bit 1 – 0: Angles correspond to the position of the tilting axes of the head/table 1: Angles correspond to the spatial angle (the iTNC calculates the position of the tilted axes of the head/table) Bit 2 – 0: The tilting axes are not positioned with Cycle 19 1: The tilting axes are positioned with Cycle 19 Bit 3 – 0: The current tilting-axis position is taken into account with respect to the machine datum 1: The 0° position is assumed for the first rotary axis Bit 4 – 0: Compensate mechanical offset during exchange of the spindle head when calling M128, M114, TCPM or "tilted working plane" 1: Compensate mechanical offset during PLC datum shift Bit 5 – 0: The current tilting-axis position is taken into account with respect to the machine datum 1: The tilting-axis position that was entered with the 3-D ROT soft key applies. Bit 6 – 0: Spatial angle C is realized through a rotation of the coordinate system. 1: Spatial angle C is realized through a rotation of the table	PLC RUN	777
	Bit 7 – 0: The current tilting-axis position is taken into account with respect to the machine datum 1: The active tilting-axis position is a) derived from the tilting angles in the 3D ROT window if manual tilting is active b) derived from the reference coordinates of the rotary axes if tilting is inactive Bit 8 – 0: The tilting axis positioning is considered depending on bit 3, bit 5 and bit 7 1: If manual tilting is active, the datum to be set for the principal axes X, Y and Z is recalculated back to the home position of the tilting element		

MP	Function and input	Software version & behavior	Page
MP7500 (is set via the kinematics table)	Tilt working plane (active preset table) Format: %xxxxxxxxxx Input: Bit 0 – "Tilted working plane" 0: Off 1: On Bit 1 – 0: Angles correspond to the position of the tilting axes of the head/table 1: Angles correspond to the spatial angle (the iTNC calculates the position of the tilted axes of the head/table) Bit 2 – 0: The tilting axes are not positioned with Cycle 19 1: The tilting axes are positioned with Cycle 19 Bit 3 – No function Bit 4 – No function Bit 5 – Test of the tilting axis during "datum setting" in X, Y and Z 0: Current tilting-axis position must fit to the defined tilting angles 1: No test Bit 6 – 0: Spatial angle C is realized through a rotation of the coordinate system. 1: Spatial angle C is realized through a rotation of the table Bit 7 – No function Bit 8 – No function Bit 9 – Reserved	340 422-01, 340 480-02 PLC RUN	778
MP7502	Functionality of M144/M145 Input: %xxx Bit 0 – 0: M144/M145 not active 1: M144/M145 active Bit 1 – M144/M145 in the automatic modes 0: M144/M145 active 1: M144 is activated automatically at the start of an NC program. It can only be deactivated with M145 during an NC program. Bit 2 – M144/M145 in the manual modes 0: M144/M145 not active 1: M144/M145 active	PLC RUN	783
MP7503	Virtual tool axis – Reapproaching the contour and manual traverse in the current tool-axis direction (FCL2 upgrade function) Input: 0: Inactive 1: Active	PLC RUN	784

MP	Function and input	Software version & behavior	Page
MP7506	Selection of kinematics at booting of the control Input: 0 to 999 -1: Function inactive	340 490-03	729
MP7507	Selecting the kinematics for the operating mode Input: %xxx Bit 0 0: Kinematics cannot be selected in Editing operating modes 1: Kinematics can be selected in Editing operating modes for simulation in Test Run mode Bit 1 0: Kinematics cannot be selected in Machining operating modes 1: Kinematics of the real machine can be selected in Machining operating modes Bit 2 0: 3D ROT soft key is available in Test Run mode 1: 3D ROT soft key is available in Test Run mode	340 490-05 340 490-06	1390
MP7510 (only possible via the old kinematics table) MP7510.0-14	Transformed axis Format: %xxxxxx Input: 0: End of the transformation sequence Bit 0 corresponds to axis X Bit 1 corresponds to axis Y Bit 2 corresponds to axis Z Bit 3 corresponds to axis A Bit 4 corresponds to axis B Bit 5 corresponds to axis C Transformation 1 to transformation 15	PLC RUN	778
MP7520 (only possible via the old kinematics table) MP7520.0-14	Additional code for transformation Format: %xx Input: Bit 0 – Tilting axis 0: Swivel head 1: Tilting table Bit 1 – Type of dimension in MP7530.x 0: Incremental dimension for swivel head 1: Absolute with respect to the machine datum for tilting table Transformation 1 to transformation 15	PLC RUN	778
MP7530 (only possible via the old kinematics table) MP7530.0-14	Type of dimension for transformation Entry: Entry of a formula is possible, *** ”Special case: Entering a formula" on page 525 *** 0: Free tilting axis Transformation 1 to transformation 15	PLC RUN	779

MP	Function and input	Software version & behavior	Page
MP7550	Home position of the tilting element	PLC	779
(only possible via the old kinematics table)	Input: –99 999.9999 to +99 999.9999	RUN	
MP7550.0	A axis		
MP7550.1	B axis		
MP7550.2	C axis		

4.4.15 Hardware

MP	Function and input	Software version & behavior	Page
MP7600.0	Only CC 422: Position controller cycle time = MP7600.0 · 0.6 ms Input:1 to 20 Proposed input value: 3 (= 1.8 ms) Proposed input value for basic version: 6 (= 3.6 ms)	RESET	843
MP7600.1	Only CC 422: PLC cycle time = MP7600.1 ·Position controller cycle time = MP7600.0 · MP7600.1 · 0.6 ms Input: 1 to 20 Proposed input value: 6 (= 10.8 ms) Proposed input value for basic version: 3 (= 10.8 ms)		843, 1603
MP7602	Only CC 424(B): PLC cycle time Input: 0 to 60 [ms] 0 to 10: 10.8 ms	340 422-03, 340 480-03	1083
MP7610.x	Only CC 424(B): Definition of the control loops as single or double speed Input: %xxxx	340 490-01, 340 492-01	1077
MP7610.0	Control loop of the 1st controller PCB		
MP7610.1	Control loop of the 2nd controller PCB		

MP	Function and input	Software version & behavior	Page
MP7620	Feed-rate override and spindle speed override Format: %xxxxxxx Input: Bit 0 – Feed-rate override if rapid traverse key is pressed in **Program Run** mode. 0: Override not effective 1: Override effective Bit 1 – No function Bit 2 – Feed-rate override if rapid traverse key and machine direction button are pressed in **Manual mode** 0: Override not effective 1: Override effective Bit 3 – Feed-rate override and spindle speed override in 1% increments or according to a nonlinear characteristic curve 0: 1% steps 1: Nonlinear characteristic curve Bit 4 – No function Bit 5 – Rapid traverse override instead of spindle override 0: Potentiometer is used for spindle override 1: Potentiometer is used for rapid traverse override Bit 6 – Feed-rate smoothing 0: Not active 1: Active Bit 7 – Reserved Bit 8 – Informational text if feed-rate or rapid-traverse override is set to 0% 0: Informational text inactive 1: Informational text active	PLC RUN 340 490-05	841, 959, 1147
MP7621	Reserved		–
MP7630	Recovery time after EMERGENCY STOP test can be configured Input: 1 to 999 [ms] 0: Previous behavior (through 340 49x-03)	340 490-03	934
MP7640	Handwheel Input: 0: No handwheel 1: Reserved 2: HR 130 3: Reserved 4: Reserved 5: Up to three HR 150 via HRA 110 6: HR 410 7 to 10: Reserved 11: HR420/HR520 without LED activation 12: In future for HR 550FS wireless handwheel 13: HR520 with LED activation	PLC RUN	1449

MP	Function and input	Software version & behavior	Page
MP7641	Handwheel settings Format: %xxxxxxxxxxxxxx Input: Bit 0 – HR 410: Entry of subdivision factor 0: Through iTNC keyboard 1: Through PLC Module 9036 Bit 1 – HR 420: With detent positions 0: Without detent positions 1: With detent positions Bit 2 – HR 420: Axis direction keys and rapid traverse 0: Controlled by the NC 1: By PLC Bit 3 – HR 420: NC Start / NC Stop 0: By the NC 1: By the PLC Bit 4 – Handwheel superimposition in the active tool-axis direction 0: Behavior as before 1: VT axis can be selected Bit 5 – Inactive behavior of HR 420 0: Report the keys of the HR 420 to the PLC only when the HR is active 1: Report the keys of the HR 420 to the PLC even if the HR is not active Bit 6 – Selecting and traversing auxiliary axes with HR 420 0: Traversing auxiliary axes not possible 1: Traversing auxiliary axes is possible Bit 7 – Teach-In button on HR 550 0: By the NC 1: By PLC Bit 8 – CTRL button on HR 550 0: By the NC 1: By the PLC Bit 9 – PLC soft keys with active HR 420 0: PLC soft keys are not active when HR is active 1: PLC soft keys are active when HR is active	PLC RUN 340 490-05	1449

MP	Function and input	Software version & behavior	Page
MP7645	Initializing parameter for handwheel	PLC	1454, 1461
MP7645.0	Assignment of the keys on handwheel HR 410 Input: 0: Evaluation of the keys by NC, including LEDs 1: Evaluation of the keys by PLC	RUN	
MP7645.0	Assignment of a third handwheel via axis selector switch S2, when MP7645.2 = 0 Input: 0: Switch position 1 (at the left stop) 3rd handwheel axis Z Switch position 2 3rd handwheel axis IV Switch position 3 3rd handwheel axis V 1: Switch position 1 3rd handwheel axis X Switch position 2 3rd handwheel axis Y Switch position 3 3rd handwheel axis Z Switch position 4 Third handwheel axis IV Switch position 5 3rd handwheel axis V 2: Switch position 3 3rd handwheel axis Z Switch position 4 Third handwheel axis IV Switch position 5 3rd handwheel axis V		
MP7645.1	Fixed assignment of third handwheel if MP7645.2 = 1 Input: 1: Axis X 2: Y axis 4: Z axis 8: Axis IV (MP410.3) 16: Axis V (MP410.4)		
MP7645.2	Assignment of a third handwheel via axis selector switch or MP7645.1 Input: 0: Assignment by axis selection switch according to MP7645.0 1: Assignment by MP7645.1		
MP7645.3-7	No function		

MP	Function and input	Software version & behavior	Page
MP7650	Handwheel counting direction (for HRA 110: for each axis) Input: Bit 0 0: Negative counting direction 1: Positive counting direction Axis-specifically only for HRA 110: Bits 0 to 13 represent axes 1 to 14 0: Negative counting direction 1: Positive counting direction	PLC RUN	1449
MP7660	Sensitivity for electronic handwheel Input: 0 to 65 535 [increments]	PLC RUN	1450
MP7670	Subdivision factor for handwheel Input: 0 to 10	PLC RUN	1450, 1455
MP7670.0	Subdivision factor for slow speed		
MP7670.1	Subdivision factor for medium speed (only HR 410)		
MP7670.2	Subdivision factor for fast speed (only HR 410)		
MP7671	Handwheel feed rate in the Handwheel operating mode with HR 410 Input: 0 to 1000 [% of MP1020]	PLC RUN	1455
MP7671.0	Slow speed		
MP7671.1	Medium speed (only HR 410)		
MP7671.2	Fast speed (only HR 410)		
MP7672	HR 410, distance per handwheel step Input: 0.0000 to 1.0000 [mm]	340 490-03	1455
MP7672.0	Slow speed		
MP7672.1	Medium speed		
MP7672.2	Fast speed		
MP7674.x	Handwheel, axis-specific subdivision factor Input: 1 to 10 0: No limitation	PLC RUN 340 490-03	1450
MP7675.x	Handwheel, axis-specific maximum path Input: 0.0001 to 10.0000 [mm] 0: No limitation	PLC RUN 340 490-03	1450

MP	Function and input	Software version & behavior	Page
MP7680	Machine parameter with multiple function Format: %xxxxxxxxxxxxxxxx Input: Bit 0 – Memory function for axis-direction keys with M4562: 0: Not saved 1: Saved if M4562 is set Bit 1 – Returning to the contour 0: Not active 1: Active Bit 2 – Block scan 0: Not active 1: Active Bit 3 – Interruption of block scan for STOP or M06 0: Interruption 1: No interruption Bit 4 – Inclusion of programmed dwell time during the block scan: 0: Include the dwell time 1: Do not include the dwell time Bit 5 – Start of calculation for block scan 0: Start from block with cursor 1: Start from beginning of program Bit 6 – Tool length in blocks with normal vectors: 0: Without R2 from tool table (south pole) 1: With R2 from tool table (center of sphere) Bit 7 – Inserting a defined rounding arc or spline: 0: Defined rounding arcs are always inserted 1: Defined rounding arcs are always inserted if the acceleration from MP1060.x or MP1070 was exceeded.	PLC RUN	895, 896, 1202, 1224, 1227, 1352, 1541

MP	Function and input	Software version & behavior	Page
MP7680	Machine parameter with multiple function	PLC	
	Bit 8 – Insertion of rounding arc or cubic spline 0: Rounding arc is inserted. 1: A cubic spline is inserted instead of a rounding arc. Bit 9 – Constant jerk on spline (bit 8 = 1) 0: No constant jerk 1: Constant jerk Bit 10 – Cutter-radius-compensated outside corners 0: Insertion of a circular arc 1: Insertion of a spline curve Bit 11 – Behavior of M116 0: Rotary axis is parallel to linear axis 1: Any position of rotary axis to linear axis Bit 12 – Behavior of Cycle 28 0: Standard behavior 1: The slot wall is approached and departed tangentially; at the beginning and end of the slot a rounding arc with a diameter equal to the slot width is cut Bit 13 – Behavior during program interruption with axis movement 0: Automatic activation of **APPROACH POSITION** 1: Do not automatically activate **APPROACH POSITION** Bit 14 – Behavior of NC start after NC stop and internal stop 0: NC start permitted 1: NC start only permitted after block scan GOTO Bit 15 – NC Start if program is aborted 0: NC start permitted 1: NC Start not permitted (message window) Bit 12 – Behavior of Cycle 39 0: Approach/departure movement on an arc is active 1: Approach/departure movement on an arc is active	RUN	

MP	Function and input	Software version & behavior	Page
MP7681	M/S/T/Q transfer to the PLC during block scan Format: %xxxx Input: Bit 0 – 0: Transfer M functions to the PLC during block scan. 1: Collect M functions and transfer them to the PLC after block scan. Bit 1 – 0: Transfer T code to the PLC during block scan 1: Transfer last T code to the PLC after block scan Bit 2 – 0: Transfer S or G code to the PLC during block scan 1: Transfer S or G code to the PLC after block scan. Bit 3 – 0: Transfer FN19 outputs to the PLC during block scan 1: Transfer last FN19 outputs to the PLC after block scan. Bit 4 – MP subfiles during block scan 0: MP subfiles are not activated during block scan 1: MP subfiles are activated during block scan	PLC RUN 340 490-05	1229

MP	Function and input	Software version & behavior	Page
MP7682	Machine parameter with multiple function Format: %xxxxxxxx Input: Bit 0 – Incremental block after TOOL CALL 0: With length compensation 1: Without length compensation Bit 1 – Reference value for calculating the preset during datum setting 0: Actual value is calculated 1: Nominal value is calculated Bit 2 – Traverse path of rotary axes with modulo display 0: Positioning without passing over zero 1: Positioning on the shortest path Bit 3 – Reserved, enter 0 Bit 4 – Tolerance for compensating movements with tilting axes (M114) 0: Tolerance will be included 1: Tolerance will not be included Bit 5 – Feed rate with M128 or TCPM 0: Feed rate refers to tool tip 1: Feed rate from interpolation of all axes involved Bit 6 – Behavior with TOOL DEF strobe 0: Depending on the NC program, the TOOL DEF strobe must be acknowledged by the PLC (TOOL DEF within a contiguous contour) 1: TOOL DEF strobe must always be acknowledged by the PLC Bit 7 – Block elements TOOL CALL and S in ISO blocks 0: Machine as programmed 1: Machine at beginning of block (block display does not change) Bit 8 – Behavior of M8 at the end of Cycles 202 and 204 0: At the end of Cycles 202 and 204, the status of M8 is restored to that before the cycle call (behavior unit now). 1: At the end of Cycles 202 and 204, the status of M8 is not restored automatically.	PLC RUN 340 490-05	1140, 1141, 1201, 779, 1549

MP	Function and input	Software version & behavior	Page
MP7682	Bit 9 – Loading of "Tilted working plane" status 0: The "Tilted working plane" status is not loaded into the Manual operating mode after a program interruption (behavior until now). 1: The "Tilted working plane" status is loaded into the Manual operating mode after a program interruption Bit 10 – Peripheral milling active/inactive 0: Peripheral milling allowed 1: Peripheral milling inactive Bit 11 – Reserved Bit 12 – Error message "Tool radius too large" is suppressed if R2 > R 0: Error message displayed 1: Error message is suppressed		1246

MP	Function and input	Software version & behavior	Page
MP7683	Executing pallet tables and NC programs Format: %xxxxxxxxx Input: Bit 0 – No function Bit 1 – **Program Run, Full Sequence** mode 0: During the start, a complete NC program is run. 1: At the start all NC programs are executed up to next pallet. Bit 2 – **Program Run, Full Sequence** mode 0: As defined in bit 1 1: All NC programs and pallets up to the end of the table are executed. Bit 3 – When the end of the table is reached, the process begins again with the first line. 0: Function is not in effect 1: Function is effective (bit 2=1) Bit 4 – Editing the active pallet table 0: Active pallet table cannot be edited. 1: In the **Program Run, Full Sequence** and **Program Run, Single Block** modes, the current pallet table can be edited. Bit 5 – AUTOSTART soft key 0: Do not display soft key 1: Display soft key Bit 6 – Display of pallet table and NC program 0: Both simultaneously in a split screen 1: Pallet table or NC program individually Bit 7 – AUTOSTART function 0: AUTOSTART function by NC 1: AUTOSTART function by PLC Bit 8 – Procedure for tool-oriented machining in the **Program Run** operating modes 0: NC start machines all workpieces on the pallet until the next tool change 1: NC start executes all NC programs until the end of the pallet	PLC RUN	1201, 1442
MP7683	Executing pallet tables and NC programs Bit 5 – EDIT PALLET soft key 0: EDIT PALLET soft key is not displayed 1: EDIT PALLET soft key is displayed	PLC RUN 340 490-05	1201, 1442

MP	Function and input	Software version & behavior	Page
MP7684	Nominal position value filter (bit 0 to bit 4) and path control with M128 or TCPM (bit 5 to bit 7 permitted) Format: %xxxxxxxx Input: Bit 0 – Nominal position value filter 0: Include acceleration 1: Do not include the acceleration Bit 1 – Nominal position value filter 0: Include the jerk 1: Do not include the jerk Bit 2 – Nominal position value filter 0: Include the tolerance 1: Do not include the tolerance Bit 3 – Nominal position value filter 0: Include the radial acceleration 1: Do not include the radial acceleration Bit 4 – Nominal position value filter 0: Include jerk and tolerance limit at changes in the curvature 1: Do not include jerk and tolerance limit at changes in the curvature Bit 5 – Reserved Bit 6 – Reserved Bit 7 – Reserved Bit 8 – Reserved Bit 9 – Accelerated 5-axis machining with M128 with many rotary axis motions that are less than 2° per positioning block (not with handwheel superimpositioning with M118) 0: Inactive 1: Active Bit 10 - Modification of the calculation of the contouring feed rate at the beginning of a contour element 0: Active 1: Previous behavior (through 340 490-02)	PLC RUN 340 490-03	831, 831, 840

MP	Function and input	Software version & behavior	Page
MP7690	Evaluation of the electronic ID labels Input: %xx Bit 0 – HEIDENHAIN power modules 0: Active 1: Inactive Bit 1 – HEIDENHAIN synchronous motors 0: Active 1: Inactive Bit 2 – HEIDENHAIN power supply units 0: Active 1: Inactive	340 422-06, 340 480-06 340 490-05	1297
MP7691.x	Size of internal diagnostics files (FILO memory) for error searching. Can only be evaluated by HEIDENHAIN. Set MP7691.x = 0.		–
MP7691.0	OS trace Log file with reports from the operating system (keystroke capture/selected programs, operating mode, external connections, etc.) **TNC:\trace.dmp** (file saved before switch-off) **TNC:\trace.act** (current recording) Input: 1 to 10 [megabytes] 0: Inactive (default)	340 420-05	
MP7691.1	TCP trace Log files with reports of the network traffic (TCP/IP). 10 files are created with the size from MP7691.1 in [megabytes]. **TNC:\tcpdump\capture1 - capture10** Input: Each file with 1 to 10 [megabytes] 0: Inactive (default)	340 490-02	
MP7691.2	NC trace Size of the log file with reports from the NC software. 10 files are created that are 10 · [MP7691.2] megabytes large **TNC:\ncpdump\capture1 - capture10** Input: Each file with 1 to 10 [10KB] 0: Inactive (default)		
MP7691.3	Kernel trace Size of the log file with reports from the NC kernel. 10 files are created that are 10 · [MP7691.3] megabytes large **TNC:\klog\0.log - 9.log** Input: Each file with 1 to 10 [10KB] 0: Inactive (default)		

4.4.16 Second Spindle

MP	Function and input	Software version & behavior	Page
MP13010 to MP13530	Machine parameter block for the second spindle Input: Function and input range are identical with MP3010 to MP3530.		981

5 Modules, Markers and Words

5.1 Overview of Modules

Module	Function	SW version	Page
9000/ 9001	Copy in the marker or word range		1778
9002	Read all inputs of a PLC input/output unit		1469
9003	Read the analog input of the MC and of the PL 4xxB		1473
9004	Edges of PLC inputs		1471
9005	Update all outputs of a PLC input/output unit		1470
9006	Set and start PLC timer		1679
9007	Diagnostic information of the PL 4xxB		1465
9008	Read certain inputs of a PL 4xxB		1470
9009	Update certain outputs of a PL 4xxB		1471
9010/ 9011/ 9012	Read in the word range		1779
9019	Size of the processing stack		1685
9020/ 9021/ 9022	Write in the word range		1780
9031	Overwrite machine parameter		529
9032	Read machine parameter		530, 1673
9033	Select machine parameter file		536
9034	Load a machine parameter subfile		537
9035	Read status information		660, 661, 696, 1152, 1477,
9036	Write status information		1451, 1476
9038	Read general axis information		659
9040	Reading the axis coordinates (format = 0.001 mm)		1142
9041	Reading the axis coordinates (format = 0.0001 mm)		1143
9042	Read the spindle coordinates (format = 0.001°)		945

Module	Function	SW version	Page
9044	Read the spindle coordinates (format = 0.0001°)		945
9045	Reading the 3-D ROT data	340 490-01	775
9048	Interrogate the operating states of axes	340 490-03	930
9050	Conversion from binary numbers → ASCII		1781
9051	Conversion from binary numbers → ASCII		1782
9052	Conversion from ASCII → binary		1783
9053	Conversion from ASCII → hexadecimal		1783
9054	Conversion from ASCII → hexadecimal binary		1784
9055	Local time		1164
9060	M function status		1237
9061	Status of non-modal M functions		1237
9063	Collision monitoring, activating/ deactivating	340 490-04	1400
9064	Status of collision monitoring	340 490-03	1401
9065	Status of the commissioning function		1109
9066	Status of HEIDENHAIN supply unit		923
9066	Status of software settings	340 490-04	307
9070	Copy a number from a string		1758
9071	Find the string length		1759
9072	Copy a byte block into a string	340 422-06, 340 480-06	–
9073	Copy a string into a byte block	340 422-07, 340 480-07	–
9074	Load texts from error/dialog files into a PLC string	340 490-06	1264
9080	Clear the small PLC window		1303
9081	Interrogate the status of the small PLC window		1303
9082	Display a string in the small PLC window		1304
9083	Display a moving-bar diagram in the small PLC window		1305
9084	Display PLC error messages with additional data	340 422-09, 340 480-09	–
9085	Display PLC error messages		1169
9086	Clear PLC error message		1170
9087	Status of PLC error message		1171
9088	Displaying the M functions		1149
9089	Control in operation		1150
9090	Select a line in the pallet table		1443
9091	Find the line number of a tool in the tool table		1542

Module	Function	SW version	Page
9092	Search for an entry in the tables selected for execution (.T/.D/.TCH)		1520
9093	Read data from tables selected for program run (.T/.D/.TCH)		1522
9094	Write data into a tool and datum table		1523
9095	Select active line in configuration file		696
9096	Delete a line from the tool table		1524
9097	Select the geometry description		741
9098	Find the active geometry description		740
9100	Assign data interface		1832
9101	Release data interface		1833
9102	Status of data interface		1833
9103	Transmit string through data interface		1834
9104	Receive string through data interface		1835
9105	Transmit binary data through data interface		1836
9106	Receive binary data through data interface		1837
9107	Read from receiving buffer		1838
9110	Transmit a message via LSV2		1839
9111	Receive a message via LSV2		1840
9112	Transmit ASCII characters via data interface		1841
9113	Receive ASCII characters via data interface		1842
9120	Start PLC axis		670
9121	Stop PLC axis		670
9122	Status of PLC axis		671
9123	Traverse the reference marks of PLC axes		672
9124	Feed rate override for PLC axis		672
9125	Stop PLC axis at next Hirth grid position		673
9128	Torque limiting by the PLC	340 490-03	888
9129	Status of torque limiting by the PLC	340 490-03	889
9130	Output of an analog voltage		1475
9133	Hardware information of the MC 42x(B)		908 910
9135	Switch on the 3-D touch probe		1484
9136	Switch the touch probe on/off	340 420-06	1484
9137	Diagnostic information of the PL 510	340 422-05, 340 480-05	1466
9138	Read analog input of the PL 510	340 422-05, 340 480-05	1474
9139	Reset short-circuit monitoring of the outputs on the PLD 16-8	340 422-05, 340 480-05	1467
9140	Set axis-specific feed-rate limit	340 422-06, 340 480-06	–

Module	Function	SW version	Page
9141	Read axis-specific feed-rate limit	340 422-06, 340 480-06	–
9143	Activate the brake test		939
9144	Activate the emergency stop test	340 490-06	937
9145	Actual-to-nominal value transfer		858
9146	Saving and reestablishing actual position values		984
9147	Assign a reference value to an axis		801
9148	Use nominal value as actual value	340 420-06	1478
9149	Only CC 424(B): Set/Read field angle via PLC	340 490-01	1111
9150	Axis-specific reading of axis traverse limits		663
9151	Select traverse range and axis designation		664
9152	Select traverse range, axis display and axis designation		665
9153	Switch the touch probe axis		1486
9155	Axis switchover from closed loop to open loop		985
9156	Axis switchover from open loop to closed loop		985
9157	Drive controller status		875
9158	Maximum torque		886
9159	Advance status report: Drives will be switched off		875
9160	Status request for temperature monitoring and I^2t monitoring		916
9161	Enable the drive controller		876
9162	Status request of the drive controller		876
9163	Switching the operating modes		976
9164	Read the actual speed value of the motor		860
9165	Sample the current motor temperature		909
9166	Momentary utilization of the drive motor		922
9167	Supply voltage monitoring		907
9168	Interrogate the commissioning status		1026
9169	Axes for which I32 does not switch off the drives		876
9170	Find the current torque	340 490-01	923
9171	Oriented spindle stop		967
9173	Speed-dependent wye/delta switchover		979
9174	Spindle status regarding wye/delta switchover		980
9175	Switch the spindle		982
9179	Status information about spindle(s)	340 422-10, 340 480-10	982

Module	Function	SW version	Page
9180	Simulation of NC keys		1341
9181	Disable individual NC keys		1341
9182	Re-enable individual NC keys		1342
9183	Disable groups of NC keys		1342
9184	Re-enable groups of NC keys		1343
9185	Disable touchpad/Interrogate status		1343
9186	Call a soft-key function		1343
9187	Status of a soft-key function call		1344
9188	Call a soft-key function	340 490-04	1344
9189	Shut down the control		1244
9190	Start the operating times		1160
9191	Stop the operating times		1160
9192	Read the operating times		1161
9193	Set the operating times		1161
9194	Alarm when operating time exceeded		1162
9195	System time		1163
9196	Find the PLC cycle time		1603
9197	Start cyclic timers		1679
9200	Display/delete PLC soft-key row		1338
9201	Display/delete PLC soft key		1339
9202	Select/deselect PLC soft keys and PLC windows		1339
9203	Activate PLC soft-key menu		1328
9204	Update the PLC soft keys		1329
9205	Set the word for acknowledgment of PLC soft keys		1330
9206	Change the settings of the PLC soft keys		1331
9207	Replace PLC soft keys		1332
9208	Status information of the PLC soft keys		1333
9210	Open or clear screen mask for the PLC window		1317
9211	Status of the large PLC window		1319
9215	Activate a PLC pop-up window		1180
9216	Pop-up window with tool selection list	340 420-03	1529
9217	Pop-up window for messages	340 422-03, 340 480-03	1182
9220	Renewed traversing of the reference marks		807

Module	Function	SW version	Page
9221	Start a PLC positioning movement		680
9222	Status request of PLC positioning movement		681
9223	Free rotation		1145
9224	Stop PLC positioning movements	340 490-01	682
9225	Compensation value for the reference mark		800
9227	Position auxiliary axes and NC axes	340 490-05	683
9228	Coupling function for auxiliary axes	340 490-06	675
9229	Status of coupling function	340 490-06	677
9230	Datum shift		1480
9231	Compensation of thermal expansion		698
9240	Open a file		1368
9241	Close a file		1369
9242	Positioning in a file		1370
9243	Read from a file line by line		1371
9244	Write to a file line by line		1372
9245	Read a field from a table		1360
9246	Write to a field in a table		1362
9247	Search for a condition in a table		1363
9248	Rename, copy or delete files	340 490-03	1356
9250	Start the PLC editor for tables		1366
9251	End the PLC editor for tables		1367
9252	Position the cursor in the PLC editor		1367
9255	Reading a field from a table as an integer value		1361
9256	Write to a field in a table		1363
9260	Receive events and wait for events		1767
9261	Send events		1768
9262	Context change between spawn processes		1769
9263	Interrupt a spawn process for a defined time		1769
9270	Read a code word		1643
9271	Write a code word		1643
9275	Write ASCII data into the log		1274
9276	Write operand contents into the log		1275
9277	Write data into the OEM log	340 422-09, 340 480-09	1275
9279	Control reset		1244
9280	Start the NC macro (Run pallet entry)		1446
9281	Select a line in the pallet table		1443

Module	Function	SW version	Page
9282	Tool usage test for pallet table	340 422-10, 340 480-10	1544
9285	Set the access level	340 490-05	1158
9290	Select a file		1356
9291	Call an NC macro		1645
9300	Lock/release the pocket table		1535
9301	Find the number of an entry in the pocket table		1537
9302	Search for a vacant pocket in the tool magazine		1537
9304	Copy columns P1 to P5 to the pocket table	340 420-03, 340 490-03	1530
9305	Tool exchange in the pocket table		1536
9306	Exchange tools between tool magazines		1538
9310	Read the machine parameter from the run-time memory		531
9311	Dynamically change values for friction compensation	340 490-03	1100
9312	Change machine parameters in the current machine parameter file	340 490-03	532
9313	Read machine parameters from current machine parameter file	340 490-03	534
9314	Activate/Deactivate machine-parameters	340 490-06	534
9320	Status of the NC program end		1200
9321	Find the current block number	340 420-06	1197
9322	Information of the current NC program	340 422-09, 340 480-09	1198
9340	Search for a pocket depending on magazine rules	340 420-03	1531
9341	Edit a pocket table depending on magazine rules	340 420-03	1532
9342	Find magazine and pocket number	340 420-06	1533
9343	Compile and activate magazine rules	340 422-10, 340 480-10	1534
9350	Read data from the tool table	340 422-07, 340 480-07	1513
9351	Write data to tool table	340 422-07, 340 480-07	–
9390	Open the online help window with the control’s browser	340 490-03	1191
9391	Display an error message with additional offset	340 490-03	1192
9392	Display PLC error message with help offset	340 490-05	1171

5.2 Overview of Markers and Words

A list of PLC operands with brief description in English and German (GLB_NC_de.DEF, GLB_NC_en.DEF) is on the control under PLC:\JH\.

Operand		Description	Set	Reset	SW version	Page
M	1900 - 1999	Decoded M function if M4571 is set	NC	NC		1236
M	4000	Spindle in position	NC	NC		966
M	4001	Nominal speed command signal of the spindle not in the ramp	NC	NC		950
M	4002	Nominal speed value = 0	NC	NC		950
M	4003	Nominal speed value output analog or digital (MP3010 = 3 to 8)	NC	NC		948
M	4004	Impermissible speed was programmed	NC	NC		955
M	4005	Status display and nominal speed value output for M03	PLC	PLC		954, 1148
M	4006	Status display and nominal speed value output for M04	PLC	PLC		954, 1148
M	4007	Status display M05 and spindle stop	PLC	PLC		954, 1148
M	4008	Disable speed output for spindle	PLC	PLC		955, 1148
M	4009	Counterclockwise spindle rotation (for gear change)	PLC	PLC		957
M	4010	Clockwise spindle rotation (for gear change)	PLC	PLC		957
M	4011	Activate rotational speed MP3520.0 and direction of rotation from M4013	PLC	PLC		969
M	4012	Open the spindle control loop	PLC	PLC		966, 1490
M	4013	Direction for spindle orientation from a standstill (M03 = 0; M04 = 1)	PLC	PLC		969
M	4014	Reverse the direction of spindle rotation	PLC	PLC		954
M	4015	Renewed evaluation of the spindle reference mark	PLC	NC		966
M	4016	Cycle 13 is executed	NC	PLC		969
M	4017	Servo-controlled spindle in motion	NC	NC		966
M	4018	Reference mark for spindle not yet traversed	NC	NC		966
M	4019	Reversing the counting direction of the position encoder on the spindle	PLC	PLC		954
M	4030	Cycle 2 or Cycle 17 active	NC	NC		971, 975
M	4031	Cycle 17 or Cycle 18 active	NC	NC		975

Operand		Description	Set	Reset	SW version	Page
M	4040	Status display M07, M08, and M09 highlighted	PLC	PLC		1148
M	4041	Status display M07, M08, M09, MK	PLC	PLC		1148
M	4042	Status display M07, M08, M09, MK	PLC	PLC		1148
M	4050	Touch probe not ready, ready signal is missing	NC	NC		1483
M	4051	Stylus deflected before start of probing cycle	NC	NC		1483
M	4052	Stylus is deflected, probing process is completed	NC	PLC		1483
M	4053	Probing process has been completed or canceled	NC	NC		1483
M	4054	Battery voltage too low (battery warning at touch probe connection); evaluated only during the probing process	NC	NC	Not supported as of 340 422-03, 340 480-03; active again as of 340 490x-04	1483
M	4055	Enable the probing process	NC	PLC		1483
M	4056	NC stop in all operating modes if stylus is deflected	PLC	PLC		1483
M	4057	Touch probe cycle active (FN17: ID990 NR2)	NC	NC	340 422-09, 340 480-09	–
M	4060	Cycle for tool measurement started	NC	NC		1507
M	4061	0: Measure the tool 1: Check the tool	NC	NC		1507
M	4062	0: Wear tolerance not exceeded 1: Wear tolerance exceeded	NC	NC/ PLC		1507
M	4063	0: Breakage tolerance not exceeded 1: Breakage tolerance exceeded	NC	NC/ PLC		1507
M	4065	Workpiece dimensions are OK	NC	PLC		1491
M	4066	Workpiece must be reworked	NC	PLC		1491
M	4067	Workpiece is scrap	NC	PLC		1491
M	4070	Strobe signal for gear code	NC	NC		957
M	4071	Strobe signal for S code	NC	NC		961
M	4072	Strobe signal for M function	NC	NC		1236
M	4073	Strobe signal T code (P code) with TOOL CALL	NC	NC		1556, 1573
M	4074	Strobe signal T code (P code) with TOOL DEF	NC	NC		1556, 1573
M	4075	Transfer active with FN19	NC	NC		1647
M	4090	Acknowledgment of "gear change completed"	PLC	PLC		957
M	4091	Acknowledgment of S code	PLC	PLC		961
M	4092	Acknowledgment of M function	PLC	PLC		1236

Operand		Description	Set	Reset	SW version	Page
M	4093	Acknowledgment of T code (P code) with TOOL CALL	PLC	PLC		1556, 1573
M	4094	Acknowledgment of T code (P code) with TOOL DEF	PLC	PLC		1556, 1573
M	4095	Acknowledgment of transfer with FN19	PLC	PLC		1647
M	4120 - 4128	PLC positioning axis 1 to 9 active	NC/PLC	NC/ PLC		685
M	4130	Activation of spindle orientation, or spindle orientation has been started with Module 9171	NC/PLC	NC		969
M	4131	Activation of Q-parameter transfer to the NC; data from D258, Q number from W516	PLC	NC		1648
M	4132	Activate datum shift from D528 to D544, or call Module 9230	PLC	NC		1480
M	4133	Start and stop the free rotation function	PLC	NC		1145
M	4134	Activation of a gear range and speed through the PLC	PLC	NC		957
M	4135	Strobe marker for selecting the traverse range	PLC	NC		662
M	4150	Operating mode: Manual Operation	NC	NC		1155
M	4151	Operating mode: Electronic Handwheel	NC	NC		1155
M	4152	Operating mode marker ■ Positioning with Manual Data Input ■ RUN ACTIVE UNIT	NC	NC	Expands 340 490-02	1155
M	4153	Operating mode marker ■ Program Run, Single Block ■ RUN SINGLE UNITS	NC	NC	Expands 340 490-02	1155
M	4154	Operating mode marker ■ Program Run, Full Sequence ■ RUN ALL UNITS	NC	NC	Expands 340 490-02	1155
M	4155	Operating mode: Traversing the reference marks	NC	NC		1155
M	4156	MANUAL TRAVERSE soft key pressed	NC	NC		1227
M	4157	Returning to the contour (MOVE TO POSITION) is active	NC	NC		1227
M	4158	Block scan active	NC	NC		1227
M	4159	PLC editor: END key or soft key pressed	NC	NC/ PLC		1366
M	4160	Pallet table selected	NC	NC		–
M	4161	M/S/T/Q transfer after block scan	NC	NC		1229
M	4162	DNC mode (0=DNC inactive, 1=DNC active)	NC	NC		
M	4163	Alternative operating mode **smarT.NC** is active	NC	NC	340 490-01	1155

Operand		Description	Set	Reset	SW version	Page
M	4170	END PGM, M02 or M30 was executed	NC	NC		1225
M	4172	1st PLC scan after power on	NC	NC		–
M	4173	First PLC scan after interruption of the PLC program	NC	NC		–
M	4174	First PLC scan after editing the MPs (MP edit was exited and the MPs were altered)	NC	NC		–
M	4175	Program interruption, control-in-operation symbol is blinking	NC	NC		1150
M	4176	Control is in operation, control-in-operation symbol is on or is blinking	NC	NC		1150
M	4177	Clearable error message displayed	NC	NC		932
M	4178	Error message EMERGENCY STOP is displayed	NC	NC		932
M	4179	Control is being shut down	NC	NC		1243
M	4180	Rapid traverse programmed (FMAX)	NC	NC		1147
M	4181	NC program selected	NC	PLC		1199
M	4182	AUTOSTART active	NC	NC		1201
M	4183	Time from AUTOSTART expired	NC	NC		1201
M	4185	Internal stop performed	NC	PLC	340 420-06	1200
M	4186	NC program is active in the Test Run mode	NC	NC	340 490-01	1155
M	4188	Compilation process of the PLC project active	NC	NC	340 490-04	1620
M	4189	Emergency off test or self-test of the control has been concluded	NC	NC	340 490-05	936
M	4190	Control for emergency off test, or self test is ready, or test is active	NC	NC	340 490-06	936
M	4191	Control is ready	NC	NC	340 490-06	936
M	4192	Request for machine control voltage ON	NC	NC	340 490-06	936
M	4200	Overflow during multiplication	NC	PLC		1718, 1731, 1761
M	4201	Division by 0	NC	PLC		1719, 1731, 1761
M	4202	Incorrectly executed modulo	NC	PLC		1720, 1731, 1761
M	4203	Error status for PLC module	NC	NC/ PLC		1731, 1761
M	4204	Reserved for errors that the PLC programmer would like to catch	NC	NC		1761
M	4210	Error from Python script with F stop active	NC	NC	340 490-04	
M	4211	Error from Python script with NC stop active	NC	NC	340 490-04	
M	4212	Error from Python script with EM. STOP active	NC	NC	340 490-04	

Operand		Description	Set	Reset	SW version	Page
M	4213	Error from Python script with NC Cancel active	NC	NC	340 490-04	
M	4223	Error from PET table with NC Cancel active	NC	NC	340 422-10, 340 480-10	1168, 1179
M	4220	Error from PET table with F stop active	NC	NC		1168, 1179
M	4221	Error from PET table with NC stop active	NC	NC		1168, 1179
M	4222	Error from PET table with EM. STOP active	NC	NC		1168, 1179
M	4223	Error from PET table with NC Cancel active	NC	NC	340 422-10, 340 480-10	1168, 1179
M	4225	Alternative error reaction active	PLC	PLC	340 490-04	1168
M	4227	PLC error message with priority 0 (error)	NC	NC	340 422-10, 340 480-10	1168
M	4228	PLC error message with priority 1 (warning)	NC	NC	340 422-10, 340 480-10	1168
M	4229	PLC error message with priority 2 (info)	NC	NC	340 422-10, 340 480-10	1168
M	4230	NC start via LSV2	NC	NC		1352
M	4231	NC stop via LSV2	NC	NC		1352
M	4260	Acknowledgment of control-is-ready signal (I3)	NC	NC	340 490-06	932
M	4300 - 4315	Value from MP4310.0	NC	NC		1672
M	4316 - 4331	Value from MP4310.1	NC	NC		1672
M	4332 - 4347	Value from MP4310.2	NC	NC		1672
M	4348 - 4363	Value from MP4310.3	NC	NC		1672
M	4364 - 4379	Value from MP4310.4	NC	NC		1672
M	4380 - 4395	Value from MP4310.5	NC	NC		1672
M	4396 - M4411	Value from MP4310.6	NC	NC		1672
M	4520	Additional T code (P code) follows with TOOL CALL	NC	NC		1557, 1573
M	4521	Tool number zero programmed	NC	NC		1556
M	4522	Tool with pocket number programmed is in effect with MP7480.0 = 3 or 4 and TOOL CALL	NC	NC		1557
M	4523	Tool without pocket number programmed is in effect with MP7480.0 = 3 or 4 and TOOL CALL	NC	NC		1557

Operand		Description	Set	Reset	SW version	Page
M	4524	Special tool called, TOOL CALL	NC	NC		1557, 1573
M	4525	TOOL CALL after expiration of tool life	NC	NC		1557
M	4526 - 4534	Axis 1 to axis 9 is the tool axis	NC	NC		660
M	4538	Geometry of the tool from W264	PLC	NC		1227, 1556
M	4539	Tool number highlighted in the status display	PLC	PLC		–
M	4540	Sequence of tool number or pocket number transfer (M4520 = 1)	PLC	PLC		1557, 1573
M	4541	Special tool in original pocket in spite of variable pocket coding	PLC	PLC		1539, 1557, 1573
M	4542	Do not update pocket number in the pocket table	PLC	PLC		1227, 1557
M	4543	Tool life 1 expired (TIME1 in the tool table)	NC	NC/ PLC		1541
M	4546	Tool life 2 expired (TIME2 in the tool table)	NC	NC/ PLC		1541
M	4547	T and G strobes with TOOL CALL	NC	NC		957, 1556
M	4560	NC stop (0: stop)	PLC	PLC		1352
M	4561	Rapid traverse	PLC	PLC		1352
M	4562	Memory function for axis direction keys (MP7680 bit 0 = 1)	PLC	PLC		1352
M	4563	Feed-rate enable for all axes	PLC	PLC		857
M	4564	NC start	PLC	PLC		1352
M	4570	Unit of measure for transfer with FN19	NC	NC		1647
M	4571	Activation of decoded M-code transfer in M1900 to M1999	PLC	PLC		1236
M	4574	Select the traverse range (with M4575)	PLC	PLC		662, 1487
M	4575	Select the traverse range (with M4574)	PLC	PLC		662, 1487
M	4576	Lock the handwheel	PLC	PLC		1450
M	4577	Disabled key was pressed	NC	PLC		1340
M	4579	INCREMENT OFF/ON soft key	NC	NC		1476
M	4580	Suppress EMERGENCY STOP, open all position control loops, NC stop	PLC	PLC		855, 932
M	4581	Open all position control loops, NC stop, activate "Approach position"	PLC	PLC		855
M	4586	Enable AUTOSTART	PLC	NC/ PLC		1201
M	4587	Rescind feed rate limit above F MAX	PLC	PLC		853

Operand		Description	Set	Reset	SW version	Page
M	4589	Activate datum management via preset table	NC	NC		1196
M	4590	Status fast PLC input from MP4130.2	NC	PLC		1682
M	4591	Status fast PLC input from MP4130.3	NC	PLC		1682
M	4592	Status fast PLC input from MP4130.4	NC	PLC		1682
M	4593	Status fast PLC input from MP4130.5	NC	PLC		1682
M	4600	Faulty internal communication between HeROS and Windows 2000	NC	NC	340 480-06	–
M	4620	Enable LIFTOFF function	PLC	PLC	340 422-06, 340 480-06	–
M	4622	Delay NC macro with **RESETINIT =** from NCMACRO.SYS	PLC	PLC	340 422-10, 340 480-10	806
M	4623	Disable starting of DNC mode (LSV2 access)	PLC	PLC	340 490-03	–
M	4624	Changed axis-traverse limits	NC	PLC	340 490-04	662
M	4625	Disable NC axes when velocity semifeedforward control is active	PLC	PLC	340 490-04	849
M	4626	Disable all key inputs of the TE keyboard unit, including the soft keys	PLC	PLC	340 490-05	1458
M	4627	Trigger condition for integrated oscilloscope	NC	PLC	340 490-06	996
M	4628	Recording of integrated oscilloscope ended	NC	PLC	340 490-06	996
M	4660	HR 420/HR 5x0 assumes control	NC	NC	340 422-09, 340 480-09	1458
M	4661	NC start on HR 420/ HR 5x0	NC	NC	340 422-09, 340 480-09	1458
M	4662	NC stop on HR 420/ HR 5x0	NC	NC	340 422-09, 340 480-09	1458
M	4663	Rapid traverse key on HR 420/ HR 5x0	NC	NC	340 422-09, 340 480-09	1458
M	4664	Spindle start on HR 420/ HR 5x0	NC	NC	340 422-09, 340 480-09	1458
M	4665	Spindle stop on HR 420/ HR 5x0	NC	NC	340 422-09, 340 480-09	1458
M	4666	Plus (+) key on HR 420/ HR 5x0	NC	NC	340 422-09, 340 480-09	1458
M	4667	Minus (–) key on HR 420/ HR 5x0	NC	NC	340 422-09, 340 480-09	1458

Operand		Description	Set	Reset	SW version	Page
M	4668	CTRL key on HR 420/ HR 5x0	NC	NC	340 422-09, 340 480-09	1458
M	4670	Potentiometer on HR 420/ HR 5x0 active	NC	NC	340 490-05	1458
M	4680	Disable activation of the HR 420/ HR 5x0	PLC	PLC	340 490-04	1458
M	4753	Write errors from PLC modules in the PLC log	PLC	PLC	340 422-09, 340 480-09	1265
M	4754	Write diagnostic information in MYDEBUG.LOG	PLC	PLC	340 422-10, 340 480-10	1265

	Marker	Description	Set	Reset	SW version	Page
W	256	Gear code	NC/ PLC	NC/ PLC		957
W	258	S code	NC	NC		961
W	260	Code for M function	NC	NC		1236
W	262	Tool pocket number	NC	NC		1556, 1573
W	264	Tool number	NC	NC		1556, 1573
W	266	Index number of a programmed indexed tool	NC	NC		1542
W	268	Tool magazine number	NC	NC		1536
W	270	Line number in help file	NC	NC		1178
W	272	Mode of operation	NC	NC	Expanded 340 490-02	1155
W	274	Code of the depressed key	NC	NC		1340
D	276	Code of the code number last entered via MOD	NC	NC		1247
D	280	First numerical value from FN19	NC	NC		1647
D	284	Second numerical value from FN19	NC	NC		1647
W	302	Number of the horizontal PLC soft key that was pressed	NC	NC		1337
W	304	Number of the vertical PLC soft key that was pressed	NC	NC		1328
W	320	Nominal speed value [min^{-1}]	NC	NC		948
W	322	Actual speed value [min^{-1}]	NC	NC		948
W	336	Setting of the AFC soft key	NC	NC/ PLC	340 490-04	1219
W	342	Value from column PLC in table AFC.TAB	NC	NC/ PLC	340 490-03	1209
W	348	Current AFC status (0=inactive, 1=learn, 2=control)	NC	NC/ PLC	340 490-03	1208
W	350	Error from AFC that led to NC stop	NC	NC/ PLC	340 490-05	1208
D	356	Programmed speed [0.001 min^{-1}]	NC	NC		948, 957
D	360	Programmed feed rate	NC	NC		853
D	364	Nominal speed value [min^{-1}]	NC	NC		948
D	368	Actual speed value [min^{-1}]	NC	NC		948
D	372	Maximum spindle speed including spindle speed override [min^{-1}]	NC	NC		948
D	388	Current contouring feed rate [mm/min]	NC	NC		853
W	480-484	Analog input at X48 [0.1 V]	NC	NC		1472
W	486 - 490	Temperature input at X48 [0.5 °C]	NC	NC		697, 1472
W	492	Percentage for spindle speed override (NC to PLC)	NC	NC		959

	Marker	Description	Set	Reset	SW version	Page
W	494	Percentage for feed rate override (NC to PLC)	NC	NC		1147
W	516	Q no. 0-7 for numerical data transfer PLC to NC	PLC	PLC		1648
B	518	Definition of the free rotation function	PLC	PLC		1145
B	519	Traverse direction for free rotation	PLC	PLC		1145
W	522	Enabling the high-speed PLC inputs	PLC	PLC		898, 1682
W	524	Open the control loop if drive enabling via X150/X151 is missing	PLC	PLC		874
D	528	Double word with multiple function, here data for transfer from PLC to NC	PLC	PLC		1648
D	528 - 544	Target position for PLC positioning	PLC	PLC		685
D	528 - 544	Datum shift for axis 1 to 5	PLC	PLC		1480
W	560 - 568	Feed rate for PLC positioning	PLC	PLC		685, 1145
W	576 - 584	Lag-tracking axis error compensation	PLC	PLC		697
D	592	Nominal position for spindle orientation	PLC	PLC		969
D	596	Max. feed rate from PLC [mm/min]	NC/ PLC	PLC		853
D	604	Maximum possible spindle speed	PLC	NC/ PLC		948
W	632	Alternative control input variable for AFC	PLC	NC/ PLC	340 490-04	1207
W	754	% function for feed-rate override for free rotation	PLC	PLC		1145
D	756	Programmed rotational speed or rotational speed from the PLC [0.001 min^{-1}]	NC/ PLC	NC/ PLC		957
D	760	Offset in tilting axes touch probe center offset [1/10 000°]	PLC	PLC		1490
W	764	Percentage for spindle speed override (PLC to NC)	NC/ PLC	NC/ PLC		959
W	766	Percentage for feed-rate override (PLC to NC)	NC/ PLC	NC/ PLC		1147
D	768 - 956	Value from MP4210.0 to MP4210.47	NC	NC		1671
W	960 - 968	Value from MP4220.0 to MP4220.4	NC	NC		1672
W	976 - 994	Value from MP4310.0 to MP4310.9	NC	NC		1672
W	1002	Last PLC run-time error that occurred	NC	NC		1683
W	1008	S code for minimum speed	NC	NC		961
W	1016	PLC module that was last processed erroneously	NC	NC		–

	Marker	Description	Set	Reset	SW version	Page
W	1018	Number of files opened by the PLC	NC	NC		1357
W	1020	Number of all open files	NC	NC		1357
W	1022	Error status of the last called PLC module	NC	NC		–
W	1024	Axis enabling	NC	NC		855
W	1026	Axes in position	NC	NC		904
W	1028	Axes in motion	NC	NC		905
W	1030	Current direction of traverse	NC	NC		651
W	1032	Reference marks not yet traversed	NC	NC		806
W	1034	Positive software limit switch was traversed	NC	NC		666
W	1036	Negative software limit switch was traversed	NC	NC		666
W	1038	Prepare to open the position control loop	PLC	PLC		856
W	1040	Axis-specific opening of the position control loop	PLC	PLC		856
W	1042	Deactivation of monitoring functions	PLC	PLC		897
W	1044	Actual-to-nominal value transfer	PLC	PLC		858
W	1046	Manual traverse in positive direction	PLC	PLC		1352
W	1048	Manual traverse in negative direction	PLC	PLC		1352
W	1054	Reference end position	PLC	PLC		806
W	1056	Lubrication pulse: Value in MP4050.x exceeded	NC	NC		667
W	1058	Reset the accumulated distance	PLC	PLC		667
W	1060	Axis-specific feed-rate enable	PLC	PLC		857
W	1062	Lock the handwheel for specific axes	PLC	PLC		1450

6 Configuring the Axes and Spindle

6.1 Control Loops

6.1.1 Selecting the Axes

With MP10 you define which machine axes are to be operable.
The bits may be changed during the run-time without a control reset.
However, the bits to be changed must have been set before the control was switched on.
Changing bits that had not been set leads to a control reset.

MP10	**Active axes**
Format:	%xxxxxxxxxxxxxx
Input:	Bits 0 to 13 represent axes 1 to 14 0: Axis not active 1: Axis active

Screen display

You can define how the axes are shown on the screen:

- In MP100.x, assign a designation to each logical axis.
- Define in MP7291.x the screen line in which the axis is to be displayed.

Rules for the display:

- NC axes are designated with uppercase letters.
- PLC axes are designated with lowercase letters.
- Axes that are not present are given a hyphen "-".

MP100	**Designation of axes**
Format:	-wvucbazyxWVUCBAZYX
Input:	Characters 1 to 14 from the right represent axes 1 to 14
MP100.0	Designation of axes for traverse range 1
MP100.1	Designation of axes for traverse range 2
MP100.2	Designation of axes for traverse range 3

MP7291	**Display of axes on the screen**
Format:	SXYZABCUVWxyzabcuvw-
Input:	Characters 1 to 14 from the right represent lines 1 to 14 Character 15 is the spindle "S", which is always output in line 15.
MP7291.0	Display in traverse range 1
MP7291.1	Display in traverse range 2
MP7291.2	Display in traverse range 3

Assignment of axis keys IV, V, VI and T

On the keyboard unit and the HR 4xx handwheel, you can assign the axis keys IV and V as desired.

The HR 5xx handwheels feature six instead of five axis-address keys. You can also enter the axis "T" in MP410.x for the virtual tool axis VT in the SIK option "Global Settings" if this option is available.

MP410	**Assignment of axis keys IV, V and VI**
Input:	Axis labels A/B/C/U/V/W/T with SW option #44
MP410.3	Axis key IV
MP410.4	Axis key V
MP410.5	VI axis key (only HR 5xx)

Demo operation for NC axes

Operation of NC axes with MP12 enables the simulation of axis motors that are not present on the machine. This permits you to put into operation and test axes that have not yet been mounted on the machine.

The following is required for this:

- The axis to be simulated must be **completely** configured in the machine parameters.
- For safety reasons, the PLC may not output any "drive and axis group enables" for the axes to be simulated while in demo operation. This might necessitate very comprehensive changes to the PLC program.

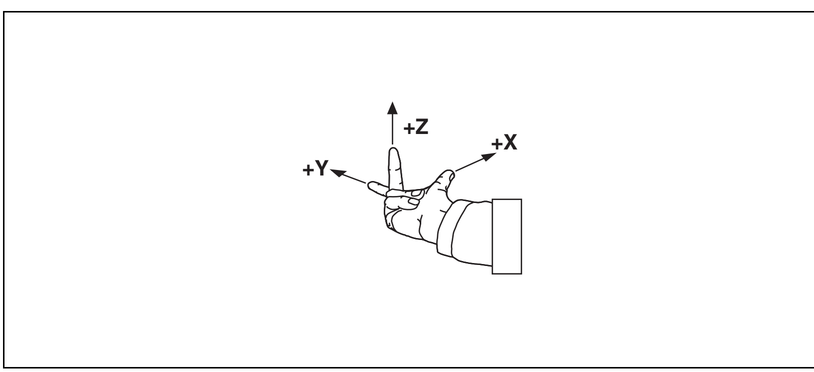
MP12	**Axis-specific demo operation for NC axes**
Format:	%xxxxxxxxxxxxxx
Input:	Bits 0 to 13 represent axes 1 to 14
	0: Demo operation not active
	1: Demo operation active

6.1.2 Axis Designation

Principal axes X, Y, Z

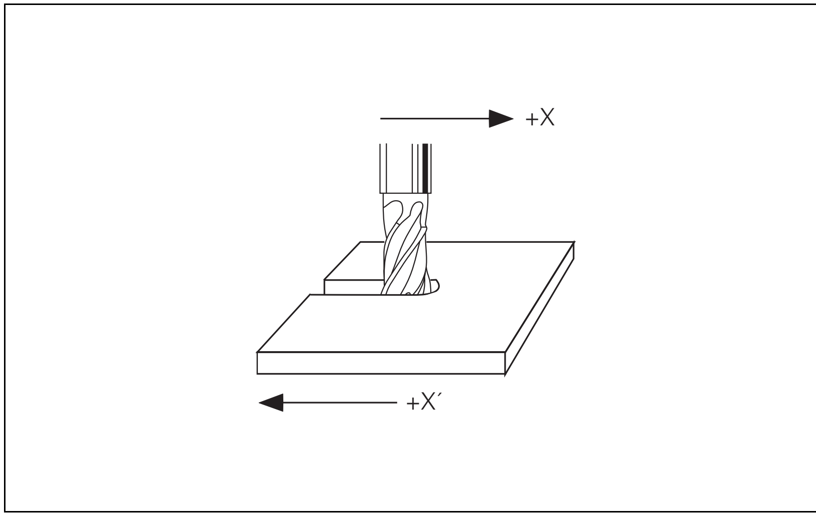
The coordinate axes and their directions of motion are defined in the international standard ISO 841.

An easy way to remember this system is to use the "right-hand rule":



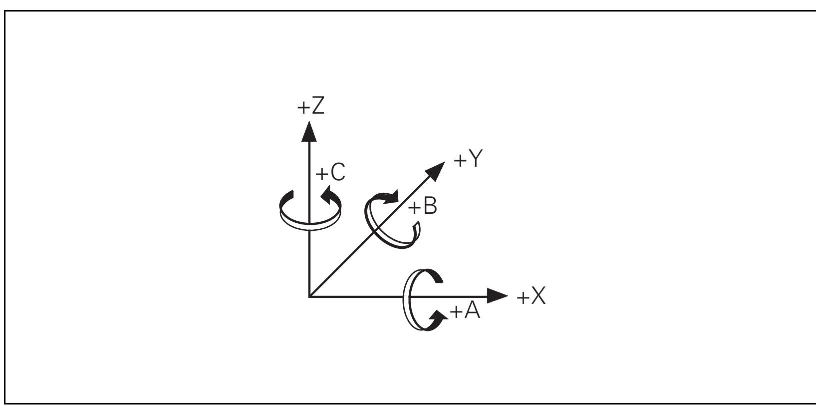
Algebraic signs of the axes

When the programmer writes an NC program, he always assumes that the tool (not the workpiece) is in motion. If the machine moves its workpiece-holding element (table) in a particular axis instead of the tool, then the direction of actual motion is opposite to the direction of axis motion. In this case the direction of motion is designated with the same algebraic sign as the axis direction, but with an apostrophe: +X´, +Y´ and +Z´:



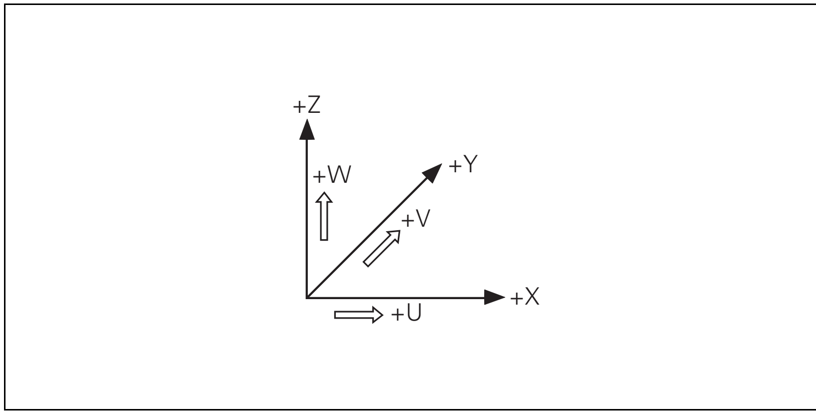
Rotary axes A, B, C

The directions of the rotary axes A, B and C follow the "right-fist rule." The fingers of the closed right hand point in the proper rotational direction of an axis when the thumb points in the direction of the associated linear axis:



Secondary linear axes

The secondary linear axes U, V and W are parallel to the principal axes X, Y and Z.



6.1.3 Encoders

Position encoders report positions and movements of the machine to the control. The iTNC 530 operates with incremental and absolute encoders with EnDat interface.

Signal period

For any given distance the position encoder supplies a fixed number of signal periods. The signal is subdivided 1024 times.

To calculate the signal period, the control requires the following data:

- In MP331.x, enter for each axis the length required for the number of signal periods given in MP332.x.
- In MP332.x, enter for each axis the number of signal periods for the length given in MP331.x.

Note

Large input values in MP331.x and MP332.x cannot be read by the PLC!

The iTNC calculates the quotient:

$$\text{Signal period} = \frac{\text{MP331.x}}{\text{MP332.x}}$$

Note

Digital axes:
If no position encoder (MP110.x = 0) is connected, the data of the speed encoder must be entered in MP331.x and MP332.x. This also applies to speed encoders with EnDat interface, since the incremental track of the speed encoder is used for position feedback control.

HEIDENHAIN offers incremental linear encoders with **distance-coded reference marks.** The nominal increment between two fixed reference marks depends on the encoder being used:

- In MP334.x, enter for each axis the nominal increments between two fixed reference marks.

If the number of grating periods between the reference end position and the first reference mark exceeds the value from MP334.x, the error message `Ref mark <axis>: incorrect spacing` appears. This monitoring is turned off with MP334.x = 0.

Example:
LS 486C:
Incremental linear encoder with distance-coded reference marks, grating period 20 µm (= one signal period covers 0.02 mm), nominal increment between reference marks is 20 mm.

MP331.x = 0.02
MP332.x = 1

$$MP334.x = \frac{20\text{ mm}}{0.02\text{ mm}} = 1000 \text{ (or 0)}$$

MP331.x **Distance for the number of signal periods in MP332**
Input: 0.0001 to +1.797693135E+308 [mm] or [°]

MP332.x **Number of signal periods for the distance in MP331**
Input: 1 to +1.797693135E+308

MP334.x **Nominal increment between two fixed reference marks on encoders with distance-coded reference marks**
Input: 1 to 65 535
0: 1 000

External interpolation

If you connect encoders with TTL signals and an external interpolation unit through the TTL/1 V_{PP} adapter to the control:

- In MP340.x, enter the interpolation factor of the external interpolation unit.

MP340.x **Interpolation factor for external interpolation**
Input: 0 to 99
0 = 1: No external interpolation

Encoder signals

Position encoders with 1-V_{PP} or 11- μA_{PP} signals can be connected to the MC 42x(B).

- With MP115.0, you set the 1-V_{PP} or 11-μA_{PP} signal.
- With MP115.2, you set the maximum input frequency.

> Note
>
> The incremental track data must be entered for the corresponding position encoder inputs for encoders with EnDat interfaces.

MP115.0 **Position encoder input 1 V_{PP} or 11 μA_{PP}**
Format: %xxxxxxxxxxx
Input: Bit 0 to bit 5: Position encoder inputs X1 to X6
Bit 6 to bit 9: Position encoder inputs X35 to X38
Bit 10: No function
0: 1 V_{PP}
1: 11 μA_{PP}

MP115.1 **Reserved**
Format: %xxxxxxxxxxx
Input: Enter %00000000000

MP115.2 **Input frequency of position encoder inputs**
Format: %xxxxxxxxxxx
Input: Bit 0 to bit 5: Position encoder inputs X1 to X6
Bit 6 to bit 9: Position encoder inputs X35 to X38
Bit 10: No function
For 1 V_{PP}: 0: 27 kHz
1: 400 kHz
For 11 μA_{PP}:0: 27 kHz
1: 140 kHz

Direction of traverse

With MP210 and MP1040 you define the direction of traverse of the axes. The counting direction depends on the position in which the encoders are mounted. Configuration errors in these parameters provoke the error message **MOVEMENT MONITORING ERROR IN <AXIS>**. Through W1030 the NC informs the PLC of the direction in which the axes traverse.

If the speed encoder is also used for position measurement, MP210 must be set for the speed encoder. Configuration errors in these parameters provoke the error message **Standstill monitoring <in axis>**.

MP210 **Counting direction of position encoder output signals**
Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Positive
1: Negative

MP1040 **Analog axes: Polarity of nominal value voltage**
Digital axes: Algebraic sign of the nominal speed value
Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Positive
1: Negative

		Set	Reset
W1030	**Current direction of traverse** Bits 0 to 8 represent axes 1 to 9 0: Positive traverse direction 1: Negative traverse direction	NC	NC

Note

The counting direction of the speed encoder signals is defined in the motor table (DIR column). If the error message **C3B0 Motor <AXIS> does not rotate** appears, you must change this entry.

Encoder monitoring

HEIDENHAIN contouring controls monitor the signal transmission from the encoders. With machine parameters MP20.x and MP21.x you activate the monitoring function for the position encoders. The following criteria are checked:

Criterion	Error message
Absolute position with distance-coded reference marks	`Encoder <AXIS> DEFECTIVE`
Amplitude of encoder signals	`Encoder AMPLITUDE TOO LOW <AXIS>`
Edge separation of encoder signals	`Encoder <AXIS>: FREQUENCY TOO HIGH`

MP20 **Monitoring functions for the axes**
Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Monitoring not active
1: Monitoring active
MP20.0 Absolute position of distance-coded reference marks
MP20.1 Amplitude of encoder signals
MP20.2 Edge separation of encoder signals

MP21 **Monitoring functions for the spindle**
Format: %xx
Input: Bit 0 – Spindle 1
0: Monitoring not active
1: Monitoring active
Bit 1 – Spindle 2
0: Monitoring not active
1: Monitoring active
MP21.0 Absolute position of distance-coded reference marks
MP21.1 Amplitude of encoder signals
MP21.2 Edge separation of encoder signals

Note

Please note:

- For digital axes the speed encoders are always monitored.
- For more information on error messages from speed encoders, see the "Error Messages" section.

Monitoring for encoders with EnDat interface:

In the event of a disturbance, the error message **`EnDat defective <error code> <axis>`** will appear.

The error code is shown in hexadecimal notation. Error codes may also appear combined, in which case they add themselves together.

There are two possible types of errors:

- The encoder reports an error.
- Access to the encoder via the EnDat interface is faulty.

Codes for errors reported by the encoder:

Error code	Meaning
0x00000001	Light source defective
0x00000002	Signal amplitude too low
0x00000004	Incorrect position value
0x00000008	Overvoltage
0x00000010	Undervoltage
0x00000020	Overcurrent
0x00000040	Replace the battery
0x00000080	Reserved
0x00000100	Reserved
0x00000200	Reserved
0x00000400	Reserved
0x00000800	Reserved
0x00001000	Reserved
0x00002000	Reserved
0x00004000	Reserved
0x00008xxx	EnDat could not be read. Possible causes: ■ Encoder defective ■ Check the wiring (cable and EnDat amplifier) ■ Encoder not connected ■ Encoder connected to wrong connector ■ Motor and position encoder switched ■ Check cable lengths

Error codes if the access to the encoder via the EnDat interface is faulty:

Error code	Meaning
0x7F	If a motor with a newer generation multiturn encoder is attached to a CC 424 with a software less than 340 49x-02 SP7, the EnDat error 0x7F is reported. In this case the NC software must be updated to 340 49x-SP7 or 340 49x-03 or higher.
0x80010000	Delete the alarm bit
0x80020000	Read the alarm status
0x80040000	Read the number of pulses
0x80080000	Read the number of signal periods
0x80100000	Read the number of differentiable revolutions
0x80200000	Read the measuring steps
0x80400000	Read the serial number
0x80800000	Read the type of encoder
0x81000000	Read the position value
0x82000000	Reserved
0x84000000	Reserved
0x88000000	Read the checksum
0x90000000	Alarm bit remains set
0xA0000000	Timeout while waiting for data – signal "high"
0xC0000000	Timeout while waiting for data – signal "low"
0x80000000	Error during access to EnDat interface

Speed encoder

The iTNC 530 uses the **Type of encoder** entry in the "motor.mot" motor table. If an encoder with Z1 track is entered in the motor table, the message **C310 Z1 track error** appears in the event of an error. If an encoder with EnDat interface is entered in the motor table, the control attempts to communicate with the encoder. If this fails, the error message **C3F0 EnDat not found <axis>** appears.

Attention

If you use the HEIDENHAIN standard motor table motor.mot and motors with EnDat encoders, you might have to change the entry for the motor in the SYS column (type of encoder) of the motor table or enter a new motor.

- SYS = 1: Incremental rotary encoder with Z1 track
- SYS = 2: Absolute speed encoder with EnDat interface

6.1.4 Assignment for Axes

With the following machine parameters you assign the position and speed encoder inputs, the speed command output and the machine parameter block of the current and speed controller to the individual logic axes:

- In MP110.x you enter the number of the position encoder input. An error message appears if an invalid number is entered.
- In MP112.x you enter the number of the speed encoder input.
- In MP120.x you enter the number of the speed command output (analog or digital).
- In MP130.x you enter index number y of machine parameter block MP2xxx.y of the current and speed controller. This way different machine parameter blocks MP2xxx.y can be used for the axis and spindle in C-axis operation.

Note

Depending on the maximum spindle speed, it might no longer be possible to use all PWM outputs (See "Maximum spindle speed" on page 656).

If MP120.x = 0, then the axis will only be displayed.

Digital axes: If MP110.x = 0, then the speed encoder (with or without EnDat interface) is also used for position control.

Note

For axes 7 to 10, only speed encoder inputs X80 to X83 and speed command outputs X57 to X60 can be used.

MP110.x	**Assignment of position encoder inputs to the axes**
Input:	0: No position encoder input 1 to 6: Position encoder inputs X1 to X6 35 to 38: Position encoder inputs X35 to X38 201 to 214: Position encoder inputs X201 to X214
MP112.x	**Assignment of speed encoder inputs to the axes**
Input:	0: No speed encoder input 15 to 20: Speed encoder inputs X15 to X20 80 to 87: Speed encoder inputs X80 to X87
MP120.x	**Nominal speed command outputs of the axes**
Input:	0: No closed-loop axis 1 to 6: Analog speed command outputs 1 to 6 (X8) 7 to 12: Analog speed command outputs 7 to 12 (X9) 51 to 64: Digital speed command outputs X51 to X64
MP130.x	**Y index of the machine parameters MP2xxx.y for the axes**
Input:	0 to 17

6.1.5 Assignment for Spindles

With the following machine parameters you assign the position and speed encoder inputs, the speed command output and the machine parameter block of the current and speed controller to the spindle/spindles:

- In MP111.x you enter the number of the position encoder input. An error message appears if an invalid number is entered.
- In MP113.x you enter the number of the speed encoder input.
- In MP121.x you enter the number of the speed command output.
- In MP131.x and MP132.x you enter index number y of machine parameter block MP2xxx.y of the current and speed controller.

First spindle			**Second spindle**		
Position	**Shaft speed**	**Nominal value**	**Position**	**Shaft speed**	**Nominal value**
X1 to X6, X35 to X38, X201 to X214	X15 to X20, X80 to X87	Digital: X51 to X64	X1 to X6, X35 to X38, X201 to X214	X15 to X20, X80 to X87	Digital: X51 to X64
X1 to X6, X35 to X38, X201 to X214	X15 to X20, X80 to X87	Digital: X51 to X64	X1 to X6, X35 to X38 X201 to X214	–	Analog: 1 to 12
X1 to X6, X35 to X38 X201 to X214	–	Analog: 1 to 12	X1 to X6, X35 to X38, X201 to X214	–	Analog: 1 to 12

Maximum spindle speed

The individual PWM outputs are assigned to different controller groups. The PWM frequency can be set separately for each of the controller groups. If PWM frequencies of > 5000 Hz are set for a controller group, it is no longer possible to use all PWM outputs of the controller group. Then only the first PWM output of the controller group can be used (See "PWM frequencies of the CC 422" on page 1012). The unused PWM outputs must not be entered in MP120.x or. MP121.x. Otherwise, the DSP error message **`C440 PWM frequency <Axis>`** will appear.

The maximum spindle speed is:

$$n_{max} = \frac{f_{PWM} \cdot 60000\ \text{min}^{-1}}{p \cdot 5000\ \text{Hz}}$$

n_{max}: Maximum spindle speed [min^{-1}]
f_{PWM}: PWM frequency [Hz]
p: Number of pole pairs

MP111 **Position encoder input for the spindle/spindles**
Input: 0: No position encoder input
1 to 6: Position encoder inputs X1 to X6
35 to 38: Position encoder inputs X35 to X38
201 to 214: Position encoder inputs X201 to X214
MP111.0 Position encoder input for the first spindle
MP111.1 Position encoder input for the second spindle

MP113	**Speed encoder for the spindle/spindles**
Input:	0: No speed encoder input 15 to 20: Speed encoder inputs X15 to X20 80 to 87: Speed encoder inputs X80 to X87
MP113.0	Speed encoder for the first spindle
MP113.1	Speed encoder for the second spindle
MP121	**Nominal speed command output of the spindle/spindles**
Input:	0: No servo-controlled spindle 1 to 6: Analog speed command outputs 1 to 6 (X8) 7 to 12: Analog speed command outputs 7 to 12 (X9) 51 to 64: Digital speed command outputs X51 to X64
MP121.0	Nominal speed command output of the first spindle
MP121.1	Nominal speed command output of the second spindle
MP131	**Y index of the machine parameters MP2xxx.y for the spindle(s) in operating mode 0**
Input:	0 to 17
MP131.0	Y index of first spindle
MP131.1	Y index of second spindle
MP132	**Y index of the machine parameters MP2xxx.y for the spindle(s) in operating mode 1**
Input:	0 to 17
MP132.0	Y index of first spindle
MP132.1	Y index of second spindle

6.1.6 Reading Axis Information

Module 9038 Reading general axis information

With Module 9038 you can interrogate the general status information of the axes. You can interrogate the status of a specific axis or of all axes at once. Bits 0 to 8 represent axes 1 to 9. Bit 15 represents the spindle. If status information is read for only one axis, only bit 0 is changed. The following table shows the meanings of the return codes:

Status information	Meaning
0	0: Axis (spindle) not active (MP10 or MP3010 or no encoder) 1: Axis (spindle) active
1	Depending on the current traverse range: 0: NC axis or not active 1: PLC axis
2	0: No closed-loop axis (spindle), only display or not active 1: Closed-loop axis (spindle)
3	Maximum temperature of the motor [°C]
4	0: Not a Hirth axis 1: Hirth axis (MP420)
5	Hirth grid [1/10 µm] (MP430)
6	Modulo value (MP810)
7	0: Linear axis or not active 1: Rotary axis in at least one of the traverse ranges
8	0: Analog axis (spindle) or not active 1: Digital axis (spindle)
9	0: Not a slave axis 1: Slave axis

Call:

```
PS    B/W/D/K  <>Axis>
               Axis-specific: 0 to 13 represent axes 1 to 14,
                  15 represents the spindle
               Bit-encoded output for all axes: –1
PS    B/W/D/K  <>Status information>
               See table above
CM    9038
PL    B/W/D    <>Information>
```

Error recognition:

Marker	Value	Meaning
M4203	0	Information was read
	1	Error code in W1022
W1022	1	Status information not available on this iTNC
	2	Axis not found

Current tool axis

You can define the current tool axis in two ways in the NC block:

- In the HEIDENHAIN conversational dialog with TOOL CALL
- In ISO programming with G17 to G20

In the PLC you can interrogate the current tool axis via markers (only axis 0 to 9) or via module: This information is only available after each tool call with "TOOL CALL," and is then updated after each tool call.

		Set	Reset
M4526	**Axis 1 is tool axis**	NC	NC
M4527	**Axis 2 is tool axis**	NC	NC
M4528	**Axis 3 is tool axis**	NC	NC
M4529	**Axis 4 is tool axis**	NC	NC
M4530	**Axis 5 is tool axis**	NC	NC
M4531	**Axis 6 is tool axis**	NC	NC
M4532	**Axis 7 is tool axis**	NC	NC
M4533	**Axis 8 is tool axis**	NC	NC
M4534	**Axis 9 is tool axis**	NC	NC

Module 9035 Reading status information (current tool axis)

Call:

PS B/W/D/K <>100>
CM 9035
PL B/W/D <>Current tool axis>
0 to n: Axes 1 to n+1
–1: No information exists about the current tool axis

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid

6.1.7 Traverse Ranges

You can divide the working range of the machine into three traverse ranges, e.g. one for each workpiece. Each traverse range is limited by a software limit switch.

For the software limit switch of a traverse range:

- The datum is the machine datum (MP960.x).
- Software limit switches for tilting axes must be activated with MP812 when MP810.x ≠ 0
- The traverse range can be limited further through the MOD function.
- If a software limit switch is activated, the error message **LIMIT SWITCH <AXIS>** appears.
- Software limit switches can be overwritten with **FN17:SYSWRITE,** e.g. for automatic tool change. This function is effective only until the next GOTO command (GOTO key or FN9 to FN12) or the end of the program.

Determining range of traverse

▶ You can determine the current range of traverse with Module 9035

Module 9035 Read status information

Call:

```
PS    B/W/D/K   <>27>
CM    9035
PL    B/W/D     <>Range of traverse>
                 0 to 2: Traverse ranges 1 to 3
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid
	20	Call was not in a submit or spawn job

Selecting the traverse range

You can switch the range of traverse in two ways:

- Select the traverse range with Module 9151 or 9152. With Module 9152 you can change the axis display at the same time.

or

- Select the traverse range with M4574 and M4575 according to the table below.
- In all operating modes you must activate the traverse range with strobe marker M4135.

You may only use the traverse range switching function via M4574 and M4575 during an M/S/T/Q strobe in all operating modes (except for **`Manual Operation`** and **`El. Handwheel`**).

M4574	M4575	Traverse range/Datum
0	0	Range 1
1	0	Range 2
0	1	Range 3

If the axis traverse limits are changed by the user (after pressing the MOD key), marker M4624 is set. The marker must then be reset by the PLC program.

		Set	Reset
M4574	**Select the traverse range (with M4575)**	PLC	PLC
M4575	**Select the traverse range (with M4574)**	PLC	PLC
M4135	**Strobe marker for selecting the traverse range**	PLC	NC
M4624	**Strobe marker for selecting the traverse range**	NC	PLC

Module 9150 Axis-specific reading of axis traverse limits

PLC module 9150 enables you to read the axis traverse limits that can be defined (after pressing the MOD key).
If the axis traverse limits are set by the user (after pressing the MOD key), marker 4624 is set. The marker must then be reset by the PLC program.

Condition:

- The default values entered in MP91x.x are the maximum possible values for the axis traverse limits.

Call:

```
PS     B/W/D/K   <>Axis>
PS     B/W/D/K   <>Traverse range>
                  0: Traverse range 1
                  1: Traverse range 2
                  2: Traverse range 3
                 –1: Current traverse range
PS     B/W/D/K   <>Mode>
                  0: Axis traverse limits that were changed with the MOD key
CM     9150
PL     D         <>Positive axis traverse limit>
PL     D         <>Negative axis traverse limit>
```

Error recognition:

Marker	Value	Meaning
M4203	0	Axis traverse limit determined
	1	Error code in W1022
W1022	1	Invalid value for mode
	2	Invalid value for axis or traverse range

Module 9151 Select traverse range and axis designation

With Module 9151 you can select the traverse range and the axis designation in one step. The axis designations in MP100.x are overwritten and cannot be activated until the module has received the axis designation –1, a traverse range has been activated with M4135, MP100.x is edited, or the control has been reset.
When the module is called it sets M4135. After switchover the NC resets M4135.

As of software 340 49x-04, PLC Module 9151 can also be called when the user has stopped the machining process with **`External Stop`**, and has then activated the MANUAL TRAVERSE function with the appropriate soft key. When RESTORE POSITION is activated, the system checks whether the traverse range has been reset to the original value. If this is not the case, an error message is output and the RESTORE POSITION function is not executed. You can, however, cancel the machining process with **`External Stop`** or Emergency Stop.
If the traverse range is switched when the MANUAL TRAVERSE function is active (activated by soft key), an active M128 function is automatically deactivated. After switching back with RESTORE POSITION, M128 is reactivated.

Call:
PS B/W/D/K/S<>String with axis designation>
Format: XYZABCUVWxyzabcuvw
Characters 1 to 9 represent axes 1 to 9
With –1 the axis designations from M100.x are valid
PS B/W/D/K <>Traverse range>
0 to 2: Range of traverse
–1: Do not change range of traverse
CM 9151

Error recognition:

Marker	Value	Meaning
M4203	0	Traverse range/axis designation switched over
	1	Error code in W1022
W1022	2	Invalid value for traverse range
	3	For the axis assignment, neither a string nor –1 was transferred
	21	The module was called during a running part program or without an M/S/T/Q strobe

Module 9152 Selecting traverse range, axis display and axis designation

With Module 9152 you can select the traverse range, the axis designation, and the axis display. The axis designations in MP100.x and the axis display in MP7291 are overwritten and cannot be activated until the module has received –1 for the axis designation and axis display, a traverse range has been activated with M4135, MP100.x or MP7291 have been edited, or the control has been reset.
When the module is called it sets M4135. After switchover the NC resets M4135.

Call:
PS B/W/D/K/S<>String with IV and V key configuration>
Format: AB
The first character represents the IV key, the second character represents the V key
With –1 the key configuration from MP410 is valid
PS B/W/D/K/S<>String with axis display>
Format: SWVUCBAZYXwvucbazyx
Characters 1 to 9 (from the right) represent lines 1 to 9
Character 10 = S always in line 10
With –1 the axis display from MP7291 is valid
PS B/W/D/K/S<>String with axis designation>
Format: XYZABCUVWxyzabcuvw
Characters 1 to 9 represent axes 1 to 9
With –1 the axis designations from M100.x are valid
PS B/W/D/K <>Traverse range>
0 to 2: Range of traverse
–1: Do not change range of traverse
CM 9152

Error recognition:

Marker	Value	Meaning
M4203	0	Traverse range, axis designation and axis display are switched
	1	Error code in W1022
W1022	2	Invalid value for traverse range, or string for axis configuration, axis display or key configuration is too long
	3	For the axis assignment, axis display or key configuration, neither a string nor –1 was transferred
	21	The module was called during a running part program or without an M/S/T/Q strobe

Setting the software limit switches

With the following machine parameters, you can set the software limit switches for the various ranges of traverse. The position values are with respect to the machine datum. Ranges of traverse 2 and 3 do not become effective until they are activated by M4574, M4575 and M4135 or with Module 9151 or 9152.

Note

The values for MP910.x, MP911.x, MP912.x, MP920.x, MP921.x and MP922.x can be transferred with the actual-position-capture key.

MP910.x **Positive software limit switches, traverse range 1 (default setting after power on)**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP911.x **Positive software limit switches, traverse range 2**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP912.x **Positive software limit switches, traverse range 3**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP920.x **Negative software limit switches, traverse range 1 (default setting after power on)**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP921.x **Negative software limit switches, traverse range 2**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP922.x **Negative software limit switches, traverse range 3**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP7490 **Functions for traverse ranges**
Format: %xxxx
Input: Bit 0 = 0: Display one traverse range via MOD
Bit 0 = 1: Display three traverse ranges via MOD
Bit 1 = 0: Each traverse range has its own datum
(and 3 memories for the positions of the swivel head)
Bit 1 = 1: One datum for all traverse ranges

The NC reports the activation of limit switches to the PLC in words W1034 and W1036:

		Set	Reset
W1034	**Positive software limit switch was approached** Bits 0 to 13 represent axes 1 to 14	NC	NC
W1036	**Negative software limit switch was approached** Bits 0 to 13 represent axes 1 to 14	NC	NC

Special function M150

If NC blocks were used in the Positioning with Manual Data Input, Program Run Single Block, and Program Run Full Sequence operating modes to program positions that are outside of the traverse ranges, then normally the blocks containing this violation are not performed, and an error message is output.

With M150 the block is traversed to at least shortly before the limit of the traverse range, despite this programming violation. Positioning is performed as close to the limit of the traverse range as possible. For example, if the limit is –600.000 and the programmed position is –700.000, M150 traverses to –599.999. This means that the limit switch information for the PLC via W1045 and W1036 is not set, since the limit switch is not traversed to.

6.1.8 Lubrication Pulse

You can define the traverse distance for each axis after which the PLC commands lubrication:

- In MP4050.x you define the traverse distance at which the lubrication pulse is to be output. The NC reports in W1056 when the entered distance in an axis has been exceeded.
- With W1058 you reset the distance counter to 0 after lubrication.

The summation of the path traversed always occurs in MP4050.x, regardless of the operating mode and how the axis was moved.

MP4050.0-8 Path-dependent lubrication of axes 1 to 9
Input: 0 to 99 999.999 [m or 1000°]

		Set	Reset
W1056	**Lubrication pulse: Value in MP4050.x exceeded** Bits 0 to 8 represent axes 1 to 9	NC	NC
W1058	**Reset the accumulated distance** Bits 0 to 8 represent axes 1 to 9	PLC	PLC

6.2 PLC axes

You can assign the controlled axes individually to the PLC.

Remember that:

- PLC axes can be operated with following error (also called lag) or with velocity feedforward control. The axis-specific jerk (MP1097.x and MP1098.x) is accounted for.
- You can start more than one axis simultaneously. However, the axes are not interpolated with each other.
- PLC axes are positioned by the shortest path if you enter a modulo value in MP810.x.
- Up to 20 commands (such as positioning, override settings, etc.) for PLC axes can be executed per run-through of the PLC program.

With MP100.x you define for every traverse range which axes the PLC controls and which the NC controls. Uppercase letters represent NC axes, and lowercase letters represent PLC axes. To indicate axes that are not present, mark them with a hyphen (-).
Furthermore, based on the settings in MP810.x, the control detects whether the PLC axis is a rotary axis.

MP100	**Designation of axes**
Format:	XYZABCUVWxyzabcuvw-
Input:	Bits 0 to 8 represent axes 1 to 9
MP100.0	Traverse range 1
MP100.1	Traverse range 2
MP100.2	Traverse range 3

Module 9120 Starting a PLC axis

This module starts the positioning of a PLC axis regardless of other processes in the control.

Conditions:

- Status changes through a PLC positioning command are not detected until the next PLC cycle.
- The axis must be activated in MP10 and identified in MP100 as a PLC axis.
- Traverse over the software limit switches is not checked.
- The axis must be stationary before positioning. Interrupt a running positioning movement with Module 9121.
- Feed-rate override is disabled. To change the feed rate, use Module 9124.
- If no reference mark has been traversed, the positioning process builds on the counter value as it was upon switch-on.

Call:

PS	B/W/D/K	<>Axis>
		0 to 8 represent axes 1 to 9
PS	B/W/D/K	<>Target position>
		Input unit: [0.0001 mm]
PS	B/W/D/K	<>Feed rate>
		Input unit: [mm/min]
PS	B/W/D/K	<>Mode>
		Bit 0: Type of target position input
		0: Absolute, i.e. relative to the machine datum
		1: Incremental
CM	9120	
PL	B/W/D	<>Error code>
		0: No error. Positioning was started.
		1: Axis does not exist
		2: Not a PLC axis
		3: Axis is already being positioned
		4: Absolute position is outside of modulo range
		5: Programmed axis not in closed loop
		6: Feed rate not permitted

Module 9121 Stopping a PLC axis

Stops a running PLC positioning process in an axis.

Condition:

- Status changes through a PLC positioning command are not detected until the next PLC cycle.

Call:

PS	B/W/D/K	<>Axis>
		0 to 8 represent axes 1 to 9
CM	9121	
PL	B/W/D	<>Error code>
		0: Positioning is canceled
		1: Axis does not exist
		2: Not a PLC axis
		3: Axis was already stationary

Module 9122 Status of PLC axis

Request for PLC positioning status.

Condition:

- Status changes through the PLC positioning command are not detected until the next PLC scan.

Call:

PS	B/W/D/K	<>Axis> 0 to 8 represent axes 1 to 9
CM	9122	
PL	B/W/D	<>Status> Bit 0 – A PLC axis? 0: NC axis or not active 1: PLC axis Bit 1 – Reference mark 0: Reference mark not yet traversed 1: Reference mark traversed Bit 2 – Positioning 0: Inactive 1: Active Bit 3 – Direction of motion 0: Positive 1: Negative Bit 4 – Positioning error 0: No positioning errors occurred 1: Positioning error Bit 5 – Closed-loop or open-loop axis 0: Closed-loop axis was programmed 1: Axis programmed which was switched to open-loop in Module 9155 Bit 6 – Target position reached? 0: Target position not yet reached 1: Target position reached

Module 9123 Traverse the reference marks of PLC axes

Traverse the reference marks as for NC axes. This module is not suitable for encoders with distance-coded reference marks (use Module 9220).

- You can use the same procedure to traverse a reference mark for PLC axes as for NC axes. Use Module 9123 only if no conventional procedure is possible.
- Module 9123 moves the axis in the given direction until the reference mark has been traversed. The axis stops next to the reference mark, offset by the braking path.

Call:

PS	B/W/D/K	<>Axis> 0 to 8 represent axes 1 to 9
PS	B/W/D/K	<>Feed rate> Input unit: [mm/min]
PS	B/W/D/K	<>Mode> Bit 0: Direction of traverse 0: Positive 1: Negative
CM	9123	
PL	B/W/D	<>Error code> 0: No error. Positioning was started. 1: Axis does not exist 2: Not a PLC axis 3: Axis is already being positioned 5: Programmed axis not in closed loop 6: Feed rate not permitted

Module 9124 Feed rate override for PLC axis

Enter a default feed-rate override for a PLC axis if positioning was started with Module 9120 or 9123.

Conditions:

- After interruption of a PLC program, the override value is set to 100%.
- When a positioning is started, the last defined override value is in effect.
- The override value can also be changed during a positioning movement.

Call:

PS	B/W/D/K	<>Axis> 0 to 8 represent axes 1 to 9
PS	B/W/D/K	<>Override> Input unit: 0 to 10 000, corresponds to 0 to 100% in 0.01% steps.
CM	9124	
PL	B/W/D	<>Error code> 0: No error, override value was set 1: Axis does not exist 2: Not a PLC axis 3: Override value incorrect

Module 9125 Stop PLC axis at next Hirth grid position

Stop an already started PLC-positioning of an axis at the next Hirth grid position.

Call:

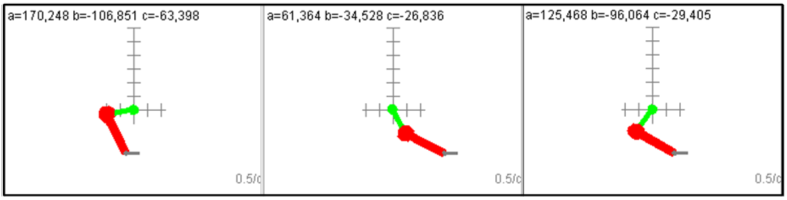
PS	B/W/D/K	<>Axis> 0 to 8 represent axes 1 to 9
CM	9125	
PL	B/W/D	<>Error code> 0: Positioning is canceled 1: Axis does not exist 2: Not a PLC axis 3: Axis was already stationary 4: Axis is not a Hirth axis (MP420.x)

6.2.1 Coupling function for PLC axes or auxiliary axes

The coupling function makes it possible to control PLC axes in dependency on other PLC axes or an NC axis. This enables you to realize complex simultaneous movements of several PLC axes. The coupling function is activated and deactivated through PLC module 9228. PLC module 9229 supplies the status information on the coupling functions.

The mutual dependence of the axes must be defined in mathematical formulas. The following example explains how to proceed:

A robot arm consists of three limbs, each of which can be controlled as three PLC axes. The arm is to move only in such a way that the third limb and its joint (shown in gray) move horizontally at a constant height.



The example is based on the following assumptions:

- At the 0° position the arm extends horizontally to the right and all joints are extended.
- The length of the second arm (marked in red) is 1.6 times the length of the first arm (marked in green).
- The third limb and its joint (shown in gray) must move horizontally at a constant height.
- a, b, c: Angles of the respective joints
- ma, mb, mc: Length of the respective components

The following condition must be fulfilled for the third joint to remain at a constant height h:

- Height of the first joint = 0
- Inclination of the first arm (green) = a
- Height of the second joint = sin(a) * ma
- Inclination of the second arm (red) = a + b
- Height of the third joint = sin(a) * ma + sin(a+b) *mb
- This results in height h = sin(a) * ma + sin(a+b) * mb

To determine the angle b as a function of a, you need the definition of b, which is derived from the formula for h as follows:

- h = sin(a) * ma + sin(a+b) * mb
- b= -a + asin((h/mb - sin(a) *ma/mb))

To keep the third arm (gray) horizontal, a + b + c must be equal to 0 (a + b + c = 0). Therefore:

- c = -a - b
- and: c = -asin ((h/mb - sin(a) * ma/mb))

Realization

For the coupling functions (PLC module 9228), the resolution of the nominal positions in the PLC module is 0.0001°. The trigonometry functions sin(), cos(), tan(), etc. require data input in radians (360° = 2*pi).

The inverse trigonometry functions supply the result in radians. To obtain the radian values from the nominal positions in internal-resolution form, the position value of the axis must be multiplied by pi / 1800000 or in shorter form by pi/18e5.

Hence, the value of an inverse trigonometry function must be multiplied by 18e5/pi to obtain the internal coordinate resolution.

The constants h , ma and mb result from the given mechanical design.

The example using the values h = 1.6 , ma=1, mb=1.6 results in the synchronous function being called in the PLC program (the error handling is missing here) as follows:

```
LBL MF49     ;activate the coupling function
L S" -a + asin(1 - 1/1.6*sin(a*pi/18e5))*18e5/pi"
= S0
PS K0        ;string number
PS K4        ;axis
PS K1        ;mode: Switch-on
CM 9228
PLX

;coupling function for additional picker axis
L S" - asin(1 - 1/1.6*sin(a*pi/18e5))*18e5/pi "
= S0
PS K0        ;string number
PS K5        ;axis
PS K1        ;mode: Switch-on
CM 9228
PLX

EM
```

The following process must be realized by the PLC program:

- Positioning the PLC axis to a suitable starting position
- Activating the coupling function
- Moving the first PLC axis appropriately (by means of positioning movements through the PLC), the other PLC axes are automatically corrected.
- Deactivating the coupling function when the respective positioning movement has been completed

Module 9228 Coupling function for auxiliary axes

Module 9228 is used to enable/disable the coupling of auxiliary axes with other auxiliary or NC axes for positioning movements. The resulting motions must be defined in a formula (coupling function) for each coupled axis. This formula is to be transmitted to the module as a PLC string.

Condition:

- The arithmetic operations and trigonometric functions that can be used in a formula are described in "Special case: Entering a formula" in the Machine Parameters Chapter in the Technical Manual.
- The coupling-function data is not forwarded until the PLC cycle has been completed.
- The status of a coupling function (readable with Module 9229) is available in the subsequent PLC cycle at the earliest.
- The coupling function can only be activated at standstill.
- The axes must initially be positioned in such a way that the activation of a formula does not result in a nominal-value step. Module 9229 (mode 1, current nominal position with the help of a formula) is used to determine whether the initial position of a coupling axis must be changed.
- For the coupling functions (PLC module 9228), the resolution of the nominal positions in the PLC module is 0.0001° or 0.0001 mm. The trigonometry functions sin(), cos(), tan(), etc. require data input in radians (360° = 2*pi). The inverse trigonometry functions supply the result in radians. To obtain the radian values from the nominal positions in internal-resolution form, the position value of the axis must be multiplied by pi / 1800000 or in shorter form by pi/18e5. Hence, the value of an inverse trigonometry function must be multiplied by 18e5/pi to obtain the internal coordinate resolution.
- The evaluation of W1022 is not necessary, because the error codes in the return value make a detailed evaluation possible (see call interface).

```
Call:
PS    K/B/W/D/S<>Coupling function>
PS    K/B/W/D   <>Coupling axis>
PS    K/B/W/D   <>Mode>
                Bit 0: Coupling function not active
                Bit 1: Coupling function active
CM    9228
PL    B/W/D     <>Error codes>
                0: Coupling function is activated/deactivated
                1: Invalid axis programmed (invalid axis number, no auxiliary
                   axis, slave axis)
                2: Invalid mode programmed
                3: Axis has not yet been homed
                4: Invalid PLC string programmed as formula
                5: Programmed axis not in closed loop
                6: Auxiliary-axis positioning (Module 912x) or free rotation
                   (Module 9223) active
                7: Coupling function is already active for programmed axis
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022 (also see return values of the module)
W1022	1	Invalid mode programmed
	2	Invalid axis or open-loop axis programmed, positioning of auxiliary axes (Module 912x) or free rotation (Module 9223) active
	11	Invalid PLC string programmed as formula
	28	Coupling function is already active for programmed axis
	33	Axis not yet referenced

Module 9229 Status of coupling function

Module 9229 is used to determine the status and error conditions of Module 9228 (coupling functions). The module can calculate the current position value to avoid nominal-position steps when axes are coupled, and to be able to pre-position the coupling axes appropriately.

Condition:

- The status or error code will be available no sooner than during the next PLC cycle after PLC module 9228 has been called. (Exception: status/error 4, error code in W1022)

Call:

PS K/B/W/D <>Coupling function>
The parameter is evaluated only if mode = 1.
PS K/B/W/D <>Coupling axis>
The parameter is evaluated only if mode = 0.
PS K/B/W/D <>Mode>
0: Status/Error feedback, possible states/errors: 0,1, 2, 3, 4 after Module 9228 has been called.
1: One-time position calculation (only possible in submit/ spawn), possible errors: 3, 4
CM 9229
PL B/W/D <>Status/Error>
If the status/error is 0 to 3, M4203 is not set:
0: No coupling function active for programmed axis
1: Coupling function is active for programmed axis
2: Coupling function canceled (see additional info)
3: Syntax error in coupling function of Module 9228 or 9229 (see additional info)
4: See error codes in W1022
PL B/W/D <>Additional info>
Mode 0:
If additional info on status/error = 2 (cancelation):
0: Canceled by emergency stop
1: Canceled due to computing error during run time
If additional info on status/error = 3 (syntax error):
0: Not yet supported
Mode 1:
Position value or additional info in the event of an error: 0x80000000 (largest negative value)

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022 (also see error code above)
W1022	1	Invalid mode programmed
	2	Invalid axis number programmed
	11	Invalid PLC string programmed as formula
	20	Call from the cyclic PLC program
	45	Canceled due to error message from operating system

6.3 PLC Positioning

You can position the axes directly through the PLC. For PLC positioning of the main spindle, see page 941.

Prerequisites

The following constraints apply to a PLC positioning command:

- It is possible in the Manual and Handwheel modes only while there is no positioning movement.
- Possible in the other modes of operation only with an M/S/T/Q strobe or if no part program is started.
- If the NC is positioning an axis, you can position additional axes only if they have already been defined as PLC axes. “PLC axes" on page 669.

Programming

You start a PLC positioning movement with Module 9221 or 9227, interrogate the status with Module 9222, and stop it ahead of time with Module 9224. After Module 9221 or 9227 has been called, markers M4120 to M4128 are set. If you reset these markers directly or program Module 9224 accordingly, positioning is canceled at the desired point in time. This is necessary if you would like to change a parameter, such as the feed rate, during positioning.

The following conditions apply to a PLC positioning command:

- If more than one axis is moved simultaneously, the axes will be interpolated.
- If you start another axis during a PLC positioning movement,
  - the first positioning command will be canceled and
  - the resulting positioning movement will be executed in all axes.
- Tool compensation is not included. Before a PLC positioning command you must end any tool compensation.
- A PLC positioning movement is not displayed in the test graphics.
- PLC positioning is done without nominal position value filter.

The NC cancels a PLC positioning movement under the following conditions:

- If in the Manual or Handwheel modes there is an NC STOP
- If in the automatic operating modes there is an NC STOP and "internal stop."
- At an EMERGENCY STOP.
- An error message that results in a STOP
- A reset of Markers M4120 to M4128
- Stopping of the positioning with Module 9224 (all axes)

Module 9221 Starting a PLC positioning movement

Starts a PLC positioning movement in one axis.
As of software 340 49x-04 you can use Module 9221 to deactivate collision monitoring for a specific PLC positioning movement. This means that it is also possible to position an axis when DCM is active and the reference marks have not yet been traversed in all axes.
A PLC positioning command can only be executed if collision monitoring is deactivated for all axes involved in the positioning movement. The deactivation of collision monitoring for the PLC positioning movement does not affect the status information provided by Module 9064.

Call:

PS	B/W/D/K	<>Axis> 0 to 13 represent axes 1 to 14
PS	B/W/D/K	<>Target position> Input unit: [0.0001 mm]
PS	B/W/D/K	<>Feed rate> Input unit: mm/min
PS	B/W/D/K	<>Mode> Bit 0 – Definition of the target position 0: Absolute, i.e. relative to the machine datum 1: Incremental Bit 1 – Software limit switches 0: Inactive 1: Active Bit 2 – Reserved Bit 3 – PLC positioning with collision monitoring deactivated
CM	9221	
PL	B/W/D	<>Error code> 0: Positioning is being started 1: Axis is not in a closed loop or is an auxiliary axis 2: Inadmissible values for the feed rate 3: Axis has not traversed the reference mark 4: No M/S/T/Q strobe during started part program 5: Programmed axis not in closed loop 6: PLC positioning already started

Module 9222 Status request of PLC positioning movement

With this module you can interrogate the status of a PLC positioning movement for an individual axis or bit-encoded.

The status of an axis or all axes remains until a new status is set when the next PLC positioning of one or more axes occurs.

Call:

PS	B/W/DK	<>Axis> **Interrogation of an axis:** 0 to 13 represent axes 1 to 14 **Interrogation of all axes:** -1: Target position reached -2: PLC positioning was started -3: PLC positioning was cancelled -4: PLC positioning reached limit switch -5: PLC positioning not possible -6: PLC positioning temporarily stopped -7: PLC positioning not started due to collision monitoring
CM	9222	
PL	B/W/D	<>Status> **Interrogation of an axis:** 0: No PLC positioning was started 1: Target position reached 2: PLC positioning was started 3: Due to cancellation, target position not reached 4: Target position is outside of traverse range 5: Positioning not possible (e.g. due to "free rotation") 6: Positioning temporarily stopped (e.g. stop in Automatic operating modes) 7: Positioning not started due to collision monitoring **Interrogation of all axes:** Status of PLC positioning movement bit-encoded in word format

Error recognition:

Marker	Value	Meaning
M4203	0	Status was transferred
	1	Axis does not exist or status interrogation faulty
W1022	1	Invalid status information was requested
	2	This status of an open-loop axis, auxiliary axis or slave axis is being interrogated

Module 9224 Stop PLC positioning movements

Individual PLC positioning movements can be stopped with Module 9224.

- If M4120 to M4128 are already set, they are reset.
- It is still possible to stop PLC positioning movements by resetting markers M4120 to M4128.
- Positioning movements of axes 10 to 14 can be stopped only with this module.

Call:

PS	B/W/D/K	<>Axis>
PS	B/W/D/K	<>Mode> Reserved: 0 transferred
CM	9224	
PL	B/W/D	<>Error> 0: Positioning is canceled 1: Invalid axis was programmed 2: This axis is not positioned by the PLC

Error recognition:

Marker	Value	Meaning
M4203	0	PLC positioning was stopped
	1	Error code in W1022
W1022	2	Invalid axis (invalid axis number, slave axis, auxiliary axis, or uncontrolled axis)
	9	Axis is not being positioned by the PLC

Module 9227 Positioning auxiliary axes and NC axes

The module starts the positioning of an NC axis (PLC positioning like with Module 9221) or the positioning of an auxiliary axis (like with PLC Module 9120). With Module 9227, acceleration and jerk can be programmed in addition to the target position and feed rate parameters.

Further modes can be programmed for positioning NC axes (like with Module 9221):

- Rapid-traverse feed rate (only for operating panels with rapid traverse override)
- Active limit switch interrogations
- Deactivated collision monitoring

See also documentation for PLC Modules 9120 and 9221

Conditions:

- The machine parameter configuration of an axis to be started with Module 9227 determines whether the positioning movement of an auxiliary axis or an NC axis is started.

Positioning of NC axes (PLC positioning):

- The same constraints apply as for starting a PLC positioning movement with Marker 4120, etc. The module may only be called if no program is running, or if a M/G/S/T/T2/Q strobe is pending. No axis direction key may be pressed in the Manual operating mode. The entered positions are referenced to the machine datum. For modulo rotary axes (MP810), positioning is by the shortest path.
- The NC axes can be started by setting the activation markers 4120 to 4128 or with the module call. The module call has priority.
- When the module is called, the corresponding activation marker is set automatically.
- The PLC positioning of this axis can be interrupted by resetting an activation marker.
- If you wish to change a parameter (e.g. target position, feed rate) of a positioning command already in progress, you must first abort positioning, then change the parameter and start again.
- A simultaneous PLC positioning movement of several axes is interpolated. If you start an additional axis while already positioning another, the first movement is aborted, and then all the programmed axes (e.g. X, Y and Z) are positioned together.
- As soon as a PLC positioning with rapid traverse is active (bit 2 is set), all active PLC positioning movements are at rapid traverse, and instead of the feed-rate override the rapid-traverse override is effective.
- A deactivation of collision monitoring (bit 3) only has an effect, if all axes positioned by the PLC at the same time were started with collision monitoring inactive.
- The deactivation of collision monitoring (bit 3) for the PLC positioning movement does not affect the status information provided by Module 9064 (collision monitoring status). With active DCM the module still reads "Monitoring active".
- Error code 7 not possible.

Positioning auxiliary axes:

- Axes with automatic reduction (modulo value in MP810.x) are always moved to the target position in the direction of the shortest traverse, unless the target position was given as an incremental value.
- The system does not check for limit switch overshoot.
- The axis must be stationary. Any positioning movement must be interrupted beforehand with Module 9121.
- The feed-rate override is not offset.
- If the axis had been in the "Search for reference mark" state, this state is canceled. The positioning movement always starts from the current counter value.
- If modules 9120, 9227 (Position PLC Axis), 9121 (Stop PLC Axis) and 9122 (PLC Axis Status Request) are called several times for the same axis during a PLC scan, only the latest command is followed.
- Error codes 3 and 4 not possible

Call:

PS	B/W/D/K	<>Axis>
PS	B/W/D/K	<>Target position/Increment> in [0.0001mm], ref system
PS	B/W/D/K	<>Feed rate> in [mm/min]
PS	B/W/D/K	<>Acceleration> in [mm/s^2] 0: Value from MP1060 is used
PS	B/W/D/K	<>Jerk> in [mm/s^3] 0: Value from MP1088 is used
PS	B/W/D/K	<>Mode> NC and auxiliary axes: Bit 0 = 0: Absolute positioning Bit 0 = 1: Incremental positioning Only NC axes: Bit 1 = 1: Software limit switch active Bit 2 = 1: Rapid traverse override effective Bit 3 = 1: DCM collision monitoring is deactivated
CM	9227	
PL	B/W/D	<>Status> 0: Function performed 1: Illegal group number 2: Incorrect parameterization via bit mask 5: Axis is not in a closed loop (Module 9155) 6: Axis is already being positioned 20: Module was not called in a spawn job or submit job

Error recognition:

Marker	Value	Meaning
M4203	0	Function performed
	1	Error code in W1022
W1022	1	Illegal group number
	2	Invalid value for bit mask
	20	Module was not called in a spawn job or submit job

PLC positioning through markers and words

To ensure compatibility, a PLC positioning command is permissible for axes 1 to 9 with M4120 to M4128, D528 to D544 and W560 to W568.

> Attention
>
> Software limit switches are ignored!

Programming:

- Enter the target position in the double words D528 to D544 in the unit [0.0001 mm].
- Enter the feed rate in words W560 to W568 [mm/min].
- To start the PLC positioning movement: Set markers M4120 to M4124 for the desired axis.

		Set	Reset
D528-544	**Target position for PLC positioning**	PLC	PLC
W560-568	**Feed rate for PLC positioning**	PLC	PLC
M4120	**PLC positioning axis 1 active**	NC/PLC	NC/PLC
M4121	**PLC positioning axis 2 active**	NC/PLC	NC/PLC
M4122	**PLC positioning axis 3 active**	NC/PLC	NC/PLC
M4123	**PLC positioning axis 4 active**	NC/PLC	NC/PLC
M4124	**PLC positioning axis 5 active**	NC/PLC	NC/PLC
M4125	**PLC positioning axis 6 active**	NC/PLC	NC/PLC
M4126	**PLC positioning axis 7 active**	NC/PLC	NC/PLC
M4127	**PLC positioning axis 8 active**	NC/PLC	NC/PLC
M4128	**PLC positioning axis 9 active**	NC/PLC	NC/PLC

6.4 Axis Error Compensation

The iTNC can compensate the following mechanical axis errors:

- Backlash
- Linear axis errors
- Nonlinear axis errors
- Thermal expansion
- Reversal peaks during circular movements
- Stick-slip friction
- Torsion compensation

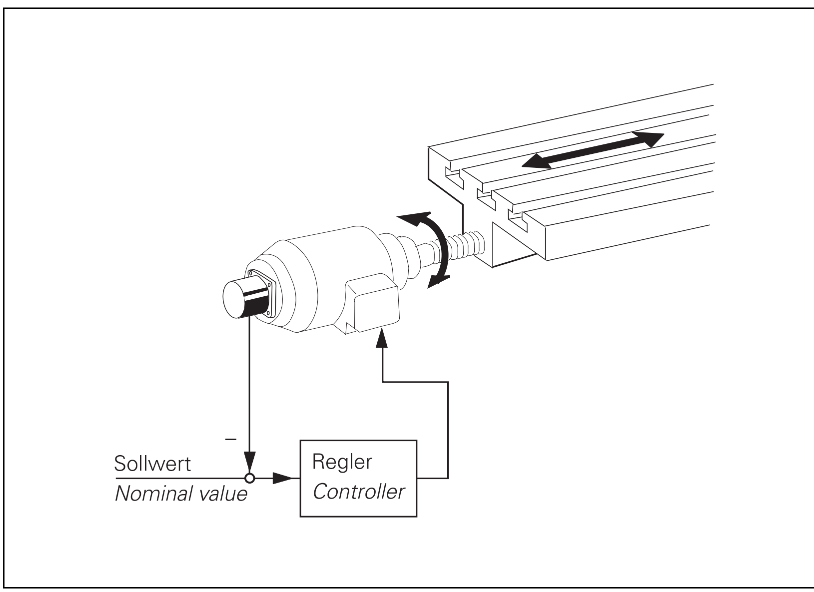
Per axis you can activate either the linear or the nonlinear axis error compensation. All other types of compensation are nonexclusive.

6.4.1 Backlash compensation

Cause outside of the control loop

During a reversal in axis direction, there is often a little play between the rotary encoder and table. This play is referred to as backlash.

Positive backlash: The rotary encoder reading is ahead of the table. The table traverse is too short.
Negative backlash: The rotary encoder reading is behind the table. The table traverse is too long.

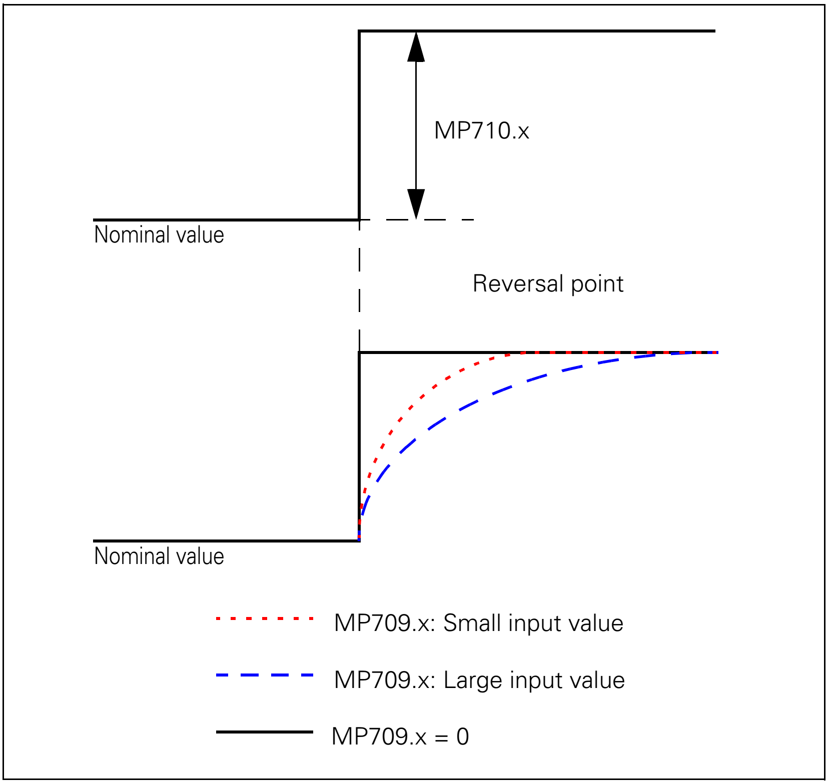


Compensation:

- Enter the backlash in MP710.x.
- In MP709.x, enter the time in which the distance to be compensated should be traversed.

The value of the backlash is added to the position value at every reversal of direction (even if it results from a nonlinear axis-error compensation, for example) and corrected by the position controller. The value of the k_V factor therefore influences the settling time for backlash compensation.

In special cases, such as long ball screws with narrow diameters, additional tension in the system can lead to undesirable dynamic side effects. In order to compensate for this tension at the reversal point, enter in MP709.x the time in which the distance to be compensated (from MP710.x) should be traversed.



Acceleration-dependent backlash compensation

In the backlash compensation, the time in which the distance to be compensated should be traversed is entered in MP709.x. The time entered is a fixed value and is used independently of axis acceleration.

Acceleration-dependent backlash compensation that is entered in machine parameter MP708.x is a different feature. It reduces the following error after direction reversal that is independent of speed, acceleration and the traversed contour path. In acceleration-dependent backlash compensation, you define a distance within which the backlash is compensated. The resulting duration for backlash compensation therefore depends on the current acceleration. Backlash compensation takes longer at low than at high acceleration.

In the machine parameter MP708.x, you enter the distance within which the backlash is to be compensated. You also need to enter the backlash in MP710.x or in the compensation table for variable backlash. In MP709.x, you need to enter the value 0 to activate acceleration-dependent backlash compensation.
MP708.x is adjusted by incrementally changing the machine parameter value in the circular interpolation test (reversal peaks).

Acceleration-dependent backlash compensation

- Enter the backlash in MP710.x.
- Enter the input value 0 in MP709.x.
- In MP708.x, enter the traverse distance within which the backlash is to be compensated.

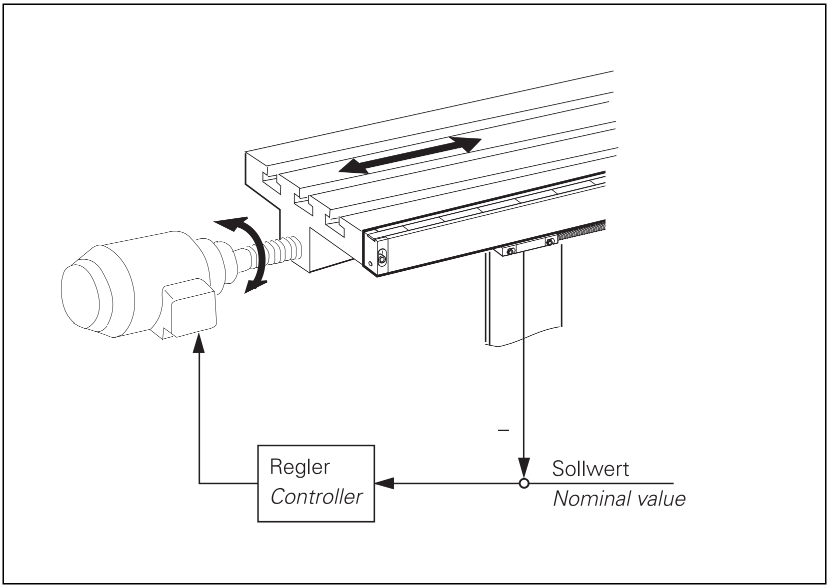
MP710.x **Backlash compensation**
Input: -9.9999 to +9.9999 [mm] or [°]

MP709.x **Time constant for backlash compensation**
Input: 0: Previous behavior of MP710.x or MP708.x is active
1 to 1000 [ms]

MP708.x **Traverse distance for acceleration-dependent backlash compensation**
Input: 0: Function inactive
0.0001 to 1.000 [mm]

Cause within the control loop

If axis movement is measured with a linear encoder, the iTNC can compensate the play between the motor and the table. At the same time, the reversal spikes during circular movements are compensated: machine parameters MP711 to MP716 for "Compensation of reversal spikes" are **not** necessary.



Compensation:

- In MP750.x, enter the reversal error in mm.
- In MP752.x, enter the time in which the distance to be compensated should be traversed.

MP750.x	**Backlash in axes 1 to 9**
Input:	-9.9999 to +9.9999 [mm] or [°]
MP752.x	**Compensation time for reversal error**
Input:	0 to 1000 [ms]

Example:

MP750.x: 0.03 mm
MP752.x: 15 ms

For every change in direction, a nominal speed command signal is output for 15 ms, which corresponds to a feed rate of 120 mm/min:

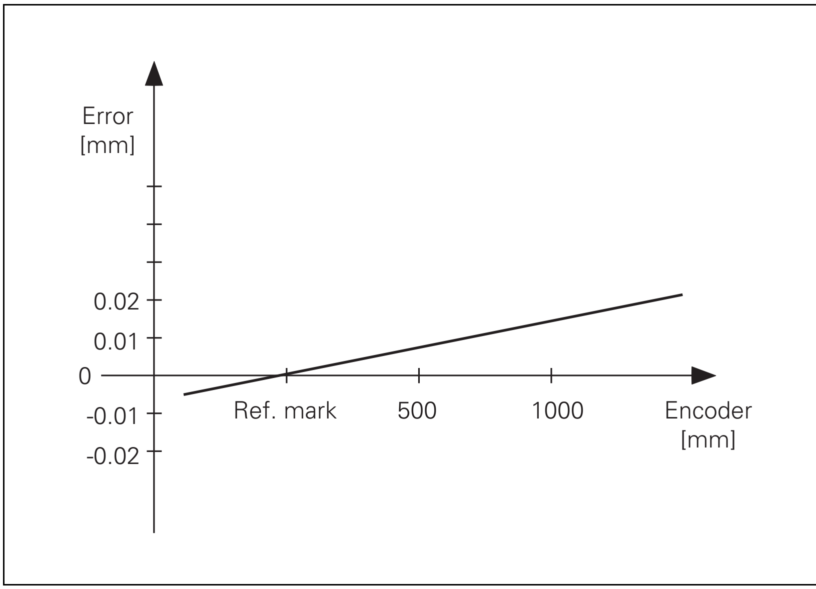
$$\frac{0.03\ \text{mm}}{15\ \text{ms}} = 120\ \text{mm/min}$$

6.4.2 Linear axis error compensation

> **Note**
>
> Linear axis error compensation is not available for rotary axes!

For every linear axis you can compensate a linear axis error.

Positive linear axis error: The table moves too far.
Negative linear axis error: The table moves short.



Compensation:

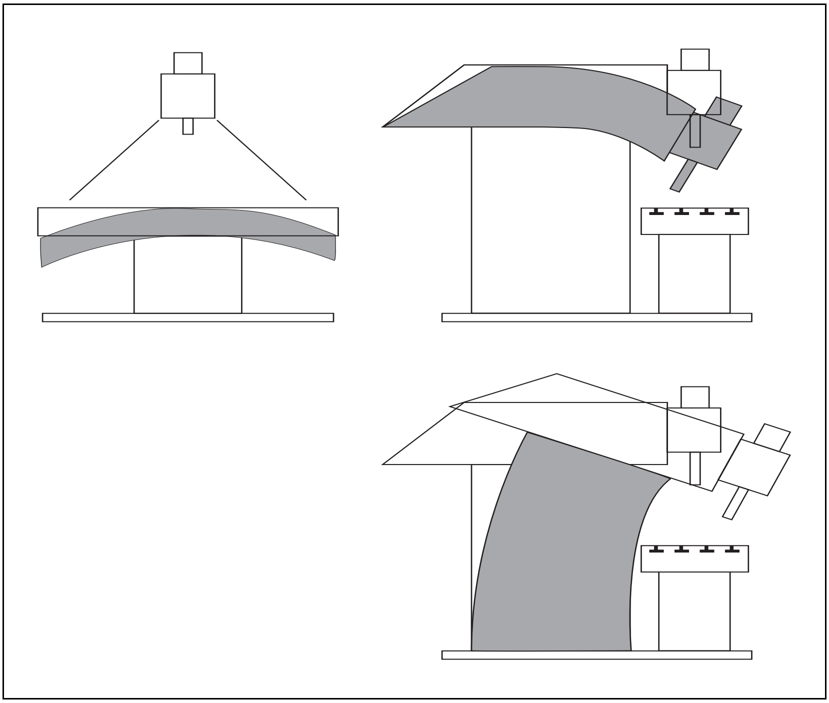
- In MP720, enter the axis error in [mm/m].
- With MP730, activate the linear axis error compensation.

MP720.x **Linear axis error compensation**
Input: –1.000 to +1.000 [mm/m]

MP730 **Selection of linear/nonlinear axis error compensation**
Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Linear axis error compensation
1: Nonlinear axis error compensation

6.4.3 Nonlinear axis error compensation

Errors in machine geometry (e.g. an error in one axis caused by the sagging of another axis) or external influences (e.g. temperature) can cause nonlinear axis errors. These graphics show typical nonlinear axis errors:

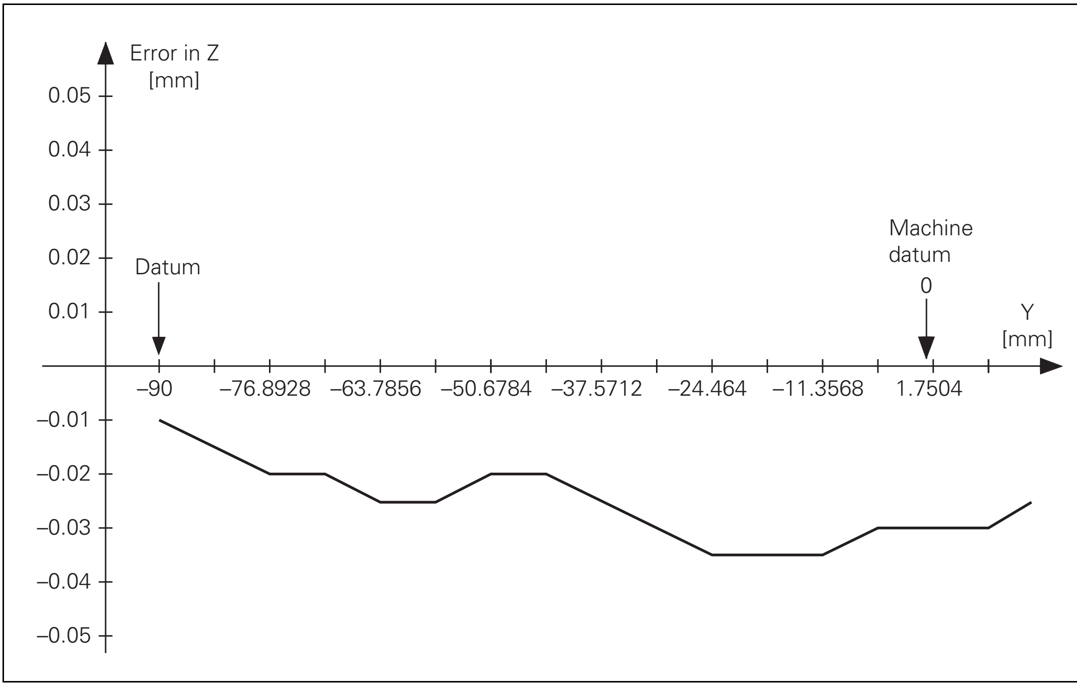


The best way to measure nonlinear axis error is with a comparator measuring system such as the HEIDENHAIN VM 101.

> **Note**
>
> The iTNC can compensate screw-pitch error and axis sag simultaneously.

Nonlinear axis error compensation is also effective for an open loop. In this case the compensation value is considered when the control loop is closed.

The following graphic shows the trace of an axis sag error as a function of Y (Z = f(Y)):



Entering the error curve

To enter the error trace in the iTNC:

- Ascertain the error trace with a comparator measuring system.
- To create a compensation-value table: Press the MOD key and enter the code number 807667; for each axis that is to be compensated for, use the program manager to create a compensation-value table with the name *.COM.
- Using soft keys (e.g. **1 OFF/ON**), activate columns in the compensation value table only for the axes whose positions affect the error of the compensated axis.
- Begin your entry with the soft key **HEAD LINE**: Enter the datum for the compensation values as a distance from the machine datum (MP960.x).
- Enter a value for the spacing of the compensation points as a power to the base of 2. The input value is ±999.9999.
- Exit the header by pressing END.
- With the soft key **APPEND N LINES,** enter the number of compensation points:
  - Up to 4000 lines
  - Maximum of 24 columns in all active compensation value tables
  - Total maximum of 24 · 4000 = 96000 compensation points
- To enter compensation values: Enter only the break points of the error trace. The iTNC interpolates linearly between the break points.

If another *.COM file is to be used for the negative direction of traverse, the file name must end with a minus character (–). The difference between two direction-sensitive compensation-value tables is corrected immediately at a reversal of direction.

Example:
Entry in the *.CMA file: **Axis_X**. The compensation value table Axis_X.COM is used. If the compensation value table Axis_X–.COM exists, it will be used for the negative traverse direction.

Example

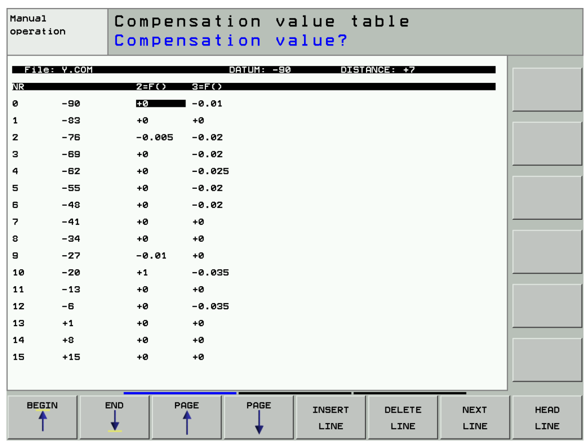
The following dependencies apply for axes 2 = Y and 3 = Z:

- Ball screw pitch error in Z and Y: Z = F(Z) and Y = F(Y)
- Axis sag in Z depending on Y
- Traverse range: Z = 800 mm, Y = 500 mm
- Datum point of compensation values: Z = –200 mm, Y = –90 mm
- Desired spacing of compensation points: 7 mm

Number of compensation points:

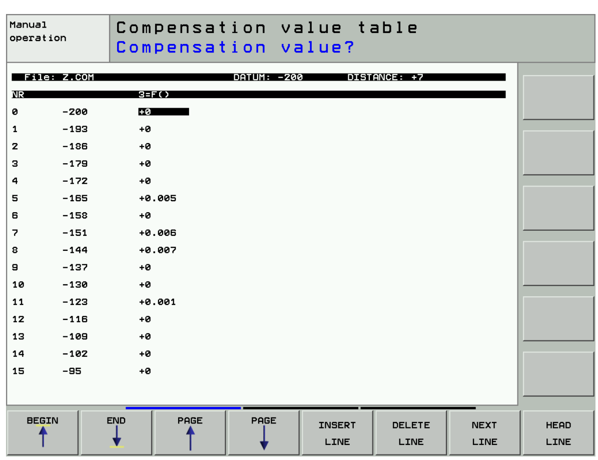
$$\frac{500\text{ mm}}{7\text{ mm}} = 71 \quad \text{Compensation points in Y}$$

$$\frac{800\text{ mm}}{7\text{ mm}} = 114 \quad \text{Compensation points in Z}$$



Axis Y: Ball screw-pitch error in column 2 = F(),

Sag error in column 3 = F()



Axis Z: Ball screw pitch error in column 3 = F()

Activate error compensation

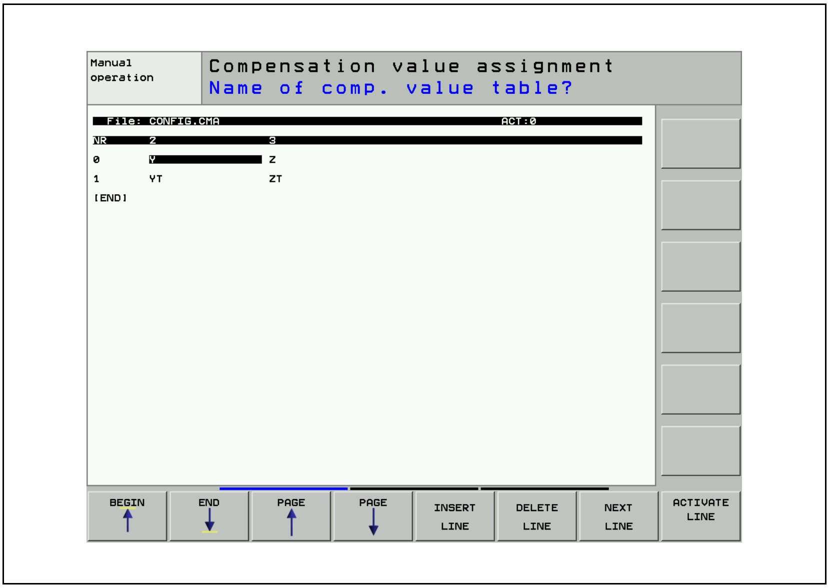
The appropriate machine parameter must be set for nonlinear axis error compensation, and the compensation value table must be registered in a configuration file:

- With MP730, activate for each axis the nonlinear axis error compensation.
- To create a configuration file: Press the MOD key and enter the code number 807667; with the program manager, create a configuration file with the extension .CMA.
- Use soft keys (e.g. **1 OFF/ON**) to activate columns for the axes for which you have created compensation value tables.
- Enter the compensation value table: You can assign more than one compensation value table to each axis; however, only one table at a time can be active. Enter the file names of the compensation value tables in the respective lines. You can select the active line either with the soft key **ACTIVATE LINE** or with Module 9095. With Module 9035 you can interrogate the active line.
- Enter the complete name of the configuration file of the *.CMA type in the system file OEM.SYS with the **TABCMA =** command.

Example

Entry in the configuration file for axes 2 = Y and 3 = Z:

Compensation value tables valid for 20 °C = Y.COM and Z.COM
Compensation value tables valid for 35 °C = YT.COM and ZT.COM



MP730 Selection of linear/nonlinear axis error compensation

Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Linear axis error compensation
1: Nonlinear axis error compensation

Module 9095 Select active line in configuration file

Call:

PS	B/W/D/K	<>Active line>
CM	9095	
PL	B/W/D	<>Error code> 0: No error 1: Entered line does not exist 2: Compensation value table does not exist 3: Compensation value table > 256 entries 4: Maximum total number of compensation points exceeded 5: Too many compensation value tables 6: CMA file does not exist 7: Call was not from a submit job 8: Call during running program without strobe 10: CMA file is protected

Module 9035 Read status information

Call:

PS	B/W/D/K	<>19>
CM	9035	
PL	B/W/D	<>Active line number> 0: Line number –1: No .CMA file active

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid
	20	Call was not in a submit or spawn job

Special case: Rotary axis

Up through software version 04, for a rotary axis only the compensation values for the entries of 0° to +360° were effective, relative to the machine datum. With MP732, the non-linear axis-error compensation can be projected onto the actual traverse range for a specific axis. This means that traverse ranges from –90° to +90°, for example, can be entered for rotary axes.

MP732 Nonlinear axis-error compensation for rotary axes

Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Not active (usual compensation)
1: Active (mapped to traverse range)

6.4.4 Compensation of Thermal Expansion

To compensate thermal expansion, exact measurements of machine thermal behavior as a function of temperature (e.g., the center of axis expansion, the amount of the expansion) are necessary.

The temperatures measured by the Pt100 thermistors are saved in the PLC words W486 to W490. Since the thermal expansion of the axes is largely proportional to the temperature: you can directly determine the amount of expansion by multiplying the temperature value by a certain factor.

Compensation:

- Transfer the distance to be compensated to Module 9231. At the same time, "lag tracking" becomes active. This means that the actual position is offset by a certain value per PLC cycle until the complete value is compensated.
- In MP4070, enter the value for the offset per PLC cycle.

For gantry axes, the compensation value must be transferred separately for each axis.

Heat compensation when using tilting axes is defined through machine parameters or via the **TEMPCOMP** column of the kinematics table. See "Temperature Compensation with Tilting Axes"on page 752.

The actual value display does not change during compensation. As an alternative, for axes 1 to 5 you can enter the value to be corrected in W576 to W584.

MP4070	**Compensation amount per PLC cycle for lagged-tracking axis error compensation**
Input:	0.0001 to 0.5000 [mm]

		Set	Reset
W486 - 490	**Temperature input at X48 [0.5 °C]** Inputs 1 to 3	NC	NC
W576 - 584	**Lag-tracking axis-error compensation** For axes 1 to 5 Input: –32 768 to +32 767 [1/10 µm]	PLC	PLC

Module 9231 Compensation of thermal expansion

With Module 9231, thermal expansion can be compensated by transferring the axis number and a compensation value.

Call:

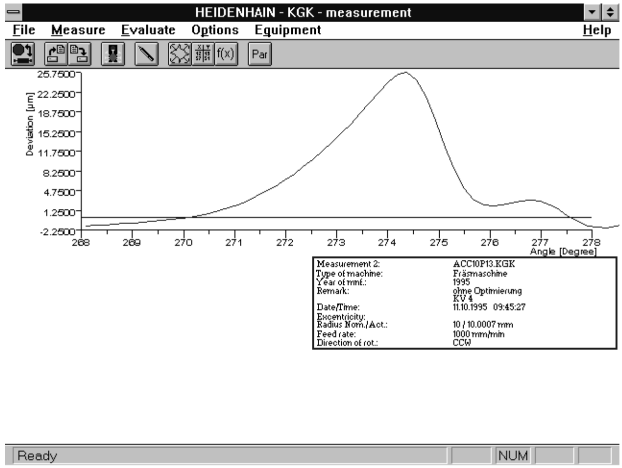
PS	B/W/D/K	<>Axis> Axes 0 to 8
PS	B/W/D/K	<>Compensation value> Range: –30000 to +30000 [1/10 µm]
CM	9231	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Invalid axis number
	2	Invalid compensation value
	24	The module was called in a spawn job or submit job

6.4.5 Compensation of Reversal Spikes during Circular Traverse

The stick-slip friction in the axis bearings during circular movement can lead to reversal spikes at the quadrant transitions. With the HEIDENHAIN KGM grid encoder and the ACCOM evaluation software you can measure the size and duration of the spikes.



Calculation

Duration of the reversal spike:

$$t_{SpD}[s] = \frac{\text{Peak width } [°] \cdot 2(\pi \cdot \text{radius [mm]} \cdot 60)}{360\ [°] \cdot \text{Feed rate [mm/min]}}$$

The spike width is [°] displayed in the diagram. The feed rate [mm/min] is the programmed tool path feed rate.

Compensation per control loop cycle time:

$$\text{Comp. [mm =]} \frac{\text{Reversal peaks [µm]} \cdot \text{control loop cycle time} \cdot \text{[s]} \cdot 10^{-3}}{0.5 \cdot t_{SpD}[s]}$$

The compensation value is entered in MP712.x.

Compensation

Digital axes:

You must compensate friction in the range of the speed controller (MP2610.x to MP2620.x). Do not compensate with MP711.x to MP716.x. See “Compensation of Sliding Friction (Only for Digital Axes)" on page 703.

Analog axes:

If you have compensated the **backlash** with MP750.x, there should be no more reversal spikes. If there are, compensate them with MP711.x to MP716.x.

Compensation:

- In MP711.x, enter the height of the spike.
- In MP712.x, enter the amount of the reversal spike that is to be compensated per control loop cycle (see "Calculation" above).

MP711.x **Height of peaks during circular movement (analog only)**
Input: -1.0000 000 to +1.0000 999 [mm] (digital: 0)

MP712.x **Compensation value per control loop cycle time**
Input: 0.000 000 to 99.999 999 [mm] (digital: 0)

If the compensation has no effect, it may be because the machine’s dynamic performance is too weak. You can selectively increase the contour accuracy with a higher k_V factor. With the M function M105 you can switch to a second set of k_V factors: In this way a second set of machine parameters becomes active for reversal spike compensation (MP715.x and MP716.x). M106 resets M105.

- Enable the M functions M105/M106 with MP7440, bit 3.
- In MP715.x, enter the height of the spike.
- In MP716.x, enter the amount of the reversal spike that is to be compensated per control loop cycle (see "Calculation" above)

MP7440 **Output of M functions**
Format: %xxxxx
Input: Bit 3 – switching the k_V factors with M105/M106
0: Function is not in effect
1: Function is effective

MP715.x **Height of peaks during circular movement (analog only) with M105**
Input: -1.0000 000 to +1.0000 999 [mm] (digital: 0)

MP716.x **Compensation value per control loop cycle time with M105**
Input: 0.000 000 to 99.999 999 [mm] (digital: 0)

6.4.6 Compensation of Stick-Slip Friction

On guideways with high stick-slip friction (static friction), a following error can occur at low feed rates during operation with velocity feedforward control. This error can be compensated by the iTNC. You can measure following error by using, for example, the integrated oscilloscope of the iTNC.

Compensation of stick-slip friction works only under velocity feedforward control. If it is also to work in manual operating modes, you must activate velocity feedforward control in each axis with MP1391.x for manual operation.

Calculations

For compensation of stick-slip friction, an additive nominal velocity is output whose value F_{zus} is calculated from the factor for stick-slip friction compensation:

$$F_{zus} = \frac{\Delta s_a}{t_R} \cdot k_v \cdot MP1511$$

F_{zus} = additional feed rate [m/min]
Δs_a = following error difference after one control loop cycle [mm]
t_R = control loop cycle time [µs]
k_v = control loop gain [(m/min)/mm]
MP1511.x = factor for stick-slip friction compensation [µs]

This additive nominal value is limited with MP1512.x. If this limit is too high, the machine vibrates while at standstill:

$$MP1512.x = \frac{s_{agrenz} \cdot 256}{TP}$$

MP1512.x = limitation of the amount of the stick-slip friction compensation [counting steps]
s_{agrenz} = limit value for Δs_a [µm]
TP = grating period of the encoder [µm]

Compensation

The compensation must be effective only at low feed rates, otherwise the nominal value increase will cause vibration at high velocity:

- In MP1511.x, enter a factor for stick-slip friction compensation (approximate value: 5000 to 10 000).
- In MP1512.x, enter a limit for the amount of the stick-slip friction compensation (approx. value: < 50).
- In MP1513.x, limit the maximum feed rate up to which the stick-slip friction compensation remains in effect.

Compensation of stick-slip friction via MP1511.x to MP1513.x is only useful for analog controlled axes. For digitally controlled axes, please use machine parameters MP2610.x to MP2614.x.

MP1511.x	**Factor for stick-slip friction compensation**
Input:	0 to 16 777 215 [µs]
MP1512.x	**Limitation of the amount of the stick-slip friction compensation**
Input:	0 to 16 777 215 [counting steps]
MP1513.x	**Feed-rate limitation for stick-slip friction compensation**
Input:	0 to 300 000 [mm/min]
MP1391.0	**Velocity feedforward control in the MANUAL and HANDWHEEL operating modes**
Format:	%xxxxxxxxxxxxxx
Input:	Bits 0 to 13 represent axes 1 to 14 0: Inactive 1: Active
MP1391.1	**Acceleration feedforward control in the MANUAL and HANDWHEEL operating modes**
Format:	%xxxxxxxxxxxxxx
Input:	Bits 0 to 13 represent axes 1 to 14 0: Inactive 1: Active

Digital axes: Limitation of the integral factor

In machines with very high static friction, a position deviation at standstill can lead to the accumulation of a very high integral factor. This can lead to a jump in the position value when the axis "tears loose." In such cases you can limit the integral-action component of the speed controller with MP2512.x.

The limitation is effective as soon as the axis is in position (W1026). If high utilization occurs when an axis is at standstill (e.g. due to static friction), the current is also limited.

MP2512.x	**Limit of integral factor of the speed controller**
Input:	0.000 to 30.000 [s] (realistically: 0.1 to 2.0)

6.4.7 Compensation of Sliding Friction (Only for Digital Axes)

Sliding friction is compensated within the range of the speed controller:

- With the integrated oscilloscope of the iTNC, define the nominal current value (I NOMINAL) at a very low speed of approx. 10 rpm.
- Enter the value for current in MP2610.x. At every change in direction, this amount is fed forward to the speed controller to compensate the sliding friction at low speeds.
- Measure the nominal value for current (I NOMINAL) at rated speed and enter it in MP2620.x. Depending on the nominal speed value, a certain current is fed forward to the speed controller and causes a sliding friction that depends on the speed.

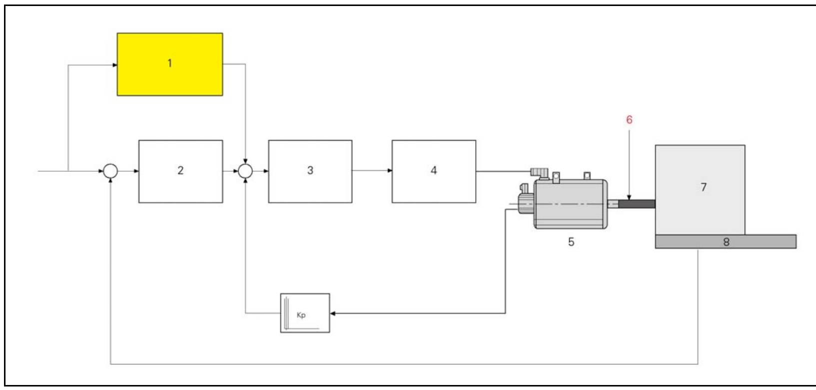
When the traverse direction is reversed at high feed rates, the sliding friction might be overcompensated. In a circular interpolation test, such overcompensation appears in the form of reversal spikes that jut inward. With MP2612.x you can prevent overcompensation by delaying the compensation.

MP2610.x	**Friction compensation at low speeds (effective only with velocity feedforward control)**
Input:	0 to 100.0000 [A] 0: No friction compensation (or axis is analog)
MP2612.x	**Delay of the friction compensation (effective only with velocity feedforward control)**
Input:	0.0000 to 1.0000 [s] (typically: 0.015 s) 0: No friction compensation (or axis is analog)
MP2620.x	**Friction compensation at rated speed**
Input:	0 to 100.000 [A] 0: No friction compensation (or axis is analog)

6.4.8 Torsion Compensation

With MP2640.x you can perform a torsion compensation between the position and speed measuring systems. The torsion compensation regulates the difference in position that results from the elasticities between the motor (rotary encoder) and the position measuring system. An additional torsion motion is added to the speed controller. In MP 2640.x you enter a drive-specific factor for the elastic constant. HEIDENHAIN recommends using MP 2640 as a replacement for MP 2606, since MP 2640 can be used to take additional effects into account.

The block diagram shows how the torsion compensation works:



- 1: Torsion compensation
- 2: Position controller
- 3: Speed controller
- 4: Current controller – power module
- 5: Motor
- 6: Elastic coupling
- 7: Machine
- 8: Linear encoder

Adjustment of MP2640.x is only worthwhile once all other feedforward-control parameters MP2600.x to MP2630.x have been adjusted. Furthermore, stick-slip friction compensation via MP2640.x should not be used if stick-slip friction compensation only takes place via MP2612.x (MP2614.x=0).

Torsion compensation via machine parameter MP2640.x is now as of 340 49x-06 also added in machines without position encoder if a value that is not equal to 0 is defined in MP2640.x. The adjustment remains unchanged, i.e. a position encoder must be used. However, the compensation values can now also be used for dimensionally identical machines without position encoder. In addition, the compensation value is filtered before it is added. This improves the behavior of the axis after direction reversal, for example.

MP2640.x	**Torsion compensation**
Input:	0.001 to 100.000 [µm/A] 0: Not active

6.5 Tilting Axes

Swivel heads and tilting tables are often used on milling machines for 5-axis machining and to machine workpieces from several sides.

The NC programs are written with a CAD system or directly at the iTNC using the **Tilt working plane** (Plane, TCPM) function. The user programs the part program in the X/Y plane and the iTNC interpolates the proper axes. All path functions, cycles, "datum setting" and "probing" can be applied in the transformed working plane.

6.5.1 Determining the Mechanical Offset

As an example, we will show how to determine the mechanical offset of a 45° double swivel head and of a forked swivel head.

> Note
>
> The 3-D ROT function must be inactive during the entire measuring process.

Double swivel head 45°

In this example, the mechanical offset of a double swivel head is determined by using a 3-D touch probe.

Input values for mechanical offset up to software 340 422-xx (via machine parameters):

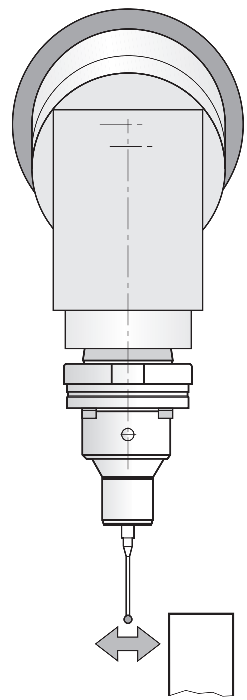
- MP7500 = %xxxx101
- MP7510.0 = %000100 ; Shift in Z axis
- MP7510.1 = %000001 ; Shift in X axis
- MP7510.2 = %001000 ; Rotate about A axis
- MP7510.3 = %000100 ; Shift in Z axis
- MP7510.4 = %100000 ; Free tilting axis C
- MP7510.5 = %001000 ; Rotate about A axis
- MP7510.6 = %000001 ; Shift in X axis
- MP7510.7 = %010000 ; Free tilting axis B
- MP7510.8 = %000000 ; End transformation

Input values for mechanical offset as of software 340 490-xx (via description table in the kinematics, ## are the values to be determined):

NR	KEY	AXIS	COORD	ON/OFF	FILE	DOC
0	**Trans**	**Z**	##			Shift in Z axis
1	**Trans**	X	##			Shift in X axis
2	Trans	**A**	##			Rotate around A axis
3	**Trans**	Z	##			Shift in Z axis
4	MachAxis	C				Free tilting axis C
5	**Trans**	A	##			Rotate around A axis
6	Trans	**X**	##			Shift in X axis
7	**MachAxis**	**B**				Free tilting axis B
:						
[END]						

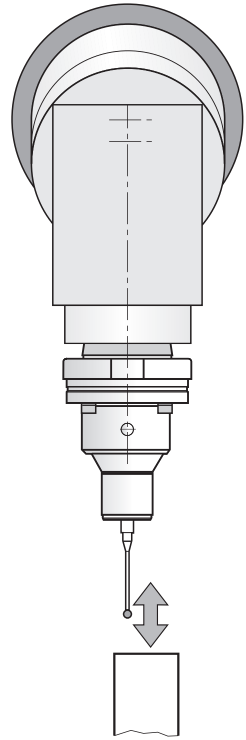
Step 1a

- Bring tilting axes B and C into 0° position
- Probe surface X1
- Set X = 0



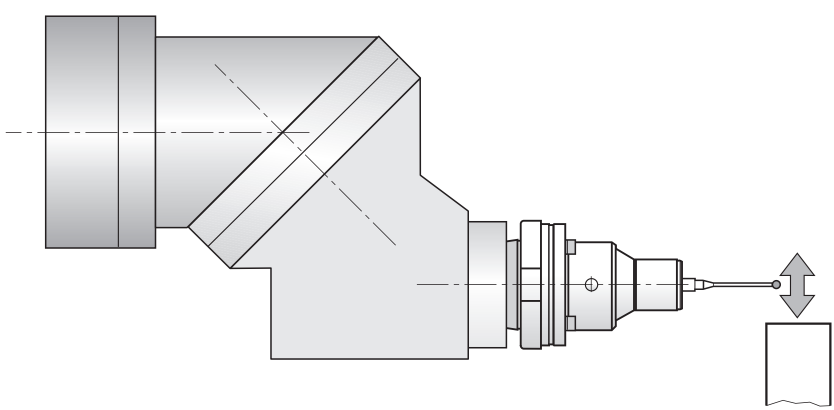
Step 1b

- Probe surface Z
- Set Z = 0



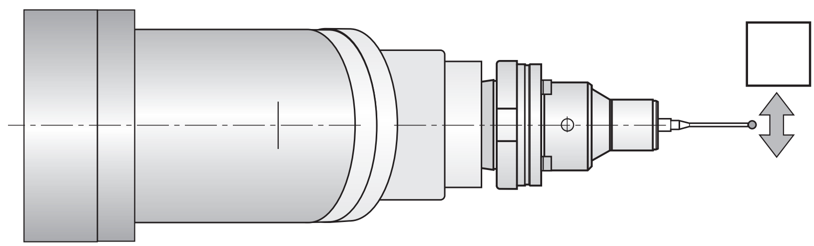
Step 2

- Position C = 180
- Probe surface Z
- Entry for the kinematics
  - Via MP: MP7530.0 = –determined value – probe length + ball radius
  - Via table: **`<Line no. [0]`** Column **`COORD>`** = –determined value – probe length + ball radius



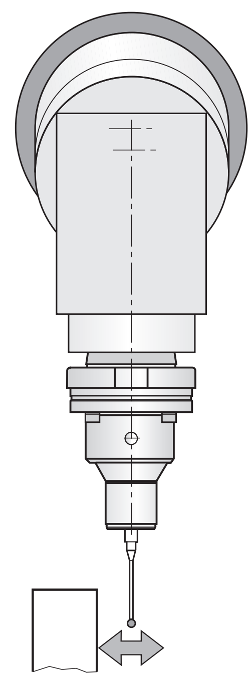
Step 3

- Probe surface X1
- Entry for the kinematics
  - Via MP: MP7530.1 = –0.5 * determined value
  - Via table: **<Line no. [1]** Column **COORD>** = –0.5 * determined value



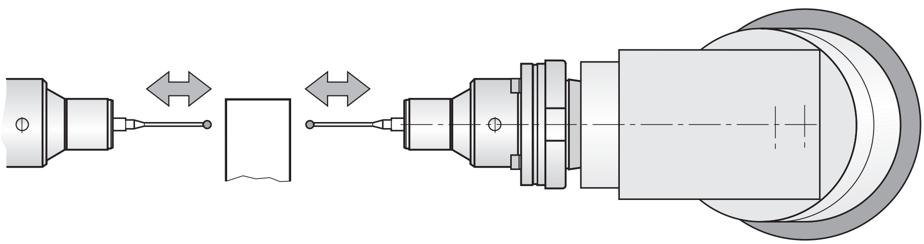
Step 4a

- Position C = 0
- Probe surface X2
- L = determined value



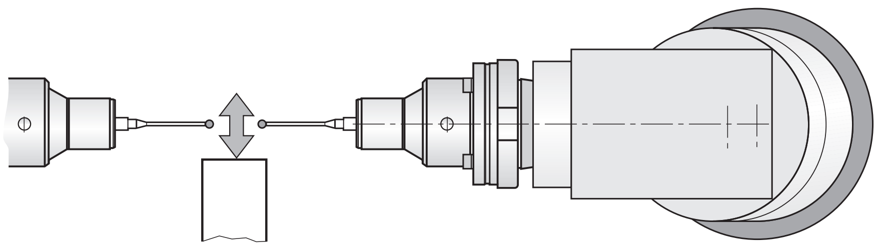
Step 4b

- Position B = -90
- Probe surface X1
- ΔX1 = determined value
- Position B = +90
- Probe surface X2
- ΔX2 = determined value
- Entry for the kinematics
  - Via MP: MP7530.3 = {[0.5 * (ΔX2 – ΔX1 – L – 2*(probe length) + 2*(ball radius))] – MP7530.0} / cos 45°
  - Via table: **<Line no. [3]** Column **COORD>** = {[0.5 * (ΔX2 – ΔX1 – L – 2*(probe length) + 2*(ball radius))] – MP7530.0} / cos 45°



Step 4c

- Position B = -90
- Probe surface Z
- Set Z = 0
- Position B = +90
- Probe surface Z
- Entry for the kinematics
  - Via MP: MP7530.6 = (–0.5 * determined value) – MP7530.1
  - Via table: **<Line no. [6]** Column **COORD>** = (–0.5 * determined value) – **<Line no. [1]** Column **COORD>**

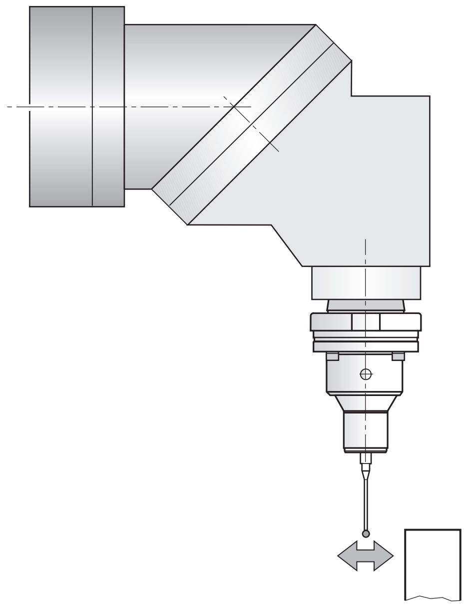


Step 5

- Probe surfaces X1, X2, Y2 and Z in a tilted working plane with the tilting angles B = –90, B = +90 and C = 180 (with MP7500 = %xxxxx0x)
- If there are differences between the individual tilting angles, the offsets from MP7530.0 and MP7530.3 or line numbers [0] and [3] in the kinematics table should be determined with a different process (steps 6 to 8), and the averages from both processes should be entered in MP7530.0 and MP7530.3 or line numbers [0] and [3] in the kinematics table.

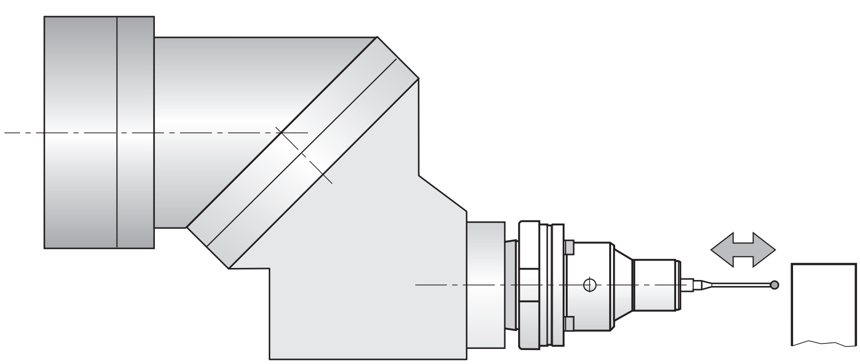
Step 6

- Probe surface Y2
- Set Y = 0



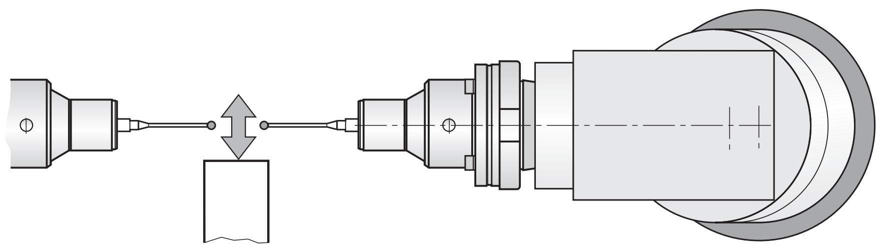
Step 7

- Position C = 180
- Probe surface Y2
- ΔY2 = determined value
- Z1 = ΔY2 – probe length + ball radius
- If there is a difference between MP7530.0 and Z1, then MP7530.0 = 0.5 * (MP7530.0 + Z)
- Entry for the kinematics if the results differ for
  - MP7530.0 and Z1, then MP7530.0 = 0.5 * (MP7530.0 + Z)
  - **<Line no. [0]** Column **COORD>** and Z1, then **COORD** = 0.5 * (**<Line no. [0]** Column **COORD>** + Z)

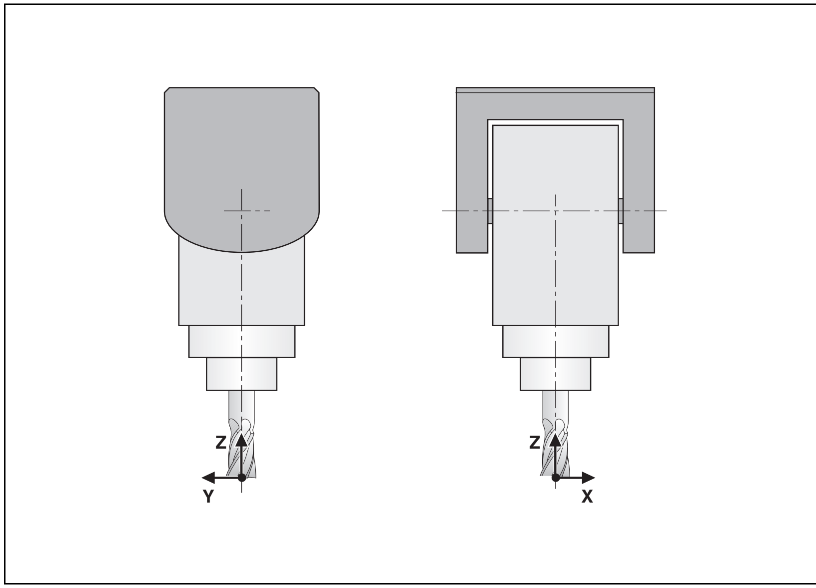


Step 8

- Position C = 0
- Position B = -90
- Probe surface Z
- ΔZ = determined value
- Position B = +90
- Probe surface Z
- Z = {[–0.5 * (ΔZ + determined value) – probe length + ball radius] – Z1} / cos 45°
  (Z1 see step 7)
- If there is a difference between MP7530.3 and Z, then MP7530.3 = 0.5 * (MP7530.3 + Z)
- Entry for the kinematics if the results differ for
  - MP7530.3 and Z, then MP7530.3 = 0.5 * (MP7530.3 + Z)
  - **<Line no. [3]** Column **COORD>** and Z, then **COORD** = 0.5 * (**<Line no. [3]** Column **COORD>** + Z)



Forked swivel head In this example, the mechanical offset of a forked swivel head is determined with a dial gauge and a cylinder with a known diameter.



Input values for mechanical offset up to software 340 422-xx (via machine parameters):

- MP7500 = %xxxx101
- MP7510.0 = %000100 ; Shift in Z axis
- MP7510.1 = %000010 ; Shift in Y axis (Y1: Offset of spindle to A axis)
- MP7510.2 = %001000 ; Free tilting axis A
- MP7510.3 = %000001 ; Shift in X axis
- MP7510.4 = %000010 ; Shift in Y axis (Y2: Offset of fork to C axis)
- MP7510.5 = %100000 ; Free tilting axis C
- MP7510.6 = %000000 ; End transformation

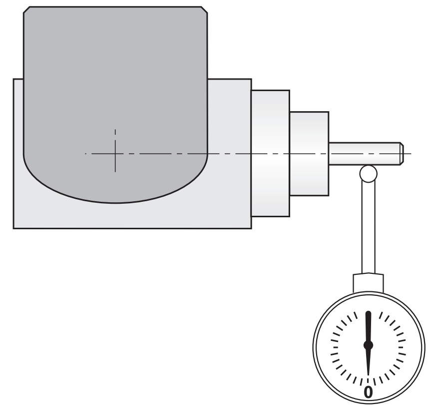
Input values for mechanical offset as of software 340 490-xx (via description table in the kinematics, ## are the values to be determined):

NR	KEY	AXIS	COORD	DOC
0	Trans	Z	##	Shift in Z axis
1	Trans	Y	##	Shift in Y axis (Y1: Offset of spindle to A axis)
2	MachAxis	A		Free tilting axis A
3	Trans	X	##	Shift in X axis
4	Trans	Y	##	Shift in Y axis (Y2: Offset of fork to C axis)
5	MachAxis	C	##	Free tilting axis C
:				
[END]				

Step 1a

Determining the Y1 offset:

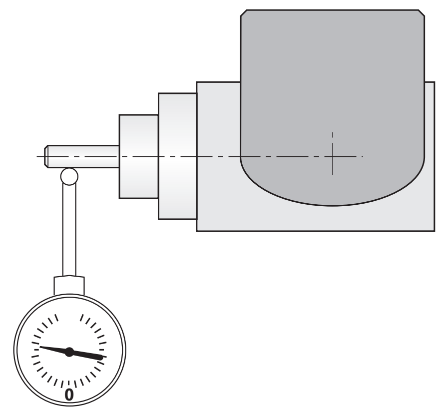
- Position A = –90
- Set the dial gauge to 0



Step 1b

Determining the Y1 offset:

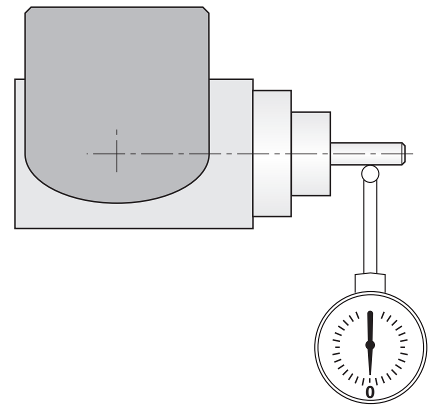
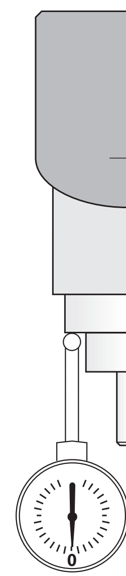
- Position A = +90
- Offset = 0.5 * determined value
- If the determined value > 0, then MP7530.1 = **–** offset or **<Line no. [1]** Column **COORD>** = **–** offset
- If the determined value < 0, then MP7530.1 = **+** offset or **<Line no. [1]** Column **COORD>** = **+** offset



Step 1c

Checking the settings for the Y1 offset:

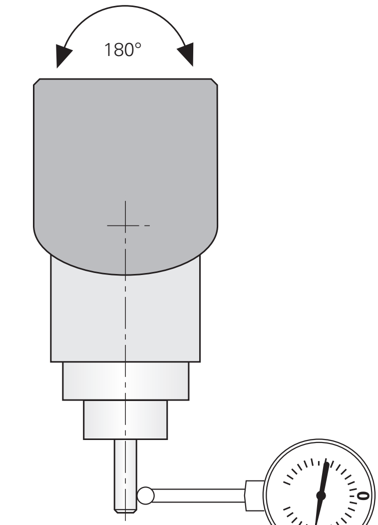
- Position A = +90
- Activate 3-D ROT
- Set the dial gauge to 0
- Set reference point to Y = 0
- Position A = –90
- Probe same position again
- Display and dial gauge must read 0

<table>
<tr>
<td>Step 2a
Determining the Z offset:
■ Deactivate 3-D ROT
■ Position A = –90
■ Set the dial gauge to 0
■ Set the Z display = 0</td>
<td></td>
</tr>
<tr>
<td>Step 2b
Determining the Z offset:
■ Position A = 0
■ Move the Z axis until the dial gauge reads 0 at the spindle tip
■ MP7530.0 = Value for Z axis – cylinder radius or <Line no. [0] Column COORD> = Value for Z axis – cylinder radius</td>
<td></td>
</tr>
</table>

Step 3

Determining the Y2 offset:

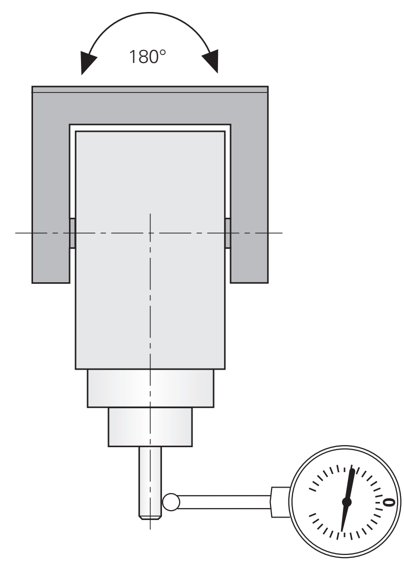
- Position A = 0
- Position C = 0 (basic setting)
- Set the dial gauge to 0
- Position C = 180
- Read Y offset from the dial gauge
- **MP7530.4** = (0.5 * determined value) – MP7530.1;
  (MP7530.1 = offset Y1)
- or **<Line no. [4]** Column **COORD>** = (0.5 * determined value) – **<Line no. [1]** Column **COORD>**;
  (**<Line no. [1]** Column **COORD>** = Offset Y1)



Step 4

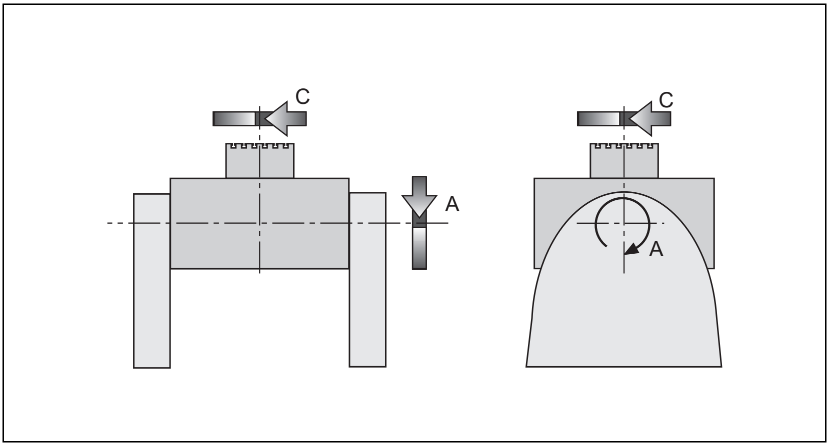
Determining the X offset:

- Position C = -90
- Set the dial gauge to 0
- Position C = 270
- Read X offset from the dial gauge
- MP7530.3 = 0.5 * determined value
- or **<Line no. [3]** Column **COORD>** = 0.5 * determined value



Tilting table / rotary table

In this example, the mechanical offset of a tilting or rotary table is determined by using a 3-D touch probe. In the A axis the table can only be tilted by -90°.

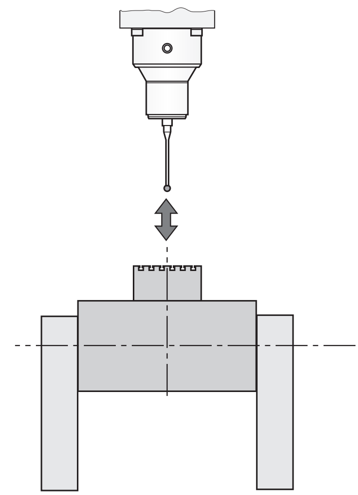
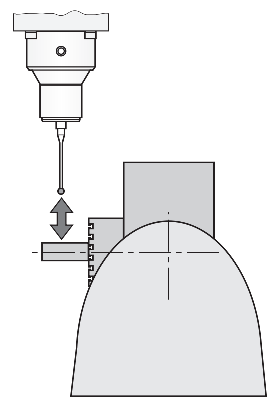
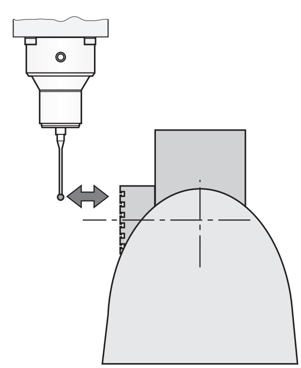


Input values for mechanical offset up to software 340 422-xx (via machine parameters):

- MP7500 = %xxxx101
- MP7510.0 = %000001 ; X coordinate of the center of rotation of C axis
- MP7510.1 = %000010 ; Y coordinate of the center of rotation of C axis
- MP7510.2 = %000100 ; Z coordinate of the center of rotation of C axis
- MP7510.3 = %100000 ; Free tilting axis C
- MP7510.4 = %000010 ; Shift in Y axis
- MP7510.5 = %000100 ; Shift in Z axis
- MP7510.6 = %001000 ; Free tilting axis A
- MP7510.7 = %000000 ; End transformation

Input values for mechanical offset as of software 340 490-xx (via description table in the kinematics):

NR	KEY	AXIS	COORD	DOC
	MachAxis	**X**		X axis
	MachAxis	**Y**		Y axis
	MachAxis	**Z**		Z axis
	Trans	**X**	[X1]	Shift in X axis to the center of rotation of the C axis
	Trans	**Y**	[Y1]+0.5 * (ΔZ – ΔY)	Shift in Y axis to the A axis
	Trans	**Z**	[Z1]+0.5 * (ΔZ + ΔY)	Shift in Z axis to the A axis
	MachAxis	**A**		Free tilting axis A
	Trans	**Y**	0.5 * (ΔZ – ΔY)	Shift in Y axis
	Trans	**Z**	0.5 * (ΔZ + ΔY)	Shift in Z axis
	MachAxis	**C**	##	Free rotary axis C
[END]				

<table>
<tr><td colspan="2">Step 1
Ascertain the center point of the center of rotation of the C axis using the 3-D touch probe (= X1 and Y1); set X = 0 and Y = 0.</td></tr>
<tr><td>Step 2
■ Probe surface Z (= Z1)
■ Set Z = 0</td><td></td></tr>
<tr><td>Step 3
Determining the Z offset:
■ Position A = –90
■ Probe at test mandrel
■ ΔZ = radius of test mandrel + Z display</td><td></td></tr>
<tr><td>Step 4
Determining the Y offset:
■ Probe surface Z
■ ΔY = Display Y</td><td></td></tr>
<tr><td colspan="2">Step 4
■ MP7530.4 = 0.5 * (ΔZ – ΔY)
■ MP7530.5 = 0.5 * (ΔZ + ΔY)
■ or entry of the values in the description table as indicated above</td></tr>
</table>

6.5.2 Calculating the Mechanical Offset up to Software 340 422-xx

Note

The description of the mechanical offset is only taken into account for tilting functions, such as TCPM or the Tilt Working Plane function.

Describing the mechanical offset

Determine the mechanical offset of the axes in the home position. For swivel heads, the starting point is the tool datum; for tilting tables, the starting point is the center of rotation of the first axis (as seen from the workpiece):

- Only for tilting tables: Define the center of rotation of the first tilting axis with respect to the machine datum.
- Determine in sequence the linear or rotary offset to the next tilting axis until you reach a point that is not separated from the machine frame by any free tilting axis.
- In MP7510.x, enter the sequence of the transformed axes, in MP7520.x the type of axis and dimensional data, and in MP7530.x enter the value of the offset. See the examples on the following pages.
  The sequence of the axes in MP7510.x is fixed (%CBAZYX), regardless of MP100.x.
- Normally the home position of the tilting element is the 0° setting. If this is not possible, enter in MP7550.x the home position in the machine coordinate system.

If a rotation has been entered, it must be canceled again in an additional transformation.

As a rule, the control takes changes in the mechanical offset into account, meaning that these changes do not have to be compensated with a PLC datum shift.

Describing the mechanical offset with tables

In order to manage several descriptions of the mechanical offset, e.g. when swivel heads are changed, the descriptions can be saved in tables. A description is activated either by the PLC or the NC.

Two types of tables are required:

Assignment table
Each row corresponds to one description (row 0 = description no. 1, etc.). The first column contains the line number. For each description (= row), the value of MP7500 for the description is entered in column two. Keep in mind that the value must be entered as a decimal number. The file name with its complete path is entered in the third column.

Description tables
The description table contains the contents of machine parameters MP7510.x, MP7520.x, MP7530.x and MP7550.x. The index x corresponds to the line number.
Of course the MP7530 column may also contain formulas, such as temperature compensation with TCPM, etc. (See "Temperature compensation" on page 753). A formula for a permanently effective temperature compensation may be entered in the TEMPCOMP column (See "Permanent temperature compensation" on page 754).

Working with the description of the mechanical offset in tables:

- Switch to the **`Programming and Editing`** operating mode, press the MOD key and enter the code number 807667.
- Choose the file PLC:\OEM.SYS from within program management.
- Enter the code word **`KINEMATIC=`**, followed by the file name with its complete path from the assignment table.
  (e.g., **`KINEMATIC= PLC:\KINEMAT\KINELIST.TAB`**)
- Leave OEM.SYS by pressing the END key.
- To create an assignment table: In program management, switch to the desired directory and enter the name of the assignment table, including the extension .TAB.
- Choose the table format with the MP7500, FILE and MPFILE fields.
- Enter the value from MP7500.x in the table for each description, and the path to the corresponding description table.
- To create a description table: In program management, switch to the desired directory and enter the name of the description table, including the extension .TAB.
- Choose the table format with the MP7510, MP7520, MP7530, MP7550 and TEMPCOMP fields.
- Enter the values of machine parameters MP7510.x, MP7520.x, MP7530.x and MP7550.x in the table.
- Activate the description table by transferring the row numbers from the assignment table
  - from the PLC with Module 9097
  - from the NC with **`FN17: SYSWRITE ID290 NR1`**
- You can ascertain the active description table in two ways:
  - With the PLC you can use Module 9098 to ascertain the name of the description table or the line number in the assignment table.
  - With the NC you can use **`FN18: SYSREAD ID290 NR1`** to ascertain the line number in the assignment table.

Attention

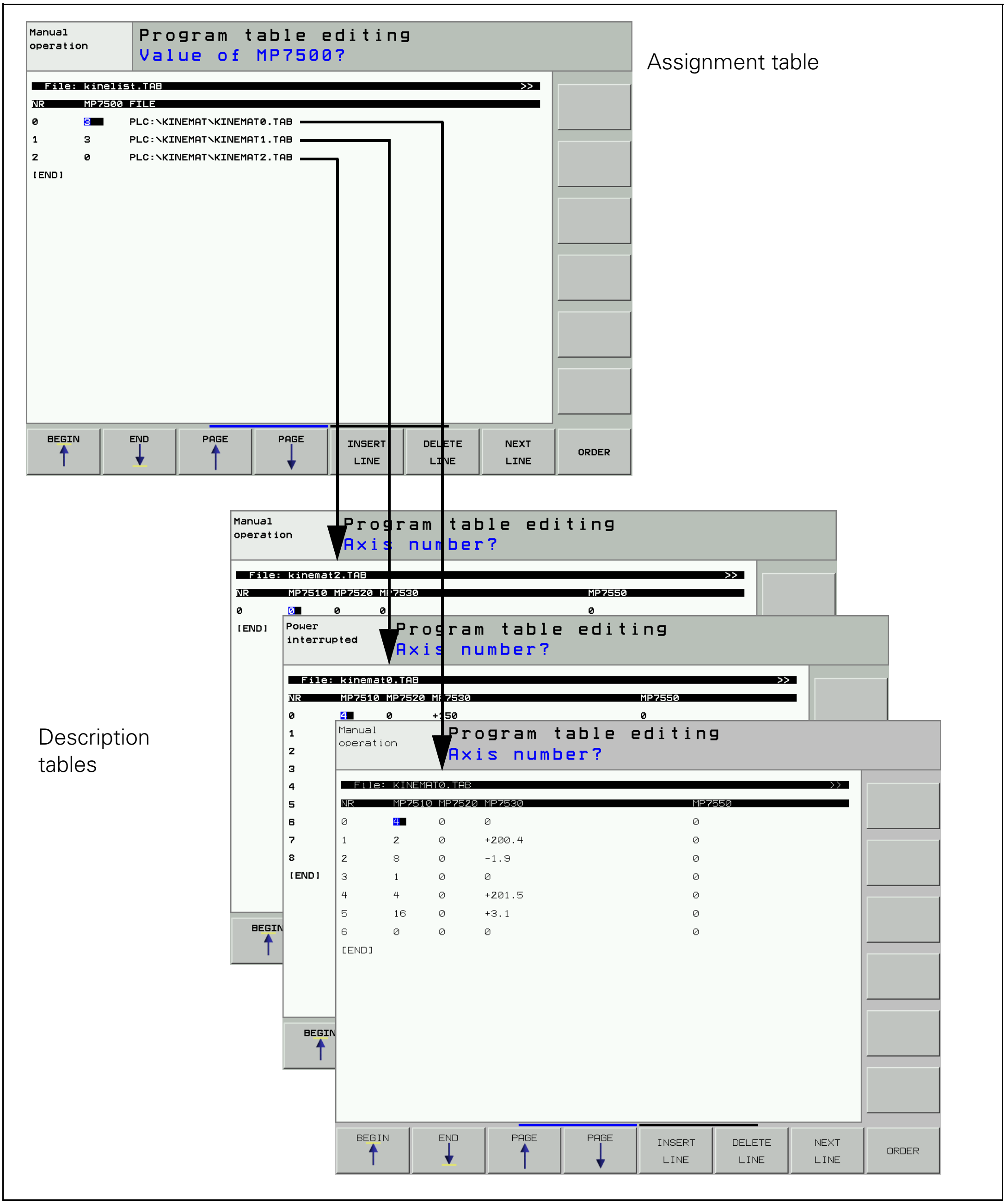
Please ensure that the objects and files referenced in the tables are written correctly. Not all possible errors can be reported.

Note

The active description table is indicated with the status "M" in program management.

Time at which changes to the descriptions become effective:

- Swivel heads: when the corresponding description table is selected again.
- Tilting tables: when the corresponding description table is selected again **and** a new datum is set.


Assignment table
Description tables
Program table editing
Value of MP7500?
File: kinelist.TAB
NR MP7500 FILE
0 3 PLC:\KINEMAT\KINEMAT0.TAB
1 3 PLC:\KINEMAT\KINEMAT1.TAB
2 0 PLC:\KINEMAT\KINEMAT2.TAB
[END]
Axis number?
File: KINEMAT0.TAB
NR MP7510 MP7520 MP7530 MP7550
0 4 0 0 0
1 2 0 +200.4 0
2 8 0 -1.9 0
3 1 0 0 0
4 4 0 +201.5 0
5 16 0 +3.1 0
6 0 0 0 0
[END]
BEGIN
END
PAGE
PAGE
INSERT LINE
DELETE LINE
NEXT LINE
ORDER

Example:

This example shows an assignment table for three description tables. The double swivel head 45° from example 2 was entered in the description table.

Assignment table KINEMATIC.TAB

NR	MP7500	FILE	MPFILE	DOC
0	7	PLC:\KINEMAT1.TAB		
1	7	PLC:\KINEMAT2.TAB		
2	7	PLC:\KINEMAT3.TAB		
[END]				

Description table KINEMAT1.TAB

NR	MP7510	MP7520	MP7530	TEMPCOMP	MP7550
0	4	0	+150.5		
1	8	0	–45		
2	4	0	+251.5		
3	32	0	0		
4	8	0	+45		
5	16	0	0		
6	0	0	0		
[END]					

6.5.3 Description for Configuring the Kinematics as of Software 340 490-xx

Note

The description of the kinematics is only taken into account for tilting functions, such as TCPM or the Tilt Working Plane function.

Differences to the previous kinematics description

The previous kinematics description (NC software 340 422-xx) contains partial transformation chains such as the description of a swivel head (from the tool reference point to the tilting axis) or a tilting table (from the center point of the rotary table to the tilting table).

By contrast, the new kinematics description (as of NC software 340 490-02), consists of one transformation chain, going from the tool reference point over the tilting head, linear axes and tilting table to the center of the rotary table.

Seen as a whole, the new kinematics description defines one transformation chain, consisting of

- fixed lengths (machine dimensions)
- variable lengths (linear axes)
- fixed rotations (machine conditions)
- variable rotations (rotary axes)

starting from the tool reference point (e.g. spindle housing, swivel head, machine base, linear axes, machine envelope, tilting table, rotary table). Each translation is described.

The difference to the previous kinematics description is that here, no coding is specified for the head or table rotary axes, or for the "ref. coordinates." This means that the sequence and algebraic signs of the table elements are reversed compared to the old description.

The following machine equation results for a closed transformation chain (tool reference point is at the center point of the rotary table):

$$T_{Head} + T_{Table} + T_{New} - T_{Ref} = 0$$

$$=>$$

$$T_{New} = T_{Ref} - T_{Head} - T_{Table}$$

T_{Head}: Translation for swivel head

T_{Table}: Translation for tilting table (algebraic sign according to the new kinematics description)

T_{New}: New translation (sum of the remaining translations after the tilting head to before the rotary table)

T_{Ref}: "Ref. coordinates" (previous description of the center point of the rotary table)

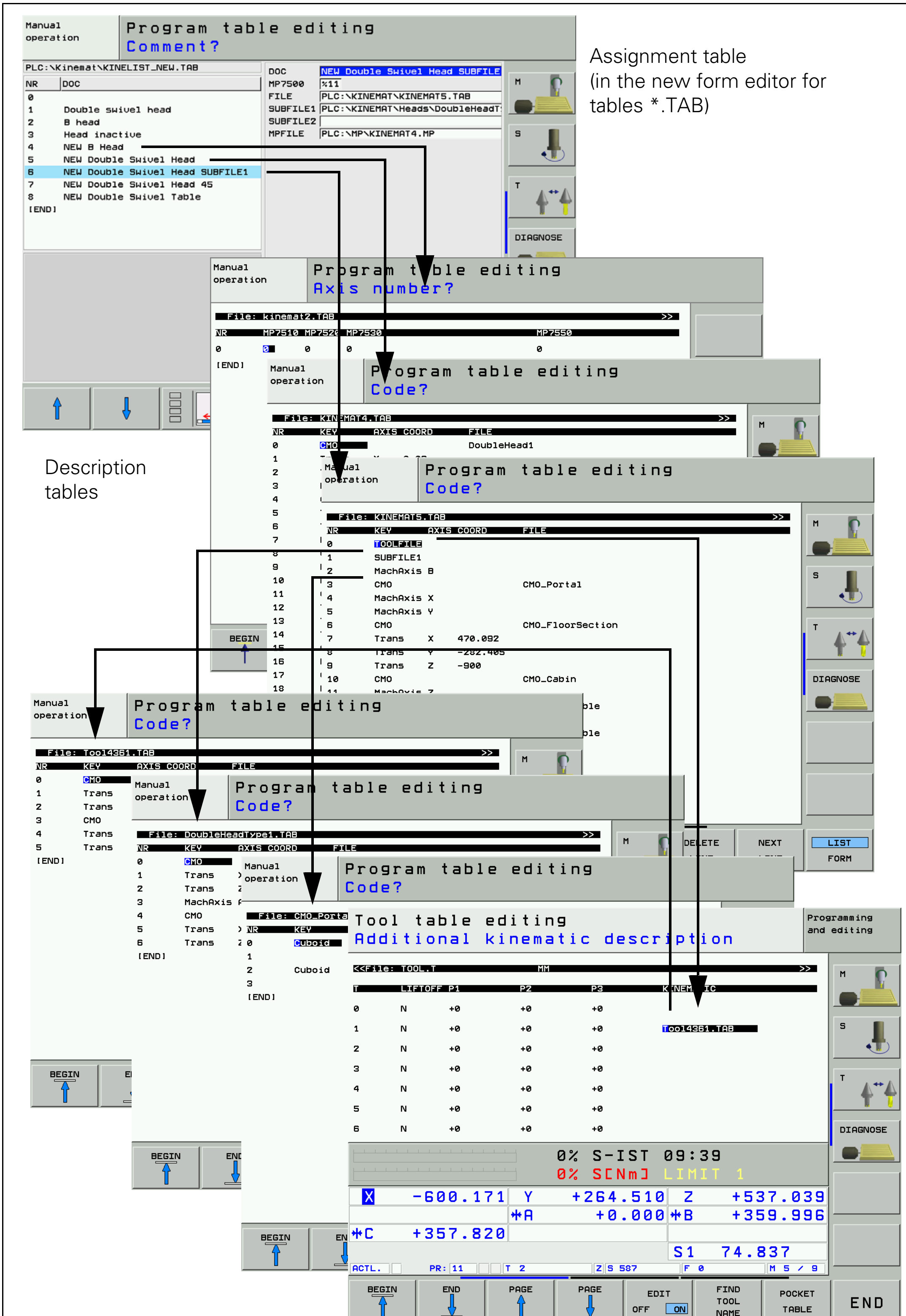
Compatibility

Basically it is possible to keep the previous kinematics descriptions in tabular form, and also for them to be available for selection in the **assignment table** parallel to the new format. Additional columns (such as **SUBFILE1**) for the new format are simply left blank in the assignment table, or, if you keep the kinematics descriptions in only the old format, you can delete them from the assignment table (**EDIT FORMAT** soft key).

Kinematics tables

In order to describe the kinematics of a machine, various tables are necessary depending on the application and the software option (option #40 "DCM – Dynamic Collision Monitoring"). Depending on the type of content, these tables are structured according to the following hierarchy when using the new kinematics description.

Table overview	Brief description
■ Assignment table	List of selectable kinematics, is called in OEM.SYS
■ Description table 1 (FILE) ■ Description table 2 (FILE)	Contains the transformation chain of the kinematics description, is referenced in the assignment table
■ Tool-carrier table (TOOLFILE)	Contains the kinematics description of a specific tool carrier, is called in the description table and referenced via the tool table
■ Partial description table (SUBFILE1, SUBFILE2)	Contains the kinematics description of an exchangeable machine component that is included in the kinematics (e.g. swivel head), is called in the description table and referenced in the assignment table
■ Definition table for collision-monitored objects (CMO)	Contains descriptions of objects located within the machine work envelope, and for which the possibility of collision with other machine objects cannot be ruled out. CMO ("Collision Monitored Object")


Assignment table
(in the new form editor for
tables *.TAB)
Description
tables
Program table editing
Comment?
Axis number?
Code?
Tool table editing
Additional kinematic description

> **Note**
>
> Definition tables, and therefore collision-monitored objects, are only processed and therefore considered on an MC 422B starting from software 340 49x-02 and with software option #40 "DCM – Dynamic Collision Monitoring" (ID 526 452-01).

Specify the machine kinematics as well as the resulting transformation model, starting from the machine reference point (REF 0, e.g. traverse block with M91).

- For machines with exchangeable components (e.g. adapter spindles, angle heads), specify partial kinematics as necessary. These can be included separately, in order to additionally increase the flexibility of the machine kinematics. Partial kinematics of this type are connected with the appropriate path in columns **SUBFILE1** or **SUBFILE2** of the **assignment table.**
- For tools with special tool-carrier kinematics, consider the possibility of connecting their additional kinematics descriptions via the tool table. Additional kinematics descriptions for tool carriers are connected in the **KEY** column of the **assignment table** via the **TOOLFILE** entry.
- Position the axes to the machine datum (transformation model with consideration of MP960).
- Now, starting from the tool reference point, define in sequence the shifts or rotations via the linear, rotary and tilting axes to a center point of a (rotary) table.
- Enter the shifts and rotations in the **description table.**

As a rule, the control takes changes in the mechanical offset into account, meaning that these changes do not have to be compensated with a PLC datum shift.

> **Note**
>
> - File paths should be entered as absolute paths. If only the file name is entered, the kinematics path from OEM.SYS is valid (e.g. KINEMATIC = PLC:\Kinemat\Kinelist.TAB).
> - Another **SUBFILE** or **TOOLFILE** cannot be called from a **SUBFILE** or **TOOLFILE,** respectively
> - File names can be entered with paths and the extension (.TAB) in the SUBFILE1, SUBFILE2 and FILE columns. If the path is missing, the path of the assignment table is used. If the file name extension is missing, *.TAB is amended (internally).

Assignment table

The assignment table "manages" the kinematics descriptions, and is the foundation for the selection dialog box called via the **KINEMATIC** keyword. The operator selects the kinematics description necessary for the current machine configuration from this dialog box. Normally only one assignment table is needed on a control.

In addition, by connecting partial description tables **(SUBFILE1** and **SUBFILE2),** the assignment table makes it possible to easily manage different machine configurations in just one assignment table. The iTNC ignores the missing entry of a SUBFILE in the assignment table, even if a **SUBFILE1** or **SUBFILE2** is entered in the description table. By combining the SUBFILE entries in the assignment table, you can cover four configurations with just one description table.

This makes the following combinations possible in the assignment table with one description table (here **KINEMAT1.TAB**):

NR	MP7500	FILE	MPFILE	SUBFILE1	SUBFILE2
0	1	PLC:\KINEMAT1.TAB			
1	7	PLC:\KINEMAT1.TAB	PLC:\MP\KINE...	PLC:\HeadA	
2	3	PLC:\KINEMAT1.TAB	PLC:\MP\KINE...	PLC:\HeadA	PLC:\HeadB
3	%11	PLC:\KINEMAT1.TAB	PLC:\MP\KINE...		PLC:\HeadB
[END]					

Specification of the assignment table:

- Switch to the **Programming and Editing** operating mode, press the MOD key and enter the code number 807667.
- Choose the file PLC:\OEM.SYS from within program management.
- Enter the code word **KINEMATIC=**, followed by the file name with its complete path from the assignment table.
  (e.g., **KINEMATIC= PLC:\KINEMAT\KINELIST.TAB**)
- Leave OEM.SYS by pressing the END key.

Creation of the assignment table:

- In program management, switch to the directory already selected via OEM.SYS, and enter the name of the assignment table specified in OEM.SYS, followed by the extension .TAB.
- Choose the table format with the MP7500, FILE, MPFILE, SUBFILE1, SUBFILE2 and DOC fields.
- In each line, enter the value from MP7500.x (integer or bit-encoded) for the information for tilting the working plane.
- In the **FILE** column, enter the absolute path to the description table.
- If necessary, enter in the **MPFILE** column an appropriate machine parameter subfile.
- If necessary, enter in the **SUBFILE1** or **SUBFILE2** columns a partial description file.

- Enter a brief descriptive text for the kinematics selection dialog.
  The texts displayed for selecting the kinematics can be created as language-sensitive texts (even with UTF8 character set). The window for selecting the kinematics is opened after the keyword KINEMATIC is entered.
  In the **DOC** column of the kinematics table, you enter a string that consists of the following elements: "**#<line number of DIALOG.A>**". Depending on MP7230.x, the texts is fetched from the specified line in DIALOG.A, and displayed in the selection window. The line numbers of DIALOG.A are identified by the preceding character **#**.
  At present, utf8 characters cannot be displayed in the selection window.
  If the entry in the **DOC** column is used as a keyword to make changes to the kinematics by using **WRITE TO KINEMATIC,** the "**#<line number of DIALOG.A>**" can be entered as KEY.
  Example:
  WRITE TO KINEMATIC AT COLUMN "HEIGHT" CAPTURE "DOC" KEY "#143" = 22

Each line of the assignment table contains information for referencing a specific kinematics description. Format of the assignment table:

Column	Input	Description
NR	0, 1, 2 ...	Automatic line numbering
MP7500	Integer (e.g. "3") or bit-encoded (e.g. %11)	Option selection for tilting the working plane (depends on the activation of the preset table). See the description of the "Tilt Working Plane" feature.
FILE	Path and file name[a]	Referencing of a description table for one machine kinematics
MPFILE	Path and file name[a]	Referencing of a machine parameter subfile, which is to be activated when these machine kinematics are selected. As of 340 490-06, relative path can also be used when entering MP subfiles in the **MPFILE** column. The relative path always starts with the folder in which the assignment table is saved.
SUBFILE1 SUBFILE2	Path and file name[a]	Referencing of two possible partial description tables for each (selected) kinematics description. These tables are activated via the **KEY** column in the description table and the **SUBFILE1** or **SUBFILE2** keyword.
DOC	Text	Brief descriptive text for selection dialog of the kinematics model via code number

a. File paths should be entered as absolute paths. If only the file name is entered, the kinematics path from OEM.SYS is valid (e.g. KINEMATIC = PLC:\Kinemat\Kinelist.TAB).

Example of an assignment table:

NR	MP7500	FILE	MPFILE	SUBFILE1	DOC
0	1	PLC:\KINEMAT1.TAB			Standard
1	7	PLC:\KINEMAT2.TAB	PLC:\MP\KINEMAT2.MP		B Head
2	3	PLC:\KINEMAT3.TAB	PLC:\MP\KINEMAT3.MP	PLC:\DoubleHeadTyp1	Double Swivel H1
3	%11	PLC:\KINEMAT4.TAB	PLC:\MP\KINEMAT4.MP	PLC:\DoubleHeadTyp2	Double Swivel H2
[END]					

With MP7506 you can make an advance selection of which entry in the kinematics table is to become active when the control is booted. Selection is made by entering the line number of the desired kinematics in **KINELIST.TAB.**

MP7506 Selection of kinematics at booting of the control
Input: 0 to 999
–1: Function not active

Note

Please note that with this feature, the same kinematics are always active whenever the control is booted. If you change the actual kinematics of the machine before booting, this is not taken into account. The kinematics set via MP7506 are still active after booting even after this change.

Description table

The description table contains the actual definition of the kinematics. The transformations are described here, subkinematics (SUBFILE1, SUBFILE2, TOOLFILE) are connected if necessary, and if software option #40 (DCM - "Dynamic Collision Monitoring") is available, objects for collision monitoring are included. Any number of description tables can be used.

Note

- The number of kinematics entries in a description table **(Trans, MachAxis)** is limited to 50. This also includes the entries in **TOOLFILE, SUBFILE1** and **SUBFILE2.** However, references to collision bodies **(CMO)** do not count as kinematics entries.
- For a correct description of the kinematics, the machine axes X, Y and Z must form a Cartesian coordinate system.
- HEIDENHAIN offers the KinematicsDesign software tool. KinematicsDesign is a program for interactive creation of control kinematics, fixture templates, completed fixtures and tool carrier kinematics.

Creating/activating/changing an assignment table:

- In program management, switch to the desired directory and enter the name of the description table, including the extension .TAB.
- Choose the table format with the KEY, AXIS, COORD, ON/OFF, FILE, DONTTEST, TEMPCOMP and DOC fields.
- With the aid of the table format for a description table shown below, enter the description of the machine kinematics.
- Activate the description table by transferring the row numbers from the assignment table
  - from the PLC with Module 9097
  - from the NC with **FN17: SYSWRITE ID290 NR1** = ... (line number)
- You can ascertain the active description table in two ways:
  - With the PLC you can use Module 9098 to ascertain the name of the description table or the line number in the assignment table.
  - With the NC you can use **FN18: SYSREAD Q1 = ID290 NR1** to ascertain the line number in the assignment table.

Note

The active description table is indicated with the status "M" in program management.

Time at which changes to the descriptions become effective:

- the iTNC is restarted
- the kinematics are selected via the code number **KINEMATIC**
- the kinematics are activated with **FN17: SYSWRITE ID290 NR1** = ...
- the kinematics are selected with PLC Module 9097
- the description table is changed with **WRITE TO KINEMATIC**

The description table contains the actual kinematics description. Format of the description table:

Column	Input	Description
NR	0, 1, 2 ...	Automatic line numbering
KEY	■ TOOLFILE	The additional kinematics description referenced in the tool table via the entry in the **KINEMATIC** column, such as a tool carrier, is included in the calculations.
	■ SUBFILE1 ■ SUBFILE2	The additional description is included in a partial kinematics table, which is referenced via the file in the **SUBFILE1** or **SUBFILE2** column of the assignment table. Basically, the same content conditions apply for subfiles as for the description tables.
	■ MachAxis	Defines the point at which a machine axis becomes effective in the kinematics sequence. From this entry onward, movements by the indicated axis (**AXIS** column) change the positions of previously defined axes or objects relative to subsequent axes or objects. The start of the sequence is always the tool reference point.
	■ MachBase	Defines the stationary part of the machine (machine base) for the graphic simulation in KinematicsDesign. All movements of a machine are simulated around the stationary part of the machine. MachBase only affects graphic simulation in KinematicsDesign. However, HEIDENHAIN recommends entering "MachBase" at the respective position.
	■ CMO	Defines a monitorable collision body in the current kinematics sequence. This machine element is described by the file (table) referenced in the **FILE** column.
	■ ToolAxis	Entering an axis (X, Y, Z, U, V, W) defines the tool axis. Only this axis can be selected with a **TOOL CALL**, see page 737.
	■ Trans	Here a transformation of the indicated axis (**AXIS** column) in the current kinematics sequence by the value entered in the **COORD** column is performed. This can be a linear translation as well as a rotation about an axis.
AXIS	X, Y, Z, A, B, C ...	Entry of the axis designation for function given in the **KEY** column (valid for the **MachAxis** and **Trans** parameters in the **KEY** column).

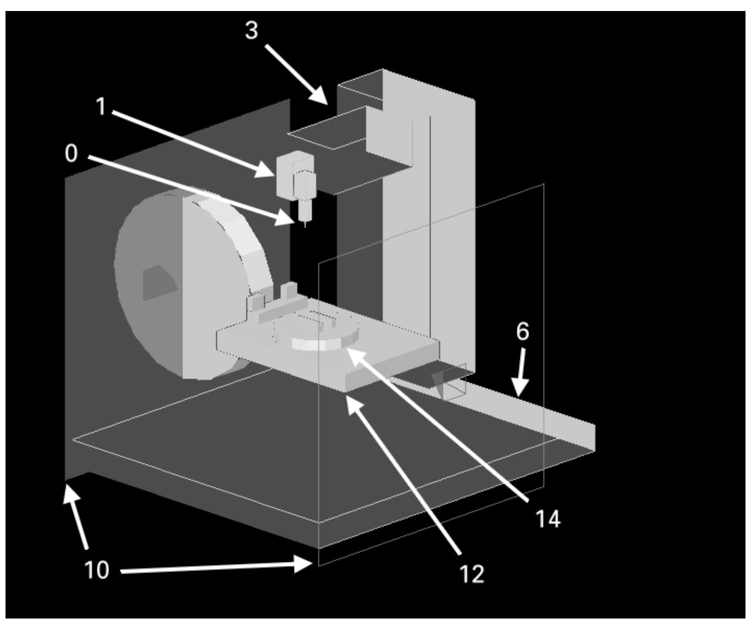
Column	Input	Description
COORD	e.g. 47.092 [mm] or 45.05 [°]	Entry of the transformation value for linear axes (X, Y, Z, ...) in mm or for rotary axes (A, B, C) in degrees. The units are not entered. The iTNC infers the units from the axis designation entered (valid for the **Trans** parameter in the **KEY** column).
ON/OFF	0 or 1 (1=CMO inactive)	Activate or deactivate a definition table for a collision-monitored object (valid for the **CMO** parameter in the **KEY** column). This makes it possible, for example, for a tool-change macro to switch off monitoring of the cabinet.
FILE	Path and file name[a]	Entry of the path and file name of a definition table for a collision-monitored object (valid for the **CMO** parameter in the **KEY** column).
DONTTEST	Path and file name[a]	Entry of the path and file name of a collision-monitored object (CMO), which is not to be monitored for collision with the CMO referenced in the **FILE** column (valid for the **CMO** parameter in the **KEY** column).
TEMPCOMP	Formula, e.g. 3.45e–3 * (W486 – 20)	Entry of a formula for a permanently effective temperature compensation for the translation given in the line. For temperature compensation, see page 697.
DOC	Comment	Entry of a comment

a. File paths should be entered as absolute paths. If only the file name is entered, the kinematics path from OEM.SYS is valid (e.g. KINEMATIC = PLC:\Kinemat\Kinelist.TAB).

The following example of a description table is intended to show the procedure for determining the kinematics of a machine.

NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST	TEMPCOMP
0	TOOLFILE						
1	SUBFILE1						
2	MachAxis	B					
3	CMO				CMO_Portal		
4	MachAxis	X					
5	MachAxis	Y					
6	CMO				CMO_FloorSection	PLC:\Kinemat\...	
7	Trans	X	470.092				1.73e–3*(W486-20)
8	Trans	Y	-282.405				0.82e–3*(W488-20)
9	Trans	Z	-900				2.3e-3* (W490-20)
10	CMO			1	CMO_Cabin	PLC:\Kinemat\...	
11	MachAxis	Z					
12	CMO				CMO_LiftTable	CMO_Cabin	
13	MachAxis	C					
14	CMO				CMO_TurnTable	CMO_LiftTable	
[END]							

Representation of the machine by the description table above



Tool-carrier kinematics table (TOOLFILE)

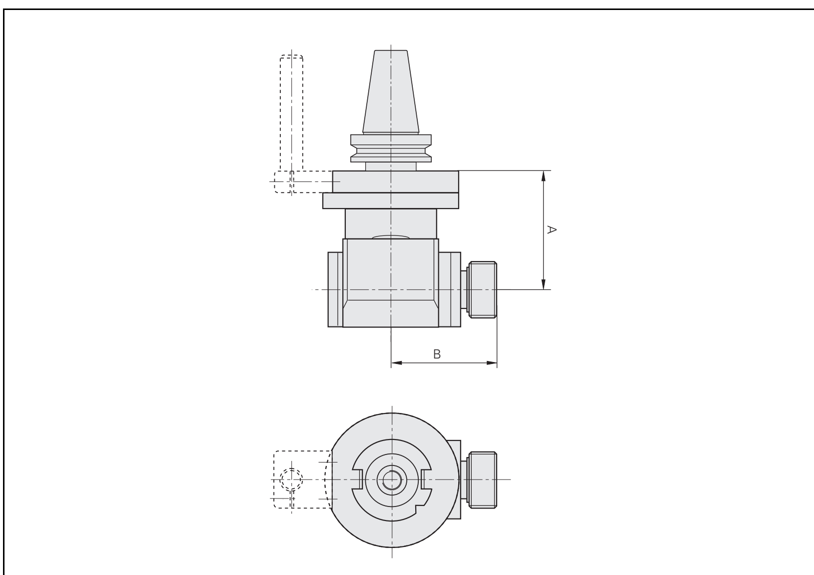
The **TOOLFILE** table is used to automatically activate specific tool-carrier kinematics when a tool-change procedure occurs. The inclusion of a **TOOLFILE** table adds an additional shift of the tool-reference point (reference point for tool dimensions) to the subsequent description of the kinematics. This occurs dynamically with the tool change.

For example, if the following **TOOLFILE** is included for an angular milling head (tool is aligned in the X direction) in the **KINEMATIC** column of the tool table, then the previous tool-reference point is shifted in the X direction by 96 mm and in the Z direction by –105 mm, and the tool alignment is rotated in the B direction by –90°. This additional kinematics description is added to the existing kinematics. This corresponds to the offset that this tool carrier would cause in the current kinematics representation.

NR	KEY	AXIS	COORD	ON/OFF	FILE
0	Trans	B	90		
1	Trans	X	-96		
2	Trans	Z	105		
3	CMO				CMO_Toolcarrier
[END]					

Example:

A = 105 mm, B = 96 mm, tool alignment in X axis



The tool holder of the tool carrier stated above (tool reference point) is now the initial point for viewing the kinematics with the included **TOOLFILE.** The subsequent kinematics description is based on this. This means that the algebraic signs of the shift are now determined by the new tool reference point.

The tool dimensions (e.g. tool length) in the tool table are automatically included in machining from this point on.

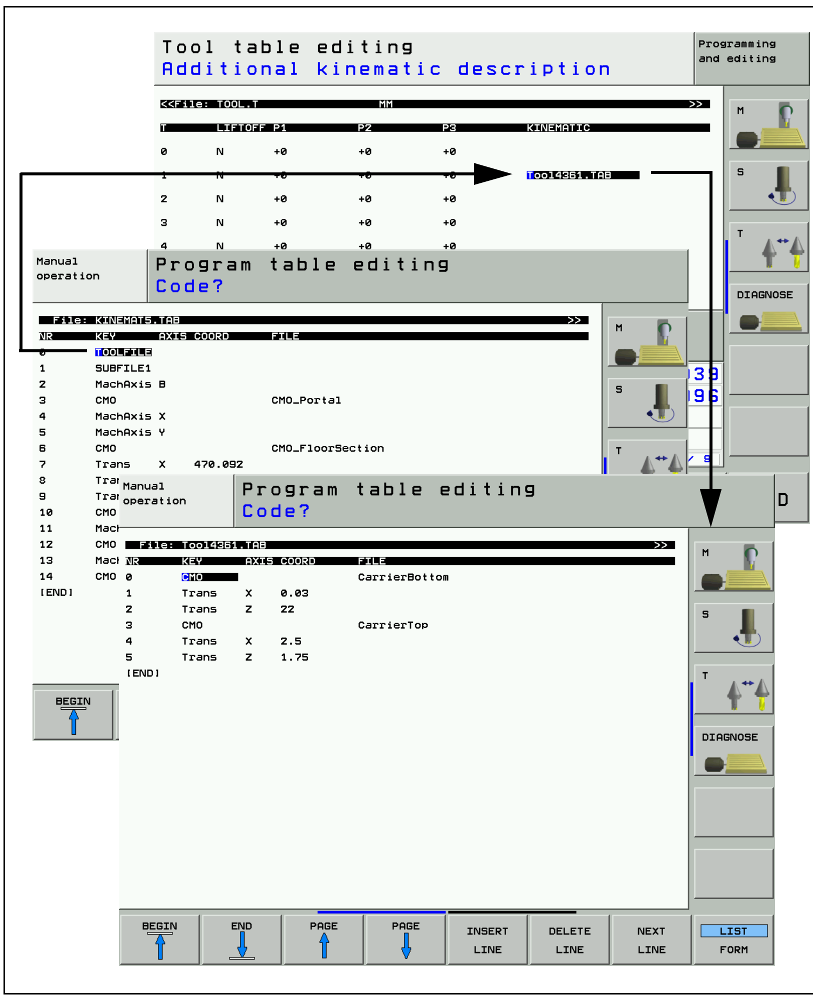
Transformations can be entered in the **TOOLFILE** table. If software option #40 has been activated (as in the table above), then collision objects (CMO) can also be entered.

The structure of these tables is identical to those of the description tables, but with the restriction that **TOOLFILEs** and **SUBFILEs** cannot be included in them.

The kinematics description of a **TOOLFILE** becomes active at the point from which it is included within the description table. Since the tool is at the beginning of the transformation chain, a **TOOLFILE** is only useful if it is included in the first line of the description table.

If no tool-carrier kinematics are referenced in the tool table (no entry in the **KINEMATIC** column), then the **TOOLFILE** entry in the description table has no effect.

In the following representation, the **TOOLFILE** referenced via the **KINEMATIC** column of the tool table is included in the description table.



Partial description table (SUBFILE)

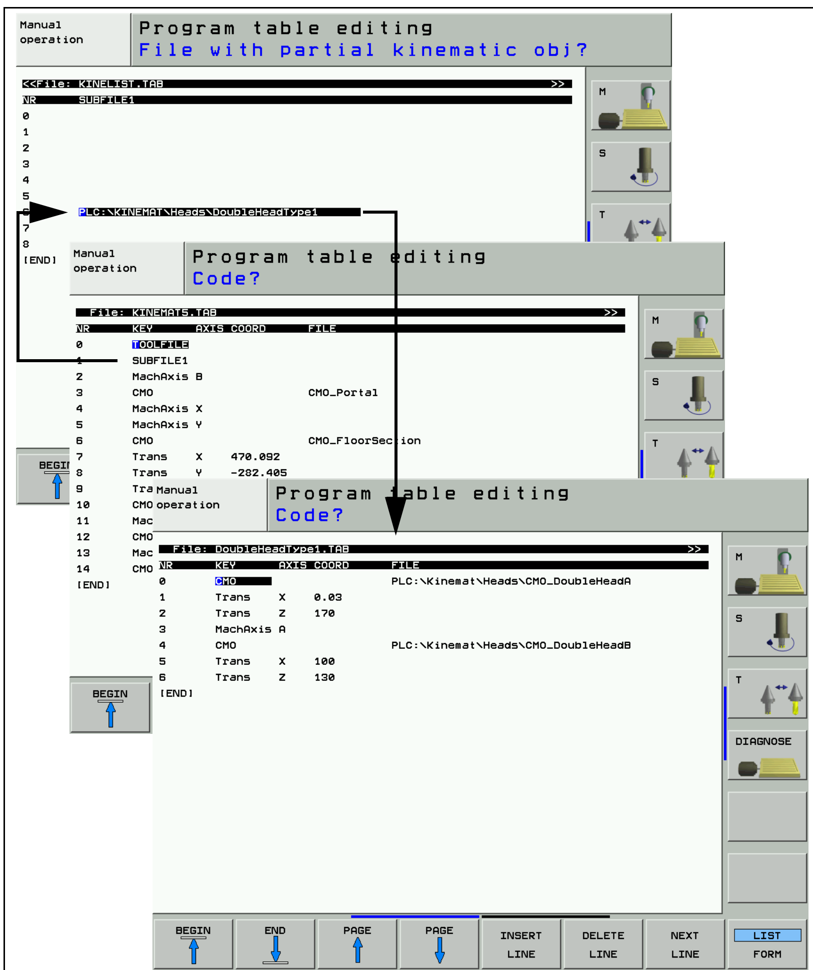
The **SUBFILE** makes it possible to integrate the kinematics of entire machine components (such as swivel heads and machine tables) from an external description table. The structure of these tables is identical to those of the description tables, but with the restriction that **TOOLFILEs** and **SUBFILEs** cannot be included in them.

Transformations, machine axes and collision objects (CMO) can be entered in the **SUBFILE.**

The kinematics description of a **SUBFILE** becomes active at the point from which it is included within the description table. However, please note that the **SUBFILE** is already referenced in the assignment table via the **SUBFILE1 or SUBFILE2** column.

If no partial description table is entered in the SUBFILE1 or SUBFILE2 columns of the assignment table, then any **SUBFILE1 / SUBFILE2** entry in the description table has no effect.

In the following representation, the **SUBFILE1** referenced via the assignment table is included in the description table.



Specifying the tool axis in the kinematics

As of software 340 49x-03 it is possible to specify a permitted tool axis in the kinematics table. This way you can permit only a certain tool axis to be selected with a **TOOL CALL.**
A tool axis is defined in the kinematics table with the **ToolAxis** keyword in the **KEY** column and by indicating an axis (X, Y, Z, U, V, W). As an alternative, the tool axis can also be defined in a SUBFILE or TOOLFILE.
If a different tool axis is selected with a **TOOL CALL,** the **Tool axis not allowed** error message is output and an NC stop is triggered. The error message is generated as well if no tool axis is programmed in the **TOOL CALL** and no tool axis has been defined modally.

So that the kinematics can be switched, the tool axis is not checked during the switch. This makes it possible to change the tool axis by switching the kinematics. The axes can be traversed without a subsequent **TOOL CALL** to activate the new, permitted tool axis, but the entry under **ToolAxis** and the active tool axis are not checked.

Attention

The following must be kept in mind when specifying a tool axis:

- After the kinematics have been switched, checking of the tool axis must be activated with a **TOOL CALL**.
- A tool axis may only be defined once via **ToolAxis** in the machine kinematics. If **ToolAxis** is defined more than once, then only the first instance is used.
- If the **ToolAxis** is to be switched via the kinematics tables, then the definition of the tool axis should be performed in the various SUBFILEs. This way you avoid multiple definitions in one kinematics description.

3-D basic rotation

Append 3-D basic rotation to kinematics

As of software 340 49x-04 it is possible to append a 3-D basic rotation to the end of the kinematics description, e.g. for chucking equipment.

When a rotation is appended to the kinematics, the coordinate system of the machine table is rotated with respect to the coordinate system of the machine axes. The preset vector from the preset table is always interpreted as a vector of the coordinate system of the machine table. As a result, the rotations must first be appended to the kinematics, and then the reference point must be determined. The working plane must be correspondingly tilted for the determination of the reference point.

The following entries must be appended to the end of the kinematics description (e.g. after the description of the rotary table):

```
Trans C ...
Trans B ...
Trans A ...
```

Then you can use the WRITE TO KINEMATIC function to enter the coordinates of the rotation in the kinematics table. The rotation becomes effective when the PLANE function is activated.

Use Cycle 431 "Measure plane" to determine the spatial angles for the rotation. Then the coordinates of the rotation must be entered in the kinematics description in the sequence described above. The coordinates of the rotation only correspond to the spatial angles of the kinematics description if they have been entered in the described sequence. Please also note that the first two measurement results of Cycle 431 define the orientation of the principal axis (principal axis X if tool axis Z). It may be necessary to align the workpiece in the plane before determining the rotation with Cycle 431. To do this, use Touch Probe Cycle 403.

Kinematics – reading and editing

The READ KINEMATICS and WRITE KINEMATICS functions are used to read and modify elements of the currently active kinematics in the NC program. In the NC program you cannot define these function FN17 (soft keys: program functions, kinematic functions) until you have entered the code number 555 343. With the iTNC 530, transformations and geometry axes are elements of the transformation chain.

READ KINEMATICS

Syntax:
```
READ KINEMATICS NOTE_QS0 MODE Q0 RESULT_Q1 KEX "KEY;AXIS" TO QS1
```

- NOTE:
  The function saves a "pointer" in this string variable. You need this pointer if you want to use WRITE KINEMATICS to describe these transformation elements again.
- MODE:
  0: Increment read-pointer
  1: Set the read-pointer to the beginning of the transformation chain
- RESULT:
  In "Result", the function provides information about possible errors.
  0: No error
  1: Error
- KEY:
  The key words whose entries are to be read are transferred in "Key." With the iTNC 530, the key words match the column names of the transformation table. The individual key words are separated by a semicolon.
- TO:
  Keys are returned, separated by a semicolon. The sequence depends on the sequence of the "KEY" key words.

WRITE KINEMATICS

Syntax:
```
WRITE KINEMATICS NOTE_QS0 MODE Q0 RESULT_Q1 KEY "COORD" = QS50
```

- NOTE:
  Pointer to a certain transformation element. This pointer must first be determined with READ KINEMATICS.
- MODE:
  0: Increment read-pointer
  1: Set the read-pointer to the beginning of the transformation chain
- RESULT:
  In "Result", the function provides information about possible errors.
  0: No error
  1: Error
- KEY:
  The key words whose entries are to be overwritten are transferred in "Key." The individual key words are separated by a semicolon.
- =:
  A string that contains the new entries is to be transferred as value. The entries are separated by a semicolon. The sequence depends on the sequence of the "KEY" key words.

Kinematics tables via PLC

Module 9098 Finding the active geometry description

Module 9098 can find the name of the active description table and/or line number in the assignment table.

Call:

PS	B/W/D/K	<>String number for table name> (line number is also determined) –1: Find only line number, no name
CM	9098	
PL	B/W/D	<>Line number in the assignment table> –1: Line number not found

Error recognition:

Marker	Value	Meaning
M4203	0	Name and/or line number found
	1	Error code in W1022
W1022	2	Incorrect parameter for string number
	20	Module was not called in a spawn or submit job

Module 9097 Selecting the geometry description

A geometry description from an assignment table can be chosen with Module 9097. The module can be called in a running NC program only in connection with a strobe. The module must be called in a submit job or spawn job, and cannot be cancelled with the CAN command.

Call:

```
PS    B/W/D/K   <>Line number in the assignment table>
PS    B/W/D/K   <>Mode, reserved>
                Transferred value must be 0
CM    9097
PL    B/W/D     <>Error condition>
                0 = Geometry description was selected
                1 = Invalid mode
                2 = Line was not found in the assignment table
                3 = Assignment table is not defined
                4 = Description table does not exist
                5 = Description table is incomplete
                6 = Module was not called in a spawn job or submit job
                7 = Call during running NC program without strobe
                8 = No KINEMATIC= entry in the OEM.SYS file
                9 = Error in the MPFILE column
                10 = Error in the MP7500 column
                11 = Error in the machine parameter subfile
                12 = Reserved
                13 = Names of collision objects (CMO) not unique
```

Error recognition:

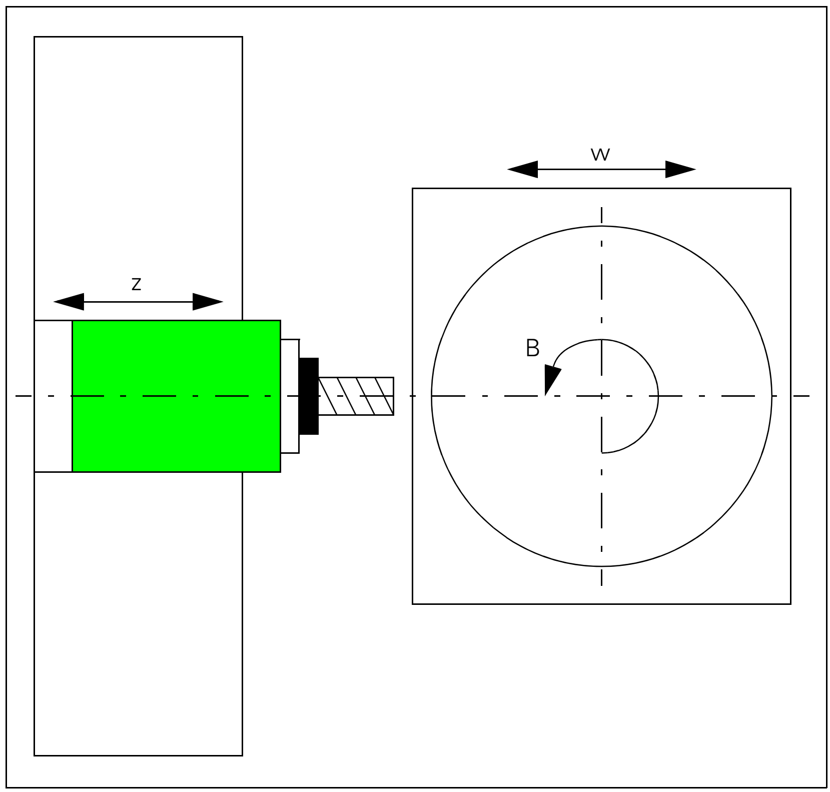
Marker	Value	Meaning
M4203	0	Geometry description was selected
	1	Error code in W1022
W1022	2	Invalid mode; or line was not found in the assignment table; or description table was not defined, does not exist or is incomplete; or there is no KINEMATIC= entry in the OEM.SYS file.
	9	Error in the MPFILE column
	10	Error in the MP7500 column
	11	Error in the machine parameter subfile
	20	Module was not called in a spawn job or submit job
	21	Call was made during a running NC program without a strobe

Overwriting the geometry description

Individual cells of the description table can be overwritten from within a machining program.

- Enter the code number 555343.
- Press the **ENTRY IN KINEMATIC TABLE** soft key.
- With the **WRITE TO KINEMATIC AT COLUMN <column to be written to> CAPTURE <searched column> KEY <keyword> =<value>** command you can overwrite individual cells in the active kinematics description in the Program Run operating modes. The line is selected by searching the **CAPTURE** column for the entry **KEY.** The column **COLUMN** is overwritten with **<value>**.

Example 1:
Vertical machining center, B-rotary table in the W axis, machine quill in Z



The position of the W axis must be considered in the kinematics table in order to use the tilting functions. For example, an NC macro can be programmed before the tilting function. This macro includes the current position of the W axis in the calculations of the kinematics table.

The following requirements apply to the correct calculation of the W axis:

- The kinematics table is determined with a defined position of the W axis.
- The Z position to be corrected is marked in a newly created column, **DOC** for example.

The kinematics table in the previous format:

NR	MP7510	MP7520	MP7530	TEMPCOMP	DOC	ORG_VALUE	Comment:
0	1	3	100				Center point in X
1	4	3	120		WPOS	120	Home position
2	16	1	0				
[END]							

- The **ORG_VALUE** column is created in the description table. Here the position of the table center point is entered when the W axis is at position 0 (in the REF system).
- Before the tilting function, the actual position of the W axis (in the REF system) must have been determined via the PLC, and the corresponding Q parameters (Q100 to Q197) transferred to the NC program.
- An NC macro that overwrites the corresponding line in the kinematics table is then carried out before the tilting function is performed.

	Comment:
0 BEGIN PGM M85 MM	
1 FN17: SYSWRITE ID290 NR1 = +3	Activate kinematics 3
2 **FN26: TABOPEN PLC:\KINEMAT\KINEMAT3.TAB**	Open KINEMAT3.TAB
3 **FN28: TABREAD Q10 = 1 / "ORG_VALUE"**	Read Z coordinate for home position of machine quill
4 **FN1: Q2 = +Q10 + +Q100**	Calculate new W position. Q100 contains the position of the W axis in the REF system (was set by the PLC)
5 WRITE TO KINEMATIC AT COLUMN "MP7530" CAPTURE "DOKU" KEY "WPOS" = +Q2	Overwrite cell in the kinematics table
6 **END PGM M85 MM**	

Example 2:
Deactivation of an active COM table so that collision monitoring is deactivated for a certain action, such as a tool change.

Table in the new kinematics description format before overwriting:

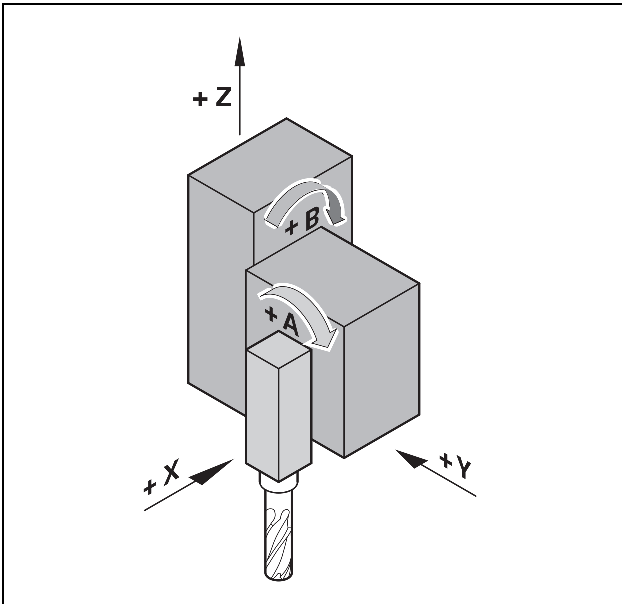
NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST	TEMPCOMP
:							
2	MachAxis	B					
3	CMO				CMO_Portal	PLC:\Kinemat\...	
4	MachAxis	X					
:							
[END]							

	Comment:
0 BEGIN PGM N545TCM MM	
:	
4 WRITE TO KINEMATIC AT COLUMN "ON/OFF" CAPTURE "FILE" KEY "CMO_Portal" = 1	Overwrite cell in the kinematics table
:	

Table after overwriting with collision monitoring switched off for the **CMO_Portal** object:

NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST	TEMPCOMP
:							
2	MachAxis	B					
3	CMO			1	CMO_Portal	PLC:\Kinemat\...	
4	MachAxis	X					
:							
[END]							

Example 1: Rectangular double swivel head



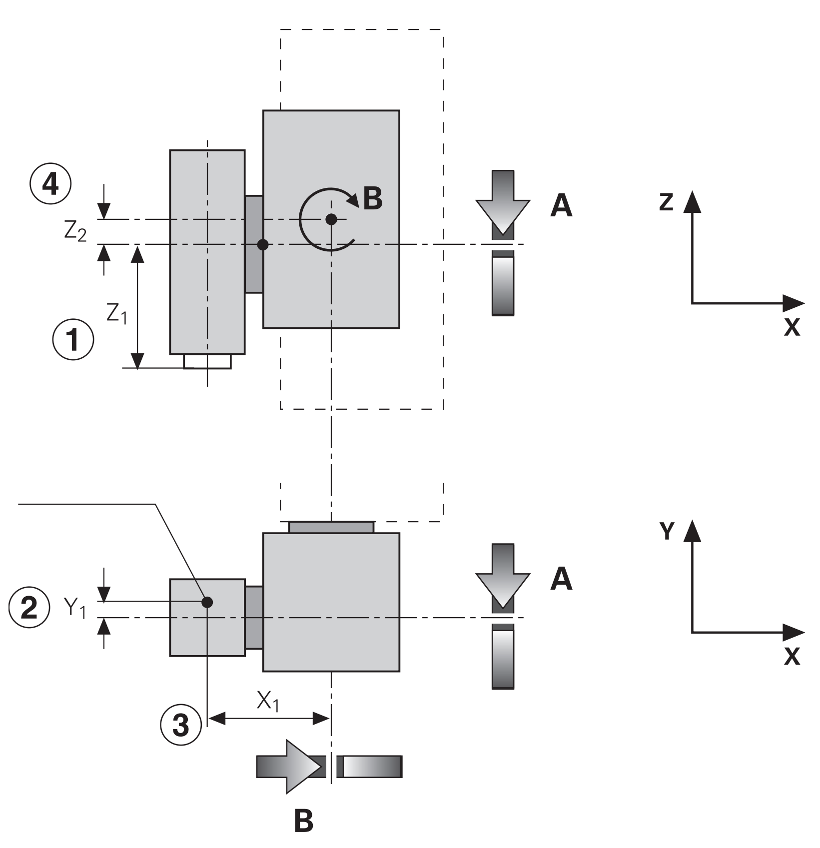
Z1 = 200.4 mm

Z2 = 3.1 mm

X1 = 201.5

Y1 = 1.9 mm

1 to 4:
Sequence of transformation



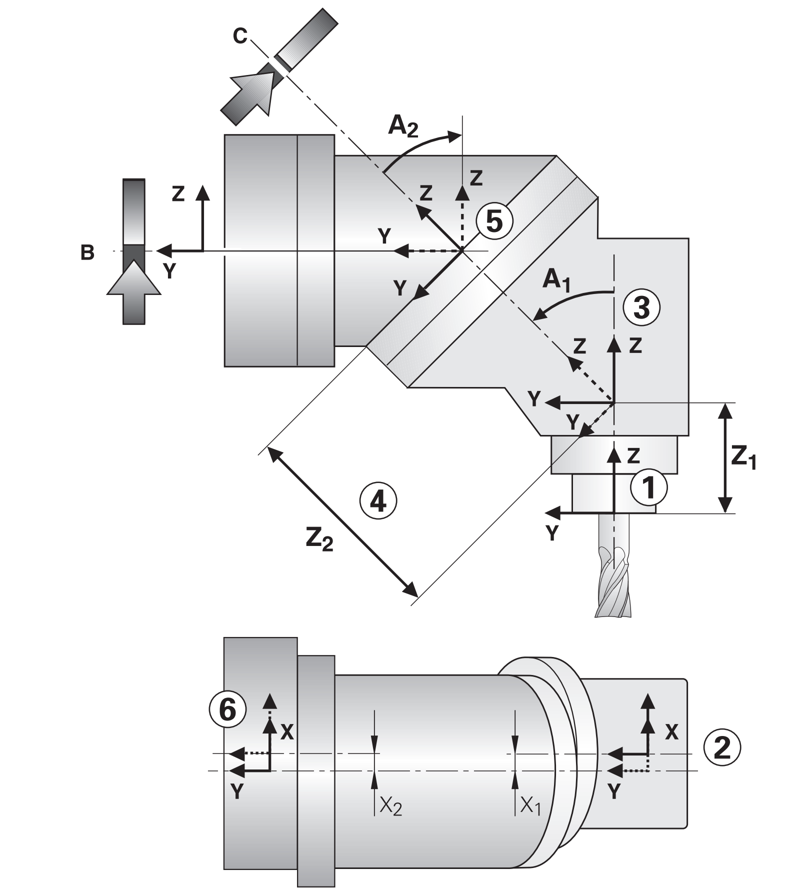
Description of the swivel head in the previous format via machine parameters (up to software 340 422-xx) or in a description table (as of software 340 490-01; MP numbers are column titles in the description table).

```
MP 7510.0    : %000100    ;Shift in Z axis (Z1)
MP 7510.1    : %000010    ;Shift in Y axis (Y1)
MP 7510.2    : %001000    ;Free tilting axis A
MP 7510.3    : %000001    ;Shift in X axis (X1)
MP 7510.4    : %000100    ;Shift in Z axis (Z2)
MP 7510.5    : %010000    ;Free tilting axis B
MP 7510.6    : %000000    ;End of the transformation chain

MP 7520.0    : %00        ;Incremental dimensions, swivel head
MP 7520.1    : %00        ;Incremental dimensions, swivel head
MP 7520.2    : %00        ;Incremental dimensions, swivel head
MP 7520.3    : %00        ;Incremental dimensions, swivel head
MP 7520.4    : %00        ;Incremental dimensions, swivel head
MP 7520.5    : %00        ;Incremental dimensions, swivel head

MP 7530.0    : +200.4     ;Dimension Z1
MP 7530.1    : –1.9       ;Dimension Y1
MP 7530.2    : +0         ;Variable dimension (free tilting axis A)
MP 7530.3    : +201.5     ;Dimension X1
MP 7530.4    : +3.1       ;Dimension Z2
MP 7530.5    : +0         ;Variable dimension (free tilting axis B)
```

Description of the swivel head as of software 340 490-02 (without elements for collision monitoring):

NR	KEY	AXIS	COORD	ON/OFF	FILE	DOC
0	Trans	Z	200.4			(Z1)
1	Trans	Y	-1.9			(Y1)
2	MachAxis	A				
3	Trans	X	201.5			(X1)
4	Trans	Z	3.1			(Z2)
5	MachAxis	B				
6	:					
:	:					
[END]						

Example 2: Double swivel head 45°



Z_1 = 150.5 mm

Z_2 = 251.5 mm

A_1 = 45°

A_2 = 45°

X_1 = 1.2 mm

X_2 = 0.8 mm

1 to 6: Sequence of transformation

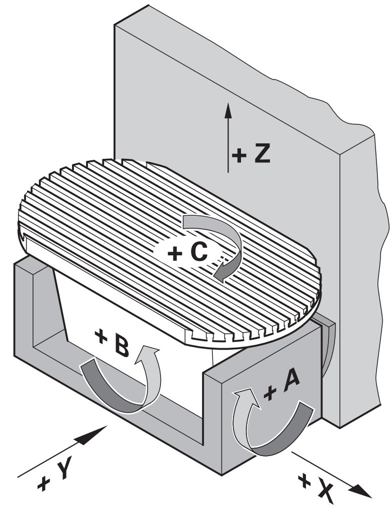
Description of the swivel head in the previous format via machine parameters (up to software 340 422-xx) or in a description table (as of software 340 490-01; MP numbers are column titles in the description table).

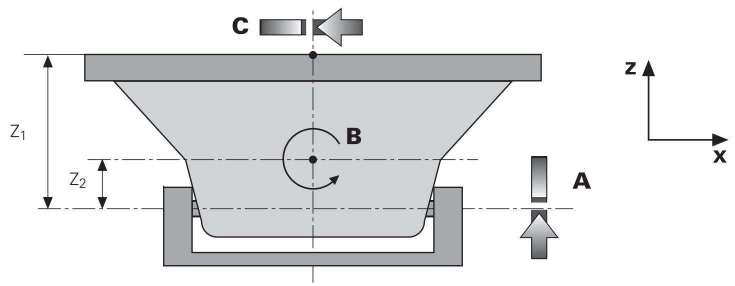
```
MP 7510.0    : %000100    ;Shift in Z axis (Z1)
MP 7510.1    : %000001    ;Shift in X axis (X1)
MP 7510.2    : %001000    ;Rotate coordinate system about axis A (A1)
MP 7510.3    : %000100    ;Shift in Z axis (Z2)
MP 7510.4    : %100000    ;Free tilting axis C
MP 7510.5    : %001000    ;Rotate coordinate system about axis A (A2)
MP 7510.6    : %000001    ;Shift in X axis (X2)
MP 7510.7    : %010000    ;Free tilting axis B
MP 7510.8    : %000000    ;End of the transformation chain

MP 7520.0    : %00        ;Incremental dimensions, swivel head
MP 7520.1    : %00        ;Incremental dimensions, swivel head
MP 7520.2    : %00        ;Incremental dimensions, swivel head
MP 7520.3    : %00        ;Incremental dimensions, swivel head
MP 7520.4    : %00        ;Incremental dimensions, swivel head
MP 7520.5    : %00        ;Incremental dimensions, swivel head
MP 7520.6    : %00        ;Incremental dimensions, swivel head
MP 7520.7    : %00        ;Incremental dimensions, swivel head

MP 7530.0    : +150.5     ;Dimension Z1
MP 7530.1    : –1.2       ;Dimension X1
MP 7530.2    : -45        ;Dimension A1
MP 7530.3    : +251.5     ;Dimension Z2
MP 7530.4    : +0         ;Variable dimension (free tilting axis C)
MP 7530.5    : +45        ;Dimension A1
MP 7530.6    : +0.8       ;Dimension X2
MP 7530.7    : +0         ;Variable dimension (free tilting axis B)
```

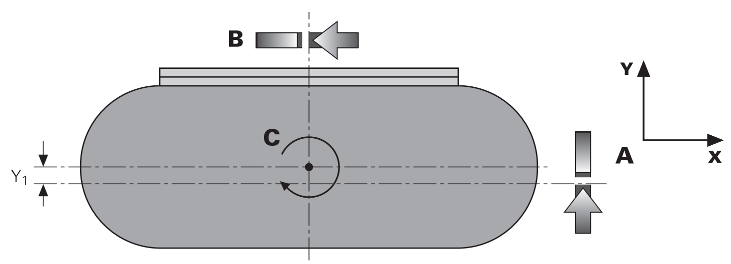
Description of the swivel head as of software 340 490-01 (without elements for collision monitoring):

NR	KEY	AXIS	COORD	ON/OFF	FILE	DOC
0	Trans	Z	150.5			(Z1)
1	Trans	X	-1.2			(X1)
2	Trans	A	-45			(A1)
3	Trans	Z	251.5			(Z2)
4	MachAxis	C				Free tilting axis C
5	Trans	A	45			(A2)
6	Trans	X	0.8			(X2)
7	MachAxis	B				
:						
[END]						

Example 3: Universal table (pitch, tilt, rotation)







Y_1 = 2.7 mm
Z_1 = 331.3 mm
Z_2 = 125.9 mm

Coordinates (with respect to machine datum) of the center of rotation of table C when all tilting axes are in their home position:

XR = 420.0 mm
YR = 151.2 mm
ZR = -395.4 mm

Description of the table in the previous format via machine parameters (up to software 340 422-xx) or in a description table (as of software 340 490-01; MP numbers are column titles in the description table).

```
MP 7510.0    : %000001    ;X coordinate of C-axis center of rotation
MP 7510.1    : %000010    ;Y coordinate of C-axis center of rotation
MP 7510.2    : %000100    ;Z coordinate of C-axis center of rotation
MP 7510.3    : %100000    ;Free tilting axis C
MP 7510.4    : %000010    ;Shift in Y axis (Y1)
MP 7510.5    : %000100    ;Shift in Z axis (Z1)
MP 7510.6    : %001000    ;Free tilting axis A
MP 7510.7    : %000100    ;Shift in Z axis (Z2)
MP 7510.8    : %010000    ;Free tilting axis B
MP 7510.9    : %000000    ;End of the transformation chain

MP 7520.0    : %11        ;Absolute dimension, tilting table
MP 7520.1    : %11        ;Absolute dimension, tilting table
MP 7520.2    : %11        ;Absolute dimension, tilting table
MP 7520.3    : %01        ;Tilting table
MP 7520.4    : %01        ;Tilting table
MP 7520.5    : %01        ;Tilting table
MP 7520.6    : %01        ;Tilting table
MP 7520.7    : %01        ;Tilting table
MP 7520.8    : %01        ;Tilting table

MP 7530.0    : +420       ;Dimension XR
MP 7530.1    : +151.2     ;Dimension YR
MP 7530.2    : -395.4     ;Dimension ZR
MP 7530.3    : +0         ;Variable dimension (free tilting axis C)
MP 7530.4    : –2.7       ;Dimension Y1
MP 7530.5    : –331.3     ;Dimension Z1
MP 7530.6    : +0         ;Variable dimension (free tilting axis A)
MP 7530.7    : +125.9     ;Dimension Z2
MP 7530.8    : +0         ;Variable dimension (free tilting axis B)
```

Description of the rotary table as of software 340 490-02 (without elements for collision monitoring) with assumed tool reference point:

NR	KEY	AXIS	COORD	DOC
0	Trans	X	420	Translation in X direction
1	Trans	Y	148.5	Translation in Y direction
2	Trans	Z	-190	Translation in Z direction
3	MachAxis	Z		
4	MachAxis	Y		
5	MachAxis	X		
6	MachAxis	B		
7	Trans	Z	-125.9	Translation in Z direction
8	MachAxis	A		
9	Trans	Y	2.7	Translation in Y direction
10	Trans	Z	331.3	Translation in Z direction
11	MachAxis	C		End of translation
[END]				

Compensation of offset of adapter spindle

It may happen that the current adapter spindle in the swivel head has a phase-angle error. This can be compensated as follows:

- Open the corresponding description table.
- Press the FORMAT EDITIEREN soft key
- Move the cursor to the END line and insert a new line by pressing the INSERT LINE soft key.
- Enter RAX_OFFS as **`Field name,`** C for **`Field type,`** 31 for **`Field width,`** 4 for the number of **`decimal places`** and a dialog text for the desired dialog languages, e.g. **`OFFSET of angular axes?`**
- Press the **`END`** key.

The new column RAX-OFFS has been added to the description table. In the first three lines of these columns, you can enter the phase-angle error of the adapter spindle.

- Line 0 corresponds to axis A
- Line 1 corresponds to axis B
- Line 2 corresponds to axis C

As soon as the description table has been activated, the phase-angle error is compensated.

> Note
>
> This function is limited to tables with the previous kinematics description.

Selecting a geometry description in case of an error

In order for the machine operator to be able to select another geometry description in case of an error, abbreviations for the geometry descriptions can be entered in the **`DOC`** column of the assignment table.

To select another geometry description in case of an error:

- While in the **`Programming and Editing`** operating mode, press the MOD key.
- Enter the code number **`KINEMATIC.`**

A pop-up window appears with the abbreviation from the **`DOC`** column in the assignment table:

- Use the arrow keys to select the appropriate geometry description.
- Press the ENT key.

The control resets and activates the selected geometry description.

> Note
>
> As of NC software 340 422-03 and 340 480-03, more than 15 entries can be shown in the pop-up window.

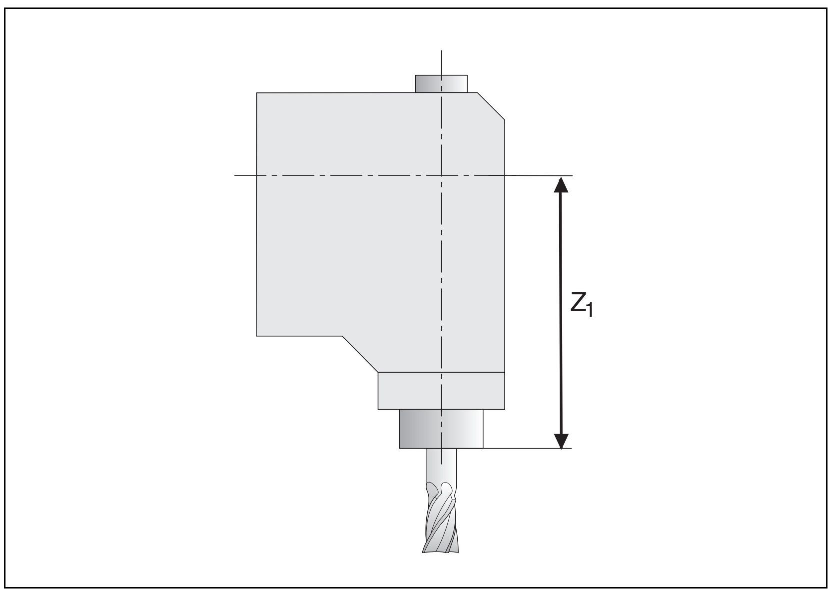
6.5.4 Temperature Compensation with Tilting Axes

A change in temperature always causes a change in length. For tilting axes, thermal growth of the spindle head must be compensated in the X, Y and/or Z axes.
There are two possibilities for temperature compensation:

- Temperature compensation only with "Tilt working plane" (not possible with the description tables in the new format)
  - by entering a formula in the **MP7530** column of the description table
- Permanently effective temperature compensation
  - by entering a formula in the **TEMPCOMP** column of the description table

In most cases, the formula to be used will be the formula for calculating a change in length: $\Delta\lambda = \lambda \langle \Delta T \langle \alpha$
Δl: Change in length
l: Length
ΔT: Change in temperature
α: Coefficient of expansion (steel: $11.5 \cdot 10^{-6}$ 1/K)

Example:



Z_1 = 300 mm (at 20 °C)
$\alpha_{steel} = 11.5 \cdot 10^{-6}$ 1/K (coefficient of expansion of steel)
W486: Temperature measured by a Pt100 thermistor

"MP7530.x" = 300 + 300 * 11.5e–6 * (W486 – 20)
"TEMPCOMP" = 300 * 11.5e–6 * (W486 – 20)
Better would be:
"MP7530.x" = 300 + 3.45e–3 * (W486 – 20)
"TEMPCOMP" = 3.45e–3 * (W486 – 20)

If the front part of the spindle gets warmer by 40 K, it results in a spindle length growth of

$$\Delta l = 300\ \text{mm} \cdot 40\ \text{K} \cdot 11,5 \cdot 10^{-6}\ \frac{1}{\text{K}} = 0{,}138\ \text{mm}$$

Constraints on the entry of a formula

- Maximum length of a formula: 31 characters
- Up to 16 variables in total
- Mathematical operations in lowercase letters, variables in uppercase letters
- The following operations are permitted in a formula:
  - Addition +
  - Subtraction –
  - Multiplication *
  - Division /
  - Logarithm to the base of 10 log10
  - Exponent ^
  - Parentheses ( )
  - Sine sin
  - Cosine cos
  - Tangent tan
  - Arc sine asin
  - Arc cosine acos
  - Arc tangent atan
  - Square root sqrt

Note

Faulty syntax in the formula is not yet detected in the kinematics tables in the new format (**TEMPCOMP** column), and when defined via MP7530, then only when an NC program is begun. The error message **MP75xx not defined** then appears.

Temperature compensation

If the "tilted working plane" function is active, the position of the tilting element is calculated for each positioning movement for the kinematics tables in the old format when temperature compensation is entered via MP7530.x.
The variables are monitored every second, and if there are any changes, MP7530.x is recalculated.

Note

Remember that the changes are compensated with a certain delay. Positioning blocks that have already been calculated can no longer be considered. This type of temperature compensation is no longer possible with the kinematics tables in the new format. For this, please use the permanent temperature compensation in the **TEMPCOMP** column of the description table.

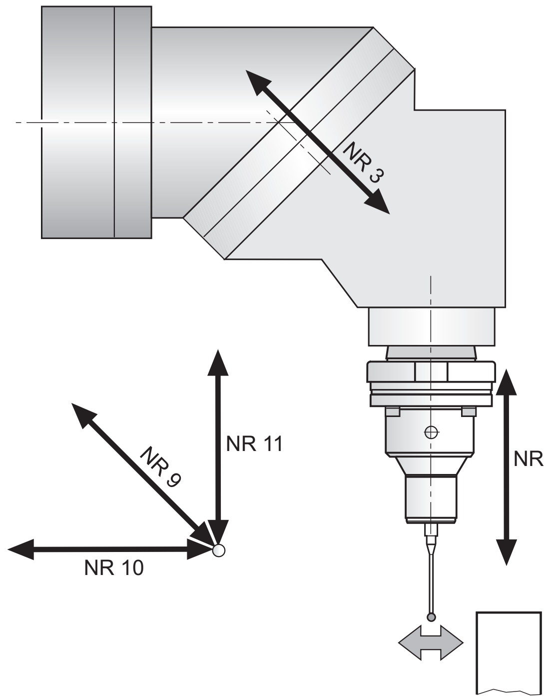
Permanent temperature compensation

The permanent temperature compensation with the formula from the **TEMPCOMP** column is only effective if the description table is active. The algebraic sign of the compensation must match that of the axis error compensation in Words W576 to W584.
The variables are monitored every second and changes are reported to the position controller. The position controller uses the formula in the **TEMPCOMP** column and the current angle of the rotary axes to calculate the compensation values.
For temperature compensation, only rows with an entry in the **TEMPCOMP** column are considered, including the entry **0.** Any missing entry (NO ENT) interrupts temperature compensation at this point, meaning that the subsequent rows will not be taken into account. If rows 1 to 7 in the following example contained no entries instead of the entry **0,** the temperature compensation would only be applied to the Z axis. **The tilting axes would not be considered.** You can use this interruption of the temperature compensation to reduce processing time if, for example, a swivel head and tilting table follow each other in a table, but the temperature compensation is only to be applied to the swivel head. You would then make no entry for the tilting table in the **TEMPCOMP** column.
With Module 9040 or 9041, transfer value 8, the value of the temperature compensation can be determined.
The formula is only entered for the transformations where compensation is to occur.

Example of a description table with permanent temperature compensation for a 45° double swivel head (also applies to the **TEMPCOMP** column in the description tables in the new format):

NR	MP7510	MP7520	MP7530	TEMPCOMP	MP7550
0	4	0	+150.5	1.73e–3*(W486-20)	0
1	1	0	–1.2	0	0
2	8	0	–45	-45	0
3	4	0	+251.5	0.0115*(W486-20)	0
4	32	0	0	0	0
5	8	0	+45	+45	0
6	1	0	+0.8	0	0
7	16	0	0	0	0
8	0	0	0	0	0
9	1	0	0	0.0345*(W486-20)	0
10	2	0	0	0.0230*(W486-20)	0
11	4	0	0	0.0173*(W486-20)	0
[END]					

The arrows in the figure show the effective directions of the entries in the column **TEMPCOMP**. **NR** x shows the respective line number of the entries:



6.5.5 Changing the Milling Heads

In order to change the milling heads, some new machine parameter values must be defined along with the new tilting axis geometry (e.g. in the assignment table via the **SUBFILE1** or **SUBFILE2** entries). For this purpose a machine-parameter subfile can be entered in the **MPFILE** column. The machine-parameter subfile contains the new axis configuration, the new axis motors, the assignments of the PWM outputs and the encoder inputs, etc. In this machine-parameter subfile there must be no machine parameters that provoke a control reset:

- In the standard machine parameter file, the bits in MP10 need to be set to 1 for all possible axes.

Note

If an axis deactivated in the standard machine-parameter file becomes activated in the machine-parameter subfile, the axis remains deactivated.

- Switch off drives for the affected axes.
- Choose a row in the assignment table in which a machine-parameter subfile is activated which sets the bits in MP10 to 0 for the affected axes.
If the encoder of a **digital** axis needs to be disconnected, then the appropriate bit must also be set in MP20.x bit x = 0. In addition, MP2200.x = "" must be set.

Note

On the CC 422, encoders with EnDat interface must not be disconnected and reconnected during operation, since the absolute value is only read when the control is started up. However, this is possible with the CC 424(B)!

- Change the milling head.
- Choose a row in the assignment table which contains a machine-parameter subfile for the new axes to set the bits in MP10 to 1 for the new axes.
If an encoder on a **digital** axis was disconnected and reconnected, then MP20.x and MP2200.x must be entered again correctly.
- Switch on drive for the new axis/axes.
- After the drive has been switched on, the affected motors should make at least one revolution.

6.5.6 KinematicsOpt

Accuracy requirements are becoming increasingly stringent, particularly in the area of 5-axis machining. Complex parts are required to be manufactured with precision and reproducible accuracy even over long periods.

KinematicsOpt (software option #48) is an important component that helps you meet these complex requirements: A 3-D touch probe cycle measures the rotary axes on your machine fully automatically, regardless of whether they are used in rotary-table or head configurations. A calibration sphere is fixed at any position on the machine table, and measured with a resolution that you define. In the cycle definition, you only have to define for each rotary axis the area that you want to measure.

From the measured values, the TNC calculates the static tilting accuracy. The software minimizes the spatial error arising from the tilting movement and, at the end of the measurement process, automatically saves the machine geometry in the respective machine constants of the kinematics table.

When using the cycles for KinematicsOpt, please also refer to the Touch Probe Cycles User's Manual.

Functions

- Testing the static tilting accuracy by probing different points on a calibration sphere with a 3-D workpiece touch probe.
- Optimizing the static tilting accuracy by adjusting the kinematics description.
- Backing up and restoring the kinematics data that can be changed with this software.

Prerequisites

- The software options #48 (KinematicsOpt) and #8 (software option 1) must be enabled.
- Feature Content Level (FCL) 3 or higher must be enabled.
- The 3-D touch probe used for the measurement must be calibrated.
- A calibration sphere with an exactly known radius and sufficient rigidity must be attached to the machine table.
- The kinematics must be described in the new table format (columns: **KEY**, **AXIS**, **COORD**, ...)
- The kinematics description of the machine must be complete and correct. The transformation values must be entered with an accuracy of approx. 1 mm.
- All machine axes involved must have adequate positioning accuracy.
- The geometry of the machine must have been measured.
- The machine datum (MP960.x) must be defined for the rotary axes (is not measured in the cycle).
- The machine parameters MP6600 to MP6602 must be defined.
- The cycles use the value entered in MP420 or MP430 to determine whether the respective axis is a rotary axis or a Hirth axis.
- Option #52 KinematicsComp, if rotational position errors of the rotary axes are to be compensated

Kinematics description

In the optimizing mode, the linear offsets (compensation translations) are adjusted to the kinematics description. Please note that two compensation transformations defining the position (not the angular position) of the rotary axis must be defined for every rotary axis. Starting with software version 340 49x-05, the TNC can find a suitable compensation translation even beyond two rotary axes. As soon as a translation is not parallel to the rotary axis to be optimized it can be accepted as a compensation translation.
If further axes lie between the rotary axis to be measured and the translation, then the translation must be parallel to the other rotary axes.

Transformations that do not have an entry in the COORD column are not changed by KinematicsOpt.

Example of a kinematics description with a B head and C table that is already **correct** for software version 340 49x-04:

NR	KEY	AXIS	COORD
xx	Trans	X	+100.014
xx	Trans	Z	+299.951
xx	MachAxis	B	
xx	Trans	Z	-600.142
xx	Trans	X	+300.021
xx	Trans	Y	-251.191
xx	MachAxis	C	

Example of a kinematics description with a B head and C table that is **incorrect** for software version **340 49x-04**: Incorrect position for the Y translation of the C axis. But **starting with 340 49x-05** this description is also permitted:

NR	KEY	AXIS	COORD	
xx	Trans	X	+100.014	
xx	Trans	Y	-251.191	Compensation translation for C
xx	Trans	Z	+299.951	
xx	MachAxis	B		
xx	Trans	Z	-600.142	
xx	Trans	X	+300.021	
xx	MachAxis	C		

Example of changes in the kinematics description with a B head and optional C table:

NR	KEY	AXIS	COORD		COORD (compensations)
0	TOOLFILE			-->	No changes
2	CMO			-->	No changes
3	Trans	X	+0.01	-->	+0.006
4	Trans	Y	+0	-->	+0
6	Trans	Z	+250.02	-->	+250.034
7	MachAxis	B		-->	No changes
8	MachAxis	Z		-->	No changes
9	MachAxis	Y		-->	No changes
10	MachAxis	X		-->	No changes
11	SUBFILE1	X		-->	No changes
12	Trans	X	+0	-->	No changes
[END]					

SUBFILE1:

NR	KEY	AXIS	COORD		COORD (compensations)
0	Trans	Z	-655.045	-->	No changes
1	Trans	X	+440.01	-->	+440.006
2	Trans	Y	+250.02	-->	+250.034
3	CMO			-->	No changes
4	MachAxis	C		-->	No changes
5	CMO			-->	No changes
[END]					

Tolerance limit

In MP6600 you can enter the maximum permissible amount of change for the cycles used in KinematicsOpt. The amount of change prevents you from accidentally changing the machine kinematics too much with KinematicsOpt. Whenever optimization requires a change greater than what is permitted in MP6600, a message appears and the change must be confirmed with NC start.

Sphere radius

In MP6601 you can enter the maximum permissible deviation of the calibration-sphere radius. The sphere radius is measured and checked with KinematicsOpt. This monitoring feature also detects incorrect probing caused by contamination.

> **Danger**
>
> After optimization the position of the presets relative to the workpiece may have changed.
>
> It is essential that you check or reset the datum points.

MP6600	**KinematicsOpt: Amount of change**
Input:	0.010 to 1.000 [mm] Default: 0.05
MP6601	**KinematicsOpt: Radius deviation of the calibration sphere**
Input:	0.010 to 0.100 [mm] Default: 0.02

PLC positioning

Some rotary axes cannot be moved by the NC, e.g. Hirth axes that can be positioned only by the PLC.

In MP 6602, you can now enter an M number that is used for positioning all rotary axes. This M function will be executed instead of the positioning movements through the NC. The rotary-axis angles required for this are saved in the Q parameters Q120 to 122. As a result, a positioning movement must always be initiated by an M macro. This is a very simple example of how this M macro may look like:

```
BEGIN PGM MM
FN 19: PLC =+4 / Q120 ;the PLC positions the A axis
FN 19: PLC =+6 / Q122 ;the PLC positions the C axis
END PGM MM
```

> **Note**
>
> Only local Q parameters (QL) are allowed in the M macro and in the NC programs called by the M macro. Otherwise, there is a great potential danger of collision during the further execution of the cycle.
>
> The cycles cannot verify that the correct position has been reached.

Be sure to remember while programming that the rotary axes must be pre-positioned to 0° degrees before the execution of the cycles if you use a PLC positioning movement. Since, in this case, there is also a great danger of collision if automatic presetting is used, this prerequisite must be confirmed by the operator.

This macro is called from all KinematicsOpt cycles when rotary axes are positioned. If an M function macro is specified, the target coordinates, not the rotary-axis positioning movement, are saved in the Q parameters 120 to 122 (axis angles) and the M function is called. The M function must ensure the correct positioning of the rotary axes to the nominal positions, as defined in Q120 to Q122.

Please keep in mind that you must not perform any transformations in the macro, and that you may only use QL parameters (no QS parameters from 0 to 99)!

MP6602	**KinematicsOpt: M-function macro for positioning the rotary axes**
Input:	0 to 999 -1: Function inactive

Unselecting a rotary axis

By expanding the number of measuring points to 0-12 with NC software 340 49x-05 (previously: 1-12), it is now possible to hide a "rotary axis" (0 = axis deselected). The angle of incidence of this axis is considered nevertheless.

Note

Please note that all transformations are effective incrementally. Compensation values for one axis lead to a corresponding shift in all subsequent positions of the axes.
By deselecting an axis, it is completely removed from the optimization with KinematicsOpt.

Positioning direction

The positioning direction of the rotary axis to be measured is determined from the start angle and the end angle. Because +270° is identical to –90° for example, the same angles can result in different measuring positions.
Examples:
Start angle = –90°
End angle = +90°
Measuring points = 4

Angle increment = (end angle – start angle) / (measuring points – 1)
= (90° - -90°) / 3
= +60°
Therefore: P1 = –90°; P2 = –30°; P3 = +30°; P4 = +90°

Start angle = +270°
End angle = +90°
Measuring points = 4

Angle increment = (end angle – start angle) / (measuring points – 1)
= (90° - 270°) / 3
= -60°
Therefore: P1 = –90°; P2 = –120°; P3 = –210°; P4 = –270°

Backlash and positioning errors

You can activate backlash measurement for all rotary axes in the input parameter Q432 during programming. You must enter an angular value that is greater than the actual backlash of the rotary axes.

Since the accuracy of this information depends on the measuring circle radius, the text log indicates the measurement uncertainty in degrees per 1 µm of system uncertainty:
The system uncertainty includes at least the position error of the linear axes and the scatter of the touch probe. However, this data is not available to the measuring cycle, and must therefore be estimated by the operator (see example below).
Additionally, the positioning error of the rotary axis is always determined. The calculated positioning error is also subject to this measurement uncertainty. In order to realistically measure the positioning accuracy, it is recommended that the angle increment when measuring the rotary axis is defined to be clearly below 90° (recommendation: less than 45°). If the angle increment is too large, then the positioning error is not calculated.
As a rule, the measurement uncertainty increases as the measuring circle radius increases.

Example for calculating the measurement uncertainty (the greater the calculated value of the measurement uncertainty, the less informative the measurement is):

- Position uncertainty of each linear axis: 10 µm
- Uncertainty of touch probe: 2 µm
- Measurement uncertainty: 0.0002 degrees/µm
- System uncertainty = SQRT (102+102+102+102) = 17.44 µm
- Measurement uncertainty = 0.0002 °/µm * 17.44 = 0.0034°

Log function

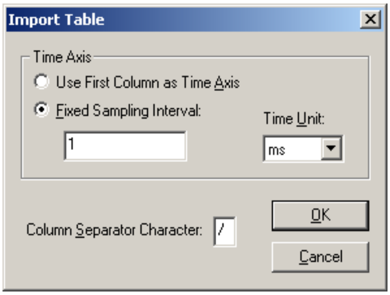
The Cycles 451 and 452 save log files in the PLC:\ directory, from which you can see the results of the individual measuring points. TNCscope can be used for a graphic evaluation (import the table). On the Extras -> Info tab, you can display additional information about the measurement and a short legend of the individual measured data.

The files are saved under PLC: as TCHPR45x_<axis name>.A. The columns created may vary depending on the conditions for measurement.

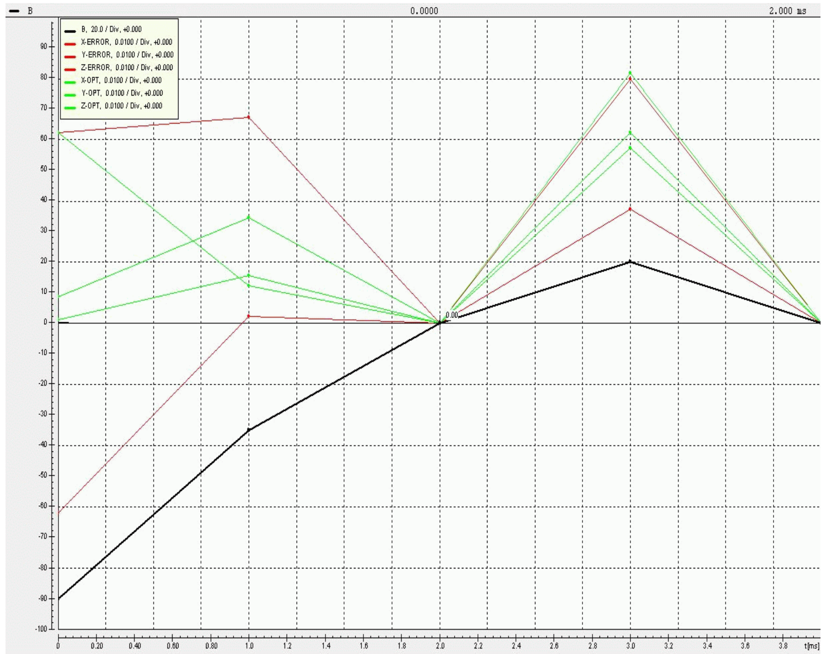
Data with the designation **REF** are to be understood with respect to the machine coordinated system. Data with the designation **ERROR** are measured errors in the coordinate system of the rotary axes. Data with the designation **OPT** are the errors to be expected in the coordinate system of the rotary axes after an optimization.
The data can be imported for evaluation (e.g. in TNCscope, Excel).

TNCscope: Open the file in TNCscope by selecting:

- File – **Import Table**. Then select the file to be opened.
- The **Import Table** dialog box opens.
- Select **Fixed sampling interval** and enter the value 1 ms in the Time unit field.
- Select "/" as the column separator and confirm your entry with OK.



Measured and optimized deviations (simulated measured data / displayed with TNCscope):



Optimized deviations as a function of the position of the rotary axis (simulated measured data / displayed with TNCscope):

- **Programming station**
  The cycle recognizes whether a programming station is being used. Because it is not possible to measure real data on a programming station, the measured data are simulated.

Touch probe cycle 450 (SAVE KINEMATICS)

Use this cycle to save the values of the active kinematics description in order to be able to restore the active kinematics description if required. For a detailed description of Cycle 450 SAVE KINEMATICS, refer to the Touch Probe Cycles User's Manual.

> Note
>
> - As a machine tool builder, you can save the shipping data for example. You enter the standard PLC code number (807667) and save the data to any memory location.
> - When an NC software version is installed, the backup file of the kinematics description that was created with Cycle 450 is overwritten. After installing the software, you must save the kinematics description again with Cycle 450 if required.

With NC software 340 49x-04 only those kinematics entries that can be changed by KinematicsOpt are saved and restored.

For 340 49x-05 this was enhanced, and now all axis and transformation entries that contain numerical values are saved. Along with the entries in the column **COORD,** now also numerical entries in the optional column **COMPTAB** are saved and restored. When restoring the COMPTAB files, the compensation limit from MP6600 is considered, independently of whether it is a translation or a rotation.

A link to a column in a compensation file can be entered in the column **COMPTAB**. These non-numeric entries are neither saved nor restored.
If, after backing up, such an entry is manually replaced by a numerical value, then this manually entered value is not changed when the old data are restored (the manual change is kept).

> Note
>
> Cycle 450 in software version 340 49x-05 is not compatible to the previous version. Data backed up with software 340 49x-04 cannot be restored!

The text log also contains the COORD entries that were backed up. Upon restoration, the text log shows the previous and the new, restored entries together. The entries in the **COMPTAB** column do not appear in the log.

Touch probe cycle 451 (MEASURE KINEMATICS)

The MEASURE KINEMATICS cycle enables you to check and, if required, optimize the kinematics of your machine. Use the 3-D TS touch probe to measure any calibration sphere that you have attached to the machine table.

Angle optimization

With NC software 340 49x-06, Cycle 451 was expanded by mode 2. In mode 1, only the position of the rotary axis is adjusted, whereas in mode 2 the position and the angular position are adjusted.

The general angular position of a rotary axis can be calculated if a minimum of two measuring points were probed per axis. Make sure that the measuring circle radius is not too small. Otherwise, the repeatability of the system can have a major influence on the result. The optimization of the angular position of a rotary axis only makes sense if the angles within the desired traverse path of the linear axes are relatively constant. If there are major changes in the angles at different positions of the linear axes due to component errors of the linear axes, the optimization of the angular position of the rotary axes is not reasonable. An adjustment of the angular position can contribute to improving the accuracy, especially on small and compact machines.

The ideal coordinate system is determined by the machine base (linear axes). The angles are calculated based on this system. The viewing direction depends on whether the rotary axis considered is a head axis or a table axis.

KinematicsOpt can perform a compensation to adjust the angular position of the rotary axes. This is only possible in conjunction with software option #52 KinematicsComp, and if the kinematics model has been prepared appropriately. This is done by changing the values of the rotation entries in the COMPTAB column. The values are only changed if the field already contains a numerical value (floating-point number). This prevents possible entries of a component-error compensation through KinematicsComp from being affected.

Angular-position compensations do not only influence the measured rotary axis, but also the complete transformation chain. After the affected rotary axis has been changed, KinematicsOpt therefore tries to undo the changes made to angles, which is desirable in some cases. This is possible within the scope determined by the correspondingly defined tolerances for the transformation chain.

Example of AC table:

NR	DOC	KEX	AXIS	COORD	COMPTAB	
0		ToolFile				
1		ToolAxis	Z			
2		MachAxis	Z			
3		MachAxis	X			
4		MachAxis	Y			
5		MachBase				Ideal coordinate system
6	YOA	Trans	Y	-300.025		
7	ZOA	Trans	Z	-449.881		
8	BOA	Trans	B		+0.00131	
9	COA	Trans	C		-0.01005	

NR	DOC	KEX	AXIS	COORD	COMPTAB
10		MachAxis	A		
11	-COA	Trans	C		+0.01005
12	-BOA	Trans	B		-0.00131
13	YOC	Trans	X	+350.152	
14	YOC	Trans	Y	-49.982	
15	Table surface	Trans	Z	+50	
16	AOC	Trans	A		-0.00115
17	BOC	Trans	B		-0.00281
18		MachAxis	C		

After the A axis has been measured, its angular position is determined and optimized. Adjustments are only possible through compensations in the **COMPTAB** column if a numerical value is entered in the corresponding lines. This is the case in lines 8 and 9 of this example, where the position errors B0A and C0A can be defined. After the A axis has been defined, Cycle 451 tries to cancel the rotations, which have just been performed. This is done by the compensations defined in lines 11 and 12 immediately afterwards. The two rotations are arranged inversely to the way they are arranged in lines 8 and 9. This simplifies traceability, but it is not absolutely necessary. After the angular position has been adjusted, a measurement is performed, in which the position of the rotary axis (Y0A and Z0A) is determined. The position is corrected in lines 6 and 7.

Then the C axis is measured and its angular position is optimized. The position errors A0C and B0C are corrected through the compensations in lines 16 and 17. Because no further elements that make an inverse transformation of the rotations possible are listed after the C axis, the workpiece coordinate system remains rotated from the ideal system (machine base).

Example of AC head:

NR	DOC	KEX	AXIS	COORD	COMPTAB
0		ToolFile			
1		ToolAxis	Z		
2	YOA	Trans	Y	+50.013	
3	ZOA	Trans	Z	+300.815	
4		MachAxis	A		
5	BOA	Trans	B		+0.0030
6	COA	Trans	C		-0.0009
7	XOC	Trans	X	+0.053	
8	YOC	Trans	Y	-0.092	
9	-BOC	Trans	B		+0.00337
10	-COC	Trans	C		+0.0103
11		MachAxis	C		
12	COA	Trans	C		-0.0103
13	BOA	Trans	B		-0.00337

NR	DOC	KEX	AXIS	COORD	COMPTAB	
14		MachAxis	Z			
15		MachAxis	X			
16		MachAxis	Y			
17		MachBase				Ideal coordinate system

During measurements in mode 2, the angular errors B0A and C0A in lines 5 and 6 of the COMPTAB column are compensated. Inverse transformation of the two angular errors B0A and C0A is not possible, because no corresponding rotations are defined. This means that the tool coordinate system is rotated from the ideal system. After the angular position has been adjusted, a measurement is performed, in which the position of the rotary axis (Y0A and Z0A) is determined. The position is corrected in lines 2 and 3.

Then the C axis is measured and its angular position is optimized. The position errors A0C and B0C can be corrected in lines 12 and 13 through the compensations in the COMPTAB column. An inverse transformation of both angular errors is possible in lines 9 and 10. As a result, further changes to other axes (also A) can be ruled out.

Example of B head (45°) and C table:

NR	DOC	KEX	AXIS	COORD	COMPTAB	
0		ToolFile				
1		ToolAxis	Z			
2	X0B	Trans	X	+50.013		
3	Z0B	Trans	Z	+300.815		
4		Trans	A	+45		
5		MachAxis	B			
6	A0B	Trans	A		-0.0027	
7	C0B	Trans	C		+0.0009	
8		Trans	A	-45		
9		MachAxis	Z			
10		MachAxis	X			
11		MachBase				Ideal coordinate system
12		MachAxis	Y			
13	X0C	Trans	X	+349.217		
14	Y0C	Trans	Y	-451.012		
15	Table surface	Trans	Z	-650.514		
16	A0C	Trans	A		-0.0005	
17	B0C	Trans	B		-0.0011	
18		MachAxis	C			

During measurements in mode 2, the angular errors A0B and C0B in lines 6 and 7 of the COMPTAB column are compensated. Inverse transformation of the two angular errors A0B and C0B is not possible, because no corresponding rotations are defined. This means that the tool coordinate system is rotated from the ideal system. After the angular position has been adjusted, a measurement is performed, in which the position of the rotary axis (X0B and Z0B) is determined. The position is corrected in lines 2 and 3.

Then the C axis is measured and its angular position is optimized. The position errors A0C and B0C can be corrected in lines 16 and 17 through the compensations in the COMPTAB column. An inverse transformation is not defined, and therefore the workpiece coordinate system is rotated from the ideal system.

For a detailed description of Cycle 451 MEASURE KINEMATICS, refer to the Touch Probe Cycles User's Manual.

Touch probe cycle 452 (PRESET COMPENSATION)

Touch probe cycle 452 can be used to optimize the kinematic transformation chain of the machine. Then the TNC corrects the workpiece coordinate system in the kinematics model in such a way that the current preset is in the center of the calibration sphere after optimization.

Cycle 452 corrects the transformations up to the last active rotary axis (like Mode 1 of Cycle 451).

After the last measured rotary axis, it is attempted to set the preset at the center of the calibration sphere via the kinematics transformations. The next possible transformations for this compensation are searched for after the last measured rotary axis of the transformation chain.

An error message is output if the preset cannot be completely compensated.

Cycle 452 can be used, for example, to compensate specific thermal behavior, and also to adapt the kinematics of tool changer heads.

Example 1 for the application of Cycle 452:
Adjust tool changer head (A axis)

SUBFILE: Head 1		SUBFILE: Head 2	
Trans	Y	Trans	Y
Trans	Z	Trans	Z
MachAxis	A	MachAxis	A
Trans	X	Trans	X
Trans	Y	Trans	Y
Trans	Z	Trans	Z
MAINFILE			
MachAxis	C		
MachAxis	Y		
MachAxis	X		
MachAxis	Z		

Once a head has been optimized (A and C axis, SUBFILE Head 1 and MAINFILE), it must be used to set the preset in the center of the calibration sphere (Cycle 451). Then Cycle 452 is simply used in combination with the second head (SUBFILE: Head 2) to optimize the A axis with preset compensation.

The transformations shown in green are recognized as compensation transformations, and the preset is thus placed at the sphere center.

Example 2 for the application of Cycle 452:
Compensation of thermal behavior

Trans	X
Trans	Y
Trans	Z
MachAxis	Y
MachAxis	X
MachAxis	Z

A present compensation is performed in this 3-axis machine. In the Y and Z directions the preset is returned to the position in the machine coordinate system at which it was located for setting the reference point of the calibration sphere.

Up to NC software 340 49x-06, the compensation translations had to be aligned according to the machine coordinate system. As of 340 49x-06, this constraint no longer applies. The three translations only need to make a shift in any desired spatial direction possible.

Example of angle head:

NR	DOC	KEX	AXIS	COORD	COMPTAB	
0		ToolFile				
1		ToolAxis	Z			
2		Trans	X	+50.013		Translations for preset compensation
3		Trans	Y	+0.024		
4		Trans	Z	+300.815		
5		Trans	B	-90		Fixed rotation
6		MachAxis	Z			
7		MachAxis	Y			
8		MachAxis	X			
9		MachBase				Ideal coordinate system

Example of non-perpendicular preset compensation

NR	DOC	KEX	AXIS	COORD	COMPTAB	
0		ToolFile				
1		ToolAxis	Z			
2		Trans	X	+0.032		
3		Trans	Z	+299.941		
4		MachAxis	B			
5		Trans	Z	+200.018		Translations for preset compensation do not result in a Cartesian (perpendicular) coordinate system
6		Trans	B	+33		
7		Trans	Z	+50		
8		Trans	Y	0		
9		Trans	B	-33		
10		Trans	X	-0.026		
11		Trans	Y	-0.012		
12		MachAxis	C			
13		MachAxis	Z			
14		MachAxis	Y			
15		MachAxis	X			
16		MachBase				Ideal coordinate system

6.5.7 "Tilt Working Plane" Feature

The user defines the position of the working plane in Cycle 19 "Tilted Working Plane." Then the iTNC performs a coordinate transformation.

With the 3D ROT soft key you can activate the tilted working plane separately for the MANUAL and PROGRAM RUN operating modes.

If bit #2 is set in MP7507, the 3D ROT soft key will be available in the Test Run mode after you have pressed the MOD key. Here you can enter (in the same manner as in the Manual Operation mode) the tilt angles in the axes A, B and C, and you can activate/deactivate tilting for the test run. Additionally, a soft key is available to load the status of the machine (active kinematics, tilt angles + active in program run). If bit#2 is set in MP7507, tilting is not deactivated anymore when a new program is selected in Test Run mode. The axis that is set for manual probing will be used as the tool axis until the first tool call. This can cause the error message "Angle cannot be calculated" if the tool axis and the selected kinematics model cannot be used together.

With MP7500 you can define the function of the tilted working plane cycle. With **FN18: SYSREAD ID290 NR2** you can request the values of the individual bits from MP7500.

Assignment of input values (Cycle 19)

With MP7500 bit 1 you define whether the input applies to the position of the tilted axes (bit 1 = 0) or the position of the working plane (bit 1 = 1).

If the input value applies to the position of the working plane, the iTNC calculates the position of the tilting axes and saves the coordinates in Q parameters:

- Q120: Coordinate of the A axis
- Q121: Coordinate of the B axis
- Q122: Coordinate of the C axis

With **FN17:SYSWRITE ID990 NR5 IDX5** you can determine if a principal axis is shown on top of another principal axis in an untilted coordinate system due to a tilt motion.

Automatic positioning

After the coordinate transformation, the Z axis remains parallel to the tool axis, perpendicular to the X/Y plane. With MP7500 bit 2 you define whether the "tilted working plane" function automatically positions the tilting axes (bit 2 = 1). In this case the user can enter the feed rate and setup clearance in the cycle.

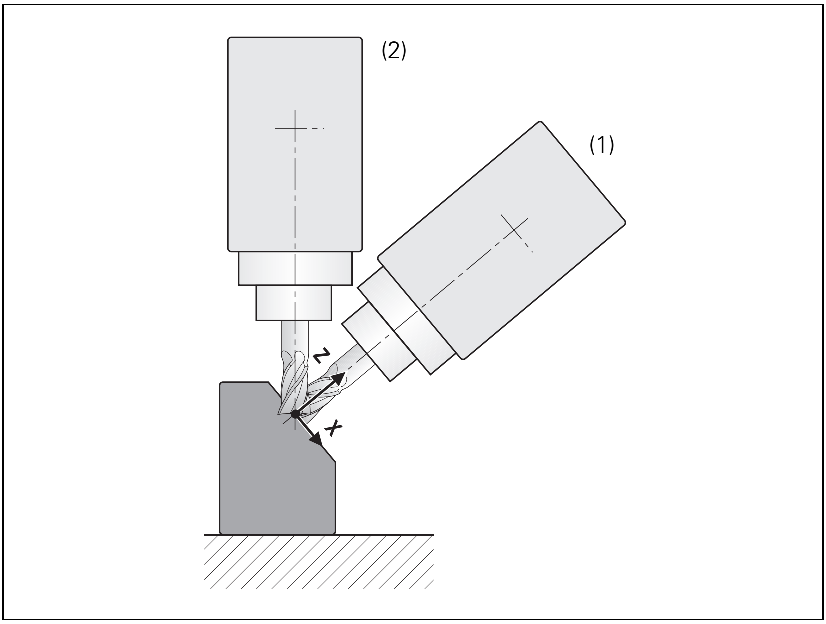
The iTNC then moves automatically to the setup clearance and interpolates the swivel and principle axes so that the tool point remains in the same position in the tilted coordinate system.

With MP7682 bit 9 you control whether the status of the **Tilt working plane** function in an NC program is applied to the **Manual** operating mode during a program interruption.

Datums

Closed-loop axes:

During "datum setting" for X, Y and Z, the datum is recalculated back to the home position of the tilting element when "tilted working plane" is **active** (1). So when "tilted working plane" is **inactive** and the tilting element is in its home position, the tool is positioned at the datum set while "tilted working plane" was **active** (2).



Behavior during datum setting can also be influenced via MP7500 bit 5:

MP7500 bit 5 = 0 (is entered in the kinematics description table)
During datum setting in X, Y and Z with an **active** tilted working plane, the current rotary-axis coordinates are checked to see if they are correct for the tilt angles, and with an **inactive** tilted working plane the rotary axes are checked to see if they are at 0. For datum setting with an **active** tilted working plane, the corresponding angles must be entered under 3-D ROT.

MP7500 bit 5 = 1 (is entered in the kinematics description table)
It can happen with titling elements with Hirth couplings that by locking the Hirth coupling, the actual value of the encoder will no longer exactly agree with the mechanical position of the tilting element. If this happens, the nominal values should be used to calculate the various datums (MP7682 bit 1).
If problems continue to occur, MP7500 bit 5 should be set to 1. The checking described in MP7500 bit 5 = 0 does not take place. The tilt angles entered under 3-D ROT are used to calculate the datums in X, Y and Z.

With MP7682 bit 1 you define whether the nominal or the actual values are used to calculate the presets during "datum setting" (is valid for MP7500 bit 5). Also see “Special case: Reference-point setting with the PLANE function, Hirth axes and M114" on page 774.

Open-loop axes:

The user must enter the current positions of the tilting axes by using the 3-D ROT soft key.

> Note
>
> In the combination of coordinate transformation cycles, note the sequence of activation and deactivation.

Special case: Reference-point setting with the PLANE function, Hirth axes and M114

If rotary axes with Hirth coupling are positioned via PLC or NC, then as mentioned above, angles can only be entered according to a certain grid. Machining can be performed in any plane by tilting the plane with the PLANE function and programming the rotary axis grid coordinates with M114 (automatic compensation of the machine geometry when working with tilting axes).
It is possible to save and set reference points in this state.
The iTNC then uses the nominal positions of the rotary axes programmed in M114 to calculate the reference point. With MP7493 the machine manufacturer can enter a maximum deviation of the rotary axes based on the nominal position resulting from the orientation of the working plane. The control accepts this deviation when setting reference points and when measuring with M114. The default value for this machine parameter is 0.005.
The iTNC uses the current ACTUAL/NOMINAL positions (MP7682, bit 1) of the rotary axes to calculate the reference point.

> Note
>
> Please note that in this procedure the tool might not be perpendicular to the tilted working plane.

MP7493	**Maximum deviation of the current tool orientation relative to the tool axis when setting a reference point with the PLANE function with M114**
Input:	0.0000 to 30.0000 [degrees] Default: 0.005

Spatial angle C ≠ 0

On machines with C tables and tool axis Z, the spatial angle C ≠ 0 (with A = 0 and B = 0) can be realized through a rotation of the coordinate system or a rotation of the table:

- With MP7500 bit 6 = 0, the spatial angle C is realized through a rotation of the coordinate system.
  With MP7500 bit 6 = 1, the spatial angle C is realized through a rotation of the table. At the same time, the angle is saved in Q122. This makes it possible, for example, to machine a workpiece by always using the same axis for paraxial linear blocks in the X/Y plane.

Interrogating the 3-D ROT data via PLC

PLC Module 9045 makes the relevant data for the "Tilt working plane" function available to the integrated PLC as well. The following data are available:

- Tilt angles (A, B, C)
- Tilted axes (A, B, C)
- In which operating mode the "Tilt working plane" function is active

When the 3-D ROT data are interrogated via Module 9045, the data are entered in four sequential double words beginning from the given starting address [n]. The returned bit mask indicates which tilt angles are possible as a result of the currently active kinematics configuration.

Note

Ensure that the addresses are available on the control, and that the given target address is a double word address.

The data are output in the following format:

- D[n+0]: Tilt angle A (unit 0.0001°)
- D[n+4]: Tilt angle B (unit 0.0001°)
- D[n+8]: Tilt angle C (unit 0.0001°)
- W[n+12]: Currently tilted axes (bit-encoded)
  Bit 0: Axis A tilted
  Bit 1: Axis B tilted
  Bit 2: Axis C tilted
- W[n+14]: "Tilt working plane" is active in operating mode
  Bit 0: Tilting active in **Program Run** operating mode
  Bit 1: Tilting active in **Manual** operating mode

Module 9045 Reading the 3-D ROT data

Call:
PS B/W/D/K <>Starting address as double-word number [n]>
CM 9045

Error recognition:

Marker	Value	Meaning
M4203	0	3-D ROT data read
	1	3-D ROT data was not read

Conditions and constraints

Conditions:

- The display position in the status window is referenced to the tilted coordinate system.
- In the combination of coordinate transformation cycles the sequence of activation must agree with the sequence of deactivation.
- The tool radius compensation in the working plane and the tool length compensation parallel to the tool axis is active.
- For machining with tilting tables, the coordinate system remains parallel to the machine coordinate system.
- If MP7500 bit 1=1 (spatial angle), Module 9045 shows all axes that will be tilted.
- If MP7500 bit 1=0 (axis angle), Module 9045 shows all axes that are actually tilted.

Constraints:

- PLC positioning movements are always parallel to an axis of the machine coordinate system (Cycle 19 has no influence).
- A datum shift via PLC also works with the "tilted working plane" function.
- The axis designations for the tilting axes are limited to A, B and C. Each designation can be used only once.
- With an active Cycle 19 "Tilted working plane," it is not possible to position with M91 or M92.

If the position of the working plane is entered, only the following swivel axes (with tool axis Z) are permissible:

- Double swivel head 45°: Axis sequence A fixed; B or C variable; A fixed; B or C variable
- Rectangular double swivel head: Axis sequence A or B variable; C variable
- Rotary or tilting table: Axis sequence C variable; A or B variable
- Swivel head and rotary table: Axis sequence A or B variable; C variable
- Tilting table 45°: Axis sequence C variable; A fixed; B variable; A fixed
- Rectangular double swivel head: Axis sequence A variable; B variable
- Universal swivel head: Axis sequence A fixed; B –90°; A variable; B +90°; A fixed; C variable
- Swivel head and rotary table: axis sequence B variable, A variable
- Swivel head and rotary table: Axis sequence C fixed, A fixed, B fixed –90°, A variable, B fixed +90°, A fixed, C fixed, C variable

With tool axis Y:

- Rotary or tilting table: Axis sequence B variable; A variable
- Double swivel head 45° and rotary table: Axis sequence A fixed; C variable; A fixed; B variable
- Rotary or tilting table: Axis sequence A or C variable; A or C variable
- Swivel head 45° and rotary table: Axis sequence A +45°, B variable, A –45°, C variable

With tool axis X:

- Universal swivel head: Axis sequence B fixed; A variable; B fixed; C variable

MP7500 **"Tilt working plane" (inactive preset table)**
Format: %xxxxxxxx
Input: Bit 0 – Switch-on "tilted working plane" function
0: Off
1: On
Bit 1 –
0: Angles correspond to the position of the tilting axes of the head/table
1: Angles correspond to the spatial angle (the iTNC calculates the position of the tilted axes of the head/table)
Bit 2 –
0: The tilting axes are not positioned with Cycle 19
1: The tilting axes are positioned with Cycle 19
Bit 3 –
0: The current tilting-axis position is taken into account with respect to the machine datum
1: The 0° position is assumed for the first rotary axis
Bit 4 –
0: Compensate mechanical offset during exchange of the spindle head when calling M128, M114, TCPM or "tilted working plane"
1: Compensate mechanical offset during PLC datum shift
Bit 5 –
0: The current tilting-axis position is taken into account with respect to the machine datum
1: The tilting-axis position that was entered with the 3-D ROT soft key applies
Bit 6 –
0: Spatial angle C is realized through a rotation of the coordinate system
1: Spatial angle C is realized through a rotation of the table
Bit 7 –
0: The current tilting-axis position is taken into account with respect to the machine datum
1: The active tilting-axis position is
a) derived from the tilting angles in the 3D ROT window if manual tilting is active
b) derived from the reference coordinates of the rotary axes if tilting is inactive
Bit 8 –
0: The tilting axis positioning is considered depending on bit 3, bit 5 and bit 7
1: If manual tilting is active, the datum to be set for the principal axes X, Y and Z is recalculated back to the home position of the tilting element

MP7500 **"Tilt working plane" (active preset table)**
As of software version:340 422-01, 340 480-02
Format: %xxxxxxxx
Input: Bit 0 – Switch-on "tilted working plane" function
0: Off
1: On
Bit 1 –
0: Angles correspond to the position of the tilting axes of the head/table
1: Angles correspond to the spatial angle (the iTNC calculates the position of the tilted axes of the head/table)
Bit 2 –
0: The tilting axes are not positioned with Cycle 19
1: The tilting axes are positioned with Cycle 19
Bit 3 – No function
Bit 4 – No function
Bit 5 – Test of the tilting axis during "datum setting" in X, Y and Z
0: Current tilting-axis position must fit to the defined tilting angles
1: No test
Bit 6 –
0: Spatial angle C is realized through a rotation of the coordinate system
1: Spatial angle C is realized through a rotation of the table
Bit 7 – No function
Bit 8 – No function

MP7510 **Transformed axis**
Format: %xxxxxx
Input: 0: End of the transformation sequence
Bit 0 corresponds to axis X
Bit 1 corresponds to axis Y
Bit 2 corresponds to axis Z
Bit 3 corresponds to axis A
Bit 4 corresponds to axis B
Bit 5 corresponds to axis C
MP7510.0–14 Transformation 1 to transformation 15

MP7520 **Additional code for transformation**
Format: %xx
Input: Bit 0 – Tilting axis
0: Swivel head
1: Tilting table
Bit 1 – Type of dimension in MP7530.x
0: Incremental dimension for swivel head
1: Absolute with respect to the machine datum for tilting table
MP7520.0–14 Transformation 1 to transformation 15

MP7530 **Type of dimension for transformation**
Input: Entry of a formula is possible, see page 525
0: Free tilting axis
MP7530.0–14 Transformation 1 to transformation 15

Note

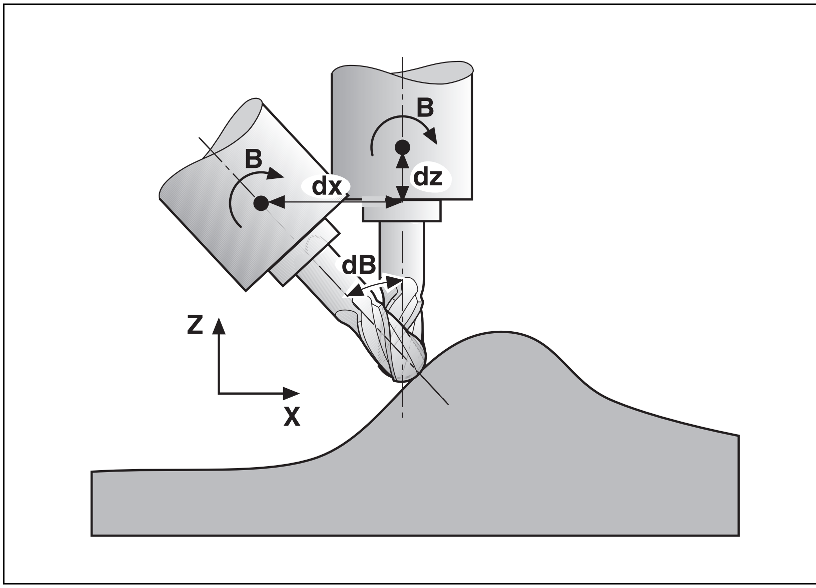
MP7530 cannot be overwritten with Module 9031 (overwrite machine parameters), since the MP contains a string, but the module transfers an integer value.

MP7550 **Home position of the tilting element**
Input: –99 999.9999 to +99 999.9999
MP7550.0 A axis
MP7550.1 B axis
MP7550.2 C axis

MP7682 **Machine parameter with multiple function**
Input: Bit 1 – Reference value for calculating the preset during "datum setting"
0: Actual value is calculated
1: Nominal value is calculated
Bit 9 – Apply tilting of the working plane
0: The status of the **Tilt working plane** function is not applied to the **Manual** operating mode during a program interruption (behavior until now).
1: The status of the **Tilt working plane** function is applied to the **Manual** operating mode during a program interruption.

6.5.8 Automatic Compensation of Offset for Tilting Axes

Unlike the "tilted working plane," here the coordinate system is not tilted. With M114, M128 or TCPM, the iTNC compensates the offset of the tool that results from tilting the axes. The tool tip is always located on the programmed nominal coordinates.



The iTNC can perform a 3-D length compensation; the radius compensation must be performed by the CAD system or the postprocessor. If the iTNC compensates the tool length, then the programmed feed rate refers to the tool point. Otherwise it refers to the tool datum.

Miscellaneous function M114

Automatic compensation with M114:

- Linear and rotational movements are superimposed. The resulting contour deviations depend on the length of the linear interpolation.
- When the table is rotated, the coordinate system is rotated against the machine coordinate system. **The iTNC does not take this into account.**
- M114 can be used with non-controlled tilting axes or PLC tilting axes. In this case, the current tilting angle and the tilting axis are entered in the NC block behind M114.
- M114 is effective locally in cycles, i.e. the function is canceled before the return to the main program. If you want M114 to also be effective in the main program, you must use **`FN17: SYSWRITE ID420 NR0 IDX0 = 0`** (globally effective coordinate transformation).

Miscellaneous function M128, TCPM

Automatic compensation with M128 or TCPM:

- Linear and rotational movements are superimposed. The resulting contour deviations are compensated.
- When the table is rotated, the coordinate system is rotated against the machine coordinate system. **The iTNC takes this into account.**
- M128 and TCPM remain in effect even after a change in operating modes. This means that the axis can be moved with the compensated machine geometry in Manual mode with the axis direction keys, or in the **El. Handwheel** mode.
- With the miscellaneous function M118, the handwheel positioning movements can be superimposed on the program run movements. The iTNC automatically performs the compensating movements in the principal axes.

A transitional element is inserted at non-tangential contour transitions when positioning with rotary axes. However, W1026 (axes in position) is not set and axes will not be clamped. This problem can be solved with M134 (exact stop at non-tangential contour transitions when positioning with rotary axes):

- Program M134 in the NC program or set MP7440 bit 6 = 1.

When M128 or TCPM are used, the principal axes make compensating movements:

- In MP7471, define the maximum velocity of the principal axes during compensating movements. The speed for compensating movements with M128 can also be limited via the PLC and the **FMAX** soft key, in addition to MP7471.

MP7440	**Output of M functions**
Format:	%xxxxxxx
Input:	Bit 6 – Automatic activation of M134 0: M134 must be activated in the NC program 1: M134 is automatically activated when an NC program is selected.
MP7471	**Maximum velocity of the principal axes during compensating movements through M128 or TCPM**
Input:	0 to 300 000 [mm/min]

Virtual tool axis (VT)

Starting with software 340 49x-04, handwheel superimposed traverse in the active tool axis system (Virtual Tool Axis VT) is now possible when TCPM is active. This function is available in the Global Program Settings (Option#44). When the HR420 is connected, the position of the virtual tool axis is displayed separately. It can be used in automatic mode when the HR 420/HR 410 is connected.

Requirements that must be met for using the Virtual Tool Axis VT:

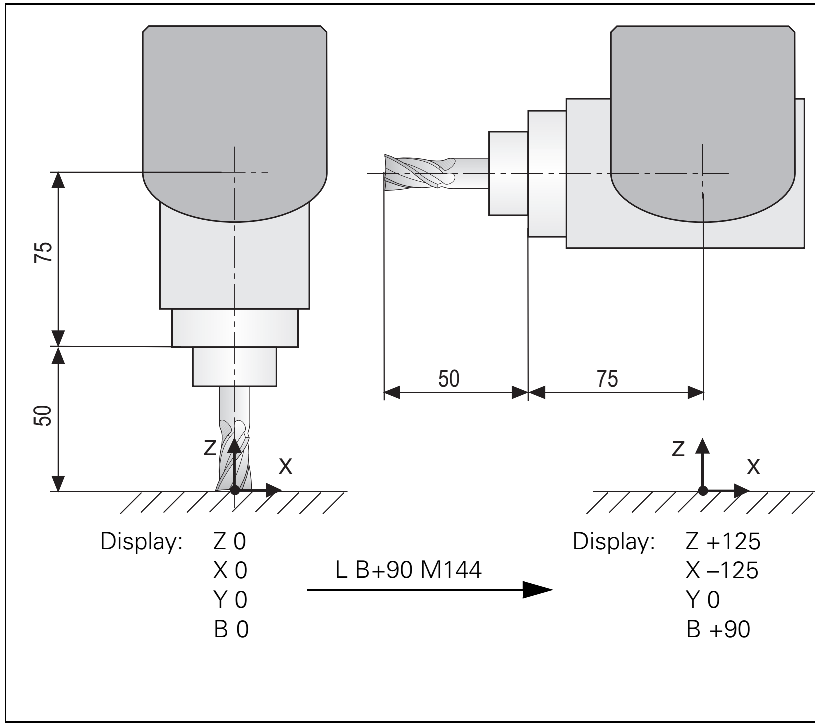
- TCPM (Tool Center Point Management) function must be active.
- DCM (Dynamic Collision Monitoring) must be deactivated.
- MP7641 bit#4 "Handwheel superimposition in the active tool-axis direction" must be set.
- MP7503 "Virtual tool axis" must be set.
- MP7682 bit#9 "Load tilted working plane" should be set for the tilting values to be loaded into the **Manual Operation** mode during a program interruption.
- T must be entered in MP410.x for the virtual tool axis (VT).
- Global Program Setting software option (Option #44) must be activated.
- In the Global Program Settings you must define the permissible traverse range for the VT axis (as with M118).
- Marker M4576 "Disable handwheel pulses" must not be set.

Associated new functions:

- In the Global Program Settings for handwheel superimposition, the axis VT can also be activated for the active tool-axis direction.
- If an HR 420 is used, the axis VT can be selected for it in the axis menu.
- When an HR 410 handwheel is connected, the axis VT can be selected via PLC module 9036 (selection parameter 6 (handwheel axis); axis value -1).
- If selection parameter 10 (handwheel axis) is selected, PLC Module 9035 supplies the value $8000000 (bit 31 is set) if the VT axis is active.
- If the handwheel traverses an offset in this axis, the tool length compensation will change (additional DL).
  This additional length compensation is retained after a tool change.

Miscellaneous function M144/ M145

M144 considers the kinematics in the display when a swivel axis is moved. However, the other axes do not perform any compensating movements. M144 is deactivated with M145.



FN18: SYSREAD ID310 NR144 can determine if M144 is active or inactive.

MP7502 **Functionality of M144/M145**

Input: %xxx
Bit 0
0: M144/M145 not active
1: M144/M145 active
Bit 1 – M144/M145 in the automatic modes
0: M144/M145 active
1: M144 is activated automatically at the start of an NC program. It can only be deactivated with M145 during an NC program.
Bit2 – M144/M145 in the manual modes
0: M144/M145 not active
1: M144/M145 active

6.5.9 Virtual Tool Axis

General information

When moving the axes in the manual operating modes, and when re-approaching the contour after a program interruption, you can select via the **3D ROT** and **TOOL AXIS** soft keys the coordinate system of the tool. The axes can be moved in the current tool axis direction even before traversing the reference marks. To accomplish this, the last position of the rotary axes before power-off is stored in non-volatile memory for incremental encoders. These non-volatile axis values are shown in a pop-up window when the control is started, and must be acknowledged by the machine operator with **ENT** (confirmation of the values) or **NO ENT** (values deviate, reference run must be performed first).

Starting with software 340 49x-04, handwheel superimposed traverse in the active tool axis system (Virtual Tool Axis VT) is now also possible when TCPM is active, see page 782.

Activation

MP7503	**Virtual tool axis – Reapproaching the contour and manual traverse in the current tool-axis direction**
Input:	0: Inactive 1: Active

6.5.10 Tilting functions with open-loop rotary axes

Inclined milling with open-loop rotary axes (counter axes)

Machines with rotary axes that only have open-loop axes (counter axes, MP120.x = 0, are not controlled) can be used for inclined milling. The following procedure is necessary:

- With M128 inactive (TCPM: the position of the tool tip remains constant when positioning tilting axes), the machine operator sets the rotary axes to the required nominal values.
- When M128 is activated, the control assumes the actual values of the non-controlled rotary axes, and uses them to calculate the changed position on the tool center point.
- The display of axes X/Y/Z is updated with the newly calculated values, and the compensating movement is performed with the next positioning.
- As long as M128 is active, the positions of the non-controlled rotary axes are monitored. If the positions of these axes deviate by more than the values defined in MP1110.x (standstill monitoring), then an error message is output and the momentary machining is interrupted.
- This function is even permitted after a program interruption with the **MANUAL TRAVERSE** soft key. In this case the new compensating movement is determined after switching with the **APPROACH POSITION** soft key.

6.5.11 Cylindrical Surface

Cycles 27 and 28, "Cylinder Surface," enable the user to machine a contour on a cylindrical surface (see the User's Manual).

Prerequisites of the previous kinematics description:

- In MP7510 to MP7530, the center of rotation of a rotary axis must be defined (see example 3).
- If PLC datum compensation is used, the same home position must apply in the description of the machine geometry in MP7510.x to MP7530.x as in the datum shift.
- After a change in MP7510.x or MP7530.x, the datum must be reset.

Prerequisites of the new kinematics tables:

- A complete description of the kinematics, from the tool reference point to the center of the rotary axis, must exist.
- If the axis geometry is changed (**MachAxis** ...), the datum must be reset.

6.6 Synchronized Axes

6.6.1 Gantry Axes

In gantry axes, tandem tables, etc., two servo-controlled axes are coupled so that they can move only simultaneously. The main axis is referred to as the master, and the tracking axis as the slave. The number of control loops available also determines the number of Gantry axes possible. Two control loops are always needed for each Gantry axis. Gantry axes can be distributed over different speed controller PCBs.

The function is effective during control both with following error and with velocity feedforward.

Activating synchronized axes:

- Assign a slave axis to a master axis.

MP850.x	**Synchronized axes**
Input:	0: Master axis 1: Slave axis to axis 1 2: Slave axis to axis 2 3: Slave axis to axis 3 4: Slave axis to axis 4 5: Slave axis to axis 5 6: Slave axis to axis 6 7: Slave axis to axis 7 8: Slave axis to axis 8 9: Slave axis to axis 9

Example

Axis 4 is slave to axis 1:

- MP850.0 = 0
- MP850.1 = 0
- MP850.2 = 0
- MP850.3 = 1
- MP850.4 = 0
- MP850.5 = 0
- MP850.6 = 0
- MP850.7 = 0
- MP850.8 = 0

Master-slave position deviation

The iTNC monitors the synchronism of the coupled axes. If the master and slave axes deviate from each other by the difference of the following errors, the iTNC displays the slave axis with the message **EXCESSIVE SERVO LAG IN <AXIS>**. The LAG display shows the current difference in position.

- In MP855.x of the slave axis, enter the maximum permissible difference in positions between the master and slave.

If an offset is caused in the axes through an emergency stop, they will be synchronized after the emergency stop.

Datum at position after switch-on (MP860.x = 0)

Entry for the slave axis

With MP860.x you can select whether the position after switch-on should be used as a synchronization reference. Master and slave axes must be at identical positions. If the defined datums are to be reproduced, then only the master needs to be moved over the reference mark.

Monitoring of synchronized axes begins immediately upon switch-on.

Datum at reference marks (set MP860.x Bit #1)

Entry for the slave axis

With MP860.x you can select whether the position should be ascertained by traversing the reference marks. After crossing over the reference mark, the master and slave axes are positioned to the same value. The default setting can be corrected with MP960.x (machine datum). In order for MP960.x to be set, the axes must traverse the reference marks with MP860.x = 0, so that no compensation movements are made. An offset in the axes is corrected after both reference marks are traversed. Reference mark traverse is ended as soon as a reference mark is traversed in both axes. The monitoring function is not active until after the compensation movement. The monitoring function is not active before the reference marks are traversed.

Conditions:

- The same type of reference mark traverse must be set for both the master and slave axes (MP1350.x).
- The velocity with which an offset (after traversing a reference mark or emergency stop) is compensated for is defined in MP1330.x for the slave axis.
- In the sequence for traversing the reference marks (MP1340.x), the master axis must be defined before the slave axis.
- The compensation movement can **not** be stopped with an NC stop (only with an emergency stop).
- The compensation movement is **not** considered in the following words:
  - W1026 (Axes in position)
  - W1028 (Axes in motion)
- If the master axis has traversed the reference mark at the time of an NC stop or an emergency stop, but the slave axis has not yet crossed it, then the slave axis can only be moved across it by using the axis-direction keys.
- Using a linear encoder: it is sufficient if the master axis has one reference end position.
- Using the speed encoder for linear measurement: One reference end position is enough, but the NC needs a reference end position signal for both axes (W1054).

Conventions

For synchronized axes:

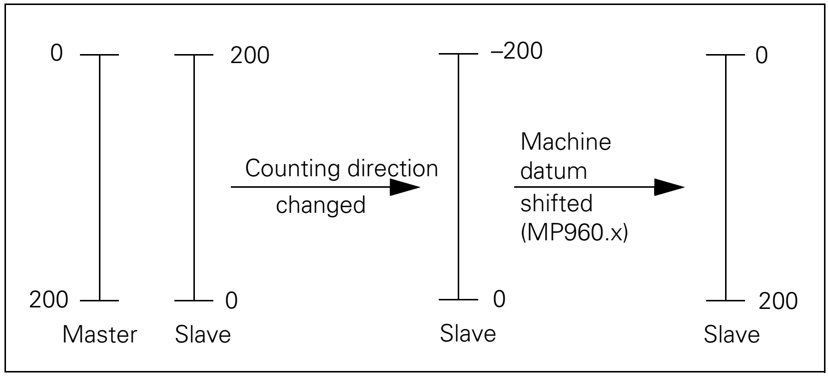
- The slave axis cannot be moved separately.
- The nominal value display of the slave axis shows the nominal value of the master axis.
- The PLC program must ensure that the master axis does not move until the slave axis is ready (clamping, feed-rate enable).
- For the slave axis, the bits for traverse direction in W1030 and axis in motion in W1028 are **not** set.
- An axis cannot be both master and slave at the same time.
- Linear and nonlinear axis error compensation as well as temperature compensation must be entered separately for each axis.
- The values for rapid traverse, acceleration, jerk, software limit switches, feed rate for reference mark traverse, and manual feed rate are also taken over from the input values of the master axis for the slave axis.
- When operating with following error, the k_V factors for master and slave must be the same.
- The axes must be either both analog or both digital.
- Master and slave axes can be linear or rotary axes.
- For gantry axes, one position encoder is sufficient.
- The nonlinear axis-error compensation can be used separately for master and slave axes.
- For the nonlinear axis-error compensation, master and slave axes may be dependent on each other.

MP855.x **Synchronization monitoring**
Input: 0 to 100.0000 [mm]
0: Monitoring not active

MP860.x **Datum for synchronous control**
Input: %xxx
Bit #0 – Datum for synchronous control
0: Datum at position after switch-on
1: Datum at reference marks (machine datum)
Bit #1 – Master-slave torque control
0: Not a torque slave axis
1: Axis is torque slave axis
Bit #2 – Brake test of slave axis
0: No automatic brake test of the slave axis
1: Brake test of the slave axis is automatically run with master axis

Example

- Gantry axes with two position encoders
- Position encoder of the slave axis is mounted mirror-inverted.



6.6.2 Master-Slave Torque Control

In master-slave torque control, two motors (master and slave) are mechanically coupled. Because of the coupling, only one position encoder is required. The motor to which the position encoder is assigned is the master.

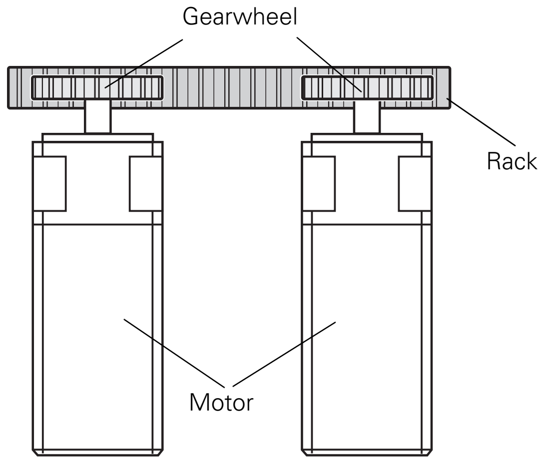
From a maximum of twelve controlled axes, six times two axes can be controlled in the torque-master-slave-control, whereby you must keep in mind that the master and slave axis are on the same speed controller PCB.
First speed controller PCB: X15 to X20
Second speed controller PCB: X80 to X85

For the CC 424 the PWM outputs of the master and slave axes must always be operated on the same DSP ("Single-speed" setting)

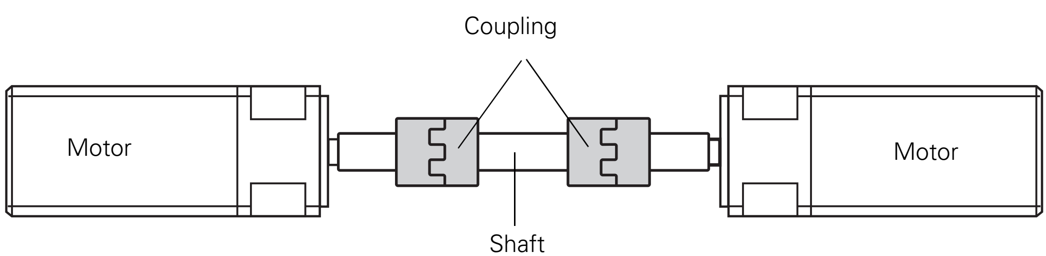
Along with X51/X53 and X52/X54, PWM outputs X55/X57 and X56/X58 can also be used on a CC424(B) for 8 and 14 control loops to operate master-slave torque axes.

In principle there are two applications:

- Minimization of mechanical play through mutual tensioning
- Distribution of torque with a rigid coupling

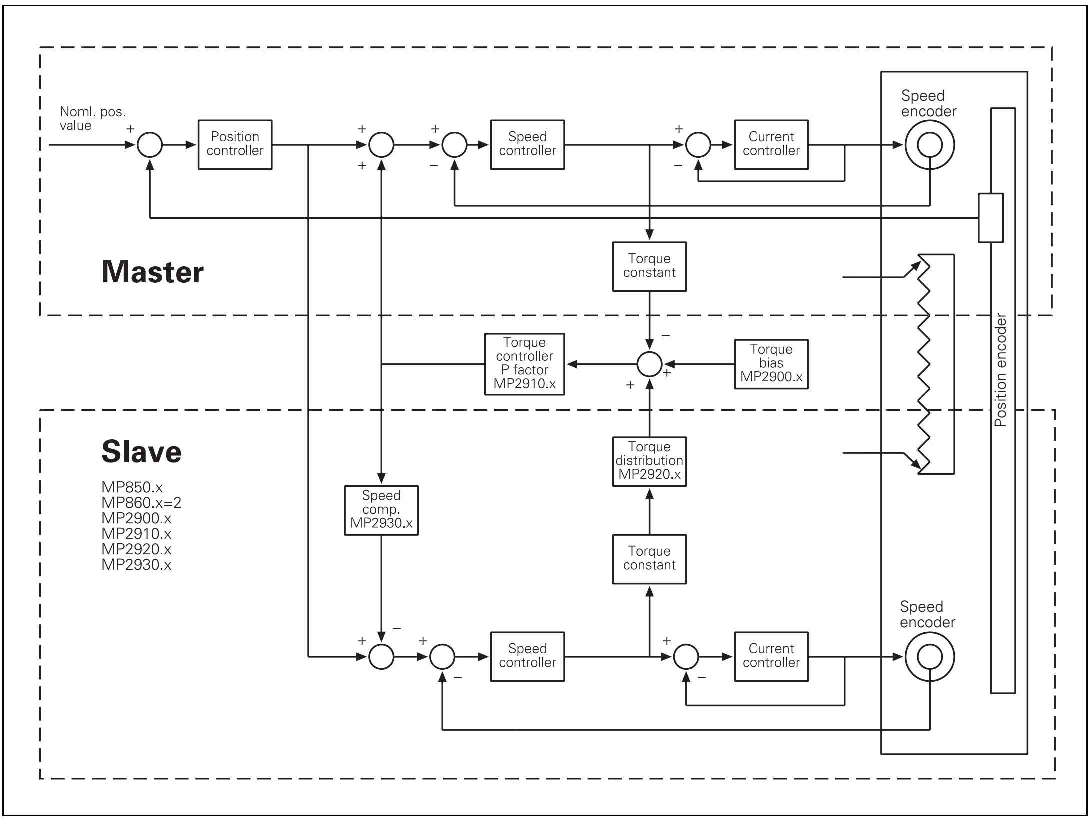


Minimization of mechanical play through mutual tensioning



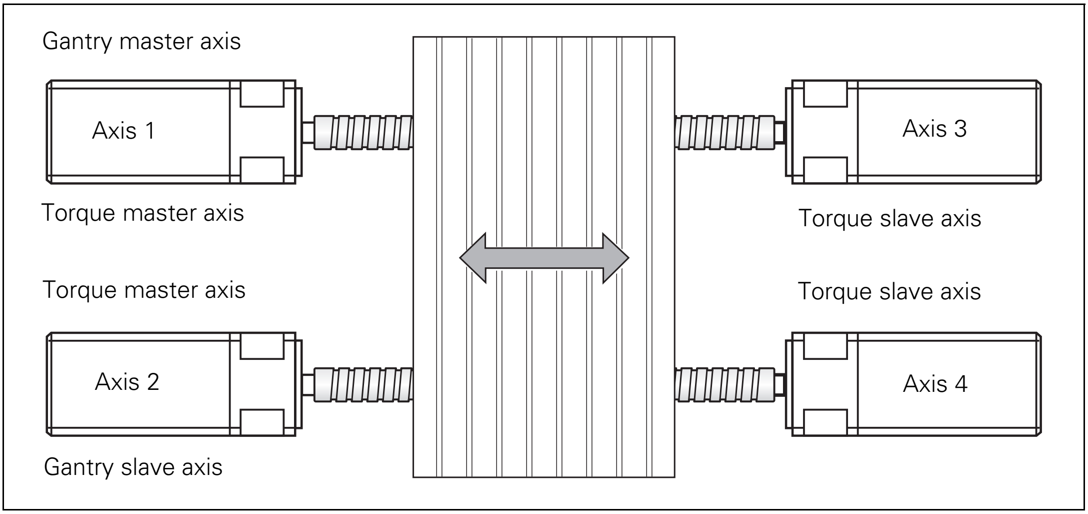
Distribution of torque with a rigid coupling

Method of function Position control is deactivated in the slave axis. The nominal velocity of the master axis is at the same time the nominal velocity of the slave axis. The speed controllers of both axes remain independent. The manipulated variables coming from the speed controllers, i.e. the nominal torque current values, are weighted with the torque constants of the motors and compared with each other. In addition, a tensioning torque (MP2900.x) can be introduced at this comparison point. To permit a distribution of drive torque, the nominal torque of the slave axis can be multiplied with a weighting factor (MP2920.x). The result at the comparison point is fed to a torque balancing controller that amplifies it proportionally (MP2910.x). The manipulated variable of the balancing controller is a speed compensation value that is added to the current speed value.



Gantry axes in master-slave torque control

It is possible to run gantry axes in master-slave torque control. The gantry master and gantry slave axes are at the same time torque master axes and have one torque slave axis each.



Example for the MP entries:

MP850.0 = 0
Axis 1 is master axis

MP850.1 = 1
Axis 2 is slave to axis 1

MP850.2 = 1
Axis 3 is slave to axis 1

MP850.3 = 2
Axis 4 is slave to axis 2

MP860.0 = 0 or 1
Axis 1: Datum for synchronous control

MP860.1 = 0 or 1
Axis 2: Datum for synchronous control

MP860.2 = 2
Axis 3 is torque slave axis

MP860.3 = 2
Axis 4 is torque slave axis

Activation of master-slave torque control

- Activate the master and slave axes with MP10.
- In MP110.x, define the position encoder for the master.
- Enter MP110.x = 0 for the slave.
- In MP850.x, define the master axis as the main axis and the slave axis as the tracking axis.
- Activate the master-slave torque control by entering MP860.x bit #1 for the slave axis.

Axes for which master-slave torque control is active can be switched by the PLC to single-axis operation during operation by overwriting MP850.x.

The new bit #2 in MP860.x is used to define that the brake of the slave axis is automatically included and checked in the brake test (PLC module 9143) of the master axis.

MP860.x	**Datum for synchronous control**
Input:	%xxx Bit #0 – Datum for synchronous control 0: Datum at position after switch-on 1: Datum at reference marks (machine datum) Bit #1 – Master-slave torque control 0: Not a torque slave axis 1: Axis is torque slave axis Bit #2 – Brake test of slave axis 0: No automatic brake test of the slave axis 1: Brake test of the slave axis is automatically run with master axis

Setting the master-slave torque control for minimizing mechanical play

- For the master and slave axes you must select in MP1040 the same or the opposite direction of rotation, depending on the application (MP210 has no effect on the slave).
- Adjust the current controller for the master and slave axes. See “Commissioning" on page 998.
- Enter the following temporary values in the machine parameters for the slave axis:
  MP2900.x = approx. 20% to 25% of the rated torque of the motor
  MP2910.x = 3
  MP2930.x = 0
- In MP2920.x, enter the ratio of the mass moment of inertia of the master to the mass moment of inertia of the slave. For identical motors, therefore, the value to be entered is 1.
- If you use a position encoder, enter 100 in MP2930.x for the slave axis; if you do not use a position encoder, enter the value 0 for the slave axis.
- Enter MP2510.x (I factor of speed controller) = 50 or, if you have one, an empirical value for your motor.
- Adjust the P and I factor of the speed controller for the master and slave axes at the same time. See See "Commissioning" on page 998. It is not permissible to commission the master and slave axes separately, since the motors must be tensioned during commissioning.
- If you do not reach the desired rise time (approx. 10 ms), you can increase the P factor with the aid of a filter. Here the band-rejection filter is preferable to the low-pass filter.
- To find the center frequency for the band-rejection filter, slowly increase the P factor to the oscillation limit and find the frequency with the integrated oscilloscope.

Note

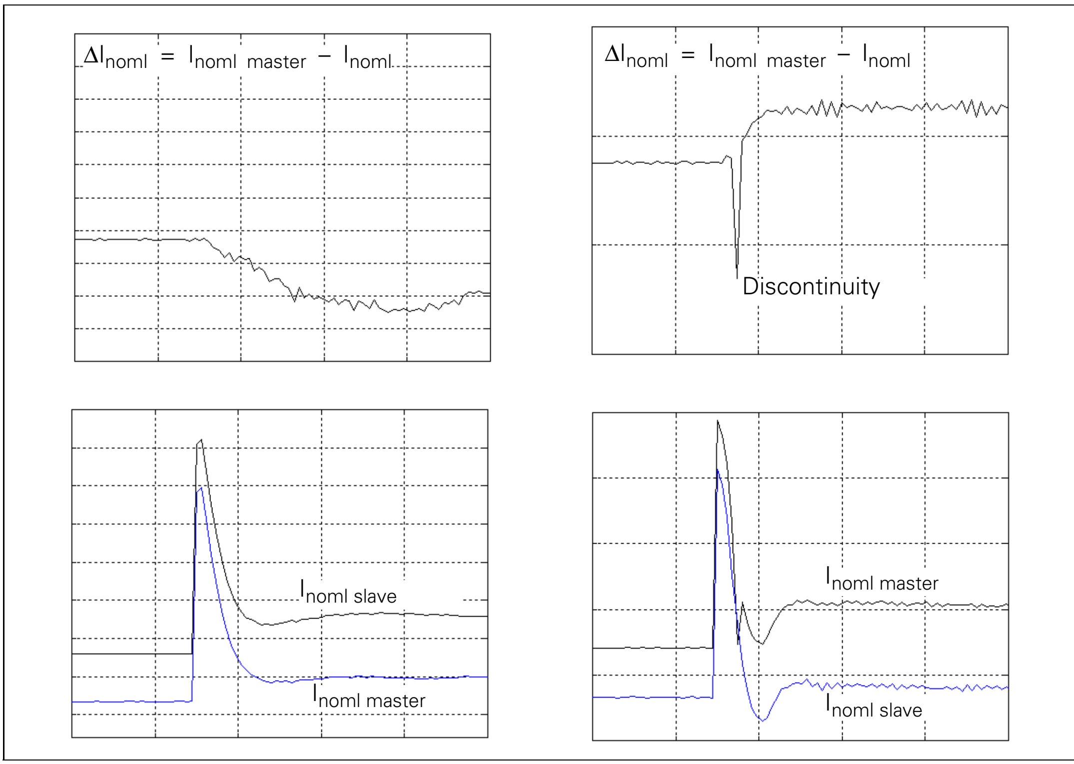
For low-frequency oscillations (< approx. 200 Hz) you should not use a filter, because it may have a negative influence on the dynamics of the control.
For the mid-range frequency (approx. 200 Hz to approx. 400 Hz) ensure that you do not excite any low-frequency oscillation.
The higher the frequency of the oscillation (> approx. 400 Hz), the less negative will be the influence of high damping on the dynamics.

Note

For identical motors, the factors of the speed controller should be identical to ensure identical dynamic behavior.

Test the tensioning torque:

- With the integrated oscilloscope, record the nominal current (I nominal) of the master and the slave axes at standstill.
- Send a step to the speed controller and, with the integral oscilloscope, record the nominal current of the master and slave axes.
- If there is a discontinuity in the course of the nominal current, increase the tensioning torque for the slave axis in MP2900.x.



> **Note**
>
> The lower the ratio of the total mass moment of inertia (transmission, machine table, etc.) to the motor mass moment of inertia, the smaller the required tensioning torque is (MP2900.x).

Test the P factor of the torque controller:

- With the integrated oscilloscope, record the actual speed value V (N ACTL).
- Increase the P factor in MP2910.x for the slave axis up to the oscillation limit.
- Enter in MP2910.x for the slave axis 50% of the resulting value.

Setting the master-slave torque control for torque distribution in a rigid design

- For the master and slave axes you must select in MP1040 the same or the opposite direction of rotation, depending on the application (MP210 has no effect on the slave).
- Adjust the current controller for the master and slave axes. See See "Commissioning" on page 998.
- Enter the following temporary values in the machine parameters for the slave axis:
  MP2900.x = 0
  MP2910.x = 3
  MP2930.x = 0
- In MP2920.x, enter the ratio of the mass moment of inertia of the master to the mass moment of inertia of the slave. For identical motors, therefore, the value to be entered is 1.
- If you use a position encoder, enter 100 in MP2930.x for the slave axis; if you do not use a position encoder, enter the value 0 for the slave axis.
- Enter MP2510.x (I factor of speed controller) = 50 or, if you have one, an empirical value for your motor.
- Deactivate the slave axis in MP10.
- For the master axis, adjust the P and I factor of the speed controller. See See "Commissioning" on page 998.
- If you do not reach the desired rise time (approx. 10 ms), you can increase the P factor with the aid of a filter. Here the band-rejection filter is preferable to the low-pass filter.
- To find the center frequency for the band-rejection filter, slowly increase the P factor to the oscillation limit and find the frequency with the integrated oscilloscope.

Note

For low-frequency oscillations (< approx. 200 Hz) you should not use a filter, because it may have a negative influence on the dynamics of the control. For the mid-range frequency (approx. 200 Hz to approx. 400 Hz) ensure that you do not excite any low-frequency oscillation. The higher the frequency of the oscillation (> approx. 400 Hz), the less negative will be the influence of high damping on the dynamics.

- Deactivate the master axis in MP10.
- Set MP850.x and MP860.x to 0 for the slave axis.
- Set the speed controller and the filter parameters for the slave axis in the same manner as for the master axis.

Note

For identical motors, the factors of the speed controller should be identical to ensure identical dynamic behavior.

Test the P factor of the torque controller:

- In MP10 reactivate the master and slave axes.
- With the integrated oscilloscope, record the actual speed value V (N ACTL).
- Increase the P factor in MP2910.x for the slave axis up to the oscillation limit.
- Enter in MP2910.x for the slave axis 50% of the resulting value.

MP2900.x **Tensioning torque between master and slave for master-slave torque control (entry for the slave axis)**
Input: –100.00 to +100.00 [Nm]

MP2910.x **P factor of the torque controller for master-slave torque control (entry for the slave axis)**
Input: 0.00 to 999.99 [1/(Nm · min)]

MP2920.x **Factor for variable torque distribution for master-slave torque control (entry for the slave axis)**
Input: 0.000 to 100.000
1: Master and slave axes have identical motors

MP2930.x **Speed compensation ratio for master-slave torque control (entry for the slave axis)**
Input: -100.00 to +100.00 [%]

6.7 Reference Marks

6.7.1 Definition

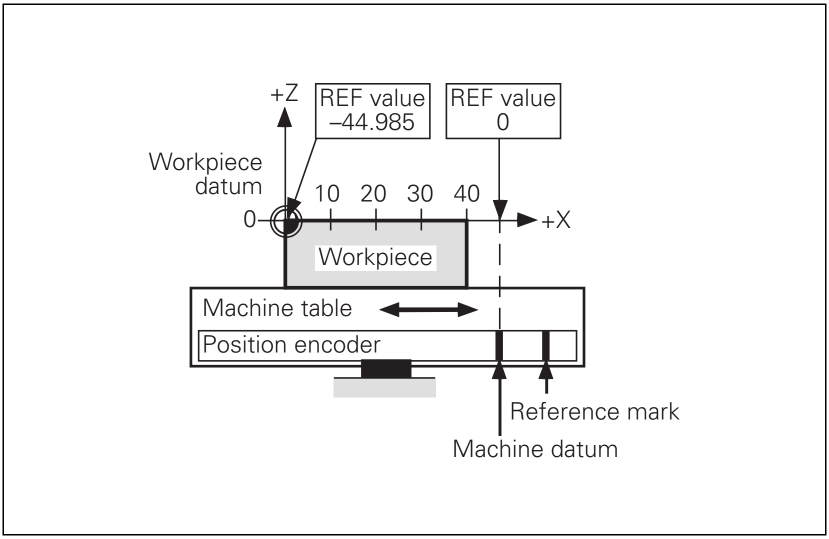
The position value (the coordinates) of an axis position is defined with respect to a freely selectable datum. When the axes are moved, the ACTUAL position is calculated incrementally. If there is an interruption in power, the reference between the axis position and the position value is lost.

Reference marks

HEIDENHAIN linear encoders are designed with one or more reference marks. The reference marks identify an axis position at a known distance from the machine datum. The position of the freely selectable datum is defined with respect to the machine datum.

The datum and the actual position can be reproduced as soon as the reference marks are traversed.

HEIDENHAIN recommends position encoders with distance-coded reference marks. With distance-coded reference marks, the position value can be reestablished after traverse of a short distance over any two reference marks.



6.7.2 Traversing the Reference Marks

The reference marks must be traversed after any interruption in power:

- Press the machine START button: The reference marks are automatically traversed. The sequence of axes is predetermined.

or:

- Press the machine axis-direction button. The user determines the sequence of the axes.

After the reference marks have been traversed:

- The software limit switches are activated.
- The most recently saved datum and machine datum are reproduced.
- PLC positioning and positioning with M91 and M92 become possible.
- The counter is set to zero for axes in an open loop.

Distance between the scale reference point and the machine datum

For position encoders with distance-coded reference marks, the machine datum is defined with respect to the scale reference point, which is at the first reference mark after the beginning of the measuring length. On angle encoders, the scale reference point is marked.

- In MP960.x, enter the distance between the scale reference point and the machine datum.

For position encoders without distance-coded reference marks but with more than one reference mark, the distance between the reference mark to be traversed and the scale reference point must also be entered:

- With Module 9225, enter the distance between the reference mark to be traversed and the scale reference point.

Module 9225 Compensation value for the reference mark
With Module 9225 you define the distance between the reference mark to be traversed and the scale reference point for the NC and PLC axes.

Call:

PS	B/W/D/K	<>Axis>
		0 to 8: Axes 1 to 9
		15: Spindle
PS	B/W/D/K	<>Compensation value in 0.1 µm>
		0: Reference mark to be traversed = scale reference point
CM	9225	
PL	B/W/D	<>Error code>
		1: Axis does not exist

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Axis not found

Assigning a reference value

In some cases a new reference value may have to be assigned to an axis, e.g. if an axis is mechanically fixed and the encoder is moved. Since due to the mechanical fixing the position of the axis cannot be changed, you can assign it a new reference value.

- Enter the new reference value in Module 9147.

Module 9147 Assigning a reference value to an axis
If a new reference value is assigned to an axis, the corresponding bit is reset in W1032.

Call:
PS B/W/D/K <>Axis number>
0 to 8: Axes 1 to 9
PS B/W/D/K <>New reference value in 0.1 µm>
CM 9147

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing strobe in M4176 = 1
	24	Module was called in a spawn or submit job

Defining the process of traversing the reference marks

In machine parameters, you define the process of traversing the reference marks:

- In MP1320.x and MP1330.x (for rotary encoders also in MP1331.x) you define the direction and velocity for traversing the reference marks.
- In MP1340.x you define the sequence of axes for traversing the reference marks.
- With MP1350.x you select the type of reference marks.

External reference pulse

If the reference mark of the encoder cannot be used, e.g. owing to an unfavorable transmission of motor and rotary axis, an external reference pulse may be evaluated.

- In MP4130.x, define the fast PLC input for the external reference pulse
- For the corresponding axis in MP1360.x, enter the number of the fast PLC input
- Enter MP1350.x = 6 for the corresponding axis

This can also be used for linear axes. Activation of the fast PLC inputs (MP4130.x) makes it possible to interrogate the inputs in the controller cycle. This controller cycle time and the traversing speed when crossing the reference mark mainly determine the accuracy of the reference run:

- Fast traverse speed --> less accurate
- Longer controller cycle time --> less accurate

It may be possible to achieve sufficient accuracy with two reference runs (first fast, then slow). Specify the logic for the fast PLC inputs with MP4131.2-5.

"Pass Over Reference Point" operating mode

The NC uses W272 to report the "Pass Over Reference Point" operating mode to the PLC.

If you switch the operating mode before all reference marks are traversed, the **PASS OVER REFERENCE** soft key prompts you traverse the remaining reference marks. In W1032 the PLC receives the information as to which axes have not yet been referenced.
In W1032, the bits for axes that are not to traverse the reference marks (MP1340.x = 0) remain set.

In the NCMACRO.SYS file, after the code number RESETINIT= you can enter the name (incl. path) of a macro that will be called when the Pass Over Reference Point mode of operation is exited. If the NC macro is terminated once with **END PGM** or **M02,** it will no longer be run when the **Pass Over Reference Point** mode is called and exited.
If M4622 is set during the first run of the PLC program, the message window **Waiting for M4622** appears after the reference marks have been traversed. The window does not disappear until you have reset M4622. In this way, you can delay the execution of the NC macro defined through RESETINIT = in the NCMACRO.SYS.
To synchronize the current machine status and the look-ahead calculation with an NC macro call, See "NCMACRO.SYS" on page 1644.

Reference end position

To prevent the axes from violating their traverse limits when traversing the reference marks, each axis requires a trip dog (at the reference end position). The trip dogs must be installed by the machine tool builder at the ends of the traverse range. The switch signals from the trip dogs are sent to free PLC inputs. The PLC program must gate these PLC inputs with W1054 for "reference end position."

The axis will only automatically be positioned to the software limit switch if

- It is beyond the positive software limit switch and is moving in the positive direction to the positive trip dog.
- It is beyond the negative software limit switch and is moving in the negative direction to the negative trip dog.

Activate the software limit switches before traversing the reference marks

Controls with software version 340 49x-04 and higher save the last axis positions that were traversed before switch-off. These positions allow you to check the traverse ranges of the axes when traversing the reference marks. This can be activated for a specific axis in MP1352 and is switched on/off with the MONITOR SW LIMIT soft key. This soft key is available after the control has been switched on in the "reference mark traverse" operating mode.

Danger

- When approaching the reference marks again (e.g. after switching the axis), this monitoring function is not active.
- If fatal error messages occur, the position at shutdown may not be exactly identical to the position at switch-on.
- Please inform your customers for which axes you have enabled the activation of the software limit switches in MP1352, and for which axes you have not enabled it. The MONITOR SW LIMIT soft key, specifically for switching monitoring off, is available if MP1352 is set for at least one axis. The user, however, is not informed for which axes the software limit-switch monitoring function before reference-mark traverse is active.

Encoders with EnDat interface

Encoders with EnDat interface can be connected to the position and speed inputs of the MC 42x(B) and CC 42x. With these encoders there is no need to traverse the reference marks. The position value is only read when the control is switched on. It cannot be read again.
When connecting a position encoder with an EnDat interface:

- Enter MP1350.x = 5.

When connecting a speed encoder with an EnDat interface:

- The iTNC automatically attempts to communicate with the encoder.

When connecting a speed encoder with an EnDat interface as a position encoder:

- Enter MP1350.x = 5.
- In MP110.x, enter 0 for the axis with the speed encoder with EnDat interface.

Note

If use of multiturn encoders with EnDat interfaces results in overruns, the corresponding information is entered in the system file NCDATA.SYS. For a control exchange, this file must be transferred or MP960.x must be readjusted.

Setting MP960.x (machine datum) when using rotary encoders with EnDat interface:

- Move the axis (encoder) to the nominal datum.
- Enter the displayed value of the deviation in MP960.x. Reverse the algebraic sign.

or:

- Set the cursor in the active machine-parameter file to the corresponding MP960.x.
- Press the actual-position-capture key. The value is automatically entered in MP960.x.

Double reference run

During the double reference run, the absolute position is first output via the EnDat interface of the speed encoder. If at a later time the reference mark of the position encoder is traversed, the control continues to work with this reference.

- Set the corresponding bits in MP1355 to 1 for the axes for which the double reference run is to be used.

The distance between the speed encoder and the position encoder must be entered in MP1356.x. When the reference mark of the position encoder is **first** traversed, the message `Set MP1356.<axis number> to <value>` appears.

- Enter this value in MP1356.x.

Up to 340 422-04, 340 480-04: Depending on the position encoder being used, W1032 (reference marks not yet traversed) can have different meanings:

- Position encoder **with** distance-coded reference marks: After reading the absolute position via the EnDat interface, W1032 is reset for the affected axis.
- Position encoder **without** distance-coded reference marks: After reading the absolute position via the EnDat interface, W1032 remains set for the affected axis.

As of 340 422-05, 340 480-05:

- Define the behavior of W1032 in MP1357.x.

MP960.x	**Machine datum**
Input:	–1.79769313486E+308 to +1.79769313486E+308 [mm] or [°] Values with respect to the scale reference point
MP1320	**Direction for traversing the reference marks**
Format:	%xxxxxxxxxxxxxx
Input:	Bits 0 to 13 represent axes 1 to 14 0: Positive 1: Negative
MP1330.x	**Velocity for traversing the reference marks**
Input:	80 to 1 000 000 [mm/min]
MP1331.x	**Velocity for leaving the reference mark end position for axes 1 to 9 (only for rotary encoders MP1350 = 2)**
Input:	10 to 1 000 000 [mm/min]
MP1340.x	**Sequence for traversing the reference marks**
Input:	0: No evaluation of reference marks 1 to 14: Axes 1 to 14

MP1350.x **Sequence for finding the reference mark**

Input: 0: Linear encoder with distance-coded reference marks (old routine)
1: Position encoder with one reference mark
2: Special type (length measurement with ROD)
3: Linear encoder with distance-coded reference marks (new routine)
4: Same as 3 except that two reference marks are evaluated
5: Encoder with EnDat interface
6: Reference pulse via fast PLC input

MP1352 **Activate the software limit switches before traversing the reference marks**

Format: %xxxxxxxxxxxxxx
Input: 0 = Software limit switch not active
1 = Software limit switch active

MP1355 **Double reference run**

Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Reference run as defined in MP1350.x
1: Double reference run

MP1356.x **Distance between speed and position encoder for double reference run.**

Input: –99 999.999 to +99 999.999 [mm] or [°]

MP1357.x **W1032 for double reference run**

Input: 0: Reset W1032 if the reference run has been over the EnDat interface of the speed encoder
1: Reset W1032 if the reference mark was traversed with the position encoder

MP1360.x **Fast PLC input for reference pulse**

Input: 0: No fast PLC input for reference pulse
1 to 5: Fast PLC input for reference pulse (MP4130.x)

		Set	Reset
M4622	**Delay NC macro with RESETINIT = from NCMACRO.SYS**	PLC	PLC
W1032	**Reference marks not yet traversed** Bits 0 to 8 represent axes 1 to 9	NC	NC
W1054	**Reference end position** Bits 0 to 8 represent axes 1 to 9	PLC	PLC

Renewed traversing of the reference marks

Module 9220 Renewed traversing of the reference marks

With this module you start an NC or PLC axis or a servo-controlled spindle for traversing the reference mark. It is possible to repeat the reference mark traverse in an axis that has already been referenced. The module can be called in all operating modes. Software limit switches are not effective. The strobe marker must remain set for the entire duration of the reference-mark traverse.

Axis:

- The sequence of functions (MP1350.x) and the velocity for leaving the reference end position (MP1331.x) are defined by machine parameter.
- The velocity and the direction for traversing the reference marks are either taken from MP1330.x and MP1320.x or they are defined in the module.

Note

The direction of traverse should be defined in the module only in exceptional cases. Since the reference end positions are not considered in this case, the limits of the traverse range may be violated.

- If an axis is started for reference point traverse although the reference mark has already been traversed, the corresponding bit is set in W1032 and the reference mark is traversed again. The same constraints apply as when traversing the reference mark for the first time.
- An axis cannot be started for referencing until all other axes are in position.

Servo-controlled spindles:

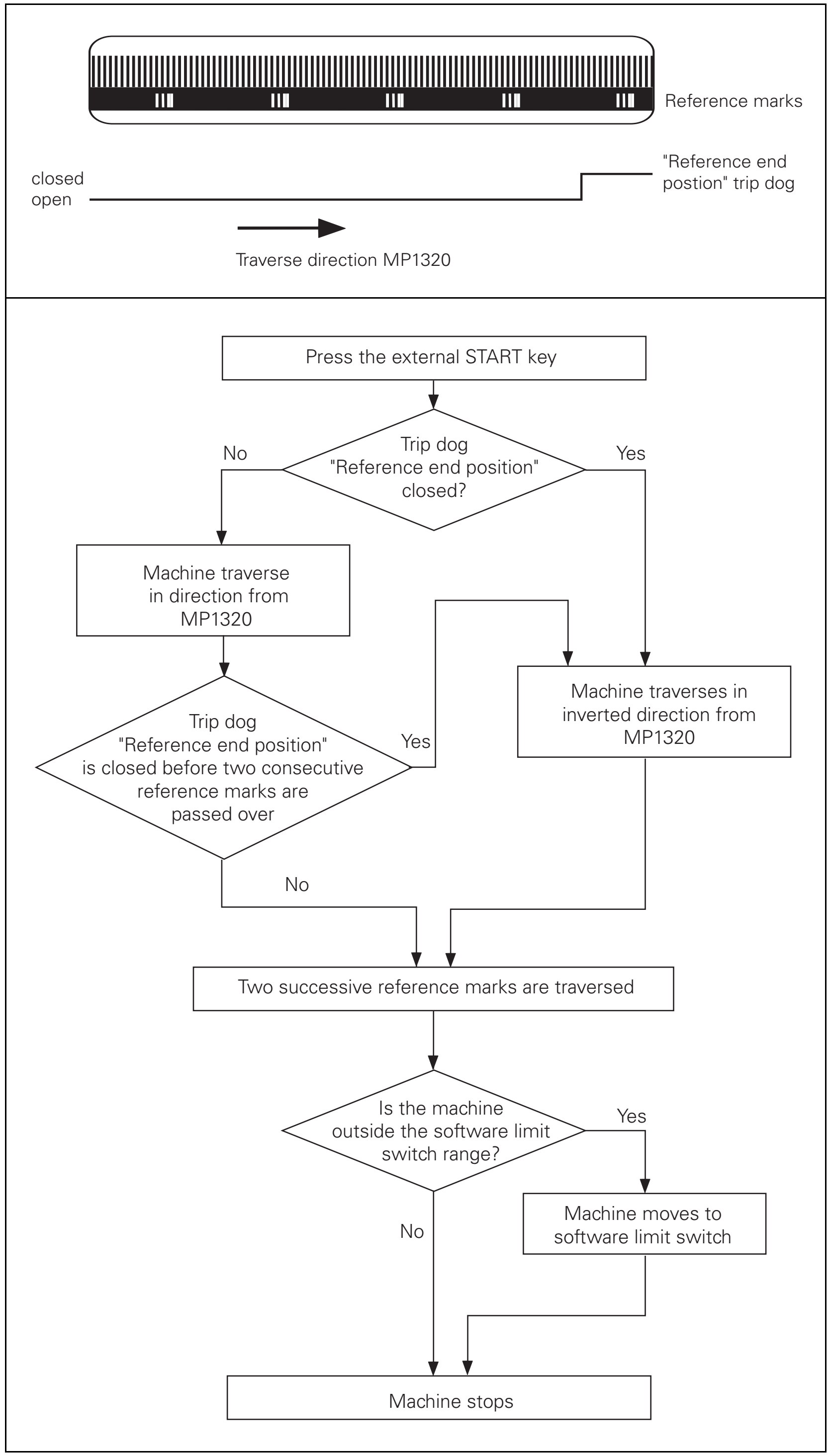
- The speed for traversing the reference mark is defined in the module.
- The spindle must be started from a standstill to traverse the reference mark.
- If the spindle is started for reference mark traverse, marker M4018 is set.

Call:

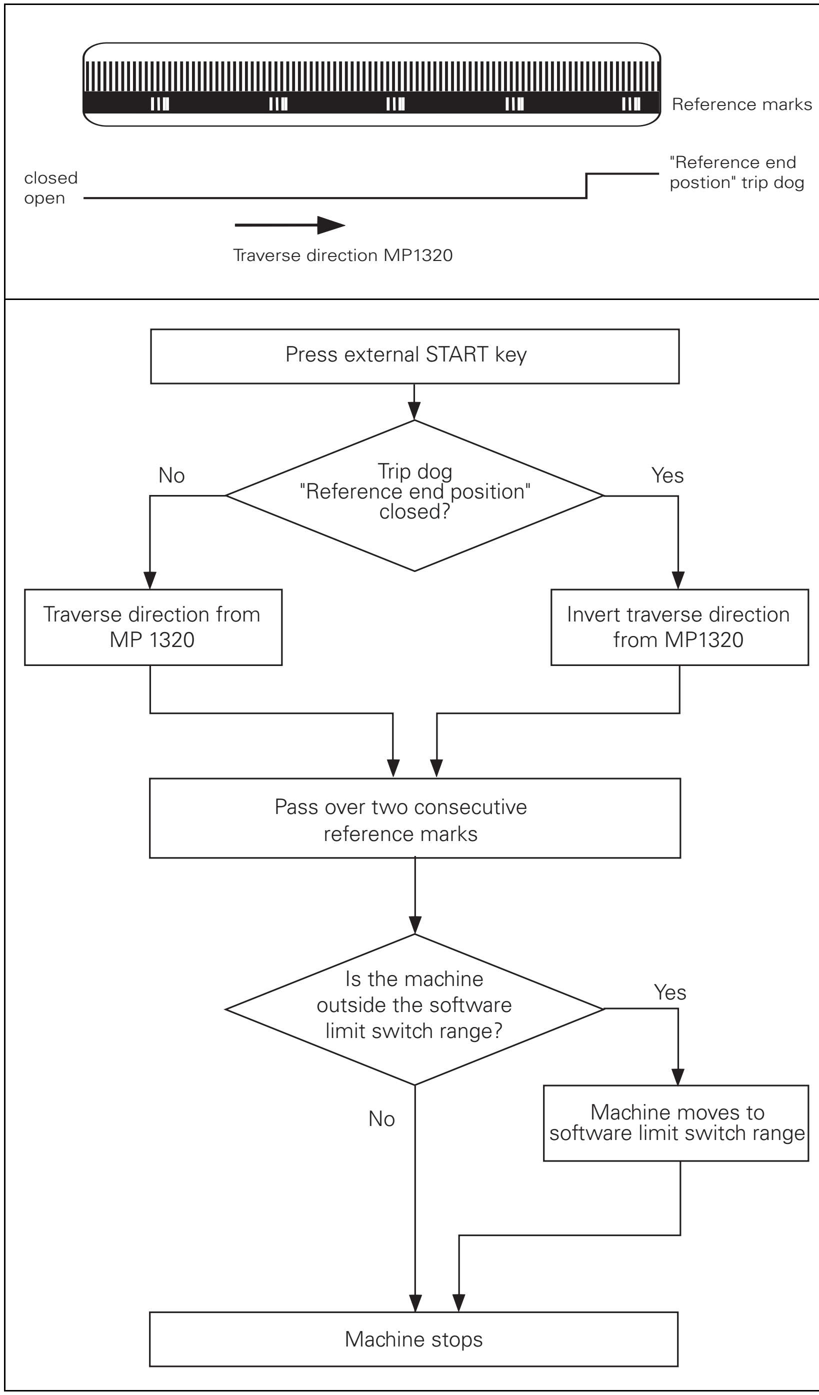
PS	B/W/D/K	<>Axis/spindle> 0 to 8: Axes 1 to 9 15: Spindle
PS	B/W/D/K	<>Feed rate/shaft speed> 0: Feed rate MP1330.x >0: Feed rate in mm/min or shaft speed in 1/1000 min^{-1}
PS	B/W/D/K	<>Direction of traverse> –1: Negative direction 0: Direction from MP1320.x 1: Positive direction
CM	9220	
PL	B/W/D	<>Error code> 0: Reference mark traverse is commanded 1: Axis does not exist, or not a servo-controlled spindle 2: Inadmissible values for the feed rate / direction 4: Reference traverse not possible because reference traverse already started 5: Axis is already being positioned or the spindle is in motion 6: Other axis is already being positioned 8: Programmed axis not in closed loop

Position encoder with distance-coded reference marks

Function when MP1350.x = 3



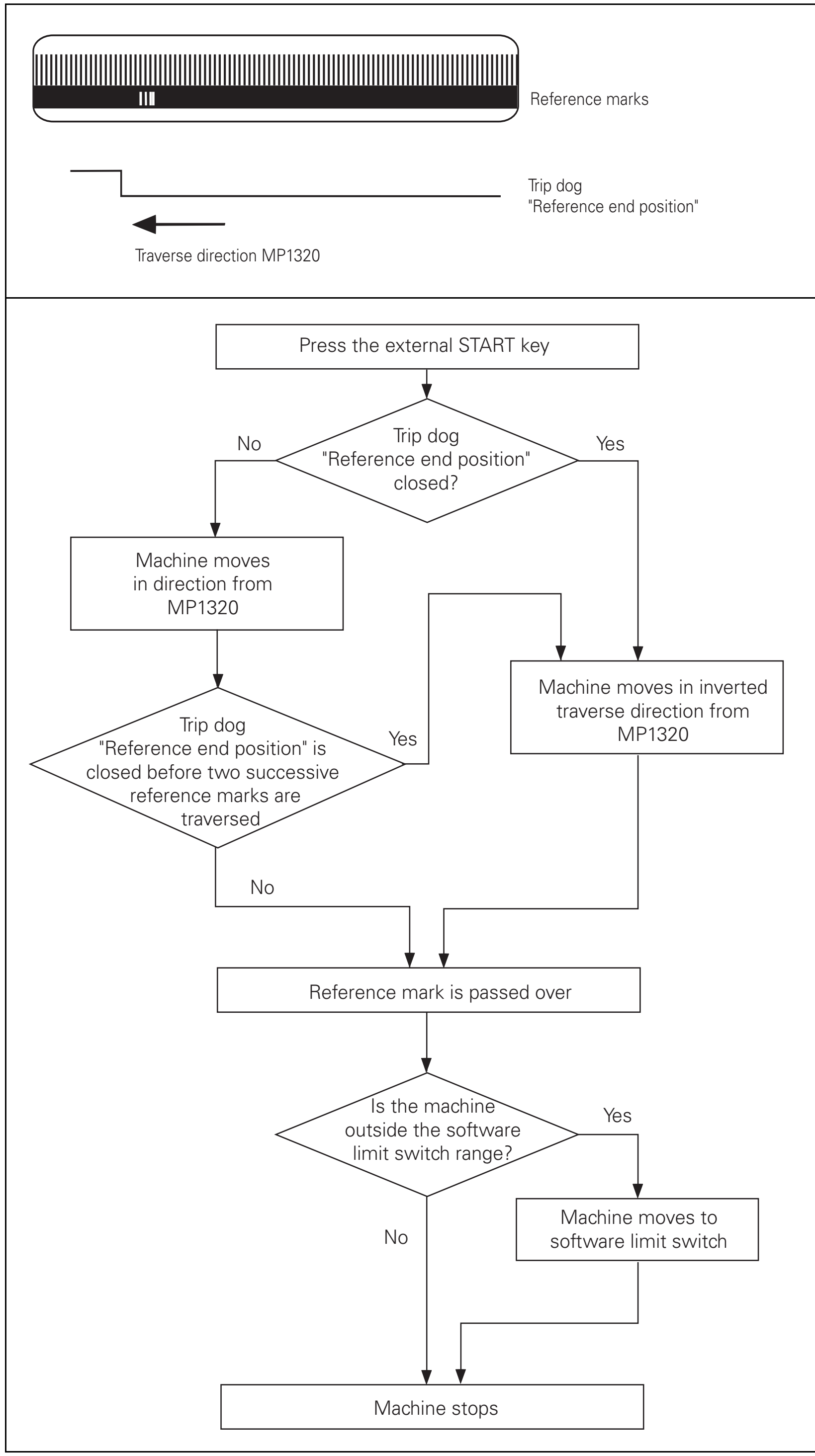
Function when MP1350.x = 0. This setting is used only to ensure compatibility. Do not use for new installations.



If during automatic referencing the trip dog is not closed until it is in the reference end position range, the contouring control will ignore this signal. It is therefore necessary that there are at least two reference marks in the range of the reference end position.

Position encoder with one reference mark

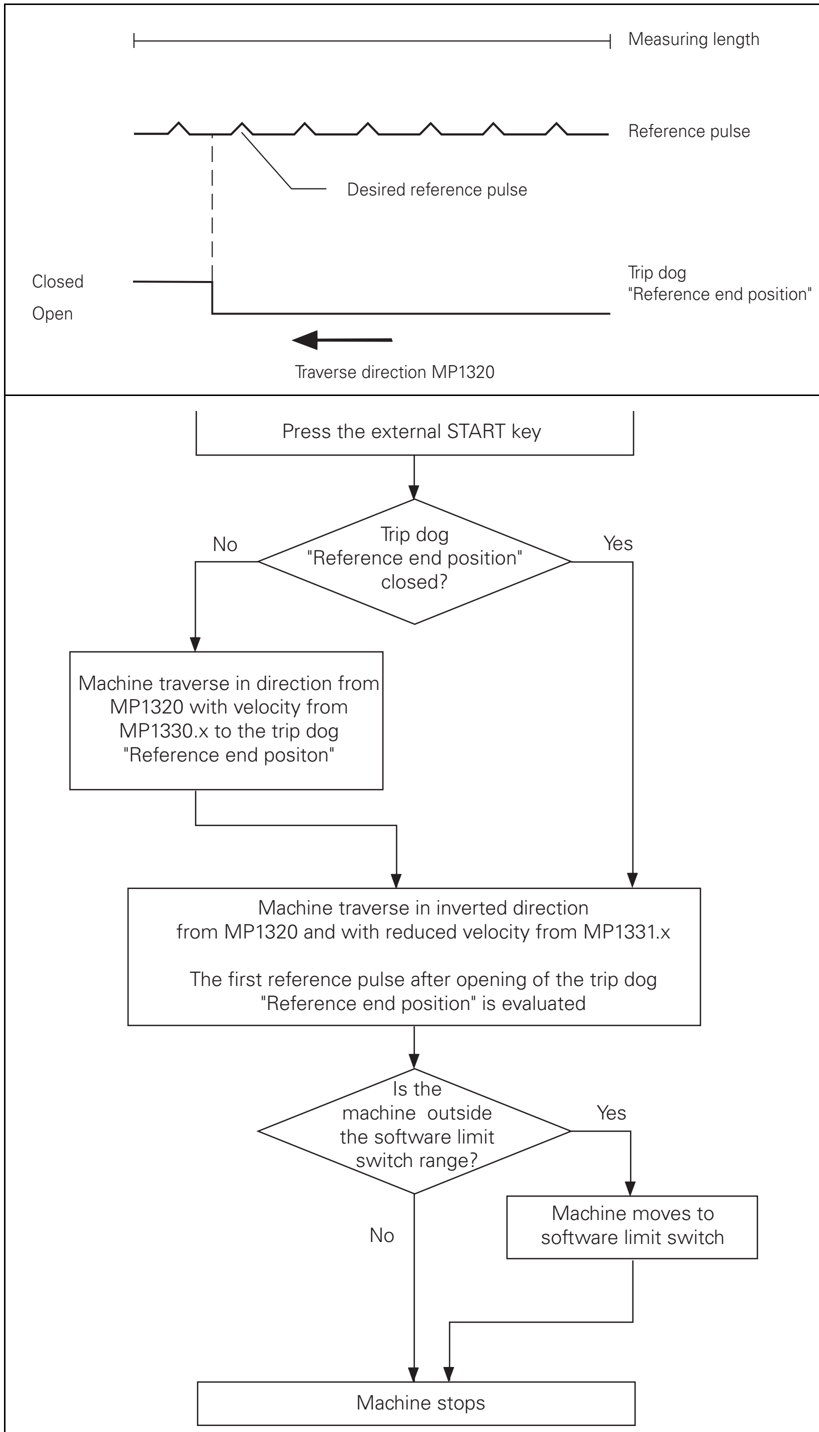
Function when MP1350.x = 1



Linear measurement through rotary encoder

Function when MP1350.x = 2

For linear measurement using a rotary encoder, a reference pulse is produced at each revolution of the encoder. Ensure that during referencing the same reference pulse is always evaluated. This can be realized with the trip dog for reference end position.



6.8 The Control Loop

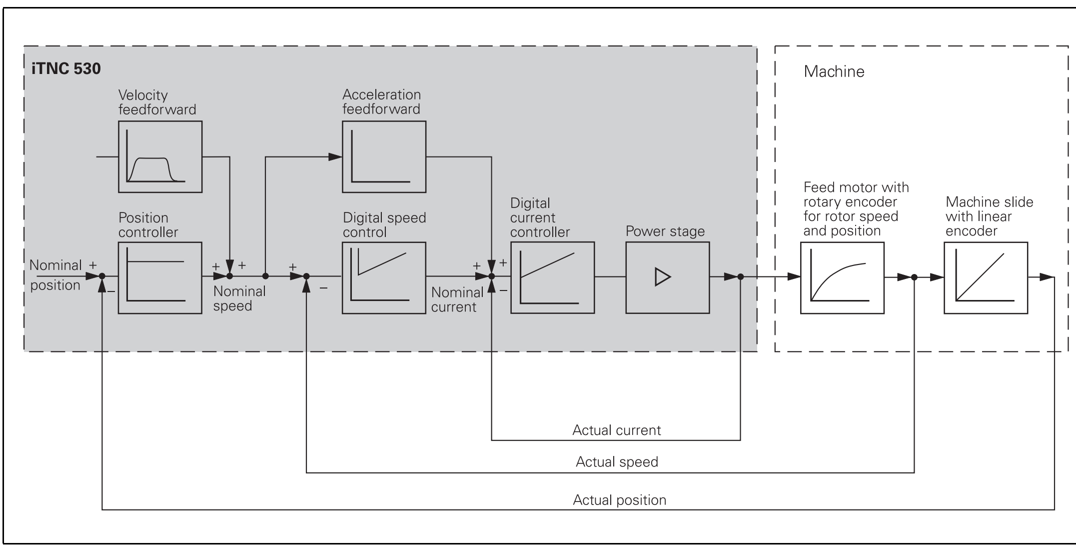
Machine tools normally function on the principle of cascade control. Here the position control loop is prior to the speed and current control loops.

Benefits of cascade control:

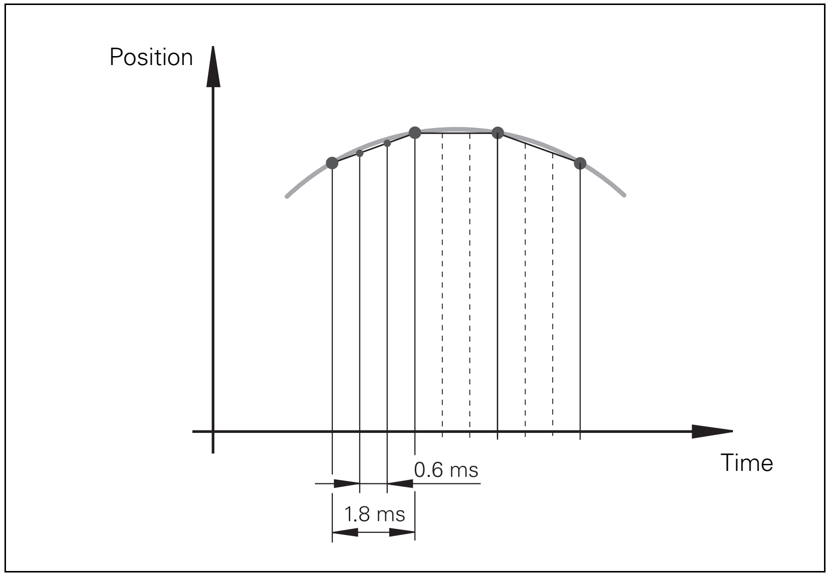
- Transparent structure of the individual control loops.
- Disturbances can be compensated through the subsequent controllers. This relieves the prior controller.
- The respective outer control loop protects the inner control loop by limiting the command variable.
- Individual commissioning of each control loop, starting with the innermost loop.

The position, speed, and current controllers, and the power module are integrated in the iTNC. The power module is driven by the CC 42x through PWM signals (PWM = pulse width modulation).

The iTNC 530 controls machines with up to 11 axes and a spindle or up to 10 axes and 2 spindles. Spindle speeds up to 40 000 min^{-1} for motors with two pole pairs are possible.



The **position controller cycle time** is the time interval during which the interpolation points on the path are calculated. The **speed controller cycle time** is the time interval in which the actual speed value is compared to the calculated nominal speed value. The **current controller cycle time** is the time interval in which the actual current value is compared to the calculated nominal current value.



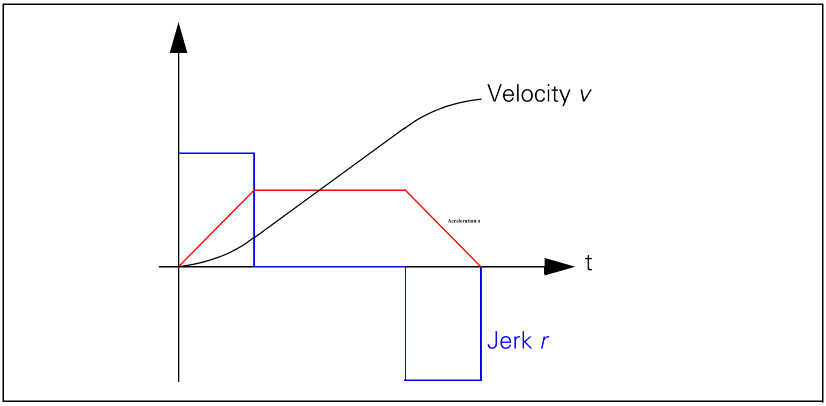
6.8.1 Relation Between Jerk, Acceleration, Velocity and Distance

To ensure proper operation of an axis, the following two conditions must be fulfilled:

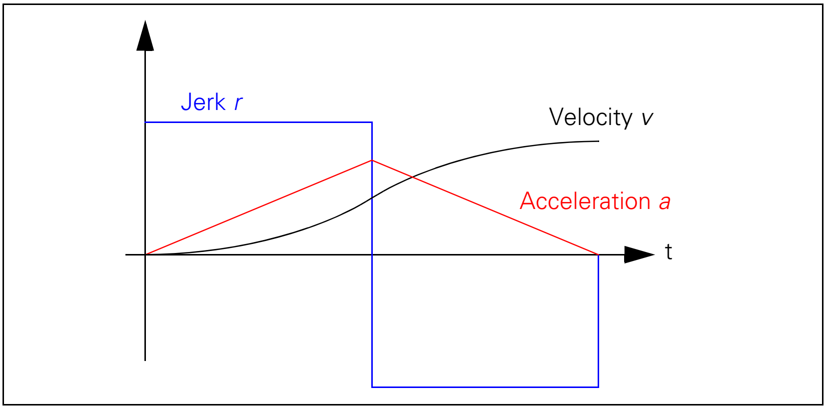
- The desired maximum speed v_{max} and maximum jerk r_{max} result in a maximum acceleration a_{max}.
- A minimum distance s_{min} must be traversed in order to attain the maximum speed v_{max}.

Maximum acceleration

Taking into account the motor and the power module, the machine should be designed in such a way that acceleration during the acceleration phase is as constant as possible. This ensures maximum utilization of the drive current.



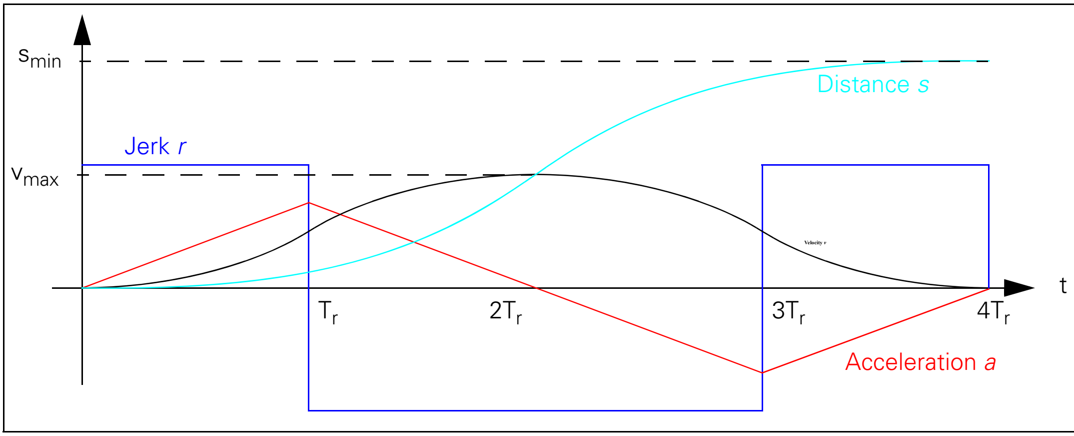
On the other hand, the machine should also be designed to fulfill the dynamic requirements. The jerk should be kept to a minimum and the jerk phase should be maximized in order to prevent the machine from oscillating. The result is no constant acceleration, but a short acceleration peak. If the maximum velocity and the maximum permissible jerk of the machine are preset, the maximum attainable velocity can be determined.



$$a_{max} = \sqrt{v_{max} \cdot r_{max}}$$

Minimum distance

To attain the maximum velocity, a minimum distance s_{min} must be traversed. If the traversed distance is greater than s_{min}, a movement with constant speed is inserted at the time $2T_r$. The minimum distance is:



$$s_{min} = 2 \cdot v_{max} \cdot \sqrt{\frac{v_{max}}{r_{max}}}$$

Example

Rapid traverse v_{max} = 30 000 mm/min (= 0.5 m/s); MP1010.x = 30000
Max. jerk with velocity v > 20 000 mm/min (= 0.33 m/s) r_{max1} = 70 m/s^3; MP1090.1 = 70, MP1092 = 20000
Max. jerk r_{max2} = 35 m/s^3 during machining; MP1090.0 = 35

Maximum attainable acceleration a_{max1} during rapid traverse:

$$a_{max1} = \sqrt{v_{max} \cdot r_{max1}} = \sqrt{0{,}5\ \frac{m}{s} \cdot 70\ \frac{m}{s^3}} = 5{,}92\ \frac{m}{s^2}$$

Maximum attainable acceleration a_{max2} during machining (v up to 20 000 mm/min):

$$a_{max2} = \sqrt{v_{max} \cdot r_{max2}} = \sqrt{0{,}33\ \frac{m}{s} \cdot 35\ \frac{m}{s^3}} = 3{,}40\ \frac{m}{s^2}$$

Distance s_{min} required to attain rapid-traverse velocity:

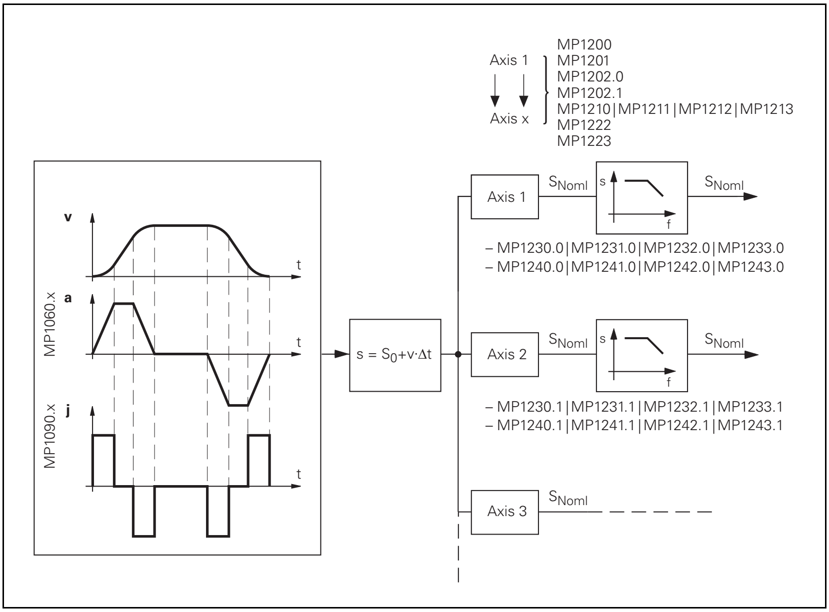
$$s_{min} = 2 \cdot v_{max} \cdot \sqrt{\frac{v_{max}}{r_{max}}} = 2 \cdot 0{,}5\ \frac{m}{s} \cdot \sqrt{\frac{0{,}5\ \frac{m}{s}}{70\ \frac{m}{s^3}}} = 0{,}085\ m = 85\ mm$$

Note

The rectangular jerk curve is rounded through the use of a nominal position value filter (MP1096.x ≠ 0). As a result, acceleration is reduced and the minimum distance required for attaining the maximum velocity is increased.

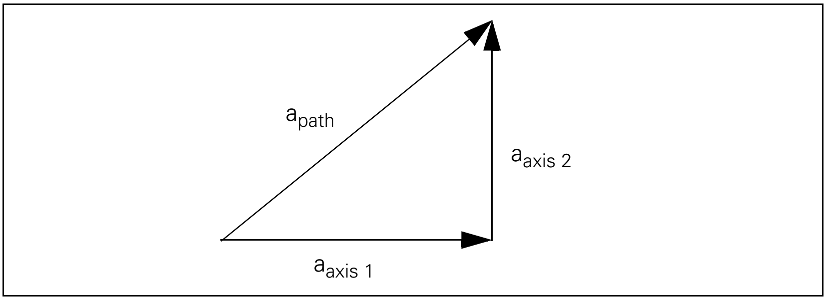
6.8.2 Interpolator

Schematic of the Interpolator:



The interpolator calculates a velocity from the programmed feed rate. With the CC 424(B) this calculation occurs every 3 ms, and with the CC 422 the frequency of the calculation depends on the time in MP7600.x. The value is also dependent on the acceleration curve and the end position.

If more than one axis is moved simultaneously, the path acceleration a_{path} is formed from the appropriate axis components. The same applies to rapid traverse in the path (See "Rapid traverse and feed rate limitation" on page 850).



If the inverter is not designed for such accelerations, you can limit the path acceleration:

- Enter in MP1061 the maximum permissible path acceleration.

You must adjust the velocity feedforward value to the dynamics of the machine:

- With MP1060.x you define the acceleration or the steepness of the velocity curve.
- In MP1090.x, you limit the jerk for the **Program run full sequence** and **Program run single block** modes of operation. Just as the acceleration is the rate of change in velocity, the jerk is the rate of change in acceleration. The greater the entered value, the more the system will tend to oscillate.
- With MP1085.x you limit the axis-specific jerk for path contours in the **Program Run, Single Block, Program Run, Full Sequence** and **MDI** operating modes, as long as the feed rate is not equal to FMAX or is less than the value in MP1092. Until now, this value was set for all axes in MP1090.0.
  The value in MP1090.0 will continue to be used as the maximum permissible jerk for motions in all axes. Even with interpolating axes this value is the maximum jerk for the entire machine.
  The formula shown can be used to calculate a guide value for MP1085. The formula is conceived for the calculated jerk to be large enough that the acceleration of the axis is not impaired.
  Input: 0.1 to 1000 [m/s3]

$$MP1085 \geq \frac{MP1060^2 \cdot 60000}{MP1010}$$

- With MP1086.x you limit the axis-specific jerk for single-axis motions at rapid traverse in the **Program Run, Single Block, Program Run, Full Sequence** and **MDI** operating modes, if the feed rate is FMAX or is greater than the value in MP1092. Until now, this value was set for all axes in MP1090.1.
  The value in MP1090.1 will continue to be used as the maximum permissible jerk for rapid-traverse movements in all axes. Even with interpolating axes this value is the maximum jerk for the entire machine.
  The formula shown can be used to calculate a guide value for MP1086. The formula is conceived for the calculated jerk to be large enough that the acceleration of the axis is not impaired.
  Input: 0.1 to 1000 [m/s3]
  0: Not active (then also for FMAX value in MP1085.x)

$$MP1086 \geq \frac{MP1060^2 \cdot 60000}{MP1010}$$

Note

HEIDENHAIN recommends entering the permissible jerk for each axis in **MP1085.x and MP 1086.x.** This way the jerk is based on the weakest axis participating in a motion. The value in MP1090.x should be chosen correspondingly greater than until now (340 49x-02) or greater than the largest axis-specific jerk (or the value 0 should be entered to omit limiting by MP1090.x) so that the control has the possibility of using the optimum jerk and therefore the optimum acceleration for interpolating axes.

- If you enter a jerk for one or more axes (e.g. rotary axes) that is lower than the value previously (before 340 49x-03) entered in MP1090.x, reduced speeds can occur during contouring movements. However, this behavior only occurs if you used MP1090.x to subject axes to a greater than ideal load for the axes. If you want to avoid this behavior, then you must enter the same values in MP1085.x or MP1086.x and MP1090.x, which results in the same behavior as previously (340 490-02). However, this leads to a loss of the speed advantage gained with the interpolation of axes with NC software 340 49x-03. The calculated path jerk is always ≥ to the smallest axis-specific jerk of the participating axes.
- Use MP1087.x to limit the axis-specific jerk in **Manual mode.**
- For unfiltered positioning movements, such as PLC positioning movements or positioning movements in measuring cycles, MP1088.x can be used to define axis-specific jerk limitation.
- Use MP1089.x to limit the axis-specific jerk in the **Pass Over Reference Point** mode of operation. This is necessary if you want to brake or accelerate faster in this operating mode than in other operating modes.

Please note:

$$\text{Jerk} = \frac{a^2}{v}$$

At high feed rates (e.g. rapid traverse) a higher jerk is permitted than at low feed rates:

- Enter the jerk for low feed rates in MP1090.0, and for high feed rates in MP1090.1. MP1090.x is the jerk on the tool path. The input value is determined by the weakest axis.
- With MP1090.x you can limit the path jerk that results from MP1085.x or MP1086.x. MP1090.0 takes effect for movements not at FMAX or if the feed rate is less than the value in MP1092. MP1090.1 takes effect for movements at FMAX or if the feed rate is greater than the value in MP1092. This limitation can be switched off by entering the value 0 in MP1090.x.
- With MP1092 you specify a feed-rate threshold for the various jerk parameters in MP1085.x and MP1086.x.
  If the programmed feed rate is less than the value in MP1092, the parameters in MP1085.x apply. If the programmed feed rate is greater than the value in MP1092, the parameters in MP1086.x apply. This way you have the same behavior as previously for all axes via MP1090.0 and MP1090.1. Enter the value 0 in MP1092 to deactivate this feed-rate limit. Then the jerk parameters from MP1085.x apply to movements not at rapid traverse. The values in MP1086.x become active for movements at FMAX.

A nominal position value is acquired every 1.8 ms from the calculated velocity. For linear interpolation:

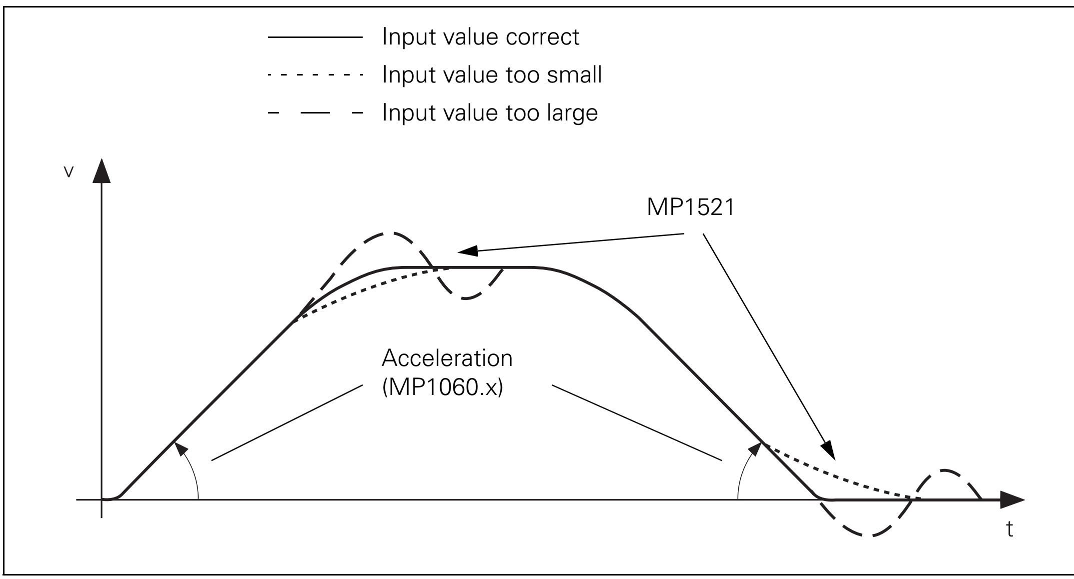
$$s = s_O + v \cdot \Delta t$$

s = nominal position value
s_o = previous nominal position value
v = calculated velocity
Δt = cycle time

The nominal position value is resolved into the individual axis components, depending on which axes have been programmed.

It may happen that the axes at first move past the target position and then oscillate onto it. This overshoot behavior during acceleration and braking can be influenced by a time constant:

- In MP1521, define the time constant for the overshoot behavior.



MP1060.x **Acceleration**
Input: 0.001 to 500 [m/s^2]

MP1061 **Limitation of the path acceleration**
Input: 0.001 to 500 [m/s^2]

MP1085.x **Axis-specific jerk for path movements in the operating modes Program Run Full Sequence, Program Run Single Block, and Positioning with Manual Data Input**
Input: 0.0 to 9999.9 [m/s^3 or °/s^3]

MP1086.x **Maximum permissible jerk during single-axis movements at rapid traverse for the operating modes Program Run Full Sequence, Program Run Single Block, and Positioning with Manual Data Input**
Input: 0: Function inactive
0.0 to 9999.9 [m/s^3 or °/s^3]

MP1087.x	**Maximum permissible axis-specific jerk for Manual mode**
Input:	0.1 to 1000.0 [m/s^3 or 1000°/s^3]
MP1088.x	**Axis-specific jerk limiting for unfiltered positioning movements**
Input:	0.1 to 9999.9 [m/s^3 or °/s^3]
MP1089.x	**Max. permissible axis-specific jerk for Pass Over Reference Point mode**
Input:	0.1 to 1000.0 [m/s^3 or 1000°/s^3]
MP1090.x	**Maximum permissible jerk on the tool path**
Input:	0.0 to 9999.9 [m/s^3 or °/s^3] 0: Not active
MP1090.0	With machining feed rate
MP1090.1	Beginning with feed rate from MP1092
MP1092	**Feed rate threshold for jerk parameters from MP1085.x and MP1086.x**
Input:	1 to 300 000 [mm/min] 0: Not active
MP1521	**Transient response during acceleration and deceleration**
Input:	1 to 255 [ms] 0: Function inactive

Feed-rate smoothing with MP1522

In addition to the nominal position value filters, you can activate feed-rate smoothing. This smoothes jerks caused by a change in the feed rate. This reduces the machine's tendency to vibrate, without significantly increasing the machining time.

An effective possibility for use is with machine oscillations under approx. 25 Hz. The advantage is that the feed-rate smoothing with MP1522 can dampen the oscillation behavior complementary to the nominal position value filters, which makes it possible to raise the limits of the nominal position value filters again. This could result in shorter workpiece machining times.

It is principally recommended that the frequencies of the nominal position value filters be set in dependency of the feed-rate smoothing.

The following calculation can be used to find the damping achieved by MP1522:

Limit frequency (f) of the feed-rate smoothing

$$fg(6dB) = \frac{1}{3} \times \frac{1000}{MP1522}$$

$$fg(12dB) = \frac{1}{2} \times \frac{1000}{MP1522}$$

Feed-rate smoothing (MP1522) with known oscillation frequency

$$MP1522(6dB) = \frac{1}{3} \times \frac{1000}{f}$$

$$MP1522(12dB) = \frac{1}{2} \times \frac{1000}{f}$$

- In MP1522, define the time constant according to the formula above for the feed-rate smoothing.
- Begin the adjustment with a smoothing of the feed rate in the 6-dB range, and increase the value step by step.
- Check the following error with a suitable NC program (e.g. feedforward adjustment program in TNCopt) until it reaches a minimal value.

Typical input value:

MP1522 = approx. 0.33 ... 0.5 · $T_{Machine}$
($T_{Machine}$ = Period of the machine resonance frequency)

Example:

$f_{Machine}$ = 50 Hz; $T_{Machine}$ = (1 / 20 Hz) = 50 ms;

MP1522 = 0.5 x 50 ms = 25 ms (input value = 25)

> **Note**
>
> There can be an increase in the following error at corners or contour transitions (change of direction) of an axis. Therefore, do not choose too large a value for MP1522, and only use feed-rate smoothing in combination with the nominal position value filters.

MP1522	**Feed-rate smoothing**
Input:	0 to 60 [ms] 0: Function inactive

Nominal position value filter

To attain a high machining velocity, the workpiece contour can be adapted to the machine dynamics by means of a nominal position value filter.

4 low-pass filters are available for limiting the bandwidth of the dynamics of nominal position and speed values.

- Single filter
- Double filter
- HSC filters
  - With Cycle 32 – mode 0 (HSC finishing filter)
  - With Cycle 32 – mode 1 (HSC roughing filter)
- Advanced HSC filters
  - With Cycle 32 – mode 0 (HSC finishing filter)
  - With Cycle 32 – mode 1 (HSC roughing filter)

Single and double filter

Single and double filters are classic low-pass filters that always smooth a contour towards the inside at changes in direction. They can be used when a very high surface quality is required, or when a high machining speed is required and larger tolerances are permitted.

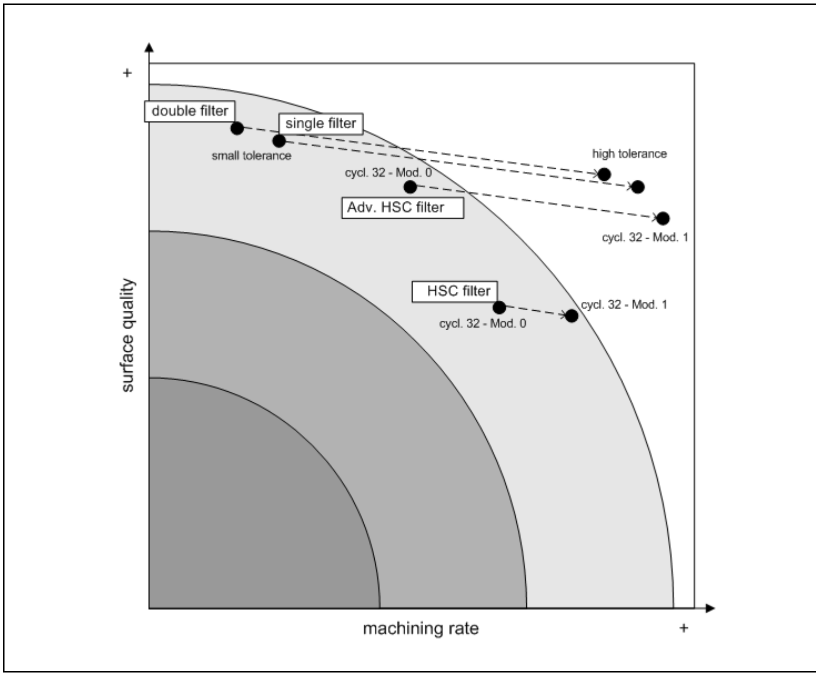
Compared to the double filter, the single filter permits a higher machining speed with a surface quality that is almost always comparable.

Depending on the cycle time, tolerance and axis-jerk limit values, the control automatically increases the filter frequency actually used (previously: filter order).

HSC filter and advanced HSC filter

The speed advantage of both HSC filters is especially large for circular contours. However, you must consider slight overshoots at corners and curvature transitions that are within the given tolerances (MP1202.x or Cycle 32). In general it is recommended that the advanced HSC filter be applied first. It has a more even effect on the speed and any possible overshoots.

The figure below gives a rough overview of when to use the nominal position value filters.



The nominal position value filters (MP1200) function in all operating modes (even in rapid traverse). For RIGID TAPPING (Cycle 17), the nominal position value filter is automatically switched off.

If the machine dynamics are adapted via a nominal position value filter, the iTNC always maintains

- the tolerance
  - from MP1202.0 (at the machining feed rate)
  - from MP1202.1 (at rapid traverse)
  - from the tolerance given in Cycle 32 – mode 0 (HSC finishing filter)
  - from the tolerance given in Cycle 32 – mode 1 (HSC roughing filter). Here the limit frequency is automatically lowered so that faster machining results for larger tolerances.
- the axis-specific jerk at corners (MP1230.x, MP1231.x, MP1232.x and MP1233.x, MP1250.x factor from MP123x.x for rapid traverse) depending on the filter type used
- the axis-specific jerk at curvature transitions (MP1240.x, MP1241.x, MP1242.x and MP1243.x) depending on the filter type used
- the axis-specific acceleration (MP1060.x)
- the path acceleration (MP1061)
- the radial acceleration (MP1070.x)
- the tolerance at curvature changes.

The tolerance consideration at curvature changes can be switched off with MP1222 for HSC filters and MP1224 for advanced HSC filters, which in most cases fully suffices for the accuracy, and has shown in the past that a significantly higher surface quality is achieved. These machine parameters are typically only used in combination with the HSC filter at resonant frequencies under 25 Hz when a high accuracy is required.

Selection criteria and settings for the nominal position value filters

The settings for the nominal position value filters mainly depend on the emphasis of the requirements for machining the workpiece. Speed and accuracy, in connection with clean and smooth surfaces, are the decisive criteria.

At the same time, the oscillation and resonance tendencies of the machining system (the machine tool) are to be considered, and taken into account in the settings for the nominal position value filters. The following recommendations can be made for the settings:

Clean surface

Definition of the term "surface":

- A clean and smooth surface has the highest priority
- Application: Finishing
- Oscillations in the axes must be damped, since following errors of 1 µm are still visible on the surface
- Tolerances are typically between 0.01 and 0.02 mm (may be slightly exceeded in order to achieve a better surface)

Settings guidelines for "surface":

- Preferential use of the advanced HSC filter (or single filter)
- Use M124 for a faulty NC program (from CAD system)
- Low jerk values in MP1090.0, MP123x.x
- Switch off tolerance consideration for curvature changes, MP1222 and MP1223 = 0.
- High jerk values for MP124x so that no limitations take effect (example: test MP124x.x up to a value of 100).
- Radial acceleration MP1070 maximum values between 1 and 3 [m/s^2].

Accuracy

Definition of the term "accuracy":

- Maintaining the tolerances has the highest priority.
- Slight oscillations can be seen on the surface.
- Tolerances typically between 0.005 and 0.01 mm.

Settings guidelines for "accuracy":

- Preferential use of HSC filter (MP1200 = 2) (with limit frequencies greater than 30 Hz).
- Lower jerk values in MP1090.0, MP1090.1, MP123x.x, MP124x.x than for "speed".
- Low radial acceleration values with MP1070 (typical values are < 1.5 m/s^2). Ideally, circular paths should be checked with a KGM grid encoder from HEIDENHAIN. However, in many cases the circular interpolation test with the integrated oscilloscope or TNCopt suffices.
- MP122x = 1 (consideration of tolerance limits at curvature changes).
- The adjustment should be tested with suitable NC programs. The **TNCopt** software from HEIDENHAIN features the "Contour-Single.h" NC program for this. The advantage is that the speed and the contour deviations can be seen directly.

Speed

Definition of the term "speed":

- Surface quality is secondary; short machining times have the highest priority.
- Application: Roughing.
- Tolerances typically between 0.1 and 0.2 mm.

Settings guidelines for "speed":

- Filter selection
  - For large tolerances (greater than 50 µm), preferential use of single filters (MP1200 = 0)
  - For small tolerances (less than 30 µm), preferential use of HSC filters (MP1200 = 2). Setting for MP1222 = 0.
- High jerk values in MP1090.0, MP1090.1
- High jerk values in MP123x.x, MP124x.x (typical: factor 3 to 5 of the path jerk)
- High radial acceleration via MP1070 (input value should not exceed the values for linear axes from MP1060.x, maximum 5 [m/s^2])
- Use M124 for a faulty NC program (from CAD system)
- When using the HSC filter: Program a high tolerance for rotary axes and roughing filters (mode 1) via Cycle 32. Typical values for the rotary-axis tolerances are around 0.01° to 5°. This can achieve significant speed advantages. However, possible undercuts must be taken into account.
- The adjustment should be tested with suitable NC programs. The **TNCopt** software from HEIDENHAIN features the "Contour-Single.h" NC program for this. The advantage is that the speed and the contour deviations can be seen directly.
- When setting the jerk and acceleration values, as well as selecting the suitable filters, take into account
  - the running noises of the machine
  - the mechanical load (wear)
  - the desired machining speed.

Setting the nominal position value filter as of SW 340 490-01

- Enter the permissible axis-specific jerk:
  - For single filter: MP1230.x (at corners), MP1240.x (at curvature changes, e.g. tangential transition from a line to an arc)
  - For double filter: MP1231.x (at corners), MP1241.x (at curvature changes)
  - For HSC filter: MP1232.x (at corners), MP1242.x (at curvature changes)
  - For advanced HSC filter: MP1233.x (at corners), MP1243.x (at curvature changes)
  - For rapid traverse at corners in MP1250.x: Here a factor of the value from the corresponding MP123x.x is entered. At transitions between rapid traverse motions, the jerk set in MP123x.x multiplied by this factor is active.
- In MP1202.0, define a tolerance for contour transitions with motions at the machining feed rate. This tolerance can be overwritten by the machine user with Cycle 32 "Tolerance."
- In MP1202.1, define a tolerance for contour transitions with motions at rapid traverse. This tolerance can **not** be overwritten by the machine user with Cycle 32 "Tolerance."
- When selecting the limit frequencies for single filters (MP1210), double filters (MP1211), HSC filters (MP1212) and advanced HSC filters (MP1213), take into account the lowest resonant frequency of your machine’s axes and the desired damping at this frequency. If the limit frequency is set to 0, the filter is switched off.

Note

The tolerance (MP1202.x, Cycle 32) always refers to the nominal value, meaning the servo lag also affects the contour accuracy.
For example, if the servo lag S = 5 µm and the tolerance T = 10 µm, then the total deviation is 15 µm.

- With MP1200, select from the single, double, HSC and advanced HSC filters the nominal position value filters for the **Program Run, Single Block, Program Run, Full Sequence** and **Positioning with Manual Data Input** operating modes.
- With MP1201, select the nominal position value filters for the **Manual, Handwheel, Jog Increment** and **Pass Over Reference Point** operating modes. Single and double filters are available here.
- With MP1222 and MP1223 you specify whether the tolerance from MP1202 or Cycle 32 is also considered for curvature changes. MP1222 affects the HSC filter, and MP1223 affects the advanced HSC filter.
- In order to achieve the optimum results for your machine or application, test the various filter settings with a test part consisting of short, straight paths.

Tips for setting the nominal position value filter

HSC filter (MP1212)

The HSC filter can be used to limit the bandwidth of the nominal command value. The corresponding limit frequency is entered in MP1212 for this.

If the HSC filter is used with limit frequencies less than 30 Hz, then ensure that the path jerk (MP1090) in dependency of the limit frequency is not set to too large a value. Otherwise, overshoots at the corners could exceed the prescribed tolerances. Independently of this, MP1090 should always be set in a manner such that the following error does not become too large during the jerk phase.

In order to maintain the accuracy at HSC frequencies under 30 Hz, the following tables should also be considered:

Controller: CC 424(B)	**Frequency to be damped [Hz]**							
MP1212	10	12,5	15	17.5	20	22.5	**25**	27.5
Max. path jerk for tolerance = 5 µm	18	18	18	18	23	45	90	150
Max. path jerk for tolerance = 10 µm	30	36	36	36	45	100	150	–
Max. path jerk for tolerance = 20 µm	45	45	64	68	90	180	–	–

Controller: CC 422	**Frequency to be damped [Hz]**							
MP1212	15	17.5	20	22.5	25	27.5	30	27.5
Max. path jerk for tolerance = 5 µm	45	45	60	68	80	80	80	96

If the HSC filter does not need to be used to limit the bandwidth, very brief machining times can be achieved by setting high limit frequencies with high jerk values. Of course, requirements for this are increased jerk (MP1232.x, MP1242.x, MP1250.x) and acceleration values (MP1060, MP1070). since starting at certain limit frequencies, the jerk and acceleration values increasingly limit the feed rate. The machining time is briefest at a certain value in MP1212. The following rule of thumb is used to find the optimum value:

MP1212 = 23 + MP1232 *0.2 (however, MP1212 should be ≥ 30)

If the tolerance is increased, the optimum value for MP1212 is shifted to smaller frequencies.

List of the nominal position value filters

The settings for the nominal position value filters are listed below by the MP number.

MP1200 **Nominal position value filter**
Input: 0: Single filter
1: Double filter
2: HSC filter
3: Advanced HSC filter

MP1201 **Nominal position value filter in the Manual, Handwheel, Jog Increment, and Pass Over Reference Point operating modes**
Input: 0: Single filter
1: Double filter

MP1202 **Predefined tolerance for Cycle 32**
Input: 0.0000 to 3.0000 [mm]
MP1202.0 Tolerance at corners for movements at machining feed rate
MP1202.1 Tolerance at corners for movements at rapid traverse

MP1205 **Reduction of the starting feed rate**
Input: 0: Not active (fast, but somewhat less precise)
1: Active (slow, but likely more precise)

MP1210 **Limit frequency for single filter**
Input: 0.0 to 166.0 [Hz]

MP1211 **Limit frequency for double filter**
Input: 0.0 to 166.0 [Hz]

MP1212 **Limit frequency for HSC filter**
Input: 0.0 to 166.0 [Hz]

MP1213 **Limit frequency for advanced HSC filter**
Input: 0.0 to 166.0 [Hz]

MP1222 **Tolerance at curvature changes with HSC filter**
Input: (only effective if MP7640 bit 4 = 0)
0: Do not include the tolerance
1: Include the tolerance

MP1223 **Tolerance at curvature changes with advanced HSC-filter**
Input: (only effective if MP7640 bit 4 = 0)
0: Do not include the tolerance
1: Include the tolerance

MP1230.x **Max. permissible axis-specific jerk at corners for single filter**

Input: 0.1 to 1000.0 [m/s^3]

MP1231.x **Max. permissible axis-specific jerk at corners for double filter**

Input: 0.1 to 1000.0 [m/s^3]

MP1232.x **Max. permissible axis-specific jerk at corners for HSC-filter**

Input: 0.1 to 1000.0 [m/s^3]

MP1233.x **Max. permissible axis-specific jerk at corners for advanced HSC filter**

Input: 0.1 to 1000.0 [m/s^3]

MP1240.x **Max. permissible axis-specific jerk at curvature changes for single filter**

Input: 0.1 to 1000.0 [m/s^3]

MP1241.x **Max. permissible axis-specific jerk at curvature changes for double filter**

Input: 0.1 to 1000.0 [m/s^3]

MP1242.x **Max. permissible axis-specific jerk at curvature changes for HSC filter**

Input: 0.1 to 1000.0 [m/s^3]

MP1243.x **Max. permissible axis-specific jerk at curvature changes for advanced HSC filter**

Input: 0.1 to 1000.0 [m/s^3]

MP1250.x **Max. permissible axis-specific jerk at curvature changes for advanced HSC filter**

Input: 0.1 to 1000.0 [m/s^3]

MP7684, described below, should not be used as of software version 340 490-xx. The nominal position value filters should be adjusted via machine parameters 12xx.

MP7684 bit 10 introduced a modified calculation of the contouring feed rate at the beginning of a contour element. The modified calculation improves the surface quality at contour transitions, which can lessen the accuracy at corners. This can be deactivated by setting MP7684 bit 10.

MP7684 **Nominal position value filter (bit 0 to bit 4) and path control with M128 or TCPM (bit 5 to bit 7 permitted)**

Format: %xxxxxxxx

Input: Bit 0 - Nominal position value filter
0: Include acceleration
1: Do not include the acceleration
Bit 1 – Nominal position value filter
0: Include the jerk at corners
1: Do not include the jerk at corners
Bit 2 – Nominal position value filter
0: Include the tolerance
1: Do not include the tolerance
Bit 3 – Nominal position value filter
0: Include the radial acceleration
1: Do not include the radial acceleration
Bit 4 – Nominal position value filter (only in connection with M114; has priority over MP1222 and MP1223)
0: Jerk and tolerance limits at changes in the curvature are included
1: Jerk and tolerance limits at changes in the curvature are not included
Bit 5 – Reserved
Bit 6 – Reserved
Bit 7 – Reserved
Bit 8 – Reserved
Bit 9 – Accelerated 5-axis machining with M128 with many rotary axis motions that are less than 2° per positioning block (not with handwheel superimpositioning with M118)
0: Inactive
1: Active
Bit 10 - Modification of the calculation of the contouring feed rate at the beginning of a contour element
0: Active
1: Previous behavior (through 340 490-02)

Compatibility with 340 422-xx: HEIDENHAIN generally recommends using the MP12xx machine parameters for the nominal position value filters. The following table is intended as an aid in this transition.

Comparison of the machine parameters

MP 340 490-xx	MP 340 422-xx	Function	Possible input value
1200		Selection of the filter type	0 = Single filter 1 = Double filter 2 = HSC filter 3 = Advanced HSC filter
1201	1095.1	Nominal position value filter in manual operation	0 = Single filter 1 = Double filter
 1202.0 1202.1	 1096.0 1096.1	Tolerance for contour transitions at corners With machining feed rate With rapid traverse	 0.0000 to 3.0000 [mm] 0.0000 to 3.0000 [mm]
1210	1099.0	Limit frequency for single filter	0.0 to 166.0 [Hz]
1211	1099.1	Limit frequency for double filter	0.0 to 166.0 [Hz]
1212	1094	Limit frequency for HSC filter	0.0 to 166.0 [Hz]
1213		Limit frequency for advanced HSC filter	0.0 to 166.0 [Hz]
1222		HSC filter: Tolerance for curvature transitions	0 = Do not include the tolerance 1 = Include the tolerance
1223		Advanced HSC filter: Tolerance for curvature transitions	0 = Do not include the tolerance 1 = Include the tolerance
1230.x	1097.x	Max. permissible axis-specific jerk for single filter	0.1 to 1000.0 [m/s^3]
1231.x	1098.x	Max. permissible axis-specific jerk for double filter	0.1 to 1000.0 [m/s^3]
1232.x	1098.x	Max. permissible axis-specific jerk for HSC filter	0.1 to 1000.0 [m/s^3]
1233.x		Max. permissible axis-specific jerk for advanced HSC filter	0.1 to 1000.0 [m/s^3]
1240.x	1097.x	Max. permissible axis-specific jerk at curvature transitions for single filter	0.1 to 1000.0 [m/s^3]
1241.x	1098.x	Max. permissible axis-specific jerk at curvature transitions for double filter	0.1 to 1000.0 [m/s^3]
1242.x	1097.x	Max. permissible axis-specific jerk at curvature transitions for HSC filter	0.1 to 1000.0 [m/s^3]

MP 340 490-xx	MP 340 422-xx	Function	Possible input value
1243.x		Max. permissible axis-specific jerk at curvature transitions for advanced HSC filter	0.1 to 1000.0 [m/s^3]
1250.x		Factor for axis-specific jerk at corners at rapid traverse (from value in MP123x.x)	0.0000 to 30.0000
1262		Filter order used for HSC filters (only useful if LIFTOFF is used during powerfail with CC 424(B))	15 to 31
1263		Filter order used for advanced HSC filters (only useful if LIFTOFF is used during powerfail with CC 424(B))	15 to 31

Matching of the filter order to limit frequencies

In software versions up to 340 422-xx, the limit frequencies of single and double filters were set via the filter order. The following table gives an overview of how the filter orders match the limit frequencies.

Note

Please note that the values of the following conversion table were calculated for use of a CC 422 controller unit with a position controller cycle time of 1.8 ms (MP7600 = 3; recommended value). If your setting for MP7600 deviates from the recommended value, please contact HEIDENHAIN or re-adjust the nominal position value filter.
The values in the table are only to be used as guidelines. The path behavior must always be checked.

Filter order	**Single filter**		**Double filter**	
	CC 422 (MP7600.0 = 3)	**CC 424(B)**	**CC 422 (MP7600.0 = 3)**	**CC 424(B)**
	Hz	**Hz**	**Hz**	**Hz**
1	Over 136	Over 82	Over 98	Over 59
2	88 - 135	53 - 81	63 - 97	38 - 58
3	65 - 87	39 - 52	47 - 62	28 - 37
4	51 - 64	31 - 38	37 - 46	23 - 27
5	42 - 50	26 - 30	31 - 36	19 - 22
6	36 - 42	22 - 25	26 - 30	16 - 18
7	32 - 35	19 - 21	23 - 25	14 - 15
8	28 - 31	17 - 18	21 - 22	13
9	25 - 27	15 - 16	18 - 20	11
10	23 - 24	14	17	10
11	21 - 22	13	15 - 16	9
12	20	12	14	9
13	18 - 19	11	13	8
14	17	10	12	8
15	16	10	12	7
16	15	9	11	7
17	14	9	10	6
18	14	8	10	6
19	13	8	9	6
20	12	8	9	6
21	12	7	9	5

Setting the nominal position value filter as of SW 340 422-12

> Note
>
> As of NC software 340 490-xx, t the nominal position value filters were restructured, expanded and set to MP12xx. The following description of the nominal position value filters only applies if you are using NC software 340 422-xx, or if the entry **`MPMODE=340422`** was saved in OEM.SYS.

To attain a high machining velocity, the workpiece contour can be adapted to the machine dynamics by means of a nominal position value filter. Here the iTNC always complies with the tolerance (MP1096.x, Cycle 32), the axis-specific jerk (MP1097.x, MP1098.x), the acceleration (MP1060.x) and the radial acceleration (MP1070.x).

The iTNC calculates the filter parameters automatically. For test purposes, you can deactivate some of the parameters in MP7684 bits 0 to 4 for the calculation.

▶ Enter the permissible axis-specific jerk:
- For single filter: MP1097.x (at corners)
- For double filter: MP1098.x (at corners)
- For HSC filter: MP1098.x (at corners), MP1097.x (at curvature changes, e.g. tangential transition from a line to an arc)

▶ In MP1096.0, define a tolerance for contour transitions with motions at the machining feed rate. This tolerance can be overwritten by the machine user with Cycle 32 "Tolerance."

▶ In MP1096.1, define a tolerance for contour transitions with motions at rapid traverse. This tolerance can **not** be overwritten by the machine user with Cycle 32 "Tolerance."

▶ Select from the following tables the input values for MP1099.x or MP1094. Note the lowest resonance frequency of your machine axes and the desired damping at this frequency.

> Note
>
> The tolerance (MP1096.x, Cycle 32) always refers to the nominal value, meaning the servo lag also affects the contour accuracy.
> For example, if the servo lag S = 5 µm and the tolerance T = 10 µm, then the total deviation is 15 µm.

Note

In order to achieve the same behavior with the single and double filters as with the TNC 426/TNC 430, you must convert the values for the minimal filter order (MP1099.x):

$$FO_{iTNC\ 530} = \frac{(FO_{TNC\ 426/TNC\ 430} + 1) \cdot t_{TNC\ 426/TNC\ 430}}{t_{iTNC\ 530}} - 1$$

$FO_{TNC\ 426/TNC\ 430}$:	Minimal filter order TNC 426/TNC 430 (MP1099.x)
$FO_{iTNC\ 530}$:	Minimal filter order iTNC 530 (MP1099.x)
$t_{TNC\ 426/TNC\ 430}$:	Position controller cycle time TNC 426/TNC 430
$t_{iTNC\ 530}$:	Position controller cycle time iTNC 530

Single filter (MP1099.0)
with 1.8 ms position controller cycle time

Damping [dB]		Frequency to be damped [Hz]									
	5	**7.5**	**10**	**12.5**	**15**	**20**	**25**	**30**	**35**	**40**	**50**	**60**
3	–	–	17	14	11	8	6	5	5	4	3	3
6	–	–	–	19	16	12	9	7	6	5	4	4
9	–	–	–	–	19	14	11	9	8	7	5	4
12	–	–	–	–	–	16	13	10	9	8	6	5
15	–	–	–	–	–	18	14	11	10	8	7	6

Single filter (MP1099.0)
with 3 ms position controller cycle time

Damping [dB]		Frequency to be damped [Hz]										
	5	**7.5**	**10**	**12.5**	**15**	**20**	**25**	**30**	**35**	**40**	**50**	**60**
3	21	14	10	8	7	5	4	3	3	2	2	1
6	–	19	14	11	9	7	5	4	4	3	3	2
9	–	–	17	14	11	8	7	5	5	4	3	3
12	–	–	19	15	13	10	8	6	5	5	4	3
15	–	–	–	17	14	10	8	7	6	5	4	3

Double filter (MP1099.1)
with 1.8 ms position controller cycle time

Damping [dB]		Frequency to be damped [Hz]										
	5	**7.5**	**10**	**12.5**	**15**	**20**	**25**	**30**	**35**	**40**	**50**	**60**
3	–	16	12	9	8	6	5	4	3	3	2	2
6	–	–	17	13	11	8	7	5	5	4	3	3
9	–	–	21	16	14	10	8	7	6	5	4	3
12	–	–	–	19	16	12	9	8	7	6	4	4
15	–	–	–	21	18	13	10	9	7	6	5	4

Double filter (MP1099.1)
with 3 ms position controller cycle time

Damping [dB]		Frequency to be damped [Hz]										
	5	**7.5**	**10**	**12.5**	**15**	**20**	**25**	**30**	**35**	**40**	**50**	**60**
3	14	10	7	6	5	3	3	2	2	2	1	1
6	21	13	10	8	7	5	4	3	3	2	2	1
9	–	16	12	10	8	6	5	4	3	3	2	2
12	–	19	14	11	9	7	5	4	4	3	3	2
15	–	21	16	12	10	8	6	5	4	4	3	2

HSC filter (MP1094)
with 1.8 ms position controller cycle time

Damping [dB]		Frequency to be damped [Hz]									
	10	**12.5**	**15**	**17.5**	**20**	**25**	**30**	**35**	**40**	**45**	**50**	**60**
3	11	12	15	18	24	28	36	41	46	51	56	66
6	–	11	12	14	18	25	29	35	40	45	50	60
9	–	–	11	12	16	22	27	32	36	41	46	56
12	–	–	–	11	14	20	24	27	30	38	42	52
15	–	–	–	–	12	19	23	25	28	35	40	50

HSC filter (MP1094)
with 3 ms position controller cycle time

Damping [dB]		Frequency to be damped [Hz]										
	10	**12.5**	**15**	**17.5**	**20**	**25**	**30**	**35**	**40**	**45**	**50**	**60**
3	10	15	18	21	23	28	33	38	43	48	53	62
6	8	11	15	17.5	20	25	30	35	40	45	50	60
9	–	10	13	16	17	22	27	32	37	42	47	57
12	–	9	12	14	16	20	25	30	35	40	45	50
15	–	8	11	13	15	19	24	29	34	39	44	49

▶ With MP1095.x you select the single or double filter. The HSC filter is switched on with MP1094.
MP1095.1 is effective in the Manual, Handwheel, Incremental Jog Positioning and Reference Mark Traverse modes. MP1095.0 and MP1094 are effective in the **Program Run, Single Block, Program Run, Full Sequence** and **MDI** operating modes. If MP1094 is used, MP1095.0 is without effect.
Example:
Set the double filter in the Program Run modes for a smooth traverse (MP1095.0 = 1), or set the single filter in the Manual mode for a shorter braking distance (MP1095.1 = 0).

▶ Test the three filter settings using a test part made of short line segments.

- Single filter
- Double filter
- HSC filters

Note

If you have selected the best nominal position value filter for your application, please note that your input value from MP1096.0 can be overwritten by the machine user through Cycle 32. If you have switched off the nominal position value filter (MP1096.x = 0), the machine user can also switch it on using Cycle 32.

MP1096.x is evaluated for each program selection, since the tolerance can be overwritten by Cycle 32. The overwriting of MP1096.x by an MP subfile is therefore without effect if a tolerance was programmed via Cycle 32 in an NC program.

The nominal position value filters function in all operating modes (even in rapid traverse). For RIGID TAPPING (Cycle 17), the nominal position value filter is automatically switched off.

Machine parameters	Single filter	Double filter	HSC filters
HSC filter MP1094	MP1094 = 0	MP1094 = 0	MP1094 = Cutoff frequency
Single/double filter MP1095.x	MP1095.x = 0	MP1095.x = 1	MP1095.0 = Nonfunctional MP1095.1 = 0 or 1
Tolerance for contour transitions MP1096.x	MP1096.x = Tolerance (Cycle 32)		
Axis-specific jerk for single filter MP1097.x	MP1097.x = Jerk (at corners)	MP1097.x = Nonfunctional	MP1097.x = Jerk (at curvature changes)
Axis-specific jerk for double filter MP1098.x	MP1098.x = Nonfunctional	MP1098.x = Jerk (at corners)	MP1098.x = Jerk (at corners)
Minimum filter configuration MP1099.x	MP1099.0 = Filter order	MP1099.1 = Filter order	MP1099.x = Nonfunctional

MP1094 **HSC filters**
Input: 0: HSC filter inactive
0.1 to 166.0: Cutoff frequency for HSC filter

MP1095 **Nominal position value filter**
Input: 0: Single filter
1: Double filter
MP1095.0 In the Program Run Full Sequence, Program Run Single Block, and Positioning With Manual Data Input operating modes
MP1095.1 In the Manual, Handwheel, Jog Increment and Pass Over Reference Point operating modes

MP1096 **Tolerance for contour transitions at corners**
Input: 0: No nominal position value filter
0.001 to 3.000 [mm]
MP1096.0 With machining feed rate
MP1096.1 With rapid traverse

MP1097.x **Maximum permissible axis-specific jerk (single/HSC filter)**
Input: 0.1 to 1000.0 [m/s^3 or 1000°/s^3]

MP1098.x **Maximum permissible axis-specific jerk (double/HSC filter)**
Input: 0.1 to 1000.0 [m/s^3 or 1000°/s^3]

MP1099 **Minimum filter order**
Input: 0 to 20
MP1099.0 Minimum filter configuration for single filter (MP1095 = 0)
MP1099.1 Minimum filter configuration for double filter (MP1095 = 1)

MP7684 **Nominal position value filter (bit 0 to bit 4) and path control with M128 or TCPM (bit 5 to bit 7 permitted)**
Format: %xxxxxxxx
Input: Bit 0 - Nominal position value filter
0: Include acceleration
1: Do not include the acceleration
Bit 1 – Nominal position value filter
0: Include the jerk
1: Do not include the jerk
Bit 2 – Nominal position value filter
0: Include the tolerance
1: Do not include the tolerance
Bit 3 – Nominal position value filter
0: Include the radial acceleration
1: Do not include the radial acceleration
Bit 4 – Nominal position value filter (only in connection with M114; has priority over MP1222 and MP1223)
0: Jerk and tolerance limits at changes in the curvature are included
1: Jerk and tolerance limits at changes in the curvature are not included

Feed-rate smoothing (do not use any longer)

Fluctuations in feed rate sometime occur during execution of NC programs consisting of short straight-line segments. MP7620 bit 6 smoothes the feed rate. However, it also reduces it somewhat.

MP7620	**Feed-rate override and spindle speed override**
Input:	Bit 6 – Feed-rate smoothing 0: Not active 1: Active

Tolerance consideration with M128 or TCPM

When working with M128 or TCPM, the rotary axes necessitate compensating movements in the main axes. Since rotary axes mostly do not have the same dynamics that linear axes have, machining is slower when rotary axes are involved.

During program run with M128 or TCPM, the head dimensions are also included in the tolerance consideration (MP1202.x, Cycle 32).

A higher tolerance is usually permissible for rotary axes, since this does not affect the accuracy (when using a spherical cutter). This higher tolerance leads to shorter positioning times of the rotary axes and the better surfaces:

- Under **TA** in Cycle 32, program a separate tolerance for rotary axes.

Because of this tolerance, a separate nominal position value filter is used for rotary axes.

6.8.3 Position controller

Position controller cycle time

With MP7600.0 you can set the position controller cycle time.

- In MP7600.0, enter a factor which, when multiplied by 0.6 ms, results in the position controller cycle time.

With the input value MP7600.0 = 3, the iTNC has a minimum position controller cycle time of 1.8 ms. The increase of the position controller cycle time also increases the PLC cycle time. To return to the previous PLC cycle time, enter the corresponding factor in MP7600.1. For entries which lead to a PLC cycle time < 10 ms, the PLC cycle time is limited to 10 ms.

MP7600.0	**Only CC 422: Position controller cycle time = MP7600.0 · 0.6 ms**
Input:	1 to 20 Proposed input value: 3 (= 1.8 ms) Proposed input value for basic version: 6 (= 3.6 ms)
MP7600.1	**Only CC 422: PLC cycle time = Position controller cycle time · MP7600.1**
Input:	1 to 20 Proposed input value: 6 (= 10.8 ms) Proposed input value for basic version: 3 (= 10.8 ms)

You can choose between two types of feedback control:

- Feedback control with following error (servo lag)
- Feedback control with velocity feedforward

- Select the type of control in the **`Positioning with manual data input`**, **`Program run, single block`** and **`Program run, full sequence`** operating modes with MP1392.
- Select the type of control in the **`Manual`** and **`Handwheel`** modes of operation with MP1391.

> Note
>
> The machine must always be adjusted for both types of control.

MP1392	**Velocity feedforward in the operating modes Program Run Single Block, Program Run Full Sequence and Positioning with Manual Data Input**
Format:	%xxxxxxxxxxxxxx
Input:	Bits 0 to 13 represent axes 1 to 14 0: Operation with following error (lag) 1: Operation with velocity feedforward control

> Note
>
> M90 (lag mode: Constant contouring speed at corners) is effective only if operation with following error is selected for all axes (MP1392 = %00000000000000).

MP1391.0 **Velocity feedforward control in the MANUAL and HANDWHEEL operating modes**

Format: %xxxxxxxxxxxxxx

Input: Bits 0 to 13 represent axes 1 to 14
0: Inactive
1: Active

MP1391.1 **Acceleration feedforward control in the MANUAL and HANDWHEEL operating modes**

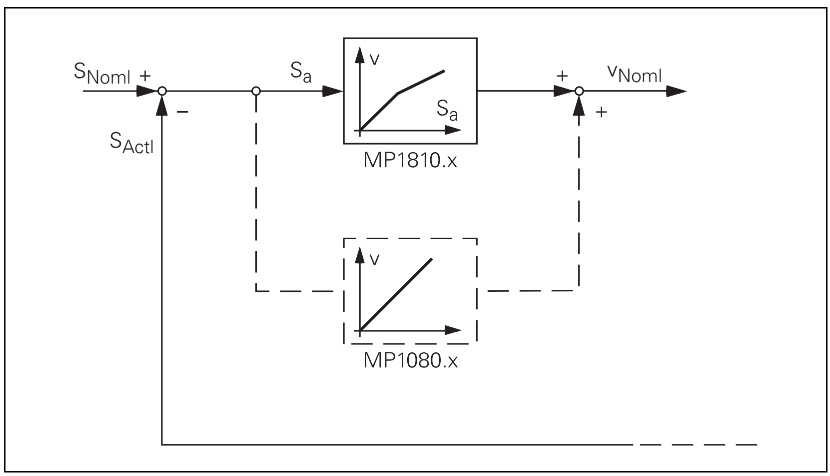
Format: %xxxxxxxxxxxxxx

Input: Bits 0 to 13 represent axes 1 to 14
0: Inactive
1: Active

Feedback control with following error

Following error (also known as servo lag) is a gap that remains between the nominal position commanded by the NC and the actual position.

Simplified representation:



The nominal position value s_{noml} for a given axis is compared with the actual position value s_{actl} and the resulting difference is the following error s_a:

s_a = $s_{noml} - s_{actl}$
s_a = following error
s_{noml} = nominal position value
s_{actl} = actual position value

The following error is multiplied by the k_v factor and passed on as nominal velocity value:

$v_{noml} = k_v \cdot s_a$

v_{noml} = nominal velocity value

Analog axes:
For stationary axes, the integral factor has an additional effect (MP1080.x). It produces an offset adjustment.

Digital axes:
There is no offset. MP1080.x has no function.

k_v factor during control with following error

The control loop gain, known as the k_v factor, defines the amplification of the position control loop. You must find the optimum k_v factor by trial and error.

If you choose a k_v factor that is too large, the following error will become very small. However, this can lead to oscillations.

If you choose too small a k_v factor, the axis will move to a new position too slowly.

For axes that are interpolated with each other, the k_v factors must be equal to prevent contour deviations.

- In MP1810.x define a set of k_v factors for operation with following error.

You can selectively increase the contour accuracy with a higher k_v factor. This k_v factor is activated with the M function M105:

- In MP1815.x define a second set of k_v factors and activate them with M105.

M105 also influences compensation of reversal spikes during circular motion. With M106 you can switch back to the original set of k_v factors:

- Enable the M functions M105/M106 with MP7440, bit 3.

Interrelation of k_v factor, feed rate, and following error

The following formula shows the interrelation of k_v factor, feed rate, and following error:

$$k_v = \frac{v_e}{s_a} \quad \text{or} \quad s_a = \frac{v_e}{k_v}$$

k_v = loop gain [(m/min)/mm]
v_e = rapid traverse [m/min]
s_a = following error [mm]

MP1810.x **k_v factor for control with following error**
Input: 0.100 to 20.000 [(m/min)/mm]

MP1815.x **k_v factor for control with following error effective after M105**
Input: 0.100 to 20.000 [(m/min)/mm]

MP7440 **Output of M functions**
Format: %xxxxx
Input: Bit 3 – Switching the k_v factors with M105/M106
0: Function is not in effect
1: Function is effective

Feedback control with velocity feedforward

The nominal velocity value consists of an open-loop and a closed-loop component.

With velocity feedforward control, the machine-adjusted nominal velocity value is the open-loop controlled component. The closed-loop velocity component is calculated through the following error. The following error is small.

In most cases, machines are controlled with velocity feedforward, since it makes it possible to machine exact contours even at high speeds.

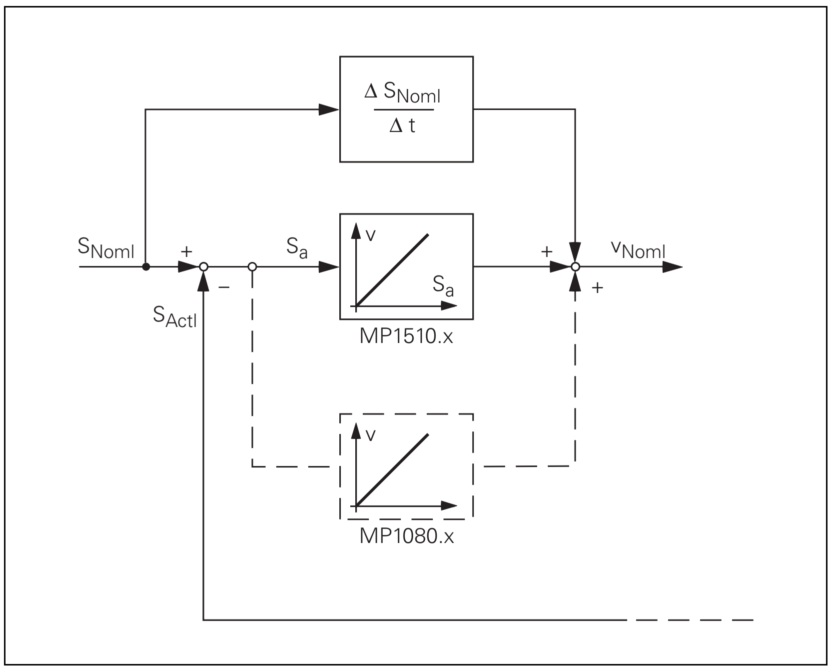
For the **Positioning with manual data input, Program run, single block** and **Program run, full sequence** operating modes:

▶ Switch on the velocity feedforward control with MP1392.

For the **Manual** and **El. Handwheel** operating modes:

▶ Switch on the velocity feedforward control with MP1391.

Block Diagram:

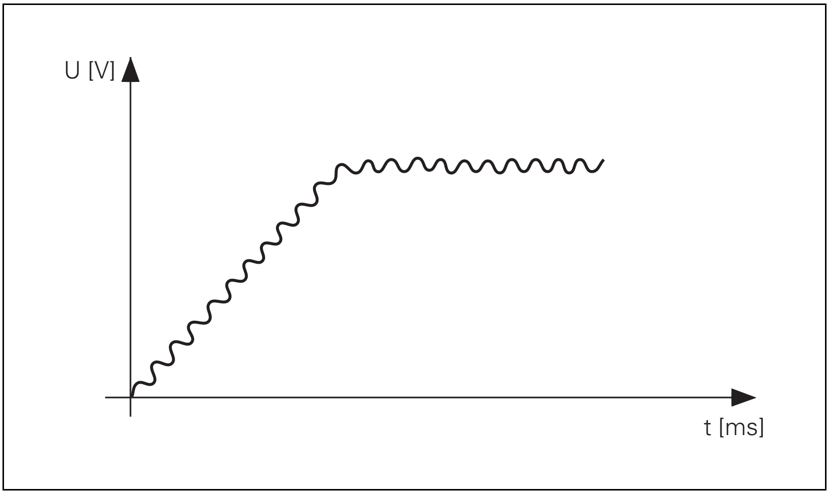


Analog axes:
For stationary axes, the integral factor has an additional effect (MP1080.x). It produces an offset adjustment.

Digital axes:
There is no offset. MP1080.x has no function.

You can influence feedback control with velocity feedforward with the k_v factor:

▶ In MP1510.x, enter a k_v factor.



Attention

If the k_v factor that you select is too large, the system will oscillate around the forward-fed nominal velocity value.

Unlike operation with following error, you must also enter the optimum k_v factor for each axis when operating with interpolated axes.

You can selectively increase the contour accuracy with a higher k_v factor. This k_v factor is activated with M105:

- In MP1515.x, define a second set of k_v factors and activate them with M105.

M105 also influences compensation of reversal spikes during circular motion. With M106 you can switch back to the original set of k_v factors:

- Enable the M functions M105/M106 with MP7440, bit 3.

MP1510.x	**k_v factor for velocity feedforward control**
Input:	0.100 to 1000.000 [(m/min)/mm]
MP1515.x	**k_v factor for velocity feedforward control effective after M105**
Input:	0.100 to 1000.000 [(m/min)/mm]
MP7440	**Output of M functions**
Format:	%xxxxx
Input:	Bit 3 – Switching the k_v factors with M105/M106: 0: Function is not in effect 1: Function is effective

Feedback control with velocity semifeedforward

MP1396.x allows the operator to switch to velocity semifeedforward control. Normally, work will be carried out using velocity feedforward. Velocity semifeedforward is activated, for example, by an OEM cycle before roughing, in order to permit a higher following error and thereby a higher velocity, combined with a lowered accuracy, in order to traverse corners.
Before finishing, another OEM cycle can be used to switch back to velocity feedforward, in order to finish with the highest accuracy possible.

In order to use velocity semifeedforward, a factor must be entered for every axis in MP1396.x, where values toward 0 control the following error more, and values toward 1 control the velocity feedforward more. The factor can be overwritten with FN17: SYSWRITE ID600. However, the factor from MP1396.x becomes valid again at the end of a program (MP7300 = 1) and whenever a new NC program is selected.

As soon as a factor between 0.001 and 0.999 has been entered in MP1396.x, the k_V factor from MP1516.x becomes effective.

Note

For axes that are interpolated with each other, the k_V factors must be equal. In this case the smaller k_V factor determines the input value for these axes.

The values for position monitoring are interpolated according to the factor in MP1396.x between the values for servo lag (MP1710.x, MP1720.x) and the values for velocity feedforward control (MP1410.x, MP1420.x).

Feedback control with following error (servo lag)	Feedback control with velocity semifeedforward	Feedback control with velocity feedforward
MP1391 bit x = 0 MP1392 bit x = 0 MP1396.x = nonfunctional	MP1391 bit x = 1 MP1392 bit x = 1 MP1396.x = 0.001 ←→ MP1396.x = 0.999	MP1391 bit x = 1 MP1392 bit x = 1 MP1396.x = 1

If the PC Marker M4625 is set, the start of an NC axis for which operation with following error or velocity semifeedforward control is active is prevented, and the error message **Semifeedforward not permitted** is displayed. The marker is only evaluated when an NC program is started or when the axes are traversed with axis-direction keys.

To use feedback control with velocity semifeedforward:

- Activate the velocity feedforward control with MP1391 and/or MP1392.
- Determine the k_V factor for velocity feedforward control (MP1510.x).
- Activate the velocity semifeedforward control by entering the desired factor in MP1396.x.
- Determine the k_V factor for velocity semifeedforward control (MP1516.x).
- Enter MP1396.x = 1 to return to velocity feedforward control.
- For example, you may now use FN17: SYSWRITE ID 600 in an OEM cycle to overwrite or reestablish the factor in MP1396.x.

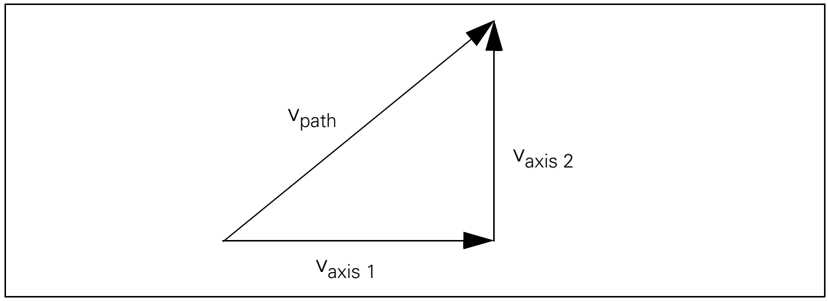
MP1396.x **Feedback control with velocity semifeedforward**
Input: 0.001 to 0.999
1: Velocity feedforward control

MP1516.x **k_V Factor for velocity semifeedforward**
Input: 0.100 to 20.000 [(m/min)/mm]

		Set	Reset
M4625	**Disabling of NC axes when velocity semifeedforward control is active**	PLC	PLC

Rapid traverse and feed rate limitation

If more than one axis is moved simultaneously, the rapid traverse on the path v_{path} is formed from the appropriate axis components. The same also applies to the path acceleration (See "Interpolator" on page 817).



If there are problems with the inverter, e.g. because the energy being generated from the axes in motion cannot be dissipated, you can limit the rapid traverse on the path:

- Enter the maximum rapid traverse on the path in MP1011.
- Enter the inputs axis-specifically in MP1010.x.
- With MP1012.x you can enter a second axis-specific rapid traverse as an alternative to MP1010.x. Use the **`FN17: SYSWRITE ID 20 NR 18 = <value>`** function to switch between the two machine parameters. MP1012 can only be used in the **`Program Run, Single Block, Program Run, Full Sequence`** and **`MDI`** operating modes.
  `<value>` = 0: MP1010 and MP1011 active
  `<value>` = 1: MP1012 active, MP1011 not active, meaning that the possible increase in the contouring feed rate is not limited.
  The rapid traverse from MP1010 in connection with MP1011 is always active at program end and when a program is first selected.

The feed rate can be limited via the PLC:

- Enter the reduced maximum contouring feed rate in D596.
- Limit the feed rate axis-specifically with Module 9140. It can be read with Module 9141.

If the value in D596 is greater than MP1011, then MP1011 becomes effective. After the control is switched on, or after an interruption of the PLC run, D596 is preassigned with the value 300 000 so that MP1011 becomes effective.

Rapid traverse can be limited by the user with the F MAX soft key. This limitation is not effective if M4587 is set. In this case only limitation through D596 is effective. After M4587 is reset, both D596 and the last limit set via the F MAX soft key are effective again. Limitation of rapid traverse via the F MAX soft key can be rescinded with the RESET soft key. The value from MP1011 then becomes active again.

The feed rate is significantly lower for manual operation than for rapid traverse:

- Enter in MP1020 a feed rate for manual operation.

The programmed feed rate and the current path feed rate are saved in D360 and D388 in mm/min. In the manual operating modes, the highest axis feed of all axes is stored in D388.

The maximum possible feed rate depends on the encoder being used.

v_{max} [mm/min] = P [mm] · f_i [kHz] · 60

v_{max} = Maximum traverse speed
P = Signal period of the encoder
f_i = Input frequency of the encoder input,
See "Encoder signals" on page 650 and
See "Encoder Connections" on page 375.

Digital axes:
For digital axes, the maximum feed rate also depends on the number of pole pairs in the drive motor and the pitch of the ball screw.

$$v_{max}\,[\text{mm/min}] = \frac{60\,000}{\text{Number of pole pairs}}\,[1/\text{min}] \cdot \text{ball screw pitch [mm]} \cdot \frac{f_{PWM}}{5000}$$

Analog axes:

- In MP1050.x, enter the desired analog voltage for rapid traverse.
- Adjust the rapid traverse feed rate (v_{max}) with the analog voltage at the servo amplifier.

Module 9140 Set axis-specific feed-rate limit
With Module 9140 you can set axis-specific feed-rate limits. The limits are saved in sequential double words for each axis. The address of the starting double word must be given. Along with a feed-rate value (≥ 0), the following limitations are possible in the double words:

- –1: Maximum feed rate (a previous limitation via Module 9140 is rescinded)
- –2: Axis-specific rapid traverse from MP1010.x (for multi-axis interpolation, the contouring feed rate is limited to the slowest respective value of the participating axes)
- –3: Axis-specific manual feed rate from MP1020.x (for multi-axis interpolation, the contouring feed rate is limited to the slowest respective value of the participating axes)

Invalid feed rates are set to 0, and M4203 is set. Other axes retain their limits.

Call:
PS B/W/D/K <>Start double word>
PS B/W/D/K <>Number of axes or double words>
CM 9140

Error recognition:

Marker	Value	Meaning
M4203	0	Feed-rate limitation set
	1	Error code in W1022
W1022	1	Invalid feed-rate value (< –3)
	2	Invalid number of axes or double words
	3	Invalid block length as of starting address
	5	Not a double-word address
	24	Module was called in a spawn or submit job

Module 9141 Read axis-specific feed-rate limit
With Module 9141 you can read axis-specific feed-rate limits. The limits are saved in sequential double words for each axis. Along with feed-rate values (≥ 0), the limitations –1, –2 or –3 (see Module 9140) are also transferred.

Call:
PS B/W/D/K <>Start double word>
PS B/W/D/K <>Number of axes or double words>
CM 9140

Error recognition:

Marker	Value	Meaning
M4203	0	Feed-rate limitation set
	1	Error code in W1022
W1022	2	Invalid number of axes or double words
	3	Invalid block length as of starting address
	5	Not a double-word address
	24	Module was called in a spawn or submit job

MP1010.x	**Rapid traverse**
Input:	10 to 1 000 000 [mm/min or °/min]
MP1011	**Limit of rapid traverse on the path**
Input:	10 to 1 000 000 [mm/min or °/min]
MP1012.x	**Second axis-specific rapid traverse**
Input:	10 to 1 000 000 [mm/min or °/min]
MP1020.x	**Manual feed**
Input:	10 to 300 000 [mm/min or °/min]
MP1050.x	**Analog axes: Analog voltage at rapid traverse**
Input:	1.000 to 9.000 [V] Digital axes: without function Input: 1

		Set	Reset
M4587	**Rescind feed rate limit above F MAX**	PLC	PLC
D596	**Max. feed rate from PLC [mm/min]**	NC/PLC	PLC
D360	**Programmed feed rate**	NC	NC
D388	**Current contouring feed rate [mm/min]**	NC	NC

Position loop resolution

The encoder signals are interpolated 1024-fold.

$$\text{Position loop resolution [µm]} = \frac{\text{Signal period [µm]}}{1024}$$

Analog axes

The iTNC outputs a voltage per position error. The 10-V analog voltage is subdivided 65536-fold with a 16-bit D/A converter. This results in a smallest voltage step of 0.15 mV.

Rapid traverse (MP1010.x) is attained at a certain voltage (MP1050.x). This results in the voltage ΔU per position error or following error s_a:

$$\Delta U = \frac{\text{MP1050.x [mV]}}{S_a\text{[µm]}}$$

If ΔU is divided by the smallest possible voltage step (0.15 mV), the result is the number *n* of the possible voltage steps per position error.

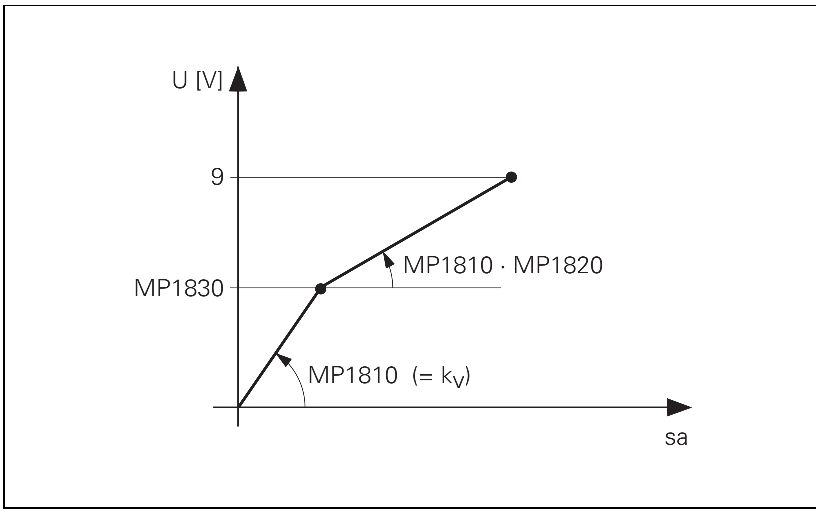
Characteristic curve kink point (for control with following error)

For machines with high rapid traverse, you can not increase the k_v factor enough for an optimum control response to result over the entire velocity range (from standstill to rapid traverse).

In this case you can define a characteristic curve kink point, which has the following advantages:

- High k_v factor in the low range
- Low k_v factor in the upper range (beyond the machining velocity range)

- ▶ Define the position of the characteristic kink with MP1830.x. In the upper range, the k_v factor is multiplied by the factor from MP1820.x.
- ▶ Enter a multiplier in MP1820.



The characteristic curve kink point must lie above the tool feed rate!

Calculating the lag (following error):

$$s_a = \left(\frac{\text{MP1830.x [\%]}}{\text{100 [\%]}} + \frac{\text{100 [\%] – MP1830.x [\%]}}{\text{MP1820.x} \cdot \text{100 [\%]}}\right) \cdot \frac{v_e}{k_v}$$

MP1820.x Multiplier for the k_v factor
Input: 0.001 to 1.00000

MP1830.x Characteristic curve kink point
Input: 0.000 to 100.000 [%]

Opening a position control loop

If M4581 has been set, the control opens the loops of all axes and of the spindle, and then performs an NC stop. This makes it possible, for example, to open the position control loops and at the same time switch off the drives. As with an external emergency stop, position monitoring is shut off for the time defined in MP1150.1, and an actual-to-nominal value transfer is executed. After the time defined in MP1150.1 has expired, position monitoring is again active, for at least the time defined in MP1150.2.

If at a standstill or with an external stop, the PLC sets M4581 in the Program Run Full Sequence, Program Run Single Block, or Positioning with MDI operating modes, and then the axes are moved, the **Approach position** function will be activated when the PLC resets M4581.
If MP4020 bit 8 = 1, then, if the axes are moved after an emergency stop, "Approach position" is automatically activated.

If M4580 has been set, an external EMERGENCY STOP (X42, pin 4 "control-is-ready") **is not** reported to the NC, but rather the function is executed like M4581.

If the position control loop is opened, the axis release in W1024 is canceled.

		Set	Reset
M4580	**Suppress EMERGENCY STOP, open all position control loops, NC stop**	PLC	PLC
M4581	**Open all position control loops, NC stop, activate "Approach position"**	PLC	PLC
W1024	**Axis enabling** Bits 0 to 13 represent axes 1 to 14 0: Position control loop open 1: Position control loop closed	NC	NC

MP1150.1 **Time period for which the monitoring function is to remain off after the fast PLC input defined in MP4130.0 is set**
Input: 0 to 65.535 [s]
Recommended: 0.2 to 0.5

MP4020 **PLC functions**
Input: Bit 8 – Behavior after an external emergency stop
0: "Approach position" is not automatically activated
1: "Approach position" is automatically activated

Clamping the axes

After running an NC block you can clamp the axes.

- If necessary, define in MP7494 the axes for which an exact stop is to occur after positioning.
- Wait until "axis in position" is set in W1026.
- Clamp the axis or axes.
- Open the position control loop with W1040.
- With Module 9161, switch the drive off.

A waiting period is necessary between "axis clamping" and "position control loop opening."

- In W1038, set the bit for the corresponding axis.

The next NC block is not run until the positioning window has been reached and the position control loop has been opened with W1040.

If the position control loop is opened, the axis release in W1024 is canceled. You can link switching off the drives via Module 9161 with W1024.

If a clamped axis is to be repositioned, the NC cancels the "axis in position" message in W1026:

- With Module 9161, switch the drive on.
- Release the clamping.
- Close the position control loop with W1040.

MP7494 **Axes for which an exact stop is to occur after positioning**
Format: %xxxxxxxxxxxxxx
Input: 0: No exact stop
1: Exact stop

		Set	Reset
W1038	**Prepare opening of the position control loop** Bits 0 to 13 represent axes 1 to 14 0: Not active 1: Active	PLC	PLC
W1040	**Axis-specific opening of the position control loop** Bits 0 to 13 represent axes 1 to 14 0: Do not open the position control loop 1: Open the position control loop	PLC	PLC

Feed-rate enable

To move the axes, you must first enable the feed rate through the PLC. Until "feed-rate enable" is set, the nominal velocity value zero is output. "F" is highlighted in the status display.

Feed-rate enable for all axes:

- Set M4563.

Axis-specific feed-rate enable:

- Reset M4563.
- In W1060, set the corresponding bits.

		Set	Reset
M4563	**Feed-rate enable for all axes**	PLC	PLC
W1060	**Axis-specific feed-rate enable** Bits 0 to 13 represent axes 1 to 14 0: No feed-rate enable 1: Feed-rate enable	PLC	PLC

Actual-to-nominal value transfer

During actual-to-nominal value transfer, the current position is saved as the nominal position value. This becomes necessary, for example, if the axis has been moved when the position control loop is open.

There are two ways to turn the actual position into the nominal position:

- To transfer the actual position value in the **Manual** and **Electronic Handwheel** modes, set the corresponding bit in W1044.
- To transfer the actual position in all operating modes, use Module 9145.

Attention

Ensure that actual-to-nominal value transfer does not occur continually, since the position monitoring will not be able to detect any uncontrolled machine movements. In this case no safe machine operation would be possible.

		Set	Reset
W1044	**Actual-to-nominal value transfer** Bits 0 to 13 represent axes 1 to 14 0: No actual-to-nominal value transfer 1: Actual-to-nominal value transfer	PLC	PLC

Module 9145 Actual-to-nominal value transfer

An actual-to-nominal value transfer for NC axes, or for PLC and NC axes together, is possible only if the control is not active (M4176 = 0) or if there is an M/S/T/T2/G strobe.
The module can always be called for an actual-to-nominal transfer only for PLC axes.
For a transfer via M strobe, MP7440 bit 2 must not be set. For a transfer via S/G strobe, MP3030 must not be set.

Call:
PS B/W/D/K <>Axes bit-encoded>
CM 9145

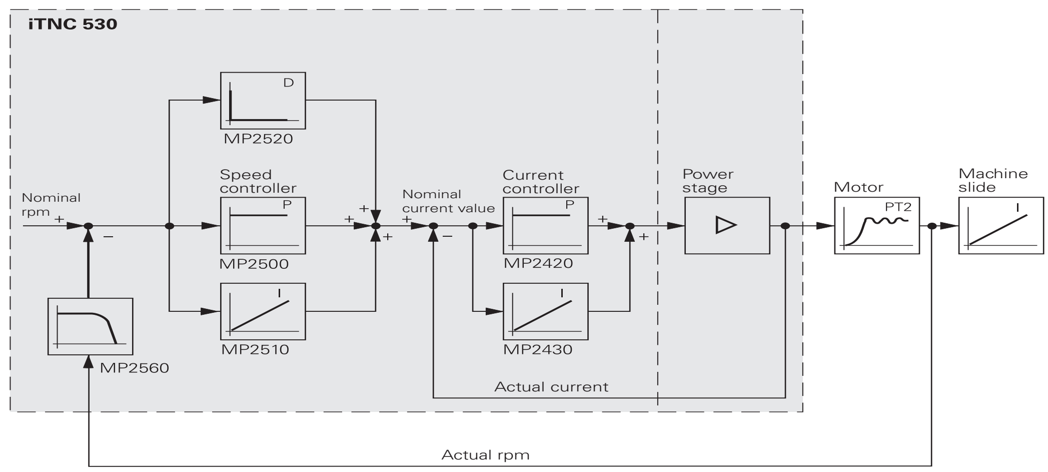
Error recognition:

Marker	Value	Meaning
M4203	0	Actual-to-nominal value transfer performed
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing M/S/T/T2/G strobe in M4176 = 1
	24	Module was called in a spawn or submit job

6.8.4 Speed Controller

6, 10 or 12 digital speed controllers for the axes and spindle(s) are integrated in the iTNC 530:

The actual speed values are measured directly at the motors with HEIDENHAIN rotary encoders. The position controller provides the nominal speed value. The speed controller is driven by the difference between nominal and actual speed values. It provides the nominal current value as output.



"Commissioning" on page 998.

With Module 9164 you can read the actual speed value of the motors.

You can adjust the step response of the speed controller:

- With the position controller switched off, enter with MP2500.x a proportional factor and with MP2510.x an integral factor for the speed controller. Adjust the step response so that only one overshoot is visible and the settling time t_{off} is as small as possible.
  Realistic values for the settling time: 3 ms to 15 ms.

MP2500.x **Proportional factor of the speed controller**
Input: 0 to 1 000 000.000 [As]

MP2510.x **Integral factor of the speed controller**
Input: 0 to 100 000 000 [A]

Module 9164 Reading the actual speed value of the motor

The resolution of the actual speed value depends on the encoder being used:

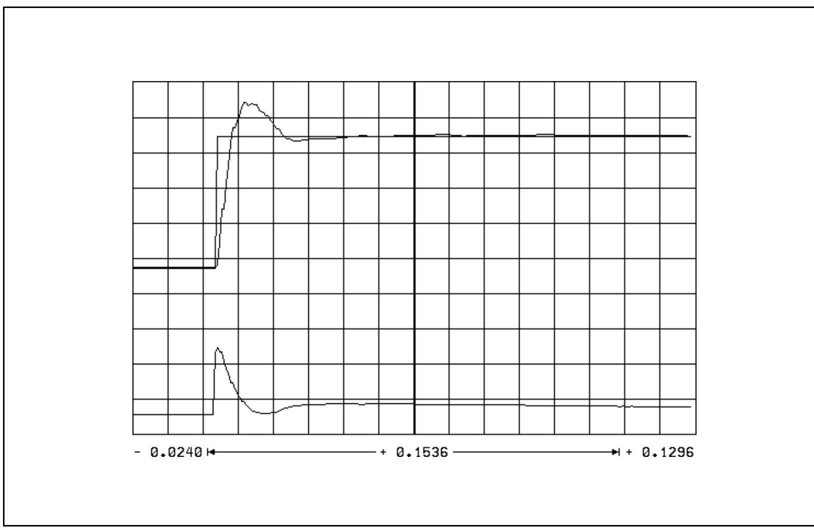
$$\text{Resolution} = \frac{1}{\text{Line count} \cdot 1024} \cdot 100\,000\ [\text{min}^{-1}]$$

Call:

PS	B/W/D/K	<>Axis>
		0 to n: Axes 1 to n+1
		15: Spindle
CM	9164	
PL	B/W/D	<>actual speed value in the format 0.001 rpm>

Error recognition:

Marker	Value	Meaning
M4203	0	Actual speed value was read
	1	Control has no integrated current controller



The step response illustrated above is idealized. In practice, interfering oscillations are superimposed on the step response.

You can reduce these interference oscillations with the differential factor, the PT_2 second-order time-delay element, the band rejection filter and the low-pass filter of the speed controller.

Differential factor

The differential factor can reduce low-frequency oscillations. However, it increases oscillations in the high frequency range.

- In MP2520.x, enter a differential factor.

Note

Ensure that the system is stable enough!

The differential factor is not recommended on machines with motors that have belt couplings. The influence of aging and temperature is too great.

Estimating the differential factor:

$$\text{MP2520.x} \approx \frac{T \cdot \text{MP2500.x}}{8}$$

MP2520.x: Differential factor of the speed controller [As^2]
MP2500.x: Proportional factor of the speed controller
T: Period of the lowest interference frequency [s]

MP2520.x **Differential factor of the speed controller**
Input: 0 to 1.0000 [As^2]

Low-pass filter

With the low-pass filter you can damp high-frequency oscillations (> approx. 450 Hz):

- Use the oscilloscope to find the fundamental frequency of the iTNC.
- Activate the 1st or 2nd order low-pass filter (see table).

Fundamental frequency of the interference oscillation	Filter order (MP2560.x)
450 Hz to 550 Hz (approx.)	First order (MP2560.x = 1)
> 550 Hz (approx.)	Second order (MP2560.x = 2)

If you cannot achieve satisfactory results with the low-pass filter, try the PT_2 element.

MP2560.x **Low-pass filter of the speed controller**
Input: 0: No low-pass filter
1: 1st-order low-pass filter
2: 2nd-order low-pass filter

PT_2 element of the speed controller

If the controlled system is insufficiently damped (e.g. direct motor coupling or roller bearings), it will be impossible to attain a sufficiently short settling time when the step response of the speed controller is adjusted. The step response will oscillate even with a low proportional factor:

- In MP2530.x, enter a value for damping high-frequency interference oscillations. If the value you choose is too high, the k_V factor of the position controller and the integral factor of the speed controller is reduced. Realistic input values: 0.0003 to 0.0020

MP2530.x **PT_2 element of the speed controller (2nd-order delay)**
Input: 0 to 1.0000 [s]

Band-rejection filter

With the band-rejection filter you can damp oscillations that you cannot compensate with the differential factor, the PT_2 element, or the low-pass filter:

- With the oscilloscope of the iTNC, find the fundamental frequency of the interference oscillations and enter them in MP2550.x.
- Increase MP2540.x incrementally until the interfering oscillation is minimized. If you set the damping too high, you will limit the dynamic performance of the control loop. Realistic input values: 3 to 9 [dB]

MP2540.x **Band-rejection filter damping of the speed controller**
Input: 0.0 to 18.0 [dB]

MP2550.x **Band-rejection filter center frequency of the speed controller**
Input: 0.0 to 999.9 [Hz]

Active damping of low-frequency oscillations

The active damping of low-frequency oscillations is suitable for damping noise oscillations of approx. 10 to 30 Hz. The damping factor is set in MP2607.x, and the damping time constant in MP2608.x. It can be calculated according to the following formula:

$$\text{MP2608.x} = \frac{k}{2 \cdot \pi \cdot f}$$

k: Factor from 0.8 to 1.0
f: Frequency to be damped (approx. 10 Hz < f < 30 Hz)

Note

The active damping should only be used if improvements actually occur, since the damping could also lead to lower and higher frequencies being fortified.

Activating the active damping:

- Ascertain the deepest resonant frequency (e.g. with the frequency diagram in TNCopt when adjusting the IPC and k_V factor).
- Set MP2607.x = 1.5.
- Calculate the damping time constant according to the above formula with k = 0.9, and enter this value in MP2608.x.
- Record **I (int rpm)** or **s diff** with the integrated oscilloscope, and move the axis with the axis-direction buttons.
- Vary the value of *k* up and down somewhat (between 0.8 and 1.0), calculate MP2608.x, and compare the oscilloscope recordings in order to find the correct value for MP2608.x (the value with the lowest amplitude).
- Vary MP2607.x, and compare with the recordings in the frequency diagram in TNCopt for the adjustment of the IPC and k_V factor.
- Select the value for MP2607.x by evaluating the advantages and disadvantages of the active damping.

MP2607.x **Damping factor for active damping**
Input: 0 to 30.000
0: No damping
1.5: Typical damping factor

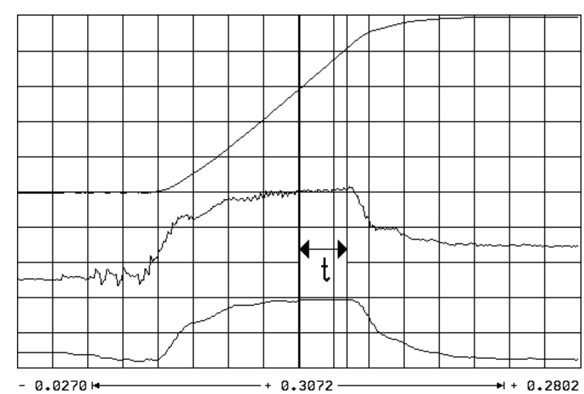
MP2608.x **Damping time constant for active damping**
Input: 0 to 0.9999 [s]
0: No damping
0.005 to 0.02: Typical damping time constant

Acceleration feedforward control

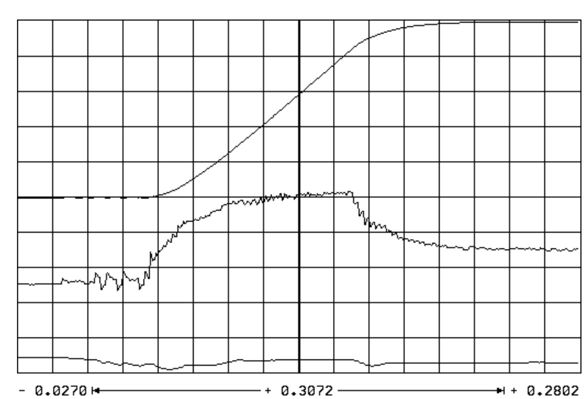
Acceleration feedforward functions only in velocity feedforward control in parallel with the speed controller.

At every change in velocity, spikes of short duration appear in the following error. With acceleration feedforward control you can minimize these spikes:

- First adjust the friction compensation. Enter values in MP2610.x to MP2620.x.
- From the integral-action component of the nominal current value I (int rpm) calculate the input value for MP2600.x.
- Adjust the acceleration feedforward control with MP2600.x.



MP2600 = 0



MP2600 = optimum

For calculation of the acceleration feedforward, the integral-action component of the nominal current value I (int rpm) is recorded with the internal oscilloscope. The actual speed value V (act rpm) and nominal current value (i nominal) are also recorded for better illustration.

$$\text{MP2600.x} = \frac{\text{I (N INT) [A]} \cdot \text{t [s]} \cdot \text{60 [s/min]} \cdot \text{MP1054.x [mm]}}{\Delta\text{V (N IST) [mm/min]}}$$

I (int rpm) = integral-action component of the nominal current value

t = acceleration time in which I (int rom) remains constant

ΔV (ACT RPM)=actual-speed-value change

MP1054.x = Traverse distance per motor revolution

MP2600.x Acceleration feedforward control
Input: 0 to 100.0000 [A/(rev/s^2)]

Limiting the integral factor

In machines with a great deal of stiction, a high integral-action component can accumulate if there is a position error at standstill. This can result in a jump in position when the axis begins moving. In such cases you can limit the integral-action component of the speed controller:

- Enter a limit in MP2512.x.
  Realistic input values: 0.1 to 2.0

MP2512.x Limit of integral factor of the speed controller
Input: 0.000 to 30.000 [s]

Integral Phase Compensation IPC

An *I* factor can be set in the speed controller of the iTNC (MP2510.x). This *I* factor is needed to attain a short setting time. However, the *I* factor has a negative influence on the position controller, i.e. the position controller tends to oscillate more easily, and it is often impossible to set the k_V factor (MP1510.x) high enough.
The IPC (Integral Phase Compensation) compensates the negative influence of the I factors on the speed controller, and makes it **possible** to increase the k_V factor (MP1510.x).

The IPC is beneficial on the following types of machines:

- Machine type 1: Machines with a dominant natural frequency between 15 Hz and 80 Hz, for which it is not possible to set a sufficiently high k_v factor.
- Machine type 2: Small-to-medium size machines that are driven directly.

Note

- The acceleration feedforward (MP2600.x) must already have been carefully adjusted for both types of machines.
- If after commissioning the IPC you wish to optimize the speed controller again, you must switch off the IPC beforehand, because the IPC influences the curve form.
- Use the same test program to commission the IPC as is used to measure the jerk and the k_v factor.

Machine type 1:

- The machine is commissioned as usual until the k_V factor is to be determined.
- Enter MP2602.x = 1 and MP2604.x = 0.
- Increase the k_V factor (MP1510.x) until you reach the oscillation limit.
- Starting value: $\text{MP2604.x} = \frac{2}{3} \cdot \frac{\text{MP2600.x}}{\text{MP2500.x}}$
- Change MP2604.x until you have found the maximum k_V factor (MP1510.x). If you cannot find a maximum k_V factor, use the default value in MP2604.x.
- Starting value: $\text{MP2602.x} = \frac{\text{MP2600.x}}{\text{MP2500.x}}$
- Increase MP2602.x until you have found a maximum k_V factor (MP1510.x). If the value found for MP2602.x is significantly greater than the starting value (> factor 2), you should adjust MP2604.x again by enlarging and reducing it to find the optimum value.
- MP1510.x = 0.65 · ascertained k_V

Machine type 2:

- The machine is commissioned as usual until the k_V factor is to be determined.
- Enter MP2602.x = 1 and MP2604.x = 0.
- Increase the k_V factor (MP1510.x) until you reach the oscillation limit.
- Starting value: $\text{MP2604.x} = \frac{2}{3} \cdot \frac{\text{MP2600.x}}{\text{MP2500.x}}$
- Change MP2604.x, normally by reducing it, until the following error is at its minimum.
- MP1510.x = 0.65 · ascertained k_V

MP2602.x	**IPC time constant T_1**
Input:	0.0001 to 1.0000 [s] 0: IPC inactive
MP2604.x	**IPC time constant T_2**
Input:	0.0001 to 1.0000 [s] 0: IPC inactive

Optimizing the jerk feedforward control

In the same manner as the other feedforward controls in a control loop, the jerk feedforward control causes following errors to be reduced or compensated during a dynamic phase (in this case the phase of the change in acceleration).

When the feedforward control is initially configured, the necessary jerk feedforward control is calculated for the known control-loop factors, but the mechanical deformation of the transmission components (e.g. due to torsion of the ball screw or the slack sides and tight sides of belt drives) and the resulting following error are unknown. This can be minimized or compensated with the dimensionless multiplier MP2606, which affects the jerk feedforward control directly.

When optimizing the jerk feedforward control all other controller parameters must already have been adjusted.

A prerequisite for the activation of the possibility of compensation in combination with the **CC 422** is an active and already adjusted IPC (Integral Phase Compensation). With the **CC 424(B)** you can also use this function without active IPC.

Commissioning:

- Enter the following test program:

```
0 BEGIN PGM TEST MM
1 LBL 1
2 L X <maximum traverse> R0 FMAX
3 L X0 FMAX
4 CALL LBL1 REP 10/10
5 END PGM TEST MM
```

- Run the program at high speed.
- Use the integrated oscilloscope to record the following error.
- **CC 422:** Adjust MP2606.x until a minimal following error occurs in the jerk phase. Here you can compensate positive following errors (MP2606.x > 1) as well as rare negative following errors (MP2606 < 1) for the jerk phase. Typical input values: 0.5 to 2.
- **CC 424(B):** Adjust MP2606.x until a minimal following error occurs in the jerk phase. Here you can compensate the following error for the jerk phase. Typical input values: 0 to 0.5 (e.g. 0.01).

Note

Please note that values over 0.5 in MP2606.x do not make sense for the CC 424(B) controller unit, and are therefore not permissible. Therefore, an error message will be issued for values greater than 0.5.

MP2606.x **Following error in the jerk phase**
Input: 0.000 to 10.000

Holding torque

The holding torque is the torque that is required to keep a vertical axis at a standstill.

The holding torque is given by the iTNC through the integral-action component of the nominal current value. In most cases the holding torque is constant. The required holding current can therefore be fed forward through MP2630.x. This relieves the speed controller.

- To prevent the effect of stiction, measure the current at low speed in both directions (e.g. 10 min^{-1}).
- Calculate the holding current from the mean of the measured current values and enter the result in MP2630.x.

Mean:

$$MP2630.x = \frac{I\ NOMINAL_1 + I\ NOMINAL_2}{2}$$

Note

- Please note that when reading the current via the internal oscilloscope, on the CC 422 you are seeing the peak value, and on the CC 424(B) you are seeing the effective value of the current.
- If the ready signal (RDY) is missing from the speed encoder inputs of vertical axes, the DSP error message **8B40 No drive < release>** appears.
A vertical axis is defined with an entry in MP2630.x.

MP2630.x **Holding current**
Input: -100.000 to +100.000 [A]

6.8.5 Switching Drives On and Off

The procedure recommended by HEIDENHAIN for switching the drives on and off, including the correct triggering of the brakes, can be configured with the PLC basic program. HEIDENHAIN recommends using the PLC basic program.

Readiness of the drives

The drives can only be switched on once the drive system is ready. This includes:

- the readiness of the supply unit (RDY.PS signal at X69 pin 17a – interrogation via Module 9066 bit 5), the inverters (RDY signal at X51 to 64 pin 10b of the PWM outputs) and of the motors.
- The global drive enabling with EMERGENCY STOP via I32 (X42/33). However, this can be switched off via MP2050.
- the conclusion of the motor orientation (determination of the field angle).
- the release via X150/X151 of the axis groups, if they are used.

Basic settings for "switching drives on"

Basic settings:

- Ensure that the clamping or brakes activate when an emergency stop is effected.
- Ensure via Module 9159 (drive controllers are switched off) that the clamping or brakes activate before the drive controllers (current and speed controllers) are switched off.
- Interrogate the readiness of the HEIDENHAIN supply module with Module 9066 bit 5.
- Before switching off the drives, save the positions of axes with common encoders.
- Determine the functionality of the global drive enabling through I32 (X42/33) with MP2050. If this function is used, ensure that connection X42 pin 33 (I32) is wired correctly. HEIDENHAIN recommends always wiring the system according to the basic circuit diagram of the iTNC 530.
- In MP2040.0 to MP2040.2, define the axis groups for drive enabling through X150/X151 pin 1 to pin 3 (e.g., MP2040.0 = %0000000000000000000000000111 determines drive enabling for axes 1 to 3 via axis group 1). Depending on the control loop being used, either X150 and/or X151 must be wired (See "X150, X151: Drive controller enabling for axis groups" on page 373). Use Module 9157 to interrogate the status of X150/X151.

If drive enabling through X150/X151 is missing, the drive motor will **always** be switched off and the position controller will be informed accordingly.

- Set W524 Bit 0 = 1, so that the control loop is opened and no error message appears when switching off the drive with X150/X151.

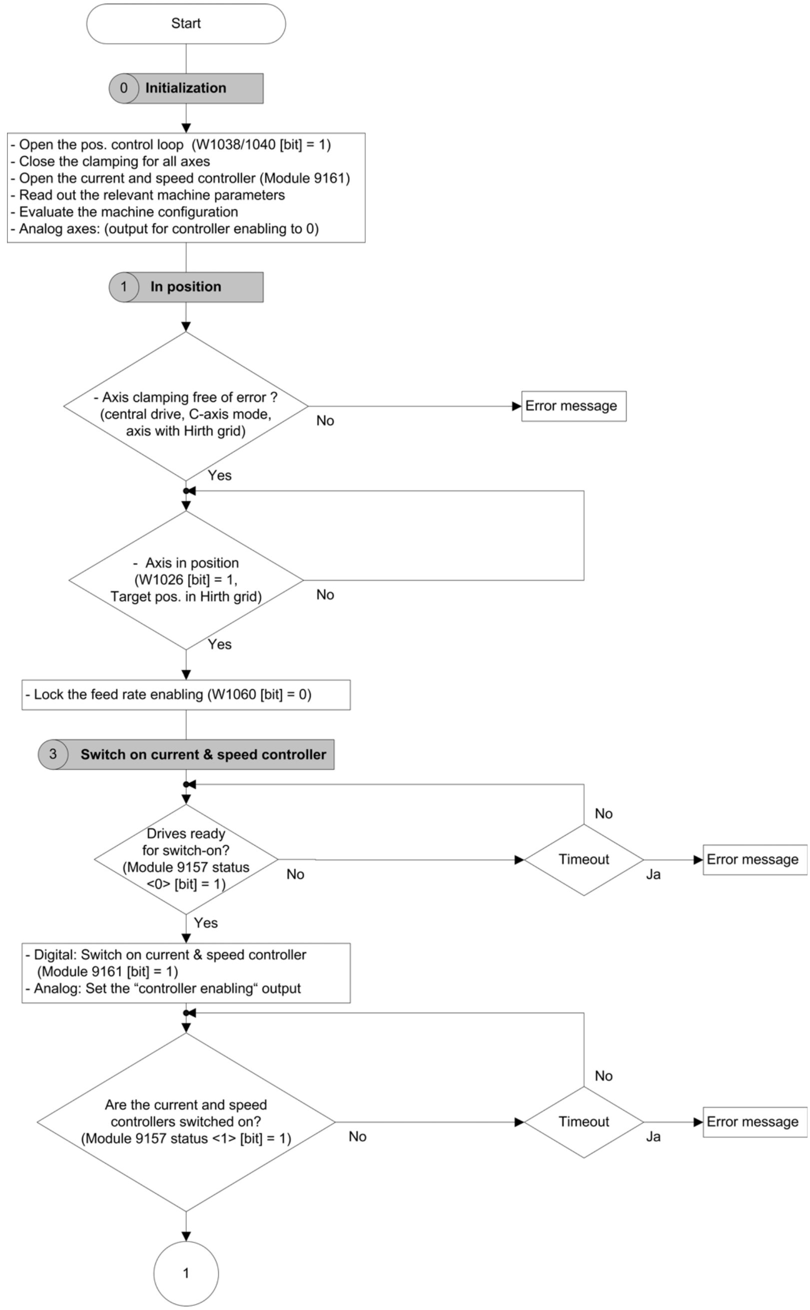
If drive enabling of axis groups via X150/X151 is used, the PLC program should perform a plausibility test to see if all axes used are defined in MP2040.x

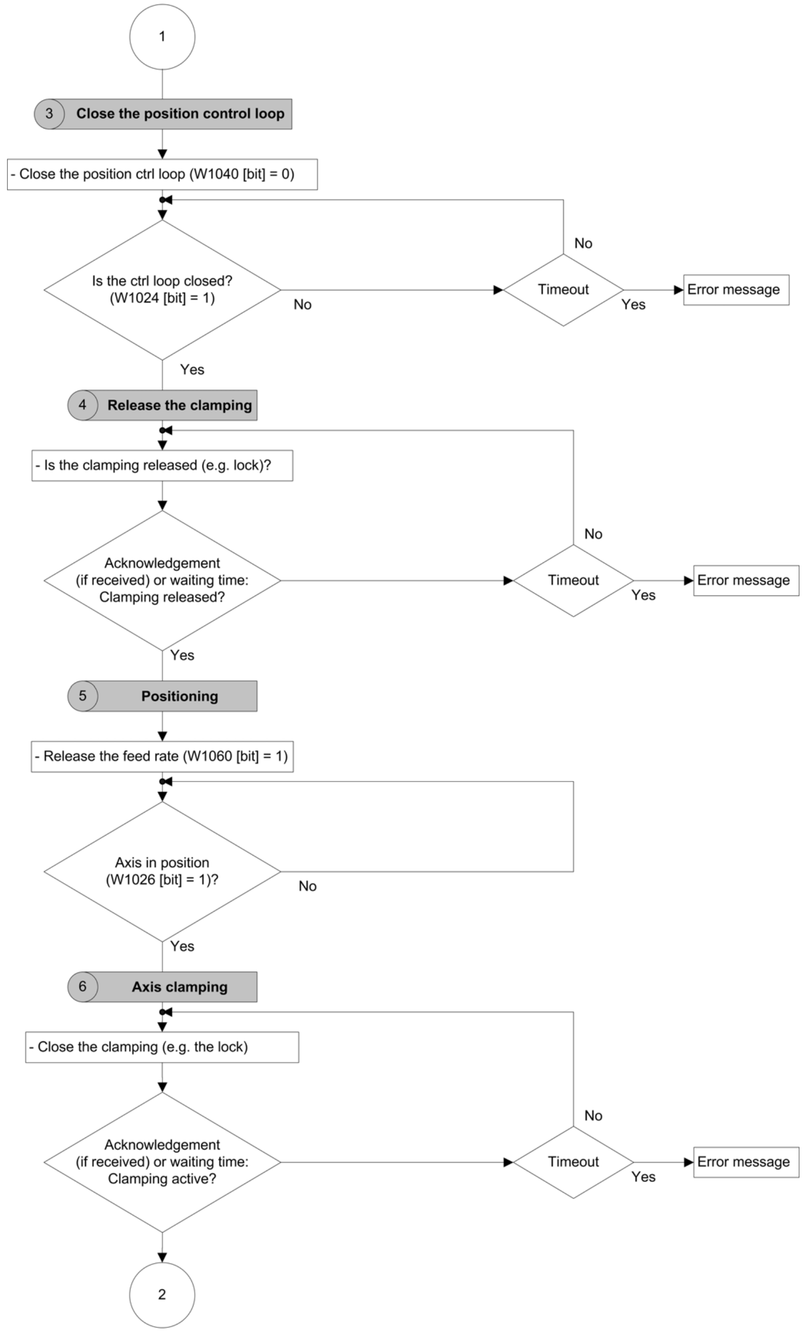
> **Danger**
>
> If a bit is defined incorrectly in MP2040.x, the drive motor might not be switched off through X150/X151.

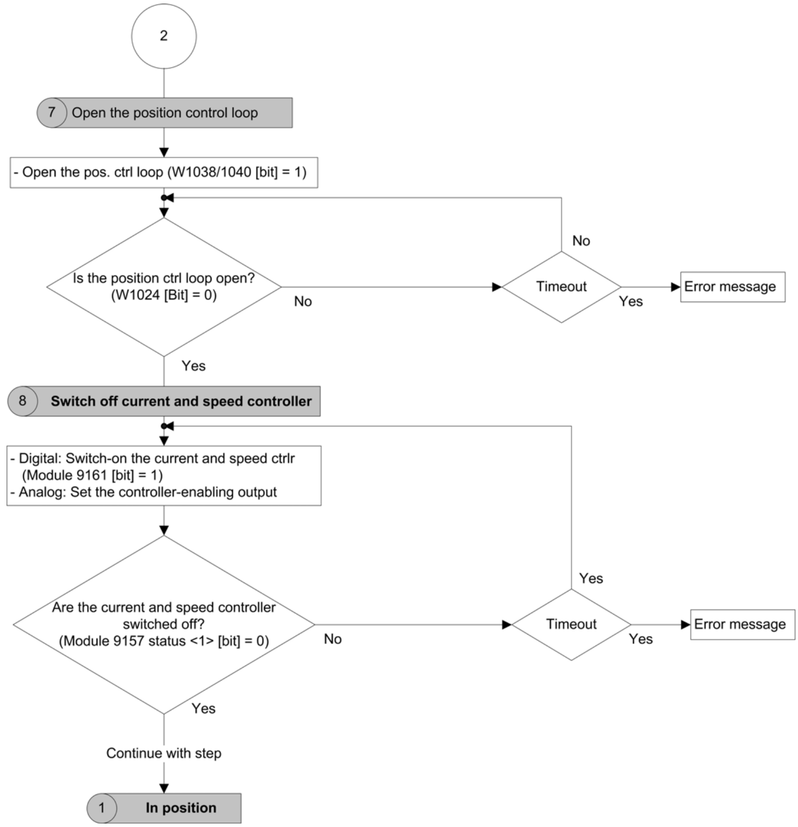
Note

- If drive enabling through X150/151 or X42/33 is missing, the error message **8B40 No drive release <axis>** appears.
- If you do **not** want to use drive enabling for axis groups, but rather just global drive enabling through I32 (X42/33), set all bits in MP2040.x to %0000000000000000000000000000 and in W524 to zero.
- The iTNC monitors the time between the switch-on of the drive controller (via Module 9161) and the READY signal of the power module (via the PWM cables). If the READY signal is missing after the waiting time has passed, the error message **8B40 No drive release <axis>** appears. In MP2170 you preset the permissible waiting time.
- If the readiness signal is reset at the PWM outputs, the drive controllers (current and speed controllers) are switched off. Normally, the error message **MOVEMENT MONITORING IN <AXIS> B** is output through the position control loop. Subsequently, the PWM signal release is switched off by the reset signal.
- As of NC software 340 420-06 the current controller is switched on 50 ms after the controller is switched on (Module 9161). This also delays the acknowledgment over Module 9162 by 50 ms.

Procedure for "switching drives on and off"




1
3 Close the position control loop
- Close the position ctrl loop (W1040 [bit] = 0)
Is the ctrl loop closed? (W1024 [bit] = 1)
No
Timeout
No
Yes
Error message
Yes
4 Release the clamping
- Is the clamping released (e.g. lock)?
Acknowledgement (if received) or waiting time: Clamping released?
Timeout
No
Yes
Error message
Yes
5 Positioning
- Release the feed rate (W1060 [bit] = 1)
Axis in position (W1026 [bit] = 1)?
No
Yes
6 Axis clamping
- Close the clamping (e.g. the lock)
Acknowledgement (if received) or waiting time: Clamping active?
Timeout
No
Yes
Error message
2


2
7 Open the position control loop
- Open the pos. ctrl loop (W1038/1040 [bit] = 1)
Is the position ctrl loop open?
(W1024 [Bit] = 0)
No
Timeout
No
Yes
Error message
Yes
8 Switch off current and speed controller
- Digital: Switch-on the current and speed ctrlr
(Module 9161 [bit] = 1)
- Analog: Set the controller-enabling output
Are the current and speed controller
switched off?
(Module 9157 status <1> [bit] = 0)
No
Timeout
Yes
Yes
Error message
Yes
Continue with step
1 In position

Avoiding global drive enabling

You can define axes for which the drives will not switch off if the global drive enabling through I32 (X42/33) is missing:

- Determine the functionality of the global drive enabling through I32 (X42/33) with MP2050.
- With Module 9169 transfer in bit code the axes that are not to be switched off.

Machine parameters, markers, modules and PLC words

MP2040 **Axis groups (for drive enabling through X150/X151)**
Format: %xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
(Spindle = bit #31)
Input: 0: Axis/spindle not assigned (disabling only through I32).
1: Axis/spindle assigned
MP2040.0-2 Axis groups 1 to 3
MP2040.3-7 Reserved, enter %00000000000000000000000000000000

MP2050 **Functionality of drive enabling I32 (X42/33)**
Input: 0: Emergency stop for all axes, Module 9169 not effective
1: Emergency stop for all axes that are not excepted with Module 9169
2: I32 and Module 9169 have no function
MP2050 can also be overwritten by the PLC and the LSV2 protocol.

MP2170 **Waiting time between the switch-on of the drive and the drive's standby signal**
Input: 0.001 to 4.999 [s]
0: 2 [s]

Bit 7 of MP2221.x enables you to deactivate the reduction of a drive's switch-on time when it is switched on with PLC module 9161. If spindles with wye/delta switchover but without evaluation of the status of the wye/delta contactor are used, it may occur that the drive is switched on again before the contactor has switched over to the appropriate mode. If this occurs, the current controller sets the voltage to the maximum possible value. If the contactor is then closed, this may lead to overcurrent, and as a result, to a switch-off (e.g. due to error message 8B60) of the drive.

MP2221.x **Bit 7 – Switch-on time of the drive**
Input: 0: Reduction of the switch-on time is active
1: Reduction of the switch-on time is not active

		Set	Reset
W524	**Open the control loop if drive enabling via X150/X151 is missing** Bit 0 = 0: Position control loop stays closed Bit 0 = 1: Position control loop is opened Bit 1 to bit 15: Non-functional	PLC	PLC

Module 9157 Drive controller status

Status information about the drive controller can be ascertained with this module.

Call:

```
PS    B/W/D/K   <>Selection of status information>
                0: Readiness of drives (bits 0 to 13)
                   Result 0: Drive is not ready and cannot be switched on
                   Result 1: Drive is ready and can be switched on
                1: Drive controller status (as in Module 9162)
                2: Axis enabled through X150/X151
                   (bits 0 to 13 = 1) or axis not enabled (bits 0 to 13 =0)
                3: Signal to X150/X151
                   (bits 0 to 7 = X150; bits 8 to 15 = X151)
                4: Spindle in operating mode 0 (bit 15 = 0) or operating
                   mode 1 (bit 15 = 1)
CM    9157
PL    B/W/D     <>Axis status information bit-encoded>
```

Error recognition:

Marker	Value	Meaning
M4203	0	Status information was ascertained
	1	Error code in W1022
W1022	2	Invalid status information was programmed
	24	Module was called in a spawn job or submit job

Module 9159 Advance status report: Drives will be switched off

Call:

```
CM    9159
PL    W/D       <>Drives, in bit code, that are switched off in the time
                defined in MP2308>
```

Module 9161 Enable the drive controller

With this module you can switch the drive controllers (speed and current controllers) on and off for specific axes. A nominal speed value is also output when the drive controller is not enabled.

Software 340 49x-03 introduced a reduction of the switch-on time when a drive is switched on by PLC Module 9161. In some cases, problems occurred after a software update. You can deactivate the reduction of the switch-on time with bit 7 of MP2221.x.

Call:

PS	W/D/K	<>Released axes>
		Bit 0...13 -> axis 1...14, Bit 15 -> spindle
		0: No drive controller enabling
		1: Drive controller enabling
CM	9161	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Control has no current controller, or the call is in a spawn or submit job

Module 9162 Status request of the drive controller

Call:

CM	9162	
PL	B/W/D	<>Drive is ready>
		Bit 0...13 -> axis 1...14, Bit 15 -> spindle
		0: Not ready
		1: Ready

Module 9169 Axes for which I32 does not switch off the drives

Call:

PS	B/W/D/K	<>Axes bit-encoded>
CM	9169	

		Set	Reset
M4563	**Global feed-rate enable for all axes**	PLC	PLC

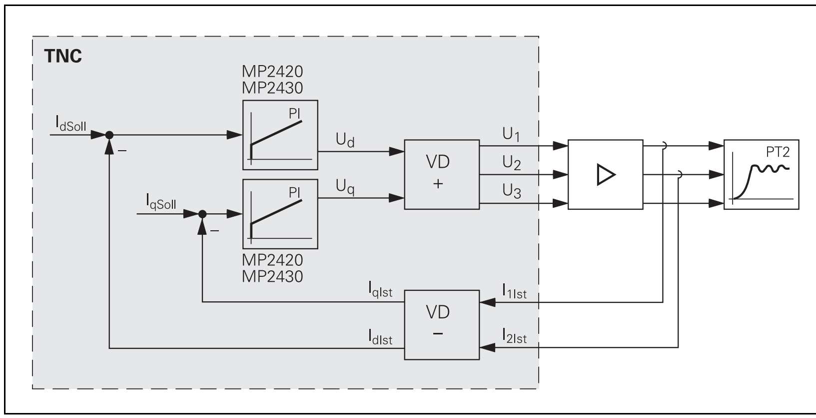
6.8.6 Current Controller

6, 10 or 12 digital current controllers for the axes and spindle(s) are integrated in the iTNC 530.

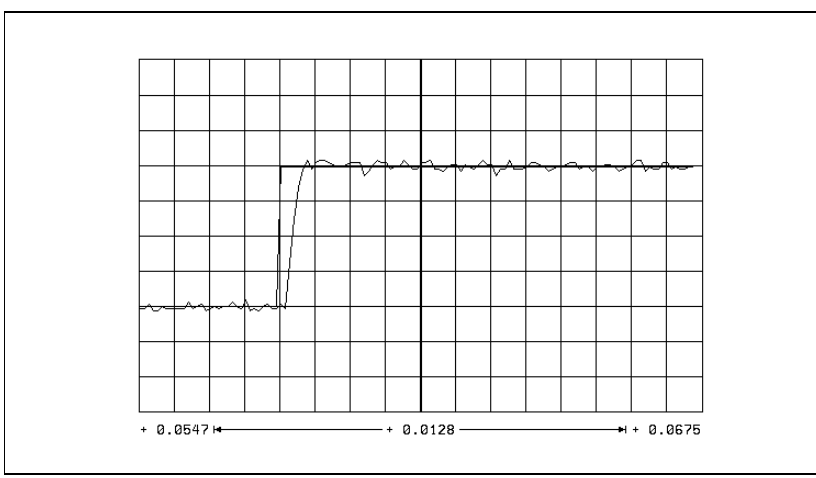
The nominal values for magnetizing current I_{dnom} and torque current I_{qnom} are divided into the PWM signals U_1, U_2 and U_3 through a PI controller and vector rotator VD+, and are transferred to the power module through X51 to X60.

The actual current values I_{1act} and I_{2act} are determined by the power module and are transferred to vector rotator VD– through X51 to X60. The vector rotator determines the actual values of magnetizing current I_{dist} and torque current I_{qnom}.

Circuit diagram:



You adjust the current controller to attain the optimum result, with the position and speed controller switched off.



The step response is adjusted such that there is no overshoot and the rise time is as small as possible:

- In MP2420.x, define the P factor of the current controller.
  Automatic calculation of the P factor for synchronous and asynchronous motors is also possible. However, automatic calculation is not to be used for linear synchronous and torque motors.
  The calculated value is entered in the MP file. An ***** is appended to the calculated value to show that the parameter was determined automatically.
- In MP2430.x, define the I factor of the current controller.
  Automatic calculation of the I factor for synchronous and asynchronous motors is also possible. However, automatic calculation is not to be used for linear synchronous and torque motors.
  The calculated value is entered in the MP file. An ***** is appended to the calculated value to show that the parameter was determined automatically.

MP2420.x **P factor of the current controller**
Input: 0 to 9999.99 [V/A]
* = automatic calculation of the P factor

MP2430.x **I factor of the current controller**
Input: 0 to 9 999 999 [Vs/A]
* = automatic calculation of the I factor

6.8.7 Braking the Drives for an Emergency Stop and a Power Fail

In an emergency stop and power failure the spindle must be braked as quickly as possible. If the braking energy cannot be drawn off quickly enough, the dc-link voltage increases sharply. Under circumstances, the inverter could switch off and the spindle coast to a stop. A powerful braking of the spindle also leads to a high strain on the mechanics of the machine. Preferably the spindle should be braked in an emergency stop by limiting the braking power.

Note

Limiting the braking power is also effective when braking the spindle with M05, if the brake ramp in M05 (MP3411 and MP3412) is steeper than the brake ramp when limiting the braking power.

Normally, in case of an emergency stop, the axes are braked at the limit of current. This can create problems:

- With gantry axes a mechanical offset can occur between the master and slave axes.
- A gear between spindle and motor can be overloaded.

In such cases, the axes should preferably be braked in an emergency stop by adding an additional braking ramp.

Note

Both of the above braking strategies are possible for axes and spindles. If both braking strategies are activated for an axis or spindle, they do not exclude each other; this means that in case of an emergency stop, whichever strategy responds first takes effect.

Problems with inverters **without** regenerative power supplies (with braking resistors) during braking mostly arise if the inverter is switched off too early. The strain on the mechanics during braking is reduced, but can also be influenced with braking strategies.
Inverters **with** regenerative power supplies usually do not develop problems if they are switched off. The main concern here is for the mechanics of the machine. If the maximum regenerative power of the inverter is exceeded when braking the drives (during an emergency stop, for example), the axes and spindle(s) coast to a stop. In this case it is sensible to define separate maximum braking powers for each drive in MP2390.x. This will ensure that each drive is braked as quickly as possible.

Braking the axes by entering an additional braking ramp

In this strategy, the braking ramp to be used in an emergency stop is entered.

Set the axis braking ramp for an emergency stop:

- Enter as a minimum value in $MP2590.x = \frac{MP1060.x \cdot 60}{MP1054.x}$
- Use the emergency stop to brake the axis from rapid traverse.
- Increase the value entered in MP2590.x until the braking time is as short as possible and the mechanics of the axis are not overloaded.

If the value entered is too small, i.e. if braking is too slow, the axis brakes at the acceleration defined in MP1060.x.

Once the value for MP2590.x has been determined and entered, the braking process at this setting must be checked. You must ensure that simultaneous braking of all axes does not lead to a switch-off due to overload of the inverters. The nearer the determined braking ramp is to the emergency stop ramp, the greater the resulting load on the inverters is.

Note

The value entered in MP2590.x refers to the motor speed, meaning the ball screw pitch is not considered.

MP2590.x **Braking ramp in an emergency stop**
Input: 0.1 to 999.9 [min^{-1}/ms]
0: Function inactive

Braking the spindle/spindles by entering the braking power

In this strategy the maximum braking power for braking the spindle/spindles in an emergency stop or power failure is entered.

If power limiting (MP2392.x) is used in normal operation, then the maximum braking performance is limited to the lower of the two values from the power limiting and the braking power.
Example:

Function	Case 1	Case 2
Power limiting (MP2392.x)	10 kW	5 kW
Maximum braking power (MP2390.x, MP2394.x)	5 kW	10 kW
Limiting the braking performance to	5 kW	5 kW

Braking the spindle/spindles in an emergency stop

- For **inverters with regenerative power supply**, enter MP2390.x = 0 so as not to limit the braking power.
- Calculate for **inverters with braking resistors** the input value for MP2390.x from the following formula:

$$\text{MP2390.x} = \frac{U_Z^2}{R \cdot 1000}$$

R = Braking resistance [Ω]
U_Z = dc-link voltage [V]

Starting with NC software 340 422-10 and 340 480-10, the maximum braking power from MP2390.x may be greater than the power limit from MP2392.x.

Braking of the spindle during an emergency stop can be delayed by the time in MP3550. This avoids tool breakage during power milling, because low spindle speeds might result in the spindle coming to a standstill before the axes do. However, the spindle braking is delayed by no more than half a spindle revolution.

Braking the spindle/spindles during a power fail

During a power fail, the "SH1B" signal at X51 to X60 is maintained for 10 more seconds, in order to brake the spindle/spindles. At the same time, the control tries to reset the PLC outputs.

- If you are using an additional braking resistor (e.g. UP 110) in connection with an **inverter with regenerative power supply,** calculate the value to be entered in MP2394.x from the above formula.
- Calculate for **inverters with braking resistors** the input value for MP2394.x with the above formula.

Note

If after entry of a value in MP2390.x or MP2394.x the mechanics are overloaded by the braking process, lower the value in MP2390.x or MP2394.x until you have found an optimum between braking time and mechanical loading.

MP2390.x **Maximum braking power**
Input: 0: No limiting of the braking power in an emergency stop
0.001 to 3000.000 [kW]

MP2394.x **Maximum brake power for power failure**
Input: 0: No limiting of the braking power in a power failure
0.001 to 3000.000 [kW]

MP3550 **Delay of EMERGENCY STOP reaction of spindles**
Input: 0.001 to 0.100 [s]
0 = Delay not active

6.8.8 Power and Torque Limiting

You can limit the power of your spindle motor to achieve wider gear ranges. Wide-range motors are characterized by a larger speed range with higher torque at low speed.
One solution for bringing about this behavior is to use an oversized motor, and to limit the maximum power. However, power limiting does not reduce the high torque to the speed at which power limiting becomes effective. This high torque (until power limiting takes effect) can be reduced with torque limiting, in order to keep the mechanics of the machine from becoming overloaded.

As of software version 340 49x-05, the maximum spindle power is always limited in order to avoid overloading the power supply module. If MP2392 is set to zero and a power supply module is defined in MP2198, the maximum spindle power is limited to the value S6-40 from the power supply module table (Supply.spy). For applications requiring a higher spindle power (i.e. the spindle should be able to take up the maximum power P-Max from Supply.spy), the desired maximum power must be entered in MP2393.

With torque limiting you can also limit the torque of the axis motors, in order to keep the mechanics of the machine from becoming overloaded. Power limiting is not useful for axis motors.

For **axes and spindles**, the torque is limited to the value taken from either the table of power modules or the motor table, whichever is lower.

On supply units where the $\overline{\text{ERR.IZ.GR}}$ signal is available, the power of the spindle is limited via MP2392.x in case of error (not for axes).

HEIDENHAIN recommends activating this monitoring function via MP2220.x bit 2 (not with UE 2xx).

The torque can be calculated for any speed:

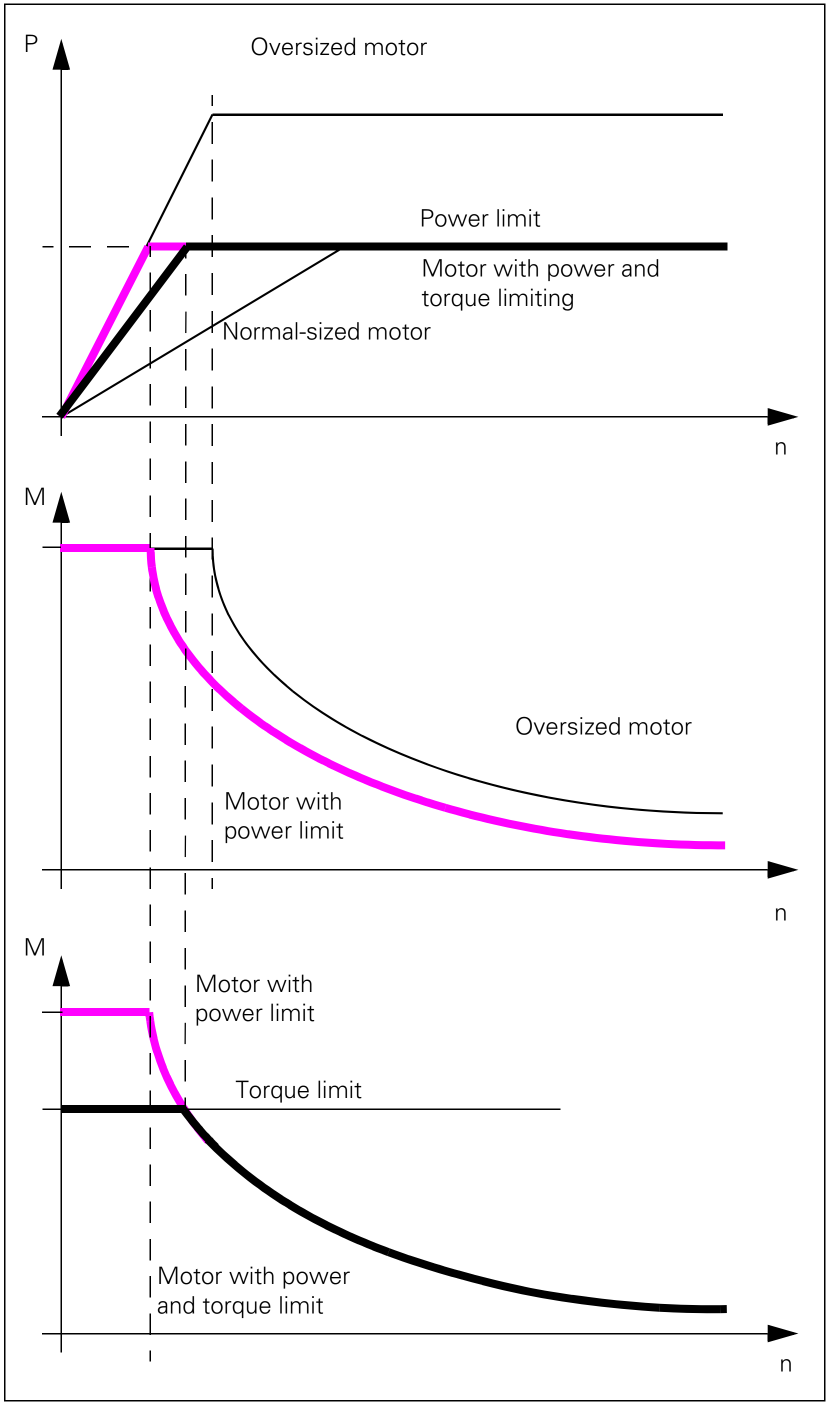
$$M = \frac{P \cdot 60}{n \cdot 2 \cdot \pi}$$

M: Torque [Nm]
P: Power [W]
n: Speed [min^{-1}]

Note

The power and torque limiting can have an effect on the braking of the spindle in an emergency stop.

- Enter the maximum power for the spindle in MP2392.x.
- Enter the maximum torque for the spindle or axis in MP2396.x.
- Activate the power limiting of the spindle at $\overline{\text{ERR.IZ.GR}}$ via MP2220.x bit 2 = 0 (not for UE 2xx)



P
Oversized motor
Power limit
Motor with power and
torque limiting
Normal-sized motor
n
M
Oversized motor
Motor with
power limit
n
M
Motor with
power limit
Torque limit
Motor with power
and torque limit
n

MP2220.x **Monitoring functions**
Input: Bit 2 – Power limit of spindle at $\overline{\text{ERR.IZ.GR}}$ (only for HEIDENHAIN inverters, except UE 2xx)
0: Power limit active
1: Power limit inactive

MP2392.x **Power limit**
Input: 0: No power limit
0.001 to 3000.000 [kW]

MP2396.x **Maximum torque**
Input: 0: No torque limiting
0.1 to 30 000.0 [Nm]

Power limiting of the spindle after PLC request

An additional maximum drive power rating can be defined with the machine parameter MP2393.x. This function must be activated via the PLC module 2314. This makes it possible to reduce power consumption for starting up the spindle. This can prevent the maximum power of the power supply module from being exceeded if an axis is positioned at the same time as the spindle is started.

The PLC knows (via M3/M4/M5) when the spindle is being started up and when the defined spindle speed has been reached. During the PLC positioning movements (tool change) you can switch to the power in MP2393. During machining (low power consumption of the axes, low traversing speeds) you can switch to the maximum spindle power (in MP2393, or no power limit).

MP2394.x **Power limiting after PLC request**
Input: 0: No power limit
0.001 to 3000.000 [kW]

Module 9158 Maximum torque

With Module 9158 you can limit the torque of an axis or spindle. The maximum torque resulting from the data in the control's motor table cannot be exceeded. In this case the torque is limited to the value from the motor specifications. After the drive is switched off, the original torque becomes effective again.
If torque limiting is active, the standstill monitoring is inactive; only the motion monitoring remains active.

The torque-producing current required for the desired torque must be transferred to the module:

Synchronous motor	Asynchronous motor
$$k_M = \frac{P_N}{I_N \cdot 2 \cdot \pi \cdot \frac{n_N}{60}}$$ $$I_q = \frac{M}{k_M}$$ I_q: Torque-producing current M: Desired torque k_M: Torque constant n_N: Rated speed (from motor table) I_N: Rated current (from motor table) P_N: Power rating (from motor table)	■ Armature control range ($n < n_{FS}$) $$I_q = \frac{M \cdot n_N \cdot 2 \cdot \pi \cdot \sqrt{I_N^2 - I_0^2}}{P_N \cdot 60}$$ ■ Field weakening range ($n > n_{FS}$) $$I_q = \frac{M \cdot n_N \cdot n \cdot 2 \cdot \pi \cdot \sqrt{I_N^2 - I_0^2}}{P_N \cdot n_{FS} \cdot 60}$$ I_q: Torque-producing current M: Desired torque n_N: Rated speed (from motor table) n: Current speed I_N: Rated current (from motor table) I_0: No-load current (from motor table) P_N: Rated power output (from motor table) n_{FS}: Threshold speed for field weakening (from motor table)

Danger

If Module 9158 is used, then certain monitoring functions regarding the drives must be switched off. Please note the following error messages and their possible consequences (see DSP error messages)

- `C380 Motor <axis> not controllable`
- `C3B0 Motor <axis> does not rotate`

Attention

If you are using a CC 422 controller unit, then please note that the current values in the oscilloscope are all peak values (and not effective values as with the CC 424(B)). Measured values then appear greater by a factor of the square root of 2 than the calculated values.

Call:

PS B/W/D/K/S<>Axis or spindle>
    0 to n: Axis 1 to axis n + 1
    15: Spindle
PS B/W/D/K/S<>Torque-producing current in mA>
    –1 = Torque given in motor specifications
CM 9158

Error recognition:

Marker	**Value**	**Meaning**
M4203	0	Torque preset active
	1	Error code in W1022
W1022	1	0 Nm torque transferred
	2	Invalid axis number, open-loop axis, or uncontrolled axis
	24	Module was called in a spawn or submit job

Module 9128 Torque limiting by the PLC

Module 9128 can be used to program a maximum torque for the programmed axis. The torque of the drive is limited to the programmed value. The value –1 cancels the torque limitation and the value from the motor data becomes effective again. The torque can be limited in [mA] or in [0.1%] of the rated current.

Condition:

- The module is only executable in the cyclic PLC program.
- The programmed value for the maximum torque may not be higher than the value in the motor data. If the programmed value is higher than the value in the motor data, the value in the motor data is used as the limit.
- A torque value of 0 cannot be programmed.
- Programming a torque value of –1 cancels limitation. The original value from the motor data becomes effective again.
- The unit of the resulting torque is [mA].
- If a drive is switched off, the torque from the motor data becomes effective when it is switched on again.
- The notes for Module 9158 must be considered!

Call:

```
PS      B/W/D/K   <>Mode>
                  0: Current in [0.1%] of the rated current
                  1: Current in [mA] (like Module 9158)
PS      B/W/D/K   <>Axis number>
                  15: Spindle
PS      B/W/D/K   <>Torque>
                  –1: Cancel the torque limiting
CM      9128
```

Error recognition:

Marker	Value	Meaning
M4203	0	Torque limiting programmed
	1	Error code in W1022
W1022	1	Invalid value for torque
	2	Invalid value for axis number or mode, axis is an open-loop axis or is temporarily not a closed-loop axis
	24	Module was called in a spawn or submit job

Module 9129 Status of torque limiting by the PLC

Module 9129 is used to determine the current status of torque limiting for the programmed axis. The momentary maximum torque can be determined in [mA] or in [0.1%] of the rated current.

Condition:

- The module is only executable in the cyclic PLC program.
- The greatest possible return value is the value resulting from the motor data.
- If torque limiting is not active, the maximum current can be determined from the motor data.

Call:

```
PS   B/W/D/K  <>Mode>
              0: Limiting active/inactive
              1: Current in [mA]
              2: Current in [0.1%] of the rated current
PS   B/W/D/K  <>Axis number>
              15: Spindle
CM   9129
PL   B/W/D    <>Status>
              Mode 0: 0 = Limiting active / 1 = Limiting inactive
              Mode 1: Current in [mA]
              Mode 2: Current in [0.1%] of the rated current
```

Error recognition:

Marker	Value	Meaning
M4203	0	Status ascertained
	1	Error code in W1022
W1022	2	Invalid value for axis number or mode, axis is an open-loop axis or is temporarily not a closed-loop axis
	24	Module was called in a spawn or submit job

6.8.9 Weakened Field Operation

General information

Asynchronous motors are usually operated with a weakened field. This operating mode can become necessary for synchronous motors if the present inverter voltage does not suffice for the require rotational speed. Typical applications for this operating mode are high-speed synchronous spindle motors and "high-speed" torque motors.

For synchronous drives the operation with a weakened field is automatically activated if MP2160.x = 1 has been entered, and the desired speed makes it necessary. Under certain conditions a voltage protection module must be used.

Because of their design, the HEIDENHAIN EcoDyn motors are treated as a special case. They must always be operated with a weakened field, but no voltage protection module is necessary.

For HEIDENHAIN EcoDyn motors, MP2160.x = 2 is to be entered.

The speed-torque characteristics from the drive manufacturer indicate whether weakened-field operation is necessary. Among other information, they show the possible speeds in combination with the necessary inverter voltage.

> Note
>
> Please note that in general there are other possible settings for weakened-field operation with the CC 424(B) (See "Peculiarities in weakened-field operation" on page 1097).

Using the voltage protection module

If the power supply fails during weakened-field operation, and the synchronous drive is running at a high speed at the same time, then the dc-link voltage can rise rapidly (generator effect of the drive). If this voltage rises to over 850 V, then inverters and possibly the motor itself can become damaged. Reliable protection against this is offered by a voltage protection module, which short-circuits the motor phases when the trigger threshold of 850 V is exceeded. (for example SM 110 or SM 130; see the "Inverter Systems and Motors" Technical Manual).

Whether a voltage protection module is necessary for a drive depends on the desired speed, the nominal speed **N-N** and the no-load voltage **U0** of the drive. See the motor table for these values.

The following formula calculates the motor speed at which a voltage greater than 850 V would be induced by the generator effect. If the desired motor speed is greater than the one calculated, then the voltage protection module (SM 110 or SM 130) **must** be used!

$$N_{max} = \frac{850\ V \times N_{noml}}{U_0 \times \sqrt{2}}$$

Please refer to the "Inverter Systems and Motors" Technical Manual to see which voltage protection module is needed for which motor types (depending on the maximum phase current of the SM 110 or SM 130).

> Attention
>
> A braking resistor, such as PW xxx or UP 110, does not offer sufficient protection.

Setting the weakened-field operation

- Enter MP2160.x = 0 if you want to operate synchronous motors **without** a weakened field.
- Enter MP2160.x = 1 if you want to operate the synchronous motor with a weakened field (check if a voltage protection module must be used).

The EcoDyn synchronous motors from HEIDENHAIN are operated with limited field weakening. No voltage protection module is necessary here.

- Select from the motor table the motors with the designation **`QSY1xxx EcoDyn`** or **`QSY1xxx EcoDyn EnDat`** for MP2200.x.
- Enter MP2160.x = 2 if you are using EcoDyn synchronous motors from HEIDENHAIN.

MP2160.x	**Field weakening for synchronous motors**
Input:	0: No weakened-field operation 1: Weakened-field operation permitted (check if a voltage protection module is necessary) 2: Operation of a HEIDENHAIN EcoDyn motor

6.9 Offset Adjustment

Digital axes:

An offset adjustment at the output of the current controller is automatically compensated.

Analog axes:

The maximum permissible offset voltage in the control is 100 mV. If this voltage is exceeded, the error message **EXCESSIVE OFFSET IN <AXIS>** appears.

6.9.1 Offset Adjustment with Integral Factor

With the integral factor you can adjust an offset automatically:

- Enter an integral factor in MP1080.x. Depending on the size of the factor the offset voltage will be reduced quickly or slowly.
- Play in the drives can result in instability in the control loop.
  In this case, enter the factor zero.

MP1080.x is effective only at a standstill.

MP1080.x	**Analog axes: Integral factor for offset adjustment**
Input:	Input 0 to 65 535
	Digital axes: No function
	Input: 0

6.9.2 Offset Adjustment by Code Number

- Activate the offset adjustment with the code number 75 368.

The iTNC displays the offset values of the analog axes in the dialog line. The values show the setting of the voltage in 0.15-mV steps.
Display value 10 means: 10 · 0.15 mV = 1.5 mV.
The displayed offset value consists of the offset values that are generated in the motor controller and in the control.

If the values are to be automatically compensated:

- Press the ENT key or the CONTINUE soft key. The control outputs a corresponding countervoltage.

If nothing is to be changed:

- Press the END soft key.

If the offset adjustment is to be switched off again:

- Enter the code number 75 368 and press the NO ENT key or the CANCEL soft key.

The offset values are saved in the control and remain safe in the event of power interruption. After a control is exchanged, the offset adjustment must be repeated by means of the code number.

6.10 Contouring behavior

6.10.1 Radial Acceleration

You can define the radial acceleration of axes in addition to the simple acceleration (MP1060.x):

- Define the radial acceleration in MP1070.

MP1070 limits the feed rate during circular movement according to the formula:

$$v = \sqrt{r \cdot MP1070}$$

v = feed rate during circular movement [m/s]

r = radius [m] (of the path of the tool center)

HEIDENHAIN recommends:

$$MP1070 = 0{,}5...1 \cdot MP1060$$

If the programmed feed rate is less than that calculated above, then the programmed feed rate becomes effective.

MP1070 functions for operation with both following error and feedforward control.

MP1070	**Radial acceleration**
Input:	0.001 to 500 [m/s^2]

6.10.2 Contour Velocity at Corners

To comply with a defined tolerance, the iTNC can reduce the tool velocity before machining corners, line-to-arc transitions and arc-to-arc transitions. The control can react to a potential violation velocity tolerance up to 256 blocks in advance. This feature is known as "look-ahead":

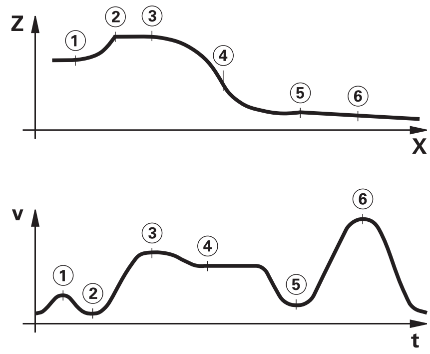
- Define the permissible tolerance for contour transitions in MP1096.x for movements at the machining feed rate and at rapid traverse. The larger the tolerance, the greater the tool velocity.

The user can overwrite this tolerance with Cycle 32 "Tolerance."

Jerk limitation (See "Interpolator" on page 817) and nominal-position-value filters enable the iTNC to machine 3-D surfaces at high speed. Prerequisite: The contour must be described with short line segments.

To ensure that cutter-compensated outside corners remain exact, a spline must be inserted into the cutter midpoint path instead of a transitional arc. The longer path of the spline (compared to the circle) results in an increased machining time. A spline also has the advantage of reducing the jerk:

- Enter bit 10 = 1 in MP7680.



Note

`FN17: SYSWRITE ID1020 NR1 = <value>` can be used from the NC program to activate machine parameter subfiles for various machining operations. See 1638.

MP1096	**Tolerance for contour transitions at corners**
Input:	0: No nominal position value filter 0.001 to 3.000 [mm]: Permissible tolerance at contour transitions
MP1096.0	With machining feed rate
MP1096.1	With rapid traverse

MP7680	**Machine parameter with multiple function**
Input:	Bit 10 – Cutter-radius-compensated outside corners: 0: Insertion of a circular arc 1: Insertion of a spline curve Proposed input value: %xx1xxxxxxxxxx Bit 11 – Reserved

Rounding of corners

If you program M90, the tool velocity in following-error mode is kept constant at corners without radius compensation. This causes a corner rounding that varies with the feed rate (see the User's Manual).

If you program M112 or M124, defined arcs will be inserted at the corners regardless of the feed rate (see the User's Manual). The rounding arcs generate twice as many NC blocks, and the feed rate is now only limited by the radial acceleration.

- With MP7680 bit 7, specify whether the rounding arcs should always be inserted or only if the acceleration from MP1060.x or MP1070 has been exceeded at the corners.
- With MP7680 bit 8, specify whether a rounding arc or a cubic spline is to be inserted between lines during the M function M112. The feed rate is reduced enough to prevent any excessive jerk. This does not apply if F MAX is programmed. The cubic spline produces an additional jerk reduction. However, it requires a longer processing time than an inserted rounding, and due to the longer path of the spline (compared to the circle), the machining time also increases.

If you have set bit 8, you can specify with bit 9 whether the jerk will remain constant on the spline. The contour speed is adjusted for constant jerk.

MP7680	**Machine parameter with multiple function**
Input:	Bit 7 – Insertion of a defined rounding arc or spline: 0: Defined rounding arcs are always inserted 1: Defined rounding arcs are always inserted if the acceleration from MP1060.x or MP1070 was exceeded. Bit 8 – Insertion of a rounding arc or cubic spline: 0: Rounding arc is inserted. 1: A cubic spline is inserted instead of a rounding arc. Bit 9 – Constant jerk on spline (Bit 8 = 1): 0: No constant jerk 1: Constant jerk

6.11 Monitoring Functions

The NC monitors the axis positions and the dynamic response of the machine. If the fixed values are exceeded, it displays an error message and stops the machine.

With W1042 you can switch off the following types of monitoring for individual axes:

- Position monitoring
- Standstill monitoring
- Movement monitoring
- Nominal speed value monitoring

		Set	Reset
W1042	**Deactivation of monitoring functions** Bits 0 to 8 represent axes 1 to 9 0: Monitoring functions active 1: Monitoring functions inactive	PLC	PLC

Attention

Safe machine operation is not possible if the monitoring functions are switched off. Uncontrolled axis movements are not detected.

If the reaction time of the PLC for switching off the monitoring functions is not sufficient, you must use a high-speed PLC input. High-speed PLC inputs are interrogated within the position control loop cycle:

- In MP4130.0, enter the number of the PLC input that is to be defined as high-speed PLC input. The number "0" is valid and means input number "0".

Note

The inputs of the PL 4xxB and PL 510 cannot be used as high-speed PLC inputs.

- Define in MP4131.0 the activation criterion for the PLC input specified in MP4130.0.
- Enable MP4130.0 with W522 bit 0. As soon as the input is set, the monitoring functions are switched off, the axes stopped, and the drive is switched off. If the following error is greater than MP1030.x (positioning window), the actual value is saved as nominal value. The monitoring functions become active again if the high-speed PLC input is reset or MP4130.0 has been disabled with W522 bit 0.

MP1150.1	**Time period for which the monitoring function is to remain off after the fast PLC input defined in MP4130.0 is set**
Input:	0 to 65.535 [s] Recommended: 0.2 to 0.5
MP4130.0	**Number of the high-speed PLC input for switching off the monitoring functions**
Input:	0 to 20 000 [no. of the PLC input] -1: Function inactive The inputs of the PL 4xxB and PL 510 may not be used!
MP4131.0	**Activation criterion for fast PLC input for switching off the monitoring functions**
Input:	0: Activation at low level 1: Activation at high level

		Set	Reset
W522	**Enabling the high-speed PLC inputs** Bit 0: Fast PLC input is defined in MP4130.0 for switching off the monitoring functions	PLC	PLC

6.11.1 Position Monitoring

The axis positions are monitored by the iTNC as long as the control loop is closed.

The input values for position monitoring depend on the maximum possible following error (servo lag). Therefore the input ranges for operation with following error and velocity feedforward are separate.

For both modes of operation there are two range limits for position monitoring.

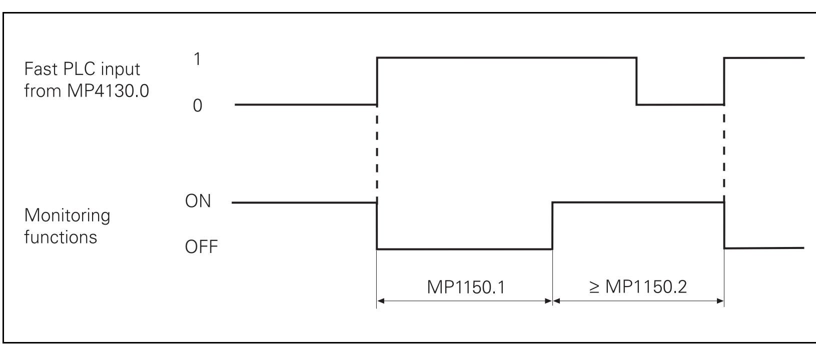
If the first limit is exceeded, the error message **EXCESSIVE SERVO LAG IN <AXIS>** appears. The machine stops.
You can clear this message with the CE key. An actual-to-nominal value transfer is then executed for the respective axes.

If the second limit is exceeded, the blinking error message **EXCESSIVE SERVO LAG IN <AXIS>** appears. The control-is-ready signal output is reset.
You cannot clear this message. You must restart the control to correct the error.

- In the machine parameters given below, define two range limits for position monitoring in each operating mode.
- Adjust the input values to the machine dynamics.

If blocked axes are the cause of the erasable error message **EXCESSIVE SERVO LAG IN <AXIS>,** a nominal velocity value may freeze, since the machine axes can no longer be moved:

- In MP1150.0, specify the time after which the nominal velocity value is to be deleted. After this time has expired, the actual position value is assumed as nominal position value. Before this time has expired, the error message cannot be cleared with the CE key. At this time the actual position value is assumed as nominal value, and the nominal velocity value is deleted.
- In MP1150.1, enter the time period for which the monitoring function is to remain off after the fast PLC input from MP4130.0 has been set. The monitoring functions reactivate after expiration of this time.
- In MP1150.2, specify the minimum time period after expiration of the time from MP1150.1 for which the monitoring functions should remain effective (e.g. if the input changes quickly).



MP1150 **Position monitoring**

MP1150.0 Delay time for deleting the nominal velocity value with the deletable error message: `Excessive servo lag in <axis>`

Input: 0 to 65.535 [s] Recommended: 0

MP1150.1 Time period for which the monitoring function is to remain off after the fast PLC input defined in MP4130.0 is set

Input: 0 to 65.535 [s]
0: Monitoring functions on
Recommended: 0.2 to 0.5

MP1150.2 Minimum time period for which the monitoring functions are to remain effective after expiration of the time from MP1150.1

Input: 0 to 65.535 [s]

MP1410.x **Position monitoring for operation with velocity feedforward control (erasable)**

Input: 0.0010 to 30.0000 [mm]
Recommended: 0.5 mm

MP1420.x **Position monitoring for operation with velocity feedforward control (EMERGENCY STOP)**

Input: 0.0010 to 30.0000 [mm]
Recommended: 2 mm

MP1710.x **Position monitoring for operation with following error (erasable)**

Input: 0.0000 to 300.0000 [mm]
Recommended: 1.2 · following error

MP1720.x **Position monitoring for operation with following error (EMERGENCY STOP)**

Input: 0.0000 to 300.0000 [mm]
Recommended: 1.4 · following error

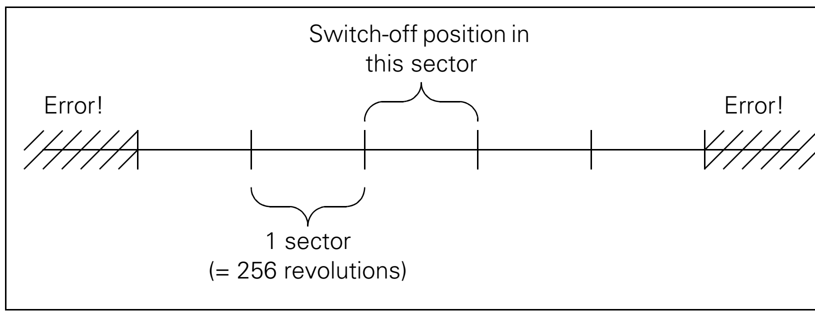
Difference between position at switch-on and shutdown

When the control is switched off, the actual position of the axes is saved with an absolute encoder. During switch-on it is compared with the position values read by the encoder.

If the positions differ by more than the difference defined in MP1146.x, a pop-up window appears with both positions. The new position must be confirmed with a soft key. If it is not confirmed, the error message `Check the position encoder <axis>` appears.

Special case: Absolute multiturn rotary encoder

The control stores an overflow (more than 4096 revolutions of the encoder) internally. Additionally, the number of traversed sectors (1 sector = 256 revolutions) is stored. After the drives are switched on, the current sector is compared to the stored sector.



If at switch-on the motor is more than two complete sectors away from the switch-off position (more than the sector after next), then immediately after the drives are switched on the `Switch-off pos. <axis> unequal ENDAT` error message appears.

Attention

The error must then be corrected!

After the control has been restarted, it is assumed that the number of revolutions is correct again.

The pop-up window may appear, stating that the positions at switch-on and shutdown differ by more than MP1146.x. If the motor is located at the correct position, you can confirm the message with the `YES` soft key.

MP1146.x Difference between the position at shutdown and the position read in via the EnDat interface

Input: 0.0000 to 300.0000 [mm] or [°]
0: No difference permitted

6.11.2 Nominal Speed Value Monitoring

Starting with the CC 424(B), nominal speed value monitoring is active as a rule, and applies to the following values:

- For axes:
  The maximum feed rate is 15% greater than the maximum motor speed (N-MAX) multiplied by MP1054.
- For spindle(s):
  The maximum spindle speed is 15% greater than the maximum motor speed (N-MAX) multiplied by the gear transmission ratio.

If monitoring responds, the error message **`88C0 Max. nominal motor speed %s exceeded`** is output and an EMERGENCY STOP reaction is initiated.

When using an EcoDyn motor, the error is triggered when the maximum permissible voltage is exceeded.

6.11.3 Movement Monitoring

Movement monitoring is possible during operation both with velocity feedforward and with following error.

During movement monitoring, the actual path traveled is compared at short intervals (several servo cycles) with the nominal path calculated by the NC. If during this period the actual path traveled differs from the calculated path, the flashing error message **MOVEMENT MONITORING IN <AXIS>** appears.

Analog axes:
An existing offset during a standstill may cause a potential at the analog output without any resulting positioning movement:

- In MP1140.x, enter a threshold from which the movement monitoring should go into effect.

Digital axes:
There is no offset.

- In MP1140.x, enter the speed from which the movement monitoring should go into effect.

For digital axes, in addition to the comparison of actual and nominal values, the calculated position from the pulses of the position encoder are compared with the pulses of the speed encoder:

- Enter in MP332.x the number of signal periods and in MP331.x the path for the number of signal periods (*** "Encoders" on page 648 ***).
- Enter the distance per motor revolution in MP1054.x. A formula can also be entered in MP1054.x.
- In MP1144.x, enter a limit value for this position difference. If you are not using a position encoder, you must enter 0 in MP1144.x as the position difference.

If the difference is greater than the input value from MP1144.x, the error message **MOVEMENT MONITORING IN <AXIS> B** appears.

Attention

If you enter the maximum value in MP1140.x or MP1144.x, no movement monitoring is active.
Safe machine operation is not possible without the movement monitoring function.

MP1140.x	**Threshold above which the movement monitoring functions**
Input:	Analog axes:0.030 to 10.000 [V] Digital axes:0.030 to 10.000 [1000 min^{-1}] Recommended: 0.030 [1000 min^{-1}]
MP1054.x	**Distance of a motor revolution (mm °)**
Input:	Analog axes: Nonfunctional Digital axes: Entry of a formula possible, See "Special case: Entering a formula" on page 525
MP1144.x	**Motion monitor for position and speed**
Input:	Analog axes: Nonfunctional Digital axes: 0 to 99 999.999 [mm] 0: No monitoring

6.11.4 Standstill Monitoring

Standstill monitoring is effective during operation both with velocity feedforward and with following error, as soon as the axes have reached the positioning window.

If the position difference is greater than the value defined in MP2800.x, the blinking error message **STANDSTILL MONITORING IN <AXIS>** appears. The message also appears if, while moving to a position, an overshoot occurs that is larger than the input value in MP1110.x, or if the axis moves in the opposite direction when beginning a positioning movement:

- In MP1110.x, enter a threshold from which the standstill monitoring should go into effect.

MP1110.x **Standstill monitoring**
Input: 0.0010 to 30.0000 [mm]

6.11.5 Positioning Window

The positioning window defines the limits within which the control considers a position to have been reached. After the position has been reached, the control begins running the next block. The position controller can correct a disturbance inside this window without activating the "Return to the Contour" function.

- In MP1030.x, define the size of the positioning window.

MP1030.x **Positioning window**
Input: 0.0001 to 2.0000 [mm]

Axes in position

Once the axes have moved into the positioning window, the corresponding bits are set in W1026. This also applies to the status after the machine control voltage is switched on. Axes that are not used are considered to be in position.

The NC resets the bits as soon as you start a positioning movement or traverse the reference marks.

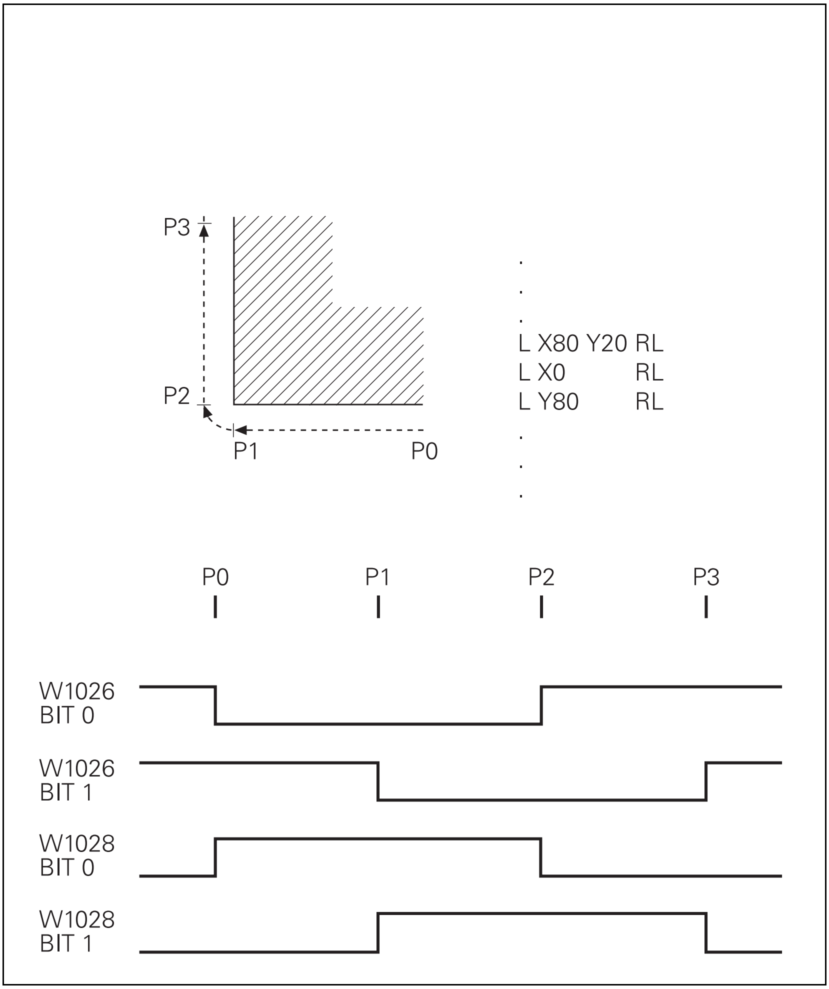
In the **Electronic Handwheel** mode of operation the bit for the current handwheel axis is reset.

W1026 is not set for contours that can be machined with constant surface speed.

		Set	Reset
W1026	**Axes in position** Bits 0 to 8 represent axes 1 to 9 0: Axis not in positioning window 1: Axis in positioning window	NC	NC

Axes in motion During axis movement, the NC sets the corresponding bits in W1028.

		Set	Reset
W1028	**Axes moving** Bits 0 to 8 represent axes 1 to 9 0: Axis not in motion 1: Axis in motion	NC	NC



6.11.6 Monitoring of the Power Supply Unit

The rectified supply voltage of the power supply unit is monitored. The supply voltage must lie within a defined range.
(400 V +/– 10%). If this is not the case the power supply unit reports an AC fail (PF.PS.AC).

At the same time, the dc-link voltage is monitored:

- If approx. 760 V– (UV 120, UV 140, UV 150, UR 2xx: approx. 800 V) is exceeded, the NC revokes the pulse release (reset) for the IGBT of the power module. The motors coast out of loop to a stop. No energy is returned to the dc link.
- If the dc-link voltage falls below approx. 385 V– (UV 120, UV 140, UV 150. UR 2xx: approx. 410 V), the power supply unit reports a powerfail (signal PF.PS.ZK)
- If the dc-link voltage falls below approx. 155 V– (UV 120, UV 140, UV 150, UR 2xx, UV 105: approx. 200 V), the control is reset (signal RES.PS).
- Below approx. 135V– (UV 120, UV 140, UV 150, UR 2xx, UV 105: approx. 180 V), the power supply unit switches off.

The UV 105 power supply unit reports a powerfail if the dc-link voltage is < approx. 385 V and the supply voltage is < approx. 330 V.

- With MP2150 you define which inverter signal is to trigger the **Powerfail** on the control.

Inverter signal	Meaning
AC fail (PF.PS.AC)	Failure of supply voltage for inverter
Power fail (PF.PS.ZK)	DC-link voltage failure

Since the AC fail is reported to the control before the powerfail, the control has more time to react to the subsequent dc-link voltage failure.

Note

Only certain HEIDENHAIN power supply units provide the AC-fail signal (see the Technical Manual for "Inverter Systems and Motors"). If you are using power supply units that do not provide this signal, you must not select the AC-fail signal in MP2150.

If a power fail is triggered on the control, all drives are brought to a controlled stop. The PLC outputs are switched off and the control freezes to ensure that the hard disk can no longer be accessed.

The control must be turned off and on again.

MP2150 **Powerfail signals on the control**
Input: 0: AC fail
1: Power fail and AC fail
2: Reserved
3: Powerfail
MP2150 can also be overwritten by the PLC and the LSV2 protocol.

Module 9167 Monitoring of dc-link voltage

With this module you can switch the dc-link voltage monitoring for powerfail (U_Z <approx. 385 V or 410 V) on and off.

If you don't call the module during the first PLC run-through, the supply voltage monitoring is automatically started after the first PLC run-through.

Call:

PS	B/W/D/K	<>Command code> 0: DC-link voltage monitoring off 1: DC-link voltage monitoring on
CM	9167	
PL	B/W/D	<>Error code> 0: Command executed –1: Transferred parameter invalid

Error recognition:

Marker	Value	Meaning
M4203	0	DC-link voltage monitoring on or off
	1	Error code in W1022
W1022	2	Transferred parameter invalid

6.11.7 Temperature monitoring

Temperature of the MC 42x(B)

The internal temperature of the MC 42x(B) is continuously monitored. At about 55 °C the temperature warning **TNC temperature warning** appears. If the temperature does not fall below 55 °C, the warning is reactivated after two minutes. Beginning at about 60 °C the error message **TNC temperature too high <temperature> °C** appears and an emergency stop is triggered. If the temperature does not fall below 60 °C when the machine is switched on again, the error message reappears after 10 to 20 seconds.

The temperature of the MC 42x(B) can be found with Module 9133.

Module 9133 Output of hardware information

Call:

```
PS   B/W/D/K  <>Mode>
              0: Internal temperature sensor in [°C]
              1: Temperature CPU1 (basic PCB) in [°C]
              2: Temperature CPU2 (additional PCB) in [°C]
              3: Voltage of buffer battery in [mV]
              4: 5-V supply voltage of main board
              5: 3.3-V supply voltage
              6: Rotational speed of the housing fan
CM   9133
PL   B/W/D    <>Value>
```

Error recognition:

Marker	Value	Meaning
M4203	0	Value was determined
	1	Error code in W1022
W1022	2	Invalid number given
	8	No second CPU present (for code 2)

Motor temperature

To measure the motor temperature, a KTY 84 must be connected at pins 13 and 25 of X15 to X20, X80 to X83. The temperature value is ascertained at least once per second. The maximum permissible motor temperature is taken from the motor table.

As soon as the given temperature is exceeded, the blinking error message **MOTOR TEMPERATURE <AXIS> TOO HIGH** appears and the drives are automatically switched off.

Module 9165 Sample the current motor temperature
Appropriate measures can be taken before the motor reaches the maximum temperature.

Call:

PS	B/W/D/K	<>Axis> 0 to n: Axes 1 to n+1 15: Spindle
CM	9165	
PL	B/W/D	<>Temperature> Range: 0 to 255 °C

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Control has no current controller

Temperature of the power module's heat sink

At X51 to X60 the temperature warning signal is available at pin 10a.

If the permissible temperature of the heat sink on the power module is exceeded, this signal is reset.

Attention

To avoid destroying the power module, the drives must be brought immediately to a standstill after a temperature warning.

Data on maximum permissible temperatures are available from the manufacturer of your power module.

The TEMPERATURE WARNING signal is not evaluated in the NC.

- Use Module 9160 or 9066 to interrogate the temperature warning, and take appropriate measures.

6.11.8 Internal power supply and housing fan

Via the PLC you can capture and evaluate the current values of the internal power supply and the speed of the housing fan.

The permanent speed monitoring (speed > 1500 min^{-1}) of the housing fan of the MC/CC can be switched off with MP4020 bit 13.

Module 9133 Output of hardware information

Call:

PS B/W/D/K <>Mode>
- 0: Internal temperature sensor in [°C]
- 1: Temperature CPU1 (basic PCB) in [°C]
- 2: Temperature CPU2 (additional PCB) in [°C]
- 3: Voltage of buffer battery in [mV]
- 4: 5-V supply voltage of main board in [mV]
- 5: 3.3-V supply voltage in [mV]
- 6: Rotational speed of the housing fan in [min^{-1}]

CM 9133

PL B/W/D <>Value>

Error recognition:

Marker	Value	Meaning
M4203	0	Value was determined
	1	Error code in W1022
W1022	2	Invalid number given
	8	No second CPU present (for code 2)

MP4020 PLC functions

Format: %xxxxxxxxxxxxxx

Input: Bit 13 – Monitoring the housing fan of the MC/CC
0: Monitoring active
1: Inactive

6.11.9 I^2t monitoring

General information

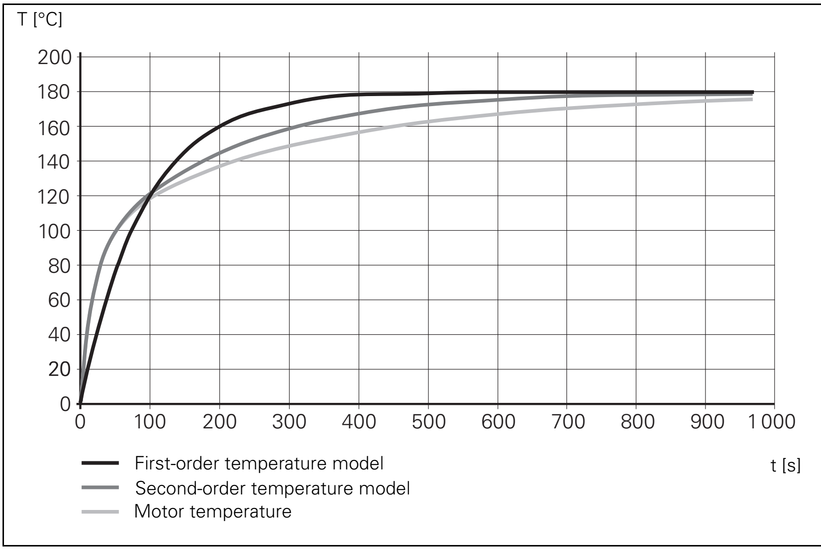
HEIDENHAIN inverter systems feature individual I^2t monitors, one for each power module and motor.

Function

An I^2t monitor calculates and supervises the temperature pattern in a thermal motor or power-stage model during operation.

The active current, the rated or stall current, (multiplied by MP2302.x for motors and by MP2304.x for power modules) and a device-specific temperature model are the basis for calculation. A first-order temperature module is available for monitoring power modules, first and second-order modules are available for motors. These modules make it possible to permanently calculate the temperature of the stator winding in the motor or the semiconductor in the power module.

Temperature model in an example comparison (motor)



The I²t monitor responds if this calculated temperature exceeds a certain limit.

Because temperature increase and heat dissipation are uneven when the motor is stationary or moving slowly, the I²t monitor distinguishes between standstill and traversing mode.

This limit range is defined in a motor table or power module table The following entries are important:

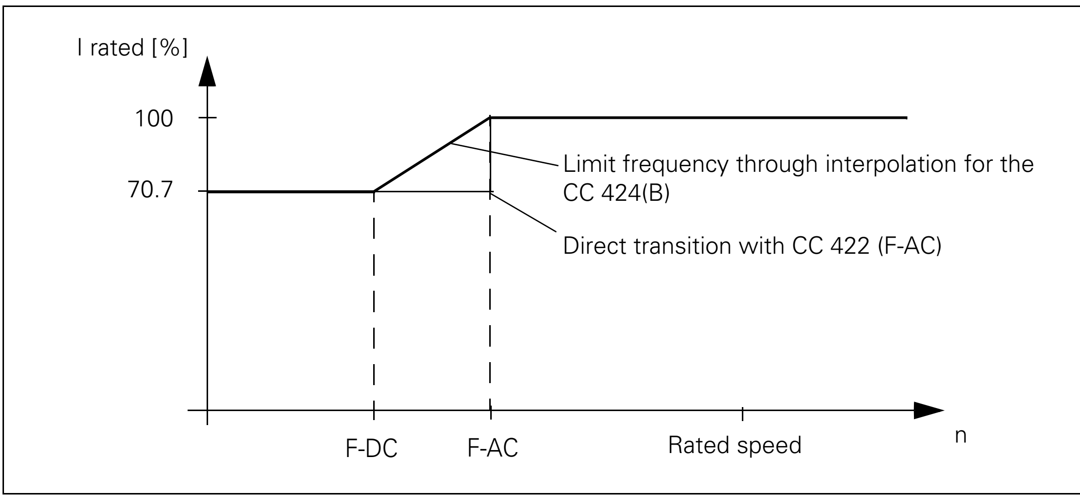
- F-AC **`(transition frequency in traversing mode [Hz])`**
- F-DC **`(transition frequency at standstill [Hz]; only CC 424(B))`**

Fundamentals

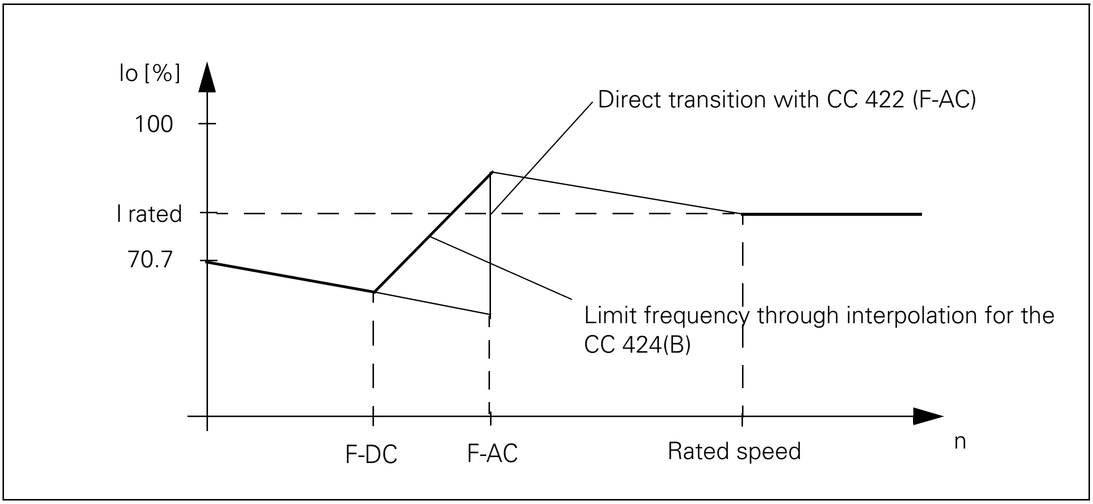
The following graphics illustrate these parameters in relation to the reference voltage. Remember here that CC 422 and CC 424(B) may have different parameters.

For the CC 422, no difference is made between F-DC and F-AC. Instead, F-AC is used as a rigid limit frequency for the transition between standstill and traversing mode. With the CC 424(B) it is possible to use an interpolated current range for the transition from standstill to traverse. This allows a more exact calculation of the temperature model.

If there is no stall torque value given in the motor table, the following model of current (with respect to the rated current) is used to calculate the temperature in the motor. The factors for MP 2302.x and MP 2304.x are not yet taken into account.



If the stall current value is given in the motor table, the following model of current (with respect to the stall current) is used to calculate the temperature in the motor. This is only used for synchronous motors, however. For asynchronous motors the above model of current applies, which is used if no stall current is given. For synchronous motors, the factors from MP2302.x and MP2304.x are not yet taken into account in the following description.



Commissioning and evaluation

- In MP2302.x, enter the factor for the I^2t monitoring of the motor. The input value is a factor for the reference current (1 = 100% of the motor's standstill current or rated current). If you enter zero, the I^2t monitoring for the motor (not for the power module) is switched off. If you enter a value less than 1, then a motor standstill current or rated current (reference current) reduced by this factor is used for calculation. The actual current is rated higher by this factor relative to the reference current, and the I^2t value is increased more quickly. This means that the results for values less than 1 provide increased protection for the motor. Values greater than 1 provide less protection for the motor.
- In MP2304.x, enter the factor for I^2t monitoring of the power module. The input value is a factor of the power module's rated current (1 = 100%). If you enter zero, the I^2t monitoring for the power module (not for the motor) is switched off. If you enter a value less than 1, then a power module current or rated current (reference current) reduced by this factor is used for calculation. The actual current is rated higher by this factor relative to the reference current, and the I^2t value is increased more quickly. This means that the results for values less than 1 provide increased protection for the power module. Values greater than 1 provide less protection for the power module.
- All required entries for calculation of a temperature model have to be available in the motor table or power module table. See "Temperature models"on page 917.
- Use Module 9160 to interrogate the I^2t monitoring (See "Interrogation through PLC module" on page 916).

Limit values

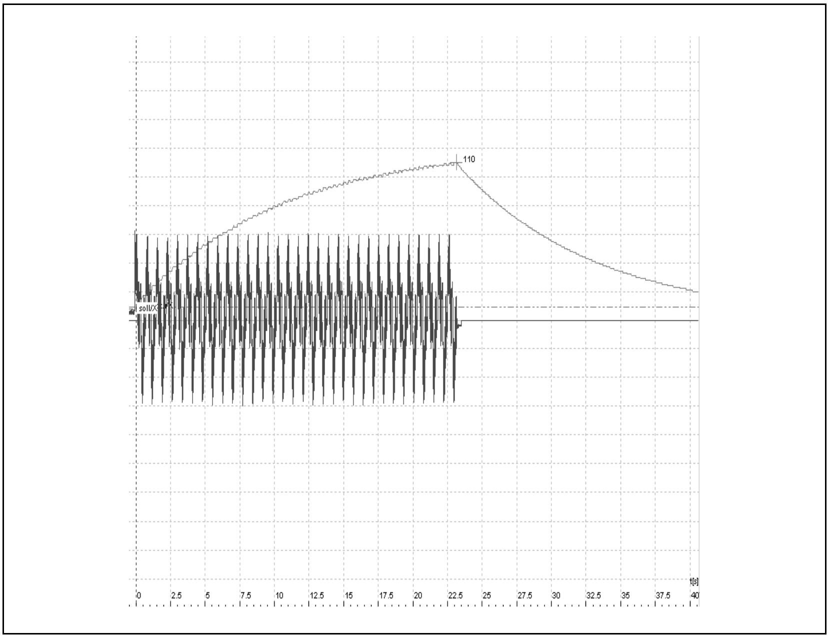
The limit values for the I^2t value (dimension for the permissible temperature in the device [%]) are handled by the NC side of the control and are composed of the following:

- Value exceeds 100%: An axis-specific I^2t early warning is sent to the PLC (for evaluation and possible countermeasures such as reduction of the feed rate with error message by PLC program with the aid of PLC Module 9160). If the value does not exceed 110% and falls below 90%, the axis-specific early warning is reset.
- 110%: An NC stop is triggered and the drives are switched off.

Note

In the oscilloscope you can display the current value of the I^2t monitoring of the motor and power module, as well as the current load of the drive.

Motor overload with **I^2t monitoring**



Machine parameters

MP2302.x **Factor for I^2t monitoring of motor**

Input: 0 to 1000.000 [· rated or stall current of the motor]
0: I^2t monitoring of motor switched off
1: Rated or stall current is reference value

MP2304.x **Factor for I^2t monitoring of the power module**

Input: 0 to 1000.000 [· rated current of power module]
0: I^2t monitoring of power module switched off
1: Rated current of power module is reference value

Interrogation through PLC module

Module 9160 Status request from temperature monitoring and I^2t monitoring

The I^2t monitoring reported by the module is given with respect to the first I^2t monitor response (power stage or motor), if both I^2t monitors are activated (MP2302.x, MP2304.x). This early warning is withdrawn as soon as the limit for reset is reached. For the response behavior, See "Limit values" on page 915.

Call:
CM 9160
PL D <>Temperature monitoring>
Bit 15876543210
Axis:Sxxxxx987654321
PL D <> I^2t monitoring I^2t early warning>
Bit 15876543210 15 876543210
Axis Sxxxxx987654321 Sxxxxx987654321

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Control has no current controller

Temperature models

The temperature model of the motor or power module is defined by the entries in the motor table or power module table, respectively (motor.mot, motor.amp). Remember that calculation of which temperature model to use depends exclusively on the availability or nonavailability of the parameters. In addition, the parameters for motors and power modules are to be evaluated separately.

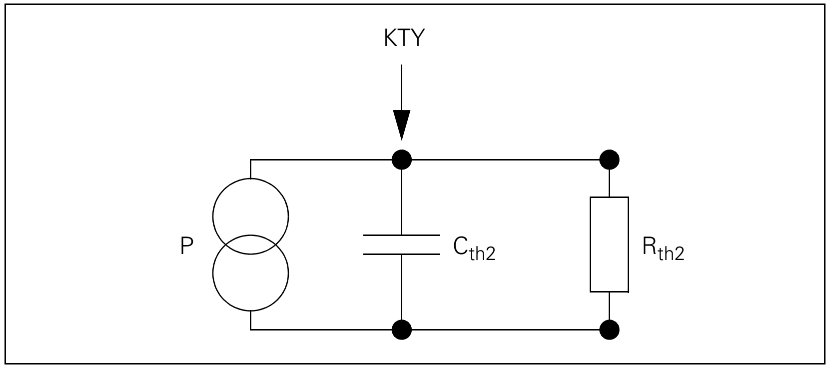
These parameters are explained below using the respective temperature model for the calculation.

Temperature model, first order

The following values (entries in the motor table or power module table) are required for the first-order temperature model to calculate the temperature.

- **F-DC** [Hz]:
  This parameter is not evaluated for the CC 422.
  Lower limit frequency for the transition of traverse to standstill with the CC 424(B).
  F-DC = 0 – Default value (0) is active
  F-DC > 0 – Input value in Hz is active
- **T-DC** [s]:
  Thermal time constant for operation at standstill (not evaluated at present)
- **F-AC** [Hz]:
  Upper limit frequency for the transition from standstill to traverse.
  F-AC = 0 – Default value (0) is active
  F-AC > 0 – Input value in Hz is active
- **T-AC** [s]:
  Thermal time constant for the motor or power stage. Identifies the point in the temperature curve at which 63% of the maximum temperature is reached.
  T-AC = 0 – Default value: 10 s for axes, 150 s for ball screw
  T-AC > 0 – Input value [s] for power modules. For motors, this input value is active if Tth_2 = 0.
- Only for motors $\mathbf{Tth_2}$ [s]:
  Thermal time constant for the motor. Identifies the point in the temperature curve at which 63% of the maximum temperature is reached.
  Tth_2 = 0 – Default value: 10 s for axes, 150 s for ball screw
  Tth_2 > 0 – Input value in [s] for motors

First-order temperature model of the motor



- P: Heat output of the three phases
- KTY: KTY temperature sensor in the winding
- C_{th2}: Thermal capacity of the motor housing
- R_{th2}: Thermal resistance on the motor housing
- T_{th2}: Thermal time constant $R_{th2} \cdot C_{th2}$

Temperature model, second order

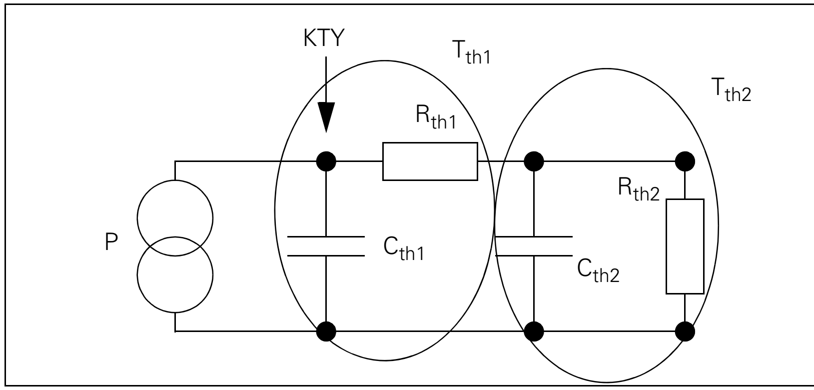
The following values (entries in the motor table or power module table) are required for the second-order temperature model to calculate the temperature (default values [axis/spindle] are valid for the entry "0"):

- **F-DC** [Hz]:
  This parameter is not evaluated for the CC 422.
  Lower limit frequency for the transition of traverse to standstill with the CC 424(B).
  F-DC = 0 – Default value (0 Hz) is active
  F-DC > 0 – Input value in Hz is active
- **T-DC** [s]:
  Thermal time constant for operation at standstill (not evaluated at present)
- **F-AC** [Hz]:
  Upper limit frequency for the transition from standstill to traverse.
  F-AC = 0 – Default value (0 Hz) is active
  F-AC > 0 – Input value in Hz is active
- **Tth_1** [s]:
  Thermal time constant for the transition from winding to housing
  Tth_1 = 0 – Default value (0 s) is active
  Tth_1 > 0 – Input value in [s] is active
- **Rth_1** [K/W]
  Thermal resistance for the transition from winding to housing.
  Rth_1 = 0 – Default value: 0 K/W
  Rth_1 > 0 – Input value in [K/W] is active
- **Tth_2** [s]:
  Thermal time constant for the transition from housing to coolant
  Tth_2 = 0 – Default value: 10 s for axes, 150 s for ball screw
  Tth_2 > 0 – Input value in [s] is active
- **Rth_2** [K/W]:
  Thermal resistance for the transition from winding to coolant
  Rth_2 = 0 – Default value: 0 K/W
  Rth_2 > 0 – Input value in [K/W] is active
- When the CC starts up, the current motor temperature (KTY sensor) is taken into the calculation model in order, for example, to compensate any excessive temperatures.

Note

- All parameters have to be entered for the model to become active. If a parameter is missing, the first-order temperature model becomes active, either with the thermal time constant "Tth2" or with "T-AC."

Second-order temperature model of the motor



- P: Heat output of the three phases
- KTY: KTY temperature sensor in the winding
- C_{th1}: Thermal capacity of the winding
- C_{th2}: Thermal capacity of the housing
- R_{th1}: Thermal resistance winding/housing
- R_{th2}: Thermal resistance housing/coolant
- $T_{th1} = R_{th1} \cdot C_{th1}$
- $T_{th2} = R_{th2} \cdot C_{th2}$

Compatibility

Old motor tables of the iTNC530 are also usable in newer software versions. If the columns/parameters in the temperature models are missing, however, it is of course impossible to calculate a second-order temperature model.

In such a case the entries **F-DC, T-DC, F-AC, T-AC** are used for a first-order temperature model. If this model, too, has no entries (entries "0"), the default values of the above temperature models apply.

6.11.10 Momentary utilization of drive motors

Module 9166 provides the momentary utilization of the given drive motor as a percentage value.

Utilization means:

Speed range	**$n_{act} < n_N$**	**$n_{act} \geq n_N$**
Asynchronous or synchronous motor	$\frac{\lvert M_{curr} \rvert}{\lvert M_{rated} \rvert}$	$\frac{\lvert P_{curr} \rvert}{\lvert P_{rated} \rvert}$

If $n_{act} = n_N$, then:

$$\frac{\lvert M_{curr} \rvert}{\lvert M_{rated} \rvert} = \frac{\lvert P_{curr} \rvert}{\lvert P_{rated} \rvert}$$

M_{curr} and P_{curr} are cyclically determined by the current controller. This determining is based on the equivalent circuit data of the motor.

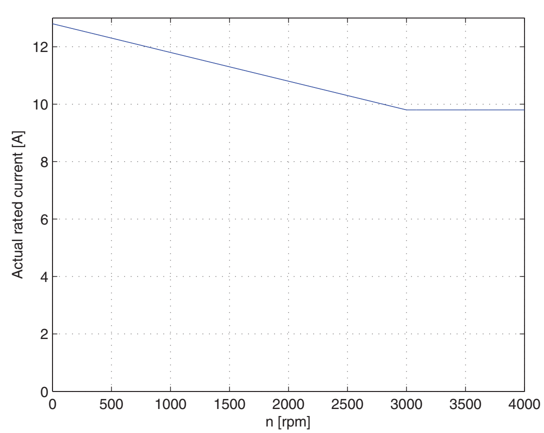
P_{rated} = P-N from motor.mot

$$M_{rated} = \frac{P_{rated}}{2 \cdot \pi \cdot \frac{n_{rated}}{60}}$$

Consideration of I_0 when using synchronous motors:

If I_0 differs too significantly from I_{rated}, then the utilization display can be adapted via MP2312.x. In this case you must set MP2312.x = I_{rated}/I_0.

Example for the behavior of I_0 and I_{rated} with an OSY155F-EcoDyn motor:



In the diagram you can see that:

I_0 = 12.8 A, I_{rated} = 9.8 A, n_{rated} = 3000 rpm

For the synchronous motor:

$M_{curr} = kT \cdot I_{curr}$

$M_{rated} = kT \cdot I_{rated}$

$M_{rated} = f(n)$

Module 9166 Momentary utilization of the drive motor

The evaluation through MP2312.x is already calculated in the utilization of the drive motor.

Call:

PS	B/W/D/K	<>Axis> 0 to n: Axes 1 to n+1 15: Spindle
CM	9166	
PL	B/W/D	<>Utilization of the drive in %>

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Control has no current controller

MP2312.x Factor for utilization of motors

Input: 0 to 1 000.000
0: Factor = 1

6.11.11 Determining the current torque of a drive

With Module 9170 you can determine the averaged, maximum and minimum torque of a drive.

The information about the torque utilization always refers to the values determined between two PLC cycles. The momentary torque is returned in tenths of per cent of the nominal torque.

Example:

For the first axis (e.g. X axis and QSY 190D EcoDyn with M_N of 23.0 Nm), an averaged torque of 281 is determined via Module 9170.

$$6.46\ \text{Nm} = \frac{23.0\ \text{Nm}}{1000} \times 281$$

$$M_{act} = \frac{M_{rated}}{1000} \times \text{Value}_{Modul9170}$$

Module 9170 Finding the current torque

Call:

PS B/W/D/K <>Axis>
0 to 13 and 15: Axes 1 to 14 and the spindle
PS B/W/D/K <>Mode>
0: Torque value in percent of the nominal torque
1: Torque value in tenths of per cent of the nominal torque
CM 9170
PL B/W/D <>Averaged torque value>
PL B/W/D <>Maximum torque value>
PL B/W/D <>Minimum torque value>

Error recognition:

Marker	Value	Meaning
M4230	0	Torque ascertained
	1	Error code in W1022
W1022	1	Invalid mode
	2	Invalid axis number

6.11.12 Status of HEIDENHAIN inverters

With Module 9066, the status information of the HEIDENHAIN supply unit can be determined.

Module 9066 Status of HEIDENHAIN supply unit

Call:

PS B/W/D/K <>Code for hardware components>
0: HEIDENHAIN inverter
1: Not available
2: I^2t early warning
3: I^2t monitoring
4: Error of analog part
CM 9066
PL B/W/D <>Status information>
Code 0: HEIDENHAIN inverter
Bit 0: Not used
Bit 1: dc-link voltage too high
Bit 2: Heat sink temperature too high
Bit 3: Short-circuit of a motor phase with Uz
Bit 4: dc-link current too high
Bit 5: Power supply unit not ready
Bit 6: Leakage current too high
Code 1: Not available
Codes 2 to 4: Axis masks

Error recognition:

Marker	Value	Meaning
M4203	0	Status ascertained
	1	Error code in W1022
W1022	2	Invalid code
	24	Module was called in a spawn or submit job

The HEIDENHAIN power supply units have several status signals which lead to error messages on the control. MP2195 is used to suppress the error message for each status signal.

HEIDENHAIN does not recommend suppressing the error messages from the power supply units. If you are using a UE 2xx, the signals must be suppressed because the UE 2xx compact inverter does not provide these signals.

Status signals that are already inactive during control power-up

The handling of status signals from HEIDENHAIN power supply units, which are already inactive during control start-up, varies depending on MP2195 bit 0:

MP2195 bit 0 = 0: Missing signals cannot be detected with Module 9066 and do not result in an error message when the drive is switched on.

MP2195 bit 0 = 1: After the PLC program has been compiled, missing signals can be detected with Module 9066 and trigger an error message when the drive is switched on. Signals that are not provided by the power supply unit must be suppressed with MP2195 (bit 1 to bit 6), because non-existent signals are always identified as errors.

The SUPPLY.SPY table for the power supply modules contains the STATUS-SIG column for the status signals of the power supply module. This column informs you, among other things, whether the PF.PS.AC and PF.PS.DC is supported by the power supply module. This information in SUPPLY.SPY is evaluated by the CC 424(B) and higher.
Bits 7 and 8 of MP2195 enable you to deactivate the error messages that may be triggered by the evaluation of these signals (PF.PS.AC, PF.PS.DC), without changing the SUPPLY.SPY table.

Note

- Signals that change their status during operation are always identified as errors.
- Only as of CC 424(B): Errors of the supply module (e.g. excessive Uz or Uv temp.) lead to an NC stop (if M148 then LIFTOFF) with a subsequent emergency stop.

MP2195 **Handling of status signals from HEIDENHAIN power supply units**

Input: Bit 0 – Status signals that are already active during control power-up.
0: Missing signals are ignored
1: Missing signals are evaluated
Bit 1– $\overline{\text{ERR.UZ.GR}}$ signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 2 – $\overline{\text{ERR.TMP}}$ signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 3 – Reserved
Bit 4 – $\overline{\text{ERR.IZ.GR}}$ signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 5 – RDY.PS signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 6 – $\overline{\text{ERR.ILEAK}}$ signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 7 - PF.PS.AC
0: Previous behavior; Error message is not suppressed
1: Error message is suppressed
Bit 8 - PF.PS.DC
0: Previous behavior; Error message is not suppressed
1: Error message is suppressed
MP2195 can also be overwritten by the PLC and the LSV2 protocol.

6.11.13 Controlling the motor brakes

The motor brakes are controlled with the $\overline{\text{BRK}}$ braking signal, which is transmitted to the **HEIDENHAIN inverters** via the PWM interface (X51 to X62). The corresponding outputs are activated there. See the basic circuit diagrams.
Control of the motor brakes via the PWM interface must be deactivated for non-HEIDENHAIN inverters that do not support this function:

- Set bit 0 = 1 in MP2234.x

The motor brakes are opened no later than 50 ms after the speed controller is switched on. For safety reasons, the controller is not switched off until the braking signal has been output:

- Enter in MP2308.x the time (overlap time) after which the controller is to be switched off (after the braking signal has been output).

If the inverter sends the $\overline{\text{RES.PS}}$ reset signal, then the $\overline{\text{BRK}}$ braking signals are output immediately upon switch-off of the controllers, i.e. without any overlap time.

Activated brakes cause a change in the controlled system. The motor with the changed controlled system is controlled during the overlap time. This can lead to oscillations when the controller is switched off. These oscillations are suppressed by the NC software. MP2220 bit 3 can be used to deactivate the suppression of vibrations. HEIDENHAIN does not recommend switching off the suppression of the oscillations.

In MP2309.x you can define a time period in which the speed and position controller parameters are adjusted to values for controlling a closed brake when the drive is switched on. This parameter can be used to avoid oscillations in the drive during switch-on when the brake is still closed and the controller is already active.

MP2220 **Monitoring functions**
Input: Bit 3 – Switching off the controller when the motor brakes are activated
0: Suppress oscillations
1: Vibrations are allowed

MP2234.x **Internal triggering of the motor brakes via the PWM interface**
Format: %xx
Input: Bit 0 –
0: Signal is transmitted
1: Signal is not transmitted
Bit 1– reserved

MP2308.x **Time between output of the braking signal $\overline{\text{BRK}}$ and switching off of the controller (overlap time)**
Input: 0.001 to 5.000 [s]
0: 0.200 s

MP2309.x **Controller parameters adjusted to closed brake**
Input: 0: Previous behavior; not active
0.001 to 5.000 [s]

Automatic test of the motor brakes at switch-on

You can carry out an automated functional test of the motor brake after switching on the drive, either before traversing the reference mark or through activation by PLC Module 9143.

This brake test only functions in combination with HEIDENHAIN inverter systems and only when using the brake output on X392/X393 if it is wired according to the basic circuit diagram from HEIDENHAIN.

For the period of one second, a torque (current) is applied while the brake is active. The path that the axis has moved is then measured. If the permissible path is exceeded, the error message **`8130 Motor brake defective <axis>`** appears, and the axis remains controlled. The test is carried out simultaneously for all affected axes.

Attention

In case of an error, the axis must be moved to a safe position, and physically supported, if necessary. Only then may the machine be switched off so that the defect can be corrected.

If no motor current flows while testing the motor brakes, the error message **`8140 No current for brake test <axis>`** appears.

- Enter in MP2230.x a factor for the motor stall current with which the motor brake test is to be carried out. If the test is not to be carried out, or for motors without brakes, enter MP2230.x = 0.

The reference value for the factor from MP2230.x is the stall current **`I0`** entered in the motor table. If **`I0`** in the motor table equals 0, then the rated current **`I-N`** from the motor table is used.

Recommended input value for MP2230.x:

$$\text{MP2230.x} \geq 1{,}3 \cdot \frac{M_L}{M_0}$$

M_L: Maximum load torque of the axis. In a standard case the holding torque of a vertical axis is used here. It is to be ensured via activation of the brake that a vertical axis does not fall down when the drive controllers are switched off.
M_0: Standstill torque of the motor

Always keep in mind:

- Torque for motor test ≥ 1.3 · maximum load torque of the axis
- Stall torque of the motor ≥ maximum load torque of the axis
- Holding torque of the motor brake ≥ torque for the motor test

Note

- Please note that when reading the current via the internal oscilloscope, on the CC 422 you are seeing the peak value, and on the CC 424(B) you are seeing the effective value of the current.
- Please note that the test torque can only be generated with a certain factor of uncertainty. Factors of influence here are the accuracy of the current sensors and the torque constant of the motor.

- Enter in MP2232.x the permissible path that the motor is allowed to move against the brake. If the test is not to be carried out, or for motors without brakes, enter MP2232.x = 0.

MP2232.x must be < MP1110.x so that the standstill monitoring does not activate!

Recommended input value for MP2232.x:

$$MP2232.x = 2 \cdot \alpha \cdot \frac{MP1054.x}{360°}$$

α: Permissible braking angle: Backlash of the motor brake as per the manufacturer specifications (for HEIDENHAIN motors, $\alpha \leq 1°$)

Example:
QSY 155B-EcoDyn: M_0 = 13 Nm, M_{Br} = 40 Nm
M_L = 11 Nm

$$MP2230.x \geq 1{,}3 \cdot \frac{11\ Nm}{13\ Nm} = 1{,}1$$

MP1054.x (distance per motor revolution) = 20 [mm]
$\alpha = 1°$

$$MP2232.x = 2 \cdot 1° \cdot \frac{20\ mm}{360°} = 0{,}111\ mm$$

MP2230.x	**Multiplier for motor current during test of motor brake**
Input:	0.100 to 30.000 [· motor stall current] 0: No test of motor brakes, or motor without brake Recommended: 1.3 · M_L / M_0
MP2232.x	**Maximum permissible path during test of motor brakes**
Input:	0 to 10.0000 [mm] or [°]

Module 9143 Activate the brake test

With this module an axis-specific brake test with the configuration from the machine parameters or with other values for machine parameters 2230 and 2232 can be started.

Constraints:

- Programming it in a submit job blocks other submit jobs until the test is completed.
- The PLC module itself gives the processing time to other spawn and submit processes.

Call:

PS	K/B/W/D	<>Axis number> 0 = 1st axis, 2 = 2nd axis, etc.
PS	K/B/W/D	<>Multiplier for rated current> Value in 1/1000 or 0: Default MP2230 (factor of nominal current)
PS	K/B/W/D	<>Permissible traverse path> Value in 0.1 [um] or 0: Default MP2232
CM	9143	
PL	B/W/D	<>Status/Error> 0: Brake OK 1: Brake defective 2: Invalid axis or negative values for rated current or traverse path 3: Call during running NC program or during other PLC jobs 4: Call was made from a cyclic PLC program 5: Error during data exchange 6: Not allowed for safe control 7: Drive not ready 8: Brake test was aborted (e.g. by emergency stop)

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis programmed (invalid axis number, not a closed-loop axis, axis currently open-loop axis or slave axis) or negative values for current or traverse path are programmed
	8	Module is not allowed for control with Functional Safety
	20	Module was not called in a spawn job or submit job
	21	Call during program run or during other active PLC jobs for the programmed axis
	40	Drive not ready
	45	Canceled due to error during data exchange or due to external influences (e.g. emergency stop)

Module 9048 Interrogating the operating states of axes

Module 9048 is used to interrogate the operating state of a certain axis or for all axes together.

Call:

```
PS    B/W/D/K   <>Axis number>
                Axis number: Individual information for a programmed axis
                –1: Information for all axes, bit-coded as axis mask
PS    B/W/D/K   <>Status information>
                0: Brake test active/inactive
                1: Free rotation active/inactive
CM    9048
PL    B/W/D     <>Status>
                Interrogation of an individual axis: 1/0 = active/inactive
                Interrogation of all axes: Bit-coded axis mask
```

Error recognition:

Marker	Value	Meaning
M4203	0	Status ascertained
	1	Error code in W1022
W1022	1	Invalid value for status information
	2	Invalid axis programmed:
		If status information 0 was transferred, then this error occurs if an invalid axis number, an open-loop axis or an axis that is temporarily not a closed-loop axis was selected.
		If status information 1 was transferred, then this error occurs if an invalid axis number, an open-loop axis, an axis that is temporarily not a closed-loop axis, a slave axis or an NC axis that is not a rotary axis was selected.

6.11.14 EMERGENCY STOP monitoring

On the control there is a PLC output (X41/34) with the designation control-is-ready, and the associated PLC input for the MC (X42-/4; I3) with the designation control-is-ready-acknowledgement for the EMERGENCY STOP routine.

Internal emergency stop

If an **internal emergency stop** is triggered (e.g. due to standstill monitoring), the iTNC switches

- the control-is-ready output off (SH1A; responsible watchdog reacts after 10 ms at the latest)
- the inverter enables off (SH1B; responsible watchdog reacts after the time set in MP2172 (1 to 6 s), and the inverters are now without power.

An error message appears and the PLC program is stopped. Depending on the error class, it might be possible that it **cannot** be cleared with the CE key.

▶ Correct the error and restart the switch-on routine.

MP2172	**Delay for the SH1B signal (inverter enable)**
Input:	0 to 6 [s] as an integer 0: 3 [s] Default

External emergency stop via MC (I3)

If an **external emergency stop** is triggered via input I3 ("acknowledgment of control-is-ready signal"),

- the nominal speed value "null" is output via the MC, braking the drives on the intended braking ramp (usually at the limit of current),
- the `EMERGENCY STOP` error message is displayed, and
- markers M4177 and M4178 are set by the NC.

The PLC marker M4260 is set when input I3 "Control-is-ready acknowledgment" is reset (=0).

You can clear the error message with `CE` after switching the machine control voltage back on.

The input I3 ("control-is-ready signal acknowledgment") is passed directly onto the NC; it can **not** be manipulated by the PLC (I3).

Resetting the "control-is-ready signal acknowledgment" inputs leads to position monitoring being shut off for the time defined in MP1150.1, and to an actual-to-nominal value transfer. After the time defined in MP1150.1 has expired, position monitoring is again active, for at least the time defined in MP1150.2.

If marker M4580 is set, then instead of the external emergency stop ("control-is-ready signal acknowledgment" input), the control loops of all axes and of the spindle are opened, and an NC stop is performed.

		Set	Reset
M4177	**Clearable error message displayed**	NC	NC
M4178	**Error message EMERGENCY STOP is displayed**	NC	NC
M4580	**Suppress EMERGENCY STOP, open all position control loops, NC stop**	PLC	PLC
M4260	**Control-is-ready acknowledgement**	NC	NC

External emergency stop via CC (axis enables)

At the same time, HEIDENHAIN recommends using the "global axis enabling" of the controller unit (CC) via (X42/33; I32) or the axis-specific "axis enables" of the CC via X150/X151, which are also integrated in the **external emergency stop.** If these are switched off, then when there is an emergency stop of the controller unit (CC)

- the nominal speed value "null" is output, braking the drives on the intended braking ramp (usually at the limit of current)
- The pulses of the power stages are normally switched off after the braking process and the overlap time (MP2308). MP2173.x serves to monitor the braking of the drives. The monitoring time for the braking process is defined in MP2173.x. After the monitoring time has expired, the control checks whether the servo drive is at a standstill. If this is not the case, the control assumes that a serious error has occurred and switches off (via SH2) the pulses of the power stages. This ensures that, after a request to switch off the servo drives (e.g. EMERGENCY STOP, X150, PLC or alarm), the pulses are safely switched off (via SH2) at the latest after the time specified in MP2173.x expires. If the standstill is detected right before expiration of the time defined in MP2173.x, the active braking process is continued and the pulses are not switched off until after the overlap time.
  The time for switching off the pulses (entry in MP2173.x) must be greater than the maximum possible braking time of the axis/spindle that can occur through electrical braking. Especially for axes/spindles without mechanical braking, you must ensure that the pulses are not switched off until after the maximum possible braking time for the axis/spindle that can occur through electrical braking. Undecelerated axes/spindles coast to a stop after pulse switch-off. In the worst case, this can cause damage to the machine. Specific operating conditions of the machine, such as maximum feed rate, overload on the axes, etc., must also be taken into account.

Attention

The time for switching off the pulses (via SH2, entry in MP2173.x) must always be greater than the maximum possible braking time of the axis/spindle that can occur through electrical braking.
However, do not set too large a value in MP2173 so that the safety function of the machine parameter is still ensured.

MP2173.x	**Pulse switch-on of the power stage**
Input:	0.2 to 100.000 [s]
	0 = 3 seconds (default value)

Testing the internal emergency stop

For test purposes, an internal EMERGENCY STOP can be simulated in order to inspect the correct wiring of the machine. The control-is-ready output is reset. The NC and PLC are no longer operable.

Danger

Hanging axes must be supported before the test in order to prevent damage to the machine in case of error.

- Enter the code number **FAILTEST** under MOD.
- Acknowledge the message window with the YES soft key in order to carry out the test.

Connection diagram

In the event of an error, a drop-off of the control-is-ready output (X41/34) must trigger an emergency stop. The control therefore checks this output every time that line power is switched on.

If you use safety protection combinations, this may lead to problems if the EMERGENCY STOP test is also performed with these combinations. The reason for this is the recovery time of such assemblies after an emergency stop. In some cases it is greater than 200 ms.
In order to maintain these specifications, the "Control is ready" output (X41/34) remains off for the time in MP7630 after detecting the 0 level at the "control-is-ready signal acknowledgment" input (X42/4). This time for step 4 (see flowchart) can be configured in MP7630. This time is to be set so that the safety protection combinations can become ready again.

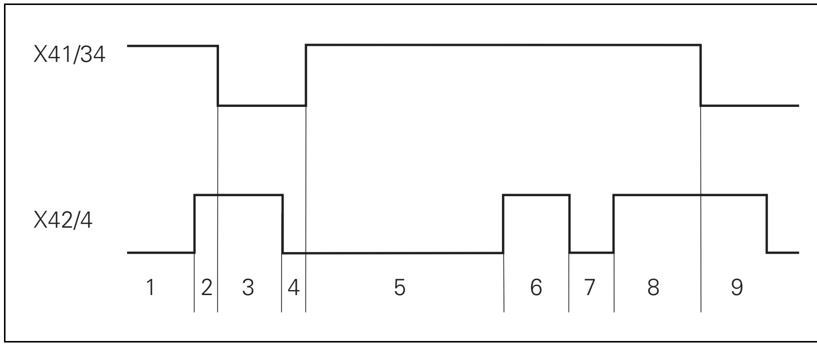
Note

The circuitry recommended by HEIDENHAIN is illustrated in the Basic Circuit Diagram.

Ensure that the control-is-ready acknowledgment occurs within 1 second.

MP7630	**Recovery time after EMERGENCY STOP test can be configured**
Input:	0: Previous behavior 1 to 999 [ms]

Flowchart



Step	Function	Screen display
1	Waiting for machine control voltage	**RELAY EXTERNAL DC VOLTAGE MISSING**
2	Recognition of the machine control voltage on X42/4 and switch-off of the control-is-ready signal on X41/34 by host computer (t < 66 ms)	
3	Maximum time within which the control-is-ready acknowledgment on X42/4 must go to zero (t < 1 s)	If exceeded, **EMERGENCY STOP defective**
4	Recognition of the acknowledgment and setting of X41/34 (t < 20 ms)	
5	Waiting for machine control voltage	**RELAY EXTERNAL DC VOLTAGE MISSING**
6	Normal control operation. Control voltage was switched on, control-is-ready output and acknowledgment are at "1".	
7	"Control-is-ready-acknowledgement" (I3) is switched off via external emergency off switch (set to "0"), for example	**EXTERNAL EMERGENCY STOP**
8	After switching the machine control voltage on again, the control operates normally.	
9	After detecting a fault, the control switches off the control-is-ready output (X41/34).	**INTERNAL EMERGENCY STOP**

As can be seen in the flowchart, after the control-is-ready output (X41/34) is reset by the external emergency stop circuitry, the signal at I3 (X42/4; "control-is-ready-acknowledgment") is also reset, in order to achieve a controlled standstill.

Repetition of the power-up test

One of the requirements of EN ISO 13849-1 is that the power-up test (emergency-stop test and braking test of the control) must be repeated within no more than 168 hours. It must be ensured by the PLC program that this requirement is met. However, a running NC program can be run to its end even after the 168 hours have passed if the guard doors have been and remain closed the entire time. However, this may only be done if the guard doors remain closed until the power-up test has been repeated! If the guard doors are open and the 168 hours have expired, the PLC program must trigger an error with an emergency stop reaction. Faulty behavior of the safety functions cannot be ruled out once the 168 hours have passed.

The emergency stop test can be started directly through the PLC module 9144, the braking test with PLC module 9143. The required timer for monitoring must be realized in the PLC program. When the control is booted after the main switch is on, the control automatically runs the power-up test.

The following PLC marker help you to realize the repeated power-up test:

- **M4189 – Emergency-stop test completed**
  The marker M4189 is set when the emergency-stop test of the control has been completed. The marker is also reset during the execution of PLC module 9144 and is not set again until the end of the emergency-stop test.
- **M4190 – Emergency-stop test active**
  The marker M4190 is set when the control is ready for the emergency-stop test after it has been started up. The marker remains set until the emergency-stop test has been completed. During the execution of PLC module 9144 the marker also remains set until the end of the emergency-stop test.
- **M4191 – Control is ready**
  The marker M4191 contains the information of the "control-is-ready" output. This information is given before the hardware output reaches the corresponding physical status (MC.RDY or STO.A.G).
- **M4192 – Request for "Control voltage On"**
  The marker M4192 is set in controls without Functional Safety (FS) if there is an NC software request for the control voltage to be switched on. You can then automatically connect the machine control voltage again through a PLC output and corresponding external circuitry.

		Set	Reset
M4189	**Emergency stop test completed**	NC	NC
M4190	**Emergency stop test active**	NC	NC
M4191	**Control is ready**	NC	NC
M4192	**Request for switching machine control voltage on**	NC	NC

Module 9144 Safety self-test / Emergency-stop test

PLC module 9144 is used to activate special functions regarding the safety self-test or emergency-stop test, as well as the Functional Safety (FS) of a HEIDENHAIN control system.
The test can be started directly through the PLC module. Also, a PLC soft key can be made available through the PLC program if all minimum requirements are fulfilled so that the user can start the emergency stop test directly by soft key.

It must be ensured by the PLC program that the following minimum requirements are met before the self-test is started:

- All guard doors must be closed and, if possible, locked.
- No active machining operation is allowed.
- All servo drives must be switched off.

Further constraints may apply, depending on the control model used:

- **Safety self-test and emergency-stop test for controls with or without HSCI and without Functional Safety (FS)**
  The PLC program decides whether and at what time a repeated test is to be run or suggested after the control has been switched on. The PLC program can use Module 9144 to start the test immediately, taking the minimum requirements (mode 0) into account.
  If the PLC program wants to test the brakes beforehand, it must command the brake test through PLC module 9143. Before the repeated self test begins, the PLC program must switch off the drives.
- The marker M4190 is set and the marker M4189 is reset when a test is started.
- The marker M4189 is set and the marker M4190 is reset when the entire test cycle has been completed.
- The marker M4192 is set when there is a request for the control voltage to be switched on.

Module 9144 and the PLC markers are used to start the repeated self-test through the PLC and to fully automate the self-test. The test is started by Module 9144 and at first triggers an external emergency stop through the signals MC.RDY and STO.A.G. Then the test continues up to the point at which the control voltage must be switched on again. This can be done by the user on request by the NC, or the procedure can be automated in such a way that marker 4192 is evaluated and the control voltage is switched on by the PLC. As soon as the control voltage has been switched on again, the self-test continues up to completion.

Call:

PS	K/B/W/D	<>Mode> 0: Start self test immediately 1: Reserved 10: Define the operating mode for functional safety 11: Request for testing the axis position
PS	K/B/W/D	<>Parameter 1> Mode 0: No evaluation, must be programmed Mode 1: No evaluation, must be programmed Mode 10: 0: Operation through machine operating panel 1: Operation through manual control unit 2: Homing and testing of axes Mode 11: Number of axis to be tested
PS	K/B/W/D	<>Parameter 2> Mode 0: No evaluation, must be programmed Mode 1: No evaluation, must be programmed Mode 10: No evaluation, must be programmed Mode 11: No evaluation, must be programmed
CM	9144	
PL	B/W/D	<>Status/Error> Mode 0: 0: Function is executed 1: Error according to W1022

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid value programmed for mode or parameter
	28	Test already active
	43	This is not an HSCI system or a system with functional safety
	51	Function is not supported by this control
	58	Control without operating-mode group
	61	Function is not supported by this control

Module 9143 Activate the brake test

With this module an axis-specific brake test with the configuration from the machine parameters or with other values for machine parameters 2230 and 2232 can be started.

Constraints:

- Programming it in a submit job blocks other submit jobs until the test is completed.
- The PLC module itself gives the processing time to other spawn and submit processes.

Call:

PS	K/B/W/D	<>Axis number> 0 = 1st axis, 2 = 2nd axis, etc.
PS	K/B/W/D	<>Multiplier for rated current> Value in 1/1000 or 0: Default MP2230 (factor of nominal current)
PS	K/B/W/D	<>Permissible traverse path> Value in 0.1 [um] or 0: Default MP2232
CM	9143	
PL	B/W/D	<>Status/Error> 0: Brake OK 1: Brake defective 2: Invalid axis or negative values for rated current or traverse path 3: Call during running NC program or during other PLC jobs 4: Call was made from a cyclic PLC program 5: Error during data exchange 6: Not allowed for safe control 7: Drive not ready 8: Brake test was aborted (e.g. by emergency stop)

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis programmed (invalid axis number, not a closed-loop axis, axis currently open-loop axis or slave axis) or negative values for current or traverse path are programmed
	8	Module is not allowed for control with Functional Safety
	20	Module was not called in a spawn job or submit job
	21	Call during program run or during other active PLC jobs for the programmed axis
	40	Drive not ready
	45	Canceled due to error during data exchange or due to external influences (e.g. emergency stop)

6.11.15 EN 13849-1

The safety functions provided for you vary depending on the hardware and software installed in the various HEIDENHAIN control systems.

On the iTNC 530 with the MC 42x and CC 42x hardware, no integrated safety functions, such as safe operating modes or safely limited speed, are available. The entire realization of these functions according to EN 13849-1 requires external safety components. TÜV Süd simply inspected these controls in regard to the safe switch-off of the drives.

The HEIDENHAIN safety system for the iTNC 530 with MC 42x and CC 42x is based on a dual-channel structure. Safety contactor combinations (SCC) are central to the measurement and evaluation of all safety-relevant assemblies (e.g. keylock switch, emergency stop button). The SCCs generate output signals that are received as input signals by the two safety channels. The safety functions are realized over dual-channel enabling of the function.

One channel is realized through external wiring with the SCCs to the enable relays in the inverters. The second channel includes the control, which consists of the main computer (MC) and one or more drive controller modules (CC= controller computer). Safety-relevant errors always lead to safe stopping of all drives through defined stop reactions.

The basic circuit diagram of the respective control with MC 42x and CC 42x serves as the basis for this dual-channel design of cut-off paths. This is a non-binding proposal from HEIDENHAIN, and must be adapted to the requirements of the machine used. As a machine manufacturer, you are responsible for adhering to the relevant standards and safety regulations.

The documents on this topic include the technical report of TÜV Süd and a system description with application conditions from HEIDENHAIN. The requirements of both documents must be complied with.

The software used in the individual controls is not relevant for the dual-channel cut-off paths and was therefore not assessed by TÜV Süd. Only the MC 42x, CC 42x hardware, the inverters, and the external safety components used are decisive for the evaluation of the cut-off paths.

In addition, the control-is-ready output of the MC 42x is only a single-channel design and is therefore not a "safe" output. The control-is-ready output in the HEIDENHAIN controls, however, serves as single-channel acknowledgment signal to the SCCs and must be wired according to the basic circuit diagrams.

6.12 Spindle

Two spindles can be controlled alternately (See "Operating a second spindle" on page 981).
The main spindle/spindles are controlled with the PLC.

The programmed speed can be output as

- Code via PLC outputs
- Analog nominal speed command signal for an analog spindle
- Digital nominal speed value for a digital spindle

The spindle functions are of varying priorities. If several functions are output at the same time, the function with the highest priority is run, and the rest are deleted. The following spindle function priorities are valid:
Highest: Spindle orientation
Second: Transmission jog
Third: M3/M4
Fourth: M5

- Specify in MP3010 the speed output for the spindle.

MP3010 **Output of speed, gear range**
Input:
0: No output of spindle speed
1: Speed code, if the speed changes
2: Speed code at every TOOL CALL
3: Nominal speed value always, G code if the gear shifts
4: Nominal speed value always, G code at every TOOL CALL
5: Nominal speed value always, no G code
6: Same as 3, but with controlled spindle for orientation
7: Same as 4, but with controlled spindle for orientation
8: Same as 5, but with controlled spindle for orientation

6.12.1 Position encoder of the spindle

Analog and digital spindles can be driven in a closed control loop. In this case the spindle needs its own position encoder:

- Define the position encoder input in MP111.x.
  - If you have a digital spindle and would like to use the speed encoder also as a position encoder, then you must set MP111.x = 0.
- Enter in MP3142 the line count of the encoder to be used. 1-V_{PP} signals undergo 1024-fold subdivision.
- Enter in MP3142 how the position encoder is mounted on the spindle. Due to the higher required accuracy, the position encoder must be mounted directly on the spindle: **MP3143 = 0**

If design considerations make this impossible:

- Define the encoder-to-spindle transmission ratio in MP3450.x and MP3451.x for each gear stage.

In this case there will be several reference pulses per revolution. For example, with a transmission of 4:1 (motor to spindle), you will receive four reference pulses (every 90°) per spindle revolution.

- Evaluate the reference mark with Module 9220. See “Renewed traversing of the reference marks" on page 807.

If **MP3143 = 1,** then X30 pin 1 is evaluated as the reference signal. If **MP3143 = 3,** then the **second** signal at X30 pin 1 is evaluated as the reference signal. This can be necessary if the spindle is located shortly before the switch for the reference signal. This might not be detected, and so only the next signal can be evaluated.
In both cases the reference mark of the position encoder is not evaluated. In this case the reference signal **must** be evaluated with Module 9220 (See "Renewed traversing of the reference marks" on page 807).

Attention

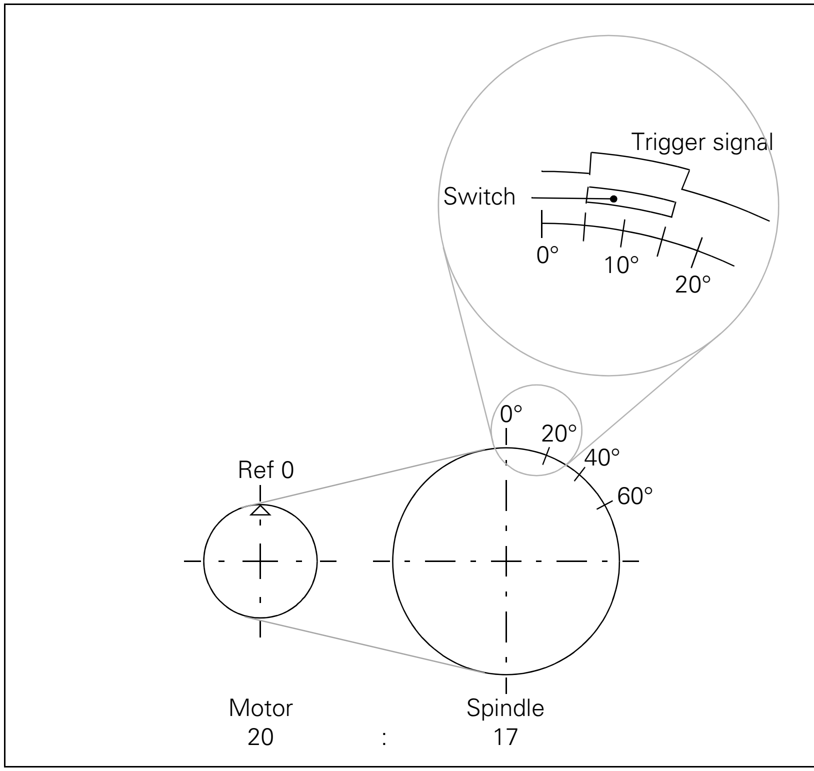
Due to its low accuracy, this solution is not recommended.

If **MP3143 = 2,** then the reference pulse release for the spindle position encoder is set with X30, pin 1. This ensures that the same reference mark is always evaluated. As soon as a reference mark is traversed, the pulses are counted and immediately after the switch is traversed the "correct" reference mark is calculated. This might mean that the spindle is no longer moved after the switch is passed over.
The signal must be available for at least as long as the position controller cycle time + 2 ms. This means that the maximum speed for detecting the signal is:

$$n_{max} = \frac{s \cdot 60000}{360 \cdot (t_{Pos.} + 2\text{ ms})}$$

s: Trigger window of the switch (10° in the example)
$t_{Pos.}$: Position controller cycle time

The example results in a maximum speed of:



> **Note**
>
> The switch for enabling of the reference mark signal must be adjusted precisely enough for the "correct" reference mark to be evaluated!

$$n_{max} = \frac{10° \cdot 60000 \text{ ms/min}}{360° \cdot (3 \text{ ms} + 2 \text{ ms})} = 333 \text{ min}^{-1}$$

The transmission ratio (MP3450.x, MP3451.x) is evaluated only if the value in MP3143.x is > 0.

Two additional input values 4 and 5 have been introduced for the machine parameter MP3143 for the type of mounting of the position encoder of the spindle:

MP111	**Position encoder input for the spindles**
Input:	0: No position encoder input 1 to 6: Position encoder inputs X1 to X6 35 to 38: Position encoder inputs X35 to X38
MP111.0	Position encoder input for the first spindle
MP111.1	Position encoder input for the second spindle

MP3142	**Line count of the spindle position encoder**
Input:	100 to 100 000 [lines]

MP3143	**Mounting configuration of the spindle position encoder**
Input:	0: Position encoder immediately on the first spindle 1: Position encoder via transmission (ratio in MP3450.x and MP3451.x); X30 pin 1: reference pulse 2: Position encoder via transmission (ratio in MP3450 and MP3451); X30 pin 1: reference pulse release 3: Same as input value 1, except that the second reference pulse is evaluated. 4: Reference-mark evaluation of the spindle via EnDat. The encoder must be mounted directly (without transmission). No reference pulse is necessary. A new reference-mark evaluation via marker M4015 may only be performed at standstill. 5: The position encoder is mounted directly (same as input value 0) and the transmission ratio is also evaluated. Entering the value 5 results in the same behavior as that of software version 340 49x-05 (rigid tapping will also be possible without a position encoder on the spindle)!

MP3450.0-7	**Number of spindle position-encoder revolutions for gear ranges 1 to 8**
Input:	0 to 65 535 0: No transmission

MP3451.0-7	**Number of spindle position-encoder revolutions for gear ranges 1 to 8**
Input:	0 to 65 535 0: No transmission

Module 9042 Reading the spindle coordinates (format 0.001°)
The following coordinate values are saved in five successive double words beginning with the specified target address:

- Actual value
- Nominal value
- Actual value in reference system
- Following error (servo lag)
- Distance-to-go

The values for actual, nominal, and reference value are standardized at 0° to +360.000°.
The values for servo lag and distance-to-go are displayed between –2879.912° and +2879.912°. Format: 0.001°.

If MP3010 < 6 (no closed-loop spindle), then all coordinates are read as zero.

During operation under open-loop control (M03 / M04 active or M05 and open position control loop), the nominal value is considered to be the actual value. The servo lag and distance-to-go are considered to be zero.

Call:
PS B/W/D/K <>Target address Dxxxx>
CM 9042

Error recognition:

Marker	Value	Meaning
M4203	0	Actual speed value was read
	1	Target address is too large or is not a double-word address

Module 9044 Reading the spindle coordinates (format 0.0001°)
Call:
SEE MODULE 9042.

6.12.2 Speed encoder of the spindle

Digital speed control requires a shaft speed encoder:

- Define the speed encoder input in MP113.x.

The iTNC 530 monitors the reference mark of the speed encoder. The monitor checks whether the line count for one revolution from reference mark to reference mark is equal to the line-count entry in the motor table.
If differences occur, the DSP error message **`C3A0 Incorrect reference position S`** appears. If this happens, check the speed encoder, encoder cable, and whether you have selected the correct motor.

With a gear wheel encoder, even if it is properly installed, monitoring can result in this error message due to its inherent inaccuracy:

- In this case, switch the monitoring off with MP2220 bit 0 = 1.

The iTNC 530 monitors the direction of rotation. If the nominal value of current exceeds the limit value for a certain time, the DSP error message **`C380 Motor <spindle 1/2, axis> not controllable`** appears.

At lower speeds, high-frequency spindles only have a low amount of torque. If such a spindle is having its speed controlled, the tool changer may slightly twist the spindle, causing the limit of current to be exceeded. This leads to the above error message:

- In this case, switch the monitoring off with MP2221 bit 1 = 1.

Attention

For axes, monitoring of the rotational direction (MP2220 bit 1) must **not** be deactivated. An error (e.g. one motor phase interchanged with another or incorrect entry in the **`DIR`** column of the motor table) might cause uncontrolled acceleration of the motor in one direction if the monitoring function for the rotational direction is deactivated.
This also applies to spindles. For spindles, however, an incorrect acceleration in one direction is less dangerous than for axes.

As of NC software 340 420-06, monitoring of the direction of rotation (MP2220 bit 1) for synchronous motors (entry **`SM`** in the column **`TYPE`** in the motor table) cannot be switched off.

MP113	**Speed encoder for the spindle/spindles**
Input:	0: No speed encoder 15 to 20: Speed encoder inputs X15 to X20 80 to 85: Speed encoder inputs X80 to X85
MP113.0	Speed encoder for the first spindle
MP113.1	Speed encoder for the second spindle
MP2220	**Monitoring functions**
Input:	Bit 0 – Monitoring the reference mark 0: Monitoring active 1: Monitoring inactive Bit 1 – Monitoring the rotational direction 0: Monitoring active 1: Monitoring inactive

6.12.3 Analog and digital closed-loop spindle control

For both analog and digital output of the nominal speed command you can program speeds from 0 to 99 999.999 min^{-1}.

The maximum controllable spindle speed is: $\frac{80\ 000}{\text{No. of pole pairs}}$

If the load increases, the spindle speed is corrected until the maximum current is attained. If the load continues to increase in spite of the maximum current, the spindle speed is reduced. For the maximum current, the value from either the motor table or the power-module table of the drive (whichever is lower) applies.

If in MP3010 you have selected the output of the nominal speed value, M4003 is set. The programmed speed is saved in D356, the nominal speed value in W320 and the actual speed value in W322. In addition, the nominal speed value is saved in D364 and the actual speed value in D368, since speeds above 32 767 min^{-1} cannot be represented in words W320 and W322.

The PLC double word D372 makes the maximum spindle speed including the spindle potentiometer available to the PLC program. This makes it possible, for example, to already acknowledge within the ramp the M functions for switching on the spindle by comparing the actual and maximum speeds.

With D604 you can limit the possible spindle speed through the PLC. To ensure compatibility, D604 is preassigned with 99 999 999 after control switch-on or after an interruption in the PLC scan.

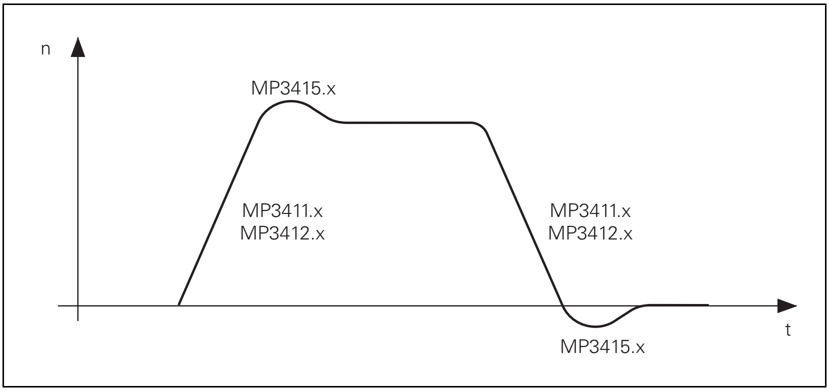
Analog spindles:
The nominal speed value of the motor is output as an analog dc voltage of ±10 V at connection X8 or X9.

Digital spindles:
The nominal speed value is transferred to the internal speed controller.

		Set	Reset
M4003	**Nominal speed value output analog or digital (MP3010 = 3 to 8)**	NC	NC
D356	**Programmed speed [0.001 min^{-1}]**	NC	NC
D364	**Nominal speed value [min^{-1}]**	NC	NC
W320	**Nominal speed value [min^{-1}]**	NC	NC
D368	**Actual speed value [min^{-1}]**	NC	NC
D372	**Maximum spindle speed including spindle override [min^{-1}]**	NC	NC
W322	**Actual speed value [min^{-1}]**	NC	NC
D604	**Maximum possible spindle speed**	PLC	NC/PLC

Nominal speed value

- In MP3411.x, define the ramp gradient for the nominal speed value at M03 and M04 for each gear range.
- With MP3412.0, specify a multiplication factor for MP3411.x, for
  - M05 (MP3412.0)
  - SPINDLE ORIENTATION (MP3412.1)
  - TAPPING (with floating tap holder) (MP3412.2)
  - RIGID TAPPING (without floating tap holder) (MP3412.3)
    The same factor applies for all gear ranges.
- Set MP3411 for M03, M04 and M05 such that the motor accelerates and brakes within the current limit.
- With MP3415, define the overshoot behavior for every operating mode when the spindle is switched on with M4011. Set MP3415.0 so that only one overshoot is visible.



If the nominal speed value is in the acceleration or deceleration ramp, then M4001 is reset. This also applies if the speed is changed with the override potentiometer.

If the nominal speed value is output as zero, M4002 is set.

MP3411.0-7	**Ramp gradient of the spindle with M03 and M04 for gear ranges 1 to 8**
Input:	Analog axes: 0 to 1.999 [V/ms] Digital axes: 0 to 1.999 [1000 min^{-1}/ms]

MP3415	**Overshoot behavior of the spindle with M03, M04 and M05**
Input:	0 to 1000 [ms]
MP3415.0	With M03, M04 and M05
MP3415.1	For spindle orientation
MP3415.2	For tapping
MP3415.3	For tapping without floating tap holder

MP3412	**Multiplication factor for MP3411.x**
Input:	0.000 to 1.999
MP3412.0	With M05
MP3412.1	With spindle orientation
MP3412.2	For tapping with floating tap holder
MP3412.3	For tapping without floating tap holder

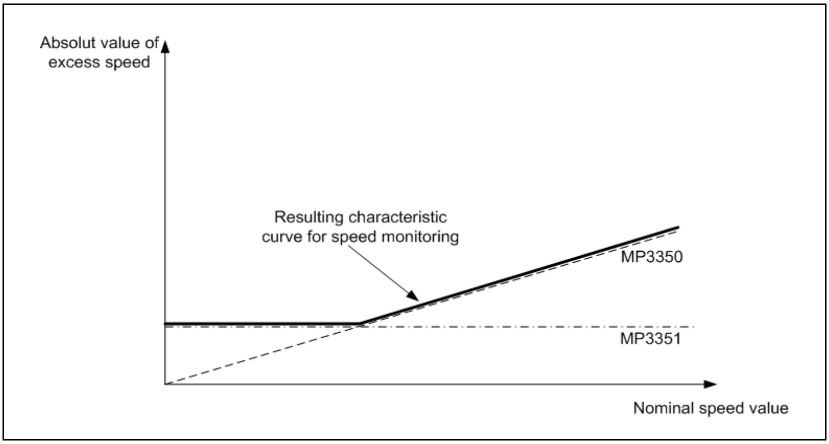
		Set	Reset
M4001	**Nominal speed command signal of the spindle not in the ramp**	NC	NC
M4002	**Nominal speed value = 0**	NC	NC

Monitoring the spindle speed

With MP3350 and MP3351 you set the maximum permitted excessive spindle speed. An NC stop with a subsequent emergency stop is triggered if the actual spindle speed is greater than the nominal spindle speed + the permitted excess speed. With this function you can also have the spindle be monitored for unexpected start-up.

In MP3350 you enter a relative value in percent, and in MP3351 and absolute value, for the permissible excessive speed. The absolute value in MP3351 is only used if the absolute value that results from MP3350 is less than the value in MP3351.

If you want to use monitoring of the spindle speed for spindles with gear stages, then you should be more generous with the tolerance that results from MP3350 and MP3351, since the spindle speed can vary significantly when switching between the gear stages.



A percentage value for the permissible shortfall of the nominal spindle speed can be entered in MP3540. You can thus use this machine parameter for monitoring whether the spindle speeds falls below a specific spindle speed limit (e.g. nominal spindle speed = 500 rpm, MP3450 = 20 % –> the monitoring function will respond if the spindle speed falls below 400 rpm). Monitoring is useful for power milling where the workpiece is to be machined at a high feed rate and a low constant spindle speed. If the monitoring function responds, an EMERGENCY STOP reaction is triggered in order to protect the machine from possible damage, for example if the spindle comes to a standstill while still engaged in the material. The error message "8BF0 Actual speed value too small" is displayed. Monitoring does not become active until the spindle speed exceeds the value defined in MP 3542.

Monitoring for falling below the lower spindle speed limit (defined in MP3540) does not become active until the actual spindle speed exceeds the value defined in MP3542. This threshold is required in order to avoid activating the monitoring function before a reasonable shaft speed is reached (for example, if the noise of encoders at the actual value of the spindle speed is of subordinate importance, or in order to prevent monitoring from becoming active when the tool plunges into the material).

In an EMERGENCY STOP, braking of the spindle is delayed compared to braking of the axes by the time in MP3550. This avoids tool breakage during power milling, because low spindle speeds might result in the spindle coming to a standstill before the axes do. However, the spindle braking is delayed by no more than half a spindle revolution.

MP3350	**Maximum permissible overshoot of the spindle speed in percent of the nominal spindle speed**
Input:	0 to 100 [%]
MP3351	**Entry of an absolute value for the permissible overshoot of the spindle speed**
Input:	0.001 to 99 999.999 [rpm] 0: Monitoring off
MP3540	**Permissible spindle speed shortfall**
Input:	1 to 99 [%] 0 = Monitoring not active
MP3542	**Minimum spindle speed as of which the monitoring in MP3540 becomes active**
Input:	0 to 100000 [rpm]
MP3550	**Delay of EMERGENCY STOP reaction of spindles**
Input:	0.001 to 0.100 [s] 0 = Delay not active

Increased spindle power for roughing

With MP3530 and MP13530 you can avoid a reduction in the maximum spindle power during roughing due to changes in load when the tool's teeth are engaged in the material.
Here the spindle speed is increased higher than previously when there is no load. It is now increased above the nominal speed, which was not the case previously. Energy is stored in the inertia of the spindle, and is then available when the teeth engage the material. This increases the mean power, and the mean speed corresponds to the nominal speed.
If MP3530 or MP13530 are set, an overshoot can occur when the nominal speed is reached. If this overshoot impairs the performance, then you may have to deactivate the increase of the spindle power.

MP3530	**Increased spindle power for roughing**
Input:	0 = Not active 1 = Increased spindle power for roughing
MP13530	**Increased spindle power for roughing (second spindle)**
Input:	0 = Not active 1 = Increased spindle power for roughing

Direction of spindle rotation

- With MP3130, define the polarity of the nominal speed value
- In MP3140, enter the counting direction of the position encoder signals.

As soon as you set M4005 for M03, or M4006 for M04, the nominal speed value is output. With M4007 for M05, the nominal speed value zero is output (spindle stop).

M4005 to M4007 also controls the miscellaneous functions in the status window.

If more than one marker is set at the same time, the error message `PLC: M4005, M4006, M4007 INCORRECT` appears.

With M4014 you can reverse the direction of rotation, in order to adjust the transmission for horizontal or vertical spindles, for example. The polarity of the nominal spindle speed is inverted.

M4019 reverses the counting direction of the spindle.

MP3130	**Polarity of the nominal spindle speed**
Input:	0: M03 positive, M04 negative 1: M03 negative, M04 positive 2: M03 and M04 positive 4: M03 and M04 negative
MP3140	**Counting direction of spindle position encoder output signals**
Input:	0: Positive counting direction with M03 1: Negative counting direction with M03

		Set	Reset
M4005	**Status display and nominal speed value output for M03**	PLC	PLC
M4006	**Status display and nominal speed value output for M04**	PLC	PLC
M4007	**Status display M05 and spindle stop**	PLC	PLC
M4014	**Reverse the direction of spindle rotation**	PLC	PLC
M4019	**Reversing the counting direction of the position encoder on the spindle**	PLC	PLC

Disable speed output for spindle

With M4008 you can block the speed output for the spindle. At the same time, M03, M04 or M05 are highlighted. The nominal speed value is zero.

		Set	Reset
M4008	**Disable speed output for spindle**	PLC	PLC

Gear ranges

You can define up to eight gear ranges:

- In MP3510.x, enter for each gear range the rated speed for "S-override 100%."
  Enter the value zero for unnecessary gear ranges.
- In MP3210.x, enter for every gear range the S analog voltage or motor revolutions at rated speed.
- In MP3240.1, define the minimum nominal speed value for the motor.
- In MP3120, define whether zero is permitted as a programmed speed.

If an impermissible speed is programmed, M4004 is set and the error message **WRONG RPM** is displayed.

Note

The gear range from W256 is output when the spindle speed is 0.

MP3510.0-7 Rated speed for the gear ranges 1 to 8
Input: 0 to 99 999.999 [min^{-1}]

MP3210.0-7 Analog nominal spindle voltage at rated speed for the gear ranges 1 to 8
Input: 0 to 100.000 [V]

MP3210.0-7 Digital spindle motor revolutions at rated speed for the gear ranges 1 to 8
Input: 0 to 100.000 [1000 min^{-1}]

MP3240.1 Analog spindle: Minimum nominal value voltage
Input: 0 to 9.999 [V]

MP3240.1 Digital spindle: Minimum motor speed
Input: 0 to 9.999 [1000 min^{-1}]

MP3120 Zero speed permitted
Input: 0: S = 0 permitted
1: S = 0 not allowed

		Set	Reset
M4004	**Impermissible speed was programmed**	NC	NC

Gear shifting

You control the gear shifting through PLC outputs. The NC enters the current gear range according to the programmed speed in W256. The gear range is calculated with MP3510.x. The output of the gear range is defined in MP3010. MP3030 bit 1 determines if the speed should be reduced to 0 when shifting between gears.

Attention

- On machines with transmission or wye/delta switchover, the RPM FACTOR (Q403) function of Cycle 209 (Tapping with chip breaking) must not be used with a value other than the default value. Switching the gear range or wye/delta switchover during retraction with the RPM FACTOR function may cause damage to the machine.
- Please inform the customers who are using these types of machines that the RPM FACTOR (Q403) function of Cycle 209 (Tapping with chip breaking) must not be used with a value other than the default value.

When the gear range is changed, the NC uses the G strobe (M4070). As soon as you confirm the gear shift with M4090, the program resumes and the G strobe (M4070) is reset by the NC.
If a TOOL CALL block is followed by the output of a T strobe and G strobe, then M4547 is set by the output of the T strobe and reset by output of the G strobe. If there is no output of either the T or G strobe, M4547 is not set.

In the PLC program you can change the programmed speed and the gear range that is calculated by the NC. The programmed speed is saved by the NC in D356 and D756. The following procedure is fundamentally possible, but not in connection with mid-program startup.

- Enter a speed in D756 and a gear range in W256. The speed must lie within the speed range of the gear.
- With M4134, activate your entries in D756 and W256.
- After the NC has reset M4134, change the gear and report with M4090 that the gear shift has been completed.

So that no problems arise during mid-program startup, or when horizontal/vertical spindles are in use, gear shifting should be realized for the spindle via various sets of parameters.

A changing nominal speed value ramp can be output to shift gears by alternately setting and resetting M4009 and M4010. This can be realized by interrogating the timers in the PLC program. This function also works if you have used M4008 to disable the speed output for the spindle:

- In MP3240.2, define the nominal speed value that is output with M4009/M4010 to the spindle motor.

MP3030	**Behavior of the spindle**
Input:	Bit 1– Zero spindle speed when shifting to another gear range 0: Reduce speed to 0 1: Do not reduce speed to 0
MP3240.2	**Analog spindle: Spindle jog voltage for gear shifting (M4009/M4010)**
Input:	0 to 9.999 [V]
MP3240.2	**Digital spindle: Motor speed for gear shifting (M4009/M4010)**
Input:	0 to 9.999 [1000 min^{-1}]

		Set	Reset
W256	**Gear code**	NC/PLC	NC/PLC
D356	**Programmed speed [0.001 min^{-1}]**	NC	NC
D756	**Programmed speed or speed from PLC [0.001 min^{-1}]**	NC/PLC	NC/PLC
M4009	**Counterclockwise spindle rotation (for gear change)**	PLC	PLC
M4010	**Clockwise spindle rotation (for gear change)**	PLC	PLC
M4070	**Strobe signal for gear code**	NC	NC
M4090	**Acknowledgment of "gear change completed"**	PLC	PLC
M4134	**Activation of a gear range and speed through the PLC**	PLC	NC
M4547	**T and G strobes with TOOL CALL**	NC	NC

Spindle override

You can change the spindle speed within certain limits with the spindle override potentiometer.

- Define the limits in MP3310.x.
- In MP3515.x, enter for every gear range a maximum attainable speed which must not be exceeded with the spindle override.

The percentage adjusted with the spindle override is entered by the NC in W492 and W764. You can change the percentage through the PLC:

- Enter the desired percentage in W764.
  The NC adopts the new value as soon as it is entered.

The spindle override functions either in 1% steps or according to a nonlinear characteristic curve:

- With bit 3 of MP7620, select the mode of the override.

Value range in W492 and W764:

- 1% steps: 1 to 150
- Nonlinear characteristic curve: 0 to 15 000
  In the lowest range, 0.01% steps are available. Beginning with a value of 2.5%, the step is 0.75%

If bit 5 of MP7620 is set, the spindle override potentiometer is used as rapid traverse override potentiometer. The value set with the potentiometer will then be entered by the NC in words W496 and W752 (if bit 5 = 1). The value in word W752 can be changed by the OEM and will be used by the NC. The entry in MP3310 for limiting the spindle override also applies for the rapid traverse override. The value range for W496 and W752 is independent of the setting in MP 7620 bit 3 for the "nonlinear curve." The potentiometer designation on the keyboard unit or the machine operating panel must match the setting in MP7620 bit 5.
The OEM must then use vacant keys on the machine operating panel for the spindle override, for example. The OEM can specify a value for the spindle override in word W764 of the PLC. The value is used by the NC.

Danger

Only use MP7620 bit#5 in accordance with the keyboard unit or the machine operating panel used. The potentiometer designation must match the function you set in MP7620 bit 5.
An incorrect or missing identification of the potentiometers can lead to errors during operation and even to property damage or personal injury!

If bit 8 of MP7620 is set, the error message "Override potentiometer" will be displayed if the feed-rate or rapid-traverse override is set to 0% at the start of an axis movement and the axis can therefore not be moved.

MP3310.0-1	**Limit for spindle override**
Input:	0 to 150 [%]
MP3310.0	Upper limit
MP3310.1	Lower limit

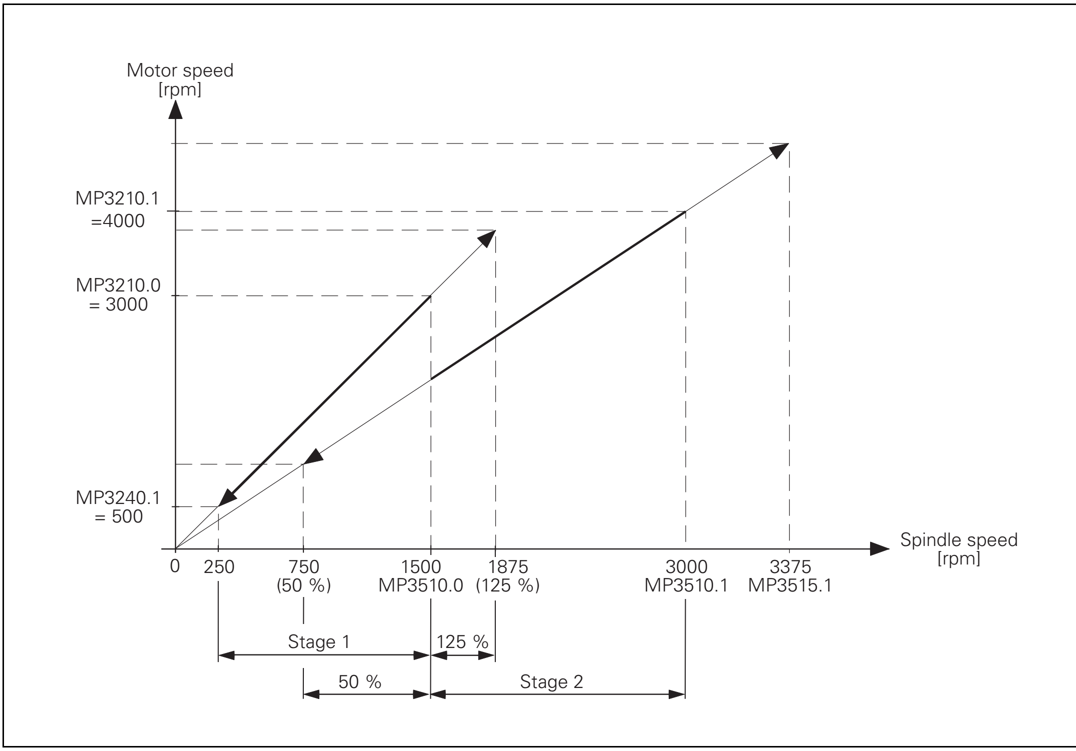
MP3515.0-7	**Maximum spindle speed for gear ranges 1 to 8**
Input:	0 to 99 999.999 [min^{-1}]

MP7620	**Feed-rate override and spindle speed override**
Input:	%xxxxxxx Bit 3 – Feed rate override and spindle speed override in 1% increments or according to a nonlinear characteristic curve: 0: 1% steps 1: Nonlinear characteristic curve Bit 5 – Rapid traverse override instead of spindle override 0: Potentiometer is used for spindle override 1: Potentiometer is used for rapid traverse override Bit 8 – Informational text if feed-rate or rapid-traverse override is set to 0% 0: Informational text inactive 1: Informational text active

		Set	Reset
W492	**Percentage for spindle override (NC to PLC)**	NC	NC
W764	**Percentage for spindle override (PLC to NC)**	NC/PLC	NC/PLC

Example: Two gear ranges for a digital spindle

- Gear range I: Spindle 1500 min^{-1} with motor 3000 min^{-1} (MP3210.0 = 3000; MP3510.0 = 1500)
- Gear range II: Spindle 3000 min^{-1} with motor 4000 min^{-1} (MP3210.1 = 4000; MP3510.1 = 3000)
- Upper limit for spindle override : 125% (MP3310.0 = 125)
- Lower limit for spindle override: 50% (MP3310.1 = 50)
- Maximum possible output speed for gear range II: 3375 min^{-1} (MP3515.1 = 3375)
- Minimum motor speed: 500 min^{-1} (MP3240.1 = 500)


Motor speed
[rpm]
MP3210.1
=4000
MP3210.0
= 3000
MP3240.1
= 500
Spindle speed
[rpm]
0
250
750
(50 %)
1500
MP3510.0
1875
(125 %)
3000
MP3510.1
3375
MP3515.1
Stage 1
125 %
50 %
Stage 2

6.12.4 Coded output of spindle speed

If you have selected speed-code output in MP3010 (entry 1 or 2), an S code is entered in W258. You must output the speed code to the spindle drive through PLC outputs.

If the speed code is changed, the NC sets the S strobe (M4071). If you acknowledge the S code with M4091, the NC program is continued and the S strobe (M4071) is reset by the NC.

If required, the programmed spindle speed is rounded off to the next standard value by the NC and given in S code as per ISO 6983 (see S-code table). Speeds of 0 to 9000 min^{-1} are possible:

- Specify in MP3010 the speed range and the speed increment. The S code for the minimum speed is stored in W1008.

Example:

Minimum speed = 1 min^{-1} (S code 20)
Maximum speed = 1000 min^{-1} (S code 80)
Speed increment = 2:

MP3020 = 20802
W1008 = 20

MP3020	**Speed range for S code output**
Format:	xxyyz xx: S code for minimum speed yy: S code for maximum speed z: Speed increment
Input:	0 to 99 999

		Set	Reset
W258	**S code**	NC	NC
M4071	**Strobe signal for S code**	NC	NC
M4091	**Acknowledgment of S code**	PLC	PLC
W1008	**S code for minimum speed**	NC	NC

S code table

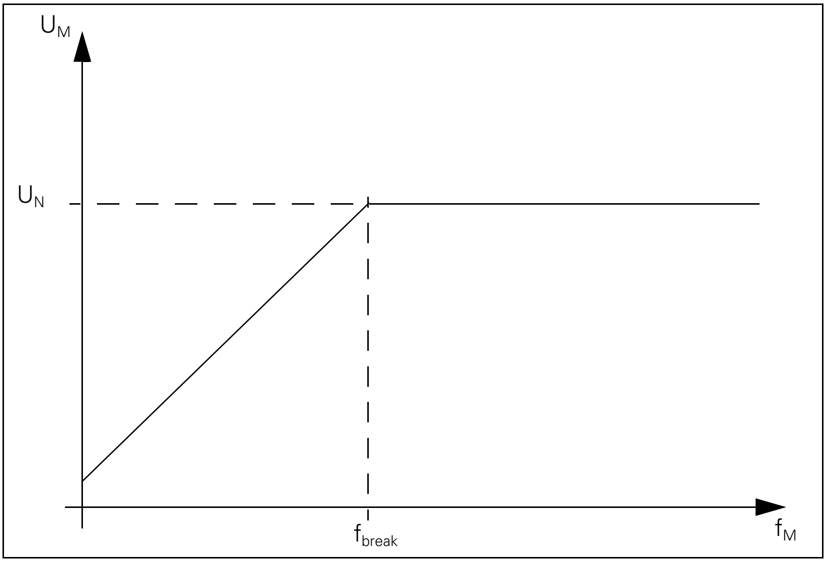
S code	min^{-1}
S 00	0
S 01	0.112
S 02	0.125
S 03	0.14
S 04	0.16
S 05	0.18
S 06	0.2
S 07	0.224
S 08	0.25
S 09	0.28
S 10	0.315
S 11	0.355
S 12	0.4
S 13	0.45
S 14	0.5
S 15	0.56
S 16	0.63
S 17	0.71
S 18	0.8
S 19	0.9
S 20	1
S 21	1.12
S 22	1.25
S 23	1.4
S 24	1.6
S 25	1.8
S 26	2
S 27	2.24
S 28	2.5
S 29	2.8
S 30	3.15
S 31	3.55
S 32	4
S 33	4.5
S 34	5
S 35	5.6
S 36	6.3
S 37	7.1
S 38	8
S 39	9
S 40	10

S code	min^{-1}
S 41	11.2
S 42	12.5
S 43	14
S 44	16
S 45	18
S 46	20
S 47	22.4
S 48	25
S 49	28
S 50	31.5
S 51	35.5
S 52	40
S 53	45
S 54	50
S 55	56
S 56	63
S 57	71
S 58	80
S 59	90
S 60	100
S 61	112
S 62	125
S 63	140
S 64	160
S 65	180
S 66	200
S 67	224
S 68	250
S 69	280
S 70	315
S 71	355
S 72	400
S 73	450
S 74	500
S 75	560
S 76	630
S 77	710
S 78	800
S 79	900
S 80	1000
S 81	1120

S code	min^{-1}
S 83	1400
S 84	1600
S 85	1800
S 86	2000
S 87	2240
S 88	2500
S 89	2800
S 90	3150
S 91	3550
S 92	4000
S 93	4500
S 94	5000
S 95	5600
S 96	6300
S 97	7100
S 98	8000
S 99	9000

6.12.5 Volts-per-hertz control mode

In volts-per-hertz control mode (U/f control mode), the motor is speed-controlled in an open loop. The motor voltage increases in proportion to frequency up to the break (= threshold rpm for field weakening). Then the motor voltage remains constant (= rated voltage of motor); only the frequency continues to increase.
If the spindle reaches the maximum current due to excessive load, the error message **C380 Motor <Spindle 1/2> not controllable** appears. For the maximum current, the value from either the motor table or the power-module table of the drive (whichever is lower) applies.



The maximum speed in the volts-per-hertz control mode corresponds to the maximum speed in closed loop operation.

To drive a motor in the volts-per-hertz control mode :

- In the motor table, enter for your motor in the column **Motor model** (TYPE) UASM, in the column **Encoder line count** (STR.) the value 0, in the column **Type of encoder** (SYS) the value 0 and in the column **Maximum temperature** [°C] T-MAX the value 255.
- The machine parameters for current controller (MP24xx.y) and speed controller (MP25xx.y, MP 26xx.y) are nonfunctional.
- The acceleration and breaking ramp (MP341x) must be set so that the maximum current is not exceeded.

Since during volts-per-hertz (U/f) control mode no speed encoder is used, W322 = 0 (actual speed value) supplies the value 0:

- Module 9164 can determine the actual speed value while the spindle is running, but not during the acceleration and braking phases.

Note

The oscilloscope shows the actual current instead of the nominal current (I NOML), since there is no nominal current with U/f components.

6.12.6 Oriented Spindle Stop

For spindle orientation the spindle must be in a closed control loop:

- Mount a position encoder for the spindle.
- With MP3010 (input value 6 to 8), specify whether the control provides for spindle orientation.

In the NC's touch probe cycles and rigid tapping cycle, the NC orients the spindle directly. In these cases, the NC sets M4017. You must reset M4012 in the PLC.

To orient the spindle to a specific angle in an NC program, use FN17: SYSWRITE ID990 NR8. The conditions above must be followed. The NC program resumes after the spindle is in position (M4000).
You can ascertain the current spindle angle with FN18: SYSWRITE ID990 NR8.

If the spindle orientation is started with an M function (e.g. M19), you must activate the oriented spindle stop in the PLC.
In MP7442, enter the number of the M function (e.g., 19) which will trigger the oriented spindle stop during the machining cycles. If MP7442 = 0 (no oriented spindle stop), the error message **`ORIENTATION not permitted`** appears when a cycle which uses oriented spindle stop is called.

The spindle orientation runs asynchronously to the NC positioning commands. You may only acknowledge the orientation once the spindle is in position (M4000).

The NC starts orienting the spindle only if the drive is switched on with Module 9161.

There are three ways to orient the spindle in the PLC:

- Module 9171
- Marker M4130
- Via initiator with marker M4011

MP7442	**Number of the M function for spindle orientation in the cycles**
Input:	1 to 999: Number of the M function 0: No oriented spindle stop –1: Oriented spindle stop by the NC

Oriented spindle stop with Module 9171

The spindle speed is reduced in open-loop control along the ramp from MP3412.1 to the speed for spindle orientation (MP3520.1). As soon as this speed is reached, the control loop closes. The spindle is oriented in feedback control along the ramp from MP3412.1 to the nominal position. As long as the spindle moves in a closed loop, M4017 remains set:

- In MP3440.x, assign each gear range a k_V factor for adjusting the gear ranges.
- In MP3415.1, define the overshoot behavior of the first spindle during spindle orientation.
- Define the positioning window in MP3420. As soon as the spindle is in the positioning window, M4000 is set.

If the spindle should not remain in the position control loop after it reaches the nominal position, then you must set M4012. Once the marker is set the spindle is no longer operated with position feedback control.

If M4012 always remains set, the control loop opens after every oriented spindle stop as soon as the positioning window is reached.

You can compensate a maladjustment resulting from mounting the rotary encoder:

- In MP3430, enter the offset between the nominal and actual position of the reference mark. The offset is then compensated during orientation.

After the spindle is switched on, the NC evaluates the reference mark, even if the position control loop is not closed. M4018 is set until the reference mark is evaluated. For special applications you can evaluate the reference mark again by setting M4015. The NC resets M4015 once the reference mark has been evaluated.

- With MP7291, select the display mode for the spindle position. If M03 and M04 are not active, the display returns to zero every 360 degrees (modulo function).

MP3412.1 **Multiplier for MP3411 during spindle orientation**
Input: 0 to 1.999

MP3415.1 **Spindle overshoot behavior during orientation**
Input: 0 to 1000 [ms]

MP3420 **Spindle positioning window**
Input: 0 to 360.0000 [°]

MP3430 **Deviation of the reference mark from the desired position (spindle preset)**
Input: 0 to 360 [°]

MP3440.0-7 **k_V factor for spindle orientation for gear ranges 1 to 8**
Input: 0.1 to 10 [(1000°/ min) /°]

MP3520.1 **Spindle speed for oriented stop**
Input: 0 to 99 999.999 [min^{-1}]

		Set	Reset
M4000	**Spindle in position**	NC	NC
M4012	**Open the spindle control loop**	PLC	PLC
M4015	**Renewed evaluation of the spindle reference mark**	PLC	NC
M4017	**Servo-controlled spindle in motion**	NC	NC
M4018	**Reference mark for spindle not yet traversed**	NC	NC

With Module 9171 you can specify the speed, nominal position and direction of rotation for spindle orientation.

M4130 is set as long as the positioning movement lasts.

Module 9171 Oriented spindle stop

The module functions only in the cyclic PLC program. If you call the module while the spindle is rotating, the transferred direction will be ignored. The spindle will be oriented in the direction of spindle rotation.

If the values 2 to 4 are transferred as direction of rotation, the spindle will be oriented to the angle last defined in CYCL DEF 13. The transferred angle is added to the value from CYCL DEF 13.

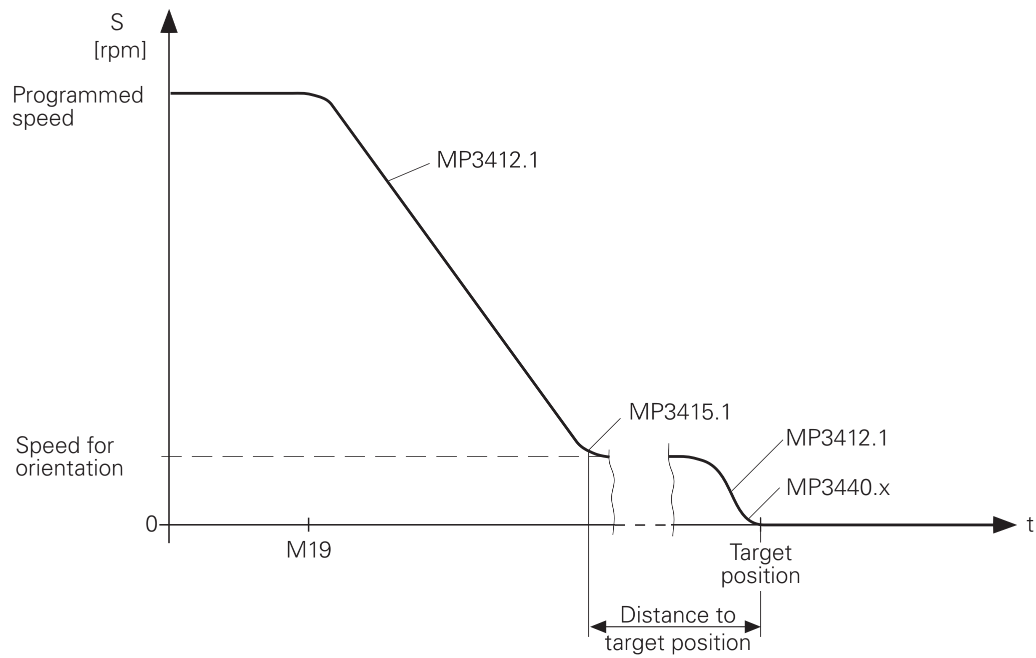
Call:

PS	B/W/D/K	<>Angle [1/10 000 °]> or additional preset if there is a value from CYCL DEF 13
PS	B/W/D/K	<>speed [1/1000 rpm]> 0: MP3520.1 is assumed
PS	B/W/D/K	<>Direction of rotation> –1: Negative direction (M04) 0: Direction of the shortest path 1: Positive direction (M03) 2: Same as –1 but angle from CYCL DEF 13 3: Same as 0 but angle from CYCL DEF 13 4: Same as +1 but angle from CYCL DEF 13 5: Incremental procedure with the entered angle. Depending on the algebraic sign given, orientation occurs in either positive or negative direction (+/–).
CM	9171	

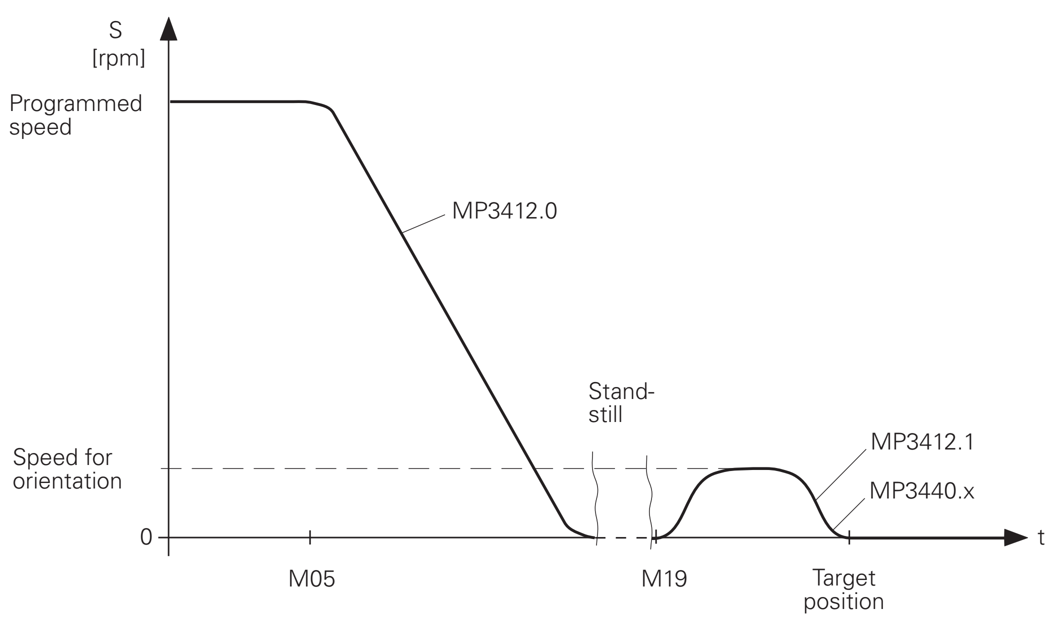
Error recognition:

Marker	Value	Meaning
M4203	0	Spindle is brought to an oriented stop
	1	Error code in W1022
W1022	1	Incorrect value for direction of rotation or rotational angle
	2	Incorrect speed
	19	No feedback-controlled spindle
	24	The module was called in a spawn job or submit job
	27	A spindle orientation is already running

Orientation of a moving spindle:



Orientation of a stationary spindle:



Oriented spindle stop with M4130

You can start the spindle orientation with M4130. The nominal position is taken from D592 and the speed from MP3520.1. The nominal position is with respect to the reference point.

For example, the nominal position can be transferred with MP4210.x or taken from the oriented spindle stop cycle (CYCL DEF 13). If the value is taken from the cycle, you must set the MSB of D592 to 1 and the other bits to 0. M4016 is set during execution of Cycle 13.

From a standstill, the spindle is oriented on the shortest path.
Prerequisite: At the start, the distance between the nominal and actual position must not be greater than the positioning window (MP3420). If the distance is greater than the positioning window, the spindle is positioned according to M4013 with M03 or M04.

		Set	Reset
D592	**Nominal position for spindle orientation**	PLC	PLC
M4013	**Direction for spindle orientation from a standstill (M03 = 0; M04 = 1)**	PLC	PLC
M4016	**Cycle 13 is executed**	NC	PLC
M4130	**Activation of spindle orientation, or spindle orientation has been started with Module 9171**	NC/PLC	NC

MP4210.0-47 Setting a number in the PLC (D768 to D956)
Input: -99 999.9999 to +99 999.9999

Oriented spindle stop via proximity switch with M4011

The spindle can be oriented through a proximity switch:

- Set M4011.

Then the spindle is moved in the direction from M4013 and at the speed from MP3520.0. The spindle is stopped as soon as you reset M4011. The current positioning value is shown in the status window.

MP3520.0 Speed activation through marker M4011
Input: 0 to 99 999.999 [min^{-1}]

		Set	Reset
M4011	**Activate rotational speed MP3520.0 and direction of rotation from M4013**	PLC	PLC

Offset compensation (only analog spindles)

After spindle orientation the offset is compensated automatically. In order to give the spindle enough time to settle to a stop, the offset compensation is delayed until the spindle has been in position for at least two seconds. The offset is then compensated in intervals of 0.152 mV per second. The spindle turns slowly due to the offset voltage.

6.12.7 Tapping with floating tap holder and nominal speed output

For tapping with floating tap holder, the position control loop is open. M4030 is set during the tapping cycle. After the spindle is switched on with M03, this is acknowledged with M4092. The nominal spindle speed must be reached before infeed begins.

During switch-on, the spindle follows the ramp in MP3411.x. During switch-off, it follows the ramp in MP3412.2:

- In MP3412.2, enter a multiplier for MP3411 during tapping.
- In MP3415.2, define the overshoot behavior of the spindle during tapping.
- Acknowledge the output of the M functions. An NC stop cannot be executed until a previous M function is acknowledged.

If the feed-rate and spindle ramps have differing gradients, the spindle follows the slower ramp.

Example:
Speed s = 1000 min^{-1}
MP3411.x = 0.025 [1000 min^{-1}/ms]

$$\frac{1000\ min^{-1}}{0{,}025 \cdot [1000\ min^{-1}/ms]} = 40\ ms$$

In this example the spindle was braked 40 ms before reaching the hole depth.

Delay times permit an optimum adjustment of the floating tap holder. You can delay the switch-off:

- In MP7120.2 enter a spindle slow-down time.

The delay cannot last longer than 30 ms before reaching the hole depth. Values above 30 ms are ignored (see the diagram).

You can delay a subsequent spindle start with M04:

- In MP7120.0 enter a dwell time. The ramp follows MP3412.2

You can delay restarting the infeed:

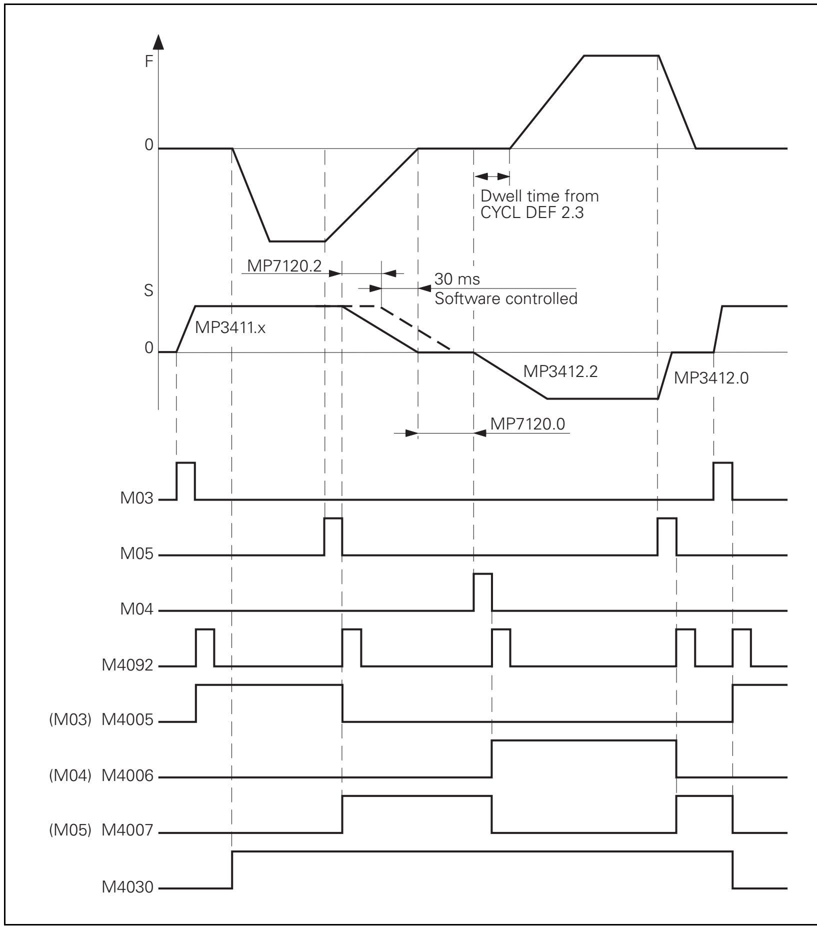
- Change the programmed dwell time in the cycle.

The NC uses M05 to switch off the spindle. The switch-off ramp follows MP3412.0. Then the spindle is switched back on with M03.

The feed rate override for tapping must be limited. Otherwise the floating tap holder may be damaged:

- Enter a limit in MP7110.x.

The following diagram shows the time sequence of the cycle:



		Set	Reset
M4030	**Cycle 2 or Cycle 17 active**	NC	NC

MP3412.2 **Multiplier for MP3411 during tapping**
Input: 0 to 1.999

MP3415.2 **Overshoot behavior of the spindle during tapping**
Input: 0 to 1000 [ms]

MP7110.0 **Minimum for feed-rate override during tapping**
Input: 0 to 150 [%]

MP7110.1 **Maximum for feed-rate override during tapping**
Input: 0 to 150 [%]

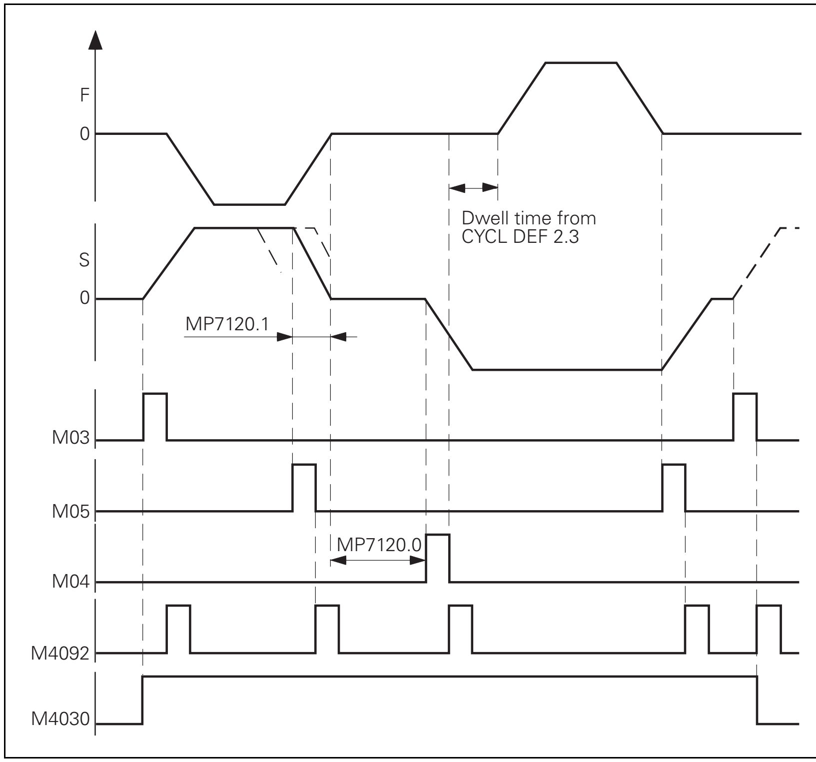
MP7120.0 **Dwell time for reversal of spindle rotational direction**
Input: 0 to 65.535 [s]

MP7120.2 **Spindle slow-down time after reaching the hole depth**
Input: 0 to 65.535 [s]

6.12.8 Tapping with floating tap holder and coded spindle-speed output

The dwell time for rotational direction reversal (MP7120.0) and the programmed dwell time have the same effect as the nominal speed value output.

The following diagram shows the time sequence of the cycle:



MP7120.1	**Advanced switching time of the spindle during tapping with coded spindle-speed output**
Input:	0 to 65.535 [s]

6.12.9 Rigid tapping

Cycle 17

- Define the rigid tapping process in the NC program with Cycle 17.
  While Cycle 17 is running, the iTNC automatically switches the tool axis to velocity feedforward mode.
- Define the dynamic response of the spindle and the machine tool axes in machine parameters. See "The Control Loop" on page 813 and „Spindle" on page 941.

With Cycle 17 the spindle can also be feedback-controlled. This results in a better speed curve:

- Set MP7160 bit 2 = 1 to drive the spindle under position feedback control with Cycle 17.

The tool axis can track the spindle or it can be interpolated with the spindle. Interpolation can result in higher speed stability of the tool axis. The path jerk (spindle and tool axis) can be set via MP3415.3:

$$r = \frac{a}{MP3415.3}$$

Whichever value is smaller from this formula and from MP1090.0 is valid.

- In MP7160, set bit 4 = 1 to interpolate the tool axis with the spindle.

With small thread depths and excessive spindle speeds it is possible that the programmed spindle speed may not be attained. The immediate transition from the acceleration phase to the braking phase can diminish the quality of the thread:

- Set MP7160 bit 1 = 1 in order to limit the spindle speed so that the spindle runs for about 1/3 of the tapping time at a constant speed.

During tapping, the position of the tool axis tracks the actual position of the spindle.

Please note that the use of acceleration feedforward control for the tool axis makes the tool axis sensitive to fluctuations in spindle speed caused, for example, by gear transmission. If this happens, the tool axis starts to run rough:

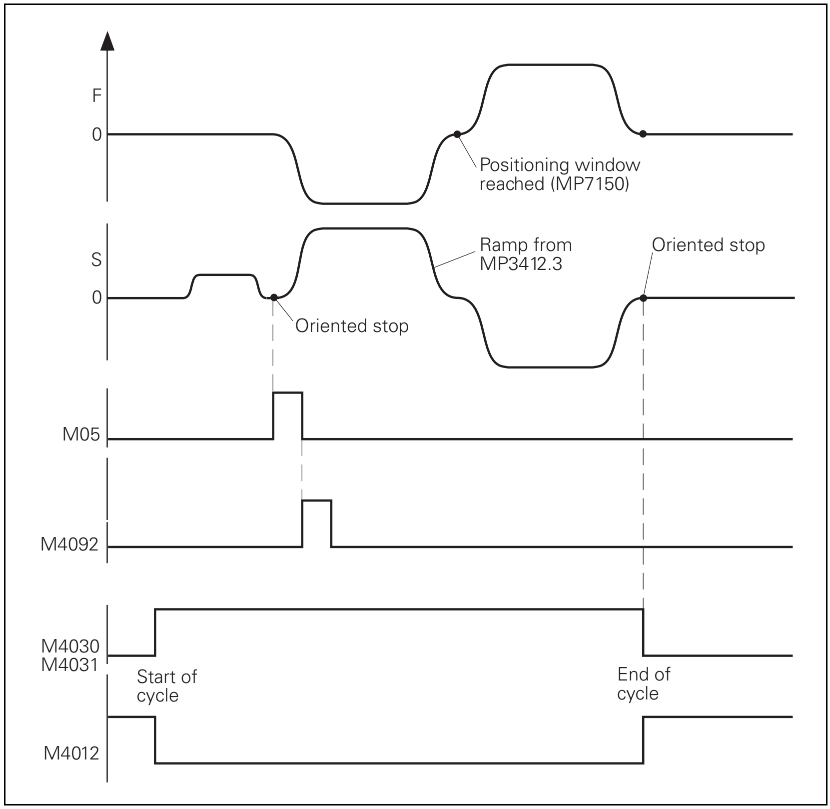
- In MP7160, set bit 3 = 1 to switch off acceleration feedforward control for Cycle 17.

Before tapping, the axes (e.g. Z and S) are synchronized through an oriented spindle stop, i.e., every Z position is assigned to a certain spindle angle. The NC orients the spindle. The NC sets M4017. The position control loop must be closed (M4012). Also see „Oriented Spindle Stop" on page 964.

Synchronization makes it possible to cut the same thread more than once. The assigned spindle angle depends on the thread pitch entered in the cycle. You can deselect this function to save machining time:

- Set MP7160 bit 0 = 1
  In this case you cannot cut the thread more than once.

M4031 and M4030 are set while the cycle runs.

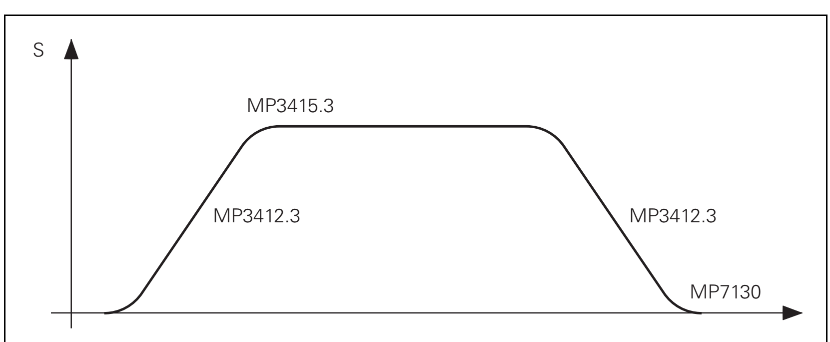


While Cycle 17 is running, the positioning window from MP7150 applies to the tool axis:

- Enter a value in MP7150 less than or equal to MP1030.x.

Define the acceleration and braking process of the spindle during rigid tapping:

- In MP3412.3 enter a multiplier for MP3411.x.
- With MP3415.3, define the overshoot behavior of the spindle.
- With MP7130, define the run-in behavior of the spindle.



MP3412.3 **Multiplier for MP3411.x for rigid tapping**
Input: 0 to 1.999

MP3415.3 **Overshoot behavior of the first spindle during rigid tapping**
Input: 0 to 1000 [ms]

MP7130 **Run-in behavior of the spindle during rigid tapping**
Input: 0.001 to 10 [°/min]

MP7150 **Positioning window of the tool axis during rigid tapping**
Input: 0.0001 to 2 [mm]

MP7160 **Spindle response during Cycles 17, 207 and 18**
Format: %xxx
Input: Bit 0 – Oriented spindle stop with Cycles 17 and 207
0: Oriented spindle stop before execution of the cycle
1: No oriented spindle stop before execution of the cycle
Bit 1 – Spindle speed
0: Spindle speed is not limited
1: Spindle speed is limited so that it runs with constant speed approx. 1/3 of the time
Bit 2 – Spindle in position feedback control
0: Spindle operated without position feedback control
1: Spindle operated with position feedback control
Bit 3 – Acceleration feedforward
0: Active
1: Not active
Bit 4 –
0: Tool axis tracks the spindle
1: Tool axis and spindle interpolated

		Set	Reset
M4030	**Cycle 2 or Cycle 17 active**	NC	NC
M4031	**Cycle 17 or Cycle 18 active**	NC	NC

Cycle 18

With Cycle 18 the tool axis tracks the actual position of the spindle. The starting position is the actual position. The target position is the hole depth:

- Program the approach and departure separately.

M4031 is set while Cycle 18 is running. M4012 must be reset for the cycle to be executed.

MP3412.3, MP3415.3, MP7130, MP7150 and MP7160 bit 1, bit 2 and bit 3 function as for Cycle 17.

6.12.10 Switching the modes of operation

For a spindle motor, two parameter sets with the same name can be saved in the motor table. This may be necessary if

- Another parameter set applies to a spindle motor at the higher speed range.
- A wye/delta connection switchover is carried out for a motor.

The switchover can be carried out during standstill or with a revolving spindle.

> **Danger**
>
> The contactor for the wye/delta switchover must not be switched under load!

As soon as the operating mode is switched with Module 9163, the NC switches the drive controller of the spindle off and activates the parameter set from the motor table and the machine parameters. You can check this with Module 9162. After the operating mode has been switched, you must reactivate the drive controller of the spindle with Module 9161.

To use the operating-mode switchover:

- Enter the two parameter blocks of your spindle motor with the same name in the motor table. Identify parameter block 1 by entering 0 in the **MODE** column, and parameter block 2 by entering 1.
- Switch between the two operating modes with Module 9163.
- With Module 9161, reactivate the drive controller.

For the two operating modes, you can use different machine parameters for the current and speed controller:

- In MP131.x you enter the y index of machine parameters MP2xxx.y for the current and speed controller in operating mode 0.
- In MP132.x you enter the y index of machine parameters MP2xxx.y for the current and speed controller in operating mode 1.

Module 9163 Switching the operating modes

Call:

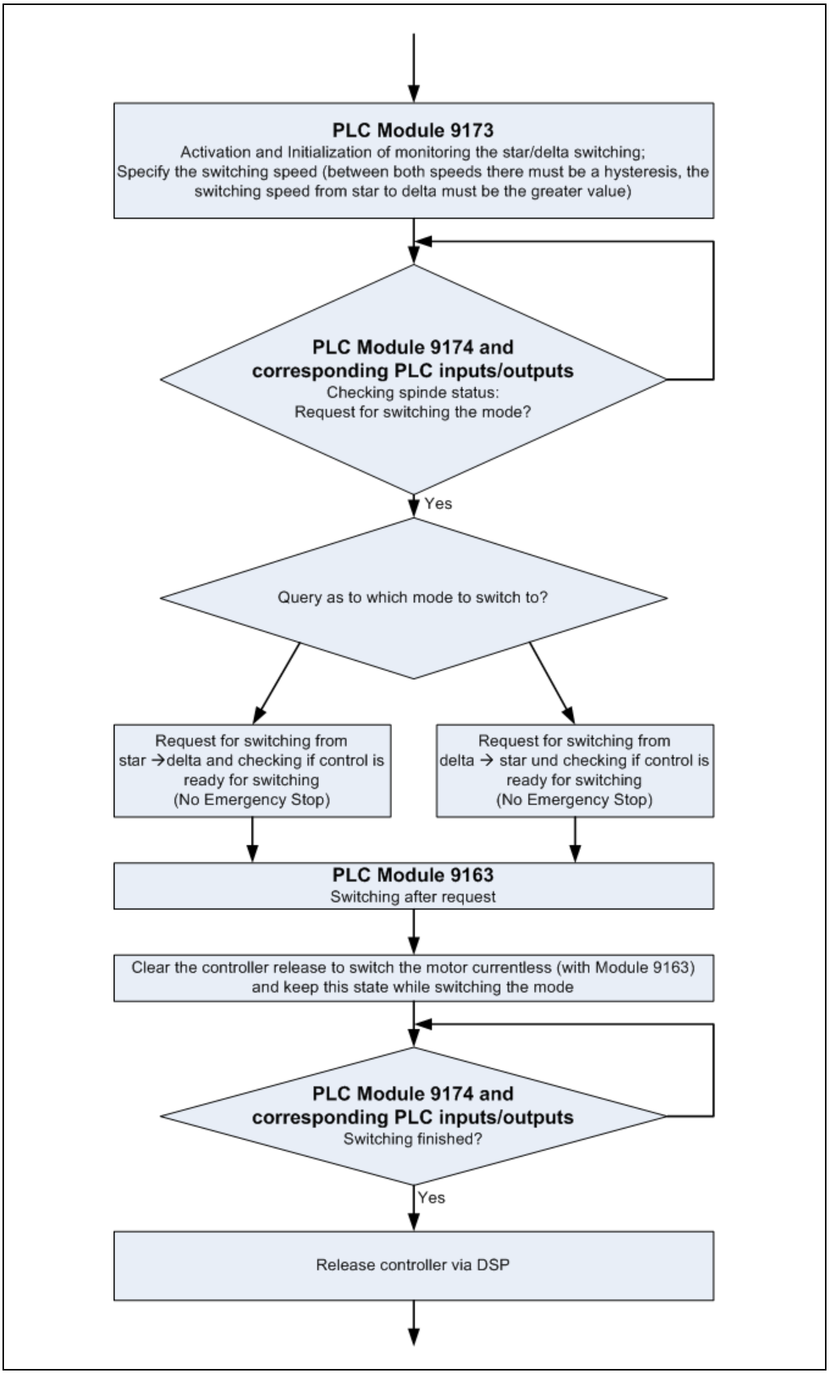
```
PS      B/W/D/K   <>Control loop>
                  15: Spindle
PS      B/W/D/K   <>Type of connection>
                  0: Operating mode 0
                  1: Operating mode 1
CM      9163
```

Error recognition:

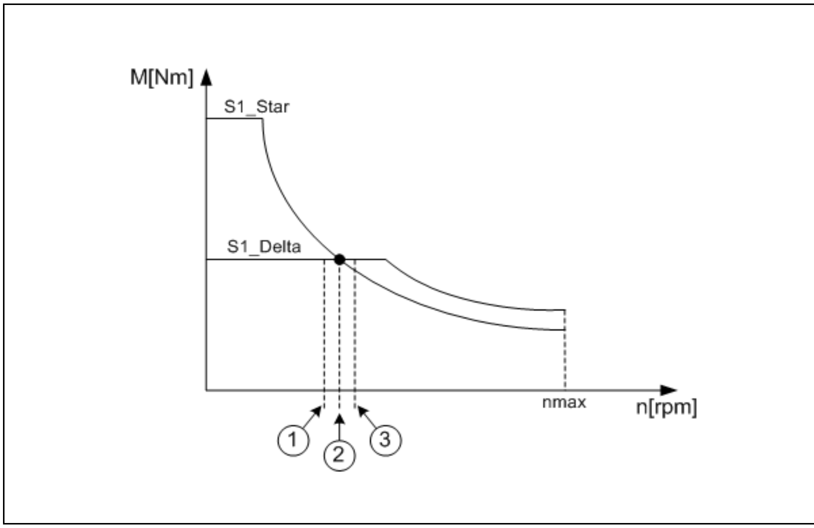
Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Switching not possible for this control loop
	2	Incorrect operating mode or incorrect control-loop number

Wye-delta connection switchover

As of software version 340 49x-03 it is possible to switch between wye and delta operation of the spindle without having to stop the spindle. However, only one gear stage can be used in order to use this direct switchover. This must be configured correspondingly via MP3010, MP3210 and MP351x. The flowchart shows the principle procedure for direct switchover of the modes via the PLC:



The diagram shows in principle how you can determine the switchover speeds for the wye/delta switchover:



- 1: Lower switching speed = n(intersection) – 2% to 4% of nmax
- 2: Speed at which the two characteristic curves intersect: n(intersection) at n(S1_wye) = n(S1_delta)
- 3: Upper switching speed = n(intersection) + 2% to 4% of nmax

Danger

The contactor for the wye/delta switchover must not be switched under load!

Attention

- On machines with transmission or wye/delta switchover, the RPM FACTOR (Q403) function of Cycle 209 (Tapping with chip breaking) must not be used with a value other than the default value. Switching the gear range or wye/delta switchover during retraction with the RPM FACTOR function may cause damage to the machine.
- Please inform the customers who are using these types of machines that the RPM FACTOR (Q403) function of Cycle 209 (Tapping with chip breaking) must not be used with a value other than the default value.

Module 9173 Speed-dependent wye/delta switchover

With Module 9173, a monitoring function dependent on the speed can be realized for the wye/delta switchover of the spindle. A switchover request is detected by Module 9174, which supplies the current status of the wye/delta operation. If a switchover is necessary, this is still done by Module 9163. The actual speed is assumed during the switchover. The readiness of the current controller is automatically rescinded when Module 9163 is called.

Condition:

- The module should not be called cyclically. A single call is enough for activation, deactivation or changing.

Additional information: Useful combinations for <Mode> parameter:

- xx0 Deactivate monitoring
- 001 Monitoring on, Switchover compatible
- 011 Monitoring on, Accelerated but safe switchover, Wait for readiness of current controller
- 111 Monitoring on, Fastest switchover possible

Call:

```
PS   B/W/D/K  <>Mode>
              Bit 0:
              0 = Monitoring off (function deactivated)
              1 = Monitoring on (automatic assumption of the actual
                  speed during switchover)
              Bit 1:
              0 = Switchover compatible to previous switchover;
                  switchover same as first switch-on of spindle
              1 = Switchover accelerated
              Bit 2:
              0 = Wait for readiness of current controller
              1 = Do not wait for readiness of current controller
PS   B/W/D/K  <>Switchover speed from wye to delta operation>
PS   B/W/D/K  <>Switchover speed from delta to wye operation>
CM   9173
```

Error recognition:

Marker	Value	Meaning
M4203	0	Speed-dependent monitoring active
	1	Error code in W1022
W1022	1	Invalid speeds (switchover speed of wye operation ≥ switchover speed of delta operation), or negative speed

Module 9174 Spindle status regarding wye/delta switchover

Module 9174 supplies the current spindle status for wye/delta operation. In order to receive one of the two "Request switchover x -> x" status values, the speed-dependent wye/delta switchover monitoring in Module 9173 must be active.

Call:

CM	9174	
PL	B/W/D/K	<>Status> 0: Spindle in wye operation 1: Request for wye -> delta switchover 2: Spindle in delta operation 3: Request for delta -> wye switchover

6.12.11 Operating a second spindle

With the iTNC 530 you can operate two spindles alternately, i.e., only one spindle can be active at a given time.
Both spindles can be operated as analog or digital spindles. If one spindle is to be operated as a digital spindle and the other one as an analog spindle, the first spindle must be operated as a digital spindle.

Assignment of encoder input and speed command output

The second spindle is driven instead of an axis, i.e., there are fewer axes available. An exception is analog operation of the second spindle without a position encoder. In this case all axes remain available. The assignment of position and speed encoder inputs as well as of speed command outputs is entered in MP111.x, MP113.x and MP121.x. See “Assignment for Axes" on page 655.

Note

If the speed encoder (with active reference mark monitoring, MP2220 bit 0) is disconnected and reconnected, the reference mark must be reevaluated (M4015) after the drive has been switched on again, otherwise the error message `Incorrect reference position` appears.

Switching between the spindles

You can switch between the two spindles through the PLC:

- Enter MP4020 bit 5 = 1 to activate double spindle operation.
- With Module 9175, switch between spindle 1 and spindle 2.

Module 9179 is used to determine the active spindle.

Commissioning the second spindle

- Digital second spindle: In MP10, deactivate one axis.
- Digital second spindle: Set MP110.x, MP112.x and MP120.x of the deactivated axis to zero.
- Machine parameters MP13010 to MP13520 are available for the second spindle. In their functions and input ranges, these parameters are identical with MP3010 to MP3520 for the first spindle. “Spindle" on page 941.
- Current and speed controller: For commissioning, use the machine parameters MP2040.x to MP2930.x. Determine the x index to be used for the second spindle with MP131.1 (for operating mode 0) and with MP132.1 (for operating mode 1).

MP4020 **PLC functions**
Format: %xxxxxxxx
Input: Bit 5 – Single- or double-spindle operation
0: Single-spindle operation
1: Double-spindle operation

MP13010 bis MP13520Machine parameter block for the second spindle
Input: Function and input range are identical to MP3010 to MP3520

Module 9175 Switch the spindle

With this module you can switch between spindle 1 and spindle 2.
When switching via an M strobe, MP7440 bit 2 must not be set. When switching via an S or G strobe, MP3030 or MP13030, respectively, must not be set. The module only needs to be called once.
Switching is only possible if

- the control is not in operation (M4176 is not set),
- the control is in operation (M4176 is set) and an M/S/T/T2/G strobe is active, or
- the machine is not currently approaching a contour. (M4157 not set)

Call:

PS	B/W/D/K	<>Spindle number>
		0: First spindle
		1: Second spindle
CM	9175	

Error recognition:

Marker	Value	Meaning
M4203	0	Indicated spindle activated
	1	Error code in W1022
W1022	2	Invalid spindle number
	6	M4157 = 1 (RESTORE POSITION active)
	20	Module was called in a spawn or submit job
	21	Missing strobe in M4176 = 1

Module 9179 Status information about spindle(s)

Status information about the spindles can be ascertained with Module 9179.

Call:

PS	B/W/D/K	<>Number of spindle>
PS	B/W/D/K	<>Code for status information>
		0: Active spindle ("Number of spindle" is not evaluated)
CM	9179	
PL	B/W/D	<>Status information>

Error recognition:

Marker	Value	Meaning
M4203	0	Status information has been ascertained
	1	Error code in W1022
W1022	1	Invalid code for status information
	2	Invalid spindle number

6.12.12 C-axis operation

In C-axis operation, an axis and a spindle are driven alternately by the same motor.
Digital or analog operation of axis and spindle is possible. Axis and spindle may each be equipped with one position encoder. Because the speed encoder is built into the motor, it measures both the axis and the spindle.

As of software version 340 49x-03, C-axis operation with the CC 424(B) is possible in the same manner as with the CC 422. The position encoder inputs of the CC 424(B) are used for C-axis operation.

Assignment of encoder inputs and speed command outputs to the axis and spindle:

- Enter in MP110.x the position encoder input of the axis (if present).
- Enter in MP111.x the position encoder input of the spindle (if present).
- Enter MP112.x = 0 for the axis (it uses the speed encoder of the spindle motor).
- Enter in MP113.x the speed encoder input of the spindle.
- Enter the same speed command output in MP121.x for the spindle and in MP120.x for the axis.

Commissioning of the axis and the spindle:

- The current and speed controllers are commissioned only for the spindle.
- The position controllers **must** be commissioned separately for the axis and spindle.

> Note
>
> The axis position controller should be commissioned in the gear range that is actually used for positioning.
> If possible, use the lowest gear range to ensure optimum control.
>
> C axis operation must be deselected for commissioning the spindle, meaning that no identical PWM outputs may be entered in MP120.x and in MP121.x.

If you use only one position encoder for both the spindle and the axis, the axis display keeps running while the spindle is in operation:

- Before switching from the axis to the spindle, save the actual position value of the axis with Module 9146. This ensures that the axis display remains at the last value, even when the spindle is rotating.
- Before switching from the spindle to the axis, recover the actual position value of the axis with Module 9146.

If you save the actual position value with Module 9146 and then close the position control loop, or if the position control loop is closed and the actual position value is then saved with Module 9146, the error message `Actual position value saved <Axis>` appears. The error message triggers an emergency stop.

Switching from **spindle to axis:**

- Stop the spindle.
- Switch to the gear range required for axis operation.
- Switch the spindle motor to the axis.
- With Module 9156, switch the axis from the open-loop to the closed-loop (servo-controlled) state.
- Enable the current and speed controls via Module 9161 with the corresponding bit for the axis.
- Release the axis clamping.
- Close the position control loop of the axis by setting the corresponding bits in W1040.
- Begin axis operation.

Switching from **axis to spindle:**

- Stop the axis.
- Clamp the axis.
- Open the position control loop of the axis by resetting the corresponding bits in W1040.
- With Module 9155, switch the feedback control for the axis off.
- Switch the spindle motor from the axis back to the spindle.
- With Module 9161 bit 15, release the current and speed controllers.
- Shift back to the original gear range.
- Start spindle operation.

Module 9146 Saving and reestablishing actual position values

Module 9146 saves and later reestablishes the actual position values of axes. If the actual position values were saved, the last value displayed remains until they are reestablished.

Call:

```
PS      B/W/D/K   <>Axes bit-encoded>
PS      B/W/D/K   <>Mode>
                  0: Save actual position values
                  1: Reestablish actual position values
CM      9146
```

Error recognition:

Marker	Value	Meaning
M4203	0	Actual position values saved or reestablished
	1	Error code in W1022
W1022	1	Invalid mode
	2	Invalid axes
	24	Module was called in a spawn job or submit job

Module 9155 Axis switchover from closed loop to open loop
With Module 9155 you can switch an axis from the closed-loop to the open-loop state. Now the bit can be transferred to the axis.

Call:
PS B/W/D/K <>Axes bit-encoded>
CM 9155

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing strobe or M4176 = 1
	24	Module was called in a spawn or submit job

Module 9156 Axis switchover from open loop to closed loop
With Module 9156 you can switch an axis from the open-loop to the closed-loop state. An automatic actual-to-nominal value transfer is executed. Now the bit can be transferred to the axis.

Call:
PS B/W/D/K <>Axes bit-encoded>
CM 9156

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing strobe or M4176 = 1
	24	Module was called in a spawn or submit job

6.13 Integrated Oscilloscope

The iTNC features an integrated oscilloscope.

This oscilloscope has six channels, of which no more than four can be used for signals from the current and speed controller. If more than four channels are to be displayed from the current and speed controller, the error message **`Channel <number> cannot be displayed`** appears.

The following signals can be recorded:

Signal	Meaning
Saved	The signal last recorded is displayed
s actual	Actual position [mm]
s nominal	Nominal position [mm]
s diff	Following error of the position controller [mm]
Volt.analog	Analog axis/spindle: Analog voltage = nominal velocity value [mV]
v actual	Actual value of the axis feed rate [mm/min]. Calculated from position encoder.
v nominal	Nominal value of the axis feed rate [mm/min]. Axis feed rate calculated from the difference from the nominal position values. The following error is not included.
Feed rate	Contouring feed rate [mm/min]
Position: A	Signal A of the position encoder
Position: B	Signal B of the position encoder
v (act rpm)	Shaft speed actual value [mm/min]; Calculated from rotary speed encoder and standardized with MP1054.
v (nom rpm)	Nominal velocity value [mm/min]: Output quantity of the position controller
I (int rpm)	Integral-action component of the nominal current value [A]; CC 422: peak value, CC 424(B): effective value
I nominal	Nominal current value [A] that determines torque; CC 422: peak value, CC 424(B): effective value
I actual	Actual current value [A]
Actl. Id	Actual magnetizing current value [A]
Max. Iq	Maximum torque current [A]
Compensat.	Position compensation value (composed of temperature compensation, axis-error compensation, backlash compensation, etc.)
PLC	The PLC operands (B, W, D, I, O, T, C) are recorded. Enter the operand in the text box next to PLC.
a nominal	Nominal acceleration value (m/s^2)
r nominal	Nominal jerk value [m/s^3]
Pos. diff.	Difference between position and speed encoder [mm]
a actual	Actual acceleration value [m/s^2]. Calculated from position encoder.
r actual	Actual jerk value [m/s^3]. Calculated from position encoder.
I2-t (mot.)	Current value of the I^2t monitoring of the motor [%]

Signal	Meaning
I2-t (pow. module)	Current value of the I^2t monitoring of the power module [%]
Utilization	Current utilization of the drive [%]
Block no.	Block number of the NC program
Gantry Diff	Difference between synchronous axes [mm]
U nominal	Nominal voltage [V]
P mech.	Mechanical power [W]
P elec.	Electrical power [W]
M actual	Actual torque value [Nm]
S noml (f.)	Nominal position as per nominal position value filter [mm]
DSP debug	Diagnosis function for internal purposes
Contour deviat.	Circular form test, contour deviation (mm)
F TCPM	Feed rate at the tool tip with TCPM
Int. diagn.	Reserved for internal purposes
DC-link P	If MP2198.x and MP2199.x are configured, the dc-link power of the power supply module for the respective axis can be displayed.
Amplitude	Amplitude of the position encoder
Motor A	Signal A of the speed encoder
Motor B	Signal B of the speed encoder
CC DIAG	The signal is axis-specific and connected with an additional input box on the right side of the oscilloscope screen. In the input box you can enter a number for selecting a channel of the DSP. See the following table for the available channels and signals.

The oscilloscope provides additional functions for commissioning the current controller. See “Commissioning" on page 998.

The recorded data remain stored until you start recording again or activate another graphic function.

CC DIAG

Available channels or signals of the CC-DIAG signal:

Number	Meaning
101	Status of emergency stop input (I32)
201	Masking the reference pulse via W1054
202	Activating the fine interpolation filter (nominal position value filter is active)
203	Position controller is closed W1040
204	Position control is fed forward (MP1392, MP 1392)
205	Deactivation of monitoring functions via W1042
206	Position control in semifeedforward mode (MP1396)
207	Rapid switch-off of speed controller (activated in MP4130)
208	DC-link voltage monitoring can be activated via PLC module 9167. The current status can be read via this channel.
300	Ready signal of power module (green LED on the power module)
301	Error signal of power module (triggers an error message)
302	Output for controlling the brake at the power module (X 344)
400	UV ready signal of the inverter
401	DC-link voltage too high
402	DC-link current too high
403	Excessive temperature of UV
404	Leakage current detected by UV
405	AC fail signal is active
600	I2T early warning threshold of drive was exceeded
601	Axis enabled (PLC module 9157)
602	Axis is active (PLC module 9162)
603	Commutation angle of motor was measured using the reference mark or EnDat
604	TNCopt function is currently being performed

Colors

- In MP7365.x, define the colors for the oscilloscope.

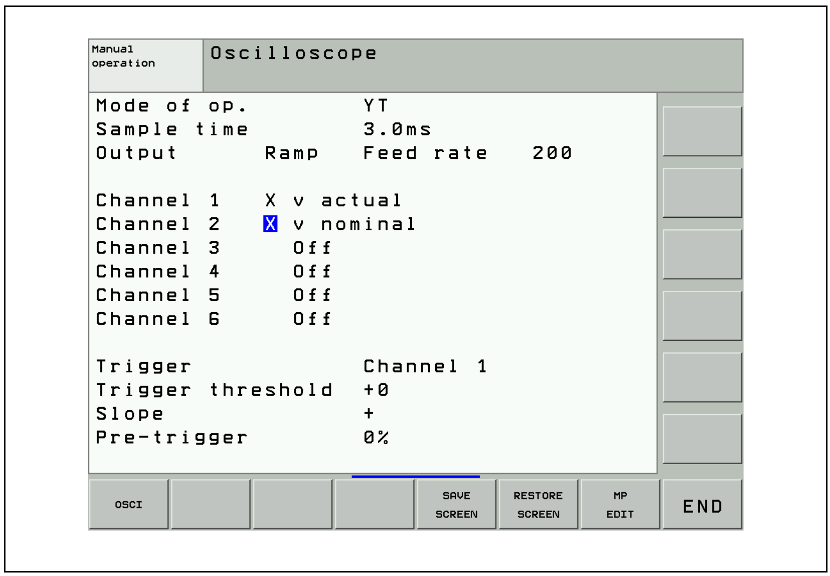
Setup

- Activate the oscilloscope with the DIAGNOSIS, DRIVE DIAGNOSTICS and OSCI soft keys.

> **Note**
>
> A step response is not possible. In order to make a step response possible, the keyword 688379 must have already been entered.

The setup menu appears:

- Choose the parameters to be entered with the cursor keys.



As of software version 340 49x-03 it is possible to operate the oscilloscope with reduced functions starting from the **`Power interrupted`** message. If you enter the code number **`DSP123`** before calling the oscilloscope, then starting from the **`Power interrupted`** message the data are recorded and can be displayed via the oscilloscope.

> **Note**
>
> Please note that the control has not finished booting at this time, and that the displayed values can therefore deviate from the correct or expected values.

Operating mode:

- Select the desired setting or choose the circular interpolation test
  - YT: Chronological depiction of the channels
  - XY: X/Y graph of two channels
  - CIRC: Circular interpolation test

Sample time:

▶ Set the time interval for recording the signals.
Possible entries: 0.6 ms, 1.8 ms and 3.6 ms
4096 samples are stored. The signals are therefore stored for the following duration:

- 0.6 ms · 4096 = 2.4576 s
- 1.8 ms · 4096 = 7.3728 s
- 3.6 ms · 4096 = 14.7456 s

Output:

▶ Select whether the nominal speed value is to be issued as a step or ramp.

- If you select ramp output, then the programmed feed rate, k_V factors, and acceleration values that you have specified with machine parameters go into effect.
- If you select step output, a step will be output as nominal velocity value when you press the axis-direction buttons in the **Manual operating mode.** During output, the position control loop is open. A step can be output only if code number 688379 or 807667 has been entered.

Feed rate:

▶ Enter the height of the step for the nominal velocity value (in mm/min). If you have defined a ramp as output, this field has no meaning.

Channel 1 to channel 6:

▶ Assign the channels of the recorded signals to the respective axes.

Trigger:

▶ Define the type of recording.
You have the following possibilities:

- **Free run:** The recording is started and ended by soft key. When you press the STOP soft key, the last 4096 events are saved.
- **Single shot**: When you press the START soft key, the next 4096 events are saved.
- **Channel 1** to **channel 6**: Recording begins as soon as the trigger threshold for the set channel is crossed.

Trigger threshold:

▶ Enter the trigger threshold (you will find the appropriate units in the signals table on 987):

Slope:

▶ Define whether recording will be triggered with the rising (positive) or falling (negative) edge.

Pre-trigger:
Recording begins at a time preceding the trigger time point by the value entered here

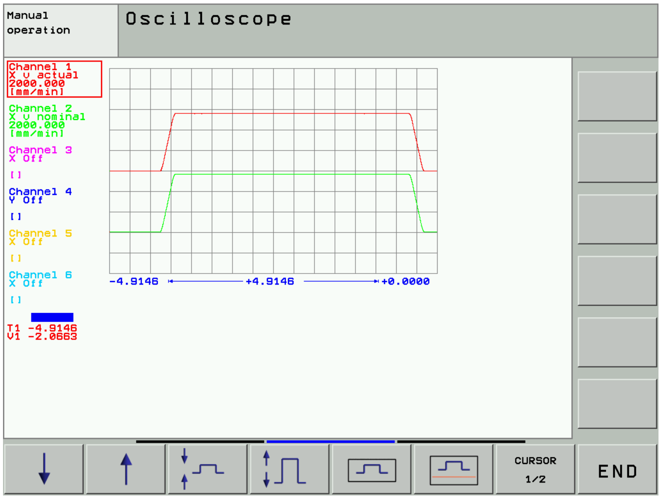
- Enter a value.

Delta trigger:
You can specify a second trigger threshold in the oscilloscope, which enables you to use a value range to define the event triggering a recording. Depending on the trigger edge and the first trigger value, you can determine whether a trigger signal is output when the value range is reached or exceeded. The inverse of the edge of the first trigger value is always selected as the active trigger edge of the delta trigger. The delta trigger is given as a value relative to the first trigger threshold. If a value of zero (0) is entered for the delta trigger (default setting), then the delta trigger is off:

- Enter the second trigger threshold (you will find the appropriate units in the signals table on 987):

Oscilloscope display:

- Press the OSCI soft key.



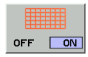
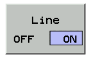
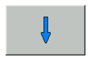
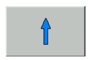
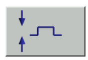
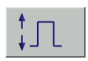
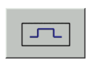
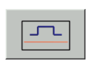
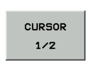
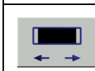
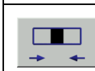
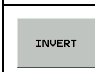
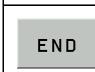
During recording, the selected signals are continuously displayed. After recording ends, the memory contents are displayed. In addition, the signal type and the resolution are displayed for each channel. The length of the recorded range, with respect to the entire memory content, is shown as a bar in the status field.

- Move the cursor with the arrow keys to select the channel. The amplitude of the selected channel as well as the time (with reference to start of recording) are shown in the status field.
- Activate a second cursor by pressing the **CURSOR 1/2** soft key. The oscilloscope displays the amplitude and time of this cursor. The time [s] of the second cursor is shown with respect to the time point of the first cursor. This function allows you to measure the acceleration time of an axis.

Note

The cursor is located on the trigger point only if it has not been moved after the measurement.
If the cursor has been moved, it will remain at the point of the time axis to which it has been moved. The cursor does not return to the trigger point until a trigger parameter has been changed.

Meaning of the soft keys:

Meaning of the soft keys:	
	Hide/show gridlines.
	Hide/show lines between measured points.
	Start recording. Recording is ended either by the trigger condition or with the STOP soft key.
	Move the signal down.
	Move the signal up.
	Decrease the vertical resolution.
	Increase the vertical resolution.
	Optimum vertical resolution. The signal is centered in the middle of the screen.
	Optimum vertical resolution. The signal is referenced to the datum line.
	Switch to second cursor.
	Move the signal to the left.
	Move the signal to the right.
	Decrease horizontal resolution.
	Increase horizontal resolution.
	Invert the signal.
	Exit the oscilloscope function.

Saving and loading a recording

You can display the signal last recorded for a channel again by selecting the **Saved** signal.

With the **SAVE SCREEN** soft key in the Setup menu you can save the recorded signals with all settings in a file on the hard disk. The file must have the extension *.DTA. This file can

- be recalled with the PLC development software PLCdesign
- be read back into the control.

In order to read a *.DTA file back into the control:

- Press the **RESTORE SCREEN** soft key in the Setup menu.
- Enter the complete file name and path of the *.DTA file.
- Press the ENT key.
- Press the **OSCI** soft key to displays the signals from the *.DTA file.

Oscilloscope via NC Program

As of software 340 49x-06, is possible to operate the integrated oscilloscope from the NC program. This makes it possible, for example, to automate certain machine analyses. When the oscilloscope is operated from the program, it behaves in the same way as during manual operation. No additional locking takes place: For example, active recording can be stopped with the STOP soft key, restarted, or the settings can be changed. Remote access to the internal oscilloscope during recording can also disturb the recording process. **FUNCTION SCOPE START** reserves the internal oscilloscope for the respective function. The oscilloscope is released automatically with **FUNCTION SCOPE STOP**, or at the end of the program (PGM END) and selection of a new NC program. To call the new functions, enter the code number 555343, press the FUNCTION SCOPE soft key and then the PROGRAM FUNCTIONS soft key:

- **FUNCTION SCOPE START JOB "INI file" RESULT "RES file"**
  FUNCTION SCOPE START JOB QS1 RESULT QS2
  This syntax starts the oscilloscope. The file specified in **JOB** initializes the oscilloscope (channels, measurands to be recorded, trigger conditions, sample time, ...). A *.DTA file is used for the initialization. This **INI file** can be created automatically. To do this, set the desired configuration in the internal oscilloscope and save the settings in a *.DTA file. Use the SAVE CONFIG. soft key in the oscilloscope to save the configuration. The file then only contains the current oscilloscope settings without the measured data. The path to which the oscilloscope is to save the measured data is indicated in **Result**. Both parameters are mandatory. However, you can specify them independently of each other as explicit text or as a QS reference.
  The INI file and the RES file must be entered with absolute paths. Both files must have the extension *.DTA for oscilloscope recordings. If an INI file already contains oscilloscope data, this data will be ignored during evaluation.
  This function reserves the internal oscilloscope, and therefore the oscilloscope cannot be started with the OSCI soft key.

- **FUNCTION SCOPE STOP Q1**
  FUNCTION SCOPE STOP
  This syntax switches the oscilloscope off again. If the execution of an NC program is to vary depending on whether a trigger condition has been met, a Q parameter can be transferred with the shutdown function. The oscilloscope management describes the value of the transferred parameter in order to give a feedback on the recording process. If the NC program is structured appropriately, and especially if a default value other than 1 is assigned to the parameter, this makes it possible for the user to derive whether a trigger condition has been met and to select appropriate branches that, for example, restart the oscilloscope or write the oscillograms to the RES file. The following parameter values can occur:
  0 = Recording had already been stopped before the **SCOPE STOP** function was triggered. The trigger condition had not been met since **SCOPE START**.
  1 = Recording was stopped by the **SCOPE STOP** function. The trigger condition had not been met since **SCOPE START**.
  2 = Recording had already been stopped before the **SCOPE STOP** function was triggered. The trigger condition had been met since **SCOPE START**.
  3 = Recording was stopped by the **SCOPE STOP** function. The trigger condition had been met since **SCOPE START**.
- **FUNCTION SCOPE STORE**
  If you call this function that does not contain parameters, the storage of the data in the buffer of the oscilloscope is forced from within the NC program. If a trigger condition has been fulfilled before the **STORE** function is called, the results for the trigger condition are written to the file indicated in **RESULT**. You need to stop recording before you can save the data.
  This function releases the internal oscilloscope, and therefore the oscilloscope can be started again with the OSCI soft key.

		Set	Reset
M4627	**Trigger condition for integrated oscilloscope**	NC	PLC
M4628	**Recording of the integrated oscilloscope ended**	NC	PLC

Circular interpolation test

A circular interpolation test can be run in the oscilloscope.

- Choose the **CIRC** operating mode in the oscilloscope.
- Select the **Contour dev.** setting for the appropriate axes in the two channels.
- Start recording.
- Start an NC program in which a circle is programmed. The circle center point must be at the origin of both axes.
- Stop recording.

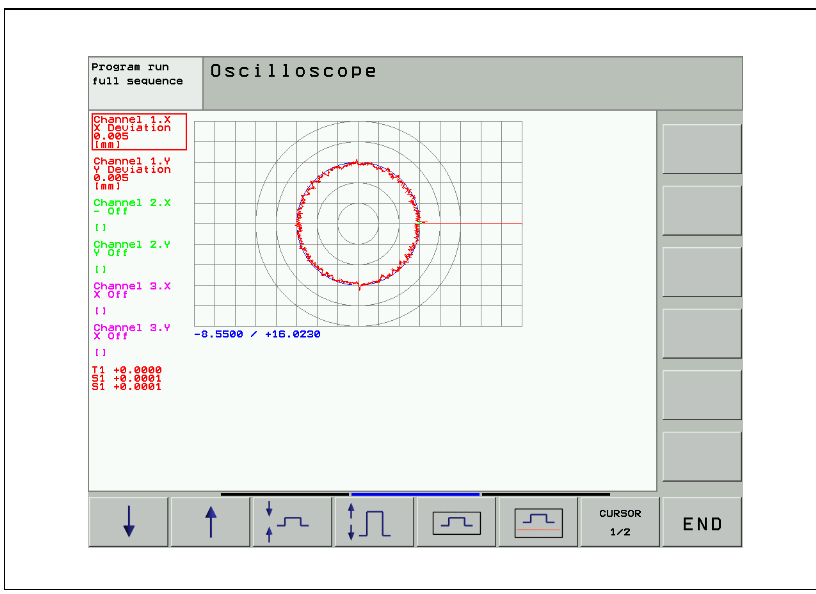
Below the grid, the recording time relative to the trigger time point is displayed.

Example of a circular interpolation test with the integrated oscilloscope:

Actual position:
X +30
Y +0

NC Program:

```
0 BEGIN PGM circular interpolation test MM
1 CC X+0 Y+0
2 CP IPA+360 DR+ F1000
3 M30
4 END PGM Circular interpolation test MM
```



6.14 Commissioning

6.14.1 Tables for power modules, supply modules and motors

In the machine parameter editor you select the installed motors, power modules and supply modules.

- Call a list of power modules, motors or supply modules with the corresponding soft key.
  In the list of motors, the type of motor (synchronous, asynchronous, or linear motor) and the operating mode are displayed in addition to the motor designation.

Meaning of the soft keys:		
▷		Call a list of power modules.
	BEGIN	Jump to the beginning of the list of power modules.
	END	Jump to the end of the list of power modules.
	PAGE ↓	Scroll one page forward in the list of power modules.
	PAGE ↑	Scroll one page backward in the list of power modules.
	SELECT AXIS	Select a power module with the arrow keys and transfer it with the corresponding STORE MP2100.X soft key.
	SHOW ACTIVE VALUE	Open the table of power modules in write-protected mode and jump to the selected power module.
	CHANGE ACTIVE VALUE	Open the table of power modules in editing mode and jump to the selected power module.
	END	Return to the machine parameter editor.

Meaning of the soft keys:		
M	Call the list of motors.	
	BEGIN	Jump to the beginning of the list of motors.
	END	Jump to the end of the list of motors.
	PAGE	Scroll one page forward in the list of motors.
	PAGE	Scroll one page backward in the list of motors.
	SELECT AXIS	Select a motor with the arrow keys and transfer it with the corresponding "STORE MP2200.X" soft key.
	SHOW ACTIVE VALUE	Open the table of motors in write-protected mode and jump to the selected motor.
	CHANGE ACTIVE VALUE	Open the table of motors in editing mode and jump to the selected motor.
	END	Return to the machine parameter editor.
	Call the list of supply modules.	
	BEGIN	Jump to the beginning of the list of supply modules.
	END	Jump to the end of the list of supply modules.
	PAGE	Scroll one page forward in the list of supply modules.
	PAGE	Scroll one page backward in the list of supply modules.
	SELECT PARAMETER INDEX	Select a supply module with the arrow keys and transfer it with the corresponding STORE MP2198.X soft key.
	SHOW ACTIVE VALUE	Open the table of supply modules in write-protected mode and jump to the selected supply module.
	CHANGE ACTIVE VALUE	Open the table of supply modules in editing mode and jump to the selected supply module.
	END	Return to the machine parameter editor.

After you have selected the motor and the power module, the models are automatically entered in MP2100.x, MP2200.x and MP2198.x.

A search function supports you in selecting motors, power stages and power supply modules for the machine parameters and also supports a text search function. The FIND soft key is available in the motor.mot, inverter.inv and supply.spy tables. In the "Find" pop-up window you can enter the complete or partial name of the device you want to search for.

If you use motors, power modules or supply modules that are not listed in the menus, please contact HEIDENHAIN.

You can overwrite standard data or add other models to the tables. If you change the table of the motors, power modules or supply modules, the changed tables are saved separately in the PLC partition, where they are used by the iTNC:

- **PLC:\MP\MOTOR.MOT** (motor table)
- **PLC:\MP\INVERTER.INV** (power module table)
- **PLC:\MP\SUPPLY.SPY** (supply module table)
- **PLC:\MP\MOTOR.AMP** (old power module table)

If at any time you want to use the HEIDENHAIN standard tables again, you must erase the above-mentioned tables in the PLC partition.

Note

Due to reasons of compatibility, the iTNC still supports the **MOTOR.AMP** power module table. However, it is not possible to use the higher currents of the HEIDENHAIN 1xx D inverters at the PWM frequencies 3.33 kHz and 4.0 kHz with this table.

The iTNC follows a specific sequence when searching for a valid table.

1st **PLC:\mp\inverter.inv**

2nd**PLC:\mp\motor.amp**

3rd **PLC:\mp\supply.spy**

4th **inverter.inv** HEIDENHAIN standard table for power modules on the system partition

5th**motor.amp** Previous HEIDENHAIN standard table for power modules on the system partition

6th **supply.spy** HEIDENHAIN standard table for supply modules on the system partition

If you use a motor that appears in the motor table, but only the data for the speed encoders differs, you can overwrite this data in the motor table with MP2202.x, MP2204.x and MP2206.x. The motor table is not actually changed. The changes only take place in the working memory.

Note

The original entry from the motor table is used only when MP2202.x = *, MP2204.x = * and MP2206.x = *.

After the commutation angle has been determined successfully, the **`8150 – Field angle ascertainment successful`** error message is now output. The drive remains controlled.

In MP2198.0 to MP 2198.4 you enter the types of power supply modules being used. This makes it possible to calculate the dc-link power for display in the oscilloscope.

In MP2199.x you assign the respective axis or spindle to one of the two power supply modules defined in MP2198.x.

MP2100.x **Type of axis power modules**
Input: Name from file <Motor.amp>

MP2198.x **Power supply model**
Input: Name from file <supply.spy>
Default setting: Empty string

MP2199.x **Assignment of the axes to the supply modules**
Input: 0: The axis/spindle is assigned to the UV in MP2198.0
1: The axis/spindle is assigned to the UV in MP2198.1

MP2200.x **Motor model**
Input: Name of the selected motor (entered by the iTNC)

MP2202.x **Overwrite "Line count" from the motor table**
Input: *: Entry from the motor table active
0: No speed encoder (volts-per-hertz control mode)
1 to 999 999

MP2204.x **Overwrite "Counting direction" from the motor table**
Input: *: Entry from the motor table active
+: Positive counting direction
–: Negative counting direction

MP2206.x **Overwrite "Type of encoder" from the motor table**
Input: *: Entry from the motor table active
0: No speed encoder (volts-per-hertz control mode)
1: Incremental rotary encoder with Z1 track
2: Absolute rotary encoder with EnDat interface (aligned)
3: Absolute linear encoder with EnDat interface
4: Linear motor with one reference mark (CC424(B))
5: Absolute rotary encoder with EnDat interface (not aligned)
6: Incremental rotary encoder without Z1 track
7: Incremental rotary encoder with distance-coded reference marks (nonaligned)
8: Incremental linear encoder with distance-coded reference marks (not aligned)

Entries in the power module table (inverter.inv)

- **NAME:** Designation of the power module
- PWM: PWM frequency in [Hz] at which the power module is driven
- **S:** Switch position of the current sensor. Is required for the D series of HEIDENHAIN inverters (UM 1xx D) in order to use the higher currents named above at the lower PWM frequencies.
  Input 0 or 1
- **I-MAX:** Maximum current of the inverter output in [A]
- **I-N:** Rated current of the inverter output in [A]
- **U-IMAX:** Current sensor voltage in [V] at I-MAX
- **I-N-DC:** Permissible continuous current in stationary rotating field or until F-DC is reached in [A]
- **T-DC:** Time constant, how long maximum current can be applied to a stationary synchronous motor in [s]
- **F-DC:** Lower motor base frequency down to which the motor can be loaded with I-N-DC in [Hz]
- **T-AC:** Cycle duration for the duty cycle S6-40% in [s]
- **F-AC:** Motor frequency from which I-MAX is permissible in [s]
- **T-IGBT:** Protection time of the IGBTs in [s]
- **R-SENSOR:** Resistance of the current sensor. The entries in this column are maintained by HEIDENHAIN and need not be changed by the machine tool builder.
- **F-Limit:** Lower frequency limit

Entries in the power module table (motor.amp)

- Designation of power module (NAME)
- Maximum current (I-MAX) in A
- Rated current (I-N) in A at a PWM frequency of 5 kHz
- Current sensor voltage (U-IMAX) in V at I-MAX
- Permissible continuous current in stationary rotating field or until F-DC is reached (I-N-DC) in A
- Time constant, how long maximum current can be applied to a stationary synchronous motor (T-DC) in seconds
- Lower motor base frequency down to which the motor can be loaded with I-N-DC (F-DC) in Hz
- Cycle duration for the duty cycle S6-40%(T-AC) in seconds
- Motor frequency from which I-MAX is permissible (F-AC) in seconds
- Protection time of the IGBTs (T-IGBT) in seconds
- Rated currents with PWM frequencies of 3333 Hz, 4000 Hz, 5000 Hz, 6666 Hz, 8000 Hz and 10000 Hz (I-N-AC-3333, I-N-AC-4000, I-N-AC-5000, I-N-AC-6666, I-N-AC-8000, I-N-AC-10000) in A

Entries in the motor table (motor.mot)

- Motor model (TYPE)
  - UASM = Uncontrolled asynchronous motor (volts-per-hertz control mode)
  - SM = synchronous motor
  - ASM = asynchronous motor
  - LSM = linear motor
- Designation of motor (NAME)
- Operating mode (MODE)
- Rated current (I-N) in A
- Rated voltage (U-N) in V
- Rated speed (N-N) in min^{-1}
- Rated frequency (F-N) in Hz
- No-load voltage (U0) in V
- No-load current (I0) in A
- Stator resistance cold (R1) in mΩ
- Rotor resistance cold (R2) in mΩ
- Stator leakage reactance (XStr1) in mΩ
- Rotor leakage reactance (XStr2) in mΩ
- Magnetizing reactance (XH) in mΩ
- Upper speed X-H characteristic (N-XH) in min^{-1}
- Threshold speed for field weakening (N-FS) in min^{-1}
- Maximum speed (N-MAX) in min^{-1}
- Factor for X-H characteristic (%-XH)
- Factor for stalling torque reduction (%-K)
- Number of pole pairs (PZ)
- Temperature coefficient (TK) in Ω/K
- Line count of the motor encoder (STR)
- Encoder being used (SYS)
  - 0 = No speed encoder (volts-per-hertz control mode)
  - 1 = incremental rotary encoder with Z1 track
  - 2 = Absolute rotary encoder with EnDat interface (aligned[1])
  - 3 = Absolute linear encoder with EnDat interface
  - 4 = Incremental linear encoder
  - 5 = Absolute rotary encoder with EnDat interface (not aligned[1])
  - 6 = Incremental rotary encoder without Z1 track (one reference mark)
  - Only CC 424(B): 7 = Incremental rotary encoder with distance-coded reference marks (not aligned[1])
  - Only CC 424(B): 8 = Incremental linear encoder with distance-coded reference marks (not aligned)
- Counting direction of the motor encoder (DIRECT.)
- Maximum temperature (T-MAX) in °C
- Maximum current (I-MAX) in A
- Rated power output (P-N) in W
- Motor mass moment of inertia (J) in kgm^2
- Inductance of the series reactor (L) in µH

---

1. See "Field Orientation"on page 1016.

- Thermal time constant for direct current (T-DC) in seconds
- Lower thermal limit frequency (F-DC) in Hz
- Thermal time constant for alternating current (T-AC) in seconds
- Upper thermal limit frequency (F-AC) in Hz; above this frequency, the maximum current I-MAX applies
- Thermal time constant for winding (Tth1) in s
- Thermal resistance between winding and lamination (Rth1) in K/W
- Thermal time constant for lamination (Tth2) in s
- Thermal resistance between lamination and coolant (Rth2) in K/W
- Stall torque (M0) in [Nm]
- Torque (Mmax) at Imax in [Nm]
- Series inductance (Ld) in [mH]
- Shunt inductance (Ld) in [mH]

Supply module table (supply.spy)

- Designation (NAME)
- Type of power supply module (E-R)
  0: No regenerative module
  1: Regenerative module
- Rated power output (P-N) in W
- Peak power (S6-40) (P-S6-40) in W
- Peak power for (0.2 s) (P-MAX02) in W
- DC-link voltage (UZ) in V
- Ratio of measuring voltage / UZ (UZ-AN) in V/V
- Ratio of measuring voltage / IZ (IZ-AN) in V/V
- Status signals (STATUS-SIG)
  Bit#0:AC-FAIL
  Bit#1:POWERFAIL
  Bit#2:TEMP
  Bit#3:READY
  Bit#4:Reserved
  Bit#5:Reserved
  Bit#6: Reserved
  Bit#7: Reserved
- Proportional factor of D controller (P-D)
- Integral factor of D controller (P-D)
- Proportional factor of Q controller (P-Q)
- Integral factor of Q controller (I-Q)
- PWM frequency (FREQ) in HZ

Series reactor

Reasons for using a series reactor:

Asynchronous motor

- Minimum leakage inductance of drive

The total leakage inductance L_{Total} of the drive is formed from the leakage inductance of the motor $L_{LeakMotor}$ and the inductance of the series reactor $L_{Reactor}$

$L_{Total} = L_{LeakMotor} + L_{Reactor}$, where:

$$L_{LeakMotor} = \frac{(X_{Str1} + X_{Str2}) \cdot 1000}{2 \cdot \pi \cdot f_N}$$

- X_{Str1}: Stator leakage reactance (see X_{Str1} in motor.mot) [mΩ]
- X_{Str2}: Rotor leakage reactance (see X_{Str2} in motor.mot) [mΩ]
- f_N: Rated frequency of the motor (see F-N in motor.mot) [Hz]
- $L_{LeakMotor}$: Leakage inductance of the motor [µH]

The total leakage inductance L_{Total} may not fall below a specified minimum value. This also depends on the PWM frequency f_{PWM}, the magnetizing current I_0 of the motor, and the dc-link voltage UZ, and is calculated as follows:

for $I_0 < 26$ A:

$$L_{Total} = \frac{700\ \mu H \cdot 5000\ Hz \cdot U_Z}{f_{PWM} \cdot 600}$$

for $I_0 \geq 26$ A:

$$L_{Total} = \frac{700\ \mu H \cdot 5000\ Hz \cdot U_Z \cdot 26\ A}{600 \cdot I_0 \cdot f_{PWM}}$$

- L_{Total}: Total leakage inductance of the drive [µH]

This way the value for the series reactor can be determined as:
$L_{Reactor} = L_{Total} - L_{LeakMotor}$

If the result is a negative value, then no series reactor is necessary.

- Impermissible temperature of the bearings

High harmonic losses in the rotor lead to excessively high bearing temperatures. One possible remedy, besides increasing the PWM frequency, is the use of a series reactor. Here there is no universally valid formula.
The amount of inductance results from the measurement of the bearing temperature.

- Limitation of dU/dt

Because of the strain placed on the motor insulation, the motor manufacturer can prescribe the use of a series reactor. The strain placed on the motor insulation results from the voltage build-up dU/dt at the inverter output.
For HEIDENHAIN inverters the voltage build-up is dU/dt = 6000V/µs.

Synchronous motor

■ Limit of current during field weakening operation

In order to ensure that the demagnetization current in the field weakening range does not exceed a permissible value of I_{DMax}, the total leakage inductance $L_{Total} = L_{Motor} + L_{Reactor}$ must have the following value:

$$L_{Total} = 10^6 \cdot \frac{\frac{U_0}{\sqrt{3}} \cdot \frac{n_{Max}}{n_N} - \frac{UZ}{\sqrt{6}}}{\frac{\pi}{30} \cdot I_{DMax} \cdot p_Z \cdot n_{Max}}$$

- U_0: No-load voltage of the motor (see U0 in motor.mot) [$Veff_{L-L}$]
- n_N: Rated speed of the motor (see N-N in motor.mot) [min^{-1}]
- n_{Max}: Maximum speed of the motor (see N-MAX in motor.mot) [min^{-1}]
- U_Z: dc-link voltage (see MP2190) [V]
- p_Z: Number of pole pairs (see PZ in motor.mot)
- I_{DMax}: Maximum demagnetization current [Aeff]
- L_{Total}: Total leakage inductance of the drive [µH]
- L_{Motor}: Total leakage inductance of the motor [µH]
- I_N: Rated current of the motor [Aeff]

The maximum demagnetization current is the no-load current of the motor during maximum speed, where $I_{DMax} \leq I_N$.

If the maximum demagnetization current is not known, it can be approximated by:

$$L_{DMax} = I_0 + \frac{I_N - I_0}{n_N} \cdot n_{Max}$$

- I_N: Rated current of the motor [Aeff]
- I_0: Continuous standstill current of the motor (see I0 in motor.mot) [Aeff]

and $I_0 \geq I_N$

■ Impermissible temperature of the bearings

High harmonic losses in the rotor lead to excessively high bearing temperatures. One possible remedy, besides increasing the PWM frequency, is the use of a series reactor.
Here there is no universally valid formula.

The amount of inductance results from the measurement of the bearing temperature. Furthermore it must be noted that, especially with synchronous spindles, the motor manufacturer requires a minimum PWM frequency and minimum inductance of the series reactor depending on this PWM value.

In this context it is important that the heating of (and possible damage to) the bearing and/or rotor cannot be detected by monitoring the motor winding temperature

Note

If required, use the speed-dependent PWM frequency switchover in order to minimize harmonic loss, see page 1078

- Compliance with the maximum permissible short-circuit current

The short-circuit current I_K is formed from:

$$I_K = \frac{U_0}{\sqrt{3} \cdot (X_{Str1} + X_H + 2 \cdot \pi \cdot f_N \cdot L_{Reactor})}$$

- U_0: Mesh no-load voltage (see U_0 in motor.mot) [$Veff_{L-L}$]
- X_{Str1}: Stator leakage reactance (see X_{Str1} in motor.mot) [Ω]
- X_M: Magnetizing reactance (see X_H in motor.mot) [Ω]
- f_N: Rated frequency of the motor (see F-N in motor.mot) [Hz]

This value may not exceed the maximum current I_{Max}: $I_K \leq I_{Max}$

Summary: Advantages and disadvantages of the series reactor:

- Advantages
  - Reduction of current harmonics
  - Reduction of losses in the motor (important for bearing temperature)
  - Reduction of the voltage build-up dU/dt at the motor winding
  - Reduction of the demagnetization current with synchronous motors in the field weakening range
  - Reduction of the short-circuit current with synchronous motors
- Disadvantages
  - Reduction of motor power at the voltage limit (field weakening range)
  - Heating of the series reactor
  - Space requirement

Note

If a series reactor is installed later, the current controller must be readjusted.

If a series reactor is used with another inductance than that shown in the motor table, the inductance value of the series reactor must be entered in MP2208.

The series reactor must fulfill the following specifications:

- The calculated inductance is required per phase.
  There are manufacturers who do not indicate the inductance per phase, but rather the mesh inductance (calculated from $Veff_{L-L}$ and Ieff).
- The series reactor must be designed for the requirements of the motor. (I_N at f_N and $I_N(N_{Max})$ at f_{Max} must be possible)
- The inductance must comply over the motor's operating range

MP2208.x **Inductance of the series reactor**
Input: * = Entry from the motor table active
Value of the series reactor in [µH]

Determining data for synchronous motors

The motor data for synchronous motors are entered in the motor table after some conversions using the values from the motor data sheet of the respective manufacturer (here using the example of a SIEMENS motor).

Values in the HEIDENHAIN motor table	Values from the motor data sheet
TYPE: SM	Permanently excited synchronous motor
NAME: 1FT6044-4AF7	1FT6044-4AF7
MODE: 0	
Rated current I-N in [A_{eff}] winding I-N: 2.9	Data sheet value Inoml (100 K) IN = 2.9 A
Rated voltage U-N in [V_{eff}] interlinked U-N: 341	Calculation from data sheet values n_{noml}, k_E, R_{Str}, $I_{noml\ (100\ K)}$, L_D: $\text{U-N} = \sqrt{3} \cdot \sqrt{(U_e + U_r)^2 + U_x^2}$ $U_e = (n_{noml} / 1000) \cdot (k_E / \sqrt{3})$ $U_e = (3000 / 1000) \cdot (108 / \sqrt{3})$ $U_e = 187.06\ V_{eff\ L,N}$ $U_r = RStr \cdot I_{noml\ (100\ K)}$ $U_r = 3.05 \cdot 2.9$ $U_r = 8.85\ V_{eff\ L,N}$ $U_x = 2 \cdot \pi \cdot (n_{noml} / 60) \cdot PZ \cdot (LD / 1.5) \cdot I_{noml\ (100\ K)}$ $U_x = 2 \cdot \pi \cdot (3000 / 60) \cdot 2 \cdot (0.016 / 1.5) \cdot 2.9$ $U_x = 19.44\ V_{eff\ L,N}$ $\text{U-N} = \sqrt{3} \cdot \sqrt{(187{,}06 + 8{,}85)^2 + 19{,}44^2}$ $\text{U-N} = 341\ V_{eff\ L,L}$
Rated speed N-N in [min^{-1}] N-N: 3000	Data sheet value n_{noml} N-N = 3000 min^{-1}
Rated frequency F-N in [Hz] F-N: 100	Calculation from data sheet value n_{noml} F-N = (n / 60) · PZ F-N = (3000 / 60) · 2 F-N = 100 Hz
No-load voltage at rated speed U0 in [V_{eff}] interlinked U0: 324	Calculation from data sheet value n_{noml} and k_Et U0 = (n_{noml} / 1000) · k_E U0 = (3000 / 1000) · 108 U0 = 324 $V_{eff\ L,L}$
No-load current I0 in [A_{eff}] winding I0: 3	Data sheet value $I_{0\ (100\ K)}$ I0 = 3 A_{eff}
Stator resistance at 20 °C R1 in [mΩ] at 20° C R1: 3050	Data sheet value R_{Str} R1 = 3050 mΩ
Rotor resistance at 20 °C R2 in [mΩ] at 20° C R2: 0	

Values in the HEIDENHAIN motor table	Values from the motor data sheet
Stator leakage reactance at F-N Xstr1 in [mΩ] Xstr1: 0	If nothing given, then zero.
Rotor leakage reactance at F-N Xstr2 in [mΩ] Xstr2: 0	
Magnetizing reactance XH for F-N at rated conditions XH in [mΩ] XH: 2295	Calculation from data sheet value L_D, n_{noml} $XH = 2 \cdot \pi \cdot (n_{noml} / 60) \cdot PZ \cdot (L_D / 1.5)$ $XH = 2 \cdot \pi \cdot (3000 / 60) \cdot 2 \cdot (0.016 / 1.5)$ XH = 6702 mΩ
Desaturation speed N-XH in [min^{-1}] N-XH: 0	
Rotational speed of beginning field weakening range N-FS [min^{-1}] N-FS: 0	
Maximum speed (mechanical) N-MAX in [min^{-1}] N-MAX: 180	Data sheet value *n* N-MAX = 180 min^{-1}
Saturation factor %-XH in % %XH: 100	
Stalling torque reduction factor %-K in % %-K: 100	
No. of pole pairs (half pole no. of motor) PZ PZ: 2	From data sheet value or model designation
Temperature coefficient of the stator winding TK in 1/K TK: 0.004	
Line count of the speed encoder STR STR: 2048	From the mounted speed encoder
Type of encoder SYS: 1	Aligned incremental encoder with distance-coded reference marks: 1 Aligned absolute encoder with EnDat interface: 2
Counting direction DIRECT. DIRECT.: +	
Max. temperature of motor at temperature feeler T-MAX in [°C] T-MAX: 150	
Maximum motor current I-MAX in [A_{eff}] winding I-MAX: 11	Data sheet value I_{max} I-MAX = 11 A_{eff}
Rated power P-N in [W] P-N: 9142	Calculation from data sheet value n_{noml} and $M_{noml\ (100\ K)}$ $P\text{-}N = 2 \cdot \pi \cdot (n_{noml} / 60) \cdot M_{noml}$ $P\text{-}N = 2 \cdot \pi \cdot (3000 / 60) \cdot 4.3$ P-N = 1351 W

Values in the HEIDENHAIN motor table	Values from the motor data sheet
Motor mass moment of inertia J in [kgm^2] J: 0.005	Data sheet value J_{mot} J = 0.00051 kgm^2
Inductance of the series reactor L in [mH] L: 0	
Thermal time constant for direct current T-DC in [s] T-DC: 2400	Calculation from data sheet value T_{th} T-DC = T_{th} ·60 T-DC = 40 ·60 T-DC = 2400
Lower thermal cutoff frequency F-DC in [Hz] F-DC: 0	
Thermal time constant for alternating current T-AC in [s] T-AC: 2400	Calculation from data sheet value T_{th} T-AC = T_{th} ·60 T-AC = 40 ·60 T-AC = 2400
Upper thermal cutoff frequency F-AC in [Hz] F-AC: 0	

6.14.2 PWM frequencies of the CC 422

For the CC 422, certain controller groups can be assigned different PWM frequencies via MP2180.x.

The PWM outputs of a controller group must be assigned the same PWM frequencies with MP2180.x. Otherwise, the DSP error message **`C440 PWM frequency <Axis> incorrect`** will appear.

- Controller group 1: X51, X53, X54
- Controller group 2: X52, X55, X56
- Controller group 3: X57, X59, X60
- Controller group 4: X58, X61, X62

▶ With MP2180.x, you can set the same PWM frequency for the PWM outputs of a controller group. The assignment between a PWM output and MP2xxx.y is done with MP120.x/MP121.x and MP130.x/MP131.x/MP132.x.

If PWM frequencies of > 5000 Hz are set for a controller group, it is no longer possible to use all PWM outputs of the controller group. Then only the first PWM output of the controller group can be used. The other PWM outputs must not be entered in MP120.x or MP121.x. Otherwise, the DSP error message **`C440 PWM frequency <Axis> incorrect`** will appear.

The following PWM outputs can be operated with a PWM frequency of > 5000 Hz:

- Controller group 1: X51 (but not X53, X54)
- Controller group 2: X52 (but not X55, X56)
- Controller group 3: X57 (but not X59, X60)
- Controller group 4: X58 (but not X61, X62)

▶ Set the required PWM frequency > 5000 Hz for the corresponding PWM output in MP2180.x. For the PWM outputs not used for the controller group, set MP2180.x = 0.

Attention

The following hardware version and later versions support the entry of different PWM frequencies for controller groups and of PWM frequencies > 5000 Hz:

- CC 422/6 control loops with ID 359 651-02
- CC 422/10 control loops with ID 359 652-02
- CC 422/12 control loops with ID 359 653-02

If you are using another hardware version, you must enter the same value (≤ 5000 Hz) in all MP2180.x.

MP2180.x **PWM frequency**

Input: 0: f_{PWM} = 5000 Hz (for HEIDENHAIN inverters)
3200 to 3999: f_{PWM} = 3333 Hz
4000 to 4999: f_{PWM} = 4166 Hz
5000 to 5999: f_{PWM} = 5000 Hz
6000 to 7999: f_{PWM} = 6666 Hz
8000 to 9999: f_{PWM} = 8333 Hz
10000: f_{PWM} = 10000 Hz

PWM frequency with INDRAMAT "POWER DRIVE" inverters

▶ In MP2180.x, enter the PWM frequency 4000 Hz.

PWM frequency with SIEMENS "SIMODRIVE" inverters

The iTNC 530 operates with a PWM frequency of 5 kHz. SIEMENS power modules are normally driven with a PWM frequency of 3.2 kHz (spindle) and 4 kHz (axes).
The rated current values I_N are defined for these frequencies. If power modules are operated with a higher PWM frequency (e.g. 5 kHz), high temperatures can be caused in these modules in some cases.

This applies particularly to these SIEMENS power modules:

- 6SN1123-1AA00-0CA0 (as axis module)
- 6SN1123-1AB00-0CA0 (as axis module)

Machines that are not under full load do not exceed the maximum permissible temperature.

There are two ways to prevent the undesired heating:

▶ In MP2180.x, enter the required PWM frequency (3200 Hz or 4000 Hz) or

▶ Reduce the factor for I^2t monitoring or

▶ Reduce rated current I_N in the table of power modules.

Note

- A reduction of the PWM frequency has no effect on the maximum rotational speed, but it means that the axes and the spindle(s) must be commissioned again.
- During derating the rated current I-N and the maximum current I-MAX are reduced by the same factor.

When a new machine is put into service, HEIDENHAIN recommends the PWM frequency appropriate for the axis modules (normally 4 kHz, see the SIEMENS documentation). If the power module of the spindle gets too warm in spite of a reduction of the PWM frequency from 5 kHz to 4 kHz, then the reference value for the I^2t monitoring or the rated current I_N must be reduced in the table of power modules.

Reduction of the reference value for I^2t monitoring or rated current I_N in the table of power modules

The reduction of the rated current I_N of the power modules, as well as the datum value for I^2t monitoring can be calculated from two values (X1, X2) that are given in the SIEMENS documentation.

The percent reduction of the rated current can be calculated with the following formula:

$$X_R[\%] = 100 - \left(\frac{(100 - X1) \cdot (8\ \text{kHz} - f_{PWM})}{8\ \text{kHz} - X2} + X1\right)$$

- X1 = Reduction factor of the current in % at a PWM frequency of 8 kHz
- X2 = PWM threshold frequency in kHz at which the electrical power reduction begins
- f_{PWM} = Frequency in kHz set in MP2180.x

This results in the reference value for I^2t monitoring:

$$X_B = 1 - \frac{X_R[\%]}{100}$$

- Reduce the rated current values I_N of your power modules in the power module table.

$$I_{Nneu} = I_N \cdot (100\ \% - X_R[\%])$$

or

- Reduce the reference value for the I^2t monitoring.

$$MP2302.x = X_B$$

Note

A reduction of the rated current of the power module can cause a reduction of the rated torque and, as a consequence, the rated power of the motor, if equal values for rated current of the power module and the rated current of the motor were chosen.

Example for a 50-A power module:

- Axis power module with 50 A, PWM frequency of 5 kHz, X1 = 40%, X2 = 4 kHz

$$X_R[\%] = 100 - \left(\frac{(100 - 40) \cdot (8\ \text{kHz} - 5\ \text{kHz})}{8\ \text{kHz} - 4\ \text{kHz}} + 40\right) = 15\ \%$$

$$X_B = 1 - \frac{15}{100} = 0,85$$

- Spindle power module with 50 A, PWM frequency of 5 kHz, X1 = 40%, X2 = 3.2 kHz

$$X_R[\%] = 100 - \left(\frac{(100 - 40) \cdot (8\ \text{kHz} - 5\ \text{kHz})}{8\ \text{kHz} - 3{,}2\ \text{kHz}} + 40\right) = 22{,}5\ \%$$

$$X_B = 1 - \frac{22{,}5}{100} = 0{,}78$$

- Axis power module with 50 A, PWM frequency of 4 kHz, X1 = 40%, X2 = 4 kHz

$$X_R[\%] = 100 - \left(\frac{(100-40)\cdot(8\ \text{kHz} - 4\ \text{kHz})}{8\ \text{kHz} - 4\ \text{kHz}} + 40\right) = 0\ \%$$

$$X_B = 1 - \frac{0}{100} = 1,00$$

- Spindle power module with 50 A, PWM frequency of 4 kHz, X1 = 40%, X2 = 3.2 kHz

$$X_R[\%] = 100 - \left(\frac{(100-40)\cdot(8\ \text{kHz} - 4\ \text{kHz})}{8\ \text{kHz} - 3{,}2\ \text{kHz}} + 40\right) = 10\ \%$$

$$X_B = 1 - \frac{10}{100} = 0,90$$

6.14.3 Field Orientation

Field orientation with absolute encoder (EnDat)

As soon as the absolute position of the encoder has been read, the absolute position and determined field angle are associated.
The field angle is associated with the zero position of the encoder.

Field orientation via encoder with Z1 track

After switching on the control, the motor orients itself (rough orientation) via the Z1 track of the encoder. The drive is ready for operation after this procedure. The field angle is determined and associated as soon as the reference mark is traversed during the first motor motion.

Field orientation via the PLC

If a synchronous drive is used along with an encoder, and the field angle cannot be determined directly via the encoder, there is no assignment between the encoder and rotor magnets.

The Field Orientation function, which must be carried out once per switch-on procedure, is used by the iTNC 530 to determine the association between the encoder and rotor magnets (field angle), and stores this information internally.

Note

Please pay attention to the following notes for determining the field angle:

- HEIDENHAIN generally recommends using either encoders with absolute values or encoders with Z1 tracks. With these encoders the field angles are available immediately after switch-on, so no additional efforts are necessary to determine them.
- The field angle can be determined only if the current controller is already adjusted!

Regarding the **motor.mot** motor table, the field orientation must be performed for the following drives:

- Linear motor with absolute encoder with EnDat interface **(SYS = 3)**
- Synchronous or torque motor with nonaligned rotary encoder with EnDat interface **(SYS = 5)**
- Synchronous or torque motor with incremental rotary encoder without Z1 track **(SYS = 6);** one reference mark per revolution
- Synchronous or torque motor with incremental rotary encoder with distance-coded reference marks **(SYS = 7)**
- Linear motor with incremental linear encoder with distance-coded reference marks **(SYS = 8)**

Methods of the iTNC for determining the field angle

The iTNC has various possibilities for orienting the field angles for the various controller units (CC). The following describes the methods for the CC 422 via MP2254.x (See "Field Orientation" on page 1101 for the CC 424(B)):

- **Automatic field orientation** (when starting the control; with motion of the motor)
- **Field orientation via soft key** (during commissioning; with motion of the motor). This method determines the field angle more accurately than the "Automatic field orientation."

Danger

- An encoder with absolute values or an encoder with a Z1 track must be used when determining the field angles for hanging axes (or braked axes that could move on their own).
- If the speed encoder is exchanged, the Field Orientation function must be rerun.

Note

For synchronous spindles, the field angle should be determined via the **FIELD ORIENT.** soft key, since this is a more exact determination.

Note

Standstill monitoring is active while during both methods for determining the field angle. If it responds, increase the threshold in MP1110.x. Afterwards, reset MP1110.x to the original value.
As of NC software 340 422-03 and 340 480-03, you can set the threshold for standstill monitoring during the field orientation separately in MP1120.x.

MP2254.x	**Determining the field angle**
Input:	0: Automatic field orientation, soft-key has no function 1: Only CC 422: Field orientation via soft key; motor motion is permitted
MP1120.x	**Standstill monitoring when determining the field angle**
Input:	0.0000 to 300.0000 [mm] or [°]

Automatic field orientation (MP2254.x = 0)

For motors that are configured in the motor table as listed above, the field angle can be determined for **supported axes** through a slight motion of the drive. This motion is so small that the field angle can be determined without any special precautions.

The field angle is determined automatically when the drive is first switched on. The motor moves back and forth for approximately four to six seconds. The **`Finding field angle`** message appears.

Do not brake the drive during this motion. The drive is ready for traverse after the field orientation has completed. The determined field angle is associated precisely and stored as soon as the reference mark is passed over during the first traverse motion.

Possible error messages

If the power module is not active before the determination of the field angle begins, the error message **`8B40 No drive release <axis>`** appears. If the power module switches off during the determination, **`8B50 Axis module <axis> not ready`** appears. At the same time, if the tolerance set in the standstill monitoring is too small (MP1110.x or MP1120.x), the message **`Standstill monitoring <axis>`** appears.

Field orientation via soft key (MP2254.x = 1)

By pressing the **`FIELD ORIENT.`** soft key once during **commissioning** of the motor. After pressing the soft key, the motor rotates. The field angle is determined and stored during this motion. This field angle is used when the motor is switched on again.

Procedure for determining the field angle via the FIELD ORIENT. soft key (MP2254.x = 1)

- Switch the control on.
- Do **not** acknowledge the **`Power Interrupted`** message. In the **`Programming and Editing`** mode of operation, use the MOD key to enter the code number **`688379.`** The oscilloscope is started.
- Press the **`I CONTROL`** soft key.
- In the **`Manual`** mode of operation, acknowledge the **`Power Interrupted`** message.
- Use the **`CHOOSE AXIS`** soft key in the oscilloscope to select the corresponding axis.
- Press the **`FIELD ORIENT.`** soft key.

  The PLC must

  - switch the drive on/off
  - release and lock the brakes.

The motor rotates back and forth for several minutes. The field angle is determined for the reference mark or datum, and is stored automatically. The **`Finding field angle`** message appears.

- Press the END soft key.

The control carries out a reset. Then the assignment of the field angle is available.

Saving the determined field angle

NC software: 340 420-xx, to 340 422-02 and to 340 480-03
The determined field angle is automatically stored on the hard disk.
If the Field Orientation function is not run, the following error message appears:

- Encoder with EnDat interface: **`8830 EnDat: no field angle <axis>`**
- Encoder without Z1 track: **`8820 Field angle unknown <axis>`**

NC software: as of 340 422-03, as of 340 480-03
The determined field angle is automatically entered in MP2556.x (See "Definition of the field angle" on page 1114).

For purposes of reliability and redundancy, either the serial number of the encoder (only for EnDat interface) or a unique control ID is entered as identification in MP2257.x.

If the current identification does not match the entry in MP2257.x, an error message appears:

- When using an encoder with EnDat interface, the error message **`8830 EnDat: no field angle <axis>`** appears. In any case the field angle must be determined anew, since the encoder does not match the field angle from MP2256.x
- When using an incremental encoder, the error message **`MP2257.<index> incorrect (ID=$<identification>)`** appears. The field angle from MP2256.x and the new identification **`(ID=$<identification>)`** for MP2257.x can only be assumed after determining that the same drive is meant (e.g. after changing controls).

Danger

In all other cases the field angle must be determined anew, since otherwise uncontrolled drive motions could occur!

MP2256.x	**Determined field angle**
Input:	0: Field angle does not need to be determined, or has not been determined
MP2257.x	**Control or encoder identification for the field angle from MP2256.x**
Input:	0: Field angle does not need to be determined, or has not been determined

6.14.4 Preparation

Proceed as follows:

- Check the wiring against the grounding diagram and the safety concept (See "Basic Circuit Diagram" at the end of Chapter 3).
- Check the control-is-ready function.
  See "EMERGENCY STOP monitoring"on page 932.
- Check the EMERGENCY STOP circuit by pressing the EMERGENCY STOP buttons and the EMERGENCY STOP limit switch.
- Select the current machine parameter file. Determine input values using the documentation on hand. Enter temporary values for machine parameters that must be optimized during commissioning.
- Create a PLC program for interfacing the control to the machine (use the PLC development software PLCdesign).
- Ensure that in the system file OEM.SYS the instruction **PLCMAIN=** refers to the current PLC program.

dc-link voltage

The iTNC 530 uses the dc-link voltage to calculate the maximum motor voltage:

- Enter in MP2190.x the level of the dc-link voltages at the power modules. Machine parameters 2190.x can be used to overwrite the entries of the SUPPLY.SPY table. The table is stored in the PLC:\mp\ directory.

MP2190.x **dc-link voltage U_Z**
Input: 0 to 3000 [V]
*: Entry from the power supply module table
HEIDENHAIN inverters:
Regenerative: 650 V
Non-regenerative: 565 V

Temporary input values

- Enter the following temporary input values when you begin commissioning:

MP	Temporary input value	Meaning
MP20.0	%00000000000000	Monitoring the absolute position of the distance-coded reference marks
MP1030.x	0.01	Positioning window
MP1054.x	?	Linear distance of one motor revolution (depends on the machine)
MP1090.x	1	Maximum permissible jerk on the tool path
MP1092	<greater than rapid traverse>	Feed rate threshold from which MP1090.1 becomes effective
MP1095	0	Single filter
MP1096.x	0	Position nominal value filter off
MP1099.0	5	Minimal filter order for single filters
MP1099.1	3	Minimal filter order for double filters
MP1110.x	2.0	Standstill monitoring

MP	Temporary input value	Meaning
MP1140.x	0.03	Movement monitoring (for digital axes the minimum value is entered)
MP1144.x	0	Motion monitor for position and speed
MP1340.x	0	No evaluation of reference marks
MP1396.x		Feedback control with velocity semi-feedforward
MP1410.x	0.5	Position monitoring in operation with velocity feedforward control (erasable)
MP1420.x	2	Position monitoring in operation with velocity feedforward control (EMERGENCY STOP)
MP1510.x	1	k_V factor for velocity feedforward control
MP1521.x	0	Transient response during acceleration and deceleration
MP1710.x	50	Position monitoring in operation with following error (erasable)
MP1720.x	50	Position monitoring in operation with following error (EMERGENCY STOP)
MP1810.x	1	k_V factor for control with following error
MP1820.x	1	Multiplier for the k_V factor
MP1830.x	100	Characteristic curve kink point
MP2220.x	%0000100	Monitoring functions
MP2420.x	0	Proportional factor of the current controller
MP2430.x	0	Integral factor of the current controller
MP2500.x	0.5	Proportional factor of the speed controller
MP2510.x	20	Integral factor of the motor speed controller (for axes with holding torque, e.g. vertical axes, the value 1 must be entered because otherwise the axis drifts away)
MP2512.x	0	Limit of integral factor of the speed controller
MP2520.x	0	Differential factor of the speed controller
MP2530.x	0	PT_2 element of speed controller
MP2540.x	0	Band-rejection filter damping
MP2550.x	0	Band-rejection filter for center frequency
MP2600.x	0	Acceleration feedforward control
MP2602.x	0	IPC time constant T_1
MP2604.x	0	IPC time constant T_2
MP2606.x	0	Following error in the jerk phase
MP2610.x	0	Friction compensation at low speed
MP2612.x	0	Delay of friction compensation
MP2620.x	0	Friction compensation at rated speed
MP2630.x	0	Holding current

Additional temporary input values for the spindle

- Enter the following additional temporary input values when you begin commissioning the spindle:

MP	Temporary input value	Meaning
MP3010.x	3 to 8	Output of speed, gear range
MP3020	991	Speed range
MP3411.x	1.999	Ramp gradient
MP3412.x	1	Multiplier for MP3411.x
MP3415.x	0	Overshoot behavior
MP3420	1	Positioning window
MP3440.x	1	k_v factor

Note

C-axis operation must be deselected for commissioning, meaning that no identical PWM outputs may be entered in MP120.x and in MP121.x.

Operating-mode switchover

During commissioning you can switch between operating mode 0 and operating mode 1 with the CONNECT STAR DELTA soft key. With Module 9168 you can interrogate the current settings in the PLC. You can switch the motor using PLC outputs and activate the corresponding machine parameters with Module 9163:

- Perform the adjustment for operating mode 0 and operating mode 1.
  - If you do not use operating mode 1, set the corresponding machine parameters to zero.

6.14.5 Commissioning digital control loops with TNCopt

In order to commission digital control loops with TNCopt, you must carry out preparations on the control as described in this chapter. Also pay attention to the notes in the documentation for TNCopt. Functions not supported by TNCopt must be commissioned manually (See "Commissioning of digital axes" on page 1026 and See "Commissioning the digital spindle" on page 1054).

When commissioning with TNCopt, the machine axes must be moved. For safety purposes, a function of this type should be enabled on the control. The entry **TNCOPT.LOCKSOFTKEYVISIBLE = YES** in OEM.SYS makes the soft key **TNCOPT OFF ON** visible after pressing the MOD key. It is used to enable such functions. As a default, the soft key is always set to **OFF** when the control is started up.

Current controller

- Switch on the control.
- Do **not** acknowledge the **Power Interrupted** message. In the **Programming and Editing** mode of operation, use the MOD key to enter the code number 688379 or 807667 (followed by the DIAGNOSIS and DRIVE DIAGNOSTICS soft keys) to switch to the **Drive Diagnostics** mode of operation.
- Press the I CONTROL soft key.
- Acknowledge the **Power Interrupted** message in the **Manual** mode of operation.
- Switch on the control voltage.
- Switch to the **Oscilloscope** mode of operation.
- Press the START STEP soft key.
- Commission the current controller with TNCopt.

Speed controller

- Position the axis or spindle to be optimized at a location where it can be commissioned safely.
- Set MP1340.x = 0 to deselect evaluation of the reference marks.
- Ensure that the loaded PLC program fulfills the following conditions:

  - Position control loop is opened, because the NC opens the position control loop only during the step function. If the position controller is not optimized, error messages appear if the position control loop is closed.
  - Enable the drive controller
  - NC stop inactive
  - Axis direction buttons active
  - Axes clamped

- In the **Programming and Editing** mode of operation, use the MOD key to enter the code number 688379 to switch to the **Drive Diagnostics** mode of operation.
- Press the OSCI soft key.
- Set the following values in the **Oscilloscope:**
  Output: Step
  Feed rate: 100
  Channel 1: I nominal
  Trigger: Free run
- Press the OSCI soft key to switch the curve representation.
- Press the START soft key to start recording.
- Set the feed-rate override potentiometer to 100%.
- Commission the speed controller with TNCopt.

Feedforward

- Position the axes to a location where the feedforward functions can be commissioned safely.
- Set the datum for the affected axes at this location to "0".
- In MP1060.x, set the acceleration to 0.5.
- In the **Program Run, Full Sequence** mode of operation, select the NC program **FF_*.H** (* = axis to be optimized) from the **TNC:\TNCOPT** folder.

> Note
>
> TNCopt generates the NC programs FF_*.H with the feed-rate values 6000 and 200. The larger feed-rate value should equal the machine's highest machining feed rate. Adjust them if necessary. The lower feed-rate value must not be changed.

- Set the feed-rate override potentiometer to 100%.
- Commission the feedforward functions with TNCopt.

Reversal spikes

- Position the axes to a location where the reversal-spike compensation can be commissioned safely.
- Set the datum for the affected axes at this location to "0".
- In MP1060.x, set the acceleration to 0.5.
- In the **Program Run, Full Sequence** mode of operation, select the NC program **CIR_*#.H** (* = axis to be optimized; # = second control loop, defines the plane) from the **TNC:\TNCOPT** folder.
- Set the feed-rate override potentiometer to 100%.
- Commission the reversal-spike compensation with TNCopt.

IPC and k_V factor

- Position the axes to a location where the IPC and k_V factor can be commissioned safely.
- Set the datum for the affected axes at this location to "0".
- Set the k_V factors in MP1510.x to 1.
  You can also start with a higher value for MP1510, as long as the value is safely below the oscillation limit.
- In the **Program Run, Full Sequence** mode of operation, select the NC program **IPC_*.H** (* = axis to be optimized) from the **TNC:\TNCOPT** folder.
- Set the feed-rate override potentiometer to 100%.

Note

Ensure that the machine parameters for the IPC (MP2602.x, MP2604.x and MP2606.x) have been set to 0.

- Adjust the IPC and k_V factor with TNCopt.

6.14.6 Commissioning of digital axes

The iTNC must be adjusted in sequence for the:

- Current controller
- Speed controller
- Position controller

The signals that you need are recorded with the integral oscilloscope.

Current controller

Use the integrated oscilloscope to adjust the current controller. The speed and position control loops are open when you adjust the current controller. You must therefore activate a special PLC commissioning program:

- Enter the name of this PLC program in the OEM.SYS file with the `PLCPWM =` instruction.

It suffices to program an EM (end module).

The drive must be enabled externally and the iTNC needs the "ready" signal.

As soon as the PLC program defined with `PLCPWM =` is active, you can use Module 9168 to interrogate the commissioning status.

Module 9168 Interrogating the commissioning status

Call:

CM	9168	
PL	D	<>Status> –1: Commissioning not active or as yet no axis is selected Bits 0 to 5 represent selected axes 1 to 6 Bit 15 – Spindle selected Bit 16 – Operating mode of spindle 0: Operating mode 0 1: Operating mode 1

Adjusting the current controller:

- Switch on the control.
- Do **not** acknowledge the **Power Interrupted** message. In the **Programming and Editing** mode of operation, use the MOD key to enter the code number 688379 or 807667 (followed by the DIAGNOSIS and DRIVE DIAGNOSTICS soft keys) to switch to the **Drive Diagnostics** mode of operation.
- Press the I CONTROL soft key.
- In the **Manual** mode of operation, acknowledge the **Power Interrupted** message.
- Use the CHOOSE AXIS soft key in the oscilloscope to select the axis to be adjusted.
- With the FACTOR P/I soft key, select the I factor and set MP2430.x = 0.
- With the FACTOR P/I soft key, select the P factor.
- Calculate the starting value of the P factor with the following formula:

$$\text{Starting value} = \frac{100\,000 \cdot L}{T_a}$$

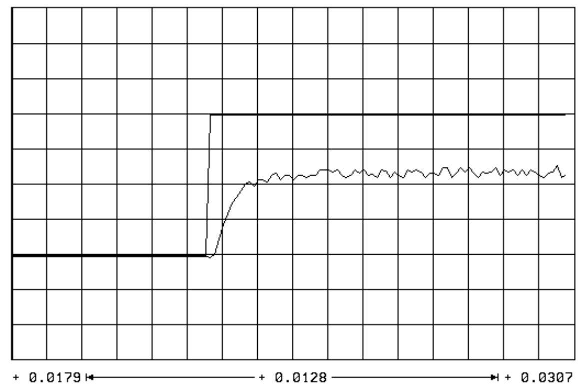
T_a	f_{PWM} **(MP2180.x)**
150	3 333 Hz
120	4 166 Hz
100	5 000 Hz
75	6 666 Hz
60	8 333 Hz
50	10 000 Hz

Synchronous motor: $$L = \frac{XH}{2 \cdot \pi \cdot (F\text{-}N) \cdot 1000}$$

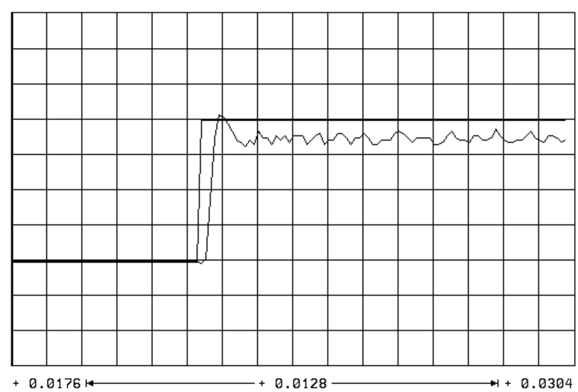
Asynchronous motor: $$L = \frac{XStr1 + XStr2}{2 \cdot \pi \cdot (F\text{-}N) \cdot 1000}$$

The values for XH (magnetizing reactance), F-N (rated frequency), XStr1 (stator leakage reactance) and XStr2 (rotor leakage reactance) can be found in the motor table. Switch to the editing mode of the motor table (APPEND MOTOR soft key).
The values for XH, XStr1 and XStr2 are specified in [mΩ] in the motor table. Use these values in the formulas. The formula already contains the conversion factor.

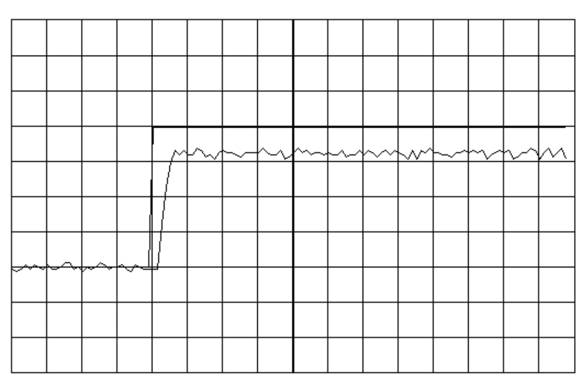
- Set this P factor (MP2420.x) with the ↑ soft key.
- Press the START STEP soft key.
  This sends a step function to the current controller and measures the step response. The height and length of the step function are automatically calculated by the iTNC.
- With the ↑ soft key, increase the P factor (MP2420.x) step by step until just barely no undershoot is visible.



MP2420.x too small

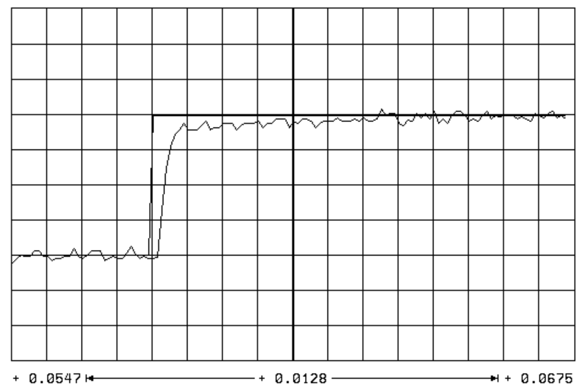


MP2420.x too large

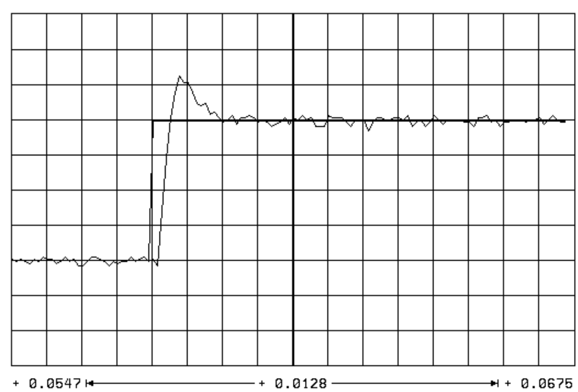


MP2420.x at optimum setting

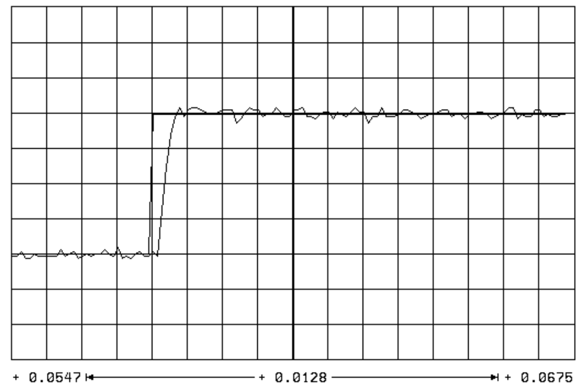
- Save this value with the STORE MP2420.x soft key.
- With the FACTOR P/I soft key, select the I factor.
- With the ↑ soft key, increase the I factor (MP2430.x) step by step until
- Spindle: Just barely no overshoot is visible, and so that the nominal value is reached as quickly as possible (short rise time).
- Axes: You see an overshoot but no undershoot.



MP2430.x too small



Axes: MP2430.x at optimum setting



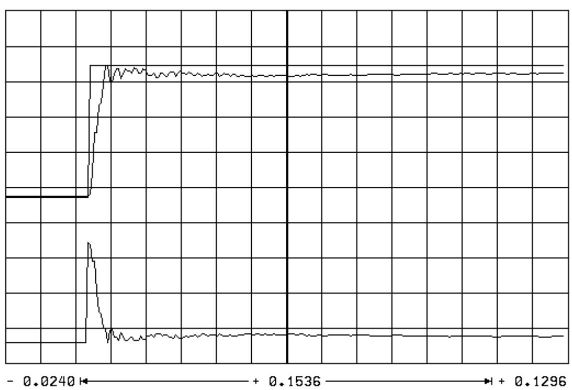
Spindle: MP2430.x at optimum setting

- Save this value with the STORE MP2430.x soft key.
- Press the END key to exit the **Commission Current Controller** mode.

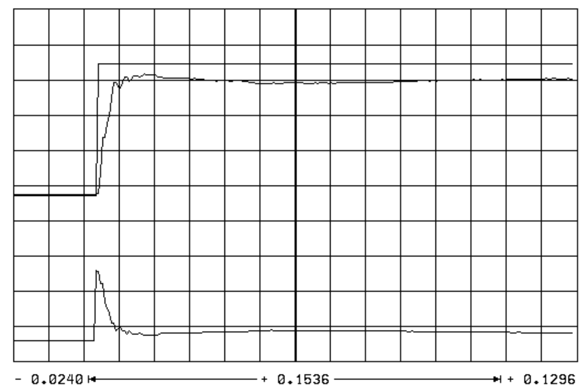
Speed controller

Adjusting the speed controller:

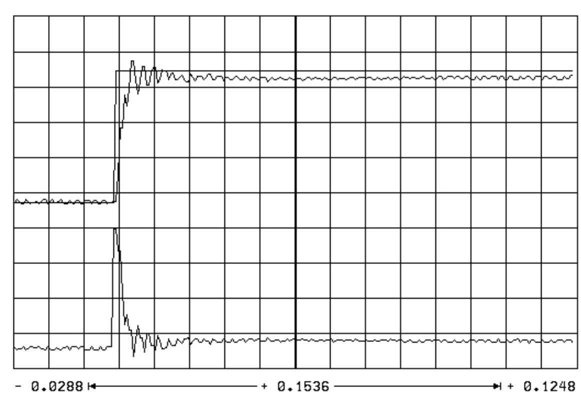
- Deselect "Pass over reference point" by setting MP1340 = 0.
- Ensure that the loaded PLC program fulfills the following conditions:
  - Position control loop is opened (W1038/W1040), because the NC opens the position control loop only during the step function. If the position controller is not optimized, error messages appear if the position controller is closed.
  - Servo drive controller is enabled (Module 9161)
  - NC stop is inactive, MP4560 = 1
  - Axis direction buttons active
  - Axes clamped
- In the MANUAL mode, use the oscilloscope function to select a step function (approx. 500 mm/min) that will not overdrive the speed controller, i.e. that does not limit I NOMINAL.
  Display the nominal velocity value V (NOM RPM), the actual speed value V (ACT RPM) and the nominal current value (I NOMINAL).
- Activate the step function with the axis direction buttons.
- To change the machine parameters, press the MP EDIT soft key in the setup menu.
- Increase MP2500.x (P factor) until the oscillation limit is reached.



MP2500.x at the oscillation limit



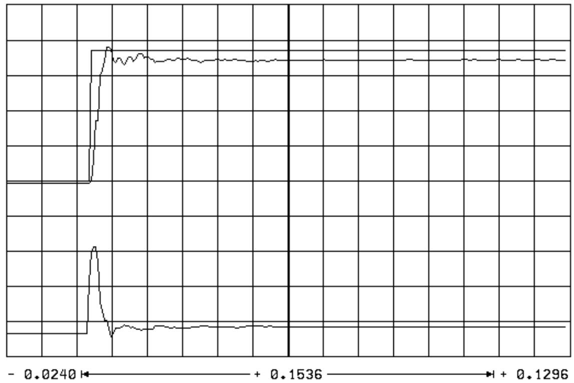
MP2500.x too small



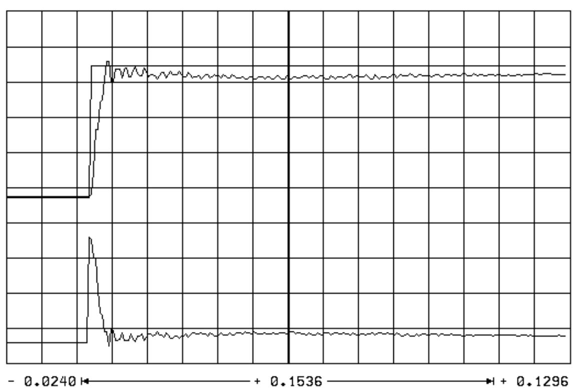
MP2500.x too large

- Input value for MP2500.x = <determined value> · 0.6

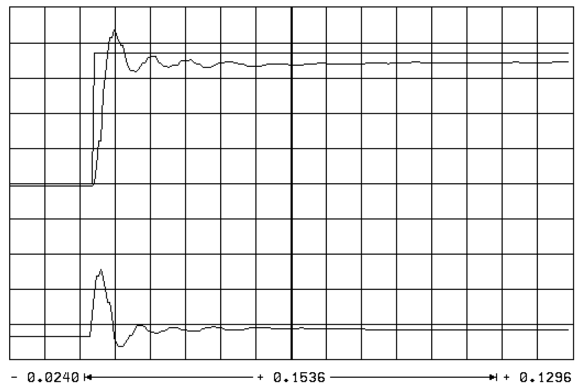
▶ Compensate high-frequency interference oscillations (> 400 Hz) with MP2530.x or MP2560.x.



MP2530.x at optimum setting



MP2530.x too small



MP2530.x too large

Attention

You can use MP2520.x to compensate low-frequency oscillation (< 100 Hz) on axes with mechanical problems.

However, HEIDENHAIN recommends that you avoid using MP2520.x if possible.

Do not use for axes with belt drive!

You can also compensate disturbance oscillations with the band-rejection filter:

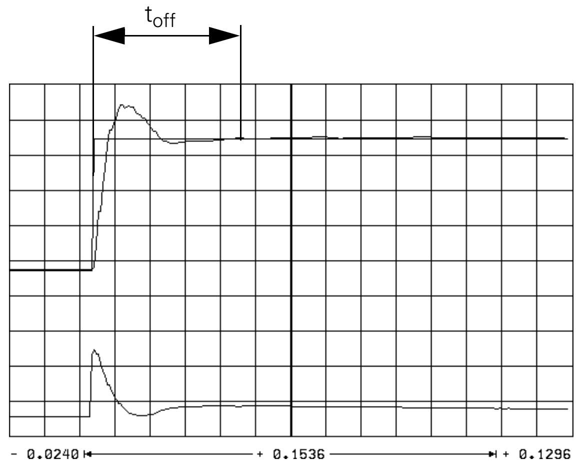
- Calculate the frequency of the oscillation and enter it in MP2550.x.
- Increase the band-rejection filter damping in MP2540.x until the interfering oscillation is minimized. Realistic input values: 3 to 9 dB.

Note

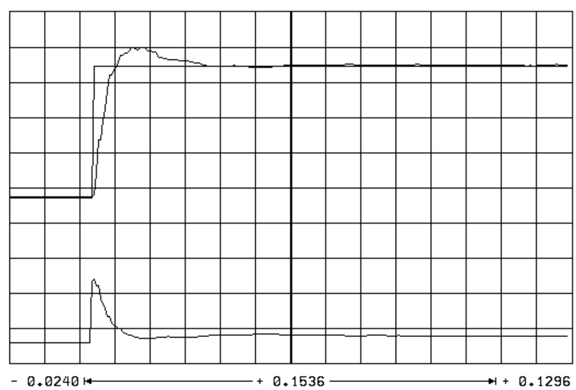
The compensation dampens the control loop. Try first to remove the mechanical causes of the disturbance oscillations.

To reduce the occurrence of disturbance oscillations, HEIDENHAIN recommends the use of motor couplings with a low tendency to oscillate (e.g. from the Rotex Company).

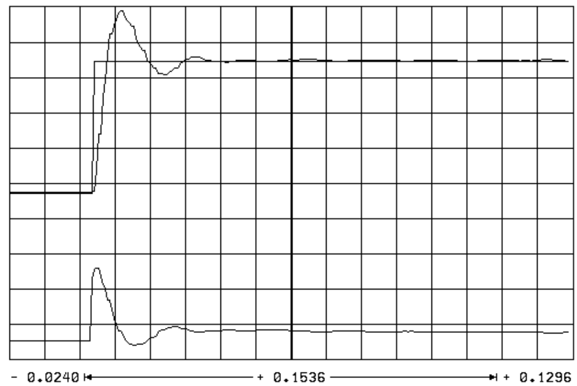
- Increase MP2510.x (I factor) until you see one overshoot followed by a slight undershoot and the settling time t_{off} is as small as possible (realistic value: 3 ms to 15 ms).



MP2510.x at optimum setting



MP2510.x too small



MP2510.x too large

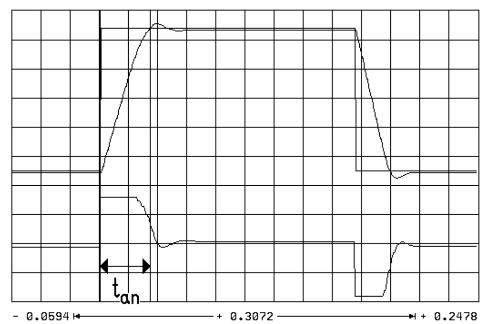
Determining the acceleration

- Load the machine with the maximum permissible weight.
- Enter the rapid traverse rate as step height
- During the step response, record the step response of the nominal velocity value V (NOM RPM), the actual speed value V (ACT RPM), and the nominal current value (I NOMINAL). It is permissible to limit the nominal current value I NOMINAL during acceleration.
- From the step response of the speed controller you determine the maximum possible acceleration (incl. 10% safety margin).

$$a = \frac{F_{max}}{t_{an} \cdot 66\,000}$$

a: Acceleration [m/s^2]
F_{max} : Maximum machining feed rate (MP1010.x) [mm/min]
t_{an}: Rise time [s]

- Enter the maximum possible acceleration in MP1060.x.



Check the counting direction

- On the oscilloscope, set TRIGGER to FREE RUN.
- Start recording.
- Change to MANUAL operating mode.
- Press the axis direction buttons.
- Check the counting direction on the display and if necessary, correct it with MP210.x.

Position controller

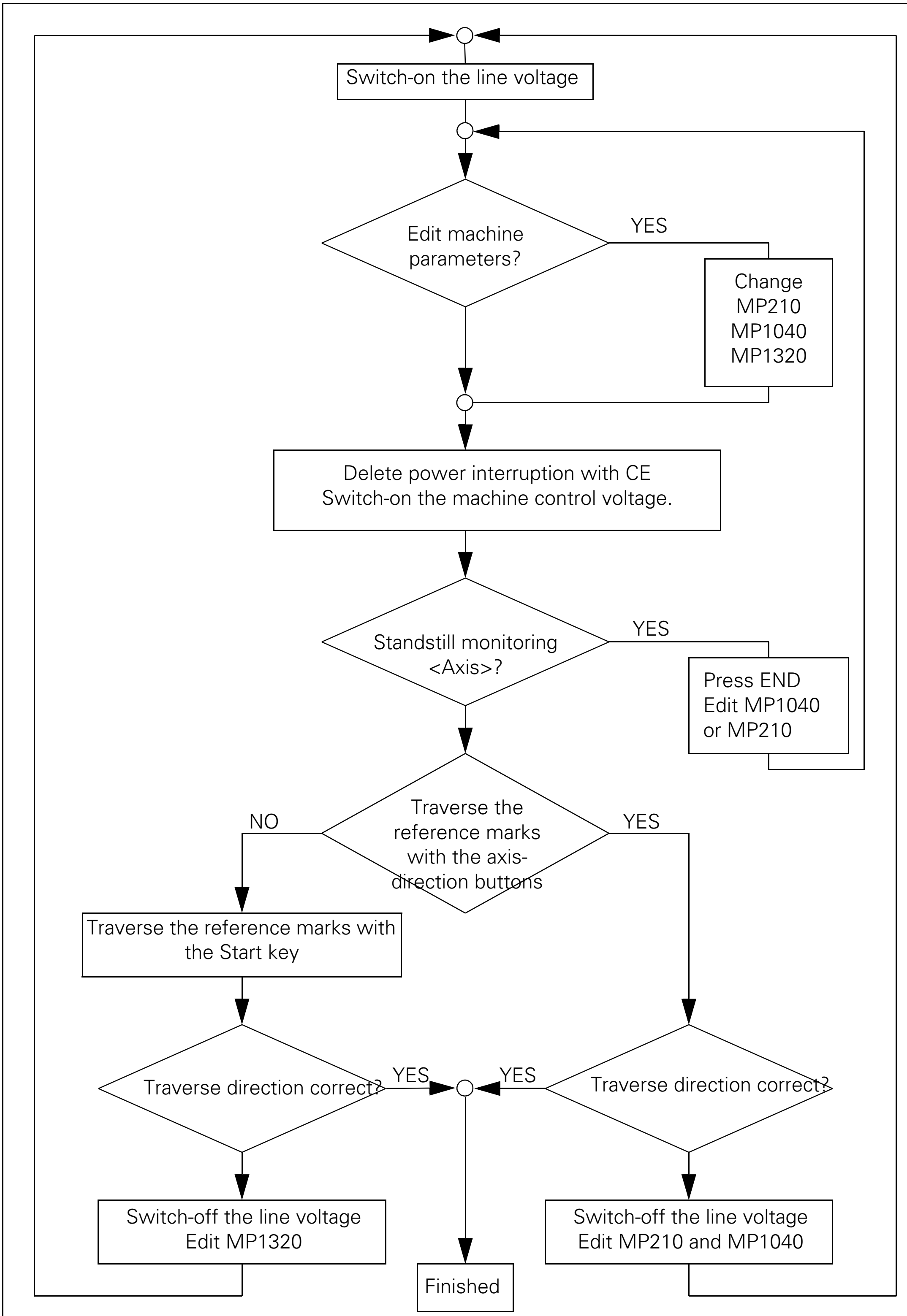
Adjusting the position control loop:

- Activate a PLC program that is adapted to the machine.
- Ensure that the position control loop is closed (W1038/W1040) and all inputs/outputs are properly operated.
- Optimize the position control loop in the following 12 steps:

If the position controller still oscillates after optimization, check the I factor (may be too high).

Step 1: Check the traversing direction (see flowchart)

- In MP1340.x, enter the sequence in which the reference points are to be traversed.


Switch-on the line voltage
Edit machine parameters?
YES
Change MP210 MP1040 MP1320
Delete power interruption with CE
Switch-on the machine control voltage.
Standstill monitoring <Axis>?
YES
Press END
Edit MP1040
or MP210
Traverse the reference marks with the axis-direction buttons
NO
YES
Traverse the reference marks with the Start key
Traverse direction correct?
YES
YES
Traverse direction correct?
Switch-off the line voltage
Edit MP1320
Finished
Switch-off the line voltage
Edit MP210 and MP1040

Step 2: Set the traverse range

You can enter up to three traverse ranges.
"Traverse Ranges" on page 661. Define the software limit switches as follows:

- In the MANUAL operating mode, press the MOD key to select the REF display. The position displays show the distance to the machine datum (MP960.x).
- With the axis direction buttons or the handwheel, move all axes in positive and negative direction until they almost reach the EMERGENCY STOP limit switches. Write down the displayed positions with algebraic sign.
- Enter the noted values in MP91x.x and MP92x.x.
- Press the MOD key and select the ACTL display.

Step 3: In MP1391 or MP1392, select the type of control

For control with velocity feedforward:

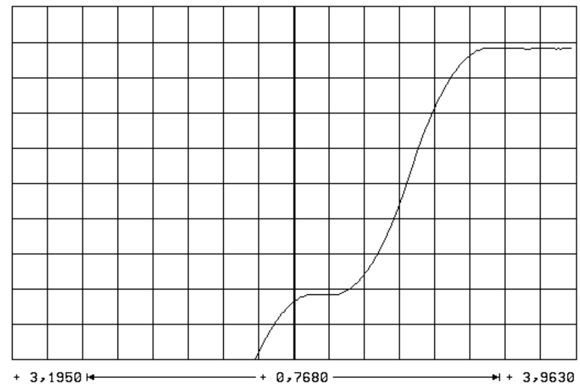
- Enter the temporary input values.

<table>
<tr><th>Machine parameters</th><th>Temporary input value</th></tr>
<tr><td>MP1391 or MP1392</td><td>Set to velocity feedforward control</td></tr>
<tr><td>MP1090.0</td><td rowspan="2">Enter a very small jerk, e.g. 1 (dependent on the machine)</td></tr>
<tr><td>MP1090.1</td></tr>
<tr><td>MP1092</td><td><greater than rapid traverse></td></tr>
<tr><td>MP1095</td><td>0</td></tr>
<tr><td>MP1096.x</td><td>0</td></tr>
<tr><td>MP1099.0</td><td>5</td></tr>
<tr><td>MP1099.1</td><td>3</td></tr>
<tr><td>MP1396.x</td><td>1</td></tr>
<tr><td>MP1521.x</td><td>0</td></tr>
</table>

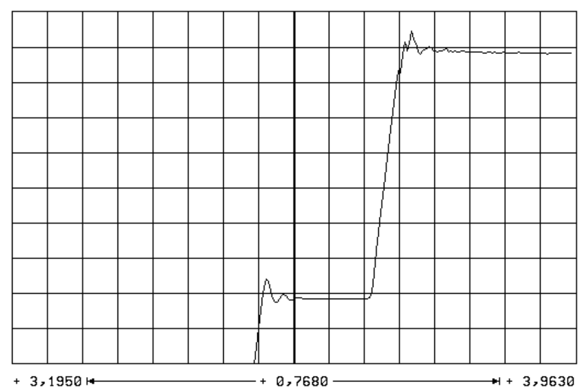
- Enter the following test program:

```
LBL 1
L X <maximum traverse>
R0 FMAX
LX0 FMAX
CALL LBL1 REP 100/100
```

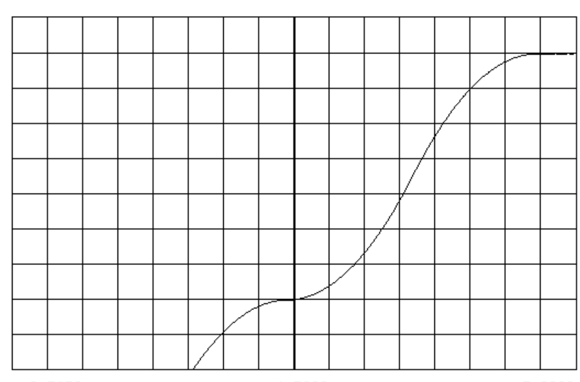
- Display the actual speed (act. speed) with the integrated oscilloscope and, if necessary, also show the following error (lag).
- Start the test program with feed-rate override = 100%.
- In MP1090.0 increase the jerk until the overshoot just disappears.



MP1090.x at optimum setting

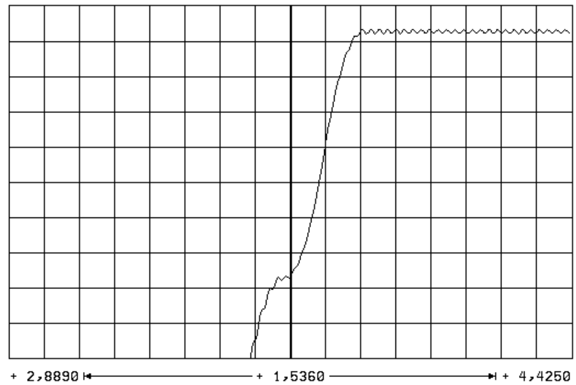


MP1090.x too large

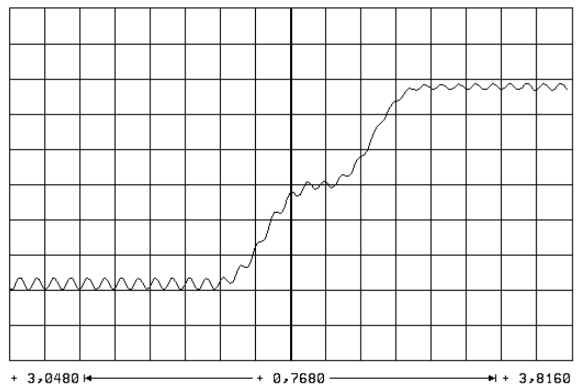


MP1090.x too small

- Enter the jerk determined from MP1090.0 in the axis-specific parameters MP1241.x and MP1242.x (previously MP1097.x and MP1098.x), and also in MP1086.x, MP1087.x and MP1089.x if required.
- Increase the k_v factor until the oscillation limit is reached.
- Calculate MP1510:
  MP1510.x = <determined value> · 0.6



Oscillation limit has been reached



kv factor too large

Unlike in operation with following error, you can also enter the optimum k_v factor for interpolated axes. You can save a number of different k_v factors in the iTNC and activate them with M functions (see “The Control Loop" on page 813). MP1090.x applies to all axes. The worst axis determines the input value.

Procedure:

- Load the axis-specific values in MP1241.x and MP1242.x (or MP1097.x and MP1098.x if using MPMODE=340422 in OEM.SYS).
- Reduce the adjusted jerk (MP1090.0) depending on the mechanical design of the machine. Do not set the jerk lower than necessary, however, because this strongly reduces the dynamic performance.
  - If at optimized jerk the maximum acceleration is not reached during the acceleration phase, enter the maximum machining feed rate in MP1092. In this case, define a higher jerk for high feed rates (> MP1092) to increase acceleration at these feed rates.

To select the nominal position value filter:

- Run a test program of short line segments.
- Use the oscilloscope to record the following error for each axis.
- Determine for each axis the oscillations on the following error. If you cannot find any oscillations, increase the jerk for the test in order to excite oscillation in the axes. Remember after the test to reset the jerk for each axis to its original value.
- Choose the appropriate type of filter characteristics from the description of the nominal position value filters on page 823 (or starting from page 835 the input values for the nominal position value filters up to software version 340 422-12). Consider the lowest determined frequency and the desired damping at this frequency.
- With MP1200 and MP1201 (previously MP1094 and MP1095) you select the single, double, HSC or advanced HSC filter.
- Test the three filter settings using a test part made of short line segments.
  - Single filter
  - Double filter
  - HSC filters
  - Advanced HSC filters

Note

If you have selected the best nominal position value filter for your application, please note that your input value from MP1096.0 can be overwritten by the machine user through Cycle 32.
If you have switched off the nominal position value filter (MP1096.x = 0), the machine user can also switch it on using Cycle 32.

Control with following error (servo lag):

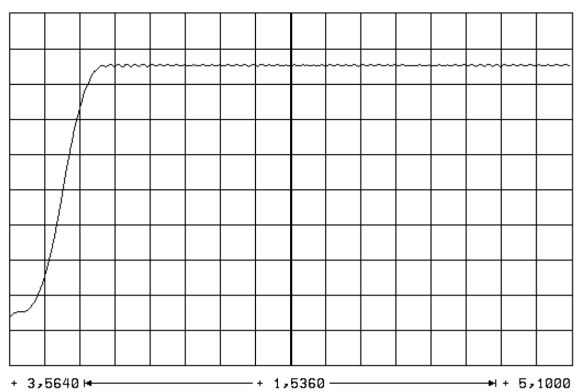
The adjusted maximum jerk works during operation with following error. MP1090 is not changed.

Procedure:

- Check the temporary input values for the machines parameters
- Specify the k_v factor for the machining feed rate:
- Enter the following test program:

```
LBL1
L X<maximum traverse> R0 F<machining feed rate>
L X0 R0 F<maximum machining feed rate>
CALL LBL1 REP 100/100
```

- Display the actual feed rate (v actual) with the internal oscilloscope.
- Start the test program with feed-rate override = 100%.
- Increase the value in MP1810.x up to the oscillation limit.
- Calculate MP1810.x:
  MP1810.x = <determined value> · 0.6



MP1810.x at the oscillation limit

For axes that are interpolated with each other, the k_v factors must be equal. The axis with the smallest k_v factor defines the input value for all axes.

You can save a number of different k_v factors in the iTNC (MP1815.x) and activate them with M functions (see “The Control Loop" on page 813).

Procedure for defining a characteristic curve kink point:

- k_v factor for rapid traverse (characteristic curve kink point):

$$\text{MP1830.x} = \frac{\text{Max. contouring feed rate} \cdot 100\ \%}{\text{Rapid traverse}}$$

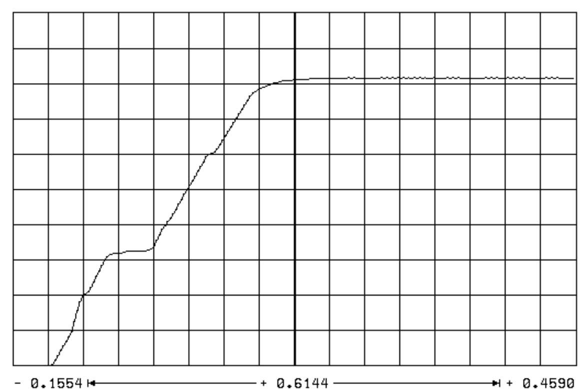
MP1820.x = 1

Set to operation with following error.

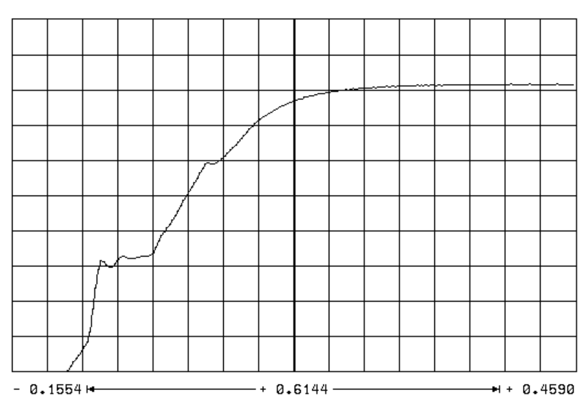
- Enter the following test program:

```
LBL2
L X <maximum traverse> R0 FMAX
L X0 R0 FMAX
CALL LBL1 REP 100/100
```

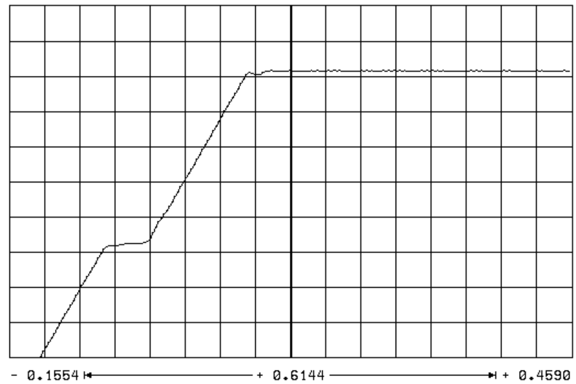
- Start the test program.
- Display the actual feed rate (v actual) with the internal oscilloscope:
  - If no oscillations are recognizable, no kink point is required.
  - If oscillations are visible, you must reduce MP1820.x until the oscillations have disappeared.



MP1820.x at optimum setting



MP1820.x too small



MP1820.x too large

Step 4: Switch-on the nominal position value filter

- In MP1096.x, enter a defined tolerance (e.g. 0.02 mm).

5. Activate monitoring functions:

> Note
>
> To ensure that the monitoring functions become effective at the right moment, you must enter meaningful values.

HEIDENHAIN recommends the following input values. You must change these values slightly to adapt them to the design of the machine.

MP	Temporary input value	Meaning
MP1030.x	0.01 mm	Positioning window
MP1110.x	2 · MP1030.x	Standstill monitoring
MP1140.x	0.03 [1000 min^{-1}]	Movement monitoring
MP1144.x	0.5 mm	Motion monitor for position and speed
MP1410.x	0.5 mm	Position monitoring in operation with velocity feedforward control (erasable)
MP1420.x	2 mm	Position monitoring in operation with velocity feedforward control (EMERGENCY STOP)
MP1710.x	1.2 · following error in rapid traverse	Position monitoring in operation with following error (erasable)
MP1720.x	1.4 · following error in rapid traverse	Position monitoring in operation with following error (EMERGENCY STOP)

Step 6: Compensate the backlash

If the cause of the backlash is outside of the control loop:

- Enter the backlash in MP710.x.

If the cause of the backlash is inside of the control loop:

- Enter the following test program:

```
LBL 1
L X100 R0 F10
L X0
CALL LBL 1 REP 100/100
```

- Use the internal oscilloscope to record V ACTUAL and V (ACT RPM). At the reversal point the actual feed rate follows the actual shaft speed by the time delay *t*.
- Set the machine parameters:
  - MP750 = $t \cdot \Delta$V ACTUAL (keep in mind the units for *t* and ΔV ACTUAL)
  - MP752 = approx. 20 ms (determined in test)

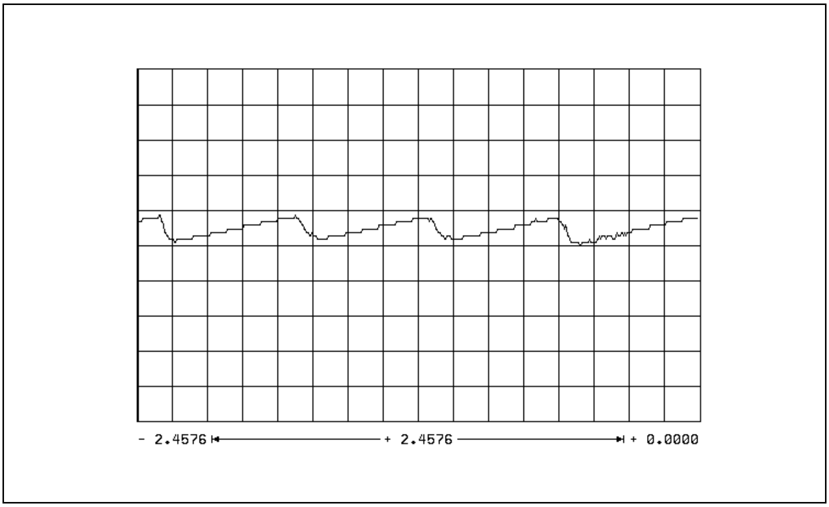
$$\Delta V\ ACTUAL = \left| V\ ACTUAL - V\ (ACT\ RPM) \right|$$

Step 7: Compensate the static (stick-slip) friction

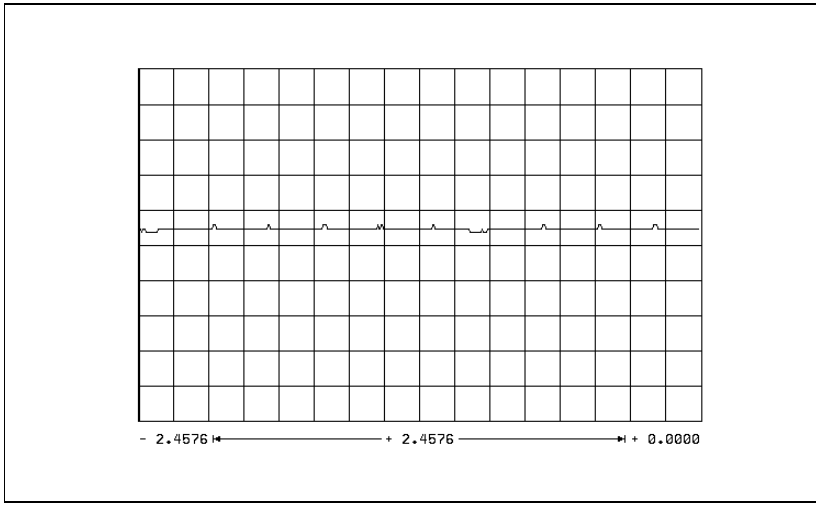
- Enter the backlash, if any exists.
- Enter the following test program (static friction in the Y axis):

```
LBL 1
L X+400 IY+0.5 R0 F200
L X0 IY+0.5 R0
CALL LBL1 REP 100/20
```

- Set the machine parameters:
  - MP1511.x = 0
  - MP1512.x = 20
  - MP1513.x = 0
- With the integrated oscilloscope, display the following error in the Y axis (Y SDIFF).
- Start the program and adjust the feed-rate override so that the following error caused by static friction becomes visible.



- Increase the feed rate until the following error is no longer measurable.
- From the current contouring feed rate, calculate the feed rate specific to the Y axis and enter the value in MP1513.1.
- Adjust the feed rate until the following error is measurable again.
- Increase MP1511.x in increments of 10 000 until the following error is no longer measurable.



If the machine oscillates at a standstill:

- Decrease MP1512.x.

Step 8: Limit the integral factor of the shaft speed controller
Very high static friction can cause an axis to jerk loose and "jump" around the target position.

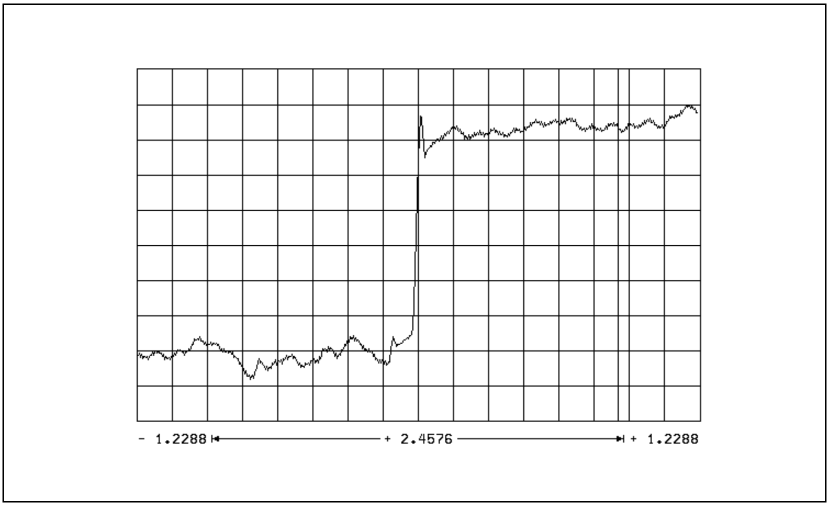
- Increase MP2512.x until the axis remains stationary.

Step 9: Adjust the holding moment

- Enter the following test program (static friction in axis Z):

```
LBL 1
L Z+2 R0 F50
L Z-2 R0 F50
CALL LBL 1/10
```

- Use the integrated oscilloscope to record the actual shaft speed (ACT RPM) and the nominal current value (I NOMINAL).
- Start the program.
- With the feed rate override knob, adjust the motor speed to ±10 min^{-1} (MP1054.x).
- Determine the current (I NOMINAL) in both directions of rotation.



▶ Calculate MP2630.x:

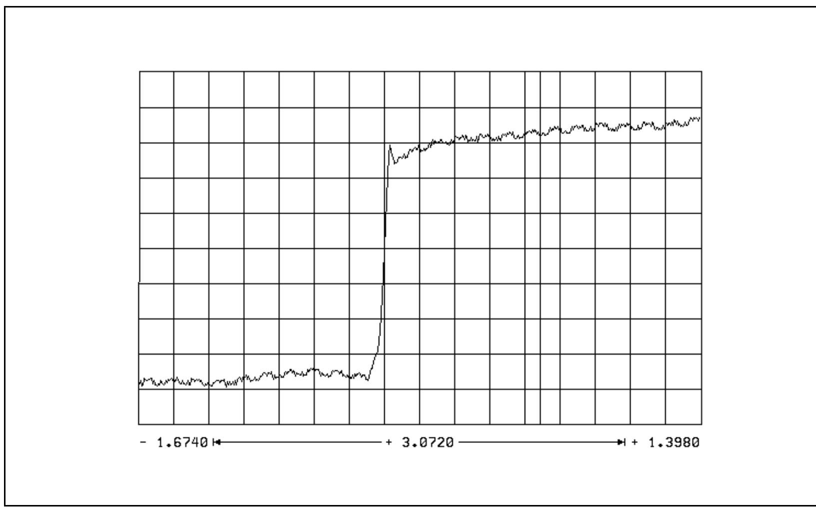
$$MP2630.x = \frac{I\ NOMINAL_1 + I\ NOMINAL_2}{2}$$

Step 10: Compensate the sliding friction

- Select operation with velocity feedforward control.
- Enter the following test program (sliding friction in the X axis):

```
LBL 1
L X+2 R0 F50
L X-2 R0 F50
CALL LBL 1/10
```

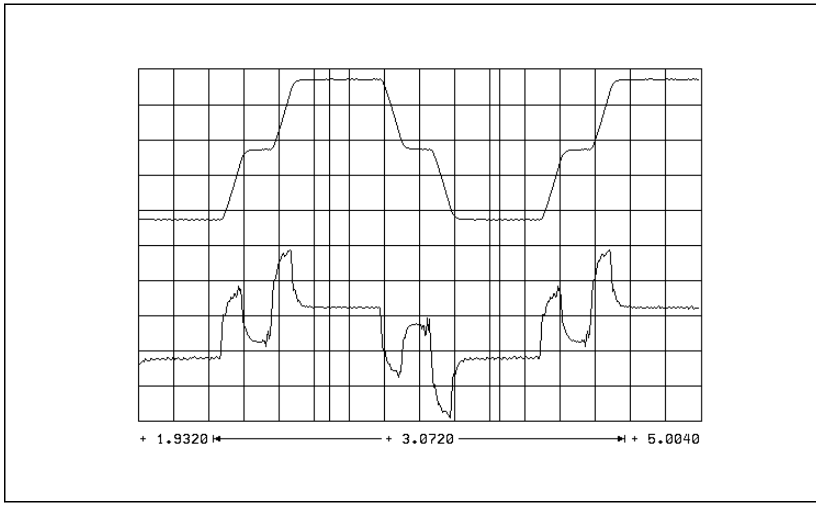
- Use the integrated oscilloscope to record the actual shaft speed (ACT RPM) and the nominal current value (I NOMINAL).
- Start the test program.
- With the feed rate override knob, adjust the motor speed to 10 min^{-1} (MP1054.x).
- Determine the current (I NOMINAL) in both directions of rotation.



- Calculate MP2610.x:

$$\text{MP2610.x} = \frac{\text{I NOMINAL}_1 + \text{I NOMINAL}_2}{2}$$

- Change the test program so that the motor rotates at its rated speed.
- Restart the test program.
- Determine the current (I NOMINAL) for the rated shaft speed.



▶ Calculate MP2620.x:

$$MP2620.x = \frac{I\ NOML_1 - I\ NOML_2}{2}$$

In the event that the motor cannot be driven at the rated speed:

▶ Measure I NOMINAL at maximum speed (rapid traverse) and calculate the current at rated speed as follows:

$$MP2620.x = \frac{(I_{max} - MP2610.x) \cdot \text{<rated speed>}}{n_{max}} + MP2610.x$$

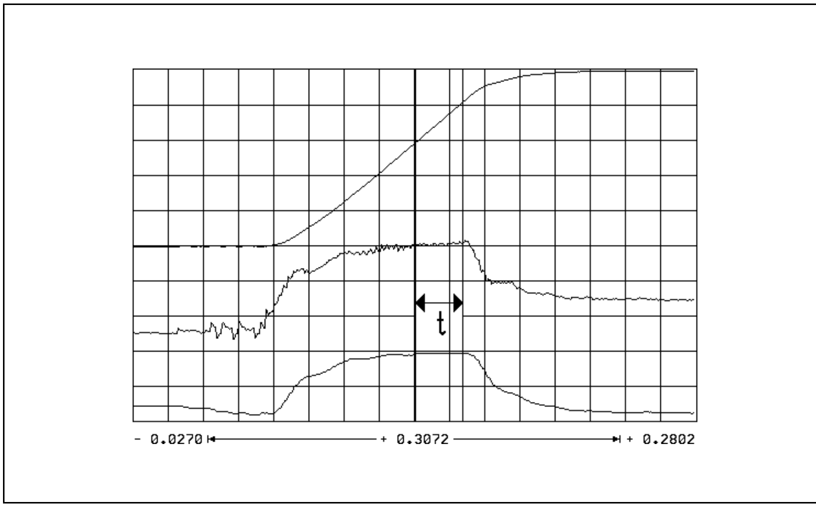
I_{nmax}: Current at rapid traverse

n_{max}: Shaft speed at rapid traverse

Step 11: Check the acceleration feedforward

- Select operation with velocity feedforward control.
- Enter the following test program:

```
LBL 1
L X+100 R0 F5000
L X-100 R0 F5000
CALL LBL 1/10
```



- Use the integrated oscilloscope to record the actual shaft speed (ACT RPM), the nominal current value (I NOMINAL), and the integral-action component of the nominal current value I (INT RPM).
- Start the test program.
- Adjust the speed with the feed rate override knob so that I NOMINAL is not limited.
- Measure the gradient of the acceleration ramp in the part in which I(INT RPM) remains constant.
- Calculate MP2600.x:

$$\text{MP2600.x} = \frac{\text{I (INT RPM) [A]} \cdot \text{t [s]} \cdot \text{60 [s/min]} \cdot \text{MP1054.x [mm]}}{\Delta\text{V (ACT RPM) [mm/min]}}$$

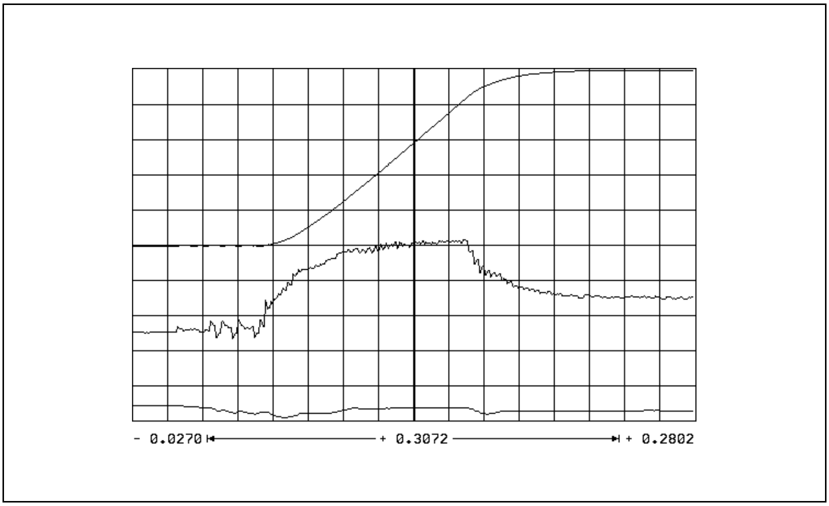
I (N INT): integral-action component of the nominal current value

t: Acceleration time in which I (INT RPM) remains constant

ΔV (ACT RPM): Change of actual rpm during *t*

MP1054.x: Traverse distance per motor revolution

- Repeat this measurement to check the input value of MP2600.x. I (INT RPM) must have approached zero.



Step 12: Run the circular test

With the circular test you can check the exact input values for compensating sliding friction:

- Determine the radial acceleration:
  MP1070 = 0.7 · MP1060.x
  MP1060.x represents the smallest acceleration in the working plane.
- At mid-range feed rate (approx. 500 mm/min) check the parameter MP2610. At the optimum setting the reversal peaks are at a minimum.

At feed rates greater than approx. 6000 min^{-1} the reversal peaks might point inward as a result of overcompensation:

- In this case, increase MP2612.x until the reversal peaks no longer point inward.

6.14.7 Commissioning the digital spindle

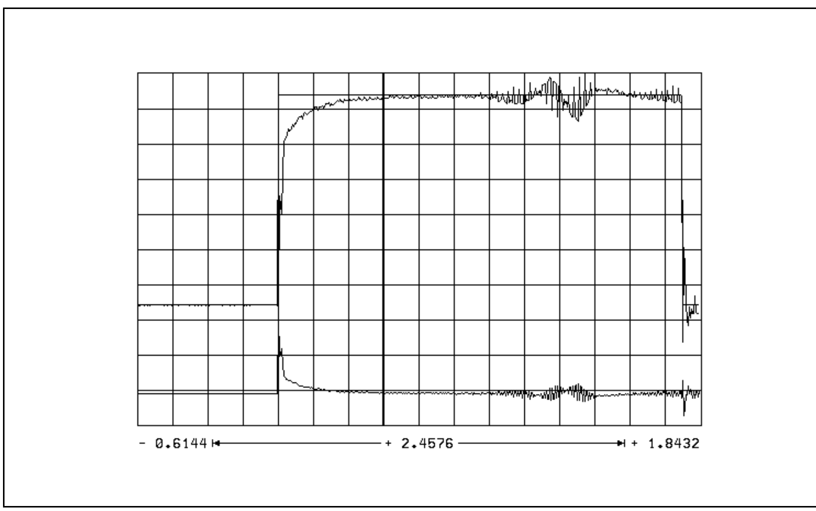
Current controller

Same procedure as for digital axes.

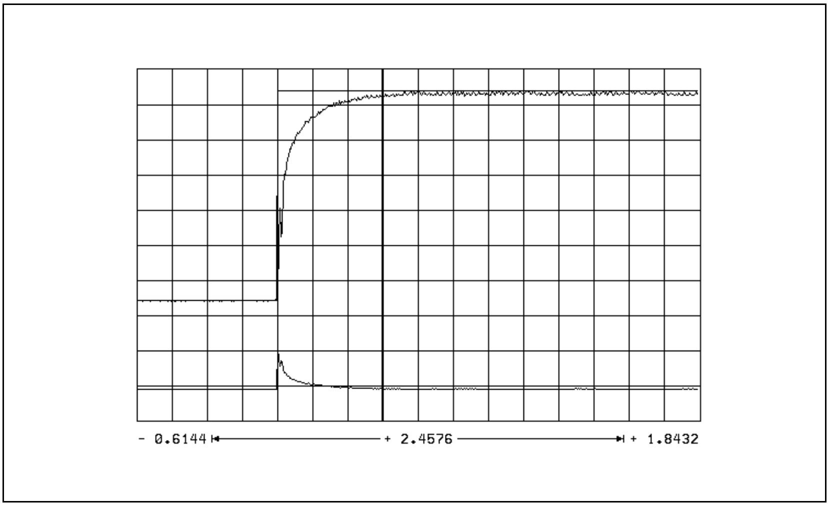
Speed controller

Define the step function:

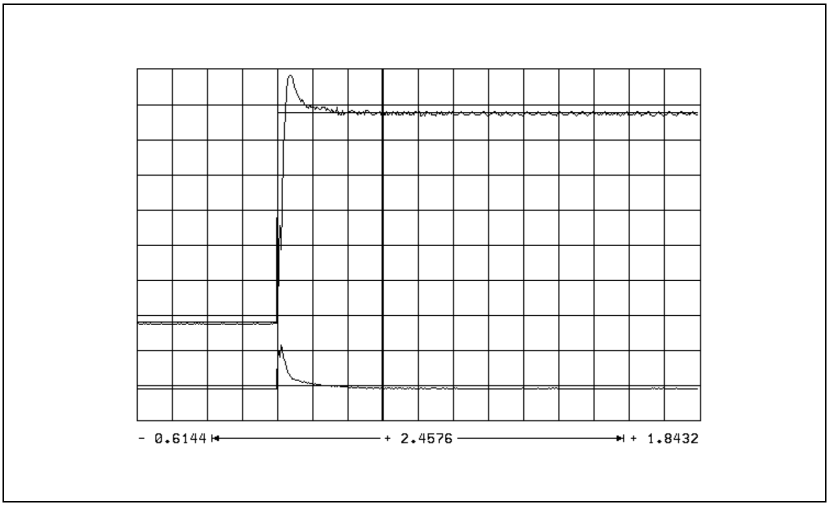
- In MP3411.x, enter the maximum acceleration and start the step by switching the spindle on.
- Activate a spindle speed from the highest gear range.
- With the integrated oscilloscope, record the nominal velocity value V (NOM RPM), the actual speed value V (ACT RPM), and the nominal current value (I NOMINAL).
- Output a step by activating the spindle on function (M03/M04).
- Choose the height of the step function for a very low speed so as not to overload the speed controller, i.e. so that I NOMINAL is not limited.
- Increase the P factor (MP2500.x) until the system oscillates or no change is visible. To edit machine parameters, press the MP EDIT soft key in the Setup menu.



- Calculate MP2500.x: MP2500.x = MP2500.x · 0.6

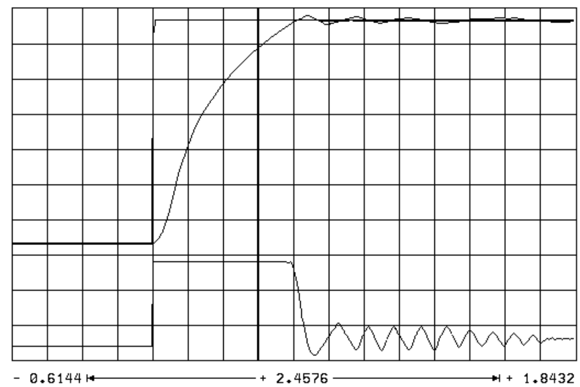


▶ Increase the I factor (MP2510.x) until you see one overshoot followed by a slight undershoot.

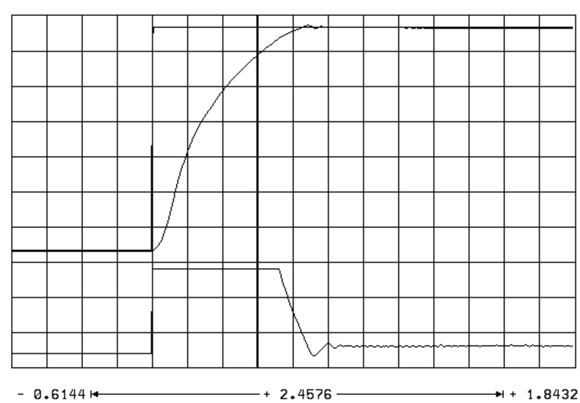


▶ Output the step with maximum shaft speed. I NOMINAL is within the limitation during acceleration. I NOMINAL must not oscillate after reaching the maximum speed. If I NOMINAL oscillates:

- Reduce MP2500.x and MP2510.x evenly until the overshoots are minimized.



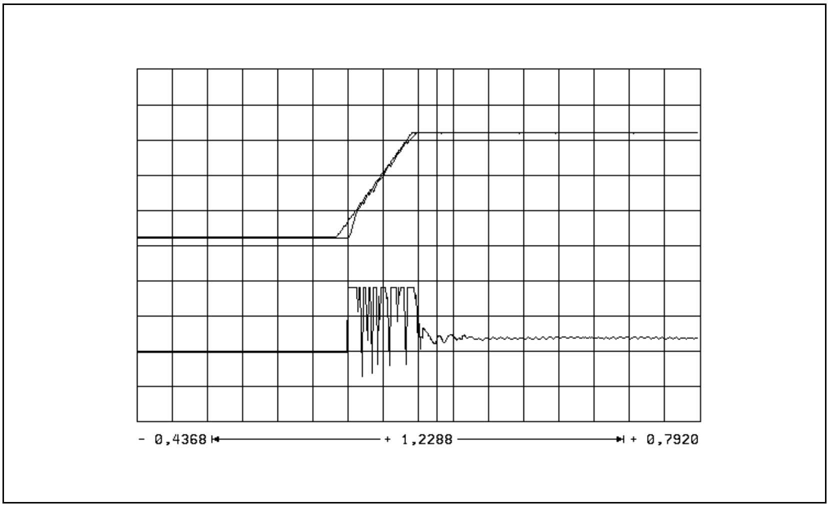
I NOMINAL oscillates



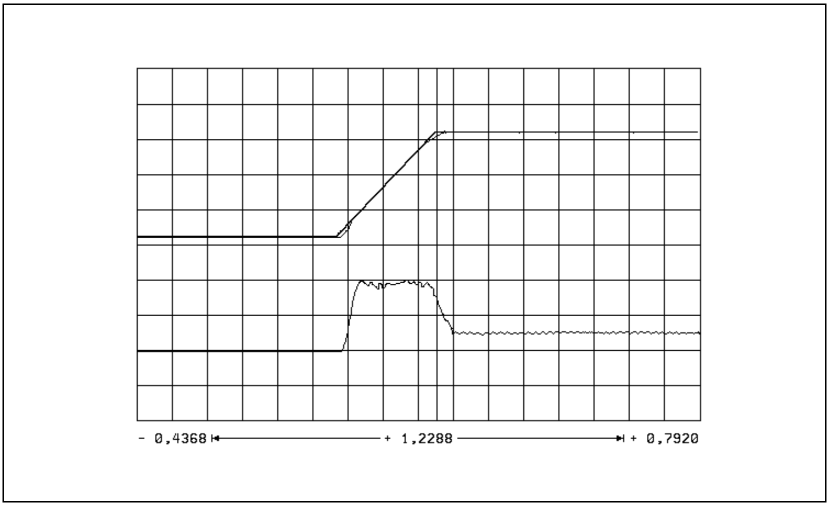
Only one overshoot

Optimize the acceleration:

- Optimize the acceleration individually for each gear range.
- Choose a ramp gradient at which the motor almost reaches the electrical current limit, and set it with MP3411.x.



- In MP3412.0, enter a factor for MP3411.x that becomes effective in the braking ramp with M05. It is the electrical current limit that is braked.



In the TAPPING and SPINDLE ORIENTATION modes, I NOMINAL must not reach the limit for acceleration:

- In MP3412.x, enter a factor for MP3411.x for these operating modes.
- With MP3415.x, specify an individual overshoot behavior for every spindle operating mode. Adapt the nominal value trace to the actual trace.

Check the direction of rotation.
You can check the direction of rotation of the spindle when M03 is output. If the spindle does not rotate in clockwise direction:

- Modify MP3130.

Position controller

The position control loop of the spindle is closed only during the spindle orientation:

- Close the position control loop of the spindle. “Oriented Spindle Stop" on page 964.
  - If the error message "Nominal speed value S too high" appears, you must modify MP3140.
- Optimize the k_v factor (MP3440.x for each gear range.
  A TOOL CALL must be run to transfer the modified gear-specific MPs.

6.14.8 Commissioning an analog axis

Temporary input values

▶ Enter the following temporary input values when you begin

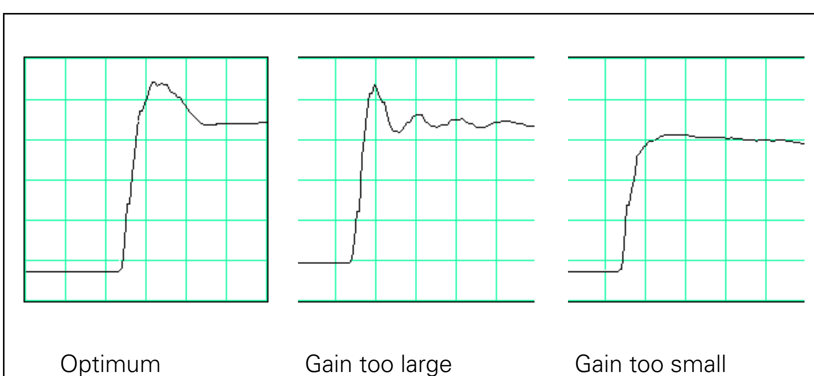
MP	Temporary input value	Meaning
MP1030.x	0.01	Positioning window
MP1090.x	1	Maximum permissible jerk on the tool path
MP1092	<Maximum rapid traverse>	Feed rate threshold from which MP1090.1 becomes effective
MP1110.x	2.0	Standstill monitoring
MP1140.x	10	Movement monitoring
MP1410.x	0.5	Position monitoring in operation with velocity feedforward control (erasable)
MP1420.x	2	Position monitoring in operation with velocity feedforward control (EMERGENCY STOP)
MP1510.x	1	k_V factor for velocity feedforward control
MP1710.x	50	Position monitoring in operation with following error (erasable)
MP1720.x	50	Position monitoring in operation with following error (EMERGENCY STOP)
MP1810.x	1	k_V factor for control with following error
MP1820.x	1	Multiplier for the k_V factor
MP1830.x	100	Characteristic curve kink point

Adjusting the servo amplifier

> **Note**
>
> For analog axes, you must adjust the servo amplifier before you optimize the position controller.

Procedure:

- Disconnect the nominal-value connection between the servo amplifier and the MC 42x(B).
- Short-circuit the nominal value input on the servo amplifier.
  The input must have a 0 V voltage.
- Activate control enabling at the servo amplifier.
- Connect the power supply to the servo amplifier.
- Perform a coarse offset adjustment:
  - If the axis moves in spite of the short-circuited nominal value input, you must adjust the offset potentiometer until the axis stops moving.
- Remove the jumper at the nominal value input and establish a nominal-value connection to the MC 42x(B).
- Perform a coarse velocity adjustment:
  - Set MP1010.x (rapid traverse) and MP1050.x (analog voltage at rapid traverse).
  - With the internal oscilloscope functions, output the nominal value step at the height for rapid traverse.
  - Record UANALOG and check the voltage.
  - Use a tachometer to measure the rotational speed of the motor and a tacho-potentiometer at the servo amplifier to adjust the nominal speed for rapid traverse.
  - Connect an oscilloscope to the tachometer of the motor.
  - Measure the step response on the tachometer during the step output.
- Adjust the proportional (P) component and the integral-action (I) component of the speed controller at the servo amplifier.



Determining the acceleration

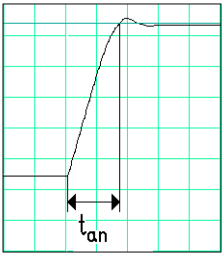
After adjusting the servo amplifier, you can determine from the step response the maximum possible acceleration:

$$a = \frac{F_{max}}{t_{an} \cdot 60\,000}$$

a: Acceleration [m/s²]

F_{max}: Maximum machining feed rate (MP1010.x) [mm/min]

t_{an}: Rise time [s]



▶ Enter the maximum possible acceleration in MP1060.x.

Position controller

Please note:

> **Note**
>
> For analog axes, you must adjust the servo amplifier before you optimize the position controller.

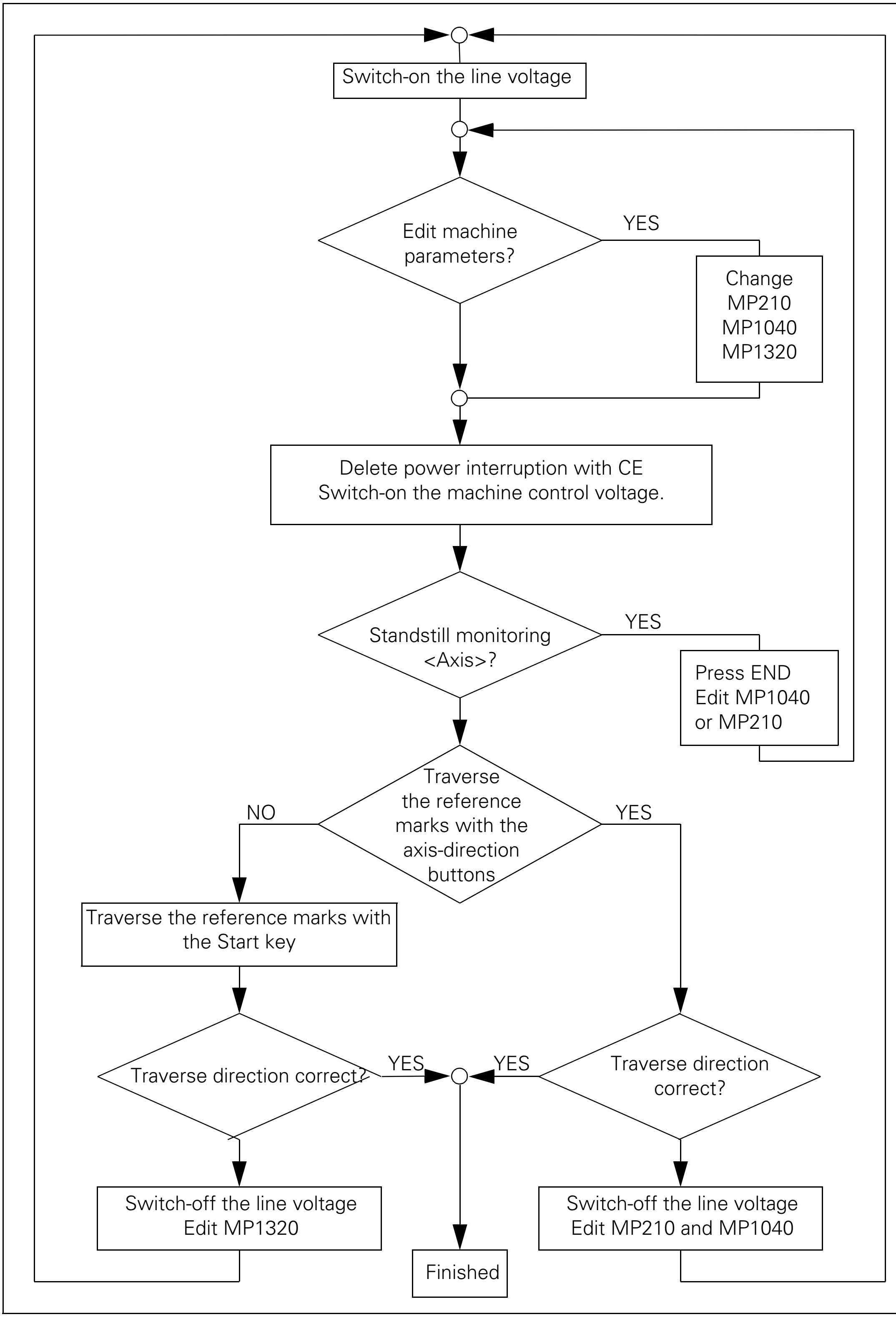
Adjusting the position control loop:

▶ Activate a PLC program that is adapted to the machine.

▶ Ensure that the position control loop is closed (W1038/W1040) and all inputs/outputs are properly operated.

▶ To optimize the position control loop take the following steps:

Step 1: Check the direction of traverse
(see flowchart)


Switch-on the line voltage
Edit machine parameters?
YES
Change MP210 MP1040 MP1320
Delete power interruption with CE
Switch-on the machine control voltage.
Standstill monitoring <Axis>?
YES
Press END
Edit MP1040 or MP210
Traverse the reference marks with the axis-direction buttons
NO
YES
Traverse the reference marks with the Start key
Traverse direction correct?
YES
YES
Traverse direction correct?
Switch-off the line voltage
Edit MP1320
Finished
Switch-off the line voltage
Edit MP210 and MP1040

Step 2: Set the traverse range
Same procedure as for digital axes.

Step 3: Specify the type of control
For control with following error, same procedure as for digital axes.

For control with velocity feedforward control, same procedure as for digital axes.

Step 4: Perform an offset adjustment
At the iTNC: "The Control Loop" on page 813.

5. Activate monitoring functions:

- Enter the following temporary input values when you begin: see "Commissioning of digital axes" on page 1026

Step 6: Compensate the backlash
Same procedure as for digital axes.

Step 7: Compensate the static (stick-slip) friction
Same procedure as for digital axes.

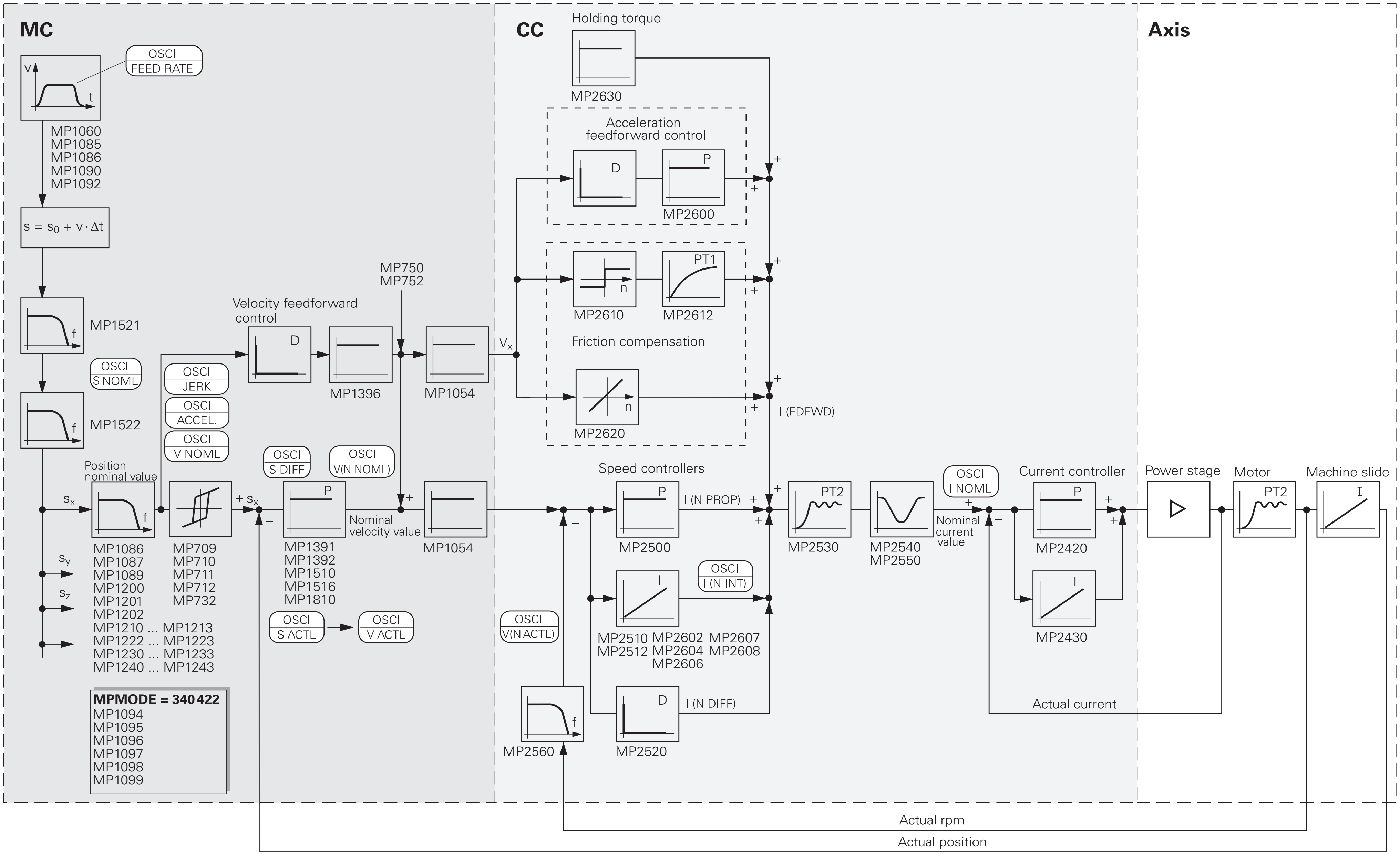
6.14.9 Commissioning the analog spindle

Adjusting the servo amplifier
Same procedure as for analog axes.

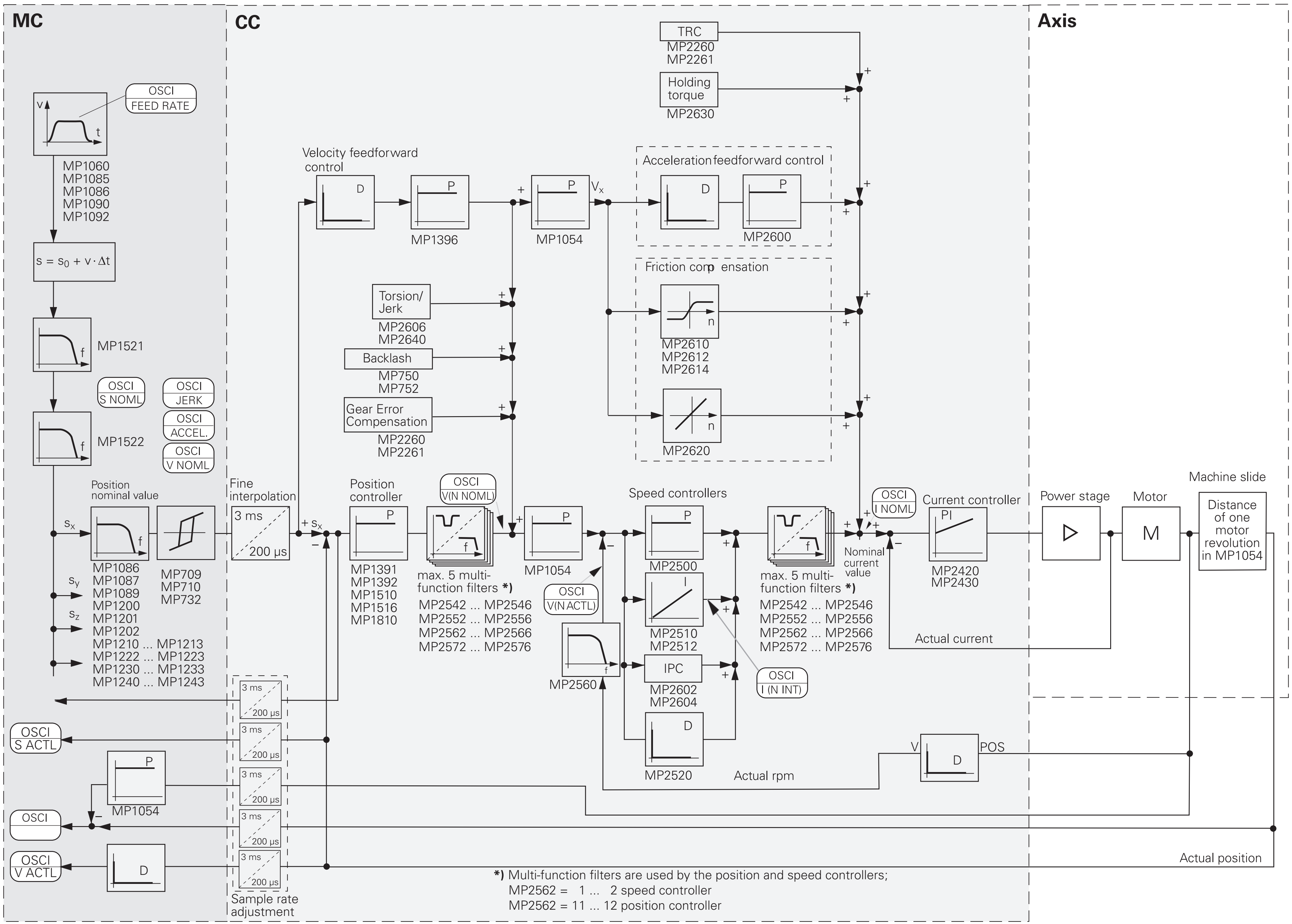
Acceleration
Same procedure as for digital spindle. You measure the signals directly at the servo amplifier with an external oscilloscope.

Direction of rotation
Same procedure as for digital spindle.

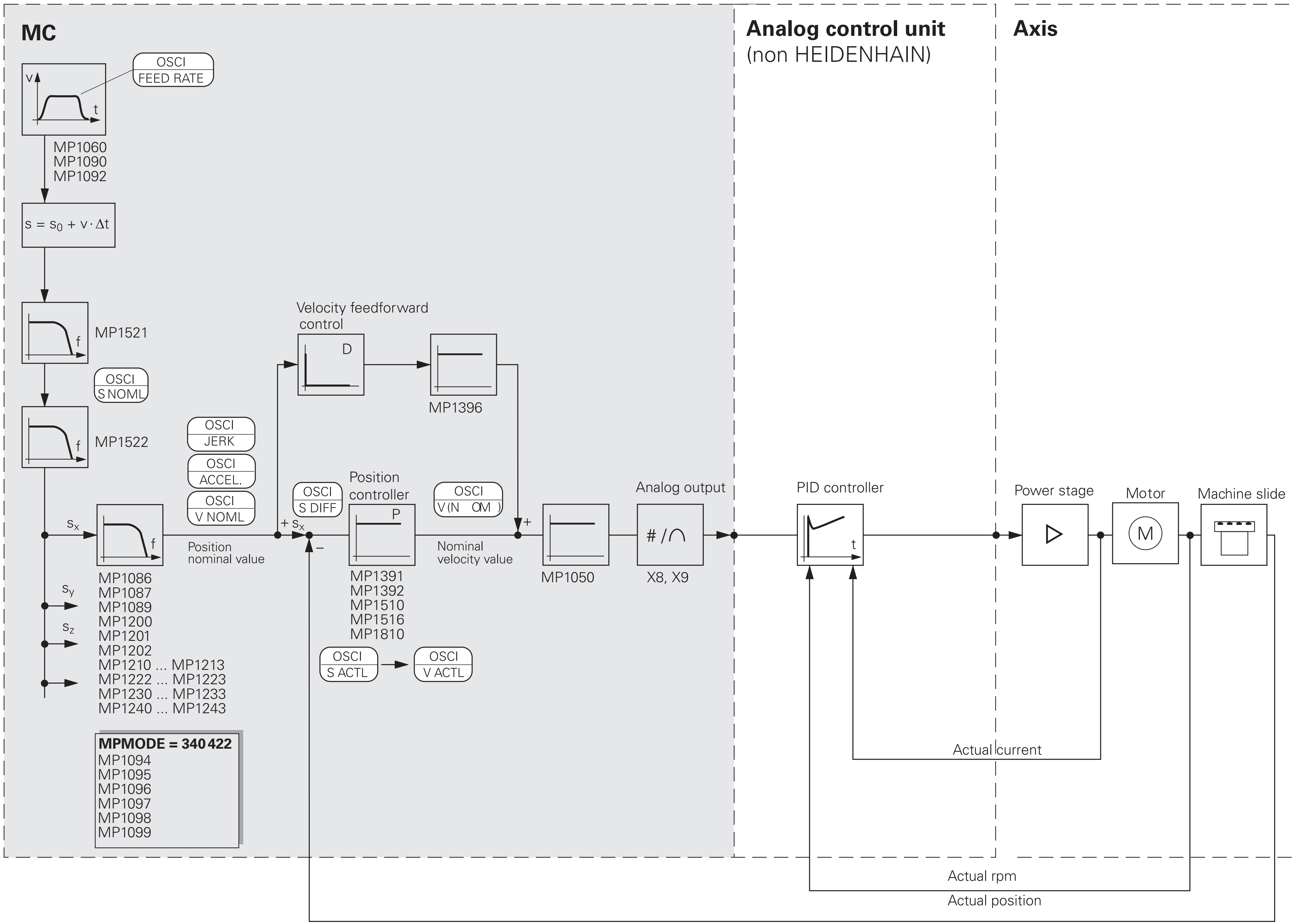
Position controller
Same procedure as for digital spindles.


MC
CC
Axis
OSCI FEED RATE
MP1060
MP1085
MP1086
MP1090
MP1092
s = s0 + v·Δt
MP1521
MP1522
OSCI S NOML
OSCI JERK
OSCI ACCEL.
OSCI V NOML
Position nominal value
MP1086
MP1087
MP1089
MP1200
MP1201
MP1202
MP1210 ... MP1213
MP1222 ... MP1223
MP1230 ... MP1233
MP1240 ... MP1243
MP709
MP710
MP711
MP712
MP732
MPMODE = 340422
MP1094
MP1095
MP1096
MP1097
MP1098
MP1099
Velocity feedforward control
MP1396
MP750
MP752
MP1054
OSCI S DIFF
OSCI V(N NOML)
Nominal velocity value
MP1391
MP1392
MP1510
MP1516
MP1810
OSCI S ACTL
OSCI V ACTL
Holding torque
MP2630
Acceleration feedforward control
MP2600
MP2610
MP2612
Friction compensation
MP2620
I (FDFWD)
Speed controllers
I (N PROP)
MP2500
OSCI I (N INT)
MP2510 MP2602 MP2607
MP2512 MP2604 MP2608
MP2606
I (N DIFF)
MP2520
OSCI V(N ACTL)
MP2560
MP2530
MP2540
MP2550
OSCI I NOML
Nominal current value
Current controller
MP2420
MP2430
Actual current
Power stage
Motor
Machine slide
Actual rpm
Actual position

6.16 Block Diagram of iTNC 530 (with CC 424)



6.17 Block Diagram of iTNC 530 (with Analog Control Unit)



7 CC 424(B) Controller Unit

7.1 Differences Between the CC 424(B) and CC 422

Function	CC 424(B)	CC 422
Hardware	Position encoder inputs X201 to X206 and X207 to X210 on the CC 424(B)	No position encoder inputs on the CC 422; they are on the MC 42x(B)
Assignment of speed encoder inputs to the PWM outputs	Permanent assignment, MP112.x removed	Variable assignment via MP112.x
MP115.x, MP116.x	When using an MC 42x(B) without position encoder inputs, MP115.x is omitted. Otherwise MP115.x applies to the position encoder inputs on the MC 42x(B); MP116.x applies to the position encoder inputs on the CC 424(B)	MP115.x applies to the position encoder inputs on the MC42x(B)
PWM frequency	Can be set via MP2180.x (switchable during operation); the calculation of the current-controller cycle time must be adapted via MP2182.x	Can be set via MP2180.x (not switchable during operation)
Control loops can be switched from single speed to double speed for higher controller performance	See "Single-Speed, Double-Speed, PWM Frequency" on page 1074.	–
Control-loop cycle times (at 5000-Hz PWM frequency) (position/speed/current)	Single-speed: 200 µs/200 µs/100 µs Double-speed: 200 µs/100 µs/100 µs (with position encoder) 100 µs/100 µs/100 µs (without position encoder)	1.8 ms/0.6 ms/100 µs (at 5000-Hz PWM frequency)
Following error in the jerk phase (MP2606.x)	Typical input values: 0.001 to 0.005	Typical input values: 0.5 to 1
Stick-slip friction compensation (MP2610.x, MP2612.x, MP2614.x)	Feed-rate independent; MP2610.x same meaning as previously (effective values, readjustment necessary), MP2612 has new meaning MP2614.x is new	Feed-rate dependent; MP2610.x and MP2612.x (peak values)
Static friction compensation (MP1511.x, MP1512.x)	MP1511.x and MP1512.x can only be used for analog axes. MP2610.x, MP2612.x and MP2614.x should be used for digital axes.	MP1511.x and MP1512.x can be used for analog and digital axes. MP2610.x, MP2612.x and MP2614.x should be used for digital axes.
Multiplication factor for k_V factor and kink point	MP1820.x and MP1830.x are only supported for analog axes.	MP1820.x and MP1830.x are supported for analog and digital axes.

Function	CC 424(B)	CC 422
Filter in the speed control loop	MP2530.x, MP2540.x, MP2550.x and MP2560.x omitted, New machine parameters MP2542.x to MP2546.x, MP2552.x to MP2556.x, MP2562.x to MP2566.x, MP2572.x to MP2576.x, MP2560.x has new meaning	Filter setting via MP2530.x, MP2540.x, MP2550.x and MP2560.x
Master-slave torque control	As of 340 422-06: The PWM outputs of the master and slave axes must always be operated on the same DSP ("Single speed" setting); the control factors MP29xx.y differ from the CC 422, i.e. a new adjustment is necessary when exchanging a CC 424(B) and CC 422	The master and slave axis must be on the same speed controller PCB
Reading the absolute value of encoders with EnDat interface	The absolute value can be read out again via the **PASS OVER REFERENCE** soft key or via Module 9220 (i.e. after the exchange of milling heads).	The absolute value is only read when the control is started.
MP2220.x	Bit 4: Monitoring for excessive temperature Bit 5: Monitoring for insufficient temperature Bit 6: Reserved Bit 7: Monitoring of encoder input frequency Bit 8: Adjust mechanical offset by gradually increasing the k_V factor	Bits 4, 5, 6, 7 and 8 are without function
MP2250.x, MP2252.x	Determining the field angle for unaligned encoders	–
MP7602	PLC cycle time [ms]	–
MP7600.0	MP7600.x removed, path interpolation fixed at 3 ms (does not influence the position controller cycle)	Factor for position controller cycle time; factor x 0.6 [ms]
Display in internal oscilloscope and in TNCopt	Effective values	Peak values

7.2 Connecting the Encoders

7.2.1 General information

HEIDENHAIN contouring controls are designed for use with incremental linear and angular encoders as measuring systems. The encoder signals are subdivided 1024-fold.
Encoders with one reference mark or distance-coded reference marks and with EnDat interface are permissible.

Please use only HEIDENHAIN encoder cables, connectors and couplings.

7.2.2 Position encoders

Incremental position encoders with 1 V_{PP} signals and absolute encoders with EnDat interface can be connected to the CC 424(B).

- Set the encoder signal with MP116.0.
- With MP116.2, you set the maximum input frequency.

Note

The incremental track data must be entered for the corresponding position encoder inputs for encoders with EnDat interfaces.

MP116.0 **Position encoder input 1 V_{PP} or 11 μA_{PP}**
Format: %xxxxxxxxxxxxxxx
Input: Bit 0 to bit 9: Linear encoder inputs X201 to X214
Bit 10: No function
0: 1 V_{PP}
1: 11 μA_{PP}

MP116.1 **Reserved**
Format: %xxxxxxxxxxxxxxx
Input: Enter %00000000000

MP116.2 **Input frequency of position encoder inputs**
Format: %xxxxxxxxxxxxxxx
Input: Bit 0 to bit 9: Position encoder inputs X201 to X210
Bit 10: No function
For 1 V_{PP}: 0: 27 kHz
1: 400 kHz
For 11 μA_{PP}:0: 27 kHz
1: 140 kHz

7.2.3 Speed encoders

Incremental position encoders with 1 V_{PP} signals and absolute encoders with EnDat interface can be connected to the CC 424(B).

7.3 Relationship between Speed Input and PWM Output

On the CC 424(B) there is a **fixed assignment of speed encoder input to PWM output.** MP112.x is omitted for this reason (special case: MP113.x). The selection of the encoder inputs and PWM outputs is only via MP120.x/ MP121.x.

Drive-control board	PWM output (MP120.x/MP121.x)	Speed encoder input
1	X51	X15
1	X52	X16
1	X53	X17
1	X54	X18
1	X55	X19
1	X56	X20
2 (1[a])	X57	X80
2 (1[a])	X58	X81
2	X59	X82
2	X60	X83
2	X61	X84
2	X62	X85
2	X63	X86
2	X64	X87

a. For CC 424(B) with 8 and with 14 control loops

As of software version 340 49x-03 you can use the CC 424(B) to operate open-loop axes (counter axes) or analog closed-loop axes (MP120 input value < 51) via the position encoder inputs.

Special case: MP113.x

MP113.x is used to switch the speed encoder inputs for the operation of a second spindle.
If only one spindle is used, MP113.x = 0, and the assignment from the above table is valid.

If, for example, two spindle motors are driven with one PWM output (and the same power module) and the power can be switched over through a contactor, the speed encoder input can be selected with MP113.x.

PWM output for the spindle	Speed encoder input	MP113.x input value
X51	X15 or X17	(0 or 15) or 17
X52	X16 or X18	(0 or 16) or 18
X53	X15 or X17	15 or (0 or 17)
X54	X16 or X18	16 or (0 or 18)

The error message **`C2A0 Encoder input <axis>`** appears if an invalid entry is given for MP113.x.

7.4 Relationship between PWM Output and Position Input

The position encoder inputs for digital control loops are no longer on the MC42x(B), but on the CC 424(B). There is an MC 42x(B) (ID 369 947-xx) without position encoder inputs.

The position encoder inputs can be assigned in any order on the drive control board. Use MP110.x/MP111.x for this.
The position encoder inputs X201 to X206 can be assigned as desired to the PWM outputs X51 to X56, and the position encoder inputs X207 to X210 to the PWM outputs X57 to X60.

Drive-control board	Position encoder input (MP110.x/MP111.x)
1	X201
1	X202
1	X203
1	X204
1	X205
1	X206
2 (1[a])	X207
2 (1[a])	X208
2	X209
2	X210
2	X211
2	X212
2	X213
2	X214

a. For CC 424(B) with 8 and with 14 control loops

7.5 Single-Speed, Double-Speed, PWM Frequency

7.5.1 General information

Depending on the version, the CC 424(B) controller unit has one or two controller PCBs, each of which has 4 DSPs (digital signal processors).

Each of these DSPs can serve two control loops. Depending on the machine or controller performance required, it can be necessary for a DSP to handle just one control loop.

That is why the CC 424(B) differentiates between single-speed control loops (SS) and double-speed control loops (DS). Double speed control loops operate with shorter controller cycle times.
With single-speed control loops, one DSP is used for two control loops. With double-speed control loops, one DSP is used for one control loop.
When setting up the modular inverter systems, pay attention to the sequence of the power modules (due to the sequence of the PWM outputs for single and double speed on the CC 424(B)).

> **Note**
>
> Due to the sequence of the PWM outputs for single and double speed on the CC 424(B), pay attention to the sequence of the power modules when setting up the modular inverter systems.

Single-speed control loops are used for:

- Spindles
- Conventional axes

Double-speed control loops are used for:

- Linear motors
- Torque motors
- High-frequency spindles
- "Axes that are difficult to control"

The configurable double-speed control loops of the CC 424(B) controller unit allow you to increase the controller performance. This enables you, for example, to achieve excellent results regarding the workpiece surface quality or speed optimization.

Double-speed control loops are needed particularly for linear drives in order to achieve proper speed/position control factors required by high-efficiency mechanical systems.

High-speed spindle motors requiring PWM frequencies greater than 5 kHz with short current controller cycle times can be operated only with double-speed control loops.

Please note that the maximum number of available control loops may be reduced through the configuration of double-speed control loops. See "Setting the controller performance in MP 7610.x" on page 1075.

7.5.2 Prerequisites

The CC 424(B) with 8 or 12 control loops is only supported as of software versions

- 340 49x-xx

and the CC 424(B) with 14 control loops only as of software version

- 340 49x-02

by TNCopt.

7.5.3 Machine parameters

Since the hardware program has been expanded to up to 14 control loops for the CC 424(B), please keep the following settings of MP7610.x in mind:

- MP7610.x was expanded (only CC 424(B)):
  MP7610.x defines control loops as single-speed or double-speed with the bit codes:
  MP7610.0: first drive control board
  MP7610.1: second drive control board
  In order to remain compatible to older, assigned settings, the two machine parameters have the following preassigned settings:
  MP7610.0: %1100
  MP7610.1: %1111

7.5.4 Setting the controller performance in MP 7610.x

We recommend that you proceed as follows for defining the controller performance of the CC 424(B) by assigning the bits in MP 7610.x to the PWM outputs:

- ▶ Determine the required performance of a control loop. Indications such as the desired machining speed, accelerations and surface quality of the workpiece are to be considered. This results in certain motors and control loop properties being required, which can be configured on the CC 424(B) by using the SS (single-speed) or DS (double-speed) functionality.
- ▶ After you have determined the number of required SS and DS control loops as described above, you may still need to modify this number if you are using special hardware configurations, such as master-slave torque control (requires two axes on the same DSP > SS operation).
- ▶ Now that you know the number of SS and DS control loops as well as the number of axes that fulfill special requirements, you must define the settings of the respective CC 424(B) version in MPs 7610.0 and 7610.1. The following table illustrates the available PWM connections for setting the SS or DS functionality in MP7610.x. The connections printed in bold emphasize the default setting.

- MP 7610.0 - %1100
- MP 7610.1 - %1111

Configuration of MP 7610.x for CC 424(B) with up to 12 control loops								
Bit	**6 ctrl. loops**		**8 ctrl. loops**		**10 ctrl. loops**		**12 ctrl. loops**	
Value No.	0 (SS)	1 (DS)	0 (SS)	1 (DS)	0 (SS)	1 (DS)	0 (SS)	1 (DS)
Bits of **MP 7610.0** for 1st drive control board (= number of DSP)								
0	**X51 X53**	X51	**X51 X53**	X51	**X51 X53**	X51	**X51 X53**	X51
1	**X52 X54**	X52	**X52 X54**	X52	**X52 X54**	X52	**X52 X54**	X52
2	X55	**X55**	X55 X57	**X55**	X55	**X55**	X55	**X55**
3	X56	**X56**	X56 X58	**X56**	X56	**X56**	X56	**X56**
Bits of **MP 7610.1** for 2nd drive control board (= number of DSP)								
0					X57	**X57**	X59 X61	**X59**
1					X58	**X58**	X60 X62	**X60**
2					X59	**X59**	X63	**X63**
3					X60	**X60**	X64	**X64**
No. of axes (default)	**6**		**6**		**10**		**10**	
No. of axes (max.)	6		8		10		12	

Configuration of MP 7610.x for CC 424(B) with 14 control loops										
Bit	**14 ctrl. loops**									
Value No.	0 (SS)	1 (DS)								
Bits of **MP 7610.0** for 1st drive control board (= number of DSP)										
0	**X51 X53**	X51								
1	**X52 X54**	X52								
2	X55 X57	**X55**								
3	X56 X58	**X56**								
Bits of **MP 7610.1** for 2nd drive control board (= number of DSP)										
0	X59 X61	**X59**								
1	X60 X62	**X60**								
2	X63	**X63**								
3	X64	**X64**								
No. of axes (default)	10									
No. of axes (max.)	14									

- The last line – **No. of axes** (default) – shows the number of usable axes resulting from the default settings in MP 7610.x. If you want to use the specified number of axes on the 8-, 12- or 14-axis version, you must modify MP 7610.0 and MP 7610.1 correspondingly.
  - **No. of axes** (standard):
    MP 7610.0 %1100
    MP 7610.1 %1111
  - **No. of axes** (for 12 axes):
    MP 7610.0 %1100
    MP 7610.1 %1100
  - **No. of axes** (for 14 axes):
    MP 7610.0 %0000
    MP 7610.1 %1100

MP7610 **Control loop speed**
MP7610.0 Control-loop speed on the 1st controller PCB
Format: %xxxx
MP7610.1 Control-loop speed on the 2nd controller PCB
Format: %xxxx

7.5.5 PWM frequency

The control loops of the CC 424(B) can be assigned different PWM frequencies in MP2180.x.

There are three fundamental PWM frequencies: 3333 Hz, 4000 Hz and 5000 Hz.
A control loop can be operated at the fundamental PWM frequency or at double this frequency. The same PWM frequency (either the fundamental PWM frequency or double this frequency) must be set for all control loops of one drive control board.
However, the fundamental PWM frequency of the first drive control board is allowed to differ from that of the second drive control board. If you have set two different fundamental PWM frequencies for one drive control board, the error message **`C013 PWM frequency error`** will appear.

- Drive control board 1: X51 to X56
- Drive control board 2: X57 to X64

▶ In MP2180.x, set the PWM frequency (either the fundamental or double the fundamental PWM frequency) for every control loop.

The cycle time of the current controller depends on the PWM frequency:

$$T_I = \frac{1}{2 \cdot f_{PWM}}$$

If a control loop is operated at double the fundamental PWM frequency, the current controller cycle time is halved. However, this is possible only with double-speed control loops.

If you want to operate single-speed control loops at double the fundamental PWM frequency,

▶ Set MP2182.x = 1 in order to calculate the current controller cycle time from the fundamental PWM frequency although the control loop will be operated at double the fundamental PWM frequency.

If you operate a single-speed control loop at double the fundamental PWM frequency and half the current controller cycle time (MP2182.x = 0), the error message **`C017 PWM frequency too high`** will appear.

Current controller cycle time depending on the PWM frequency:

PWM frequency	Single-speed control loop		Double-speed control loop	
	MP2182.x = 0	MP2182.x = 1	MP2182.x = 0	MP2182.x = 1
3333 Hz	150 µs	Error C013!	150 µs	Error C013!
4000 Hz	125 µs	Error C013!	125 µs	Error C013!
5000 Hz	100 µs	Error C013!	100 µs	Error C013!
6666 Hz	Error C017!	150 µs[a]	75 µs	150 µs[a]
8000 Hz	Error C017!	125 µs[b]	62.5 µs	125 µs[b]
10000 Hz	Error C017!	100 µs[c]	50 µs	100 µs[c]

a. Fundamental PWM frequency of 3333 Hz
b. Fundamental PWM frequency of 4000 Hz
c. Fundamental PWM frequency of 5000 Hz

Switching the PWM frequency during operation

It may sometimes be necessary to increase the PWM frequency at high speeds, for example, if problems with the spindle temperature occur. With the CC 424(B), the PWM frequency can be switched during operation. You can switch only between the fundamental PWM frequency and double the fundamental PWM frequency. Since the current controller cycle time cannot be changed during operation, the entry MP2182.x = 1 is required. This means that at the same time that you enter double the fundamental PWM frequency in MP2180.x, you must set MP2182.x = 1.

Example:
Fundamental PWM frequency of 5000 Hz; a control loop is to be operated at double the fundamental PWM frequency and high speeds.

Default setting	**Machine parameters to be edited**	
	Single-speed control loop	**Double-speed control loop**
MP2180.x = 5000	MP2180.x = 10000	MP2180.x = 10000
MP2182.x = 0	MP2182.x = 1	MP2182.x = 1

MP2186.x and MP2188.x can by used to switch between simple and double PWM frequency, depending on the speed.

Set MP2182.x = 2 if you want to use automatic PWM frequency switching. The PWM frequency of a power stage is then doubled at a certain speed, based on a PWM frequency ≤ 5 kHz. In MP2186.x and MP2188.x you define the speeds at which switching occurs. The current controller cycle time is always (if MP2182.x = 2) based on the high PWM frequency.

MP2180.x **PWM frequency**

Input: 0: f_{PWM} = 5000 Hz
3200 to 3999: f_{PWM} = 3333 Hz
4000 to 4999: f_{PWM} = 4166 Hz (CC 424(B): 4000 Hz)
5000 to 5999: f_{PWM} = 5000 Hz
6000 to 7999: f_{PWM} = 6666 Hz
8000 to 9999: f_{PWM} = 8333 Hz (CC 424(B): 8000 Hz)
10000: f_{PWM} = 10000 Hz

MP2182.x **Cycle time of current controller at double the fundamental PWM frequency**

Input: 0 = Standard case:
MP2180 = [3333 to 5000 Hz] with single-speed axes
MP2180 = [3333 to 10000 Hz] with double-speed axes
(current controller cycle time = 1 / (2 * MP2180))
1 = Special case 1, only CC 424:
MP2180 = [6666 to 10000 Hz] for high PWM frequencies with single-speed axes
(current controller cycle time = 1 / MP2180)
2 = Special case 2, only CC424, CC61xx:
MP2180 = [3333 to 5000 Hz] with speed-dependent doubling (MP2186, MP2188) of the fundamental PWM frequency from MP2180 with double-speed axes
(current controller cycle time = 1 / (4 * MP2180))

Switching the PWM frequency depending on the speed

This function is used with high-speed spindle drives. This switchover is only possible for double-speed control loops.

In MP2186.x and MP2188.x, a speed-dependent hysteresis for switching the PWM frequency is specified. It only takes effect if the value in MP2188 is less than the value in MP2186.

This function is connected with MP2182.x and MP2180.x. Only if MP2182.x has the value 2 and MP2180.x is ≤ 5 kHz, does the switching of the PWM frequency take effect. Please note that the adjustment of the current controller (MP2420, MP2430) is based on the lower PWM frequency ≤ 5 kHz. Adaptation of the current-controller parameters and consideration of the power-module derating are performed automatically.

Using this function provides several benefits:

- At lower speeds and therefore a lower PWM frequency, the power module provides a comparatively high current. This results in a relatively high maximum motor torque.
- On the one hand, losses due to harmonics in the motor become more important as the speed increases, and on the other hand, the relationship between the electrical frequency and the PWM frequency worsens. These two disadvantages can be counteracted by increasing the PWM frequency. The resulting reduction of the current normally is insignificant, since in part due to motor characteristics, very high motor currents are mostly no longer possible or needed at higher speeds.

Note

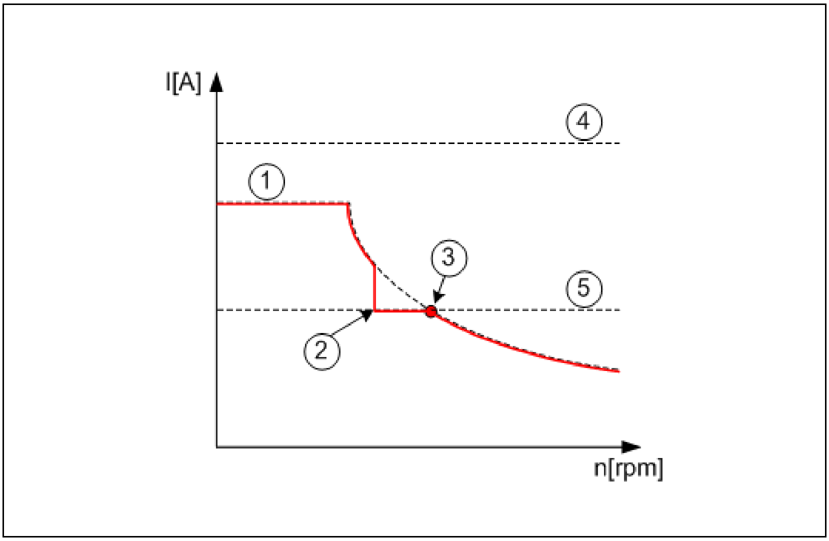
The speed-dependent switching of the PWM frequency is only permitted with power modules from HEIDENHAIN.

Danger

Speed-dependent switching of the PWM frequency with non-HEIDENHAIN power modules can lead to malfunctions, and possibly to damage of the power modules. Therefore, only use this function with power modules from HEIDENHAIN.

When determining the optimum switching speed for the PWM frequency, you should consider that the maximum motor current decreases as the speed rises, due to the finite dc-link voltage.
The current for the drive depends on the maximum permissible motor current and power-module current. The smaller of the two values limits the current for the drive. The value of the maximum power-module current is reduced by approx. 30% because of the derating when doubling the PWM frequency. During derating the rated current I-N and the maximum current I-MAX are reduced by the same factor.

The optimum switching speed results from the intersection of the maximum current curve of the motor with the maximum current curve of the power module for the high PWM frequency. You can determine the maximum current curve of the motor spindle by using the TNCscopeNT software. Record the current Inoml/S in dependency of the speed. The spindle must be accelerated to the maximum speed so that the derating behavior can be seen in the curve. The figure shows the behavior when the values entered for the switching speed (MP2186.x, MP2188.x) are too low. This then results in a speed range where the current for the motor is less than the permitted and maximum current, resulting in inconsistencies in the motor's torque behavior.



- 1: Red line: Maximum current for the drive, resulting from the entries in MP2186.x and MP2188.x
- 1: Broken line:
  Maximum possible current for the drive (Imax of motor)
- 2: Switching point for the PWM frequency set too low
- 3: Optimum switching point for the PWM frequency
- 4: Maximum power-module current at low PWM frequency
- 5: Maximum power-module current at high PWM frequency (Imax of power module)

Summary:

- The speed is switched at the intersection of the two current curves (Imax of motor, Imax of power module) so that no inconsistencies in the torque behavior of the motor occur.
- For better controllability (no harmonics at higher PWM frequencies), it might already make sense to switch at lower speeds.
- The best speed to switch at must be determined by experimenting. The value above should serve as an initial value.

MP2186.x **Speed-dependent switching of the PWM frequency**
Input: 0 to 100 000 [rpm]
MP2186.x specifies the shaft speed at which the PWM frequency is switched to twice the PWM frequency

MP2188.x **Speed-dependent switching of the PWM frequency**
Input: 0 to 100 000 [rpm]
MP2188.x specifies the shaft speed at which the original PWM frequency is returned to from the doubled PWM frequency (as the result of MP2186.x).
MP2188.x must be < MP2186.x

7.6 PLC Cycle Time

MP7600.x is of no importance on the CC 424(B).

- Enter the PLC cycle time in ms in MP7602.

MP7602	**PLC cycle time**
Input:	0 to 60 [ms]
	0 to 10: 10.8 ms

7.7 Monitoring Functions

The KTY temperature sensor of the motors is monitored by the control for excessive and insufficient temperatures. If the KTY is not to be evaluated (e.g. because the temperature sensor is not doubly isolated), this function must be deactivated with MP2220.x bit 4 = 1.

The CC 424(B) monitors the input frequency of the speed encoders. If this monitoring leads to problems (e.g. unjustified responses), it can be deactivated with MP2220.x bit 7 = 1. The following error messages can appear:

- Speed encoder: **8860 Input frequency from speed encoder <axis>**
- Position encoder: **8870 Input frequency from position encoder <axis>**

Only one encoder is used for capturing the speed and position when linear and torque motors are being used. If such an axis is removed from the closed-loop control and later reintroduced, a mechanical offset can occur. This offset is not fixed "in one blow," but instead is adjusted by gradually raising the k_V factor from 0 to the original value. This function is deactivated with MP2220 bit 8 = 1.

MP2220.x **Monitoring functions**
Input: Bit 4 – Monitoring for excessive temperature
0: Active
1: Inactive
Bit 5 – Monitoring for insufficient temperature
0: Active
1: Inactive
Bit 6 – Reserved
Bit 7– Monitoring of encoder input frequency
0: Active
1: Inactive
Bit 8 – Adjust mechanical offset by gradually increasing the k_V factor
0: Active
1: Inactive

7.8 Special Functions

7.8.1 Multifunction Filter

With the CC 424(B), you can influence the manipulated variable of the speed controller (= nominal current) and the position controller (= nominal speed) by means of up to five freely definable filters per axis. These filters are multifunctional filters, which means that the filter type of each individual filter order can be selected as desired. They are also effective for the spindle(s).

Objective of the filters

The first objective when adjusting a machine is the optimization of the control loop in the current and speed controller. The increase of the P component of the control loops in order to raise the dynamics of the machine is the main aspect of this. If a control loop is at the oscillation limit, these oscillations can be damped with filter functions, so that the P components can be increased again.

The second objective when adjusting a machine is the optimization of the position controller. Here it is attempted to increase the k_V factor in the position controller, in order to simultaneously increase the machine's performance (the acceleration behavior, for example). The procedure is always to increase the k_V factor to the oscillation limit, damp these oscillations with the filters, and then increase the k_V factor again.

Types of filters

Three different types of filters per axis are available for selection:

- **PT2 low-pass**
  - Use:
    - Oscillations in the upper frequency range (typically: from 500 Hz)
    - High-frequency noises on axes (such as during switch-on)
- **Band-rejection filter**
  - Use:
    - Oscillations in the middle frequency range (typically: between 100 Hz and 2.5 Hz)
  - Typical settings:
    - Damping from 6 to 9 dB
    - Bandwidth: equal to the center frequency, constant from 500 Hz
  - Disadvantage:
    - These can strengthen oscillations in the lower frequency range
- **Phase increase**
  - Use:
    - Oscillations in the lower to middle frequency range, which occur because of an insufficient phase reserve
    - Oscillations in the lower frequency range, for which band rejection would excessively decrease the amplitude
  - Typical settings:
    - Phase from 20° to 80°
    - Center frequency: Frequencies from 3 to 400 Hz
    - Bandwidth: Equal to the center frequency (oscillation frequency)
  - Disadvantage:
    - The control-loop gain above the center frequency is increased. The increased use of band-rejection filters can become necessary, or the P component might need to be reduced.
  - Note: After the settings have been made, the stability of the control loop must be checked again (P and I component)

Recommended types of filters

Experience has shown that the band rejection of the multifunction filters is to be used for damping oscillations in the **speed controller.**

On the other hand, the tendency of the **position controller** to oscillate should be counteracted with IPC (Integral Phase Compensation). Only if this adjustment does not lead to the desired result can the multifunction filters such as the phase increase (better, since it does not facilitate oscillations at lower frequencies as much) or the band-rejection filter be used.

Since the ambient conditions can be so different, the use of the filters must be checked separately in every case. The TNCopt PC software from HEIDENHAIN should always be used, so that the sequence of the adjustment matches the ideal case. This manual can only present recommended guidelines and procedures.

Possible settings for the multifunction filters

	Filter 1	**Filter 2**	**Filter 3**	**Filter 4**	**Filter 5**
Filter selection **0** = Filter not active **1** = PT2 low-pass filter (speed controller) **2** = Band-rejection filter (speed controller) **3** = Phase increase (speed controller) **11** = PT2 low-pass filter (position controller) **12** = Band-rejection filter (position controller) **13** = Phase increase (position controller)	MP2562.x	MP2563.x	MP2564.x	MP2565.x	MP2566.x
■ **PT2 low-pass filter:** No effect ■ **Band rejection:** Damping [dB] ■ **Phase increase:** Phase [0 - 90°]	MP2542.x	MP2543.x	MP2544.x	MP2545.x	MP2546.x
■ **PT2 low-pass filter:** Corner frequency [Hz] ■ **Band-rejection filter:** Center frequency [Hz] ■ **Phase increase:** Center frequency [Hz]	MP2552.x	MP2553.x	MP2554.x	MP2555.x	MP2556.x
■ **PT2 low-pass filter:** No effect ■ **Band-rejection filter:** Bandwidth [Hz] ■ **Phase increase:** Bandwidth [Hz]	MP2572.x	MP2573.x	MP2574.x	MP2575.x	MP2576.x

The filters can be used in the position or speed controller in any combination, i.e. even if the first filter is deactivated for the 1st axis,

- MP2542.0
- MP2552.0
- MP2562.0 = 0
- MP2572.0

the second filter, for example (in this case phase increase in the speed controller), can be activated for the first axis:

- MP2543.0 = 40
- MP2553.0 = 120
- MP2563.0 = 3
- MP2573.0 = 120

Changes to CC 422

Due to the multifunction filters on the CC 424(B), MP2540.x and MP2550.x (band rejection) are no longer active.

Filters in the position encoder

The k_V factor can be increased by using the filters in the position controller. After the k_V factor has been increased up to the oscillation limit, a band-rejection filter can be defined for the oscillation frequency so that the k_V factor can be further increased.

MP2542.x **Damping/phase increase for filter 1**
Input: 0 to 99.0 [dB]

MP2543.x **Damping/phase increase for filter 2**
Input: 0 to 99.0 [dB]

MP2544.x **Damping/phase increase for filter 3**
Input: 0 to 99.0 [dB]

MP2545.x **Damping/phase increase for filter 4**
Input: 0 to 99.0 [dB]

MP2546.x **Damping/phase increase for filter 5**
Input: 0 to 99.0 [dB]

MP2552.x **Center/corner frequency for filter 1**
Input: 0 to 30 000.0 [Hz]

MP2553.x **Center/corner frequency for filter 2**
Input: 0 to 30 000.0 [Hz]

MP2554.x **Center/corner frequency for filter 3**
Input: 0 to 30 000.0 [Hz]

MP2555.x **Center/corner frequency for filter 4**
Input: 0 to 30 000.0 [Hz]

MP2556.x **Center/corner frequency for filter 5**
Input: 0 to 30 000.0 [Hz]

MP2562.x **Filter type for filter 1**
Input: 0: No filter
1: PT2 low-pass filter (speed controller)
2: Band-rejection filter (speed controller)
3: Phase increase (speed controller)
11: PT2 low-pass filter (position controller)
12: Band-rejection filter (position controller)
13: Phase increase (position controller)

MP2563.x **Filter type for filter 2**
Input: 0: No filter
1: PT2 low-pass filter (speed controller)
2: Band-rejection filter (speed controller)
3: Phase increase (speed controller)
11: PT2 low-pass filter (position controller)
12: Band-rejection filter (position controller)
13: Phase increase (position controller)

MP2564.x **Filter type for filter 3**
Input: 0: No filter
1: PT2 low-pass filter (speed controller)
2: Band-rejection filter (speed controller)
3: Phase increase (speed controller)
11: PT2 low-pass filter (position controller)
12: Band-rejection filter (position controller)
13: Phase increase (position controller)

MP2565.x **Filter type for filter 4**
Input: 0: No filter
1: PT2 low-pass filter (speed controller)
2: Band-rejection filter (speed controller)
3: Phase increase (speed controller)
11: PT2 low-pass filter (position controller)
12: Band-rejection filter (position controller)
13: Phase increase (position controller)

MP2566.x **Filter type for filter 5**
Input: 0: No filter
1: PT2 low-pass filter (speed controller)
2: Band-rejection filter (speed controller)
3: Phase increase (speed controller)
11: PT2 low-pass filter (position controller)
12: Band-rejection filter (position controller)
13: Phase increase (position controller)

MP2572.x **Bandwidth for filter 1**
Input: 0 to 30 000.0 [Hz]

MP2573.x **Bandwidth for filter 2**
Input: 0 to 30 000.0 [Hz]

MP2574.x **Bandwidth for filter 3**
Input: 0 to 30 000.0 [Hz]

MP2575.x **Bandwidth for filter 4**
Input: 0 to 30 000.0 [Hz]

MP2576.x **Bandwidth for filter 5**
Input: 0 to 30 000.0 [Hz]

7.8.2 Filter order for separate low-pass filter in the speed controller

Application

The function of MP2560.x (CC 422: low-pass filter) has changed:
If a low-pass filter is used with the CC 424(B), the filter order of the low-pass filter can be set in MP2560.x However, MP2560.x = 0 should be entered under standard conditions.

Spindle: High-frequency spindles often cause considerable current noise (shown by **`I nom`** or **`Utilization`** on the oscilloscope). The optimization goal is as little current noise as possible at high spindle speeds. MP2560.x = 10 to 20 should be used as a starting value.

Axes: The low-pass filter should be used if the actual speed has "a lot of noise," for example MP2560.x = 1 or 2. If there is no improvement, set MP2560.x = 0.

For adjustment of the filters, See "Adjustment of the speed controller" on page 1126.

Machine parameters

MP2560.x	**Filter order of the low-pass filter**
Input:	0 to 20 Recommended input value: 0 Recommended input value if much current noise from high-frequency spindles: 10 to 20

7.8.3 Dynamic determination of load

General information

During traverse motions, each machine axis behaves differently depending on the various loads, such as the differing inertias of the workpieces.

In connection with the CC 424(B) controller unit, the iTNC offers you the possibility of optimizing the machine according to load classes. Before the machining process, the optimum settings for a workpiece with a certain load class can be entered.

Determining the load

The effects of the inertia of the workpieces can be detected with a measuring cycle created by the OEM. A measurement that determines the maximum occurring torque (target value in % of the rated torque without decimal places) between two points in time is started. At the same time, the machine is optimized for this load class, and the machine parameters determined are saved in a machine-parameter subfile.

The load class is determined for each workpiece before machining starts (via manual entries or by determining the load with an NC macro), and the corresponding, optimized machine-parameter subfile is selected.

Procedure

The following procedure is recommended for determining and using a load class.

By the OEM: Determining the load class and creating the machine-parameter subfiles

- Optimizing the machine and saving the results without load
- Loading the machine with a mass appropriate for the load class to be determined
- Starting the measurement, traversing the axis to be measured, and stopping the measurement
  Example of a corresponding NC program:

```
0   BEGIN PGM measure_a MM
1   L C+0 R0 FMAX
2   FN 17: SYSWRITE ID 621 NR0 IDX4 = +0
3   L IA+180 R0 FMAX
4   L IA-180 FMAX
5   FN 18: SYSREAD Q1 = ID621 NR0 IDX4
6   END PGM measure_a MM
```

- Optional: Saving the measured results in a variable:
  FN 17 SYSWRITE ID 590 NR 2 IDX <axis> IDX = 1 to 30 (is not deleted during program selection)
  FN 17 SYSWRITE ID 590 NR 3 IDX <axis> IDX = 1 to 30 (protected during a power failure)
- Optional: Reading the saved value
  FN 18 SYSREAD ID 590 NR <2 or 3> IDX <axis> (IDX = 1 to 30)
- Optimizing the machine and saving the results in a machine-parameter subfile for this load class
- Performing the measurements for each load class as described above

By the customer: Selecting the machining method optimized for the load

- Entering the load class (or mass) before machining, or determining the load class with the NC macro from the OEM as described above.
- Activating the correct machine-parameter subfile depending on the measured result
  FN 17 SYSWRITE ID 1020 NR 1 = <line number> (when using the keyword **MPFRAGMENTFILE = <path/file name>** in OEM.SYS)

Note

- Retention forces and friction forces are included in the measured value
- Deviations in the utilization due to oscillations or torque ripple are included in the measured value
- The override of the feed-rate potentiometer is taken into account
- The determined value depends on the axis parameters. Use axis parameters with stable behavior for the determination.

7.8.4 LIFTOFF function in case of a power failure

General information

If the power fails and LIFTOFF is enabled (M148 must be active, column **LIFTOFF** in the tool table = Y, PLC: M4620=1), an attempt is made to lift the tool from the contour by the distance given in MP1160 with the help of the remaining energy of the dc-link.
Certain conditions must be maintained before and during LIFTOFF.

- The 24-V power supply must be maintained for at least 1 second (USB for 24 V, or buffer capacity or capacitor). HEIDENHAIN offers the CML 110 (Capacitor Module for Low Voltage) for this.
- The current and speed controllers may not be switched off (e.g. via PLC Module 9161)
- AC-Fail may not be evaluated (MP2150 = 3).
- The wye-delta contactor combination may not fail, otherwise the spindle could not be controllable during liftoff.

Note

LIFTOFF only functions with HEIDENHAIN inverters.

Note

Perform the LIFTOFF function on the machine to check whether the tool lifts off the complete distance defined in MP1160 without problems.
Due to the dynamics of a machine, it may occur that the LIFTOFF (if values > 10 mm are defined in MP1160) cannot be executed completely and is aborted with an error message.

Danger

- Limit-switch monitoring is not active during LIFTOFF.
- LIFTOFF must be deactivated while exchanging tools.

MP1160	**LIFTOFF in case of power failure**
Input:	0 to 30.0000 [mm] Default: 0.1 [mm]

Additional settings

Since the residual energy of the dc-link is only available for a controlled LIFTOFF for a very short time, under certain circumstances the prescribed distance may not be reached. Since even times on the scale of milliseconds can be of use here, a special possibility was created in order to react to a power failure more quickly. The starting point is the processing time for the HSC nominal position value filters, which, due to the number of filter coefficients (filter order) to be calculated, lead to a delay in the reaction time. This number of filter coefficients to be calculated can be reduced with MP1262 (filter order for HSC filters) and MP1263 (filter order for advanced HSC filters). The starting point is the filter order of 31 recommended by HEIDENHAIN, which is entered as a default value and should suffice for the LIFTOFF function in most cases.

If this is not the case, then you can reduce the reaction time via the filter order. Since the coefficients are calculated in the interpolator clock cycle of usually 3.0 ms, the reaction time decreases correspondingly per decrement of the filter order.

Note

A modification of the filter order means a **basic change** in the filter characteristics of the nominal position value filters set, and can therefore have significant influences on the behavior of the machine. The effects must be checked individually, and the nominal position value filters must be changed if necessary.

Starting from the following settings for MP1262 and MP1263, checking and re-adjustment of the nominal position value filters is strongly recommended:

- MP1262 ≤ 1.1 / (MP1212*interpolator cycle time)
- MP1263 ≤ 0.67 / (MP1213*interpolator cycle time)

MP1262 **Filter order used for HSC filters**
Input: 0 to 31 [filter order]
Default: 31

MP1263 **Filter order used for advanced HSC filters**
Input: 0 to 31 [filter order]
Default: 31

With MP2192 you can enter a percent value of the dc-link voltage Uz as trigger threshold for the detection of a power failure. If the dc-link voltage Uz drops below the value resulting from MP2192, an NC stop and LIFTOFF are performed.
If you enter the value 0 for MP2192, only the UV ready signal is monitored. If you enter a value in MP2192 greater than the typical value of 400 V for a powerfail, LIFTOFF is performed correspondingly earlier and with more energy in the dc-link. MP2192 must be > MP2194.
The entered value must be > 400 V for this function to be useful. HEIDENHAIN recommends a value of 80%.

Attention

- This function can only be used in combination with regenerative HEIDENHAIN inverters.

With MP2194 you enter a voltage value in [V] of the dc-link voltage. If the dc-link voltage Uz drops below the value in MP2194, the spindle is actively braked.
Normally energy is recovered during braking of the axes with LIFTOFF. This does not apply to linear motors with poor efficiency. Here energy is needed in order to brake an axis. This means that a load is placed on the dc-link during the braking procedure. In certain circumstances this can mean that there is no longer enough energy to perform LIFTOFF. The only energy source in the system may be a rotating spindle. If the dc-link voltage drops below the value in MP2194, the spindle is braked and energy is recovered. This should make enough energy available in order to perform LIFTOFF completely.
The entered value must be > 400 V for this function to be useful. HEIDENHAIN recommends a value of 450 V.

Attention

- This function can only be used in combination with regenerative HEIDENHAIN inverters.

MP2192 **Threshold sensitivity for LIFTOFF**
Input: 0 to 100 [%]

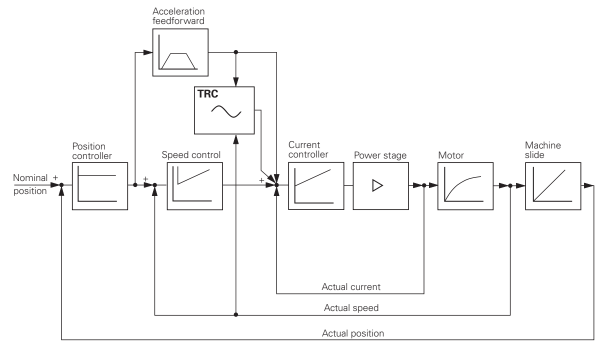
MP2194 **DC-link voltage as of which the spindle is braked in a powerfail**
Input: 0 to 3000 [V]

7.8.5 TRC – Torque Ripple Compensation

General information

Certain motors with permanent magnets (linear, torque and some synchronous motors) have an increased, position-dependent variation of the motor torque (not QSY motors from HEIDENHAIN). This can be the result of two things:
- During idle running, the cogging due to attractive forces of the permanent magnets
- When under load, the torque ripple from higher harmonics, resulting from the opposing electromotive forces (generator effect)

In practice, both causes always occur together, i.e. the torque of the motor is subject to periodic oscillations. This can have a negative effect on the controllability of the motor, which can result in a greater following error, and under circumstances, lower surface quality of the workpiece.
To compensate for the **cogging,** a compensation current ascertained specifically for each motor can now be added.



Activating TRC

TRC can only be activated via a special compensation file. The settings in this file can only be made with the TNCopt commissioning software from HEIDENHAIN (as of version 2.3). Please refer to the TNCopt documentation. The compensation current is determined with a special method for measurement, and the parameters for calculating this compensation are stored in a compensation file on the control. The iTNC 530 then takes these parameters into account when calculating the controller parameters.

- Directory: **`PLC:\MP\TRC`**
- File name: **`xx_<Motor_name>.TRC`** or new **`xx_<Motor_name>.CMP`**
  - **`xx`**: Index of the axis (e.g. 00 = 1st axis, X axis)
  - **`<Motor_name>`**: Name of the motor from the motor table (max. 29 characters)
  - **`.CMP`** (old **`.TRC`**): File extension for "Torque Ripple Compensation"

An entry in MP2260.x specifies whether torque ripple compensation is to be performed for an axis. If MP2260.x is followed by a blank line (no entry), no compensation current is calculated for this axis.

Example:

```
MP2260.0:    00_MotNameFromMotTab;Motor of 1st axis
MP2260.1:    ;Motor of 2nd axis, no compensation
```

With MP2261.x, compensation values contained in the compensation file (.*cmp, defined in MP2260) can be deactivated for specific axes. For master-slave torque control, a separate TRC file now be defined for the master axis and the slave axis in order to compensate gear-transmission errors separately.

Note

- The TRC function can only be used with PWM frequencies up to 5 kHz.
- A TRC file can only be used on the control on which the adjustment has been made.
- A TRC file must be re-created if the motor, rotary encoder or angle encoder is exchanged.
- A TRC file can only be generated for synchronous motors or for linear or torque motors.

MP2260.x **Name of the file for torque ripple compensation (TRC)**
Input: xx_<MotorNameFromMotorTable>.CMP (generated in TNCopt). If no .CMP is found, .TRC is accessed
No entry: No compensation

MP2261.x **Deactivate compensation**
Input: %00000000000000
1 = Compensation not active
Bit 0: Torque ripple compensation
Bit 1: Gear error compensation

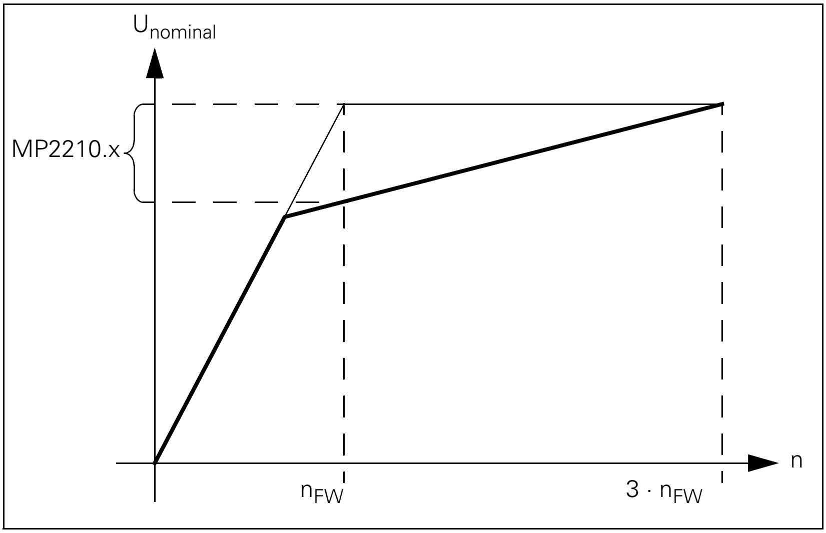
7.8.6 Peculiarities in weakened-field operation

Reduction of the magnetization current

Some of the asynchronous spindle motors require a high magnetizing current at low speeds ($n < n_{field\ weakening}$). This magnetizing current can, however, lead to thermal motor problems at the rpm for field weakening $\mathbf{n}_{FW}$ when there is no load.

The nominal voltage (and so also the nominal magnetization current) at the rpm for field weakening $\mathbf{n}_{FW}$ when there is no load can be reduced with MP2210.x The entered reduction results in a profile of the nominal voltage as shown in the figure below. The maximum nominal voltage is reached when $n = 3 \cdot n_{field\ weakening}$. The nominal voltage can be reduced by max. 60 % (MP2210.x = 60).

If a load is placed on the drive, then the nominal voltage is increased again automatically in order to improve the dynamics.



MP2210.x	**Reduction of the nominal voltage at the rpm for field weakening when there is no load**
Input:	0 to 60 [%] 0: Function inactive

Note

Please note that the reduction of the magnetization current can lead to a limitation of the drive's dynamics. However, this depends on the drive, and must be judged separately in each case.

Setting the reduction of the magnetization current

If thermal problems arise on an asynchronous spindle with no load during weakened-field operation, and these problems cannot be traced to an overload or other mechanical problems, then please reduce the magnetization current as follows:

- Reduce the nominal voltage in steps of 10% in MP2210.x.
- Reduce MP2210.x until a stable temperature within the motor specifications is reached when there is no load.

7.9 Stick-Slip Friction Compensation at Quadrant Transitions

Stick-slip friction compensation for the CC 424(B) has been improved in comparison to stick-slip friction compensation for the CC 422.
With the CC 422, the quadrant transition is influenced by MP2612.x only after the zero crossover of the speed. With the CC 424(B), the compensation already begins before the zero crossover of the speed.

With the CC 424(B), the parameters MP2612.x and MP2614.x now function with respect to distance rather than time (unit: [mm]).
This makes it possible to compensate quadrant transitions independently of velocity, acceleration, and diameter.

If stick-slip compensation is used at the same time that an HSC filter is being used, then in rare cases unwanted movements of the machine can occur (brief movement in wrong/reversed direction). You can avoid this effect by using the triangle filter instead of the HSC filter.

MP2610.x	**Friction compensation at low motor speed**
Input:	0 to 30.0000 [A] (effective value) 0: No friction compensation
MP2612.x	**Distance before the reversal point from which a reduction of the current from MP2610.x is to go into effect**
Input:	0 to 1 000 [mm] or [°] 0: No friction compensation 0.1: Typical input value
MP2614.x	**Distance after the reversal point from which the current from MP2610.x is to go into effect again**
Input:	0 to 1 000 [mm] or [°] 0: Friction compensation same as CC 424 0.1: Typical input value

Module 9311 Dynamically change values for friction compensation
Module 9311 is used at run-time to prescribe other values for the friction compensation. The original values from MP2610.x, MP2612.x and MP2614.x are temporarily overwritten in the DSP. The MP file remains unchanged.

Conditions:

- This function is supported as of the DSP hardware CC 424(B).

Call:
PS B/W/D/K <Axis number>
PS B/W/D/K <Current in [mA]>
0 to 30000 replaces the value in MP2610.x
PS B/W/D/K <Path in [0.1 µm]>
0 to 10000 replaces the value in MP2612.x
PS B/W/D/K <Path in [0.1 µm]>
0 to 10000 replaces the value in MP2614.x
CM 9311

Error recognition:

Marker	Value	Meaning
M4203	0	New values assumed for axis number
	1	Error code in W1022
W1022	1	Invalid value as replacement for machine parameter
	2	Invalid axis number programmed
	19	Function is not supported by the DSP board (CC 422)
	24	Call was not from a cyclic program

7.10 Field Orientation

General information

If a linear, torque or synchronous motor is used with an incremental encoder without a Z1 track or an unaligned encoder with EnDat interface, there is no association between the encoder and the rotor magnets. The field angle must be determined before this motor can be moved.

The iTNC 530 uses the "field orientation" function to determine the field angle for the motors listed above. The association between the encoder and the rotor magnet (field angle) is determined and stored.

> Note
>
> The "field orientation" function can be performed only if the current controller is already adjusted!

Regarding the **motor.mot** motor table, the field orientation must be performed for the following drives:

- Linear motor with absolute encoder with EnDat interface **(SYS = 3)**
- Synchronous or torque motor with unaligned rotary encoder with EnDat interface **(SYS = 5)**
- Synchronous or torque motor with incremental rotary encoder without Z1 track **(SYS = 6);** one reference mark per revolution
- Synchronous or torque motor with incremental rotary encoder with distance-coded reference marks **(SYS = 7)**
- Linear motor with incremental linear encoder with distance-coded reference marks **(SYS = 8)**

Absolute encoder with EnDat interface	Incremental encoders
As soon as the absolute position of the encoder has been read, the absolute position and determined field angle are associated. The field angle is associated with the zero position of the encoder.	After switching on the drive, the motor orients itself (rough orientation; the message **Finding the field angle** appears). The drive is ready for operation after this procedure. The field angle is determined and associated as soon as the reference mark/s is/are traversed during the first motor motion.

> Danger
>
> If the speed encoder is exchanged, the "field orientation" function must be rerun.

Field orientation with absolute encoder (EnDat)

As soon as the absolute position of the encoder has been read, the absolute position and determined field angle are associated.
The field angle is associated with the zero position of the encoder.

Field orientation via encoder with Z1 track

After switching on the control, the motor orients itself (rough orientation) via the Z1 track of the encoder. The drive is ready for operation after this procedure. The field angle is determined and associated as soon as the reference mark is traversed during the first motor motion.

General information about encoders for direct drives

- An absolute encoder with EnDat interface should be used, since the absolute position value is available directly after switch-on, and the field angle can be associated immediately. This means that the motor can be controlled immediately.
- The encoder should have a high line count. This leads to better controllability.
- With incremental encoders the motor must first be moved a "minimum" distance in order to determine a field angle with which the motor can be moved until the reference mark. Only after the reference mark has been traversed can the field angle determined during commissioning be assigned.
- If excessive clamping of the axis prevents the "minimum" motion for determining the field angle, then no field angle can be determined and the axis cannot be controlled. In this case the clamping must be undone for the field angle to be determined. If this is not possible, because the axis would fall down, then an absolute encoder with EnDat interface must be used.

7.10.1 Possibilities for determining the field angle

There are various possibilities for determining the field angle:

- The field angle is determined automatically when the drive is switched on, without any motion of the motor. The method of determination is set in MP2250.x. The field angle is stored after it has been determined. This field angle is used when the motor is switched on again. The **FIELD ORIENT.** soft key has no function.
- By pressing the **FIELD ORIENT.** soft key once while the motor is being commissioned. The soft key appears in the **Commissioning Current Controller** operating mode. After pressing it, the motor moves. The field angle is determined and stored during this motion. This field angle is used when the motor is switched on again. A plausibility test is run during the field angle determination.

> **Attention**
>
> This method cannot be used for hanging axes (with 100% weight compensation), since the brakes are not applied and the monitoring functions are deactivated!

▶ Select the method for field angle determination in MP2254.x.

HEIDENHAIN recommends using MP2254.x = 2 when commissioning new drive systems (such as machine prototypes), because the plausibility tests will be run. After successful commissioning, MP2254.x = 0 can be used to save time (such as for series production of the machine).

In certain cases it can be of advantage to determine the field angle via MP2254.x = 3. This mode can be used if

- there are no brakes,
- in the **Commissioning Current Controller** mode of operation, where the brakes are always open,
- the user ensures that the brakes can be opened manually or with the PLC.

MP2254.x **Determining the field angle**

Input: 0: Field angle is determined during operation; soft key has no function (without plausibility test)
2: Only CC 424(B): Field angle is determined via soft key; motor motion is permitted (with plausibility test)
3: Only CC 424(B): Same as 2, but the drive must no longer be switched on by the PLC. The drive is moved immediately after the **FIELD ORIENT** soft key is pressed.

> **Note**
>
> For synchronous spindles, the field angle should be determined via the **FIELD ORIENT.** soft key (MP2254.x = 2), since this is a more exact determination.

Plausibility test

This tests several machine parameters and parts of the circuitry for their plausibility:

- Encoder line count
- Number of pole pairs
- Rotational direction of the electrical field
- Traverse distance per electrical revolution

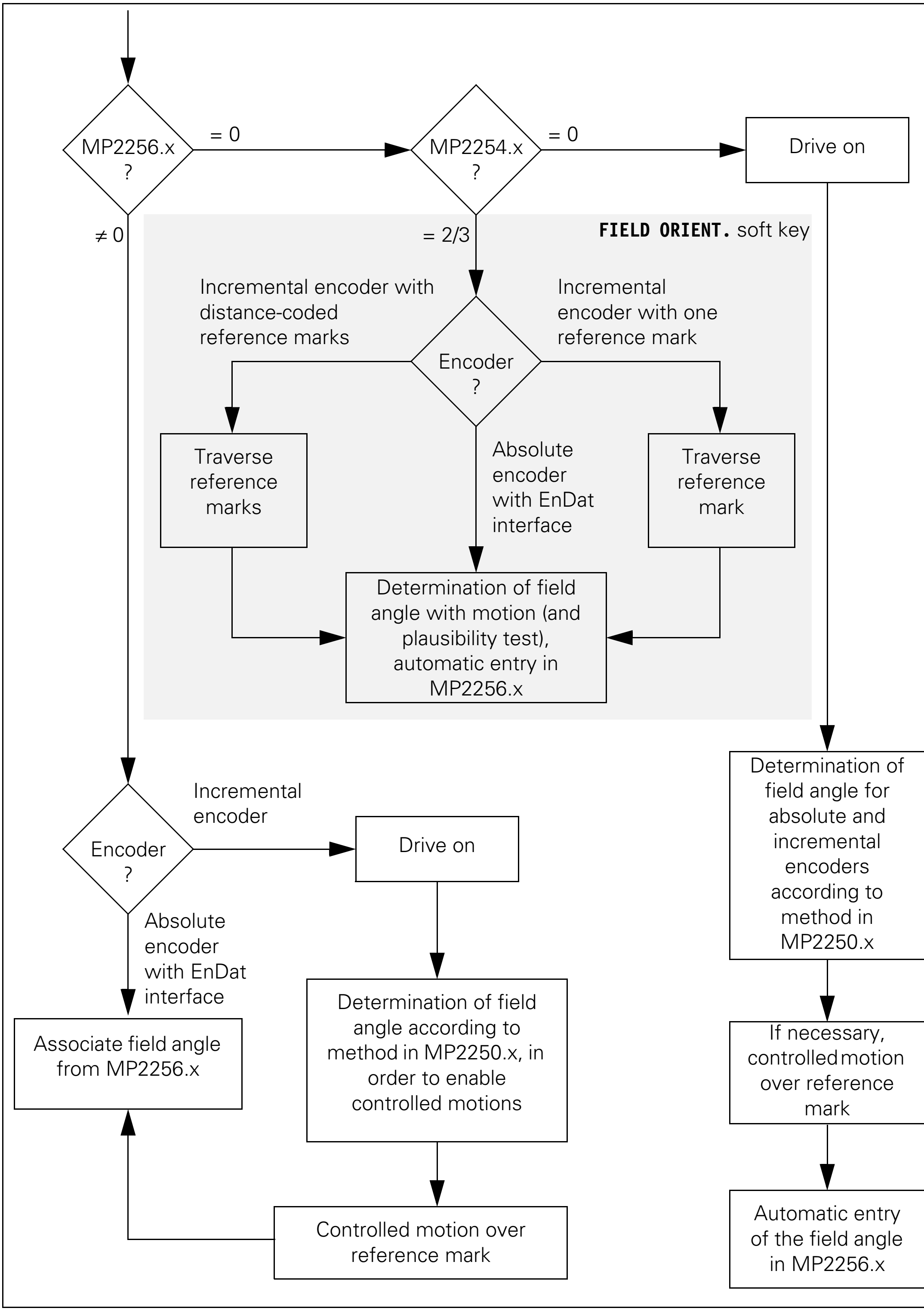
Note

This method for determining the field angle is recommended for commissioning, new designs, and other similar situations.

The following messages can appear during the plausibility test:

- **8630 Field orient. successful**
  Indicates that the field angle was successfully determined and stored in MP2256.x.
- **8B10 Wrong traverse direction**
  Indicates that the rotational direction of the electrical field does not match the counting direction of the encoder.
  Error fix: Change the entry in MP2204.x.
- **8B20 Error field orientation**
  Indicates that no usable measuring results could be determined.
  A common cause is incorrect values in MP331.x, MP332.x and MP1054.x.
  A further cause could be that the motor is moving against a resistance (e.g. brake is still active, bellows, limit stop) or that the mechanics are too stiff.

Overview of the field orientation



7.10.2 Determination of the field angle without motor motion (MP2254.x = 0)

A distinction must be made if you intend to determine the field angle without motor motion:

- Commissioning: No field angle has been determined yet (MP2256.x = 0)
- A field angle has been already determined (MP2256.x ≠ 0)

The field angle is determined automatically after switching on the drive. This process lasts approximately 4 to 6 seconds (the PLC program must not rescind the drive release during this time). The **`Finding field angle`** message appears.

If the power module is not active before the determination of the field angle begins, the error message **`8B40 No drive release <axis>`** appears. If the power module switches off during the determination, **`8B50 Axis module <axis> not ready`** appears.

Field angle not yet determined (MP2256.x = 0)

If the field angle on this machine was not yet determined after the control was started (MP2256.x = 0), the determination starts automatically. The method for determining the field angle is stored in MP2550.x. The determined field angle is stored in MP2556.x.

Field angle already determined (MP2256.x ≠ 0)

If the field angle on this machine was already determined after the control was started (MP2556.x ≠ 0), a distinction must be made:

- If an absolute encoder with EnDat interface is being used:
  The absolute position of the encoder is read immediately after the control has been started. The field angle from MP2256.x is assigned to this position. Therefore, the first motor motion already occurs with the determined field angle.
- If an incremental encoder is being used:
  Immediately after the control has been started and the control voltage has been switched on, then depending on MP2250.x a field angle is determined with which the motor can be traversed over the reference mark. After traversing the reference mark, the field angle from MP2256.x is assigned. The subsequent motor motions utilize the field angle from MP2256.x.

Determining the field angle

There are three methods for determining the field angle without motor motion:

- Method 2: Current pulses are output with the brakes applied, and the absolute rotor position is determined from the reaction. A "minimum" movement of the motor must be possible when the brakes are applied.
- Method 3: Method 3 functions in the same manner as Method 2, but with the difference that the motor brakes are not applied. Therefore, this method is not suitable for hanging axes. However, this method can lead to more exact results than Method 2, so it should be used for synchronous spindles. Minimal spindle movements can occur during field angle determination.
- Method 4: This new method was introduced for determining the field angle so that the relationship between the position of the incremental encoder and the position of the rotor magnets can be established even if there is considerable noise in the encoder signals.

- In MP2250.x, select the method to be used for determining the absolute rotor position.

> Note
>
> Standstill monitoring is active while determining the field angle. If it responds for motors without motor brakes, increase the threshold in MP1110.x. Afterwards, reset MP1110.x to the original value.

MP2250.x	**Determining the field angle without motor motion**
Input:	0: Same as input value 2 1: Reserved 2: Method 2 (brakes applied) 3: Method 3 (same as Method 2, but motor brakes not applied) 4: Method 4 (if there is a lot of noise in the encoder signals)
MP2252.x	**Reserved**
Input:	Enter 0

7.10.3 Determination of the field angle with motor motion (MP2254.x = 2/3)

Since the motor moves in a certain direction while finding the field angle, it should be near the midpoint of the traverse path before the field angle is determined.
Axes with linear/torque motors can be slid "by hand" if the brakes are not applied.

While the field angle is being found, the speed controller and position controller are opened and the drive controller is active. This means that the motor is moved (approx. 2 pole pairs) and the brake must be open until the field angle is determined.

Danger

Hanging axes require a 100% compensation for weight.
Please contact HEIDENHAIN if this is not the case.

Attention

Limit switches are ignored!
If axes move into an illegal area, press the emergency stop button!

Note

When using incremental encoders with distance-coded reference marks, MP334.x (nominal increment between two fixed reference marks) must be set correctly.

MP2254.x = 2

The PLC initial servicing program, whose name and path is entered in the OEM.SYS file after the **PLCPWM =** entry, must ensure that the inverters are ready after "Switch on external dc voltage," but that the motor brakes are only open while determining the field angle. Alternately, the motor brakes can be opened manually for the duration of the field angle determination.

MP2254.x = 3

Under certain conditions, determination of a field angle with the help of the PLC is not necessary or desired. Here the motor is moved immediately after the **FIELD ORIENT.** key is pressed, and the field angle is determined. This mode can be used if

- there are no brakes,
- in the **Commissioning Current Controller** mode of operation, where the brakes are always open,
- the user ensures that the brakes can be opened manually or with the PLC.

Module 9065 Status of the commissioning function

Module 9065 is used to interrogate status information of commissioning functions dealing with the determination of the field angle, and – as of software version 05 – with the commissioning of the current controller of an axis.

Conditions for the determination of the field angle:

- Synchronous, linear and torque motors determine the field angle each time the control is started if no EnDat or Z1-track encoders are used. For the duration of determining the field-angle (about 5 to 7 seconds), <Mode> 1 returns bit-encoded the axes for which field-angle determination is active.
- Module 9162 reports that the rotary encoder is not ready while the field angle is being determined. A PLC error message can be suppressed if determining is active.

Conditions for commissioning the current controller:

- If no commissioning function is active, the value –1 is returned.

Call:

PS	B/W/D/K	<Mode> Finding the field angle: 0: Axes for which the field angle is being determined with a commissioning aid and an internal oscilloscope 1: Axes for which automatic determination of the field angle is active Commissioning the current controller: 2: Axes for which the commissioning of the current controller is active 3: Interrogation whether the spindle is in delta operation during the commissioning of the current controller (bit#0 =1)
CM	9065	
PL	B/W/D	<Axes bit-encoded>

Error recognition:

Marker	Value	Meaning
M4203	0	Axes have been determined
	1	Error code in W1022
W1022	1	Invalid value for mode

Before determining the field angle (**FIELD ORIENT.** soft key not yet pressed) the inverter must be in the following mode of operation:

- Green “READY” LED on
- Red “SH1” LED off
- Red “SH2” LED on (drive controller not ready, brakes closed)

As soon as the drive enable comes from the PLC, the **Finding field angle** message appears, otherwise **8B40 No drive release** appears.

The motor moves and the field angle is determined. Limit switches are not taken into account.

In order to avoid a possible error message about standstill monitoring, an appropriately large input value is to be provided in MP1110.x (MP1120.x, which is intended for standstill monitoring during determination of the field angle, is available as of NC software 340 490-02).

- Switch the control on.
- Do **not** acknowledge the **Power Interrupted** message. In the **Programming and Editing** mode of operation, use the MOD key to enter the code number **688379.** The oscilloscope is started.
- Press the **I CONTROL** soft key.
- In the **Manual** mode of operation, acknowledge the **Power Interrupted** message.
- Use the **CHOOSE AXIS** soft key in the oscilloscope to select the corresponding axis.
- Press the **FIELD ORIENT.** soft key.

  The PLC must

  - switch the drive on/off
  - release and lock the brakes

The motor moves back and forth. The field angle is determined for the reference mark or datum, and is stored automatically. The **Finding field angle** message appears. Then another message appears (see page 7 – 1104).

- Press the END soft key.

The control carries out a reset. If the message **8630 Field orient. successful** appears, then the field angle was associated and is available.

7.10.4 Reading or setting the field angle via the PLC

For axes without encoders with Z1 tracks, which cannot be moved when switched off (e.g. due to Hirth coupling), or if it has been ensured that they cannot moved when switched off, Module 9149 can be used to read out the field angle after positioning. This axis-specific field angle must be stored via the PLC in nonvolatile memory.

It is also important that the saved field angle is cleared (set to 0) before each positioning, so that the axis (motor) is not started with an incorrect field angle after a power failure. This way an incorrect writing of the field angle is prevented if a power failure occurs.

Setting via Module 9149 of the nonvolatile field angle must be performed before the drives of the affected axis are switched on. This means that it is not necessary to determine the field angle again.

Danger

Please note the following items when setting/reading the field angle via the PLC:

- An incorrectly set field angle can lead to undesirable reactions of the motor, including uncontrollability. It might even move in the wrong direction!
- If the axis position is moved again after the field angle has been read, then the determined field angle may no longer be used.
- The commutation angle may be set only after you have ensured that the stored commutation angle corresponds to the current position (e.g. due to Hirth coupling).
- The module is suitable only for synchronous, torque, or linear motors in conjunction with nonaligned encoders without EnDat interface.
- The module responds with a value only if the reference mark has been traversed.
- The commutation angle for an axis can be set only once after the control is switched on and before the drives are first switched on.

Module 9149 Set/Read commutation angle

Call:

PS	B/W/D/K	<Axis>
PS	B/W/D/K	<Commutation angle in 0.001°> 1 to 720 000
PS	B/W/D/K	<Mode> 0: Read commutation angle 1: Set commutation angle
CM	9149	
PL	B/W/D	<Error code> 0: Commutation angle set/read 1: Module was not called in a spawn job or submit job 2: Invalid mode 3: Invalid axis number 4: Invalid commutation angle Error code from controller when mode 0 is active (read commutation angle): 100: Unknown reference position Error code from controller when mode 1 is active (set commutation angle): 200: Invalid motor type (no synchronous or linear motor) 201: Invalid encoder type (not "non-aligned") 202: Invalid commutation angle 203: Commutation angle already set
PL	B/W/D	<Commutation angle>

Error recognition:

Marker	Value	Meaning
M4203	0	Commutation angle set/read
	1	Error code in W1022
W1022	1	Invalid mode
	2	Invalid axis number
	20	Module was not called in a spawn job or submit job
	45	Error code from controller

7.10.5 Saving the determined field angle

NC software: 340 420-xx, to 340 422-02 and to 340 480-03
The determined field angle is automatically stored on the hard disk.
If the "field orientation" function is not run, the following error message appears:

- Encoder with EnDat interface: **`8830 EnDat: no field angle <axis>`**
- Encoder without Z1 track: **`8820 Field angle unknown <axis>`**

NC software: as of 340 422-03, as of 340 480-03
The determined field angle is automatically entered in MP2556.x.

For purposes of reliability and redundancy, either the serial number of the encoder (only for EnDat interface) or a unique control ID is entered as identification in MP2257.x.

If the current identification does not match the entry in MP2257.x, an error message appears:

- When using an encoder with EnDat interface, the error message **`8830 EnDat: no field angle <axis>`** appears. In any case the field angle must be determined anew, since the encoder does not match the field angle from MP2256.x
- When using an incremental encoder, the error message **`MP2257.<index> incorrect (ID=$<identification>)`** appears. The field angle from MP2256.x and the new identification **`(ID=$<identification>)`** for MP2257.x can only be assumed after determining that the same drive is meant (e.g. after changing controls).

Danger

In all other cases the field angle must be determined anew, since otherwise uncontrolled drive motions could occur!

Note

You can force a new field angle determination be entering MP2256.x = 0 (for example, after exchanging a motor or encoder).

MP2256.x	**Determined field angle**
Input:	0: Field angle does not need to be determined, or has not been determined
MP2257.x	**Control or encoder identification for the field angle from MP2256.x**
Input:	0: Field angle does not need to be determined, or has not been determined

7.10.6 Definition of the field angle

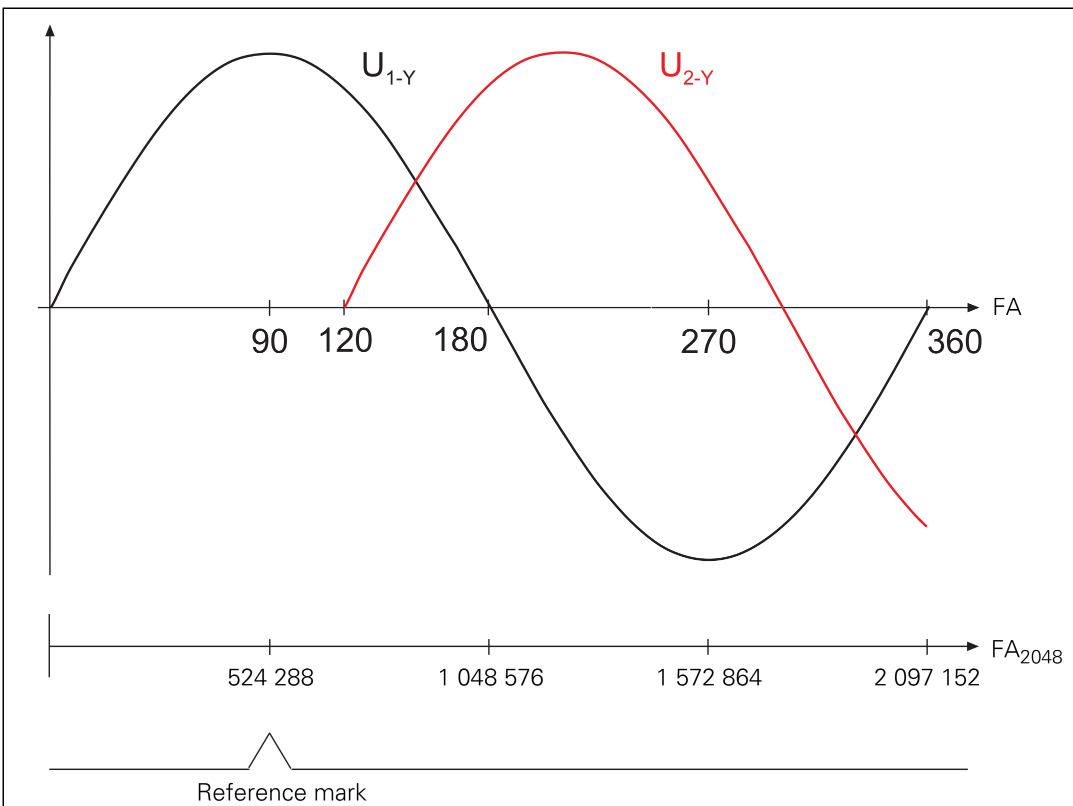
The following applies to the determined field angle, which is entered in MP2556.x:

The field angle is determined for 360° electrical rotation of the motor, not for mechanical rotation.

The motor is moved with external power in the positive direction (when viewing the shaft, the shaft rotates clockwise). The voltages $U_{1\text{-}Y}$ (phase 1 to star point) and $U_{2\text{-}Y}$ (phase 2 to star point) are measured. The positive peak value of $U_{1\text{-}Y}$ corresponds to a field angle of 90°.
The field angle at the reference mark is saved in increments in MP2256.x An increment is formed from <line count> · <interpolation factor>.
Therefore:
MP2256.x = (<field angle at reference mark> · <line count> · 1024) / 360°

In the example below, the reference mark is at the field angle 90°, i.e. MP2256.x = (90° · 2048 · 1024) / 360° = 524288.

On a "standard" synchronous motor (with aligned speed encoder), the reference mark is at the field angle 0°. If the field angle were to be determined for this motor, the result would be MP2256.x = approx. 2097152.



- $U_{1\text{-}Y}$:Motor voltage between phase 1 and star point
- $U_{2\text{-}Y}$:Motor voltage between phase 2 and star point
- FA:Field angle in degrees
- FA_{2048}:Field angle in increments for an encoder with 2048 lines and 1024-fold interpolation (2048 · 1024 = 2 097 152)

7.11 Adjustment of Linear and Torque Motors

7.11.1 General information

Linear and torque motors should be used only in connection with the CC 424(B).
These motors should be connected to controllers with double the computing power (double speed; PWM output X55, X56, X57 to X60).

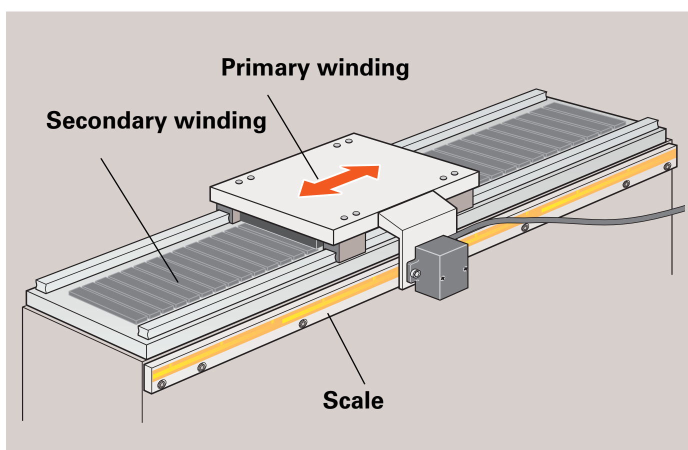
The internal oscilloscope of the iTNC can operate at a maximum resolution of only as fine as 600 µs, but the feedback control can operate at resolutions as fine as 100 µs.
Frequencies higher than 1/(600 µs · 2) = 833 Hz result in undersampling. This can result in misinterpretation of the oscilloscope image. High frequencies are mirrored downward. For example, a 1000-Hz oscillation appears as a 833 Hz - 167 Hz = 666-Hz oscillation.

Note

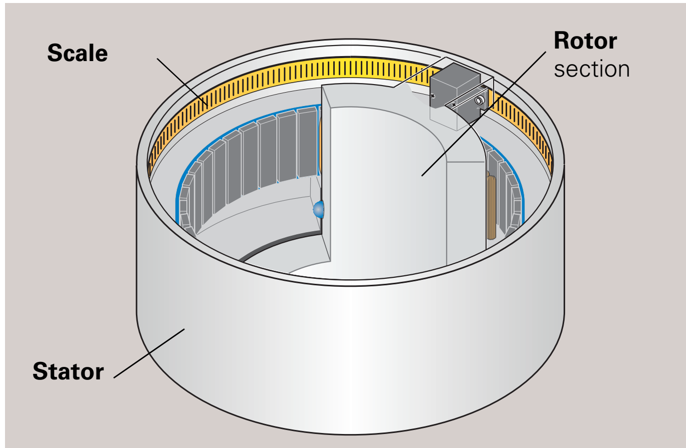
In order to avoid misinterpreting the oscilloscope image, **TNCopt** should be used when searching for oscillation frequencies or optimizing the controller.

For the CC 424(B), TNCopt and the internal oscilloscope display effective values, as opposed to the peak values of the CC 422.

Linear motor setup



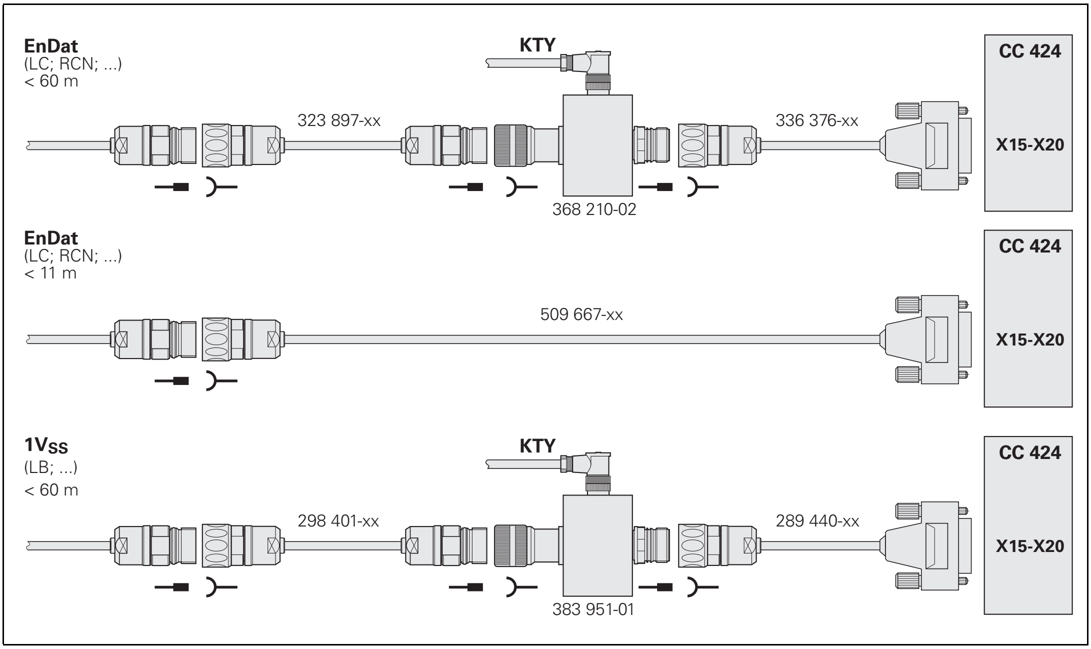
Torque motor setup



Position and speed encoders

For linear and torque motors, position encoders are used as speed encoders. Therefore, they must be connected to the speed encoder inputs (X15 to X20, X80 to X87).

In order to adapt the pin layouts (See "Encoder Connections" on page 375) for absolute encoders with EnDat interface (e.g. LC, RCN), you must use the connecting cable with the ID number 336 376-xx and the line drop compensator with the ID number 368 210-02 (for up to 60 m) or the connecting cable with the ID number 509 667-xx (for up to 11 m), and for incremental encoders with 1-V_{PP} signals (e.g. LB, ROD) you must use the line drop compensator with the ID number 383 951-01.
The temperature sensor (KTY 84) can also be connected to both line drop compensators.



Temperature sensor

Linear motors usually have a KTY and several PTC thermistors or thermoswitches for temperature measurement.

> Attention
>
> The PTC thermistors or thermoswitches must be **galvanically isolated** and evaluated by the PLC.
>
> The KTY requires **double insulation** to the motor windings, which must be provided by the motor manufacturer. Otherwise, do **not** connect the KTY to the control!

The KTY is monitored by the control (NC). The temperature signal is conducted to the control together with the encoder signals (X15 to X20, X80 to X83). If the KTY is not to be evaluated, this function must be deactivated over MP2220.x bit 4 = 1 (See "Monitoring Functions" on page 1084).

For linear and torque motors, the conductor for the temperature signal of the KTY is frequently in the motor power cable, which can cause interference. Since the conductor for the temperature signal is then led into the conductor of the speed encoder, the interference causes noise in the encoder signals. HEIDENHAIN therefore recommends conducting the temperature signals over the line drop compensator, so that the interference signals are filtered.

> Note
>
> HEIDENHAIN recommends the additional temperature monitoring of the PTC thermistors or thermoswitches via the PLC, since these are distributed over the entire length (linear motors) or circumference (torque motors) (as opposed to the KTY, for which there are only spot measurements).

For example, PTC thermistors can be connected to a PLC input via the securely grounded 3RN1013-1BW10 thermistor motor-protection device from SIEMENS.

7.11.2 Safety precautions for linear and torque motors

Note

Comply strictly with the warnings and safety precautions printed in this chapter. They help to prevent damage to material through improper handling!

Danger

Linear and torque motors are equipped with strong magnets and exercise strong magnetic forces of attraction!

This can endanger health directly (e.g. for people with pacemakers), or indirectly (e.g. through fast motor movements and high thrust force).

Danger

- **Please comply with all safety regulations of the motor manufacturer!**
- **Never put your hands in the traverse range of a machine that is switched on.**
- Always switch off the machine before working within the traverse range (machine must be free of potential).
- Ensure free traverse for the axis.
- Before switch-on, check the commutation.
- Monitor the end positions.
- Keep the motor area free of chips.
- Watch out for unusual noises.
- Ensure proper function of the motor coolant system.
- Check at regular intervals the primary and secondary surfaces on the side toward the air gap.
- Check for mechanical stability. The integrated buffer must be able to absorb the energy at Vmax in case of a fault.

Attention

Please comply with the following instructions during all servicing work:

- Mounting, servicing for initial operation, and maintenance are to be performed only by trained personnel.
- Wear work gloves during installation and maintenance work.
- Persons with pacemakers must not service the machine.
- Keep clocks and magnetized data media (e.g. credit cards, floppy disks, etc.) at a distance.
- Do not allow heavy metallic objects to contact the secondary winding of the motor.
- Never allow magnetic surfaces to contact metal.
- Never place the primary winding directly onto the secondary winding.
- Keep a good grip on steel tools and bring them to the secondary winding only slowly and from the side.

7.12 Commissioning Linear and Torque Motors

In this example a linear motor (Siemens 1FN3900-4WB0) and a torque motor (ETEL TMA0360-070-3UC) are adjusted.
If the motors are not yet found in the motor table, enter them in the table using the data sheet values and the conversion rules (See "Determining Entries for Motor Tables" on page 1132), or contact HEIDENHAIN.

On a CC 424(B) the input value 4 can be entered in MP 2206.x for linear motors with one reference mark. In this case the commutation angle must not be determined by using the current controller window. It is determined automatically when the drive is switched on (MP2254.x = 0). When the reference mark is traversed for the first time, it is saved in MP 2256.x.

7.12.1 Machine parameters for linear motors

The following machine parameters can be defined for the 1FN3900-4WB0 linear motor (rotational speed and position measurement through LC 181):

- MP331.x = 0.016 [mm] (with use of an LC 181)
- MP332.x = 1
- MP1350.x = 5 (linear encoder with EnDat interface)
- MP1054.x = 46 [mm/rev]
- MP2100.x = HEIDENHAIN-UM114
- MP2200.x = 1FN3900-4WB0
- MP2202.x = * (distance traveled per electrical period as in the motor table)
- MP2204.x = * (counting direction as in the motor table; if the rotating field does not match the counting direction, enter a minus sign)
- MP2206.x = * (encoder as in the motor table)
- MP2220.x bit 4 = 1 (no monitoring for excessive temperature, since the temperature sensor only has a single insulation layer)

If speed and position are measured by an LB 3xx C, the following machine parameters change:

- MP331.x = 0.004 [mm] (with use of an LB 3xx C)
- MP332.x = 1
- MP334.x = 2000
- MP1350.x = 4 (linear encoder with distance-coded reference marks)
- MP2204.x = * (counting direction as in the motor table; if the rotating field does not match the counting direction, enter a minus sign)
- MP2206.x = 8 (incremental linear encoder with distance-coded reference marks (not aligned))

7.12.2 Machine parameters for torque motors

The following machine parameters can be defined for the ETEL TMA0360-070-3UC torque motor (rotational speed and position measurement through RCN 723):

- MP331.x = 360 [°]
- MP331.x = 32768 (with use of an RCN 723)
- MP1350.x = 5 (linear encoder with EnDat interface)
- MP1054.x = 360
- MP2100.x = HEIDENHAIN-UM114
- MP2200.x = ETEL TMA0360-070-3UC
- MP2202.x = * (line count as in the motor table)
- MP2204.x = * (counting direction as in the motor table; if the rotating field does not match the counting direction, enter a minus sign)
- MP2206.x = * (encoder as in the motor table)

7.12.3 Adjustment of the current controller

- Enter as many machine parameters as possible
- Assign empirical values to machine parameters that must still be determined (See "Preparation" on page 1020).

Danger

During current controller adjustment of linear and torque motors, the rotor position of the motor is not yet known.
For this reason, if the motor brakes are not active, the motor might move slightly when the current pulses switch on. In other words, it might oscillate about a preferred position. It is also possible to position "manually" to the preferred position. Do not do this, however, during a measurement.

Use the integrated oscilloscope or TNCopt to adjust the current controller.

During adjustment of the current controller the speed controller and position controller are open. During output of the current pulses the drive controller becomes active.

The PLC initial servicing program, whose name and path is entered in OEM.SYS after the **PLCPWM =** entry, must ensure that the inverters are ready after "Switch on external dc voltage," but that the motor brakes are not opened.

Before and after the output of the current pulses (**START STEP** soft key is not pressed) the inverter must be in the following operating mode:

- Green "READY" LED on
- Red "SH1" LED off
- Red "SH2" LED on (drive controller not ready)

During output of the current pulses:

- Green "READY" LED on
- Red "SH1" LED off
- Red "SH2" LED off (drive controller ready)

During current adjustment, proceed as follows:

- Switch the control on.
- Do not acknowledge the **Power Interrupted** message. In the **Programming and Editing** mode of operation, use the MOD key to enter the code number **688379.** The oscilloscope is started.
- Press the **I CONTROL** soft key.
- In the **Manual** mode of operation, acknowledge the **Power Interrupted** message and switch the control voltage on.
- Use the **CHOOSE AXIS** soft key in the oscilloscope to select the axis to be adjusted.
- With the **P/I FACTOR** soft key, select the I factor and set MP2430.x = 0.
- With the **P/I FACTOR** soft key, select the P factor.
- As a starting value for the P factor first enter the value 0, select the step increment **SLOW,** and press the ↑ soft key about 5 to 10 times.
- Press the **START STEP** soft key.
  This sends multiple step functions to the current controller and measures the step responses. The height and length of the steps are automatically calculated by the control.

Applying a voltage causes a brief humming noise.

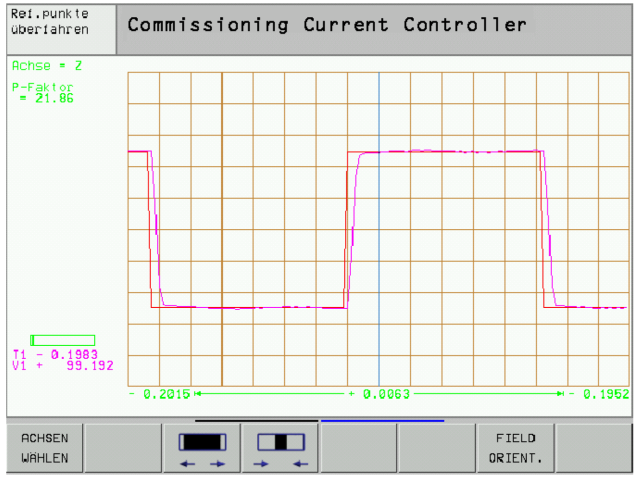
> Danger
>
> If the brakes are not active, the motor can move somewhat (in the direction of a preferred position)!
> The error message **Standstill monitoring <axis>** might be displayed. It can be deleted.

- With the ↑ soft key, increase/decrease the P factor (MP2420.x) step by step just enough so that no undershoot is visible
- Save this value with the **STORE MP2420.x** soft key.
- With the **P/I FACTOR** soft key, select the I factor.
- As a starting value for the I factor first enter the value 0, select the step increment **SLOW,** and press the ↑ soft key about 5 to 10 times.
- With the ↑ soft key, increase/decrease the I factor (MP2430.x) step by step just enough so that no undershoot is visible
- Save this value with the **STORE MP2430.x** soft key.
- Press the END key to exit the **Commission Current Controller** mode.

The control carries out a reset.

> Note
>
> For **linear and torque motors,** adjust the P and I component of the current controller so that no overshoots are visible in the step response. This reduces motor noise.

Properly adjusted step response:



7.12.4 Adjustment of the speed controller

The speed controller should be adjusted to largely suppress oscillation frequencies but to permit short rise times.

The following table shows approximate input values for commissioning the speed controller so that axes can be moved:

Machine parameters	Value
MP1510.x	1 to 5; k_V factor for velocity feedforward control
MP2500.x	1 to 50 for linear motors (starting value 1); 50 to 5000 for torque motors (starting value 50)
MP2510.x	10 to 2000 for linear motors (starting value 10); 1000 to 800 000 for torque motors (starting value 1000)
Increase positioning window and position monitoring	See "Commissioning" on page 998
MP2562.x to MP2566.x	0 (deactivate all filters)
MP2602.x, MP2604.x	0 (without IPC)

The current controller must have been adjusted and the field angle ascertained (See "Adjustment of the current controller" on page 1123 and See "Field Orientation" on page 1101).

Preparation on the control

- Position the axis or spindle to be optimized at a location where it can be commissioned safely.
- Ensure that the loaded PLC program fulfills the following conditions:

- Enable the drive controller
- NC stop inactive
- Axis direction buttons active

As opposed to the CC 422, the internal oscilloscope of the control no longer needs to be started when adjusting the speed controller of the CC 424(B) with TNCopt.

Setting the filters in the speed controller

- To determine the oscillation frequencies of the speed controller, activate the **Optimization/speed controller step response** function in TNCopt.

The oscillation frequencies of the speed controller for linear and torque motors must be determined at slow feed rate in positive and negative traverse direction, and at the midpoint and end of the traverse range.
Later you enter the ascertained frequencies, bandwidths and damping in the corresponding machine parameters (MP2542.x to MP2566.x). Normally no more than three filters are required.

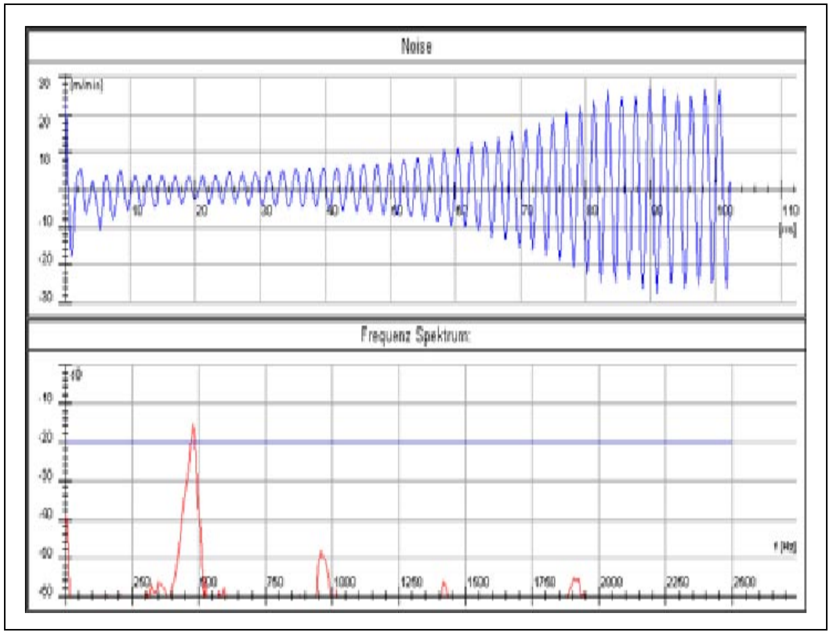
Experience recommends:

- At very small frequencies (< 200 Hz) use no filters
- Use very little damping (3 dB to 6 dB) at low frequencies (200 Hz to 400 Hz)
- The bandwidth should be approximately at center frequency
- Damping values are normally between 3 dB and 18 dB. Damping above 18 dB usually brings no further improvement.
- The use of a PT2 filter on linear motors does not bring positive results. For torque motors it is sometimes necessary to activate a PT2 filter (not below approx. 400 Hz).

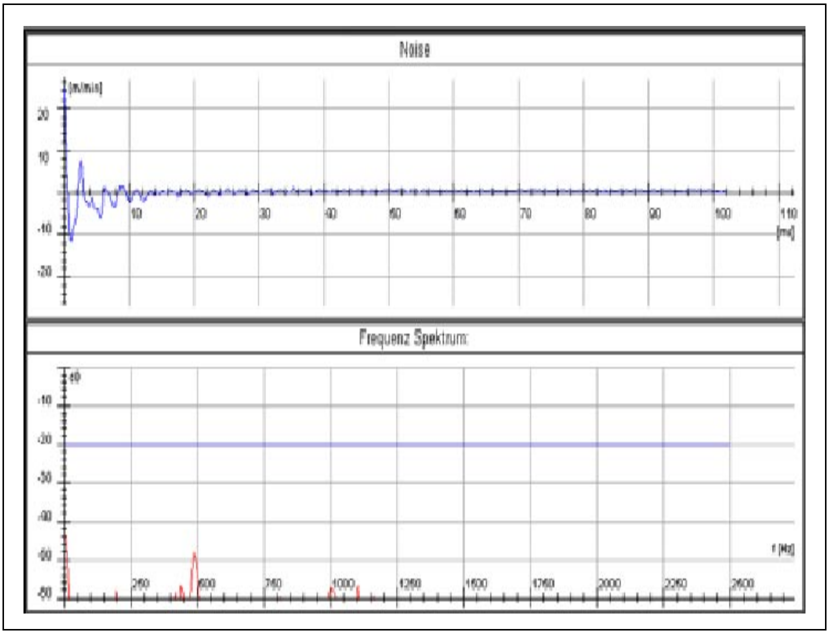
Note

Only use the filters in the speed controller if they are necessary, i.e. if the P factor can be increased noticeably.

- For P and I factors, set low values at first:
  (for linear motor: P factor = 1, I factor = 10;
  for torque motor: P factor = 50, I factor = 100).
- Increase the P factor up to the oscillation limit in 10% steps
  (see the TNCopt User's Manual).



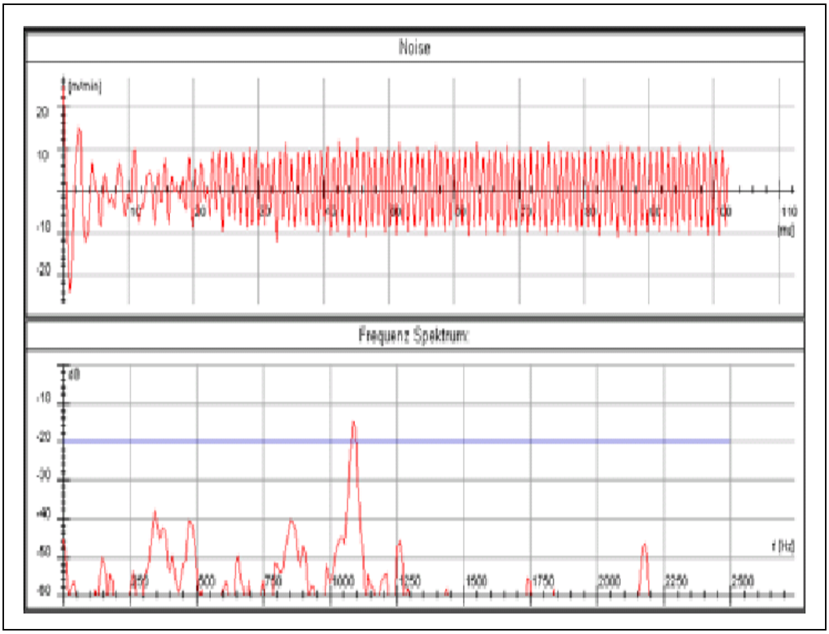
- Find the characteristic values of the first filter (F1):
  Enter the oscillation frequency (e.g. at 479 Hz),
  Enter the bandwidth (e.g. 479 Hz, center frequency),
  Enter the damping (3 dB).
- Examine the step response.
- Check whether a constant P factor decreases the excessive value.
- Increase the damping in 3-dB steps (damping greater than approx. 18 dB normally has no benefit).
- Examine the step response. The oscillations should be greatly decreased.



▶ Increase the P factor further.

This may cause new oscillations.
When selecting filters, watch whether your selection causes or aggravates negative characteristics. If it does, do not activate any further filters.

Use filters only if the P factor can be significantly increased!



After finding the oscillation frequencies, several filters can be activated.

- On the control, enter the determined filter type (MP2562.x to MP2566.x), the determined center frequency (MP2552.x to MP2556.x), the determined bandwidth (MP2572.x to MP2576.x) and the determined damping (MP2542.x to MP2546.x).

Adjusting synchronous and asynchronous motors

Synchronous and asynchronous motors are adjusted as described in the TNCopt User's Manual.

If a sufficiently short enough rise time is not achieved (approx. 3 ms), use a filter (See "Setting the filters in the speed controller" on page 1127). Using a filter also results in an increased P factor. The I factor must then be redetermined.

In most cases automatic adjustment of synchronous and asynchronous motors determine rise times that are too short (< 2 ms). Rise times that are too short result in irregular control responses. In this case both the P and I factors must be reduced (until a rise time of approx. 2 ms is achieved).

Adjusting linear and torque motors

- First determine the oscillation frequencies and adjust the filters (See "Setting the filters in the speed controller" on page 1127).
- To adjust linear and torque motors, activate the **Optimization/Linear motor adjustment** function in TNCopt.

This makes it possible to automatically find the controller parameters in the sequence P factor, k_V factor, I factor (also see the TNCopt User's Manual). An NC program must be started in which the motor is moved back and forth.

Because inhomogeneity of magnets, changing the air gap, and other factors can cause linear motors to oscillate more strongly at certain positions, HEIDENHAIN recommends adjusting the motor over a large range of traverse (approx. 300 mm to 500 mm).

During adjustment, TNCopt automatically increases the newly adjusted controller factor after each reversal point until it detects an oscillation limit. The respective current value is displayed in TNCopt. Usually you can already hear the oscillation limit being reached. In this case the oscillation limit is set per command button (see the TNCopt User's Manual). The factors are reduced after detection of the oscillation limit. Then they are considered to be optimized values.

Particularly the optimization of the I factor often results in very low frequency oscillations that the controller does not recognize as oscillations. Here the oscillation limit must be set by command button.

- Move the axis back and forth.

The P factor, k_V factor or I factor is automatically increased by 10% of the current value.

- As soon as you hear an oscillation in the axis (also visible on the iTNC's oscilloscope with **`I noml`**), set the oscillation limit via command button (see the TNCopt User's Manual).

The determined P factor, k_V factor or I factor is reduced and assumed. TNCopt automatically switches to finding the next factor.

- Also repeat the finding of the k_V factor and I factor.

The P, I and k_V factors were found.

- Enter the determined values at the control in MP2500.x, MP2510.x and MP1510.x.

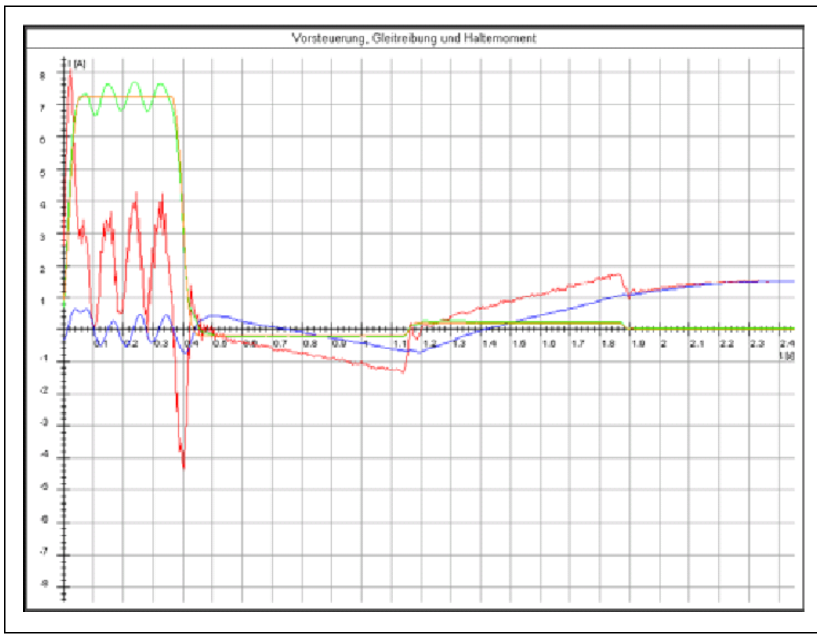
Finding the feedforward values

Automatic adjustment of feedforward values in TNCopt often fails for linear motors. The reason is the often extreme ripple of the nominal current due to the pronounced torque ripple. But even if the automatic adjust does not work, TNCopt can still be used for manual adjustment.

The proposed curves are sometimes not the optimum for linear motors. It can therefore be better for manual adjustment if you increase the traverse paths and the feed rate (by approximately doubling it, for example) and adjust the acceleration in MP1060 to about 1 m/s^2 to 2 m/s^2.

If the integral current should approach the nominal current only very slowly, reduce the k_V factor and increase the I factor. However, these changes should be undone after determining the feedforward values.

The image below shows an adjustment of a highly critical linear motor with a distinctly inhomogeneous magnetic field. This can be seen from the nominal current, which fluctuates very strongly depending on the position:



7.13 Determining Entries for Motor Tables

> **Note**
>
> If you use a linear or torque motor that is not listed in the current HEIDENHAIN motor table, please contact HEIDENHAIN. As an alternative, you can also determine the input values yourself.

7.13.1 Determining data for linear motors

The motor data for linear motors are entered in the motor table after some conversions using the values from the motor data sheet of the respective manufacturer (here using the example of a Siemens motor).

Values in the HEIDENHAIN motor table	Values from the motor data sheet
TYPE: LSM	Permanently excited linear synchronous motor
NAME: 1FN3900-4WB0	1FN3900-4WB0
MODE: 0	
Rated current I-N in [A_{eff}] winding I-N: 49.4	Data sheet value IN IN = 49.4 A
Rated voltage U-N in [V_{eff}] interlinked U-N: 394	Calculation from data sheet values FN, IN, RP.120, LP $\text{U-N} = \sqrt{3} \cdot \sqrt{(U_e + U_r)^2 + U_x^2}$ U_e = N-N / (1000mm/m) · FN / (IN ·3) = (2.66 m/s) · 8100 N / (49.4 A ·3) = 145.4 $V_{eff\ L\text{-}N}$ U_r = RP.120 · IN = 0.8 Ω · 49.4 A = 39.5 $V_{eff\ L\text{-}N}$ U_x = 2 · π · F-N · LP · IN = 2 · π · 57 Hz · 0.0075 H ·49.4 A = 132.69 $V_{eff\ L\text{-}N}$ U-N = 394 $V_{eff\ (L\text{-}L)}$
Rated speed N-N in [mm/s] N-N: 2666	Calculation from data sheet value $v_{MAX,FN}$ N-N = (1000 mm/m) · $V_{MAX,FN}$ / (60 s/min) = (1000 mm/m) · (160 m/min) / (60 s/min) = 2666 mm/s Note: Other meaning than for rotating motors!
Rated frequency F-N in [Hz] F-N: 57	Calculation from V-N, distance per electrical period (see entry in STR column) F-N = N-N · (1000 µm/mm) / (distance per elec. period [µm]) F-N = (2666 mm/s) · (1000 µm/mm) / (46 000 µm) = 57 Hz

Values in the HEIDENHAIN motor table	Values from the motor data sheet
No-load voltage at rated velocity U0 in [V_{eff}] interlinked U0: 252	Calculation from V-N and data sheet value $k_{F,3}$ U0 = N-N / (1000 mm/m) · ($k_{F,3}$ / 3) · √3 = 2.66 m/s · (164 N/A / 3) · √3 = 252 V_{eff} L,L
No-load current I0 in [A_{eff}] winding I0: 0	
Primary winding resistance at 20 °C R1 in [mΩ] at 20° C R1: 600	Calculation from data sheet value $R_{P,20}$ R1 = $R_{P,20}$ · 1000 = 0.6 Ω ·1000 = 600 mΩ
Rotor resistance at 20 °C R2 in [mΩ] at 20° C R2: 0	
Primary winding leakage reactance at F-N Xstr1 in [mΩ] Xstr1: 0	If nothing given, then zero.
Rotor leakage reactance at F-N Xstr2 in [mΩ] Xstr2: 0	
Magnetizing reactance XH for F-N at rated conditions XH in [mΩ] XH: 2685	Calculation from F-N and data sheet value L_P XH = 2 · π · F-N · L_P = 2 · π · 57 Hz · 7.5 mH = 2685 mΩ
Desaturation velocity N-XH in [min^{-1}] N-XH: 0	
Rotational speed of beginning field weakening range N-FS [min^{-1}] N-FS: 0	
Max. velocity (mechanical) N-MAX in [mm/s] N-MAX: 2666	Calculation from data sheet value $v_{MAX,FN}$ N-MAX= 160 m/min · (1000 mm/m) / (60 s/min) = 2666 mm/s Note: Other meaning than for rotating motors!
Saturation factor %-XH in % %XH: 100	
Stalling torque reduction factor %-K in % %-K: 100	
No. of pole pairs (half pole no. of motor) PZ PZ: 1	
Temperature coefficient of the primary winding TK in 1/K TK: 0.004	

Values in the HEIDENHAIN motor table	Values from the motor data sheet
Distance per electrical period STR STR: 46 000	Calculation from data sheet value τ_M Path $= \tau_M \cdot 2 \cdot (1000\ \mu m/mm)$ $= 23\ mm \cdot 2 \cdot (1000\ \mu m/mm)$ $= 46\ 000\ \mu m$ Note: Other meaning than for rotating motors!
Type of encoder SYS: 3	Incremental encoder (e.g. LB): 8 Absolute encoder with EnDat interface (e.g. LC): 3
Counting direction DIRECT. DIRECT.: +	
Max. temperature of motor at temperature sensor T-MAX in [°C] T-MAX: 120	Data sheet value $T_{P,\ MAX}$ $T_{P,\ MAX}$ = 120 °C
Maximum motor current I-MAX in [A_{eff}] winding I-MAX: 138.9	Data sheet value I_{MAX} I_{MAX} = 138.9 A_{eff}
Rated power P-N in [W] P-N: 21600	Calculation from data sheet values $v_{MAX,FN}$, F_N P-N = $v_{MAX,FN} \cdot (1\ min/60\ s) \cdot F_N$ P-N = 160 m/min · (1 min/60 s) · 8100 N P-N = 21600 W
Mass of primary winding J in [kg] J: 56.2	Data sheet value m_P m_P = 56.2 kg Note: Other meaning than for rotating motors!
Inductance of the series reactor L in [mH] L: 0 µH As long as (X1str+Xh) / (2 · π · (N-N / 60) · PZ) is greater than 700 µH, no series reactor is required.	
Thermal time constant for direct current T-DC in [s] T-DC: 0	
Lower thermal cutoff frequency F-DC in [Hz] F-DC: 0	
Thermal time constant for alternating current T-AC in [s] T-AC: 0	
Upper thermal cutoff frequency F-AC in [Hz] F-AC: 0	

7.13.2 Determining data for torque motors

The motor data for torque motors are entered in the motor table after some conversions using the values from the motor data sheet of the respective manufacturer (here using the example of an ETEL motor).

Values in the HEIDENHAIN motor table	Values from the motor data sheet
TYPE: SM	Permanently excited synchronous motor
NAME: Etel-TMA0360-070-3UC	TMA0360-070-3UC
MODE: 0	
Rated current I-N in [A_{eff}] winding I-N: 32.6	Data sheet value IN IN = 49.4 A
Rated voltage U-N in [V_{eff}] interlinked U-N: 252	Calculation from data sheet values FN, IN, RP.120, LP $\text{U-N} = \sqrt{3} \cdot \sqrt{(U_e + U_r)^2 + U_x^2}$ Ue = 2 · π · (N-N / 60) · Tcw105 / Icw105 / 3 Ue = 2 · π · (180 / 60) · 485 / 32.6 / 3 Ue = 93.48 $V_{eff\ L,N}$ Ur = (R105L;L / 2) · Icw105 Ur = (1.92 / 2) · 32.6 Ur = 31.30 $V_{eff\ L,N}$ Ux = 2 · π · (n / 60) · (2p / 2) · (L1L,L / 2) · Icw105 Ux = 2 · π · (180 / 60) · (66 / 2) · (0.00738 / 2) · 32.6 Ux = 74.83 $V_{eff\ L,N}$ U-N = 252.0 $V_{eff\ L,L}$
Rated speed N-N in [min^{-1}] N-N: 180	Data sheet value *n* N-N = 180 min^{-1}
Rated frequency F-N in [Hz] F-N: 99	Calculation from data sheet value *n* in 2p F-N = (n / 60) · (2p / 2) F-N = (180 / 60) · (66 / 2) = 99 Hz
No-load voltage at rated speed U0 in [V_{eff}] interlinked U0: 252	Calculation from data sheet value *n* in Kt U0 = 2 · π · (n / 60) · (Kt / 3) · √3 U0 = 2 · π · (180 / 60) · (15.7 / 3) · √3 U0 = 170.9 $V_{eff\ L,L}$
No-load current I0 in [A_{eff}] winding I0: 0	
Stator resistance at 20 °C R1 in [mΩ] at 20° C R1: 670	Calculation from data sheet value $R20_{L-L}$ R1 = $R20_{L-L}$ / 2 = 1.34 Ω / 2 = 670 mΩ
Rotor resistance at 20 °C R2 in [mΩ] at 20° C R2: 0	

Values in the HEIDENHAIN motor table	Values from the motor data sheet
Stator leakage reactance at F-N Xstr1 in [mΩ] Xstr1: 0	If nothing given, then zero.
Rotor leakage reactance at F-N Xstr2 in [mΩ] Xstr2: 0	
Magnetizing reactance XH for F-N at rated conditions XH in [mΩ] XH: 2295	Calculation from data sheet value $L1_{L-L}$, *n* and 2p XH $= 2 \cdot \pi \cdot (n / 60) \cdot (2p / 2) \cdot (L1_{L,L} / 2)$ $= 2 \cdot \pi \cdot (180 / 60) \cdot (66 / 2) \cdot (0.00738 / 2)$ $= 2295$ mΩ
Desaturation speed N-XH in [min^{-1}] N-XH: 0	
Rotational speed of beginning field weakening range N-FS [min^{-1}] N-FS: 0	
Maximum speed (mechanical) N-MAX in [min^{-1}] N-MAX: 180	Data sheet value *n* N-MAX = 180 min^{-1}
Saturation factor %-XH in % %XH: 100	
Stalling torque reduction factor %-K in % %-K: 100	
No. of pole pairs (half pole no. of motor) PZ PZ: 33	From data sheet value 2p PZ = 2p/2 PZ = 66/2 = 33
Temperature coefficient of the stator winding TK in 1/K TK: 0.004	
Line count of the speed encoder STR STR: 32768	
Type of encoder SYS: 5	Incremental encoder with Z1 track: 1 Aligned absolute encoder with EnDat interface: 2 Unaligned absolute encoder with EnDat interface: 5 Unaligned incremental encoder with distance-coded reference marks: 7
Counting direction DIRECT. DIRECT.: +	
Max. temperature of motor at temperature sensor T-MAX in [°C] T-MAX: 120	
Maximum motor current I-MAX in [A_{eff}] winding I-MAX: 53	Data sheet value Ip I-MAX = 53.0 A_{eff}

Values in the HEIDENHAIN motor table	Values from the motor data sheet
Rated power P-N in [W] P-N: 9142	Calculation from *n* and Tcw105 P-N = 2 · π · (n / 60) · Tcw105 P-N = 2 · π · (180 / 60) · 485 Nm P-N = 9142 W
Motor mass moment of inertia J in [kgm^2] J: 0.157	Data sheet value J J = 0.157 kgm^2
Inductance of the series reactor L in [mH] L: 0 µH As long as (X1str+Xh) / (2 · π · (N-N / 60) · PZ) is greater than 700 µH, no series reactor is required.	
Thermal time constant for direct current T-DC in [s] T-DC: 0	
Lower thermal cutoff frequency F-DC in [Hz] F-DC: 0	
Thermal time constant for alternating current T-AC in [s] T-AC: 0	
Upper thermal cutoff frequency F-AC in [Hz] F-AC: 0	

With MP2209.x you can overwrite the value for "Mass moment of inertia J" in the motor table. This makes it possible to react to additional, rigidly coupled inertias.

MP2209.x **Mass moment of inertia of a drive motor**
Input: * = Entry from the motor table active
Value of the mass moment of inertia in [kgm^2]

8 Machine Interfacing

8.1 Display and Operation

You can modify the display and operating modes of the iTNC by editing the machine parameters.

The display screen is divided into separate windows. The user can select the operating functions by soft key. Refer to the User's Manual.

8.1.1 Position and status display

The status display shows the status of the control.

With a soft key you can activate an additional status display in the graphic window instead of the graphic. The status display is shown in forms that are divided into tabs. You switch between the tabs via soft keys. The STATUS PGM, STATUS POSITIONS, STATUS TOOL and STATUS COORDINATE TRANSFORMATIONS tabs can be selected directly, and the other status displays are selected with soft keys.

This information includes:

- Axis positions
- Tools
- Feed rate
- M functions
- Active spindle (S1 or S2)
- Current time
- The most important status information is collected on an overview page.
- In the PGM status a moving-bar graphic shows the current program run time as a percentage. In addition, the actual and programmed feed rate are shown as numerical values.
- The active machining cycle and the active parameters programmed in Cycle 32 Tolerance are shown in the CYC status.
- The active basic rotation is also shown in the TRANS status.
- Only the status information important for test run is shown in the Test Run operating mode.
- If the **Global Program Settings** software option is available, two additional tabs containing information about the activated functions are shown.
- If the **Adaptive Feed Rate Control** software option is available, the additional AFC tab containing information about the feed-rate control is shown.

Position display step

To define the position display step for axis and spindle positions:

- Enter the desired display step for the axes in MP7290.x and for the spindle in MP7289.

The position loop resolution is not influenced by this parameter.

MP7290.x **Position display step for the axes**
Input: 0: 0.1 mm or 0.1°
1: 0.05 mm or 0.05°
2: 0.01 mm or 0.01°
3: 0.005 mm or 0.005°
4: 0.001 mm or 0.001°
5: 0.0005 mm or 0.0005°
6: 0.0001 mm or 0.0001°

MP7289 **Position display step for the spindle**
Input: 0: 0.1°
1: 0.05°
2: 0.01°
3: 0.005°
4: 0.001°
5: 0.0005°
6: 0.0001°

Position display of the tool axis

The tool length can be offset in the position display of the tool axis. If it is, the displayed position value then refers to the tool point:

- With MP7285, select whether the tool length should be offset.

The behavior of an incremental block after a **TOOL CALL** can be specified:

- With MP7682 bit 0, select whether the tool length should be offset.

MP7285 **Tool length offset in the tool-axis position display**
Input: 0: Tool length is not offset
1: Tool length is offset

MP7682 **Machine parameter with multiple function**
Input: Bit 0 – Incremental block after TOOL CALL
0: With length compensation
1: Without length compensation

Position display for rotary axes and PLC auxiliary axes

For these axes you can set the modulo value for the counting mode (i.e. the value after which the axis display returns to zero). The software limit switches become inactive when the modulo counting mode is activated.

However, for the modulo counting mode you can activate software limit switches for a traverse-range limit (e.g. 10 to 180 in MP91x.x or 92x.x) via MP812. This traverse-range limit cannot be restricted additionally in the manual operating mode.

When using encoders with distance-coded reference marks (MP1350), you can define a traverse range of more than 360° for rotary axes with limit switches (e.g. ± 190 degrees). When the machine is switched off, the position is stored by the control. This enables the control to ascertain whether the axis is, for example, at a position of –1° or +359° after the machine is switched back on.

Note

Please note that the traverse-range limits set earlier in the manual operating mode also become active in the modulo counting mode when MP812.x is set. However, these can no longer be edited or viewed in the manual operating mode.

▶ Select the display mode with MP810.x.

▶ Activate or deactivate the software limit switches with MP812.

Rotary axes with modulo display can be positioned either without crossing zero or always along the shortest path:

▶ Select the type of positioning with MP7682.

- For bit 2=0: Programming with M126
- For bit 2=1: You need not program with M126.

Furthermore, based on the settings in MP810.x, the control detects whether the PLC axis is a rotary axis:

- MP810.x ≠ 0: PLC axis is a rotary axis

MP810.x	**Display mode for rotary axes and PLC auxiliary axes**
Input:	0.0000 to 99 999.9999 [°] 0: Display +/–99 999.9999 ≠ 0: Modulo value for display
MP812	**Activate software limit switches for tilting axes with modulo display, M94 and encoders with EnDat interface**
Input:	%xxxxxxxxxxxxxx Bits 0 to 13 represent axes 1 to 14 0: Software limit switch not active 1: Software limit switch active
MP7682	**Machine parameter with multiple function**
Input:	Bit 2 – Traverse path of rotary axes with modulo display 0: Positioning without passing over zero 1: Positioning on the shortest path

Reading of axis coordinates

▶ Read the axis coordinates with Module 9040 or Module 9041.

The values are saved in double words beginning at the given address. Enough double words must be reserved. The number of required addresses is according to the **AXISNUMBER =** entry in OEM.SYS. If this entry does not exist, the number is according to the enabled control loops.

To define the coordinate value of an axis, the reference point of the axis must first be traversed.

Module 9040 Reading of axis coordinates (format 0.001 mm)
Module 9040 loads the axis coordinates from the control loop for all NC axes. Different values can be loaded: Actual value, nominal value, servo lag, distance-to-go, the deflection of a triggering touch probe, temperature compensation values, and as of software 340 49x-05 even the distance counter (counts the distance traveled by an axis).

The datum system equals the reference system shifted by a datum compensation. All values, except for the distance counter, are saved in the format 1/1000 mm in double words beginning at the programmed target address. The number of double words that are written to varies depending on the configuration.

Conditions:

SEE MODULE 9041.

Call:
SEE MODULE 9041.

Module 9041 Reading of axis coordinates (format 0.0001 mm)

Module 9041 loads the axis coordinates from the control loop for all NC axes. Different values can be loaded: Actual value, nominal value, servo lag, distance-to-go, the deflection of a triggering touch probe, temperature compensation values, and as of software 340 49x-05 even the distance counter (counts the distance traveled by an axis).

The datum system equals the reference system shifted by a datum compensation. All values, except for the distance counter, are saved in the format 1/10000 mm in double words beginning at the programmed target address. The number of double words that are written to varies depending on the configuration.

Condition:

- Regardless of whether individual axes have been excluded through MP10, the coordinate values are always loaded for all axes. The values for excluded axes remain undefined.
- The coordinate value of an axis remains undefined until the reference point has been traversed.
- In a tilted working plane, the axis coordinates are referenced to the non-tilted coordinate system.
- The tool length is ignored for the actual values in the datum system.
- The coordinate type "Actual values in the datum system" must only be read when the working plane is NOT tilted.
- Distance counter in [mm] (as of 340 49x-05):
  Updating in minute cycle if a change of at least 10 mm has occurred. The maximum value of a counter corresponds to the value from MP4050.x and is cleared with W1058 (resetting the accumulated distance).

Call:

```
PS      K/B/W/D   <Target address Dxxxx>
PS      K/B/W/D   <Type of coordinate>
                  0: Actual values
                  1: Nominal values
                  2: Actual values in the reference system
                  3: Following error
                  4: Distance-to-go
                  5: Deflection (measuring touch probe)
                  6: Actual values in the shifted reference system (datum
                     shift)
                  7: Reference values with backlash compensation offset
                     from MP710.x
                  8: Temperature compensation from the description tables
                     of the tilting-axis geometry
CM      9040 OR
CM      9041
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Invalid coordinate type, target address too large, or given target address is not a double-word address

Free rotation

Free rotation means that the rotary axis rotates as often as required (with a display range of 0 to 360°) without being affected by software limit switches. You can define the free rotation function through words (axis 4 and 5) or with Module 9223.

No uncontrolled axes can be used for this function, but PLC axes (auxiliary axes) are possible along with main axes.

The maximum feed rate is 300 000 °/min or the limiting value in MP1010.x. The feed rate is not shown in the status window. You can change the feed rate with the override potentiometer (W754), for example by copying W494 (active feed rate override) to W754. MP7620 bit 3 is accounted for.

Use Module 9048 to inquire whether free rotation is active for a specific axis.

The following conditions apply to the free rotation function:

- Collision monitoring (DCM) does not work in combination with the "free rotation" PLC function, since this special function for the defined "free" rotary axis runs asynchronously to the system. For the work envelope this means that all CMOs moved with the "free" rotary axis cannot be monitored by DCM.
- All monitoring functions, such as for position, following error, signal amplitude of the encoder, etc., remain active.
- All settings for the rotary axes via machine parameters, such as jerk, acceleration, etc., remain valid.
- The axis that is in free rotation behaves asynchronously to (independently of) all other axes. For example, an NC program can be run while the "free rotation" axis rotates independently. This axis is not considered when calculating the path.
- The axis for free rotation must be a rotary axis that can rotate over 360° without any restrictions.
- As of software version 340 49x-02 the free rotation function can be used not only for principal axes (NC axes), but also for auxiliary axes (PLC axes).

Free rotation defined by words

- W566 Feed rate in axis 4 for free rotation
- W567 Feed rate in axis 5 for free rotation
- W754 Feed rate override percentage for free rotation
- B518 Defining the free rotation function
- B519 Traverse direction for free rotation
- M4133 Starting and stopping the free rotation function
  If you set M4133, the NC takes the information from B518 and B519 and resets M4133.

Free rotation with Module 9223

If a program has been started, the module may be called only in conjunction with an M/S/T/Q strobe. Auxiliary axes (e.g. rotary axes that are configured as auxiliary axes) are an exception; for them the PLC module can also be used without active strobe signal during a running NC program.

Module 9223 Free rotation

When the module is called, M4133 is set (start and stop).
The feed-rate override in W754 remains in effect.

Call:

PS	B/W/D/K	<Axis> 0 ... max
PS	B/W/D/K	<Feed rate [°/min]>
PS	B/W/D/K	<Mode> 0: Stop +1: Start in positive direction –1: Start in negative direction
CM	9223	
PL	B/W/D	<Error code> 0: No error: Positioning is started/stopped 1: No rotary axis transferred 2: Impermissible feed rate 3: Axis has not traversed the reference mark 4: No M/S/T/Q strobe during running program 5: Programmed axis not in closed loop 6: Other traversing command active for axis at start of function 7: Free rotation not active for axis at stop of function

		Set	Reset
M4133	**Start and stop the free rotation function**	PLC	NC
B518	**Definition of the free rotation function** 0: Cancel the function 8: Free rotation for axis 4 16: Free rotation for axis 5	PLC	PLC
B519	**Traverse direction for free rotation** 0: Axis 4 and axis 5 = + 8: Axis 4 = – , axis 5 = + 16: Axis 4 = +, axis 5 = – 24: Axis 4 and axis 5 = –	PLC	PLC
W754	**% function for feed-rate override for free rotation**	PLC	PLC
W566 - 568	**Feed rate for free rotation** Axis 4 to axis 5	PLC	PLC

Feed rate display

The programmed contour feed rate is displayed in the **Program run, single block** and **Program run, full sequence** operating modes. With the feed rate potentiometer you can change the feed rate from 0 to 150%.

If rapid traverse was programmed, **FMAX** is displayed and M4180 is set.

The percentage adjusted with the feed rate override is entered by the NC in W494 and W766.

You can change the percentage through the PLC:

▶ Enter the desired percentage in W766. The NC immediately takes over the new value

The feed-rate override is effective either in 1% steps or according to a nonlinear characteristic curve:

▶ With MP7620 bit 3, select the mode of the override.

Value range in W494 and W766:

- 1% steps: 1 to 150
- Nonlinear characteristic curve: 0 to 15 000

In the lowest range, 0.01% steps are available. Beginning with a value of 2.5%, the step is 0.75%

Note

If bit 3 of MP7620 is set, then, due to the nonlinear characteristic curve, the labeling of the override only matches the actual value at the 0% and 100% positions.

In the manual modes of operation the axis feed rate is shown instead of the contouring feed rate.

You can choose between two types of display:

- The axis feed rate is shown after you press an axis-direction key. If two keys are pressed simultaneously, no feed rate is displayed.
- If no key is pressed, the smallest axis feed rate is always shown. The PLC axes are not included in the selection of the smallest feed rate. If more than one key is pressed simultaneously, a feed rate is also displayed.

▶ Define the type of display in MP7270.

MP7270	**Feed rate display in the operating modes MANUAL OPERATION and ELECTRONIC HANDWHEEL**
Input:	0: Display of axis feed rate through pressing an axis direction key (axis-specific feed rate from MP1020) 1: Display of axis feed rate also before an axis direction key is pressed (smallest value from MP1020 for all axes)

MP7620	**Feed-rate override and spindle speed override**
Format:	%xxxxxxx
Input:	Bit 0 – Feed-rate override if rapid-traverse key is pressed in Program Run mode: 0: Override not effective 1: Override effective Bit 1 – Non-functional Bit 2 – Feed-rate override if rapid-traverse key and machine-direction button are pressed in Manual mode: 0: Override not effective 1: Override effective Bit 3 – Feed-rate override and spindle speed override in 1% increments or according to a nonlinear characteristic curve: 0: 1% steps 1: Nonlinear characteristic curve

		Set	**Reset**
W494	**Percentage for feed rate override (NC to PLC)**	NC	NC
W766	**Percentage for feed rate override (PLC to NC)**	NC/PLC	NC/PLC
M4180	**Rapid traverse programmed (FMAX)**	NC	NC

Feed rate for rotary axes

The iTNC interprets the programmed feed rate for a rotary axis in degrees per minute. The contour feed rate depends on the distance of the tool center from the center of the rotary axis.

With the M116 function the contouring feed rate can be converted to mm/min. In this way the feed rate is independent of the distance from the tool center to the center of axis rotation:

- Define the rotation center of the rotary axis with MP75xx. Also see page 705.

Display of the M functions

The following functions are displayed in the status window:

- M03, M04, M05: Miscellaneous functions for spindle control
- M07, M08, M09: Miscellaneous functions for coolant control

You can control the display of the these functions through the PLC.

- M4005, M4006: Status display, M03 and M04 change the direction of rotation of the spindle.
- M4008: Blocks the speed output for the spindle. The programmed spindle speed continues to be displayed. At the same time, M03, M04 or M05 are highlighted. The nominal speed value is zero.

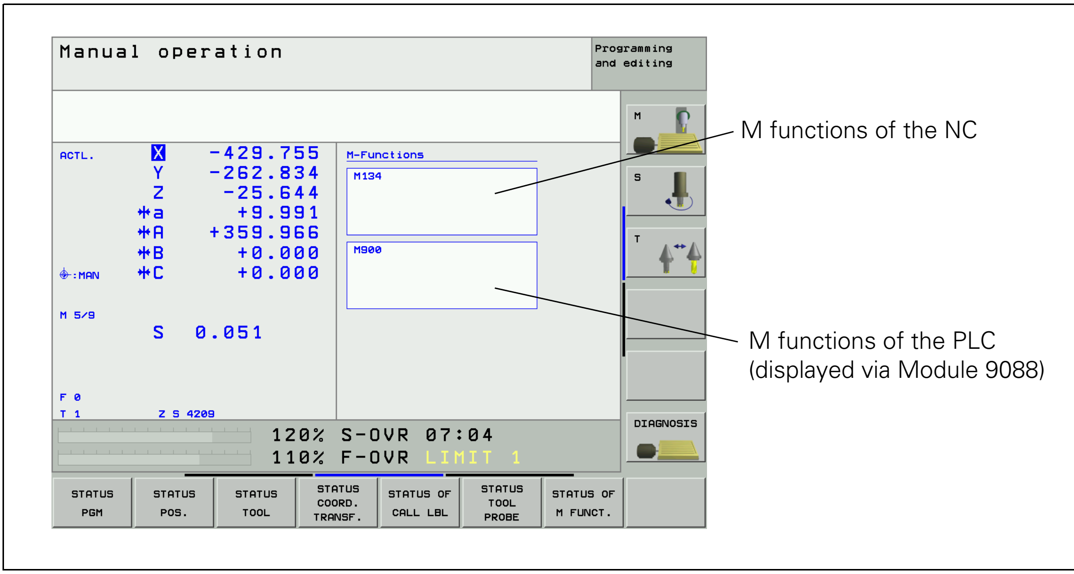
Other M functions of the NC are shown in the status window.

		Set	Reset
M4005	**Status display and nominal speed value output for M03**	PLC	PLC
M4006	**Status display and nominal speed value output for M04**	PLC	PLC
M4007	**Status display M05 and spindle stop**	PLC	PLC
M4008	**Disable speed output for spindle**	PLC	PLC
M4040	**Status display M07, M08, and M09 highlighted**	PLC	PLC
M4041	**Status display M07, M08, M09, MK**	PLC	PLC
M4042	**Status display M07, M08, M09, MK**	PLC	PLC

M4041	M4042	Display
0	0	M09
1	0	M07
0	1	M08
1	1	MK

M functions of the PLC can be displayed in the status window:

- With Module 9088 you can display M functions in the status window or delete them.



Module 9088 Displaying the M functions

Call:

PS	B/W/D/K	<Number of the M function to be displayed>
PS	B/W/D/K	<Mode> –1: Delete all M functions in the status window 0: Delete M function 1: Display M function
CM	9088	

Error recognition:

Marker	Value	Meaning
M4203	0	M function displayed or deleted
	1	Error code in W1022
W1022	1	Invalid M-function number
	2	Invalid mode number

Control in operation

If the control is at work, e.g. executing a positioning movement or an M function, a symbol resembling an asterisk (*) is shown in the status window.

If a running NC program is interrupted with an external stop key, the control-in-operation symbol blinks in the status display window.

In the **Positioning With Manual Data Input, Program Run, Single Block** and **Program Run, Full Sequence** operating modes, these conditions are reported to the PLC with M4175 and M4176.

To delete or display the control-in-operation symbol through the PLC:

- Ensure that the control-in-operation symbol is not already blinking or being displayed by the NC.
- Enter the command code in Module 9089.

Module 9089 Control in operation

The control-in-operation symbol can be set only if it is not already being displayed by the NC, since the NC has priority over the PLC. If the symbol is being displayed by the NC, it cannot be erased.

M4176 is not influenced by the display of the control-in-operation symbol through the PLC.

Call:

PS	B/W/D/K	<Command code> 0: Clear the control-in-operation symbol 1: Display the control-in-operation symbol
CM	9089	
PL	B/W/D	<Error code> 0: Control-in-operation symbol was cleared/displayed 1: Incorrect error code 2: Control-in-operation symbol is already being displayed by the NC 3: Control-in-operation symbol is blinking 4: Control-in-operation symbol was not cleared because it is already being displayed by the NC

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Incorrect transfer parameter

		Set	Reset
M4175	**Program interruption, control-in-operation symbol is blinking**	NC	NC
M4176	**Control is in operation, control-in-operation symbol is on or is blinking**	NC	NC

Clearing the status information

To erase the status information, tool data and contents of the Q parameters:

▶ Select the conditions with MP7300.

Status information is all programmed values, such as scaling factor, datum shift, feed rate, tolerance from Cycle 32, etc. These are all reset, and the tolerance programmed in Cycle 32 is cleared (is only effective again with MP1096.x). The Q parameters and tool data are set to zero.

MP7300 **Erasing the status information, tool data and Q parameters**

Input: 0: Erase the status information, Q parameters and tool data if a program is selected.
1: Erase the status information, Q parameters and tool data if a program is selected and in the event of M02, M30, and END PGM.
2: Erase the status information and tool data if a program is selected.
3: Erase the status information and tool data if a program is selected and in the event of M02, M30, END PGM.
4: Erase the status information and Q parameters if a program is selected.
5: Erase the status information and Q parameters if a program is selected and in the event of M02, M30, END PGM.
6: Erase the status information if a program is selected.
7: Erase the status information if a program is selected and in the event of M02, M30, END PGM.

Input	Erase if PGM MGT	Erase with M02, M30, END PGM	Status information	Tool data	Q parameters
0	x	–	x	x	x
1	x	x	x	x	x
2	x	–	x	x	–
3	x	x	x	x	–
4	x	–	x	–	x
5	x	x	x	–	x
6	x	–	x	–	–
7	x	x	x	–	–

Interrogating the status display through the PLC

Module 9035 Reading status information

With this module you can interrogate the status display or read the status information. You transfer a number indicating the desired information. In order to receive information about the status of the alternative **smarT.NC** operating mode, please use the operating mode markers or the operating mode word (see "Operating modes" on page 1155).

Transferred number		Return Value
0	Editor mode in foreground	0: **Programming and Editing** 1: **Test Run**
1	Machine mode in foreground	0: **Traverse Reference Points** 1: **Manual Operation** 2: **Electronic Handwheel** 3: **Positioning with Manual Data Input** 4: **Program Run, Single Block** 5: **Program Run, Full Sequence**
2	Editor mode in background	0: None (main operating mode active) 1: MOD active 2: Program management/external interfaces active 3: MP editor active 4: PLC programming active
3	Machine mode in background	0: None (main operating mode active) 1: MOD active 2: Program management/external interfaces active 3: Tool table selected 4: Pocket table selected 5: Other table editor active
4	Displayed screen window	Bit encoded Bits 0 to 7: Editing screen Bit 0=1: Editing screen is displayed Bit 1=1: Operating-mode window active Bit 2=1: Block display/program select/setup window active Bit 3=1: Position display active Bit 4=0: Always 0 / Not possible Bit 5=1: Status/Graphics window active Bits 6/7: Reserved Bits 8 to 15: Machining screen Bit 8=1: Machining screen is displayed Bit 12=1: PLC status window active in machining screen Bits 9 to 15: Reserved
5	Selected file in **Programming and Editing** or **Test Run**	0: No file 1: *.H (conversational NC PGM) 2: *.I (ISO NC PGM) 3: *.T (tool table) 4: *.D (datum table) 5: *.P (pallet table) 6: *.A (ASCII file) 7: *.TCH (pocket table)
6	Selected file in **Program Run, Full Sequence** and **Program Run, Single Block**	0: No file 1: *.H (conversational NC PGM) 2: *.I (ISO NC PGM)

Transferred number		Return Value
7	Selected axis for actual position capture in **Programming and Editing** mode	0 to 13: Axes 1 to 14
8	Selected axis for actual position capture in **Positioning with MDI** mode	0 to 13: Axes 1 to 14
9	Handwheel axis	–1: None or more than one 0 to 2: X, Y, Z 3 to 13: axes 4 to 14
10	Handwheel axis, bit-encoded	Bit 0: X axis Bit 1: Y axis Bit 2: Z axis Bits 3 to 13: Axes 4 to 14
	Handwheel subdivision factor	
11 12 13 14 15	X key Y key Z key IV key V key	0 to 10
16	Input format of the $MDI file	0: *.H (conversational NC PGM) 1: *.I file (ISO NC PGM)
17	Display format	0: mm 1: inches
18	Tilting the working plane	Bit 0=1: Tilting is active (manual and/or during program run) Bit 1=1: Tilting is active in manual operation Bit 2=1: Tilting is active in program run
19	Active line in the *.CMA file	>=0: Line number –1: No *.CMA file
20	HR 410 speed	0: Slow 1: Medium 2: Fast
21	Control model	0: TNC 310 1: TNC 370 2: TNC 410 3: TNC 426 CA/PA 4: TNC 426 CB/PB/M or TNC 430 CA/PA/M 5: iTNC 530 6: iTNC 530 (with Windows 2000/XP) 20: ATEK M
22	Status of M128	0: M128 not active 1: M128 active
23	Handwheel superimposition with M118	0: M118 not active Bits 0 to 13: Axes 1 to 14
26	Jog increment	
27	Traverse range	
28	Query of the superimposed operating mode "Table editing" in the "Machine" operating mode	0: No table in editing mode 3: Tool table in editing mode 4: Pocket table in editing mode

Transferred number		Return Value
	Handwheel subdivision factor	
31	Axis 1	0 to 10
32	Axis 2	
33	Axis 3	
34	Axis 4	
35	Axis 5	
36	Axis 6	
37	Axis 7	
38	Axis 8	
39	Axis 9	
	Tool change	
50	Tool change sequence (see **`FN18: SYSREAD ID61 NR0`**)	
51	Pocket number for reserve	
52	Magazine number for reserve	
53	Pocket number for insertion	
54	Magazine number for insertion	
100	Number of the tool axis	
1000	Table editor (only in a spawn job or submit job)	>= 0: Active line in the table editor –1: Table editor not active
1001	Pallet table (only in a spawn job or submit job)	>= 0: Active line in the pallet table –1: Pallet table not active
1002	Status of pallet machining	–1: Main program is not a pallet table 0: Machining was not started 1: NC program is selected but not started 2: NC program was started 3: Pallet-change macro was started 4: Macro from the **`PALEPILOG`** entry in NCMAKRO.SYS was started 5: Pallet-change macro was started by the PLC (Module 9280)

Call:

```
PS    B/W/D/K   <Number of the desired status information>
CM    9035
PL    B/W/D     <Status information>
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid
	20	Call was not in a submit or spawn job

8.1.2 Operating modes

The iTNC features main operating modes, background operating modes, and the alternative `smarT.NC` operating mode. The background operating modes are each called from the main operating modes.

Each operating mode and the varied corresponding information can be interrogated and evaluated via the PLC.

The "Machine" and "Editor" operating modes can be interrogated via markers, W272 and a PLC module (see "Module 9035 Reading status information" on page 1152), and the alternative `smarT.NC` operating mode can be interrogated via Marker M4163 in combination with W272.

		Set	Reset
M4150	**"Manual operation" main operating mode is active**	NC	NC
M4151	**Electronic Handwheel main operating mode is active**	NC	NC
M4152	**Operating mode marker** ■ Positioning with Manual Input main operating mode (if smarT.NC inactive --> M4163 = 0) ■ RUN ACTIVE UNIT smarT.NC main operating mode (if smarT.NC active --> M4163 = 1)	NC	NC
M4153	**Operating mode marker** ■ Program Run, Single Block main operating mode (if smarT.NC inactive --> M4163 = 0) ■ RUN SINGLE UNITS smarT.NC main operating mode (if smarT.NC active --> M4163 = 1)	NC	NC
M4154	**Operating mode marker** ■ Program Run, Full Sequence main operating mode (if smarT.NC inactive --> M4163 = 0) ■ RUN ALL UNITS smarT.NC main operating mode (if smarT.NC active --> M4163 = 1)	NC	NC
M4155	**Traversing the Reference Points main operating mode is active**	NC	NC
M4163	**Alternative operating mode "smarT.NC" is active**	NC	NC
M4186	**NC program was started in the Test Run mode**	NC	NC

		Set	Reset
W272	**Operating mode word** ■ If smarT.NC inactive --> M4163 = 0 1: MANUAL OPERATION 2: ELECTRONIC HANDWHEEL 3: POSITIONING WITH MANUAL DATA INPUT 4: PROGRAM RUN, SINGLE BLOCK 5: PROGRAM RUN, FULL SEQUENCE 7: TRAVERSING THE REFERENCE POINTS ■ If smarT.NC active --> M4163 = 1 3: RUN ACTIVE UNIT 4: RUN SINGLE UNITS 5: RUN ALL UNITS	NC	NC

"Machine" operating modes

- "Machine" main operating modes
  - **Traverse the reference marks** (not with absolute encoders)
  - **Manual Operation**
  - **Electronic Handwheel**
  - **Positioning with Manual Data Input**
    Machine individual NC blocks without a special program context
  - Program Run, Single Block
    Machine an NC program block by block. Each NC block must be started manually.
  - **Program Run, Full Sequence**
    Completely machine the NC program
- "Machine" background operating modes (cannot be called from every main operating mode)
  - **MOD**
    Basic settings and information
  - **Program Management/Interfaces**
    File manager and configuration of external interfaces
  - **Edit Tool Table**
  - **Edit Pocket Table**
  - **Edit Other Tables**

Operating modes "Editor"

- "Editor" main operating modes
  - **Programming and Editing**
  - **Test Run**
    Virtual machining of a workpiece with nearly real-time graphic simulation
- "Editor" background operating modes
  - **MOD**
    General settings and settings via keyword
  - **Program Management/Interfaces**
  - **MP editor**
  - **PLC Programming**

smarT.NC alternative operating mode

Overview of the smarT.NC operating modes

- **smarT.NC: Programming** operating mode
  Program creation. This smarT.NC operating mode is comparable to **Programming and Editing** of the Editor main operating mode.
- **smarT.NC: File Management** operating mode
- **smarT.NC: Testing** operating mode
  This smarT.NC operating mode is comparable to **Test Run** of the Editor main operating mode.
- **smarT.NC: Machining** operating mode
  Program run. This smarT.NC operating mode is comparable to **Program Run, Single Block** and **Program Run, Full Sequence** of the Editor main operating mode.
  It is divided into the following sub-operating modes:
  - „**RUN SINGLE UNITS**
    Machine individual units without a special program context
  - „**RUN ALL UNITS**
    Completely machine the NC program
  - „**RUN ACTIVE UNIT**
    Machine an NC program unit by unit. Each unit must be started manually.

Disable operating modes

PLC Module 9285 can be used to disable operating modes and other functions.

Module 9285 Set the access level

PLC Module 9285 locks/enables the following predefined functions. If the bit is set in the mask, the function is locked. If the bit is set to 0, the function is enabled:

Group 0: Locking operating modes	Bit mask
Manual Operation operating mode	0x01
Electronic Handwheel operating mode	0x02
Positioning with Manual Data Input operating mode	0x04
Program Run, Single Block operating mode	0x08
Program Run, Full Sequence operating mode	0x10
smarT.NC operating mode	0x20

Group 1: Locking the write access to files	Bit mask
NC programs (*.H, *.I, *.HU, *.HP, *.HC, *.DXF)	0x01
Tool table	0x02
Pocket table	0x04
Preset table	0x08
Pallet table	0x10

Group 2: Locking other functions	Bit mask
Manual probing	0x01
Code numbers	0x02
Fixture management	0x04

Call:

```
PS      B/W/D/K   <Group number>
                  0: Locking operating modes
                  1: Locking the write access to tables
                  2: Locking other functions
PS      B/W/D/K   <Bit mask>
CM      9285
PL      B/W/D     <Status>
                  0: Function performed
                  1: Illegal group number
                  2: Incorrect parameterization via bit mask
                  20: Module was not called in a spawn job or submit job
```

Error recognition:

Marker	Value	Meaning
M4203	0	Function performed
	1	Error code in W1022
W1022	1	Illegal group number
	2	Invalid value for bit mask
	20	Module was not called in a spawn job or submit job

8.1.3 Operating times

The iTNC can measure up to 16 operating times and store them in a file in the SYS partition:

Operating time	Meaning
TNCTIME	Control on
MACHINETIME	Machine on
PROGTIME	Program run
PLCTIME0 to PLCTIME12	Definable times of the PLC

For all operating modes except **Programming and Editing** and **Test Run:**

- Press the MOD key and press the **MACHINE TIME** soft key.
- With MP7237.x, specify the times that can be reset with the code number 857282, and the PLC operating times that you wish to display.
- In MP7238.x, define the dialog messages to be displayed for the individual operating times.

The time is measured in seconds and is updated every minute during the run time. When the control is switched off, no more than one minute is lost.

The NC measures the time for **Control on, Machine on** and **Program run.**

For PLC operating times 1 to 13:

- Start with Module 9190.
- Stop with Module 9191.

Except for **Control on,** all operating times are saved during a hard-disk backup. see "Data backup" on page 329.

With the following modules you can evaluate and change the operating times:

- Module 9190: Starting the operating times
- Module 9191: Stopping the operating times
- Module 9192: Reading the operating times
- Module 9193: Setting the operating times
- Module 9194: Alarm when operating times are exceeded

MP7237 **Display and reset the operating times**
Format: %xxxxxxxxxxxxx
MP7237.0 Displaying PLC operating times
Input: Bits 0 to 12 represent PLC operating times 1 to 13
0: Do not display
1: Display
MP7237.1 Reset PLC operating times with code number 857282
Input: Bits 0 to 12 represent PLC operating times 1 to 13
0: Do not reset
1: Reset
MP7237.2 Reset PLC operating times with code number 857282
Input: Bit 0 – Nonfunctional
Bit 1 – "Machine on" operating time
Bit 2 – "Program run" operating time
0: Do not reset
1: Reset

MP7238.0-12 Dialog messages for PLC operating times 1 to 13
Input: 0 to 4095
Dialog no. from the file **PLCDIALOG=** (OEM.SYS)

Module 9190 Starting the operating times
You start one or more operating times.

Call:
PS B/W/D/K <PLC operating time>
Bits 0 to 12 represent PLC operating times 1 to 13
CM 9190

Error recognition:

Marker	Value	Meaning
M4203	0	PLC operating time started
	1	Incorrect parameter

Module 9191 Stopping the operating times
You stop one or more operating times.

Call:
PS B/W/D/K <PLC operating time>
Bits 0 to 12 represent PLC operating times 1 to 13
CM 9191

Error recognition:

Marker	Value	Meaning
M4203	0	PLC operating time started
	1	Incorrect parameter

Module 9192 Reading the operating times

You read the current value of an operating time. The current value is transferred in seconds. If the value is greater than 2 147 483 648 (approx. 69 years), a negative number will be transferred.

Call:

PS	B/W/D/K	<Number of the operating time>
		–3: `Control on`
		–2: `Machine on`
		–1: `Program run`
		0 to 12: PLC operating times 1 to 13
CM	9192	
PL	B/W/D	<Current time [s]>
		–1: Error

Error recognition:

Marker	Value	Meaning
M4203	0	PLC operating time started
	1	Incorrect transfer value, or module was not called in a spawn job or submit job

Module 9193 Setting the operating times

You overwrite the current value of the operating time. The old value is lost irretrievably.

The time for `Control on` cannot be overwritten.

Transfer all values greater than 2 147 483 648 (approx. 69 years) as negative numbers.

Call:

PS	B/W/D/K	<Number of the operating time>
		–2: `Machine on`
		–1: `Program Run`
		0 to 12: PLC operating times 1 to 13
PS	B/W/D/K	<New time [s]>
CM	9193	

Error recognition:

Marker	Value	Meaning
M4203	0	Operating time was overwritten
	1	Incorrect transfer value, or module was not called in a spawn or submit job

Module 9194 Alarm when operating time exceeded

You define a marker that is set when a certain threshold is exceeded. The marker is set every minute after the threshold is passed. The marker can be delayed by max. 59 s the first time it is set. All values greater than 2 147 483 648 (approx. 69 years) must be transferred as negative numbers.

If you enter the value zero as the alarm threshold, the function is deactivated.

Call only in a submit job or spawn job.

Call:

PS	B/W/D/K	<Number of the operating time> –3: **Control on** –2: **Machine on** –1: **Program Run** 0 to 12: PLC operating times 1 to 13
PS	B/W/D/K	<Alarm threshold [s]>
PS	B/W/D/K	<Number of the alarm markers>
CM	9194	

Error recognition:

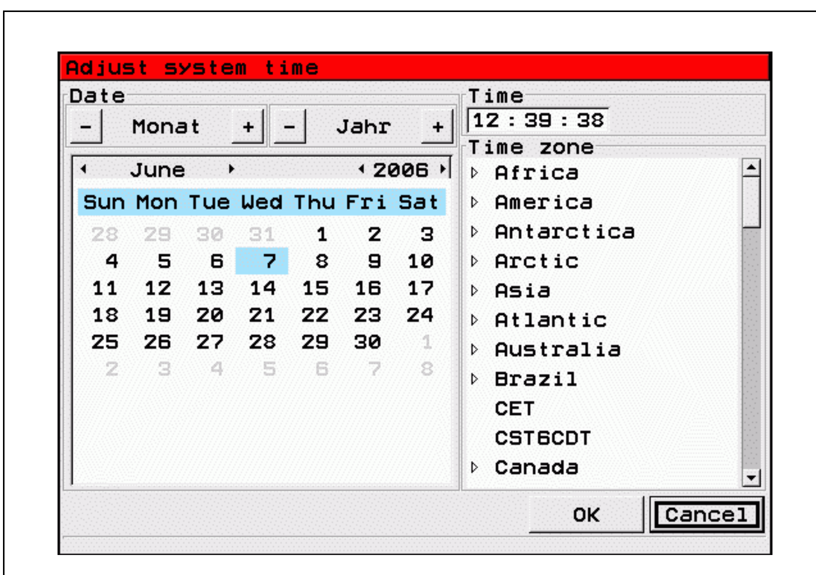
Marker	Value	Meaning
M4203	0	Alarm function activated
	1	Incorrect transfer value, or module was not called in a spawn or submit job

System time

Setting the local time via soft key (only for single-processor control)

Press the MOD key in the Programming and Editing mode of operation to display the SET DATE/TIME soft key. After pressing this soft key, the following window in which you can enter the current local time appears. If you make any changes to these settings, the control must be restarted.

Set the time zone for the control to automatically switch between standard and daylight-saving time.



Module 9195 System time

At the factory, the system time of the iTNC is set to Central European Time or Central European Summer Time in the BIOS. The iTNC internally operates with UNIX system time. The UNIX system time contains the number of seconds accumulated since 0:00 hours on January 1, 1970. The iTNC calculates from the system time (BIOS) to the UNIX system time.

Enter the current local time via the SET DATE/TIME soft key so that the program manager time matches the local time.

With Module 9195 you can read the current value of the UNIX system time. The value read with Module 9195 is independent of the indicated local time and always refers to Universal Time (Greenwich time).

Call:

CM	9195	
PL	D	<System time> Number of seconds since 0:00 hours on January 1, 1970.

MP7235	**Time difference to Universal Time (Greenwich Mean Time)**
Input:	–23 to +23 [hours] 0: Universal Time (Greenwich Mean Time) 1: Central European Time (CET) 2: Central European daylight-saving time

Module 9055 Local time

With Module 9055 you can convert the value read with Module 9195 into a legible ASCII format. Module 9055 corrects the transferred value to local time.

Call:

PS	B/W/D/K	<System time> Number of seconds since 0:00 hours on January 1, 1970.
PS	B/W/D/K	<String number for the result>
PS	B/W/D/K	<Format> 0: DD.MM.YYYY hh:mm:ss 1: D.MM.YYYY h:mm:ss 2: D.MM.YYYY h:mm 3: D.MM.YY h:mm 4: YYYY-MM-DD- hh:mm:ss 5: YYYY-MM-DD- hh:mm 6: YYYY-MM-DD h:mm 7: YY-MM-DD- h:mm 8: DD.MM.YYYY 9: D.MM.YYYY 10: D.MM.YY 11: YYYY-MM-DD 12: YY-MM-DD 13: hh:mm:ss 14: h:mm:ss 15: h:mm
CM	9055	

Error recognition:

Marker	Value	Meaning
M4203	0	String was created
	1	Incorrect transfer value

8.1.4 Error messages

Error messages are displayed by the NC or PLC below the operating mode:

- If the error message blinks, switch the machine off and correct the error. If the error message does not blink, M4177 is set.

You can call PLC error messages with Module 9084 or 9085, or by activating a marker (M4800 to M4999):

- Define your PLC error messages in the *.PET table.
- Assign the markers to the error messages.

With Module 9086 you can delete PLC error messages, and with Module 9087 you can interrogate the current status of the error message.

> Note
>
> A *.PET table is absolutely mandatory, since without it the PLC program cannot be compiled or activated.
>
> If a *.PET table contains more than 2048 error messages, the excessive messages are ignored and the error message **`PET table: Too many lines`** appears.

- In the OEM.SYS file, use the command **`PLCERRTAB=`** to enter the name of the *.PET table.
- You can automatically generate the entry by calling COMPILE.

If more than one PLC error message is activated at once:

- Press the CE key to read the error messages in succession. With the special command **`ERRQUE=`** you can display the list of active error messages in the large PLC window.
- Press the ERR key to have a list of all current error messages displayed. These can be selected and cleared in a targeted manner.

Error list

The columns of the error list (ERR key) have the following meanings:

- **`Number:`** Error number (–1: no error number defined), assigned by HEIDENHAIN or the machine tool builder
- **`Class:`** Error class. Defines how the iTNC processes this error.
  - **`ERROR`**: Collective error class
    The error reaction depends on the status or current operating mode of the control
  - **`FEED HOLD`**: The feed-rate entry is cleared
  - **`PGM HOLD`**: Program run is interrupted (control-in-operation symbol blinks)
  - **`PGM ABORT`**: Program run is canceled (internal stop)
  - **`EMERG. STOP`**: EMERGENCY STOP is triggered
  - **`RESET`**: iTNC performs a reset
  - **`WARNING`**: Warning message, program run is continued
  - **`INFO`**: Informational message, program run is continued
- **`Group:`** Error source. Shows the cause of the error:
  - **`GENERAL`**: Other error
  - **`OPERATING`**: Error during machining
  - **`PROGRAMMING`**: Error during editing
  - **`PLC`**: Error by the PLC
- **`Error message`**: Error text displayed by the iTNC

PET Table

The PLC error message table (*.PET) consists of the following columns, to which you can assign special attributes:

- **`NR`**
  Line number in the table. The modules select the PLC error message by assigning the line number.
- **`ERROR`**
  With the HELP key you can display information on the error messages. (see "Help" on page 1177).
  There are three ways to specify the error text:
  - Direct entry of the error text (max. 32 characters)
  - Line number of the PLC error text file (# <line no.>) specified with **`PLCERROR`**= in OEM.SYS.
  - Number of the string memory containing the error text (# <string no.>)
- **`MARKER`**
  The PLC error message can be activated without module call by setting the marker defined here (M4800 to M4999). The marker is also set if the error message was activated through Module 9085.
  Entry 0: No error marker
- **`RESET`**
  0: No NC reset upon activation of the error message. Error display does not blink.
  1: NC reset upon activation of the error message. Error display flashes.
- **`NC-STOP`**
  0: No NC stop upon activation of the error message
  1: NC stop upon activation of the error message (M4221 is set).

- **NC CANCEL**
  0: No NC stop with subsequent INTERNAL STOP upon activation of the error message
  1: NC stop with subsequent INTERNAL STOP upon activation of the error message (M4223 is set).
- **F STOP**
  0: Feed-rate enable is not influenced
  1: Feed rate-enable is reset upon activation of the error message (M4220 is set).
- **EMER.STOP**
  0: No EMERGENCY STOP upon activation of the error message
  1: EMERGENCY STOP upon activation of the error message (M4222 is set).
- **AE**
  You can use marker M4225 to activate an alternative error reaction, provided that it is available in the PET table. The column **AE** (Alternative error reaction E/C/S/F) of the PET table is evaluated for this. However, the error reactions that are active at the time an error message occurs will become effective. This means that error messages that were initiated before marker M4225 was set, but have not been displayed yet, will trigger the original error reactions. The following entries can be made in the **AE** column:
  - **E** = **EMER.STOP**
  - **C** = **NC-CANCEL**
  - **S** = **NC-STOP**
  - **F** = **F-STOP**
  - **N** = **No Reaction**
  - **X** = **Error will be ignored**
- **CE**
  0: Error message can be cleared with the CE key
  1: Error message cannot be cleared with the CE key
- **PRIOR**
  A priority of 0 to 2 can be entered for the error message. Priority 0 (error) is the highest priority, then priority 1 (warning) and priority 2 (info). PLC error messages triggering an Emergency Stop receive the highest priority (independent of the priority from the *.PET table). Therefore, these error messages always appear at the first position in the error list. The error message **External EMERGENCY STOP** has a lower priority, but still a higher priority than the top PLC priority. This means that PLC error messages triggering an Emergency Stop always appear at the first position in the error list. They are followed by **External EMERGENY STOP** and then by further PLC error messages, depending on their priority. Depending on their priority, M4227 to M4229 are set for active error messages.
  Different colors can be set for the priorities (see "Color settings" on page 1248).
- **MType**
  This column is reserved for future applications. Enter E.
- **ONL-NAME**
  Name of a help file (*.CHM).
- **ONL-NUMMER**
  Help number within the help file under ONL-NAME.
- **SF**
  0: No service file is generated upon an error message.
  1: A service file is generated upon an error message.

- **REACT**
  This attribute results in an NC stop at block end if the error marker is set at this time:
  0: No NC stop at block end upon activation of error message
  1: NC stop or INTERNAL STOP at block end upon activation of the error message

Note

With the attribute **REACT**, not that

- Cycles are executed completely before an NC stop is triggered.
- With smarT.NC units, the current block is executed completely before an NC stop is triggered.
- An NC stop is triggered only if the error marker is (still) set at the end of the block or the end of the cycle.

		Set	Reset
M4220	**Error from PET table with F stop active**	NC	NC
M4221	**Error from PET table with NC stop active**	NC	NC
M4222	**Error from PET table with EM. STOP active**	NC	NC
M4223	**Error from PET table with NC Cancel active**	NC	NC
M4225	**Alternative error reaction active**	PLC	PLC
M4227	**PLC error message with priority 0 (error)**	NC	NC
M4228	**PLC error message with priority 1 (warning)**	NC	NC
M4229	**PLC error message with priority 2 (info)**	NC	NC

Module 9084 Display PLC error messages with additional data

With Module 9084 you can display PLC error messages with additional data. You can insert place holders (%s, %d, %f) at any position of the error messages. The place holders are assigned the data from the module at run time. Only those place holders that are defined in the PLC error message will be replaced. %s is replaced by the string or the string content. The first occurrence of %d or %f in the PLC error message is replaced by the content of variable 1, and the second occurrence of %d or %f is replaced by the content of variable 2. %d is an integer, %f is a floating point number with three decimal places. Alternatively, you can define the number of decimal places with %.1f to %.6f. If the module is called several times with the same line number of the *.PET table, the error message is entered only once in the queue. A maximum of 32 PLC error messages can be entered in the queue. If necessary, the error marker assigned is set. If the *.PET table or the line number is not found, the following error message appears:

PLC-ERROR <line number>

Call:
PS B/W/D/K <line number of the *.PET table>
0 to 2047: Line number
PS B/W/D/K/S<Data for %s>
PS B/W/D/K <Data for %d or %f; variable 1>
PS B/W/D/K <Data for %d or %f; variable 2>
CM 9084

Error recognition:

Marker	Value	Meaning
M4203	0	PLC error message with additional data displayed
	1	Error code in W1022
W1022	1	Line number missing
	8	Incorrect operating mode, compatibility error marker set
	23	Overflow of PLC error message queue

Module 9085 Display PLC error messages

Up to 32 error messages can be placed in the queue, of which up to eight can be from the string memory.

Blinking error message: Is displayed without entry in the queue.

Error number –1: Blinking error message **EMERGENCY STOP PLC** is displayed, even if no *.PET table was defined.

Error number not equal to –1 and no *.PET table selected:
Blinking error message **PLC: NO ERROR TABLE SELECTED**

Call:
PS B/W/D/K <Line number of the *.PET table>
0 to 2047: Line number
–1: Blinking error message **EMERGENCY STOP PLC**
CM 9085

Error recognition:

Marker	Value	Meaning
M4203	0	Error message displayed or in queue
	1	Error code in W1022
W1022	1	Line number missing
	8	Incorrect operating mode, compatibility error marker set
	23	Overflow of PLC error message queue, or too many error messages from string memory

Module 9086 Delete PLC error message

With this module you can erase all set PLC error messages or a specific (nonblinking) error message in the queue.

Call:

PS B/W/D/K	<Line number of the *.PET table> 0 to 2047: Line number –1: Erase all PLC error messages
CM 9086	

Error recognition:

Marker	Value	Meaning
M4203	0	Error message displayed or in queue
	1	Error code in W1022
W1022	1	Line number missing
	8	Incorrect operating mode, compatibility error marker set

Module 9087 Status of PLC error message

Call:

PS B/W/D/K	<Line number of the *.PET table, status code> 0 to 2047: Line number –1: PLC error message, general –2: Number of the active PLC error message –3: Number of error messages in the *.PET table
CM 9087	
PL B/W/D	<Status/error code> For code 0 to 2047: 0: No error message with the number, or message deleted –1: Line number not found (see W1022) Bit 0 – PLC error message is displayed Bit 1 – PLC error message in queue For code –1: 0: No PLC error message 2: PLC error message in queue For code –2: ≥ 0: Number of the displayed error –1: No error in the *.PET table For code -3: ≥ 0: Number of errors in the *.PET table

Error recognition:

Marker	Value	Meaning
M4203	0	Status information was read
	1	Error code in W1022
W1022	1	Invalid line number of status code

Module 9392 Display PLC error messages with help offset

With Module 9392, you can display a PLC error message with an additional help offset and further parameters. The error message must be defined in the .PET table.

The behavior of the module corresponds to the behavior of PLC Module 9084 used in conjunction with Module 9391.

The wild cards %s, %d and %f can be defined at the appropriate places in the error texts. The individual module parameters are assigned to the wild cards (as described in the module interface), and are entered in the error texts at run time. Only the wild cards that occur in an error text are replaced. In order to show the places after the decimal point, the wild cards %.1f ... %.6f can be used. If %f is entered, three decimal places are used.

The module can be called from a cyclic PLC program or from a spawn job or submit job.

An offset is added to the value for the help number in the .PET file in order to generate the actual help number. This way a group error number can be defined for an (OEM) device in the .PET table. The error number (used as an offset) supplied by the device in case of error then leads to the appropriate help text.

Condition:

- The .CHM help file must be stored language-sensitive in the TNC:\tncguide\de directory, or in TNC:\tncguide\en etc.
- An OEM-specific OEMx.CHM file is necessary.
- If no OEMx.CHM file is indicated, the online help is not called. All other reactions that are saved for this error in the .PET table are performed.

Call:

```
PS     K/B/W/D   <Line number in the .PET error table>>
PS     K/B/W/D   <Additional text from S0 to S99 or constant string>
PS     K/B/W/D   <Variable 1>
PS     K/B/W/D   <Variable 2>
PS     K/B/W/D   <Offset for the help number in .PET>
CM     9392
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Transferred parameter outside of value range or error number is not in .PET table
	2	Invalid parameter
	3	String address outside of value range
	8	Incorrect operating mode, compatibility error marker set
	23	Overflow of PLC error message queue

Example:

```
PS     K10               ; Error 10 from .PET
PS     S"ERROR-ERROR"    ; %s
PS     K9000             ; %d
PS     W100              ; %.1f
PS     K2                ; Offset for the help number in .PET
CM     9392
```

Assumption: Text in .PET "%s error in module %d, W100 = %.1f" and W100 = 1234

Therefore, the following output text results:
"ERROR-ERROR error in module 9000, W100 = 123.4"

8.1.5 Service files

Software version 340 49x-04 introduced a function that enables you to save data relevant for service purposes in a .zip file. The appropriate data from the NC and PLC are saved in the file **`TNC:\service\service<xxxxxxxx>.zip`**. The name of the file is generated automatically, whereby **`<xxxxxxxx >`**is the system time shown as an unambiguous character string in hexadecimal code. You can generate a service file in the following ways:

- Press the ERR key and then the SAVE SERVICE FILES soft key.
- It can be provoked by certain configured errors. Here a "1" must be entered in the SF column (**`Create Servicefile`**) of the PET table. A service file is generated no more than once per minute.
- Crash of the NC software due to a fatal error
- Request via TNCremoNT

After a service file has been created, an informational text is displayed. This text indicates that the service file also contains your and your customer's NC data (NC programs, etc.). You and your customers must be aware of this when you give the service file to third parties.

The informational text must be confirmed with the OK button or the ENTER key.

When creating the service file via telemaintenance, the informational text is not displayed. You and your customers must be aware that telemaintenance enables third parties to access your data anyway.

The following data (and other information) is saved in the service file:

- Log
- PLC log
- Selected files (*.H/*.I/*.T/*.TCH/*.D) of all operating modes
- *.SYS files
- Machine parameters
- Information and log files of the operating system (can be partially activated via MP7691)
- Contents of PLC memory
- NC macros defined in PLC:\NCMACRO.SYS
- Information about the hardware
- Error outputs and configuration files of the PLC compiler
- Current Feature Content Level (FCL) and active software options, including the option designations through the file PLC:\SIK.INFO
- The changes to machine parameters via the PLC, LSV2 and NC programs through the MPSERVER.TXT file.

Please remember that OEM logs are not automatically recognized and saved by HEIDENHAIN. However, you can store your own **`PLC:\oemservicefiles.sys`** control file, in which you can enter additional files to be saved.

Behavior when saving the HEIDENHAIN logs:
The size of the log data saved is always between 5 MB and 15 MB.

- Log >= 5 MB
  –>Only the current log is saved.
- Log < 5 MB
  –> The current and the last archived log are automatically saved.

Note

HEIDENHAIN recommends that you encrypt your passwords in order to protect your passwords in *.SYS files when a service file is generated.

When a service file is generated, the *.SYS files are also saved in the .zip file. As a result, however the passwords in the *.SYS files can be accessed by other persons. If you also want to protect your passwords from unauthorized access when the service file is generated, proceed as follows:

- Open the *.SYS file in which you have stored passwords (e.g. OEM.SYS).
- Place the cursor on the line containing a password (e.g. PLCPASSWORD, MPPASSWORD).
- Press the ENCRYPT PASSWORD soft key.
  This encrypts the password and places it after the keyword.

In the default setting, the data are automatically converted to ASCII format when being saved to the .zip file. If however the conversion to ASCII fails, the binary version of the file is saved in the .zip file.

Note

Files that are saved on the encrypted PLC partition **PLCE:** cannot be written to the service file.

You can also store your own control file **PLC:\oemservicefiles.sys**, in which you can enter additional files to be saved.
In addition, your service agency can help the end user save another control file **TNC:\service\userfiles.sys.**

Control files have the following syntax:

- **FOLDER <folder> [<fmask>][-a=<n><u>][-s=<n><u>][-b]**
  Saving all files of a directory. The effect of the command can be influenced by various parameters. The individual parameters must be entered in the described sequence. If more than one criterion is specified, all of the criteria must apply for the file to be saved.

Event	Function
<folder>	Directory that is to be searched for files to be saved. Only the specified directory is searched for files, and not the subdirectories.
[<fmask>]	Optional parameter as a selection criterion for the file name. The wild cards ***** can be used for any number of characters, and the character **?** can be used for exactly one character. Example: ***.TXT**, **PGM12?.H** If no parameter is entered, all files of the directory are saved.
[-a=<n><u>]	Optional parameter for specifying the age of the files to be saved. Files that are older than the specified period of time will not be saved. <n> for the time period: ■ Input of a number <u> for the unit of time: ■ s = seconds ■ m = minutes ■ h = hours ■ d = days Examples: **-a=30s**, **-a=7d, -a=24h**
[-s=<n><u>]	Optional parameter for specifying the size of the files to be saved. Files exceeding the specified size will not be saved. The parameter is to prevent that an NC program with 1 GB is saved, for example. <n> for the size: ■ Input of a number <u> for the unit of size: ■ B = byte ■ k = kilobyte ■ m = megabyte Examples: **-s=100k**, **-s=1m**
[-b]	Binary files are not converted to ASCII files. All binary files are saved in binary format.

- **SETUPFILE <file> <token> [-a=<n><u>][-s=<n><u>][-b]**
  Saving a file whose name is saved in a system file. You can use the same parameters that are used for FOLDER.

Event	Function
<file>	Specify one of the following keywords for the system files that are to be searched for the token: ■ OEM (= PLC:\OEM.SYS) ■ NCPATH (= SYS:\NCPATH.SYS) ■ NCMACRO (= PLC:\NCMAKRO.SYS)
<token>	Enter the token under which the file to be saved is entered in the specified system file. Example: **SETUPFILE OEM MPFILE**, **SETUPFILE NCMACRO PALETT**

- **FILE <fname> [-a=<n><u>][-s=<n><u>][-b]**
  Saving an individual file. You can use the same parameters that are used for FOLDER.

Event	Function
<fname>	Complete path and name of the file to be saved.

- **INCLUDE <file>**
  When a file is specified, it is evaluated as another control file. In order to avoid that a file is specified twice, the INCLUDE statement must not be nested more than three levels.

Event	Function
<file>	Complete path and name of the additional control file.

8.1.6 Help

Help soft key in MOD

With the help file you can display help text, useful information or machine commands:

- Ensure that a help file of the type *.HLP is defined in the system file OEM.SYS with the **MODEHELP =** command.
- Press the MOD key.
- Press the HELP soft key.

You can edit the help file in the PLC editor:

- Press the MOD key and enter the code number 807667.
- Create a *.HELP file.

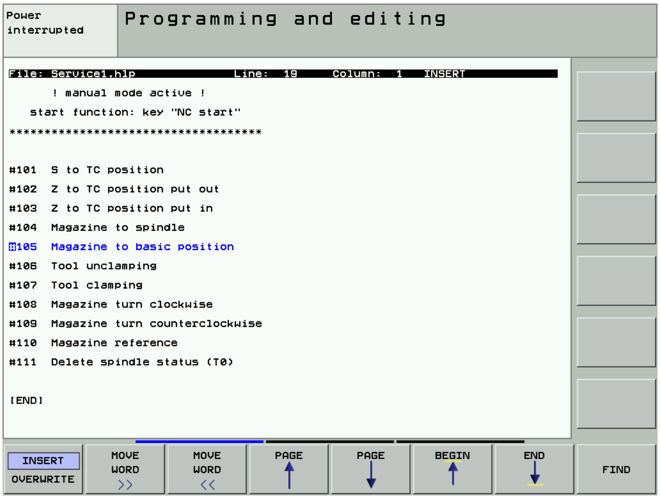
With machine commands:

- Define at the beginning of the line a numerical value in the format #xxxx.

If the user moves the cursor to a line with a numerical value, this number is displayed in W270. In the PLC program you can interrogate W270 and execute the command.

If the cursor is moved to a line without a valid numerical value, the value –2 is entered in W270.

If no HELP file is selected, W270 contains the value –1.



You can create several help files:

- Select the conversational language for help files with MP7230.3. The entry behind **MODEHELP =** in OEM.SYS is overwritten with the language-specific path **(PLC:\LANGUAGE\<Language>).**

The user selects a file:

- By pressing the HELP soft key
- By pressing the PGM MGT soft key

If a HELP file is selected in the foreground and background operating mode, the error message **PARALLEL OPERATION NOT POSSIBLE** appears.

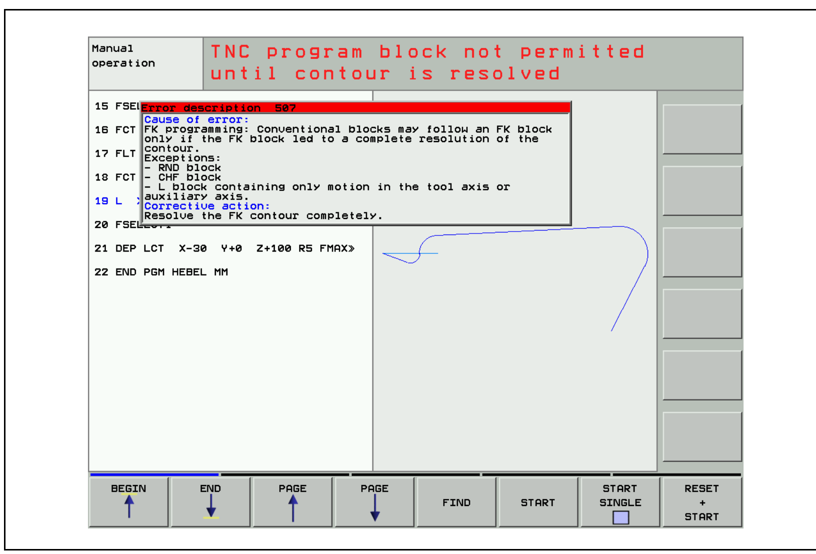
HELP files are saved externally with the identifier J.

		Set	Reset
W270	**Line number in help file** –1: No help file selected –2: Not a valid numerical value 0 to 9999: Line number	NC	NC

Help window with HELP key

If an NC error message appears:

- Press the HELP key.
  You will receive information on the cause and correction of the error.



To make such information available for PLC error messages as well, you must save the texts in files:

▶ Create two files:

- Texts under the heading "Cause of Error" in **REASON.A,** for example.
- Texts under the heading "Corrective Action" in **FIX.A,** for example.

▶ Define the names of both files in the system file OEM.SYS with keywords:

- Cause of error: **PLCERRREASON = REASON.A** (for example)
- Corrective action: **PLCERRFIX = FIX.A** (for example)

▶ Save the files in the corresponding language directories **(PLC:\LANGUAGE\<Language>).**

▶ With MP7230.3, select the active language (file).

The files are divided into text blocks. Each text block can contain up to 10 lines, each with 60 characters. It is ended with <FF>. On the iTNC you can enter an <FF> with the key combination SHIFT + RET. Through the error number (line number) in the PET table, the TNC finds the associated text block in the "error correction" and "error cause" files. Error number zero is the first text block. The error number is also shown in the heading of the help window.

With the PLC programming software PLCdesign, you can generate a PET table and text files, and then transfer them to the iTNC.

In the PLC you can use markers M4220 to M4223 to interrogate the activity of an error from the PET table.

		Set	Reset
M4220	**Error from PET table with F stop active**	NC	NC
M4221	**Error from PET table with NC stop active**	NC	NC
M4222	**Error from PET table with EM. STOP active**	NC	NC
M4223	**Error from PET table with NC Cancel active**	NC	NC

8.1.7 PLC pop-up window

The PLC pop-up (i.e. superimposed) window is shown in the following operating modes:

- **Manual Operation**
- **Positioning with Manual Data Input**
- **Program Run, Single Block**
- **Program Run, Full Sequence**

▶ Activate a pop-up window with Modules 9215 or 9217

If you transfer file names without paths, the iTNC looks for the file in the language-specific directory PLC:\LANGUAGE\<language>:

▶ With MP7230.3, select the active language (file).

When the PLC pop-up window is called, other pop-up windows (such as the help window) are moved to the background and become active again after the PLC pop-up window has been closed. The PLC pop-up window can be displaced to the background by another pop-up window.

The modules do not return until the pop-up window is closed. They must be called in a spawn job, not in a submit job, because otherwise the subsequent submit jobs will not be run until the pop-up window is closed.

A pop-up window called with Module 9215 or 9217 can be ended with Module 9261.

Module 9215 Activating a PLC pop-up window
The module uses the following events:

- $01 000 000 Window build-up
- $00 010 000 Closing the pop-up window

The event for window build-up is generated internally and must not be started externally. If the event for closing the pop-up window is transferred, the module closes without waiting for keyboard input.

The user can make his selections from a list by using the cursor keys and the ENTER key, or the hot keys. The module then returns the line number of the selected menu item. (Line 1 = number 0). You separate the individual entries with <LF>.

Call only in a submit job.

Call:

PS B/W/D/K/S	<String with window heading>
PS B/W/D/K/S	<String with file name of the list> [complete path or only file name]
PS B/W/D/K	<Start position> [line in which the highlight is located]
PS B/W/D/K	<Mode> Bit 0/Bit 1 – Character size 00: Automatic 01: Small character 10: Medium character 11: Large character Bit 2 – Frame 0: With frame 1: Without frame Bit 3 – Heading 0: Display heading 1: Do not display heading Bit 4 – Hot keys 0: No hot keys 1: Hot keys (0 to 9 and A to Z) before menu items Bit 5 – Vertical / Horizontal 0: Vertical arrangement 1: Horizontal arrangement (bit 4 = 0)
CM 9215	
PL B/W/D	<Selected line> 0 to n: Line number from list –1: No selection made (END, NOENT) 2: For error see W1022

Error recognition:

Marker	Value	Meaning
M1022	0	Incorrect mode transferred
	3	Not a valid string for file name or heading
	6	Window cannot be displayed (internal error, e.g. problems with memory or operating system)
	20	Module was not called in a spawn job
	28	Another PLC pop-up window is open
	36	File with the list could not be found

Module 9217 Display pop-up window for messages

Module 9217 can be used in the Machine operating modes to display a pop-up window with the text of a file (max. 16 KB) and various soft-key rows. The pressed key or soft key is reported. Starting at a certain number of lines (up to 500 lines possible), a scroll bar is displayed automatically. Use the arrow keys or the PgUp and PgDn keys to scroll through the pop-up window. The keyboard is assigned to the pop-up window. The width of the window is determined by the longest line.

A displayed window can be closed via Module 9261 (sending of events) through event $010000.

Call:

PS	S	<Window title>
PS	S	<ASCII file with message text>
PS	B/W/D/K	<Mode> 0: No soft-key row; clear window with CE 1: Soft-key row with OK soft key 2: Soft-key row with YES and NO soft keys 3: Soft-key row with YES, NO and END soft keys
CM	9217	
PL	B/W/D	<Pressed key or soft key> –1: Error 0: CE key 1: OK soft key 2: YES soft key 3: NO soft key 4: END soft key

Error recognition:

Marker	Value	Meaning
M4203	0	Pop-up window displayed
	1	Error code in W1022
W1022	1	Invalid mode
	6	No connection to display server
	11	Invalid string for title
	20	Module was not called in a spawn or submit job
	28	Pop-up window already active
	36	.ASCII file with message text does not exist

8.1.8 TNCguide – context-sensitive help system for the iTNC 530 (user documentation)

(FCL 3 function)

The help system was integrated on the basis of the CHM help format known from Windows. The CHM format was introduced by Microsoft in 1997 with the HTML help system, and is now used by many Windows programs. It is a collection of individual HTML files that are collected in a single compressed file.

Mozilla Firefox is used as the browser on the single-processor version of the iTNC 530. The dual-processor version of the iTNC 530 and the iTNC 530 programming station use Internet Explorer or the standard browser configured for your PC.

In principle, the iTNC 530 will be able to open and display any CHM files, including those prepared by the OEM. HEIDENHAIN will generally supply the following documentation in the form of CHM files together with the iTNC software:

- Conversational Programming User's Manual
- Touch Probe Cycles User's Manual
- smarT.NC Pilot
- Collection of all NC error messages

The User's Manual for the programming station is also supplied with the iTNC 530 programming station. Context-sensitive entry points are defined for these files, meaning that pressing the HELP key brings the user directly to the appropriate location in the documentation. If no context-sensitive entry point is available, the TNC opens the parent **main.chm** book file, in which all CHM files in the respective (language-sensitive) help directory are shown. The user navigates to the desired entry with the mouse or the arrow keys.

TNCguide can also be opened with the mouse: after clicking the help symbol, shown to the right directly above the soft-key row, the mouse pointer changes to a question mark. If the user then clicks a soft key, the TNC starts the help system and displays the description of this soft key, if a context-sensitive entry point exists. If no context-sensitive entry point is available, the control opens the parent **main.chm** book file, as described above.

Basic conditions for the OEM

So that CHM files created by OEMs can be shown in the **main.chm** book file, the conditions listed below must be followed:

- The OEMx.CHM files must be stored in the **TNC:\tncguide\de** directory, or in **TNC:\tncguide\en,** etc. When the dialog language is switched (with MP7230), the iTNC searches the corresponding language subdirectory when the help system is called.
  HEIDENHAIN recommends placing an OEMx.CHM file in English in each language subdirectory if you don't translate your documentation into every language. This ensures that online help is available for all topics, regardless of the language settings on the control (MP7230). If no OEMx.CHM file exists in the language subdirectory, no OEM-specific help is shown when the online help is called.
- HEIDENHAIN has already defined the names for the CHM files created by the OEM, so that these files can be displayed as books (if they exist) within the parent **main.chm** book file:

Name of CHM file	Help-number range from	to
OEM1.chm	10.000.000	10.999.999
OEM2.chm	11.000.000	11.999.999
OEM3.chm	12.000.000	12.999.999
OEM4.chm	13.000.000	13.999.999
OEM5.chm	14.000.000	14.999.999
OEM6.chm	15.000.000	15.999.999
OEM7.chm	16.000.000	16.999.999
OEM8.chm	17.000.000	17.999.999
OEM9.chm	18.000.000	18.999.999
OEM10.chm	19.000.000	19.999.999

- The help-number range shows the context-sensitive entry points that are permanently defined for each file in order to simplify entry via the parent main.chm book file.
- The following links contain useful information about HTML help as well as software for downloading:
  http://msdn.microsoft.com/library/en-us/htmlhelp/html/vsconHH1Start.asp
  http://www.helpware.net/
- HEIDENHAIN recommends using Mozilla Firefox 1.0.x to check how the HTML pages are displayed, since it is used on the control. The view differs somewhat from the view in Internet Explorer, especially regarding the page layout. However, this does not replace a thorough test of the CHM file on the control.

Using PLC modules to call CHM files created by the OEM:

- PLC Module 9391 is introduced, with which a PLC error message is displayed, and in addition an offset is added to the value of the error number (= ONL Number) in the .PET table in order to generate the actual help number. This way a group error number can be defined for an (OEM) device in the .PET table, and the error number (used as an offset) supplied by the device in case of error then leads to the appropriate help text. This requires an OEM-specific *.CHM file, which must be indicated in the .PET table (= ONL Name).
- Additionally, PLC Module 9390 is also introduced, with which a help window can be opened directly by the PLC.

Every possible way of calling your OEM help file offers you the possibility of showing the entire directory (including HEIDENHAIN help files) in the directory tree, or just the directory of your OEM help file. This selection is made when calling the help file. If you enter **main.chm** as the help file in the call (via PLC module, *.PET file, soft key or NC error), then the entire directory is shown. If you enter your OEM help file OEMx.CHM as help file in the call, then only the directory of your help file is shown.

Files and structure of the help system

All online help available on a control can be called separately as well as within the online help system. In order for the call within the entire system to function, the conditions described below must be followed:

The help system is structured as follows:

- main.chm – Welcome page of the help system
  - jh1.chm
  - jh2.chm
  - ...
  - oem1.chm
  - oem2.chm
  - oem3.chm
  - ...

The HTML pages of the individual online help topics exist completely independently of each other. Special entries combine the tables of contents and collate the index entries (see below).

Table of contents

The table of contents of main.chm contains "merge" objects:

```
<OBJECT type="text/sitemap">
   <param name="Merge" value="oem1.chm::/merged.hhc">
</OBJECT>
```

HEIDENHAIN has specified here the name of the CHM file and the file name **merged.hhc** of the table of contents contained therein. When the welcome page of the help system (main.chm) is called, all existing help files linked with merge commands are included. Each table of contents is only displayed if it exists in a file named **merged.hhc.** This is why it is essential that you include the directory structure of your help as **merged.hhc** when your generate your **OEMx.CHM** files.
The first level of this table of contents should have exactly one entry: the title of the help file. This title than appears as a "book" in the entire table of contents, and can be opened by the user in order to show the subordinate headings.

Along with the actual HTML pages, the following files must be included when generating the OEMx.CHM files:

- *.hhc file
  In this file you describe the structure and format of your help system. If you call your help file via main.chm, then this file must be named **merged.hhc.** If your help file is to be called directly, without **main.chm,** then the directory tree in the *.hhc file that you indicated as content file when you created the CHM file is used.
- *.hhk file
  In this file you list all entries that are to be shown in the index later, and create the links to the corresponding HTML pages. At least one entry is necessary here in order for your help file to be displayed.
- *.hhp file
  This file is the project file that is necessary for generating an OEMx.CHM file.
- *.h file (only necessary for context-sensitive help)
  In this file you use the **#define** command to establish connections between error numbers, from the respectively valid ranges of error numbers, and any variables.
  Example:
  #define IDH_OEM1_CHAP110000000
  #define IDH_OEM1_Page1_110000100
- *.txt file (only necessary for context-sensitive help)
  In this file you establish the connection between the variables and the corresponding HTML pages, which are then called.
  Example:
  IDH_OEM1_CHAP1=chapter1.html
  IDH_OEM1_Page1_1=page11.html

INDEX

Note: "Merging" of the indexes only functions if each file involved contains at least one index entry, meaning at least one index entry is also necessary for the OEM help file. In addition, the entry "Binary Index=Yes" must be set in the project file (*.hhp).

The project file of main.chm contains the following entries:

- [MERGE FILES]
  - jh1.chm
  - jh2.chm
  - ...
  - oem1.chm
  - oem2.chm
  - oem3.chm
  - ...

This collates and displays the index entries of all present and named help files when the index of main.chm is called.

Context-sensitive call

In a context-sensitive call of the OEM help, the index display starts from the OEM help, and here the index entries can only be collated if all other help files *except* the current OEM file itself are entered in the project file of the OEM help.
HEIDENHAIN makes a complete list of the file names available.

Summary for OEM help

- Project file: Binary Index=Yes
- Project file: [MERGE FILES] with current list *except* its own file name!
- Table of contents: "merged.hhc" – Title of the help, can be opened as a "book."FrameMaker
- Index: At least one index entry.

Designing soft keys for context-sensitive help

In order to establish context sensitivity in combination with soft keys, the additional HELPID and HELPFILE soft keys must be entered in the descriptions of the soft keys. Use HELPID to assign to a soft key an ID (= error number) from the respectively valid range of error numbers. With the HELPFILE attribute you indicate in which *.chm file the error number can be found (e.g. OEM1.chm) or via which file the error number is to be searched for (e.g. main.chm). The assigned error number, in combination with the *.txt and *.h files, is used to call the appropriate HTML page when the help is called.
If you enter main.chm as the HELPFILE, the entire directory structure with the HEIDENHAIN help files is shown. If you enter only one file (e.g. OEM1.CHM), then only the directory of this file is shown. The OEMx.CHM files must be stored in the **TNC:\tncguide\de** directory, or in **TNC:\tncguide\en,** etc. When the dialog language is switched (with MP7230), the iTNC searches the corresponding language subdirectory when the help system is called.
You can enter HELPFILE for each soft key, or once for all soft keys.

Example:

```
PLC SOFTKEY Project File - Version 1.0

; Path for the soft-key help file
HELPFILE    'TNC:\tncguide\de\oem1.chm'

...

; here the assignment of a HELPID to the soft keys
;without indication of a *.chm file. This automatically links to
; the help file indicated above.
ACTION Action2_Softkey HELPID:10000000

PULSE  Pulse1_Softkey  HELPID:10000100

BLANK
ENDSKMENU
```

or:

```
PLC SOFTKEY Project File - Version 1.0

...

; here the assignment of a HELPID and a HELPFILE to
; the soft keys. This automatically links to the
; indicated help file.
ACTION Action2_Softkey HELPID:10000000
HELPFILE:TNC:\tncguide\de\oem1.chm

PULSE  Pulse1_Softkey  HELPID:10000100
HELPFILE:TNC:\tncguide\de\eoem1.chm

BLANK
ENDSKMENU
```

Enhanced error notification

- PLC error messages
  For each entry (i.e. error message) in the PET file, the machine manufacturer can enter the name of a help file (*.CHM) and a help number within this *.CHM file. This is done with the two new columns in the PET table, ONL Name and ONL Number. The called help files must—as mentioned earlier— be present and language-sensitive in the TNC:\tncguide\de, TNC:\tncguide\en, etc. directories. When a PLC error message is current and the user then presses the ERR key and the HEIDENHAIN TNCguide soft key, the appropriate chapter from this file is shown (context-sensitive call). If a help number but no *.CHM file is indicated, the main.chm file is automatically shown. If the OEM has adhered to the permanently defined help-number ranges for the respective files (see above for the help-number ranges), then the correct, context-sensitive help page is shown in this case as well.
- NC error messages
  The machine manufacturer has the possibility of calling his own additional error descriptions for NC errors. This is also done with a separate CHM file. With the OEMERRORCHM.FILE and OEMERRORCHM.OFFSET entries in OEM.SYS, context-sensitive links to any page in the OEM file can be created. Under OEMERRORCHM.FILE you must enter the *.CHM file in which the error number is to be searched. If main.chm is entered under OEMERRORCHM.FILE, the error number is searched for from there and the entire directory tree of **main.chm** is displayed. If the error number is not found, the start page of main.chm is shown.
  Under OEMERRORCHM.OFFSET you must enter an error number that is used as an offset. The number of the NC error is added to the number entered here. The resulting error number is then searched for.
  Example:
  ```
  OEMERRORCHM.FILE       = main.chm
  OEMERRORCHM.OFFSET     = 10000000
  ```
  If, for example, the NC error message **Key nonfunctional** is shown (NC error number 938), the resulting error number = 10000938.
  If you press the ERR key now, two soft keys are offered.
  If you press the HEIDENHAIN TNCguide soft key, the HEIDENHAIN error description appears. If you press the MACHINE MANUFACTURER soft key, the resulting error number (e.g. 10000938) is searched for in the given CHM file. The appropriate page is opened if the error number exists, otherwise the start page of the given file opens.
  The help files to be called must—as mentioned earlier— be present and language-sensitive in the TNC:\tncguide\de, TNC:\tncguide\en, etc. directories.
  The MACHINE MANUFACTURER soft key is only displayed if one of the two entries named exists in OEM.SYS.

Including an OEM-specific online help file

Once you have created a valid *.chm file, proceed as follows in order to display your own OEM-specific help file in the HEIDENHAIN TNCguide:

- You may need to rename your **`*.chm`** file. You must use one of the names reserved by HEIDENHAIN for OEM help files.
  e.g. **`OEM1.CHM`**
- Use TNCremo to transfer the help file to the control.
- Store your help file in the appropriate language directory:
  `TNC:\tncguide\de, TNC:\tncguide\en,` etc. If you have created only an English help file, HEIDENHAIN recommends placing it in the other language directories as well.
- Press the HELP key to call the TNCguide. Your help file should now automatically be included in the TNCguide directory tree.

National languages

CHM files will not be available for all possible TNC dialog languages at the time the NC software is released. We are planning on delivering the German and English languages with the software, and the CHM files in other languages will be made available for free downloading (for unregistered users as well, of course) from our FileBase. The user then simply downloads the appropriate file(s) for the respective language(s), and stores them in the directory defined by us on the TNC hard disk: **`TNC:\tncguide\de`** or the appropriate language subdirectory.

A **`readme.a`** file is located in the respective language directories on the TNC hard disk under **`TNC:\tncguide\`**. This file describes the procedure for loading the CHM files from our FileBase.

Online help files and TNCremoNT

The online help files (with file name extension .chm) can be managed with TNCremoNT:

- **Transfer of *.chm files:**
  Online help files are binary files. If TNCremoNT is updated to version 2.5, the file extension .chm is automatically added to the list of binary file types. Otherwise the list of binary file types must be amended manually under Extras > Configuration on the Mode tab in order to transfer them correctly.
- **Performing a backup of the TNC via TNCremoNT:**
  Online help files are automatically untagged during creation of the scan list used for the backup. This also applies to CHM files that the machine manufacturer has saved on the control.
  Reason:
  The *.chm files saved on the control require a large amount of memory, and do not need to be backed up, since they are freely available from the HEIDENHAIN homepage.
  Please note that only online help files from HEIDENHAIN are available here. This setting can be applied to other file types as well in TNCbackup under Edit > Settings, if necessary.

Module 9390 Open the online help window with the control's browser

With Module 9390 you open an online help window with the control's browser. The module can be called from a cyclic PLC program or from a spawn job or submit job. The extension .CHM can be given in the call, but is not necessary.

Constraints:

- The .CHM help file must be stored language-sensitive in the **TNC:\tncguide\de** directory, or in **TNC:\tncguide\en** etc.
- If an empty string is given as file name, the **main.chm** file is used. Depending on the link in **main.chm** the context number then branches to the **OEMx.CHM** specified there.
- If 0 was given as the help number, or if the given number could not be found in the help system, the start page of the given help file is opened.
- If the help system could not be started (e.g. the .CHM file is missing), an NC error message with a reference to this PLC module is generated.

Call:

PS	SXX	<File name> from string number S0/S1/... or S "file" or S""
PS	K/B/W/D	<Help number of the HTML page to be opened from the respectively valid help-number range>
CM	9390	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid parameter or parameter does not exist
	22	Help system could not be started

Module 9391 Display an error number with additional offset

With Module 9391 you can display a PLC error message from the .PET table with an additional offset. Module 9391 corresponds in behavior to Module 9085. The module can be called from a cyclic PLC program or from a spawn job or submit job.

An offset is added to the value for the help number in the .PET file in order to generate the actual help number. This way a group error number can be defined for an (OEM) device in the .PET table. The error number (used as an offset) supplied by the device in case of error then leads to the appropriate help text.

Two new columns were added to the .PET table for this. In the column ONL-Name you enter the file name of the help file that is to be called additionally in connection with an offset. In the ONL-Number column you enter the ID (=help number) of the HTML page in the help file to be opened. The offset is added to this ID (=help number) during the module call, and is given by the module. This makes it possible to refer to specific HTML pages in addition to the more general PLC error message.

Constraints:

- The .CHM help file must be stored language-sensitive in the **TNC:\tncguide\de** directory, or in **TNC:\tncguide\en** etc.
- An OEM-specific **OEMx.CHM** file is necessary.
- If no **OEMx.CHM** file is indicated, the online help is not called. All other reactions that are saved for this error in the .PET table are performed.

Call:

PS	K/B/W/D	<Line number in the .PET error table>
PS	K/B/W/D	<Offset for the help number in .PET>
CM	9391	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid parameter or parameter does not exist

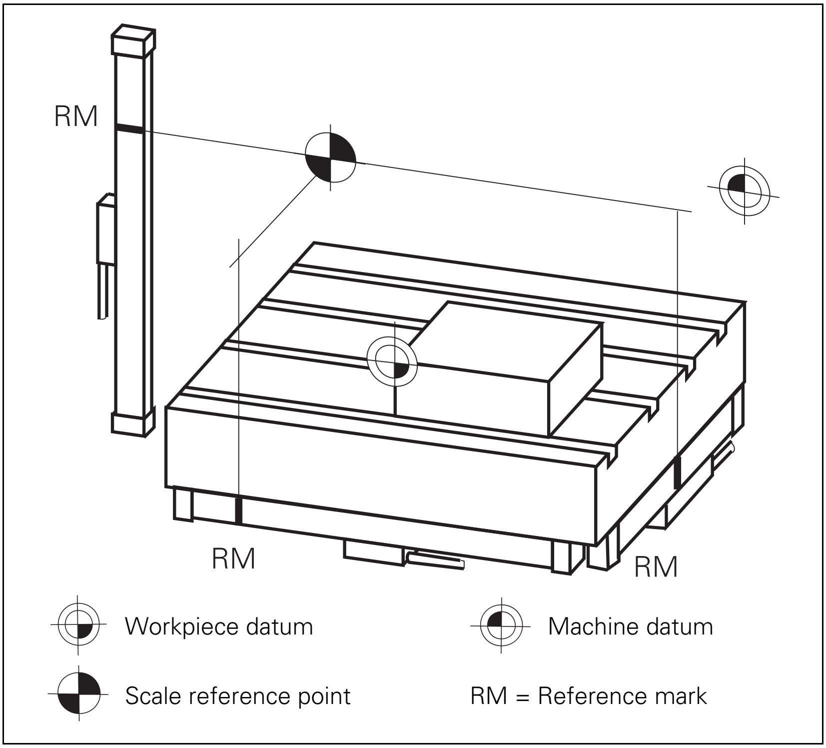
8.1.9 Machine datum

The machine is built with a fixed **machine datum**. All referenced displays and positioning blocks refer to this machine datum.

In the `Manual` and `Electronic Handwheel` operating modes you can define a **workpiece datum** with the "datum setting" function. NC programming blocks are entered with respect to this defined datum.

Other reference points:

- If the user programs M91 in an NC program, the NC programming block refers to the machine datum.
- ▶ In MP960.x, enter the distance between the machine datum and the scale reference point.
- All NC programming blocks are referenced to the machine datum.
- ▶ With MP7295, disable the "datum setting" function for specific axes.
- NC program block values are defined with respect to fixed positions of the machine if the user programs M92 in the NC program.
- ▶ In MP950.x, enter the distance between the machine datum and the machine-referenced position. The values for MP950.x can be assumed with the "actual position capture" key.



Note

M91 and M92 are active only in the block in which they are programmed.

Datum setting

The datum can be set either by only the "Datum setting" soft key or by the soft key plus the axis keys:

- With MP7296, define how the datum should be set.

To change the datum in the OEM cycles:

- Press the MOD key.
- Enter the code number 555 343.
- Enter: **`FN25: PRESET <axis>/<value to be converted>/<new datum>`**.

<Axis>	Axis for which the datum is to be set
<Value to be converted>	Coordinate in the active coordinate system to which the datum is to be set, or the number of the Q parameter that contains this coordinate
<New datum>	Desired value of the datum

Note

The currently effective datum, but not the corresponding value in the preset table, is overwritten with **`FN25: PRESET.`**

If two axes are being operated paraxially, for example with two alternating spindle motors, you can specify that the datum being set in one axis with "Datum setting" should also apply to the other axis.

- Enter in MP7492.x the number of the axis that is to also use the same datum. The index of MP7492.x defines the axis whose datum is to be applied.

Example:
Axes 0 and 1 are parallel axes. When setting the datum for Axis 0, the datum is also to apply to Axis 1. In this case you must set MP7492.0 = 1.

MP950.x **Datum for positioning blocks with M92**
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
Values with respect to the machine datum

MP960.x **Machine datum**
Input: –1.79769313486E+308 to
+1.79769313486E+308 [mm] or [°]
Values with respect to the scale reference point

MP7295 **Disable "Datum setting"**
Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Not disabled
1: Disabled

MP7296 **"Datum setting" through axis keys**
Input: 0: Datum can be set by axis keys and soft key
1: Datum can be set only by soft key

MP7492.x **Number of axis in which the same datum is to be set during datum setting (with active preset table)**
Input: 0 to 17
–1: Do not set a datum
MP7492.0 Datum set in the first axis
to
MP7492.13 Datum set in the fourteenth axis

Datum management via preset tables

As of NC software 340 422-01 and 340 480-01, the datums are managed in the preset table TNC:\PRESET.PR:

- With MP7294, disable the "datum setting" function in the preset table for specific axes.

Each traverse range uses its own preset table. The preset table of the current traverse range can be viewed in the **Manual** operating mode. **FN17: SYSWRITE** and **FN18: SYSREAD** refer to the active preset table.
FN17: SYSWRITE ID503 is used for entering a value directly (i.e. without conversion) into the preset table. This way a defined state can be set as a default value in the preset table.
FN17: SYSWRITE ID502 and **FN18: SYSREAD ID502** are used for entering a value with conversion into the active coordinate system or preset table, or for reading it.

Note

The currently effective datum, but not the corresponding value in the preset table, is overwritten with **FN25: PRESET.**

The **PR.LINESLOCKED =** entry in OEM.SYS is used to write-protect lines in the preset table, such as for the machine datum. Separate each line number with a comma, and connect line ranges with a dash: **PR.LINESLOCKED = 1,4-8,22**. Write-protection can only be assigned to the first 255 lines. Write-protected lines are shown in a different color. The active line and line 0 are always write-protected. The manually set datum, which was set by the user with the axis keys, is automatically written into line 0 of the preset table.

Up through software version 340 49x-02, the entry **PRESETTABLE = OFF** in OEM.SYS can be used to deactivate the preset table after a control reset. The stored datums are no longer valid after the reset, and a new datum must be set in each axis (possible in all traverse ranges). The active status of preset table is reported to the PLC with M4589=1.

Note

The meanings of the bits in MP7500 depend on whether the preset table is active.

When the control is started up, new cycle data is generated (with or without support from the preset table).
It is also possible to take the presence or absence of the preset table into account in OEM cycles. See page 1640.

		Set	Reset
M4589	**Activate datum management via preset table**	NC	NC

MP7294 **Disable axis-specific "Datum setting" in the preset table**
Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Not disabled
1: Disabled

8.1.10 NC program

Setting "look-ahead"

In order to adapt the feed rate to the workpiece machining process, the iTNC 530 precalculates the geometry. This way changes in directions (corners, curvatures, and changes in curvatures) are detected in time, and the participating NC axes can be braked or accelerated accordingly.

You can set the number of NC blocks to be used for precalculation. The greater the number of blocks for advance calculation, the higher the possible feed rate. Shorter machining times can be achieved.

This improved effect is especially noticeable with many short traverse blocks in the micron range, which are exported from CAD/CAM files, for example. However, the default setting of 256 blocks usually suffices.

MP7400 **Setting look-ahead**
Input: 0: 256 [blocks] (default)
1: 512 [blocks]
2: 1024 [blocks]

Block number increment for ISO programs

▶ In MP7220 enter the block number increment for DIN/ISO programs.

MP7220 **Block number increment for DIN/ISO programs**
Input: 0 to 250

Finding the block number

Module 9321 Find the current block number
Please use Module 9322 if possible!
The current block number is ascertained with Module 9321.

Call:
PS B/W/D/K <String number (reserved for future applications)>
CM 9321
PL B/W/D <Current block number>

Error recognition:

Marker	Value	Meaning
M4203	0	Block number has been found
	1	Error

Module 9322 Information of the current NC program

With Module 9322, you can determine the current block number of the active NC program. If the module is called from the cyclic PLC program, only the block number of the current NC main program is read in real time. If the module is called from a spawn job or submit job, the path of the current NC program is determined in addition to the block number (from the block scan).

Constraints:

- Because of the geometry look-ahead, the call from the cyclic PLC program only supplies the block number in real time, but no information about the NC program.
- Call from the cyclic PLC program:
  For all traverse blocks that are not generated from an NC program, a cycle or an NC macro, block number –1 is read.
- Call from the cyclic PLC program:
  After the NC program is cancelled or after the end of the NC program, the last block number executed is returned.

Call:

PS	B/W/D/K	<Mode> When called from a cyclic PLC program, the setting of <Mode> is without effect. The block number of the active NC program is always returned. When called from a spawn job or submit job: 0: String / block number and path refer only to the active NC (sub)program. Block number from block scan. 1: String / block number and path refer to the active NC (sub)program or cycle. Block number from block scan. 2: Only the name of the NC main program without information about the block number (block number is set to 0 when executed correctly).
PS	B/W/D/K	<String number for path of the NC program or cycle> Call from a cyclic PLC program: Without effect.
CM	9322	
PL	B/W/D	<Block number of current NC program or cycle> –1: Error if error marker is set –1: Block number in certain cases, if call was from a cyclic PLC program

Error recognition:

Marker	Value	Meaning
M4203	0	Information of current NC program has been read
	1	Error code in W1022
W1022	1	Invalid mode
	2	Invalid string number

NC program selected

With marker M4181 it is possible to interrogate whether an NC program is selected in the `Program Run, Full Sequence` or `Program Run, Single Block` mode of operation. The marker is not set if an NC program is selected from a pallet table.

		Set	Reset
M4181	**NC program selected**	NC	PLC

Display of the NC program

The NC program can be displayed in various layouts:

- Set MP7281 = 0 to show all blocks completely.
- Set MP7281 = 1 to show only the current block completely. All other blocks are shown as only one line.
- Set MP7281 = 2 to show all blocks as one line. A block is shown completely only when it is created or edited.

MP7281	**Depiction of the NC program**
Input:	0: All blocks completely 1: Current block completely, others line by line 2: All blocks line by line; complete block when editing

Deleting lines from an NC program

Individual blocks of an NC program can be deleted without a confirmation question.

- Set MP7246 bit 1=1 to have to the confirmation question `Really delete NC block?` and the soft keys YES, NO and END appear when the DEL key is pressed.

MP7246	**Machine parameter with multiple function**
Input:	Bit 1 – Clear with DEL key 0: Does not need confirmation 1: Must confirm via soft key

Checking the NC program during editing

The control checks an NC program while editing it. The duration depends on the number of lines to be checked. Errors that occur after the defined line number are not recognized during editing.

- Enter in MP7229.0 the line number to which the NC program is to be checked (LBL and TOOL DEF blocks).
- Enter in MP7229.1 the line number to which FK blocks are permitted. If FK blocks do not appear until after this line number, they are not checked.

MP7229	**Properties of the NC program**
MP7229.0	Line number for program testing
Input:	100 to 9999
MP7229.1	Program length to which FK blocks are allowed
Input:	100 to 9999

Status information about the end of an NC program

Module 9320 Status of the NC program end

Module 9320 can ascertain status information on the termination of the NC program.

Call:

CM	9320	
PL	B/W/D	<Cause of NC program end> 1: Emergency stop 2: Positioning error 3: Programmed stop (stop, M00) 4: Normal end 5: Geometry error 6: END PGM, M02 7: Internal stop 8: RS-232-C transmission error
PL	B/W/D	<Error class> Bit 0: Reserved Bit 1: Control loop Bit 2: Probing Bit 3: Limit switch Bit 4: Error from FN14 Bit 5: Tool management Bit 6: Programming error Bit 7: Program selection/preparation Bit 8: Pallet administration Bit 9: Emergency stop
PL	B/W/D	<Help number> –1: No help number
PL	B/W/D	<Block number in the NC program>
PL	B/W/D	<Additional information> In the "tool management" error class: Tool number

		Set	Reset
M4185	**Internal stop performed**	NC	PLC

Canceling an NC program

An NC macro can be called automatically if an NC program was canceled by an error message or an external or internal stop. You can use it to exchange information between the NC and the PLC. This NC macro may not contain any positioning commands, or the error message **`Program data erroneous`** will appear.

- In NCMACRO.SYS enter the name (and path) of the NC macro after the code word **`RUNCANCEL =`**.

Automated NC program start

NC programs and pallet tables can be started by the iTNC automatically at a date and time set by the user. To use the Autostart function:

- Use MP7683 bit 5 to show the AUTOSTART soft key.
- With MP7683 bit 7, specify whether the NC program should be started by the NC or the PLC after the expiration of the programmed time.
  Two markers are available for starting the program by the PLC:
  - M4182 indicates whether the AUTOSTART function was activated.
  - M4183 indicates whether the time programmed by the user has expired.
- Switch to **Program Run, Full Sequence** mode and use **PGM MGT** to activate the NC program or pallet table to be processed. No current error messages are allowed.
- Press the AUTOSTART soft key.
- Enter the date and time at which the machine is to be switched on.
- Set M4586 for the PLC to enable the Autostart function. If the PLC does not enable the function, the error message **Autostart not enabled** appears.
- Activate the Autostart function with the AUTOSTART ON soft key. **Active** blinks in the window.

		Set	Reset
M4182	**AUTOSTART active**	NC	NC
M4183	**Time from AUTOSTART expired**	NC	NC
M4586	**Enable AUTOSTART**	PLC	NC/PLC

MP7683 **Executing pallet tables**

Input: Bit 5 – AUTOSTART soft key
0: Do not display soft key
1: Display soft key
Bit7 – AUTOSTART function by PLC
0: AUTOSTART function performed by the NC
1: AUTOSTART function is performed by the PLC. The NC does not trigger an NC start.

Sorted machining of block elements

The DIN/ISO editor of the iTNC 530 does not automatically sort block elements. This means that the block elements TOOL CALL and S in DIN/ISO blocks are machined at the location they were programmed at. If in NC programs (such as from a postprocessor) these block elements are not programmed until the end of the block, then they won't be machined until the end of the block.

- Set MP7682 bit 7 for the block elements TOOL CALL and S in DIN/ISO blocks to automatically be machined at the beginning of the block. The display of the block elements does not change.

MP7682 **Machine parameter with multiple function**

Input: Bit 7 – Block elements TOOL CALL and S in DIN/ISO blocks
0: Machine as programmed
1: Machine at beginning of block (block display does not change)

Behavior during program interruption

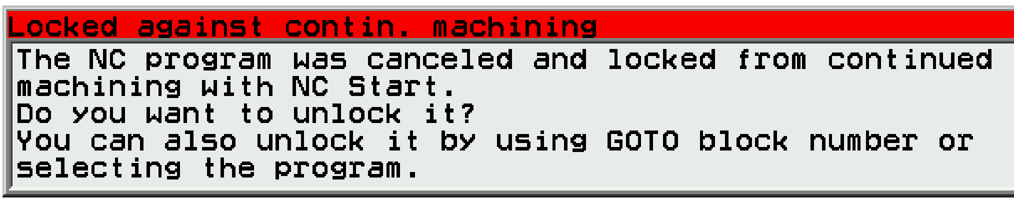
If an NC program is interrupted and subsequently an axis is moved, then before continuing the NC program the **RESTORE POSITION** function must be executed in order to return the axes to the positions current after the last processed NC block. This function is automatically activated by the TNC.

- Set MP7680 bit 13 so that the **RESTORE POSITION** function will **not** be activated automatically.

If the spindle is stopped by an NC stop and an internal stop, and does not restart automatically with the NC start, you can force a mid-program startup:

- Set MP7680 bit 14 to permit an NC start only with mid-program startup or a GOTO command

A message window can be displayed automatically if NC start is pressed after program cancellation.



The machine operator must then decide whether an NC start is permissible:

- Set MP7680 bit 15 to permit an NC start only after acknowledging a message, renewed program selection or a GOTO command.

MP7680 **Machine parameter with multiple function**

Input: Bit 13 – Behavior during program interruption with axis movement
0: Automatic activation of RESTORE POSITION
1: Do not automatically activate RESTORE POSITION
Bit 14 – Behavior of NC start after NC stop and internal stop
0: NC start permitted
1: NC start only permitted after mid-program startup or GOTO
Bit 15 – NC Start if program is aborted
0: NC start permitted
1: NC Start not permitted (message window)

Retract the tool automatically from the contour

Through the **LIFTOFF** column of the tool table, a function can be activated that retracts the tool from the workpiece by 0.1 mm in tool-axis direction after an NC stop.

- Enable the LIFTOFF function via M4620.
- The function is also activated in the NC program with M148. M149 deactivates M148.

		Set	Reset
M4620	**Activate LIFTOFF function**	PLC	NC/PLC

8.1.11 Adaptive feed control (AFC)

With adaptive feed control (AFC, software option #5), the contouring feed rate is regulated depending on the respective spindle power in percent. This is done with the help of the feed-rate override factor, which is normally determined in the PLC by the setting of the override potentiometer. If AFC is active, then this factor is no longer formed from the potentiometer, but rather from the spindle power and other process data, and the contouring feed rate is calculated. The iTNC then automatically changes the value of the feed-rate override factor, with the goal of maintaining the spindle power at a constant reference value.
The main benefit of this process control, along with ensuring the quality of the machining procedure (e.g. detection of cutter breakage or wear), is the optimization of the machining time, which is intended to ensure or improve productivity and efficiency. Changes in the material (harder sections), deviations in oversizes and tool wear lead to the danger of overloading the spindle. AFC can be effective in counteracting this.

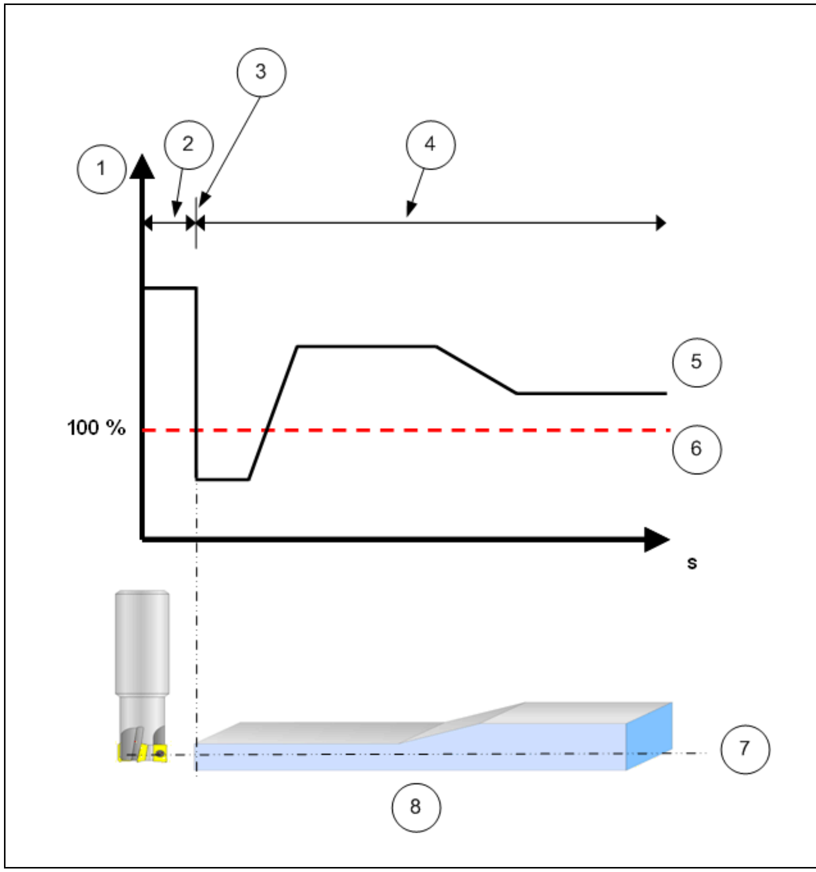
Benefits of adaptive feed control:

- Optimizing the machining time:
  By regulating the feed rate, the attempt is made to maintain the maximum spindle power learned over the entire machining time.
  The total machining time is reduced by increasing the feed rate in machining zones where less material is being removed.
- Tool monitoring:
  The tool life of a tool is increased by reducing the feed rate when the maximum spindle power learned is exceeded, until the time when this reference spindle power is reached again.
  If it is detected that the maximum spindle power was exceeded and at the same time the minimum feed rate could not be maintained, a programmed switch-off reaction is performed. This avoids subsequent damages due to cutter breakage or wear.
- Protection of the machine's mechanical elements
  Damages to the machine due to overload can be avoided by reducing the feed rate ahead of time and with appropriate switch-off reactions.
- Documentation by capturing and saving the learning and process data.

Limitations regarding the use of AFC:

- AFC cannot be used with analog spindles.
- AFC cannot be used in volts-per-hertz control mode.

The following graphic schematically shows the principle of adaptive feed control:



- 1: Feed rate
- 2: Air cut
- 3: Beginning of machining
- 4: Feed-rate calculation during machining
- 5: Feed rate calculated by AFC
- 6: Feed rate programmed in the NC program
- 7: Cutting depth
- 8: Workpiece

Operating states of AFC

The two main operating states of adaptive feed control are **Learning** and **Controlling and Monitoring.**

General information about Learning

A learning phase must be completed before AFC can be activated. This learning phase serves to ascertain the corresponding reference power of the spindle **PREF** for each individual machining step. The learning phase is divided into one or more learning sections. The beginning of a learning section or machining section is defined by the output of M function M03 or M04 (Spindle on right/left) in an NC program after a **TOOL CALL** has been processed. The end of the cut is defined by the spindle stop (M function M05). The NC macros M3.H, M4.H and M5.H must be assigned to these spindle M functions in the **MFUNCT.TAB** table.
At the beginning of cutting with M3 or M4, the FN17 block:
FN 17:SYSWRITE ID 622 NRO IDX 0.0 =+1 must be programmed in the M function macro for starting a cut after the M function.
At the end of cutting with M5, the FN17 block
FN 17:SYSWRITE ID 622 NRO IDX 0.0 =+0 for ending a cut must be programmed before M5.
This way AFC can be used without having to change the existing NC program. The only restriction is that no M-function macros can be executed with active radius compensation. Before calling the macros, the radius compensation must be deactivated, and reactivated after the macro if necessary.
The beginning of another section can be defined for the same tool with another call of M03 or M04. In order to exclude a tool from AFC (e.g. touch probe, tap, …), the AFC column must be undefined (empty) in the **TOOL.T** tool table. Otherwise in this column you must enter the identifier from the global control parameter table **AFC.TAB.** This identifier selects the control parameter block from the table **AFC.TAB.**
If a cutting data table **.CDT** exists and is used, the basic control setting in the AFC column of this table is read. This means that the entry in the cutting data table has precedence over the entry in the tool table. The data ascertained during each learning section of the reference machining, especially the maximum spindle power, are saved with a generated cutting number in a table. They form the foundation for the subsequent controlled machining with active AFC.
As an alternative to the generated cutting number, the value in NR can be used as the cutting number by programming discrete FN17 cutting blocks. However, the programmer must then ensure that the numbers are entered correctly and sequentially. The learning phase can consist of an entire machining process with multiple tools, but it also can also be stopped after a certain time in which the important parameters for the respective tools have been determined. The learning phase can be ended either manually via a soft key or automatically after a programmed time for the machining step has expired.

With **FN 17: SYSWRITE ID 622 NRO IDX1.0 = <value>** you can transfer a **<value>** for the time in seconds after which the teach-in cuts are ended automatically. The function behaves like the pressing of the **EXIT LEARNING** soft key after the appropriate time. This function is deactivated again by programming **<value>** = 0.
At any time during the teach-in cut, you can change the machining feed rate with the feed-rate override potentiometer any way you want, and so influence the ascertained reference load. The file generated in this step is given the name of the selected NC program and receives **.AFC.DEP** as the file extension. **.DEP** is appended because this file depends on an existing **.H** file. E.g. an NC program with the name **MACHINING.H** results in a file with the name **MACHINING.H.AFC.DEP.** This file, and the global parameter table **AFC.TAB** as well, are set up as freely definable tables, and form view is also possible (switch via soft key). However, the format is prescribed and cannot be changed.

Details of the learning phase:

- The spindle power is only recorded once the nominal speed has been reached, since the spindle is operated at a constant speed, and acceleration and braking phases are not included in the machining.
- Saving of the spindle power in air (idle power) in order to later evaluate only the power used during the actual machining process.
- Detection of entry into the material by the cutter when the idle load is exceeded by 2%.
- Saving of the maximum value of the spindle power of a learning section as reference for the control.
- Saving of the machining time of each learning section.
- Monitoring for stationary spindle or deviation from the nominal speed during active feed not at FMAX.

Note

- Below a certain minimum tool diameter (e.g. 5 mm), in some circumstances it may not make sense to use AFC, since the actual data may become too small for safe control. This can be specified in the tool or cutting data table by leaving the AFC column empty. The entry in the cutting data table makes it possible for the basic setting to depend on the pairing of the tool and workpiece materials.

- If it becomes necessary in other cases for AFC to abort a machining step, there are two possibilities:

  - By leaving the AFC column empty in the tool or cutting-data table.
  - By deactivating AFC via targeted switch-off with FN 17:SYSWRITE ID 622 NR0 IDX 0.0 =+0 in the NC program or a corresponding M-function macro. If you use the M-function macros for M3 and M4, then the FN 17 block must follow the call of the M-function macro. This way you end the cut for AFC before machining actually takes place.

If the learning process is ended during a learning step, either via the **EXIT LEARNING** soft key or because the programmed time (FN17 - ID622) has expired, the control immediately begins regulating the feed rate adaptively. The value determined up to that point is used as the spindle reference value, and is entered in **<NAME>.H.AFC.DEP.** This is useful if it can be seen that the spindle power will not increase significantly any further in this step.

General information about Controlling

The maximum spindle power was determined for each learning and machining step. By changing the feed-rate factor within programmable limits (e.g. 70% to 130%), the software continuously attempts to match the momentary spindle power to the reference value during machining.

This way the feed rate can be increased for areas where only a small amount of material is being cleared, and a slower feed rate can be used when there is more material than usual or the cutter is becoming dull. This protects the spindle drive and the tool from overload. If a disturbance in the process occurs during machining, a programmable switch-off reaction is performed. This occurs if the feed rate falls below the minimum feed-rate factor (e.g. 70%) at the reference spindle load. You can then assume that the cutter has become dull or has broken. If the feed rate falls below the defined value of 30% of the programmed feed rate, an NC stop is performed. This defined error reaction cannot be influenced, and occurs independently of the programmed overload reaction.
Controlling can be switched off and on again via soft key at any point during machining. Controlling is also always switched off if the value set for the override potentiometer is manually reduced by more than 10%. The potentiometer is then effective again instead of AFC. Adaptive feed control must then be switched on again via soft key. A potentiometer value less than or equal to 50% is also always effective, i.e. AFC is inactive then. If the potentiometer value exceeds these 50% while AFC is on, then AFC becomes active again. The maximum spindle power that occurs while under control is ascertained and saved for evaluation later. The machining time of each machining section is also saved.
The events AFC ON, AFC OFF, potentiometer manually changed -> AFC OFF are entered in the log of the iTNC.

Tasks of AFC in each machining section:

- Detection of entry into the material by the cutter when the idle load is exceeded by 2%. Regulate to entry speed until a path corresponding to the cutter radius has been cleared, but at most 600 ms.
- Calculation of the optimum feed-rate factor.
- Monitoring for overload and switch-off reaction.
- Saving the maximum value of the spindle power for the evaluation.
- Detection of exit from the material by the cutter when the power is below the programmed exit load, and regulation to the exit speed until a path corresponding to the cutter radius has been traversed.
- After detection of traverse in air, regulate to the programmed idle feed rate.
- AFC is not in effect when machining at rapid traverse, since FMAX is used in the IPO, and at times override for FMAX.

As an alternative method for controlling with AFC, you can use the value in PLC word W632 [value in tenths of percent] as control input variable, instead of using the spindle power in percent. A spindle power of 50% corresponds to the value 500 to be entered in W632 (0.1% corresponds to the value 1). This may be necessary for individual applications (e. g. pecking).
The control input variable in W632 must be calculated in the PLC program, which can best be done with a fast PLC cycle (e.g. 15 ms). Also, when word W632 is used, no idle load for the spindle is determined and no transitional period for the tool entering or leaving the workpiece is considered.
This function must be activated by entering **`AFC.PLCCONTROL = ON`** in the OEM.SYS.
However, the function only works if a value is entered in W632.

Automatic selection of the Learning / Controlling status

If an NC program is selected for machining and started, and if AFC is active, then it is first checked whether an associated **<name>.AFC.DEP** file exists in the same directory as the NC file. If this is the case, then the data in this file are immediately used as control parameters, and machining is performed with adaptive feed control. If this file does not exist, the first machining run is used as the learning phase and the file is created. If the AFC file is determined to be incomplete during machining, the missing steps are automatically performed in the Learning mode, thereby generating the missing data.
When Learning is deactivated (M05, manually via soft key, or with FN 17), the learned data are saved and the status of this data block is set to "Controlling." During Controlling, the data blocks are read in the same sequence they were learned, the step numbers are evaluated and other parameters (tool number, index) are checked to see if they match.

The current status or mode of AFC is stored in PLC Word W348 (inactive = 0, learning phase = 1, controlling = 2).

- W348 = 0: AFC inactive (OFF)
- W348 = 1: AFC in learning phase
- W348 = 2: AFC is controlling

Additional FN functions

Further FN 17 functions are available for software option #45 "AFC - Adaptive Feed Control":

- Definition of reference power. The learning phase can therefore be omitted. The reference power is specified in percent with respect to the rated power of the spindle. Example:
  FN 17:SYSWRITE ID 622 NR0 IDX 3.0 = 85
- Activating cut-based tool-wear monitoring.
  The monitoring limit is specified in percent of the reference power. If the spindle power exceeds the monitoring limit, the PLC marker M4510 will be set. In this way, the PLC can initiate the desired reactions, such as setting the tool life of a tool to "expired". Example:
  Monitoring limit at 20% with a reference power of 50%. In this way, 60% of the rated power output of the spindle is monitored:
  FN 17:SYSWRITE ID 622 NR0 IDX 4.0 = 20
- Activating cut-based spindle-load monitoring.
  The monitoring limit is specified in percent of the reference power. If the spindle power exceeds the monitoring limit, an NC stop is triggered.
  Monitoring limit at 40% with a reference power of 50%. In this way, the system monitors for 70% of the rated power output of the spindle:
  FN 17:SYSWRITE ID 622 NR0 IDX 5.0 = 20

  The PLC can read in word W350 why the NC stop was triggered:
  - (W350 = 1 or 2:
    NC stop due to falling below the permissible minimum power
  - W350 = 3:
    NC stop due to detected cutter breakage (FN17 function ID 622 IDX 6.0)
  - W350 = 4:
    NC stop because the spindle load was exceeded (FN17 function ID 622 IDX 5.0)

- **FN17 – ID 622 IDX 6.0**
  The function **FN 17: SYSWRITE ID 622 NR<> IDX 6.0** is used to activate spindle load monitoring independently of the cut in order to detect cutter breakage. The monitoring limit is specified in percent with respect to the rated power of the spindle. If the value exceeds or falls below the monitoring limit, an NC stop is triggered.
  A filter constant must be specified in parameter **NR<>** in order to avoid an NC stop when the tool leaves the material. Only abrupt changes in spindle power that are typical of tool breakage may be evaluated. The filter constant can be between 1 and 60 ms. If a constant of 0 is entered, the default value 10 ms will be used.
  The values for the filter constant and the tolerance limit must be determined for each individual application. Example:
  Filter constant of 15 ms, tolerance limit for cutter breakage at 10% of the rated power output of the spindle:
  FN 17:SYSWRITE ID 622 NR15 IDX 6.0 = 10

  The PLC can read in word W350 why the NC stop was triggered:
  - (W350 = 1 or 2:
    NC stop due to falling below the permissible minimum power
  - W350 = 3:
    NC stop due to detected cutter breakage (FN17 function ID 622 IDX 6.0)
  - W350 = 4:
    NC stop because the spindle load was exceeded (FN17 function ID 622 IDX 5.0)

> Note
>
> Due to the difficult conditions surrounding setting the values for **FN 17: SYSWRITE ID 622 NR<> IDX 6.0**, you should only use this function after consultation with HEIDENHAIN.

Associated files

Files for machining

The following requirements are mandatory for controlling to be activated:

- An NC program must have been selected **(<name>.H).**
- The column AFC must be enabled in the tool table via MP7266.40.
- If you are working with a **<name>.CDT** cutting table, then you must create an AFC column and enter any desired control strategy there. If an entry in the cutting data table intersects with an entry in the tool table, the entry in the cutting data table has priority.
- M-function macro files M3.H, M4.H, M5.H (and possibly more).
- **MFUNCT.TAB** macro definition.

The three files of adaptive feed control:

File of the global control parameters: **AFC.TAB**
In the **AFC.TAB** table, which must be saved in the **TNC:** root directory, you enter the basic control settings with which the TNC is to perform the feed-rate control.
The data in this table are default values that are copied during the teach-in cut to the file associated with the respective machining program, where they serve as the basis for controlling.
Control parameters for the subsequent standard control strategy are already saved there. Other settings can be programmed as desired.
The static control parameters that must be programmed in **AFC.TAB** for each cutting-data set are:

Column	Function
NR	Consecutive line number in the table (has no further functions)
AFC	Name of the control strategy. You must enter this name in the AFC column of the tool table. It specifies the assignment of control parameters to the tool.
FMIN	Minimum feed rate at which the TNC is to perform an overload reaction if the feed rate falls below this value for one second. Enter the value in percent of the programmed feed rate.
FMAX	Maximum feed rate in the material up to which the TNC can automatically increase the feed rate. Enter the value in percent of the programmed feed rate.
FIDL	Feed rate for traverse when the tool is not cutting (feed rate in the air). Enter the value in percent of the programmed feed rate.
FENT	Feed rate for traverse when the tool moves into or out of the material. Enter the value in percent of the programmed feed rate.
OVLD	Desired reaction of the iTNC to overload: ■ M: Perform an NC-function macro defined by the OEM ■ S: Immediately perform an NC stop ■ F: Perform an NC stop once the tool has retracted ■ E: Only display an error message on the screen ■ –: Do not perform an overload reaction The iTNC conducts a shutdown response if the maximum spindle power is exceeded for more than one second and at the same time the feed rate falls below the minimum you defined.
POUT	Spindle power at which the TNC is to detect tool exit from the workpiece. Enter the value in percent of the learned reference load. Recommended input value: 8%

Column	Function
SENS	Sensitivity (aggressiveness) of regulation. A value between 50 and 200 can be entered. 50 is for slow control, 200 for a very aggressive control. An aggressive control reacts quickly and with strong changes to the values, but it tends to overshoot. Recommended value: 100
PLC	The value entered here is written to word W342 at the beginning of a cut. The module cannot be called in the cyclic PLC program. Depending on this, any necessary fine adjustments to the adaptive feed control can be performed from the PLC program. Maximum input range: 3 digits.

Example of an **AFC.TAB**:

NR	AFC	FMIN	FMAX	FIDL	FENT	OVLD	POUT	SENS	PLC
0	Standard	70	125	100	80	-	5	100	0

Proceed as follows to create the AFC.TAB file (only necessary if the file does not yet exist):

- Select the **Programming and Editing** operating mode.
- Press the PGM MGT soft key to call the file manager
- Select the **TNC:** directory
- Create the new **AFC.TAB** file and confirm with the ENT key. The TNC displays a list with table formats.
- Select the **AFC.TAB** table format and confirm with the ENT key. The TNC creates the table with the **Standard** control strategy.

Settings file with the control parameters: **<name>.H.AFC.DEP**
This file is automatically created during the learning phase, and filled with values. It contains all control information for the cuts that occur in the NC program. You can manually change all entries, except TNR and IDX, at any time in order to adapt the control parameters specifically to the NC program. Call the **<name>.H.AFC.DEP** via the **SETTINGS TABLE** soft key. It contains the following additional information:

Column	Function
NR	Number of the machining step
TOOL	Number or name of the tool with which the machining step was made (not editable)
IDX	Index of the tool with which the machining step was made (not editable)
PREF	Reference load of the spindle. The TNC measures the value in percent with respect to the rated power of the spindle.
N	Code whether a tool number (0) or tool name (1) was programmed in the NC program.
POUT	Exit load of the spindle. The TNC uses the value as a percent of the reference load **PREF** of the spindle.
ST	Status of the machining step ■ L: (= Learning) In the next program run, a teach-in cut is recorded for this machining step. The iTNC overwrites any existing values in this line. ■ C: (= Controlling) The teach-in cut was performed successfully. The next time it is machined, AFC will control the feed rate.
AFC	Name of the control strategy used. If the control strategy given in the tool table is not in AFC.TAB, then the entry **default** is in this column. Default settings are then used for adaptive feed control.

Example of a **<name>.H.AFC.DEP** file after a successful teach-in cut:

NR	TOOL	IDX	FMIN	FMAX	FIDL	FENT	OVLD	POUT	PREF	SENS	ST	PLC	AFC
0	2	0	60	140	150	70	-	5	64	100	C	0	Standard

Proceed as follows to select (and edit) the **<name>.H.AFC.DEP** file.

- Select the **Program Run, Full Sequence** operating mode.
- Shift the soft-key row
- Select the table of AFC settings with the **SETTINGS TABLE** soft key
- If necessary, enter any improvements. Edited entries are identified by an * before the name of the control strategy.

The DELETE LINE soft key is also available in the Programming and Editing mode of operation. However, only use this function if you specifically want to delete an incorrectly created line from the <NAME>.H.AFC.DEP table for control settings.
This may be reasonable when lines were created in the table, which would not have been created during normal program run. Such lines might, for example, be produced when program run is canceled, or by a GOTO command to a previous beginning of a cut. You must confirm deletion in a confirmation request, because deleting a line moves up the following lines with their control parameters.

Note

Please note that the control parameters will be assigned to the wrong cuts if you delete lines incorrectly.
If you are not sure which line may be deleted, then perform the learning phase anew for the complete NC program.

Protocol file with the actual data of the control: **<name>.H.AFC2.DEP**
The iTNC stores various pieces of information for each machining step of a teach-in cut in this file. During control, the iTNC updates the data and makes various evaluations.
Call the **<name>.H.AFC2.DEP** file via the **EVALUATE TABLE** soft key.

Column	Function
NR	Number of the machining step
TOOL	Number of the tool with which the machining section was performed (cannot be edited)
IDX	Index of the tool with which the machining step was made (not editable)
SNOM	Nominal spindle speed [rpm]
SDIF	Maximum difference of the spindle speed in % of the nominal speed
LTIME	Machining time for the teach-in cut
CTIME	Machining time for the control cut
TDIFF	Time difference in % between the machining time during teach-in and control
PMAX	Maximum recorded spindle power during machining. The TNC shows the value as a percent of the spindle’s rated power.
PREF	Reference load of the spindle. The TNC shows the value as a percent of the spindle’s rated power.

Column	Function
OVLD	Reaction by the iTNC to overload: ■ M: Perform an NC-function macro defined by the OEM ■ S: Immediately perform an NC stop ■ F: Perform an NC stop once the tool has retracted ■ E: Only display an error message on the screen ■ –: Do not perform an overload reaction The iTNC conducts a shutdown response if the maximum spindle power is exceeded for more than one second and at the same time the feed rate falls below the minimum you defined.
BLOCK	Block number at which the machining cut begins.

Example of a **<name>.H.AFC2.DEP** file after a cut with active AFC:

NR	TOOL	IDX	SNOM	SDIFF	LTIME	CTIME	TDIFF	PMAX	PREF	OVLD
0	2	0	1000	3.5	00:00:40	00:00:32	-20.0	70.2	64	-

Proceed as follows to select the **<name>.H.AFC2.DEP** file.

- Select the **Program Run, Full Sequence** operating mode
- Shift the soft-key row.
- Select the table of AFC settings with the **SETTINGS TABLE** soft key
- Select the protocol file with the **EVALUATE TABLE** soft key

NC-function macro at overload

The entry **M** in the **OVLD** column of **AFC.TAB** can be used to call an NC-function macro for each control strategy as a reaction to an overload. The AFC detects that the spindle load has been exceeded if the actual feed rate is below the value in **FMIN** for more than one second. No more than one minute after the overload has been detected is the NC macro performed. The path for the NC macro must be entered in the **PLC:\ncmacro.sys** file via the keyword AFC as follows:

AFC = <path><ncmacro>.H
e.g: **AFC = PLC:\NC_MACRO\AFC_STOP.H**

If this entry is not in the ncmacro.sys file or the NC macro does not exist, then the NC program is stopped with the corresponding error message. The FN17 function SYSWRITE ID990 NR11can also be used for the AFC NC macro. It makes it possible to perform a tool change with subsequent positioning. This way replacement tool can be inserted in a tool-change procedure as part of an AFC NC macro. Use the following NC macro for this:

```
1  ;Read current tool number: Q1
2  FN 18: SYSREAD Q1 = ID20 NR1
3  ;Read maximum tool life of the tool (at TOOL CALL): Q2
4  FN 18: SYSREAD Q2 = ID50 NR10 IDXQ1
5  ;Set current tool life of the tool to maximum tool life
6  FN 17: SYSWRITE ID50 NR11 IDXQ1 = Q2
7  ;Insert the replacement tool using repositioning logic
8  FN 17: SYSWRITE ID990 NR11 = 0
```

The FN function FN 17: SYSWRITE ID990 NR11 cannot be used with active radius compensation if the tool change is performed in a change macro. This new FN function works with a tool change only with PLC positioning.

As an alternative to block number 8 in the macro **FN 17: SYSWRITE ID990 NR11 = 0**, as of software version 340 49x-05 an automatic tool call with M101 (on) and M102 (off) can be performed at defined points in the NC program. For this, you must delete block 8 from the macro and enter M101/102 at the possible tool change points in the NC program. If the machining run-time between M101/102 pairs is shorter than one minute, it is ensured that a possible tool change will be performed one minute after the overload message M has occurred.
Otherwise, the time until the tool change will be extended. Depending on the duration of the longest individually occurring NC block, the change time may always be extended by the duration of this block.
Example:
An NC block with a duration of 40 seconds, and five blocks, each with a duration of two seconds, are executed in a loop.
The pre-calculated times (limited to a machining time of one minute) result in: 40 + 2 + 2 + 2 + 2 + 2 = 50 seconds. Because the result is < 1 minute, the long block of the next loop is added. The result is 50 + 40 = 90 seconds of pre-calculation. The tool change macro can be inserted at this point, if required.
If the machining process is at the beginning of the long block when the overload occurs, the remaining time of the long block will also be added. This can result in a time of 90 seconds + remaining time, e.g. 35 seconds = 125 seconds plus a further delay through feed rate reduction by AFC (e.g. 80% = 125 * 1.25 = 156 seconds) from the occurrence of the overload to the execution of the tool change macro.

Default settings for using AFC

Adaptive feed control is a software option (option #45), and must therefore be enabled separately for each machine. The following settings must then be made on the control:

- Create the **TNC:\AFC.TAB** file, and define one or more control strategies.
- Set MP7246 bit 3 so that the <NC name>.H.AFC.DEP table for control settings can be generated for adaptive feed control. A separate file is created for each NC program in the learning phase.
- Use MP7266.40 to enable the **AFC** column in the tool table, or to create the **AFC** column in a ***.CDT** cutting data table.
- Enter the name of the control strategy appropriate for each tool in the **AFC** column of the tool table or cutting data table.
  If you do not enter a control strategy for the tool used in the NC program, AFC will not be in effect for this tool.
  If you are in the **AFC** column in the tool table, you can use the SELECT AFC CONTROL SETTING soft key to select a control strategy from **AFC.TAB.**
- Set the macro execution for the M functions M3, M4, M5 and any other M functions in the **PLC:\MFUNCT\MFUNCT.TAB** table.
- Create the M-function macros M3.H, M4.H, M5.H and any others in the **PLC:\MFUNCT** directory. In order to edit **FN17,** the appropriate code number (555 343) must first be entered.

Note

The M-function macros cannot be called if radius compensation is active in the NC program. Before calling the macros, the radius compensation must be deactivated, and reactivated after the macro if necessary.

M-function macros with which M3 or M4 automatically define the beginning of a cut for AFC, and M5 defines the end of a cut:

```
0  BEGIN PGM M3 MM
1  M3
2  FN 17:SYSWRITE ID 622 NR0 IDX 0.0 =+1;after M3
3  END PGM M3 MM

0  BEGIN PGM M4 MM
1  M4
2  FN 17:SYSWRITE ID 622 NR0 IDX 0.0 =+1;after M4
3  END PGM M4 MM

0  BEGIN PGM M5 MM
1  FN 17:SYSWRITE ID 622 NR0 IDX 0.0 =+0;before M5
2  M5
3  END PGM M5 MM
```

Note

The M-function macros for M3, M4 and M5 (as listed above) are used to write the control settings for the various cuts sequentially, in the same order as the cuts in the NC program, into the **<NC name>.H.AFC.DEP** file. You cannot use these M-function macros to assign a permanent number to the individual cutting data blocks.
This means that you cannot use the **Program Run, Single Block** operating mode to perform a teach-in cut for just a single cut.
You must always run the entire NC program so that AFC can update the correct lines in **<NC name>.H.AFC.DEP.**

However, the FN 17:SYSWRITE ID 622 function offers you the possibility of numbering all the cuts of an NC program, making a permanent assignment possible. This is considered in the **Program Run, Single Block** operating mode, and you can retroactively carry out the learning process for individual cuts.

You must deactivate the execution of the given M-function macros for this. The definition of the beginning and end of a cut via the FN 17:SYSWRITE ID 622 function must then occur separately for each cut in the NC program.

Note

You cannot use just any numbers when numbering the cuts in the NC program.
The beginning of the first cut must be started after M3 or M4 with FN 17:SYSWRITE ID 622 NR0 IDX 0.0 =+1 and be ended before M5 with FN 17:SYSWRITE ID 622 NR0 IDX 0.0 =+0.
NR must be incremented for each further cut. Targeted access to individual cutting data blocks is only possible with sequential numbering.

MP7246	**Machine parameter with multiple function**
Input:	Bit 3 – Settings file for AFC 0: Do not generate settings file for AFC 1: Generate settings file for AFC

Initial learning and controlling with AFC

After you have entered the settings described above, proceed as follows to carry out a learning phase for an NC program:

- In the **Program Run, Full Sequence** operating mode, load the NC program for which the learning procedure is to be carried out, e.g. **<NC name.H.>**
- Activate AFC with the **ADAPTIVE FEED CONTROL** and **FEED CONTROL ON** soft keys. This creates and displays the **<NC name.H.AFC.DEP>** table of control settings.
- Start the NC program.
- The learning procedure is performed automatically for each individual machining cut.
- The spindle reference power determined with the teach-in cut until this time is displayed in a pop-up window.

Note

You do not have to run the entire machining step in the learning mode. If the cutting conditions do not change significantly, you can switch to the control mode immediately. Press the **EXIT LEARNING** soft key, and the status changes from **L** to **C.**

At any time during the teach-in cut, you can change the machining feed rate with the feed-rate override potentiometer any way you want, and so influence the ascertained reference load.

You can repeat a teach-in cut as often as desired. Manually change the status from **ST** back to **L.** It may be necessary to repeat the teach-in cut if the programmed feed rate is far too fast, and forces you to sharply decrease the feed rate override during the machining step.

The PREF RESET soft key enables you to reset the spindle reference power determined by the teach-in cut until this moment, and to restart reference-value determination.

You can teach any number of machining steps for a tool. A machining step always starts with **M3** or **M4,** and ends with **M5.** For machining steps in which the tool remains the same, but it appears useful to define multiple steps for AFC, you can program another **M3** or **M4.** Each **M3** or **M4** defines the beginning of a new cut.

- All control settings for the individual cuts of an NC program are saved as a table in the **<NC name>.H.AFC.DEP** file (**SETTINGS TABLE** soft key), where they are used for future machining.
  The status of each cutting data block changes from L (learning) to C (controlling) after a successful teach-in cut.
- The data determined in the learning phase is saved in the **<NC name>.H.AFC2.DEP** file (**EVALUATION TABLE** soft key).

Once the NC program has been run, the learning procedure is finished for adaptive feed control. The status in the **<NC name>.H.AFC.DEP** file (**SETTINGS TABLE** soft key) must have switched from L (learning) to C (controlling) for each cut. This way a valid settings file exists for adaptive feed control. As long as the status is not set to L (learning) and adaptive feed control has been activated with the **AFC ON** soft key, the NC program is run with AFC.

- If you start the NC program again, the feed-rate override is changed by the adaptive feed control. The control tries to maintain the spindle power as constantly as possible at the reference value **PREF** of the spindle power.
- The data in the **<NC name>.H.AFC2.DEP** file (**EVALUATION TABLE** soft key) determined in the learning phase is compared with the values from machining with AFC.

Activating/ deactivating AFC

Proceed as follows to activate or deactivate adaptive feed control:

- Select the **Program Run, Full Sequence** operating mode.
- Shift the soft-key row.
- Activate adaptive feed control: Set the **AFC** soft key to **ON**.
- Deactivate adaptive feed control: Set the **AFC** soft key to **OFF**.

The setting of the AFC OFF/ON soft key is saved, and even if the control is reset, an emergency stop occurs, or the machine is switched off via the main switch, the momentary setting of the soft key is stored.
The setting of the AFC OFF/ON soft key is also mapped in PLC word W336.

- W336 = 0: Soft key is set to OFF
- W336 = 1: Soft key is set to ON

Note

If adaptive feed control is active, the TNC assumes the functions of the feed-rate override. If you reduce the feed-rate override by more than **10%** of the current setting, then the TNC switches AFC off. In this case the TNC displays a window, which you can acknowledge with the CE key.

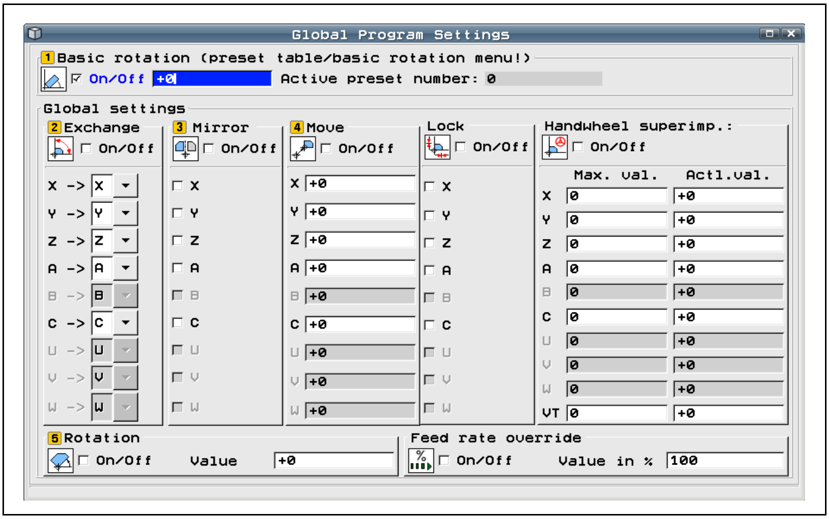
If you set the feed-rate override to less than **50%,** the value of the override potentiometer takes effect. However, AFC is not automatically switched off here. If the value of the override potentiometer later exceeds the 50% again, AFC assumes control again.

The TNC shows various pieces of information in the additional status display when adaptive feed control is active.

8.1.12 Global program settings (PGM)

Global PGM Settings (software option #44) offers you the possibility of superimposing various coordinate transformations and settings in the program-run modes of operation.

For a detailed description of the individual functions, see the iTNC 530 User's Manual.



Attention

The following must be observed when using the **global PGM settings:**

- Basically, all settings in the Global PGM Settings remain active when executing cycles and NC programs. However, when executing macros or PLC positioning movements, all Global PGM Settings are automatically deactivated.
- If you disable one or more axes via Global PGM Settings, then this disabling is only canceled within a macro or during the PLC positioning. However, if you transfer axis position values to the PLC (e.g. with FN19) within a cycle or NC program and then perform a PLC positioning, then this positioning really is performed with all axes. However, if the axis position values transferred from a cycle or an NC program are not approached, due to the Global PGM Settings, then this can lead to a crash during the subsequent PLC positioning.
- M118 cannot be deselected in cycles if handwheel superimpositioning has been activated in the Global PGM Settings. However, handwheel superimpositioning is automatically deactivated in the Global PGM Settings when executing a macro.
- If you read the position values of an axis via the PLC, then when you interrogate the nominal values, the nominal value programmed in an NC program is returned, and not the actual value of the axis. If further actions derive from the nominal value, then this can lead to problems, since these value do not necessarily reflect the actual position values of the axes. If you interrogate the actual value of an axis, then the actual position is returned.

8.1.13 Cycles

HEIDENHAIN contouring controls feature standard fixed cycles (e.g. peck drilling, tapping, pocket milling), which can be called in the NC program. In addition to the standard HEIDENHAIN cycles, you can program so-called Original Equipment Manufacturer (OEM) cycles (see the Cycle Design User's Manual). You can influence the function of many HEIDENHAIN standard cycles through machine parameters.

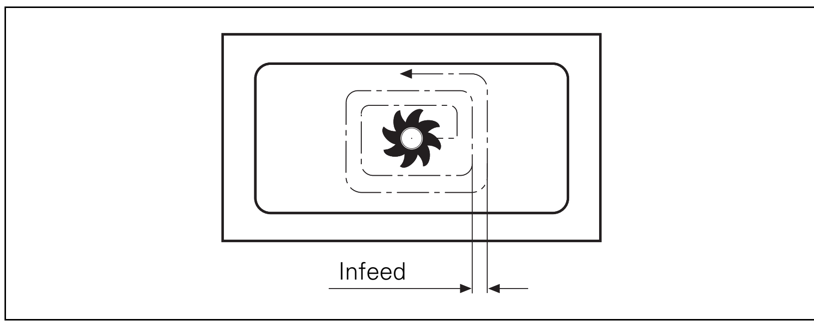
For more information on the tapping cycle and the oriented spindle stop cycle, see page 941.

See page 1481 for more information on the touch probe cycles.

Pocket milling

Cycles 4 and 5:

▶ In MP7430, enter the overlap factor for roughing out a rectangular or circular pocket.



Infeed = (MP7430) · cutter radius

MP7430 is only effective for Cycles 4 and 5.

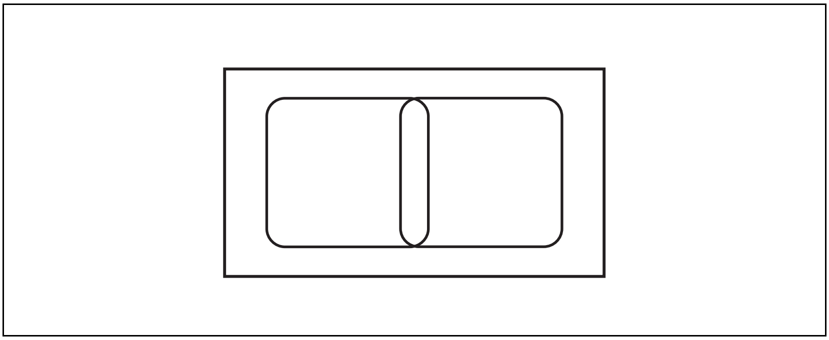
MP7430	**Overlap factor for pocket milling**
Input:	0.001 to 1.414

Cycles for milling pockets with combined contours

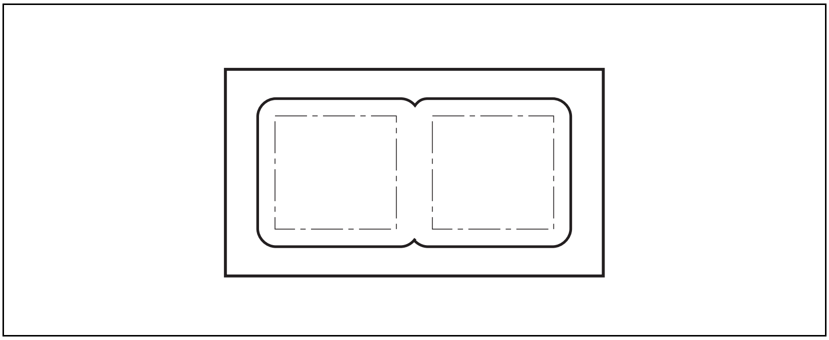
Cycles 6, 14, 15, 16:

▶ With MP7420, specify:

- Bit 0: The milling direction for channel milling
- Bit 1: The sequence for rough-out and channel milling
- Bit 2: The conditions under which programmed pockets should be merged (see graphics below)
- Bit 3: Whether each process (channel milling or pocket clearing) is to be completed for all pecking depths before performing the other process, or whether both are to be performed alternately for each pecking depth
- Bit 4: Position after completion of the cycle

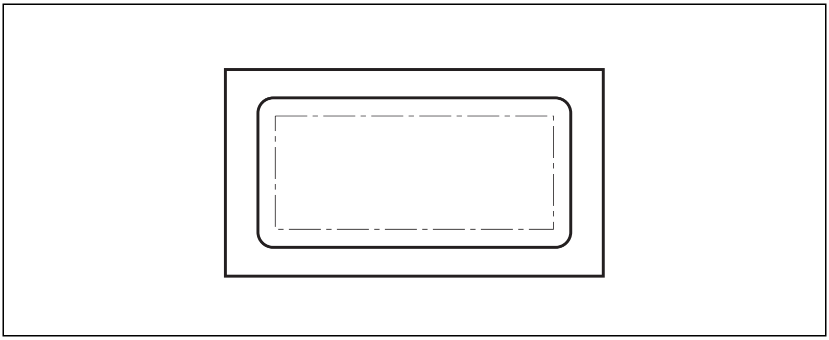


The programmed contours of two pockets intersect slightly.



MP7420 bit 2 = 0:

The control clears the pocket separately because the paths of the tool center do not intersect. Material will remain at inside corners.



MP7420 bit 2 = 1:

The control clears the pockets together because the programmed contours intersect. No material will remain at inside corners.

MP7420 **Cycles for milling pockets with combined contours**
Format: %xxxxx
Input: Bit 0 – Milling direction for channel milling
0: Counterclockwise for pockets, clockwise for islands
1: Clockwise for pockets, counterclockwise for islands
Bit 1 – Sequence for rough-out and channel milling (only for SL 1):
0: First channel milling, then pocket rough-out
1: First pocket rough-out, then channel milling
Bit 2 – Merging of listed contours
0: Contours are merged only if the tool-center paths intersect
1: Contours are merged if the programmed contours intersect
Bit 3 – Rough-out and channel milling to pocket depth or for every infeed
0: Each process uninterrupted to pocket depth
1: Both processes for each pecking depth before proceeding to the next depth
Bit 4 – Position after completion of the cycle
0: Tool moves to the same position as before the cycle was called
1: TNC retracts the axis to the "clearance height"

Scaling factor

Cycle 11:

- In MP4710, enter the effective range of the scaling factor.

MP7410 **Scaling cycle in two or three axes**
Input: 0: Scaling cycle is effective in all three principal axes
1: Scaling cycle is effective only in the working plane

Cylindrical surface

With Cycles 27, 28 and 39 you can machine a contour on a cylindrical surface (see the User's Manual).

In standard approach/departure movements of Cycle 39, the tool approaches/departs the contour tangentially on an arc that depends on the tool radius. If contours are close to each other, the approach/departure movements might damage adjacent contours.

- Define the rotation center of the rotary axis with MP75xx. Also see page 705.
- Define the behavior of Cycle 28 with MP7680 bit 12.
- Define the behavior of Cycle 39 with MP7680 bit 16.

MP7680 **Machine parameter with multiple function**
Input: Bit 12 – Behavior of Cycle 28
0: Standard behavior
1: The slot wall is approached and departed tangentially; at the beginning and end of the slot a rounding arc with a diameter equal to the slot is cut
Bit 16 – Behavior of Cycle 39
0: Standard behavior
1: Approach/departure movement on an arc is not active

8.1.14 End of program run

If the program end is reached in the operating modes **Program Run, Single Block** and **Program Run, Full Sequence,** the NC sets M4170. This marker is reset with the next program start.

You can evaluate the information "Program end" during program run with pallet changers, for example.

		Set	Reset
M4170	**END PGM, M02 or M30 was executed**	NC	NC

8.1.15 Returning to the contour

With HEIDENHAIN contouring controls you can resume an interrupted NC program at a specified block number by scanning the previous blocks (see "Mid-Program Startup" in the User's Manual).

Note

- M functions M142 and M143 are not permitted with mid-program startup.
- If you perform a mid-program startup in a program containing M128, then the TNC performs any compensation movements necessary. The compensation movements are superimposed over the approach movement.

If the program is aborted due to a power outage, or if the iTNC 530 with Windows 2000/XP was shut down in Windows, and **power-fail monitoring is active**, then after the **Power interrupted** message is acknowledged, the **NC program cancelled** message appears. The point of interruption is remembered, and is offered for the mid-program startup.

Note

You must enable these functions through machine parameters.

You must prepare the PLC program accordingly.

Markers inform the PLC about individual conditions during mid-program startup (block scan). Depending on these markers you can enable certain functions such as the axis-direction buttons for **MANUAL TRAVERSE.**

M4156 is set if the **MANUAL TRAVERSE** soft key is pressed.

M4157 is set if the **RESTORE POSITION** soft key is pressed ("Return to Contour").

M4158 is set if the **RESTORE POS. AT** soft key is pressed. M4158 is reset if the **RESTORE POSITION** or **INTERNAL STOP** soft key is pressed.

During the block scan, PLC positioning commands are included in calculation only if they are also executed. The TOOL CALL block normally initiates PLC positioning commands for tool change. If you want these positioning commands to be calculated in the block scan:

- In MP951.x, enter the absolute position with respect to the machine datum. The values for MP951.x can be assumed with the "actual position capture" key.
- Activate the calculation for the specific axes with MP7450.

With flexible tool-pocket coding in the central tool file (see "Tool Changer" on page 1513), the change of pocket number in the tool file must be prevented during block scan if the TOOL CALL blocks are not collected:

- Set M4542.

The block scan can be interrupted by a programmed STOP or with M06, whereby you can have the programmed dwell time included:

- With MP7680, bits 3 and 4, select the parameters for the block scan.
- With MP7451.x, define the feed rate for returning to the contour.

If an NC program block is interrupted in **Single Block mode** or by a **STOP** block and the positions of NC axes are changed, the NC program can be restarted at the changed positions. If in OEM.SYS **STRICTREPOS = YES,** the function for restoring the position is activated (see "OEM.SYS" on page 1637).

Note

If you change the tool data in the PLC, update them with M4538, or change or update them with an NC macro, then the new tool data cannot be correctly offset in the block scan.

MP951.x **Simulated tool-change position for TOOL CALL during mid-program startup (block scan)**
Input: –99 999.9999 to +99 999.9999 [mm] or [°]

MP7450 **Offsetting the tool change position from MP951.x in block scan**
Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Do not offset
1: Offset

MP7451.x **Feed rate for returning to the contour**
Input: 10 to 1 000 000 [mm/min]

MP7680 **Machine parameter with multiple function**
Input: Bit 1 – Returning to the contour
0: Not active
1: Active
Bit 2 – Block scan
0: Not active
1: Active
Bit 3 – Interruption of block scan for STOP or M06
0: Interruption
1: No interruption
Bit 4 – Inclusion of programmed dwell time during the block scan
0: Include the dwell time
1: Do not include the dwell time
Bit 5 – Start of calculation for block scan
0: Start from block with cursor
1: Start from beginning of program

		Set	Reset
M4156	**MANUAL TRAVERSE soft key pressed**	NC	NC
M4157	**Return to the contour (RESTORE POSITION) is active**	NC	NC
M4158	**Block scan active**	NC	NC
M4538	**Geometry of the tool from W264**	PLC	NC
M4542	**Do not update pocket number in the pocket table**	PLC	PLC

M/S/T/Q transfer during block scan

The PLC can collect the M/S/T/Q signals during the block scan in order to output them after the block scan ends:

- Set MP7681 to a value other than zero so that after a block scan the message **RESTORE MACHINE STATUS** is displayed and output with the M/S/T/Q signals.
- With the external start key, activate the output of the displayed signals. As long as these signals are being output, M4161 remains set.
- In the system files **PLC:\MGROUPS.SYS** and **PLC:\MSPLIT.SYS,** define the M functions to be output after a block scan.

The M/S/T/Q signals are output in this sequence (exception: see the **ORDER=PRIO** instruction):

- 1. M function that was defined with **MFIRST**
- 2. M/S/T/Q signals in the programmed sequence
- 3. M function that was defined with **MLAST**

After **RESTORE MACHINE STATUS,** the control checks whether the status set by the PLC agrees with the status calculated by the NC. No error message appears if this is the case; if for example another traverse range is selected in a tool change macro but the original traverse range is set at the end of the macro. If the NC status and PLC status do not match, the error message **PLC function not permitted** appears.

In order to run the above named functions on machines that have executed them through the PLC, leading to the **PLC function not permitted** error message, as an alternative there are functions that can be executed through an NC macro:

- **FN17: SYSWRITE ID20 NR13,** to switch between two spindles
- **FN31: RANGE SELECT,** to switch the traverse range **(RANGE),** the axis assignment **(ASSIGNED),** and the axis display **(DISPLAYED)**
- **FN32: PLC PRESET,** to execute a PLC datum shift

The functions are only visible with code number 555343.

Function	NC	PLC
Spindle switchover	**FN17: SYSWRITE ID20 NR13**	Module 9175
Range of traverse, axis assignment, axis display	**FN31: RANGE SELECT**	Module 9152
Datum shift	**FN32: PLC PRESET**	Module 9230

Canceling block scan

If block scan is cancelled, it is for possible for the NC status and PLC status not to match. A macro can be entered in NC MACRO.SYS after the keyword **STARTUPCANCEL =** for this. This macro is always called when block scan is not ended with **RESTORE MACHINE STATUS.** This macro brings the NC into concordance with the actual condition of the machine (traverse range, spindle, etc.).

MP7681 bit#4 enables you to activate an MP subfile during block scan. The MP subfile is activated with FN17 ID1020.

Danger

If bit#4 is set and the block scan is canceled nevertheless, the respective changes will remain effective!
If required, you have to undo these changes in order to restore a consistent state.
You can use a suitable NC macro for this, for example.

MP7681 **M/S/T/Q transfer to the PLC during block scan**

Format: %xxxx

Input: Bit 0:
0: Transfer M functions to the PLC during block scan.
1: Collect M functions and transfer to PLC after block scan.
Bit 1:
0: Transfer T code to the PLC during block scan
1: Transfer last T code to the PLC after block scan
Bit 2:
0: Transfer S or G code to the PLC during block scan
1: Transfer last S or G code to PLC after block scan
Bit 3:
0: Transfer FN19 outputs to the PLC during block scan
1: Transfer last FN19 outputs to the PLC after block scan
Bit 4:
0: MP subfiles are not activated during block scan
1: MP subfiles are activated during block scan

		Set	Reset
M4161	**M/S/T/Q transfer after block scan**	NC	NC

Delay time for the change signals (M/S/T)

MP7444 allows you to define the time for the output of the change signals M/S/T after a positioning movement. This enables you to reduce the machining time by avoiding filter run times. It is not always necessary to wait until the filter run times have been completed (e.g. switch-on of cooling water). The delay time can be deactivated in MP7444, if required.

MP7444	**Delay time for the change signals (M/S/T)**
Input:	0: Previous behavior, change signal after complete filter run time 1 = Change signal if the nominal feed rate reached the value 0 before the actual value (formed by the filters) did. 2 = Change signal if the nominal feed rate reached the value 0 before the actual value (formed by the filters) did, and an additional delay until the actual value reaches the time window the first time.

Instructions in MGROUPS.SYS

`GROUP =`
You divide M functions into groups. After a block scan, the last programmed M function in a group is transferred to the PLC. An M function can be entered in up to eight groups. During block scan, the M function is collected for all groups in which it occurs, and is then considered during restoration. However, this could lead to such M functions being executed more than once when restoring the machine status. This function only makes sense if the **`ORDER = PRIO`** option has not been entered.
Example: **`GROUP = M3,M4,M5`**

`SPECIAL =`
You define all M functions that are not defined in a group, and that should be sent to the PLC after a block scan.
Example: **`SPECIAL = M19,M22,M25`**

`MFIRST =`
`MLAST=`
You define two M functions to be sent to the PLC at the start and end of an output sequence after a block scan. This enables the PLC program to recognize that a sequence of M/S/T/Q strobes that was collected during the block scan is being transferred. You can omit these instructions if you do not need this information.
Example: **`MFIRST = M80, MLAST = M81`**

REMAIN = OUTPUT
All M functions that are not defined in MGROUPS.SYS are transferred during the block scan to the PLC. If you do not enter this instruction such M functions are ignored.

Note

You must use **`REMAIN = OUTPUT`** in the following functions:

- Datum shift with M4132
- PLC positioning except with TOOL CALL
- Traverse range switchover with M4135, if MP7490 = 1
- Switchover spindle 1/spindle 2 with Module 9175

If the PLC shifts the datum or switches the traverse range, the error message **PLC function not permitted** appears.

ORDER = PRIO
The M functions are transferred in the sequence in which they are entered in the MGROUPS.SYS file. If you do **not** enter this instruction, the M functions are transferred after a block scan in the sequence in which they were programmed.
HEIDENHAIN recommends that you **avoid** using this instruction.

TOOLGROUP =, TDEFGROUP =, SPINDLEGROUP =, FN19GROUP =
In conjunction with **ORDER = PRIO,** the output sequence of the S/T/Q strobes after a block scan is specified in the MGROUPS.SYS file. HEIDENHAIN recommends that you **avoid** using these instructions.

NCMACRO = TC, M
With an M function or during a TOOL CALL, you can also call an NC program instead of a T strobe (see "Tool Changer" on page 1513 and "Calling an NC macro with an M function" on page 1240).
The instruction **NCMACRO =** prevents NC macros for tool change (**TC**) or NC macros for M functions (**M**) from running during the block scan. Rather they are started at the end of the block scan.

Instructions in MSPLIT.SYS

M functions that are effective in several groups are divided in the MSPLIT.SYS file into function components.

Example: **M13=M3, M8**

8.1.16 M functions

In the iTNC you can program miscellaneous functions, also known as M functions. The code of an M function is transferred to the PLC before or after execution of the NC block.

M89 to M299 are reserved for the NC, and several M functions between M00 and M88 have fixed meanings for the NC. The other M functions are freely available.

Effective at A = beginning of block
E = end of block

M function	Meaning	Effect
M00	Program STOP/Spindle STOP/Coolant OFF	E
M01	Optional program STOP	E
M02	Program STOP/Spindle STOP/Coolant OFF/possible clearing of the status display[a]/go to block 1	E
M03	Spindle ON clockwise	A
M04	Spindle ON counterclockwise	A
M05	Spindle STOP	E
M06	Tool change/Program STOP[b]/Spindle STOP	E
M07		A
M08	Coolant ON	A
M09	Coolant OFF	E
M10		E
M11		A
M12		E
M13	Spindle ON clockwise/Coolant ON	A
M14	Spindle ON counterclockwise/Coolant ON	A
M15 - M18		A
M19		E
M20 - M29		A
M30	Same as M02	E
M31		A
M32 - M35		E
M36 - M51		A
M52 - M54		E
M55 - M59		A
M60		E
M61		A
M62		A
M63 - M70		E
M71 - M88		A

M function	Meaning	Effect
M89	Vacant miscellaneous function **or** cycle call, modally effective[b]	E
M90	Operation with following error: Constant feed rate in corners	A
M91	Within the positioning block: Coordinates are referenced to machine datum	A
M92	Within the positioning block: Coordinates are referenced to a position defined by the machine tool builder, such as tool change position	A
M93		A
M94	Reduce the rotary axis display to a value below 360°	A
M95 - M96	Approach behavior at the starting point of the contour	E
M97	Machine small contour steps	E
M98	Machine open contours completely	E
M99	Blockwise cycle call	E
M100		E
M101	Automatic tool change with replacement tool if maximum tool life has expired	A
M102	Reset M101	E
M103	Reduce feed rate during plunging to factor F	A
M104	Reactivate most recently set datum	A
M105	Machining with second k_V factor	A
M106	Machining with first k_V factor	A
M107	Suppress error message for replacement tools with oversize	A
M108	Reset M107	E
M109	Constant contouring speed on the tool cutting edge (increasing and decreasing the feed rate)	A
M110	Constant contouring speed on the tool cutting edge (only decreasing the feed rate)	A
M111	Reset M109/M110	E
M112	Insert rounding radius between nontangential straight lines	A
M113	Reset M112	E
M114	Automatic compensation of machine geometry when working with tilted axes	A
M115	Reset M114	E
M116	Feed rate for rotary axes in mm/min	A
M117	Reset M116	E
M118	Superimpose handwheel positioning during program run	A
M119		
M120	Pre-calculate the radius-compensated contour (LOOK AHEAD)	A
M121 - M123		

M function	Meaning	Effect
M124	Ignore points for calculating the rounding arc with M112	A
M125		
M126	Shortest-path traverse of rotary axes	A
M127	Reset M126	E
M128	Retain position of tool tip when positioning tilting axes (TCPM)	A
M129	Reset M128	E
M130	Within the positioning block: Points are referenced to the non-tilted coordinate system	A
M131		A
M132		A
M133		E
M134	Exact stop at nontangential contour transitions when positioning with rotary axes	A
M135	Reset M134	E
M136	Feed rate F in mm per spindle revolution	A
M137	Reset M136	E
M138	Selection of tilted axes	A
M139		A
M140	Retraction from the contour in the positive tool axis direction	A
M141	Suppress touch probe monitoring	A
M142	Delete modal program information	A
M143	Delete basic rotation	A
M144	Compensating the machine's kinematics configuration for ACTUAL/ NOMINAL positions at end of block	A
M145	Reset M144	E
M146	Save the current geometry information in a temporary file (tool-oriented pallet machining)	A
M147		A
M148	Retract the tool automatically from the contour at NC stop	A
M149	Reset M148	E
M150	Utilize the traverse range completely in NC blocks	A
M151 - M199		
M200	Laser cutting: Direct output of the programmed voltage	A
M201	Laser cutting: Output of voltage as a function of distance	A
M202	Laser cutting: Output of voltage as a function of speed	A
M203	Laser cutting: Output of voltage as a function of time (ramp)	A
M204	Laser cutting: Output of voltage as a function of time (pulse)	A
M205 - M299		A
M300 - M999		

a. depends on MP7300
b. depends on MP7440

- In the PLC, evaluate the M functions that have no fixed meaning for the NC.

When an M function is transferred to the PLC, the code of the M function is saved in W260 and the strobe marker M4072 is set:

- Set M4092 in order to report the execution of the M function. The next NC block is run. M4072 is reset by the NC.

The Mfunctions M00 to M99 can also be transferred decoded to the markers M1900 to M1999:

- Activate this function with M4571.

Note

M functions greater than 99 are not transferred to the PLC. They have a fixed meaning for the NC to activate certain functions.

		Set	Reset
W260	**Code for M functions**	NC	NC
M4072	**Strobe signal for M functions**	NC	NC
M4092	**Acknowledgment of M functions**	PLC	PLC
M4571	**Activation of decoded M-code transfer in M1900 to M1999**	PLC	PLC
M1900 - M1999	**Decoded M function if M4571 is set**	NC	NC

Status of M functions

- With Module 9060 you can ascertain the status of M functions M100 to M199.
- With Module 9061 the status of the non-modal M functions M94, M142, M143 and M146 can be ascertained.

Module 9060 M function status

Module 9060 can determine whether an M function between M100 and M199 is active.

Call:

PS	B/W/D/K	<Number of the M function (100 to 199)>
CM	9060	
PL	B/W/D	<Status> 0: M function inactive 1: M function active

Error recognition:

Marker	Value	Meaning
M4203	0	Status was ascertained
	1	Error code in W1022
W1022	1	Invalid number of M function

Module 9061 Status of non-modal M functions

With module 9061 the status of the non-modal M functions M94, M142, M143 and M146 can be interrogated. The status of the interrogated M function is then set until the module is called again, even if the NC program has finished.

Call:

PS	B/W/D/K	<Number of the M function (90 to 199)>
CM	9061	
PL	B/W/D	<Status> 0: M function was not active 1: M function was active

Error recognition:

Marker	Value	Meaning
M4203	0	Status was ascertained
	1	Error code in W1022
W1022	1	Invalid number of M function

Program stop with M functions

In the **Program Run, Single Block** and **Program Run, Full Sequence** operating modes the next NC block is not run until you have reported execution of the M function:

- Set M4092 in order to report the execution of the M function.

For special machines you can deselect the program stop (see "Special Functions for Laser Cutting Machines" on page 1509).

- Select the program stop with MP7440, bit 2.

Program stop with M06

According to ISO 6983, the M function M06 means "tool change."

- With MP7440 bit 0, select program stop when M06 is transferred to the PLC. After the program stop and the tool change, the NC program must be restarted through an NC start or by the PLC.

Modal cycle call M89

You can use the M function M89 to program a modal cycle call. The possibilities for calling a cycle are:

- NC block **CYCL CALL**
- Miscellaneous function M99. M99 is non-modal, i.e. it must be specially programmed each time is it to be executed.
- Miscellaneous function M89. M89 depends on MP7440 bit 1. M89 is effective modally as a cycle call, i.e. in each subsequent positioning block, the fixed cycle last programmed is called. M89 is cancelled by M99 or by a CYCL CALL block.

If M89 is not defined as a modal cycle call, it is transferred to the PLC as a normal M function at the beginning of the block.

Reduced feed rate of tool axis with M103

With M103 F.. you can reduce the contouring feed rate for motion in the negative direction of the tool axis. The tool axis share of the feed rate is limited to a value that the iTNC has calculated from the most recently programmed feed rate.

$F_{max} = F_{prog} \cdot F_{\%}$
F_{max} = maximum feed rate in the negative direction of the tool axis
F_{prog} = most recently programmed feed rate
$F_{\%}$ = programmed factor behind M103 as a percentage

M103 F.. is canceled by re-entering M103 without a factor.

- Enable the M103 F.. function with MP7440 bit 2.

Automatic activation of M134

In the standard setting, a transition element is inserted for positioning with rotary axes at non-tangential transitions (depending on the acceleration, jerk and tolerance). With M134, a precision stop is performed at these transitions:

- Enable the automatic activation of M134 with MP7440 bit 6.

MP7440 **Output of M functions**
Format: %xxxxxxx
Input: Bit 0 – Program stop with M06:
0: Program stop with M06
1: No program stop with M06
Bit 1 – Modal cycle call M89:
0: Normal code transfer of M89 at beginning of block
1: Modal cycle call M89 at end of block
Bit 2 – Program stop with M functions:
0: Program stop until acknowledgment of the M function
1: No program stop. Acknowledgment is not waited for.
Bit 3 – Switching of k_V factors with M105/M106:
0: Function is not in effect
1: Function is effective
Bit 4 – Reduced feed rate in the tool axis with M103
0: Function is not in effect
1: Function is effective
Bit 5 – Reserved
Bit 6 – Automatic activation of M134
0: M134 must be activated in the NC program
1: M134 is automatically activated when an NC program is selected

Error messages during cycle call

Before execution of a fixed cycle, the spindle must be started with M3 or M4. If this is not the case, the error message **Spindle ?** appears. If you are using a high speed cutting (HSC) spindle that is started by its own M function (not M3 or M4):

- Suppress the error message **Spindle ?** with MP7441 bit 0.

If a positive depth is programmed in machining cycles, the error message **Enter depth as negative** appears:

- Suppress the error message **Enter depth as negative** with MP7441 bit 2.

MP7441 **Error message during cycle call**
Format: %xxx
Input: Bit 0 –
0: Error message **Spindle ?** is not suppressed
1: Error message **Spindle ?** is suppressed
Bit 1 – Reserved, enter 0
Bit 2 –
0: Error message **Enter depth as negative** is suppressed
1: Error message **Enter depth as negative** is not suppressed

Auxiliary cycles

Cycles 18 (thread cutting) and 33 (thread on taper) are auxiliary cycles. You cannot use them alone, but you can use them for your OEM cycles (see the User's Manual).

▶ Set MP7245 = 1 to enable the auxiliary cycle.

MP7245	**Disable auxiliary cycles**
Input:	0: Auxiliary cycles disabled 1: Auxiliary cycles permitted

Calling an NC macro with an M function

The M functions M0 to M88 and M300 to M999 can call an NC macro in all operating modes. First the table PLC:\MFUNCT\MFUNCT.TAB must exist. Line number 0 represents M0, line number 1 represents M01, etc. The NC macros must be entered in the directory PLC:\MFUNCT\ with the name of their M function (e.g. M301.H).
M functions that call an NC macro are not sent to the PLC. If an M function defined in MFUNCT.TAB is programmed in a macro, the M function is reported to the PLC.
To synchronize the current machine status and the look-ahead calculation with an NC macro call, see "NCMACRO.SYS" on page 1644.

For behavior during a block scan, see "Instructions in MGROUPS.SYS" on page 1230.

With **`FN17: SYSWRITE ID420 NR0 IDX0 = 0,`** all coordinate transformations (e.g. cycles 7, 8, 10, 11, 19) performed in the NC macro become globally effective. Without this block, they remain locally effective (only in the NC macro).

Explanation of the columns in the table MFUNCT.TAB:

Column name	Description	Input
NR	Number of the M function	–
MACRO	Is the macro present?	YES: "Y", "y" or "1" NO: "N", "n" or "0"
EFFECTIV	Is the M function effective at the beginning or end of block (only M0 to M88 and M300 to M999)?	
WAIT	Only for M functions that do **not** call NC macros: Should the NC wait for acknowledgment from the PLC (only for M0 to M88 and M300 to M999)? If there is no entry in this column, MP7440 bit 2 is valid.	
MANLOCK	Is the M function locked in the manual operating modes?	
NONESTED	May the macro **not** be called by another macro?	
RK	Should the macro stored for the M function also be used for active radius compensation (RR/RL)? Until now, the error message **Cancel radius comp. before PLC pos.** was displayed. A macro that can also be run during active radius compensation should only contain NC functions that do not influence radius compensation in the calling program.	

Note

A maximum of six NC programs can be nested (subprograms, cycles, macros).

8.1.17 Powering up and shutting down the control

Powering up the control

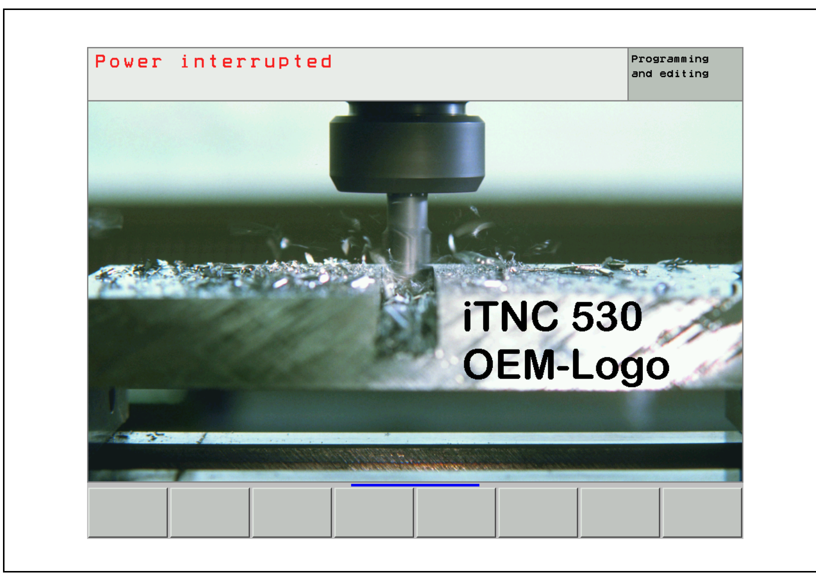
While the control is starting, a customer-specific company logo can be displayed instead of the HEIDENHAIN logo. If a service pack is installed, then a different logo may appear instead of the message window.

Requirements of the logo:

- The logo must be a bitmap file (*.BMP) with a color resolution of 16 or 24 bits. It is displayed in 16 bits.
- Maximum picture size:
  - BF 120: 640 x 480
  - BF 150: 1024 x 768

> **Note**
>
> If the picture is larger than the window, it will be cropped symmetrically. If the picture is smaller than the window, it will be displayed centered.

- The logo must be designed to remain recognizable even with a reduced window:
  While the control is powering up, the complete logo is displayed. If power is interrupted, only a reduced window remains available:
  - BF 120: 636 x 424
  - BF 150: 1020 x 681



- Standard logo: Enter the keyword **LOGO =** in OEM.SYS, followed by the complete path of the logo, e.g. **LOGO = PLC:\LOGO\OEM-LOGO.BMP**
- Logo with service pack installed: Enter the keyword **LOGOSP =** in OEM.SYS, followed by the complete path of the logo, e.g. **LOGOSP = PLC:\LOGO\SP-LOGO.BMP**
- On the TNC, create a new directory, for example **PLC:\LOGO.**
- Transfer the logo to this directory

Shutting down the control

The control must be shut down before it can be switched off. This is done either with the soft key in **Manual mode** or by the PLC with Module 9279 or 9189. If the control is shut down (either with a PLC module or with the soft key), M4179 is set, the ready signal of the drives is removed, and the hard disk is set to sleep mode.

If a PLC output is to be set after shutting down the control, this function must be activated with MP4040 = 1 or 2. Shutting down via Module 9279 must be done with Mode 2. After the control has been shut down and after the time from MP4041 has expired, the PLC output specified in MP4042 is set.

Note

On the iTNC 530 with Windows 2000/XP, no PLC output can be set after shutdown (MP4040 to MP4042).

		Set	Reset
M4179	**Control is being shut down**	NC	NC

MP4040 **Set PLC output after shutdown**
Input: 0: Do not set a PLC output
1: Only after shutdown via Module 9279
2: After shutdown via Module 9279 or soft key

MP4041 **Time after shutdown until setting of the PLC output from MP4042**
Input: 0 to 1000 [s]

MP4042 **PLC output to be set after shutdown**
Input: 0 to 31

Module 9279 Shut down control (configurable)
With Module 9279 the control can be

- Shut down
- Shut down and restarted (reset)
- Shut down, and then a PLC output specified in MP4040, MP4041 and MP4042 is set.

In each case the PLC is not executable after shutting down, and no message appears on the screen that the control is being shut down.

Call:
PS B/W/D/K <Mode>
0: Shut down the control
1: Shut down and restart the control
2: Shut down the control depending on MP4040, MP4041 and MP4042
CM 9279

Error recognition:

Marker	Value	Meaning
M4203	0	Control reset was carried out
	1	Error code in W1022
W1022	2	Invalid mode
	20	Module was not called in a spawn job or submit job

Module 9189 Shut down the control
Module 9189 shuts down the control. The PLC is not executable after shutdown. The message windows, which appear during shutdown via soft key, do not appear.

Call:
CM 9189

Error recognition:

Marker	Value	Meaning
M4203	0	Control was shut down
	1	Error code in W1022
W1022		Module was not called in a spawn job or submit job

Message for power interruption

After the control powers up, the **`Power interrupted`** message appears.

- Press the CE key to acknowledge this message and compile the PLC program.

With MP7212 you can suppress this message, e.g. for unattended operation.

MP7212 **Power interrupted message**
Input: 0: Acknowledge the **`Power interrupted`** message with CE key
1: **`Power Interrupted`** message does not appear

8.1.18 Arc end-point tolerance

The iTNC uses the entered NC data to calculate the deviation of the arc radius between the beginning and end of the arc:

- Enter a tolerance value in MP7431.
  If the entered tolerance is exceeded, the error message **CIRCLE END POS. INCORRECT** appears.

MP7431	**Arc end-point tolerance**
Input:	0.0001 to 0.016 [mm]

8.1.19 Limit-switch tolerance for M140 / M150

With MP7432 you can enter a tolerance for the limit switches. The limit switches can then be traversed with M140/M150 by this tolerance without an error message.
Retraction with M140/M150 can still be performed if a limit value has been entered and the axes are still within the tolerance. If the axes are still within the tolerance, they can even be moved outside of the traverse range, but only to another limit switch.
If one or more axes that are already slightly outside the limit-switch range are to be moved to the same limit switch, the axes are not moved. No limit-switch error message appears if the axes are still within the tolerance of the limit switch.

MP7432	**Limit-switch tolerance for M140 / M150**
Input:	0.0001 to 1.0000 [mm]
	0: Limit-switch tolerance off

8.1.20 Radius compensation

A path to be traversed can be increased or decreased by the tool radius by entering "R+" or "R–."

The input dialog is not initiated with the "L" key but directly with the orange axis-direction key. For reasons of compatibility, this function has been retained for point-to-point and straight cut controls.

Example:	X + 20	R+	Conversational programming
	G07	X + 20 G49	ISO programming

Paraxially compensated positioning blocks (R+/R–) and radius-compensated positioning blocks (RR/RL) must not be entered in succession.
To avoid erroneous entries:

- Disable the generation of paraxial positioning blocks by setting MP7246 bit 0.

You can switch off the peripheral milling function (3-D radius compensation with tool orientation) by setting MP7682 bit 10. As a result, only 2-D radius compensation is active even if M128 with RR/RL radius compensation is active.

MP7246 Machine parameter with multiple function
Input: Bit 0 – Paraxial positioning blocks
0: Generation permitted
1: Generation disabled

MP7682 Machine parameter with multiple function
Input: Bit 10 – Peripheral milling
0: Peripheral milling allowed
1: Peripheral milling inactive

8.1.21 User Parameters

You can provide the machine tool operator with easy access to up to 16 machine parameters known as user parameters. He can then call them through the MOD function by simply pressing the USER PARAMETER soft key.

- In MP7330.x, enter the numbers of the machine parameters that you wish to make available.

Example:

If MP7230.1 should be the first available user parameter:

- Enter the input value 7230.01 in MP7330.0.

If the user selects a user parameter, a message appears on the screen. You can specify this message:

- In the system file OEM.SYS, enter the name of the PLC dialog message file with the command **PLCDIALOG=**.
- In MP7340.x enter the line number of the PLC dialog message to be displayed.

MP7330.0-15 Specification of user parameters 1 to 16
Input: 0 to 9999.00 (no. of the user parameter)

MP7340.0-15 Dialog messages for user parameters 1 to 16
Input: 0 to 4095 (line number of the PLC dialog message file)

8.1.22 Code numbers

You can enter certain code numbers through the MOD function. With these code numbers you can activate certain functions.

The following code numbers have a fixed meaning:

Code number	Function
95148	Select machine parameter file
807667	Select the PLC mode
857282	Reset the operating times
75368	Automatic offset adjustment
123	Call machine parameters that are accessible to the user.
531210	Delete M0 to M999 and B0 to B127
688379	Oscilloscope
555343	FN17: Overwrite system data FN25: Overwrite datum
75368	Offset adjustment
NET123	Ethernet settings
LOGBOOK	Read out the log
FAILTEST	Simulate internal emergency stop
SIK	Option menu
KINEMATIC	Selection window for the machine kinematics

The code of the entered code number is entered in double word D276. You can evaluate this code and define your own functions for code numbers, or disable fixed code numbers.

		Set	Reset
D276	**Code of the code number last entered via MOD**	NC	NC

8.1.23 Programming Station mode

You can switch the control into programming-station mode with MP7210. This way the control can be used as a simple programming station. No drives are enabled.

In this setting NC programs cannot be executed. You can only create and test NC programs.

You can select whether the PLC should be active. In addition, you can activate the emergency-stop loop if the PLC is active. The emergency-stop loop must already be connected correctly (X41/34 and X42/4) in order to switch the control on correctly.

If a programming-station mode is active (MP7210 > 0), this is shown in the first line of the main menu of the **`PLC Programming`** operating mode via "**`MP7210 = x`**."

MP7210 **Programming station**

Input: 0: Controlling and programming
1: Programming station with PLC active
2: Programming station with PLC inactive
3: Programming station with PLC and emergency stop active (X41/34 and X42/4)

8.1.24 Color settings

The colors of the display unit can be defined by machine parameters.

The following color settings **cannot** be changed:

- HEIDENHAIN logo after machine switch-on (standard color)
- Error message for invalid machine parameters (red)
- Blinking error message (red)
- Plan view in the graphic display (blue)
- Cursor (inverse)

You define the desired color by mixing the basic colors red, green and blue. Every basic color has 256 difference stages of intensity. The input values for color settings are byte-oriented. HEIDENHAIN recommends hexadecimal input.

Color	**Red**		**Green**		**Blue**	
Adjustment	Rough	Fine	Rough	Fine	Rough	Fine
HEX ranges	0 to F	0 to F	0 to F	0 to F	0 to F	0 to F
Input for yellow: $0....	3	9	3	9	0	0

The colors can also be poorly adjusted (e.g. red error message on red background). HEIDENHAIN therefore supplies the controls with a standard color setting, which is suggested by the control during creation of the MP list.

To configure the screen saver:

- Enter in MP7392.0 the time in minutes after which the screen saver should activate itself. Enter 0 to disable the screen saver.
- With MP7392.1 you select the type of screen saver that starts after the time in MP7392.0 expires.

If a (new) error message occurs, the screen saver will be exited automatically.

MP7350	**Window frames**
MP7351	**Error messages**
MP7351.0	Priority 0 (error)
MP7351.1	Priority 1 (warning)
MP7351.2	Priority 2 (information)
MP7352	**"Machine" operating mode display**
MP7352.0	Background
MP7352.1	Text for operating mode
MP7352.2	Dialog
MP7353	**"Programming" operating mode display**
MP7353.0	Background
MP7353.1	Text for operating mode
MP7353.2	Dialog
MP7354	**"Machine" program text display**
MP7354.0	Background
MP7354.1	General program text
MP7354.2	Active block
MP7354.3	Comments and unused machine parameters in the machine parameter file
MP7354.4	Background of inactive windows
MP7355	**"Programming" program text display**
MP7355.0	Background
MP7355.1	General program text
MP7355.2	Active block
MP7355.3	Comments and unused machine parameters in the machine parameter file
MP7355.4	Background of inactive windows

MP7356	**Status window and PLC window**
MP7356.0	Background
MP7356.1	Axis positions in the status display
MP7356.2	Status display other than axis positions
MP7357	**"Machine" soft-key display**
MP7357.0	Background
MP7357.1	Text color
MP7357.2	Inactive soft-key row
MP7357.3	Active soft-key row
MP7358	**"Programming" soft-key display**
MP7358.0	Background
MP7358.1	Text color
MP7358.2	Inactive soft-key row
MP7358.3	Active soft-key row
MP7360	**Graphics: 3-D view**
MP7360.0	Background
MP7360.1	Top surface
MP7360.2	Front face
MP7360.3	Text display in the graphics window
MP7360.4	Lateral face
MP7360.5	Lowest point of blank form
MP7360.6	Highest point of blank form (below surface)
MP7361	**Graphics: Projection in three planes**
MP7361.0	Background
MP7361.1	Top view
MP7361.2	Front and side view
MP7361.3	Axis cross and text in the graphic display
MP7361.4	Cursor
MP7362	**Additional text display in the graphic window and pocket calculator**
MP7362.0	Background of graphic window and pocket calculator
MP7362.1	Background of status display and keys of the pocket calculator
MP7362.2	Status symbols and symbols of the pocket calculator (c in "cos")
MP7362.3	Status values and texts of the pocket calculator (os in "cos")
MP7362.4	Color of the unselected tabs in the graphics window
MP7362.5	AFC tab – Background color
MP7362.6	AFC tab – Color of actual override factor
MP7362.7	AFC tab – Color of actual spindle load
MP7363	**Programming graphics**
MP7363.0	Background
MP7363.1	Resolved contour
MP7363.2	Subprograms and frame for zooming
MP7363.3	Alternative solutions
MP7363.4	Nonresolved contour
MP7363.5	Rapid traverse movements

MP7364	**Color of the help illustrations for cycles**
MP7364.0-6	Colors 1 to 7 of the graphic program used
MP7364.7	Line color (color 8 of the graphic program)
MP7364.8	Color for highlighted graphic elements if defined in the help illustration
MP7364.9	Background

MP7365	**Oscilloscope**
MP7365.0	Background
MP7365.1	Grid
MP7365.2	Cursor and text
MP7365.3	Selected channel
MP7365.4-9	Channel 1 to 4

MP7366	**Pop-up window (HELP key, pop-up menus etc.)**
MP7366.0	Background
MP7366.1	Text or foreground
MP7366.2	Active line
MP7366.3	Title bar
MP7366.4	Scroll-bar field
MP7366.5	Scroll bar
MP7366.6-14	Reserved

MP7367	**Large PLC window**
MP7367.0	Background
MP7367.1-7	Colors 1 to 7 (Color 8: MP7350)
MP7367.8-14	Colors 9 to 15

MP7368	**Calculator**
MP7368.0	Background
MP7368.1	Background of displays and keys
MP7368.2	Key texts ("os" in "cos")
MP7368.3	Key symbols

MP7369	**Directory tree in PGM MGT**
MP7369.0	Text background
MP7369.1	Text
MP7369.2	Text background of the active folder
MP7369.3	Line color of the tree structure
MP7369.4	Folders
MP7369.5	Drives
MP7369.6	Text background of the heading in the browser window

MP7370	**Small PLC window**
MP7370.0	Background
MP7370.1-15	Colors 1 to 15

MP7392.0	**Time after which the screensaver is activated**
Input:	1 to 99 [min] 0: No screen saver

MP7392.1	**Type of screensaver**
Input:	0: No screen saver 1: Default screensaver of the X server 2: 3-D line graphics

The standard color setting is shown in the following list:

Machine parameters	Standard setting
MP7350	$0808080
MP7351.0	$0FF2020
MP7351.1	$000FF00
MP7351.2	$00000FF
MP7352.0	$0ECECEC
MP7352.1	$0000000
MP7352.2	$00000FF
MP7353.0	$0C0C0C0
MP7353.1	$0000000
MP7353.2	$00000FF
MP7354.0	$0FFFFFF
MP7354.1	$0000000
MP7354.2	$00000FF
MP7354.3	$0A0A0A0
MP7354.4	$0C8C8C8
MP7355.0	$0FFFFFF
MP7355.1	$0000000
MP7355.2	$00000FF
MP7355.3	$0A0A0A0
MP7355.4	$0C8C8C8
MP7356.0	$0ECECEC
MP7356.1	$00000FF
MP7356.2	$00000FF
MP7357.0	$0C0C0C0
MP7357.1	$0000000
MP7357.2	$0000000
MP7357.3	$00000FF
MP7358.0	$0C0C0C0
MP7358.1	$0000000
MP7358.2	$0000000
MP7358.3	$00000FF
MP7360.0	$0AAAAAA
MP7360.1	$08888F0
MP7360.2	$00011AA
MP7360.3	$0FFFFFF
MP7360.4	$00000FF
MP7360.5	$0550000
MP7360.6	$0FFFFFF

Machine parameters	Standard setting
MP7361.0	$0AAAAAA
MP7361.1	$00000E8
MP7361.2	$00000E8
MP7361.3	$0FF0000
MP7361.4	$0FF00FF
MP7362.0	$0ECECEC
MP7362.1	$0FFFFFF
MP7362.2	$00000FF
MP7362.3	$00000FF
MP7362.4	$0C0C0C0
MP7362.5	$0ECECEC
MP7362.6	$000FF80
MP7362.7	$000AAFF
MP7363.0	$0ECECEC
MP7363.1	$00000FF
MP7363.2	$0FF00FF
MP7363.3	$000EC00
MP7363.4	$0FF0000
MP7364.0	$0DBD3DB
MP7364.1	$0FF0000
MP7364.2	$0202020
MP7364.3	$0000000
MP7364.4	$00000FF
MP7364.5-6	$00000FF
MP7364.7	$0AA0000
MP7364.8	$000EEEE
MP7364.9	$0DBD3DB
MP7365.0	$0FFFFFF
MP7365.1	$0808080
MP7365.2	$00000FF
MP7365.3	$0FF0000
MP7365.4	$0C08030
MP7365.5	$000FF00
MP7365.6	$0FF00FF
MP7365.7	$00000FF
MP7365.8	$0FFCF00
MP7365.9	$000CFFF

Machine parameters	Standard setting
MP7366.0	$0ECECEC
MP7366.1	$0000000
MP7366.2	$00000FF
MP7366.3	$0FF0000
MP7366.4	$0FFFFFF
MP7366.5	$0FF0000
MP7366.6	$0000000
MP7366.7	$0202020
MP7366.8	$0404040
MP7366.9	$0606060
MP7366.10	$0808080
MP7366.11	$0A0A0A0
MP7366.12	$0C0C0C0
MP7366.13	$0E0E0E0
MP7366.14	$0FFFFFF
MP7367.0	$0ECECEC
MP7367.1	$0000000
MP7367.2	$0D0D0D0
MP7367.3	$0A0A0A0
MP7367.4	$0808080
MP7367.5	$0404040
MP7367.6	$0202020
MP7367.7	$000FF9C
MP7367.8	$0FFFF7C
MP7367.9	$0FF0000
MP7367.10	$000FF00
MP7367.11	$0001C5C
MP7367.12	$000FFFF
MP7367.13	$00000FF
MP7367.14	$0FF00FF

Machine parameters	Standard setting
MP7368.0	$0ACACAC
MP7368.1	$0FFFFFF
MP7368.2	$00000FF
MP7368.3	$0FF0040
MP7369.0	$0ECECEC
MP7369.1	$0000000
MP7369.2	$00000FF
MP7369.3	$0000000
MP7369.4	$0FF6000
MP7369.5	$0FF0040
MP7369.6	$0FF0000
MP7370.0	$0C0C0C0
MP7370.1	$0000000
MP7370.2	$0D0D0D0
MP7370.3	$0A0A0A0
MP7370.4	$0808080
MP7370.5	$0404040
MP7370.6	$0202020
MP7370.7	$000FF9C
MP7370.8	$0FFFF7C
MP7370.9	$0FF0000
MP7370.10	$000FF00
MP7370.11	$0001C5C
MP7370.12	$000FFFF
MP7370.13	$00000FF
MP7370.14	$0FF00FF
MP7370.15	$0FFFFFF

8.1.25 Graphic display

In the graphics window you can view the following graphics:

- Program verification graphics
- Parallel graphics
- Programming graphics
- Help graphic

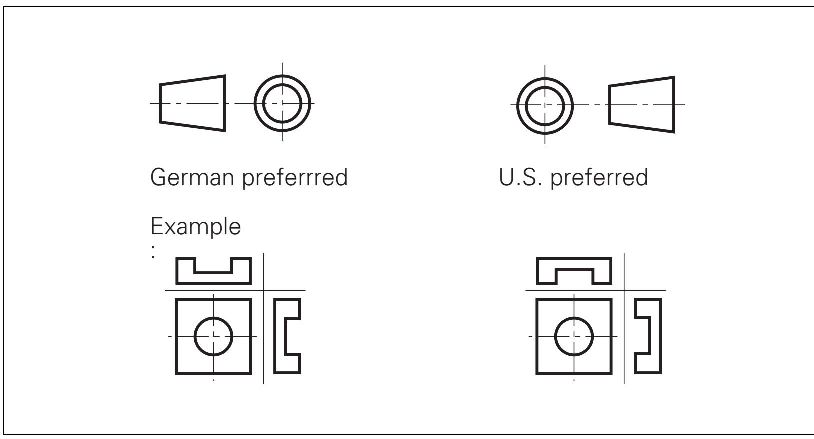
For the test graphics and parallel graphics you can choose one of three display modes:

- Projection in three planes
- Plan view
- 3-D view

Projection in three planes

The display in three planes can be shown in 1st-angle projection as preferred in Germany or in the American-style 3rd-angle projection:

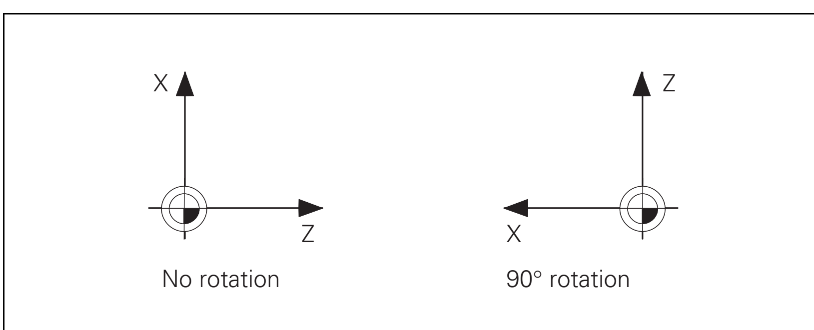
- ▶ Select the type of projection with MP7310, bit 0.



Rotation of the coordinate system

You can rotate the coordinate system for graphic display by +90° if, for example, the Y axis is defined as tool axis.

- ▶ Select the angle of rotation with MP7310, bit 1.



Graphic display for datum shift

In an NC program you can program several BLK forms in succession.

After datum shift with Cycle 7, the shift can be interpreted to apply also to subsequent blank forms:

- Define the BLK form shift in bit 2 of MP7310.

Position of the cursor

In the display in three planes you can display the position of the cursor:

- Switch this function on with MP7310, bit 3.

MP7310	**Graphic display mode**
Format:	%xxxxxxxx
Input:	Bit 0 – Projection in three planes: 0: German-preferred projection 1: US-preferred projection Bit 1 – Rotating the coordinate system in the working plane by 90° 0: No rotation 1: Rotation by +90° Bit 2 – BLK form after datum shift: 0: Shifted 1: Not shifted Bit 3 – Display of the cursor position: 0: Not displayed 1: Displayed Bit 4 – Reserved Bit 5 – 3-D graphics during program test 0: 2.5-D and 3-D 1: 2.5 D Bit 6 – Stock removal with an inclined tool 0: Not active 1: Active Bit 7 – Exact evaluation of the column LCUTS (cutting length) from the TOOL.T table in order to display special tools (e.g. saw blade). 0: Free evaluation 1: Exact evaluation for special tools

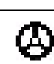
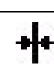
8.1.26 Special characters

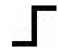
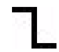
To enter special characters, use the following key combinations:

<table>
<tr><th>Key combination</th><th>Special characters</th></tr>
<tr><td>SHIFT + "</td><td>’</td></tr>
<tr><td>SHIFT + &</td><td>@</td></tr>
<tr><td>SHIFT + (</td><td>[</td></tr>
<tr><td>SHIFT +)</td><td>]</td></tr>
<tr><td>SHIFT + –</td><td>_</td></tr>
<tr><td>SHIFT + /</td><td>\</td></tr>
<tr><td>SHIFT + !</td><td>|</td></tr>
<tr><td>SHIFT + ^</td><td>~</td></tr>
<tr><td>SHIFT + #</td><td>ESC</td></tr>
<tr><td>SHIFT + RET</td><td><Form feed></td></tr>
<tr><td>SHIFT + SPACE</td><td>Switch between uppercase and lowercase</td></tr>
</table>

8.1.27 iTNC character set

Small characters

No.	Character	No.	Character	No.	Character	No.	Character
01 - 1D		54	T	8B	Ï	CB	Ë
1E		55	U	8C	Î	CC	Ì
1F		56	V	8D	Ì	CD	Í
20	<SPACE>	57	W	8E	Ä	CE	Î
21	!	58	X	8F	Å	CF	Ï
22	“	59	Y	90	É	D0	<SPACE>
23	#	5A	Z	91		D1	Ñ
24	$	5B	[	92	Æ	D2	Ò
25	%	5C	\	93	ô	D3	Ó
26	&	5D	]	94	ö	D4	Ô
27	’	5E	^	95	ò	D5	Õ
28	(	5F	_	96	Û	D6	Ö
29	)	60	’	97	ù	D7	Œ
2A	*	61	a	98	Ÿ	D8	Ø
2B	+	62	b	99	Ö	D9	Ù
2C	,	63	c	9A	Ü	DA	Ú
2D	-	64	d	9B - 9F		DB	Û
2E	.	65	e	A0	Á	DC	Ü
2F	/	66	f	A1	¡	DD	Ÿ
30	0	67	g	A2	Ó	DE	<SPACE>
31	1	68	h	A3	Ú	DF	ß
32	2	69	i	A4	ñ	E0	à
33	3	6A	j	A5	Ñ	E1	á
34	4	6B	k	A6	O	E2	â
35	5	6C	l	A7	A	E3	ã
36	6	6D	m	A8 - AD		E4	ä
37	7	6E	n	AE	<<	E5	å
38	8	6F	o	AF	>>	E6	æ
39	9	70	p	B0	°	E7	ç
3A	:	71	q	B1		E8	è
3B	;	72	r	B2		E9	é
3C	<	73	s	B3		EA	ê

No.	Character	No.	Character	No.	Character	No.	Character
3D	=	74	t	B4		EB	ë
3E	>	75	u	B5	µ	EC	ì
3F	?	76	v	B6		ED	í
40	@	77	w	B7		EE	î
41	A	78	x	B8		EF	ï
42	B	79	y	B9		F0	<SPACE>
43	C	7A	z	BA		F1	ñ
44	D	7B	(	BB		F2	ò
45	E	7C	|	BC		F3	ó
46	F	7D	)	BD		F4	ô
47	G	7E	~	BE		F5	õ
48	H	7F		BF	¿	F6	ö
49	I	80		C0	À	F7	œ
4A	J	81	ü	C1	Á	F8	ø
4B	K	82		C2	Â	F9	ù
4C	L	83		C3	Ã	FA	ú
4D	M	84	Ä	C4	Ä	FB	û
4E	N	85	À	C5	Å	FC	ü
4F	O	86	Å	C6	Æ	FD	ÿ
50	P	87	Ç	C7	Ç	FE	<SPACE>
51	Q	88	Ê	C8	È	FF	<SPACE>
52	R	89	Ë	C9	É		
53	S	8A	È	CA	Ê		

Medium characters

No.	Character	No.	Character	No.	Character	No.	Character
01 - 1D		53	S	89	Ë	CA	Ê
1E		54	T	8A	È	CB	Ë
1F		55	U	8B	Ï	CC	Ì
20	<SPACE>	56	V	8C	Î	CD	Í
21	!	57	W	8D	Ì	CE	Î
22	“	58	X	8E	Ä	CF	Ï
23	#	59	Y	8F	Å	D0	
24	$	5A	Z	90	É	D1	Ñ
25	%	5B	[	91		D2	Ò
26	&	5C		92	Æ	D3	Ó
27	’	5D	]	93	ô	D4	Ô
28	(	5E	^	94	ö	D5	Õ
29	)	5F	_	95	ò	D6	Ö
2A	*	60	’	96	Û	D7	Œ
2B	+	61	a	97	ù	D8	Ø
2C	,	62	b	98	Ÿ	D9	Ù
2D	-	63	c	99	Ö	DA	Ú
2E	.	64	d	9A	Ü	DB	Û
2F	/	65	e	9B - 9F		DC	Ü
30	0	66	f	A0	Á	DD	Ÿ
31	1	67	g	A1	¡	DE	
32	2	68	h	A2	Ó	DF	ß
33	3	69	i	A3	Ú	E0	à
34	4	6A	j	A4	ñ	E1	á
35	5	6B	k	A5	Ñ	E2	â
36	6	6C	l	A6	O	E3	ã
37	7	6D	m	A7	A	E4	ä
38	8	6E	n	A8 - AD		E5	å
39	9	6F	o	AE	<<	E6	æ
3A	:	70	p	AF	>>	E7	ç
3B	;	71	q	B0	°	E8	è
3C	<	72	r	B1		E9	é

No.	Character	No.	Character	No.	Character	No.	Character
3D	=	73	s	B2		EA	ê
3E	>	74	t	B3		EB	ë
3F	?	75	u	B4		EC	ì
40	@	76	v	B5	µ	ED	í
41	A	77	w	B6		EE	î
42	B	78	x	B7		EF	ï
43	C	79	y	B8		F0	
44	D	7A	z	B9		F1	ñ
45	E	7B	(	BA		F2	ò
46	F	7C		BB		F3	ó
47	G	7D		BC -BE		F4	ô
48	H	7E	~	BF	¿	F5	õ
49	I	7F		C0	À	F6	ö
4A	J	80	Ç	C1	Á	F7	œ
4B	K	81	ü	C2	Â	F8	ø
4C	L	82		C3	Ã	F9	ù
4D	M	83		C4	Ä	FA	ú
4E	N	84	Ä	C5	Å	FB	û
4F	O	85	À	C6	Æ	FC	ü
50	P	86	Å	C7	Ç	FD	ÿ
51	Q	87	Ç	C8	È	FE	
52	R	88	Ê	C9	É	FF	

Large characters

No.	Character	No.	Character	No.	Character	No.	Character
01 - 1D		35	5	4D	M	65	E
1E		36	6	4E	N	66	F
1F		37	7	4F	O	67	G
20	<SPACE>	38	8	50	P	68	H
21	!	39	9	51	Q	69	I
22	“	3A	:	52	R	6A	J
23	#	3B	;	53	S	6B	K
24	$	3C	<	54	T	6C	L
25	%	3D	=	55	U	6D	M
26	&	3E	>	56	V	6E	N
27	’	3F	?	57	W	6F	O
28	(	40	@	58	X	70	P
29	)	41	A	59	Y	71	Q
2A	*	42	B	5A	Z	72	R
2B	+	43	C	5B	[	73	S
2C	,	44	D	5C	\	74	T
2D	-	45	E	5D	]	75	u
2E	.	46	F	5E	^	76	v
2F	/	47	G	5F	_	77	w
30	0	48	H	60		78	x
31	1	49	I	61	a	79	y
32	2	4A	J	62	b	7A	z
33	3	4B	K	63	c	7B - FF	
34	4	4C	L	64	D		

8.1.28 Conversational language

The TNC is delivered with all NC-dialog human languages already loaded:

- In MP7230.0 select the conversational language in which you wish to work.

If the NC dialog messages for the selected language are not on the hard disk, the error message **LANGUAGE LOAD ERROR** appears. You can continue working in the default language English.

You can write your own dialog messages and save them in several languages:

- Save your dialog messages in permanently defined directories in the PLC partition.

These directories are:

PLC:\LANGUAGE\ CHINESE\
CZECH\
DANISH\
DUTCH\
ENGLISH\
ESTONIA\ (Option #41 – ID 530 184-01)
FINNISH\
FRENCH\
GERMAN\
HUNGARIA\
ITALIAN\
KOREAN\ (Option #41 – ID 530 184-01)
LATVIAN\ (Option #41 – ID 530 184-011)
LITHUANIAN\ (Option #41 – ID 530 184-01)
NORWEGIAN\ (Option #41 – ID 530 184-01)
POLISH\
PORTUGUE\
ROMANIAN\ (Option #41 – ID 530 184-01)
RUSSIAN\
SLOVAK\ (Option #41 – ID 530 184-01)
SLOVENIAN\ (Option #41 – ID 530 184-01)
SPANISH\
SWEDISH\
TURKISH\ (Option #41 – ID 530 184-01)

- With MP7230.1–3, switch to the desired language.

You can store PLC dialog message files, PLC error message files, and help files with identical file names in the different languages and in UNICODE (e.g. with PLCtext V3.1):

> **Note**
>
> If texts from these files are also used in the PLC window, then they cannot be in UNICODE, but must rather use UTF8-coding.

- In the system file OEM.SYS, enter only the file names with the commands **PLCDIALOG =** and **PLCERROR =**. The NC looks for the paths given in MP7230.1 or MP7230.2. The entry behind **MODEHELP =** is overwritten with the selected path whenever MP7230.3 is changed.

The PLC texts for PLC soft keys, FN14 error messages (UTF8 code in ERROR.A or FN14 file) and PLC mask files (ERROR.A or DIALOG.A) can also be displayed as UTF8-encoded texts (e.g. for Asian languages). For the FN14 error messages and the PLC mask files, the UTF8 code must then exist in the files stated.

PLC texts that are to be displayed in the PLC status window may not be UTF8-encoded. As an alternative, the English character set may be used.

Dialogs for user parameters and soft-key descriptions for the HR 420 cannot be displayed in Asian languages. If such dialogs are used, the English language is displayed instead.

The ASCII characters of the values 0xEF, 0xBB and 0xBF serve to indicate the UTF8 code. These three characters must be in the correct sequence as the first three characters in the appropriate file, and only serve to indicate the UTF8 code.

MP7230 **Switching the conversational language**
Input: 0: English
1: German
2: Czech
3: French
4: Italian
5: Spanish
6: Portuguese
7: Swedish
8: Danish
9: Finnish
10: Dutch
11: Polish
12: Hungarian
13: Reserved
14: Russian (Cyrillic characters)
15: Chinese (simplified)
16: Chinese (traditional)
Additional dialog language – 17: Slovenian (option #41)
18: Norwegian (option #41)
19: Slovak (option #41)
20: Latvian (option #41)
21: Korean (option #41)
22: Estonian (option #41)
23: Turkish (option #41)
24: Romanian (option #41)
25: Lithuanian (option #41)
14, 15, 16 and 17 only if a BF 150 is being used

MP7230.0 NC conversational language
MP7230.1 PLC conversational language (user parameters), soft keys for OEM cycles
MP7230.2 PLC error messages
MP7230.3 Help files

Module 9074 Load texts from error/dialog files into a PLC string

Module 9074 is used to copy texts from error files and dialog files (ERROR.A, DIALOG.A) to a PLC string directly and without coding.

Condition:

- Due to the file access the module must be called from a submit job or spawn job.
- The module returns the number of loaded characters (string length). With UTF-8 texts this value may be larger than the number of visible characters. Also, with UTF-8 texts the indexed access to individual characters in the string may not be correct in some cases.

Call:

```
PS    K/B/W/D   <PLC string>
                Source or constant string (e.g. S"D012")
PS    K/B/W/D   <PLC string>
                Target
CM    9074
PL    B/W/D     <Number of characters in target string>
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	11	Invalid address of the source string or target string
	20	Module was not called in a spawn job or submit job

Example: Transmission of text 13 from the error file to PLC string 54

```
PS    S"#E013"          ; Source: Line 13 from error.a
PS    K9000             ; Target string: S54
CM    9074
PL    W_STRLEN          ; Number of char in the string
L     M4203
IFT
      L     NP_W1022_Module_Error status
      =     Wxxx
ENDI
```

Additional conversational languages

HEIDENHAIN offers additional conversational languages, which can only be selected in combination with option #41 (ID 530 184-01). If option #41 is activated via **Programming and Editing** > **MOD** key and the keyword **SIK,** then the "additionally" declared dialog language is activated.

Decimal point

- With MP7280 you specify whether the decimal point will be a comma or a period.

MP7280 **Decimal character**
Input: 0: Decimal comma
1: Decimal point

8.1.29 Logs

PLC log

There is a separate log for the PLC in the PLC log under PLC:\PLCDEBUG.LOG. The marker should be set only for debugging purposes. Otherwise, unnecessary access to the hard disk will put a strain on the system. The following events are entered:

- Start of the PLC after switch-on.
- Start and stop of the PLC.
- Errors from PLC modules (this entry is only made if M4753 = 1).
- Run-time errors

		Set	Reset
M4753	**Write errors from PLC modules in the PLC log**	PLC	PLC

Diagnostics log

If M4754 is set, internal diagnostic information is entered in the log MYDEBUG.LOG. The marker should be set only for debugging purposes. Otherwise, unnecessary access to the hard disk will put a strain on the system.

		Set	Reset
M4754	**Write diagnostic information in MYDEBUG.LOG**	PLC	PLC

OEM log

> Note
>
> Writing to OEM logs must only take place in worthwhile intervals, since under circumstances the processing time could be affected negatively, and the hard disk written to unnecessarily.

Module 9277 Writing data into the OEM log

With Module 9277 the PLC can write data into a specific OEM log. Up to eight OEM logs can be used at the same time. The module can be called from a cyclic PLC program or from a spawn job or submit job. The string for the log entry may contain two wildcards (data1 and data2). Only wildcards that occur are replaced. The output format is controlled through the entry %d for integers or the entry %f for floating point numbers with three decimal places. Alternatively, you can define the number of decimal places with %.1f to %.6f.
Example of a string for the log entry: **`S"data1: %.2f data2: %d"`**
If the maximum log size of 1 MB is exceeded, the log is copied to <name>.LOG.OLD and a new log with the same name is created. Once the logs have been called, they remain open until the control is shut down.

Call:
```
PS    B/W/D/K/S<Path with file name (without .LOG extension)>
PS    B/W/D/K/S<String with placeholder for log entry>
PS    B/W/D/K   <Value for data1>
PS    B/W/D/K   <Value for data2>
PS    B/W/D/K   <Switch for additional entries>
                Bit 0 = 0/1: Entry without/with time stamp
                Bit 1 = 0/1: Entry without/with PLC cycle counter
CM    9277
```

Error recognition:

Marker	Value	Meaning
M4203	0	Data written into OEM log
	1	Error code in W1022
W1022	2	Invalid string number or invalid string
	22	Message cannot be transmitted

Standard log

The standard log serves as a troubleshooting aid. There are 4 MB of memory available for this purpose.
All entries in the log are marked with the momentary date and time.
You can read out the standard log in two ways:

- After entering the code word LOGBOOK, enter the path and name of an ASCII file and the time and date from which the log should record, as well as the time and date up to which it should record. After that, an ASCII file is generated and opened with the log entries.
- The PC software programs PLCdesign, TNCremo, and TNCremoNT offer you several functions for reading out the log.

Entry		Description
RESET		Powering up the control
BERR		Blinking error message
BREG		Register contents with a blinking error message
ERR		Error messages ■ P: PLC error message with the line number in the PLC error text file ■ N: NC error message with number ■ `Power fail interrupt:` Control was switched off by a POWERFAIL ■ Result of the file system test (in case the control was not properly shut down previously)
ERR	dosfsck -a	Check of the PLC: and TNC: hard-disk partitions if the control was not shut down correctly
INFO	"xxx"	"xxx": Name of the control process that enters the information in the log ■ PLC ■ SYS ■ MAIN ■ REMO ■ CTRL ■ SMARTNC ■ SMARTNC_LDF ■ BDEHAN ■ SOKY ■ GEO ■ PYTHON ■ FIXTURE ■ FILEMAN
INFO	CTRL KINEMATIC	Active tool with tool number, radius (R =) and length (L =)
INFO	CTRL collision	Status of collision monitoring in the Manual and Automatic operating modes
INFO	CTRL REG AXIS xx START REF=yy	Position yyyy of axis xx at start of movement
INFO	CTRL REG AXIS xx STOP REF=yy	Position yyyy of axis xx at stop of movement

Entry		Description
INFO	MAIN ERRCLEARED	Acknowledgment of an error message
INFO	MAIN ERR_RECURED	Error message entered several times
INFO	MAIN ENDAT	Entry for the position upon switch-on, if the switch-off and switch-on positions of an EnDat encoder do not match
INFO	MAIN RUNPROC	Status information about the current process
KEY		Key strokes
INFO	MAIN SOFTKEY	Path with associated image file of a pressed soft key
STIB[a]	ON	Control-in-operation on
	OFF	Control-in-operation off
	BLINK	Control-in-operation symbol blinking
INFO	MAIN START	Type of control, NC software and valid Feature Content Level (FCL)
INFO	MAIN FILE DEL	Faulty files on the hard disk, to be erased during booting
INFO	MAIN HDD	Hard disk designation
INFO	MAIN DSP	ID number of the active controller software
INFO	MAIN CYCLES	Test results for fixed cycles and touch probe cycles
INFO	MAIN KEYSOURCE	Source of the keystrokes ■ KEYBOARD ■ PLC ■ PLCNCSTART ■ HANDWHEEL ■ LSV2
INFO	MAIN KINEMATIC	Listing of the definition tables with collision bodies that are monitored for collision with option #40, DCM.

a. Control-in operation symbol = " * " in the screen display

Entry		Description
INFO	MAIN PGM	Started NC program or NC macro
INFO	MAIN LINE	Line number of the running NC program or NC macro
INFO	MAIN PGMEND	Information about the program end in program run Byte 0/1 00 01 Emergency stop 00 02 Positioning error 00 03 Programmed stop 00 04 Block end in single block mode 00 05 Geometry error 00 06 END PGM, M02 00 07 TNC STOP button 00 08 Data transmission error (RS-422/RS-232) Byte 2/3 xx xx Internal error class Byte 4...7 xx xx xx xx Internal error code Byte 8 ... 11 xx xx xx xx Line number Byte 12 ... 15 xx xx xx xx Reserved In addition, when a program is stopped by an error message, the following information is entered in the log: NC program, line number, actual position, datum, datum shifts, tool number
INFO	MAIN MACEND	Information about the end of an NC macro Byte 0/1 00 01 Emergency stop 00 02 Positioning error 00 03 Programmed stop 00 04 Block end in single block mode 00 05 Geometry error 00 06 END PGM, M02 00 07 TNC STOP button 00 08 Data transmission error (RS-422/RS-232) Byte 2/3 xx xx Internal error class Byte 4...7 xx xx xx xx Internal error code

Entry			Description
INFO	MAIN PATH	PLCEDIT	File for PLC editor
		NCEDIT	File for NC Editor
		RUNPGM	Main program for program run
		RUNPALET	Pallet table for program run
		RUNDATUM	Datum table for program run
		RUNTOOL	Tool table for program run
		RUNTCH	Pocket table for program run
		SIMPGM	Main program for program test
		SIMDATUM	Datum table for program test
		SIMTOOL	Tool table for program test
		RUNBRKPGM	Stopping point for block scan
		SIMBRKPGM	Stopping point for program test
		RUNPRINT	Path for FN15: PRINT for program run
		SIMPRINT	Path for FN15: PRINT for program test
		MDIPGM	File for positioning with manual data input
		NCFMASK	Mask for file management in the NC area
		PLCFMASK	Mask for file management in the PLC area
		EASYDIR	Paths for standard file management
		TCHPATH	Datum table for manual measurement
		SIMTAB	Freely definable table in program test
		RUNTAB	Freely definable table in program run
		KINTAB	Active kinematics table
INFO	MAIN NCEVENT		Entries via **FN38: SEND** from the **Program Run, Full Sequence** or **Program Run, Single Block** operating modes
	MAIN NCTEVENT		Entries via **FN38: SEND** from the **Test Run** operating modes
INFO	MAIN BUTTON MOUSE "x" "y" "z" "a" "b"		Recording of mouse movements/buttons ■ "x": P = Press, R = Release ■ "y": L = Left button, R = Right button ■ "z": Key pressed simultaneously N = None, S = SHIFT, C = CTRL, A = ALT, W = Windows, L = Left button, M = Middle button, R = Right button ■ "a": Position of the mouse pointer in X ■ "b": Position of the mouse pointer in Y
INFO WARNING ERROR	PLC <log identifier>		Entries through PLC Modules 9275 and 9276
INFO	SYS	SHUTDOWN	Control was shut down
		REBOOT-TNC	Control was rebooted (automatically)
		REBOOT-BIOS	Control was rebooted (automatically)

Entry			Description
INFO[a]	REMO A_LG		Log in with LSV2 protocol
	REMO A_LO		Log out with LSV2 protocol
	REMO Delete		Deletion of a file via the LSV2 protocol
	REMO Receive		Reception of a file via the LSV2 protocol
	REMO C_LK		LSV2 protocol: Locking and releasing the keyboard; the key codes between locking and releasing are sent via LSV2 protocol
INFO	SOKY	KEYSOURCE:	Source of information on a key: ■ KEYBOARD ■ PLC ■ PLCNCSTART ■ HANDWHEEL ■ LSV2 ■ KEYLOGGER
		PROCESS:	Name of the target process to which the key information is sent.
		IDENT:	Control-internal name of the soft key
		SOFTKEY:	Name of the BMX image file of the soft key
		OVERLAY:	Current overlay number of the soft key
		Autorepeat ...	Status information on the Autorepeat function (start, stop, waiting times, ...)
		Key Logger:	Status information on key recording (start, stop, repeat, ...)
INFO	SYS WINEVENT FILEMAN.STARTUP.READY		File Manager (PGM MGT) started
ERROR	"xxx": "yyy"		"xxx": Name of the control process that enters the information in the log ■ PLC ■ SYS ■ MAIN ■ REMO ■ CTRL ■ SMARTNC ■ SMARTNC_LDF ■ BDEHAN ■ SOKY ■ GEO ■ PYTHON ■ FIXTURE ■ FILEMAN "yyy": Optional information: Name of the process causing the error

a. For test purposes, all LSV-2 telegrams can be entered in the log. This function must be enabled with the LSV-2 TELEGRAM OFF/ON soft key.

Example of a standard log entry

The following example shows possible entries in the log:

```
STIB:      ON                              20.01.2001 14:01:42
INFO:      MAIN PGM                        20.01.2001 14:01:42
           TNC:\mercedes\Mbprog5a.I
INFO:      MAIN LINE                       20.01.2001 14:01:42
           0
STIB:      OFF                             20.01.2001 14:01:44
INFO:      MAIN PGMEND                     20.01.2001 14:01:44
           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
           00 02 00 02 00 00 00 33 00 00 00 11
           Byte 0                                    Byte 15

INFO:      MAIN PGMEND                     20.01.2001 14:01:44
           Stop reason: Positioning error
           Error      : 51
           Error class: "Positioning"
           NC program : TNC:\mercedes\Mbprog5a.I line 17
INFO:      MAIN PGMEND                     20.01.2001 14:01:44
           Actual pos.:
            X = 1.8251
            Y = -9.2372
              Z =   45.0030
            A = 0.0000
            B = 359.9999
            C = 0.0000
            W = 65.9894
           Preset     : (Range = 0)
            X = -8.6201
            Y = 7.5515
              Z =   -1835.3142
            A = 0.0000
            B = 0.0000
            C = 0.0000
            W = -178.8965
           Datum shift:
            X = 0.0000
            Y = 0.0000
```

Write to the log

The log can be written to by the PLC or from an NC program.

Write data **from an NC program** into the log:

- Use the **FN38: SEND** function in the NC program. It is available after you have entered the code number 555343.

Example for programming a block with **FN38: SEND:**

```
...
FN38: SEND /"Q-Parameter Q1: %f  Q2: %f" /+Q1 /+Q2
...
```

Entry in the log from the **Program Run, Single Block** or **Program Run, Full Sequence** operating mode:

```
INFO:              MAIN NCEVENT             <Date and time>
                   Q parameter Q1: <Value Q1> Q2: <Value Q2>
```

Entry in the log from the **Test Run** operating mode:

```
INFO:              MAIN NCTEVENT            <Date and time>
                   Q parameter Q1: <Value Q1> Q2: <Value Q2>
```

Write data for diagnostic purposes **from the PLC** into the log:

- With Module 9275 you can write ASCII data into the log.
- With Module 9276 you can write the contents of the operands into the log.

Note

Do not use Modules 9275 and 9276 in the PLC program as shipped. Instead, use them only for debugging. Otherwise the processing times could be increased and the hard disk could be written to unnecessarily, so that the log can no longer fulfill its function of recording keystrokes and error messages.

Additional log

Software 340 49x-03 introduced a second log parallel to the existing log . It is in the battery-buffered RAM, and so enables access to the log data on the control even without a hard disk. Enter the code number **LOGBOOK1** to output it and save it as an ASCII file, just as the standard log.
This log is not available for the programming station.

Module 9275 Write ASCII data into the log
With Module 9275 you can write ASCII data into the log. For later editing the entry can be given an identifier.

Call:

PS	B/W/D/K/S<Log entry>	
		–1: No entry
PS	B/W/D/K/S<Log identifier>	
		–1: No entry
PS	B/W/D/K	<Priority>
		0: Information
		1: Warning
		2: Error
CM	9275	

Error recognition:

Marker	Value	Meaning
M4203	0	Entry was written
	1	Error code in W1022
W1022	1	Invalid priority
	2	Invalid string number or invalid immediate string
	12	No string end identifier
	20	Module was not called in a spawn or submit job

Module 9276 Write operand contents into the log

With this module you can write the contents of operands (inputs, outputs, markers, bytes, words, double words, timers, counters) into the log. For later editing the entry can be given an identifier.

Call:

PS	B/W/D/K	<Identifier for operand name> 0: M (marker) 1: I (input) 2: O (output) 3: C (counter) 4: T (timer) 5: B (byte) 6: W (word) 7: D (double word)
PS	B/W/D/K	<Address of the first operand>
PS	B/W/D/K	<Number of operands>
PS	B/W/D/K/S	<Log identifier> –1: No entry
PS	B/W/D/K	<Priority> 0: Information 1: Warning 2: Error
CM	9276	

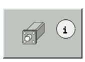
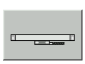
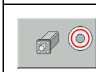
Error recognition:

Marker	Value	Meaning
M4203	0	Entry was written
	1	Error code in W1022
W1022	1	Invalid priority
	2	Invalid identifier for operand name
	3	Invalid first operand address
	4	Sum of first operand address and number of operands invalid
	5	Address is not a word/double-word address
	12	No string end identifier
	20	Module was not called in a spawn job or submit job
	36	Entry in the log was truncated after 210 characters

8.1.30 Diagnostic functions

The iTNC 530 up to and including software version 340 49x-03 provided the diagnostic functions for error diagnosis, which were available after pressing the DIAGNOSIS soft key (see table below).

The following table compares the organization of the diagnostic functions in software 340 49x-03 and lower with the organization in software 340 49x-04 and higher.

Soft key	Soft key	Function
PROFIBUS DIAGNOSIS	Replaced by BUS DIAGNOSIS soft key	
DRIVE DIAGNOSIS	Remains available	
	OSCI	The integrated oscilloscope is started.
	DSP	Replaced by DRIVE DIAG
	AMP	
		
		
	The following soft keys appear only if the **Power Interrupted** message was not acknowledged, and if the code number 688379 or 807667 was entered.	
	I CONTROL	The integrated oscilloscope for commissioning the current controller is started. For more detailed information, refer to the Technical Manual of your control.
	TEST	Replaced by DRIVE DIAG. Remains available (only after the code number has been entered).
		
HEROS DIAGNOSIS	The file TNC:\herosdiagnose.txt is created after pressing this soft key. HEIDENHAIN uses this file for diagnosis of the operating system.	

The iTNC 530 with software version 340 49x-04 and higher provides the following diagnostic functions for error diagnosis.

To call the diagnostic functions:

- While in the **Programming and Editing** operating mode, press the MOD key.
- Press the DIAGNOSE soft key.

The following diagnostic functions are available:

Soft key	Soft key	Function
BUS DIAGNOSIS	After pressing this soft key, you can test various HSCI and Profibus settings, provided that you are using one of the two bus systems.	
	PROFIBUS	After pressing this soft key, you can test various Profibus settings, provided that you are using a Profibus system.
	HSCI DIAGNOSIS	After pressing this soft key, you can test various HSCI settings, provided that you are using an HSCI system.
DRIVE DIAGNOSIS	Various drive diagnosis functions can be selected after pressing this soft key. Before selecting the diagnostic function, under **Supply unit** you must select the power supply unit being used, so that the signals present are not interpreted as errors.	
	OSCI	The integrated oscilloscope is started.
	DRIVEDIAG	The diagnosis tool DriveDiag is opened, see page 1278.
	The following soft keys appear only if the **Power Interrupted** message was not acknowledged, and if the code number 688379 or 807667 was entered.	
	I CONTROL	The integrated oscilloscope for commissioning the current controller is opened.
HEROS DIAGNOSIS	The file TNC:\herosdiagnose.txt is created after pressing this soft key. HEIDENHAIN uses this file for diagnosis of the operating system.	

DriveDiag

The diagnosis tool DriveDiag is available on iTNCs 530 with software version 340 49x-04 and higher.

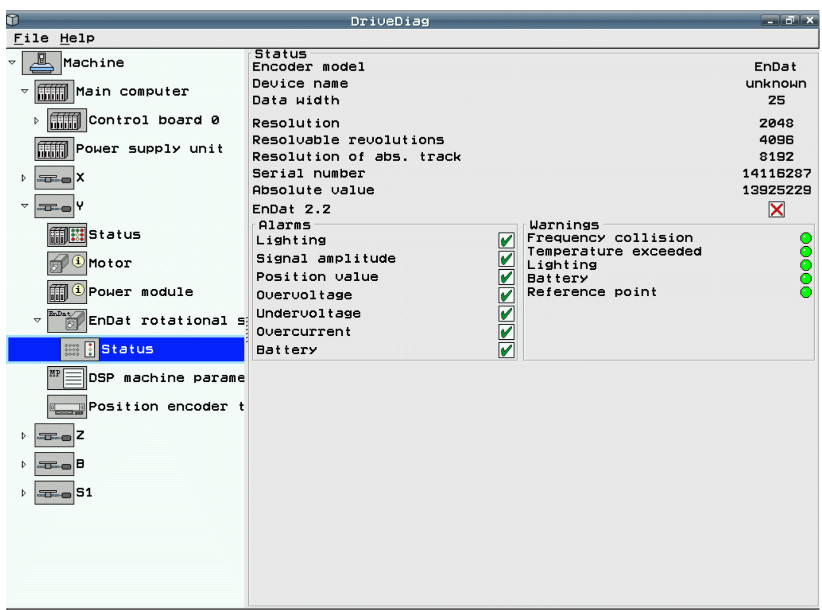
To start DriveDiag:

- While in the **`Programming and Editing`** operating mode, press the MOD key.
- Press the DIAGNOSIS soft key
- Press the DRIVE DIAGNOSIS soft key.
- Press the DRIVE DIAG soft key.

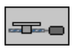
This opens an additional task for the DriveDiag diagnosis tool.

The structure of the drive system is displayed on the left side of the screen in a tree structure. Detailed information on the currently selected component is displayed on the right side of the screen.

To navigate within the tree structure (left side of screen) and the tabs (right side of screen), use the arrow keys. Use the ENT key to move from the left side of the screen to the right side of the screen. Use the END key to move from the right side of the screen back to the left side of the screen. If the cursor is already located on the left side of the screen, you can exit DriveDiag with the END key.



To switch back to the screen of the control or to DriveDiag, press the screen switchover key.

Symbol	Symbol	Function
	**Machine**: Higher-level folder containing all components of the complete machine ■ **Connection:** Internal IP address (127.0.0.1)	
		**Main computer**, see page 1279
		**Power supply unit**, see page 1285 ■ Information about ID label and status
		Axis (e.g. **X, Y, Z**, etc.), see page 1287 Higher-level folder containing all axis components

Main computer

Symbol	Symbol	Function
	**Main computer** **Version** tab: ■ **Control model:** Type of control ■ **NC software:** Installed NC software version ■ **PLC software:** PLC software being used	
	Main computer **Status** tab: ■ Status information about various signals, see page 1281	

Symbol	Symbol	Function
		**`Drive control board x`** **`Version`** tab: ■ **`Speed controller software:`** Installed speed-controller software version ■ **`Current-controller software:`** Installed current-controller software version ■ **`Hardware code:`** Additional information for identifying the hardware (only CC 422). ■ **`Version of additional info:`** Date and time of installation ■ **`Degree of support:`** Degree of support for controller software ■ **`SG software available:`** Information about whether the installed software supports functional safety ■ **`SG software active:`** Functional safety is activated/deactivated
		`Drive control board x` **`Voltages and currents`** tab: ■ **`Supply voltage +5V:`** Current value of voltage in [V] ■ **`DC-link voltage:`** Current value of voltage in [V] ■ **`DC-link current:`** Current value of current in [A] ■ **`DSP computer board temp.`**: Current temperature value on the DSP computer board in [°C] ■ **`Supply voltage +15V:`** Current value of voltage in [V] ■ **`Supply voltage –15V:`** Current value of voltage in [V] ■ **`Supply voltage +3.3V:`** Current value of voltage in [V] ■ **`Auxiliary voltage UL:`** Current value of voltage in [V] ■ **`Auxiliary voltage UH:`** Current value of voltage in [V]
		`Drive control board x` **`Status`** tab: ■ Status information about various signals, see page 1283

Meanings of the signals under "main computer / status"

Signal	Meaning	Colors
External enabling signals		
Acknowledgment: Control is ready (–NE1)	The $\overline{NE1}$ signal (emergency stop input 1, MC) is active if a 0 level is present (low active). For the iTNC 530 the corresponding input is at connector X42/ I3 (PLC input), and is looped to the MC as a hardware line.	■ Gray: No information about the signal available ■ Green: Signal is not active, enable ■ Red: Signal is active, no enabling
External signals from MC		
Ref. signal of spindle (X30)	For more detailed information, refer to "Input: Spindle Reference Signal" in the Technical Manual.	■ Gray: No information about the signal available ■ Green: Signal is active ■ Red: Signal is not active
Trigger signal (X12) of TS touch probe	For more detailed information, refer to "X12: Connecting the Touch Probe" in the Technical Manual.	■ Gray: No information about the signal available ■ Green: Signal is active ■ Red: Signal is not active
Battery (X12) of TS touch probe	For more detailed information, refer to "X12: Connecting the Touch Probe" in the Technical Manual.	■ Gray: No information about the signal available ■ Green: Signal is not active, warning ■ Red: Signal is active, no warning
TS touch probe is ready (X12)	For more detailed information, refer to "X12: Connecting the Touch Probe" in the Technical Manual.	■ Gray: No information about the signal available ■ Green: Signal is active ■ Red: Signal is not active
Trigger signal (X13) of TT probe	For more detailed information, refer to "X13: Connecting the Touch Probe" in the Technical Manual.	■ Gray: No information about the signal available ■ Green: Signal is active ■ Red: Signal is not active
TT touch probe ready (X13)	For more detailed information, refer to "X13: Connecting the Touch Probe" in the Technical Manual.	■ Gray: No information about the signal available ■ Green: Signal is active ■ Red: Signal is not active

Signal	Meaning	Colors
MC internal signals and status		
Current controller commissioning mode	The signal is active if the control is in the operating mode for adjusting the current controller.	■ Gray: No information about the signal available ■ Green: Operating mode for adjusting the current controller is active ■ Red: Operating mode for adjusting the current controller is not active
Power interruption acknowledged	The signal is active if the **Power interrupted** message has been acknowledged with the CE key.	■ Gray: No information about the signal available ■ Green: Power interruption was acknowledged ■ Red: Power interruption was not acknowledged

Meanings of the signals under "drive control board / status"

Signal	Meaning	Colors
External enabling signals		
Drive enable (–NE2)	The $\overline{NE2}$ signal (emergency stop input 2, CC) is active if a 0-level is present (low active). For the iTNC 530 the corresponding input is at connector X42/I32 (PLC input), and is looped to the CC as a hardware line.	■ Gray: No information about the signal available ■ Green: Signal is not active, enable ■ Red: Signal is active, no enabling
Powerfail	The $\overline{PF}$ signal shows the status of the "effective" powerfail signal for the drive controller. The signal is the result of gating the $\overline{PF.PS.ZK}$ (dc-link powerfail) and $\overline{PF.PS.AC}$ (AC fail) signals. The gating process can be defined in the machine parameters and in the PLC.	■ Gray: No information available ■ Green: Enable, $\overline{PF}$ is inactive (1-level) ■ Red: $\overline{PF}$ is active (0-level): the dc-link voltage has decreased below a permissible (inverter-specific) level or the phase monitoring responded; no enabling
X50 Machine On	No longer relevant in the current HEIDENHAIN controls.	■ Gray: No information available
MC is ready (–WD)	This signal shows that the MC is ready for control. This signal is a possible reason that the power module was switched off via SH1.	■ Gray: No information available ■ Green: Enable: $\overline{ME}$ not active (1-level) ■ Red: No enabling: $\overline{WD1}$ is active (0-level), the MC's watchdog is not retriggered. This signal is relayed to the inverter as $\overline{SH1}$ ($\overline{SH1}$ also has other signal sources).
Powerfail (DC)	The signal is generated at the inverter, and is led via the supply bus to the drive controller. The input at the drive controller is displayed. Depending on the wiring, either this signal or **Powerfail (AC)** is relayed on the drive control board to the powerfail signal.	■ Gray: No information available ■ Green: Enable: $\overline{\text{Powerfail (DC)}}$ is inactive (1-level) ■ Red: No enabling: $\overline{\text{Powerfail (DC)}}$ is active (0-level): the dc-link voltage has decreased below a permissible (inverter-specific) level.
Powerfail (AC)	The signal is generated at the inverter, and is led via the supply bus to the drive controller. The input at the drive controller is displayed. Depending on the wiring, either this signal or **Powerfail (DC)** is relayed on the drive control board to the powerfail signal. **Powerfail (AC)** does not exist for all supply units (e.g. not for UV 130).	■ Gray: No information available ■ Green: Enable: $\overline{\text{Powerfail (DC)}}$ is inactive (level 1) ■ Red: No enabling: $\overline{\text{Powerfail (AC)}}$ is active (0-level), phase monitoring responded, at least one power supply phase failed

Signal	Meaning	Colors
Internal enabling signals		
Switch-off (speed 0)	This signal causes all axes to be switched off or locked at 0-level. Thus the drives of all control boards are switched off or locked.	■ Gray: No information about the signal available ■ Green: Enable: –N0 inactive (1-level) ■ Red: No enabling: –N0 active (0-level); all drives are locked
CC controller ready	If no error is present in the drive controller and the controller unit was started, "ready for control" is reported.	■ Gray: No information about the signal available ■ Green: Enable: CC is ready for control ■ Red: No enabling
Clearable DSP error	Clearable DSP errors are 2nd class errors (such as motor temperature). The CC can only resume control after the error has been cleared (by pressing the CE key).	■ Gray: No information about the signal available ■ Green: Enable: There is no (clearable) 2nd class error ■ Red: No enabling: A 2nd class error is present
Watchdog current controller	This signal is activated by the current controller's watchdog. It affects $\overline{\text{SH2}}$ on the power stage via the PWM interface.	■ Gray: No information about the signal available ■ Green: Enable: Current controller watchdog OK ■ Red: No enabling: Current controller watchdog is active (0-level). No pulse release from the current controller via the PWM interface

Power supply unit

Symbol	Symbol	Symbol	Function
	**Power supply unit** **ID Label**, see page 1297 tab: ■ **Plug and Play:** Information about whether the device is Plug and Play capable ■ **Device:** Designation of the device ■ **Model:** Designation of the device type ■ **Serial number:** Serial number of the device ■ **ID number:** HEIDENHAIN ID of the device		
	Power supply unit **Status** tab: ■ Status information about various signals, see page 1281		

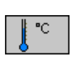
Meanings of the signals under "supply unit / status"

Signal	Meaning	Colors
DC-link voltage	Current value of voltage in [V]	
DC-link current	Current value of current in [A]	
Powerfail	The $\overline{\text{PF}}$ signal shows the status of the "effective" powerfail signal for the drive controller. The signal is the result of gating the $\overline{\text{PF.PS.ZK}}$ (dc-link powerfail) and $\overline{\text{PF.PS.AC}}$ (AC fail) signals. The gating process can be defined in the machine parameters and in the PLC.	■ Gray: No information available ■ Green: Enable, $\overline{\text{PF}}$ is inactive (level 1) ■ Red: $\overline{\text{PF}}$ is active (0-level): the dc-link voltage has decreased below a permissible (inverter-specific) level or the phase monitoring responded; no enabling
Powerfail (DC)	The signal is generated at the inverter, and is led via the supply bus to the drive controller. The input at the drive controller is displayed. Depending on the wiring, either this signal or **Powerfail (AC)** is relayed on the controller PCB to the powerfail signal.	■ Gray: No information available ■ Green: Enable: $\overline{\text{Powerfail (DC)}}$ is inactive (level 1) ■ Red: No enabling: $\overline{\text{Powerfail (DC)}}$ is active (0-level): the dc-link voltage has decreased below a permissible (inverter-specific) level.
Powerfail (AC)	The signal is generated at the inverter, and is led via the supply bus to the drive controller. The input at the drive controller is displayed. Depending on the wiring, either this signal or **Powerfail (DC)** is relayed on the drive control board to the powerfail signal. **Powerfail (AC)** does not exist for all supply units (e.g. not for UV 130).	■ Gray: No information available ■ Green: Enable: $\overline{\text{Powerfail (DC)}}$ is inactive (level 1) ■ Red: Not enabled: $\overline{\text{Powerfail (AC)}}$ is active (level 0), phase monitoring responded, at least one power supply phase failed

Signal	Meaning	Colors
`dc-link voltage>>`	The signal reports the status of the dc-link voltage: Either it is OK or too high. This signal also switches off all power modules (via the unit bus). Possible error causes: ■ Missing or faulty braking resistor ■ Excessive braking power	■ Gray: No information about the signal available ■ Green: DC-link voltage OK ■ Red: DC-link voltage too high
`Temperature`	The signal reports the status of the heat sink temperature in the inverter: Either it is OK or too high.	■ Gray: No information about the signal available ■ Green: Temperature OK ■ Red: Temperature too high
`DC-link current >>`	The signal reports the status of the dc-link current: Either it is OK or too high. Both the positive and the negative dc-link currents are evaluated.	■ Gray: No information about the signal available ■ Green: DC-link current OK ■ Red: DC-link current too high
`Power supply unit ready`	The signal reports the ready status of the supply unit: Supply unit OK, main contactor on, or supply unit not ready.	■ Gray: No information about the signal available ■ Green: Supply unit OK, main contactor on ■ Red: Power supply unit not ready
`Ground fault`	The signal reports the status of the leakage current monitoring: Either it is OK or too high or there is a ground fault.	■ Gray: No information about the signal available ■ Green: Leakage current is OK ■ Red: Leakage current too high or ground fault
`Reset from CC to UV`	Signal –RES.LE. Reset from controller unit to supply module. Resets the error memory in the supply module.	■ Gray: No information about the signal available ■ Green: Reset signal is not active ■ Red: Reset signal is active

X, Y, Z, etc. (axis)

Symbol	Symbol	Symbol	Function
	X, Y, Z, etc. (axis) Higher-level folder containing all axis components		
		Status **Speed Controller** tab: ■ Status information about various signals, see page 1290 **Status** **Position Controller** tab: ■ Status information about various signals, see page 1293 **Status** **PLC** tab: ■ Status information about various signals, see page 1294	
		Motor **ID Label**, see page 1297 tab: ■ **Plug and Play:** Information about whether the device is Plug and Play capable ■ **Device:** Designation of the device ■ **Model:** Designation of the device type ■ **Serial number:** Serial number of the device ■ **ID number:** HEIDENHAIN ID of the device ■ **Brake present:** Information about whether the motor has a brake	
		Motor **Motor Data** tab: ■ Information about motor-specific data from the motor table	
		Motor **Result** tab: ■ Information about motor data derived from the motor table or from the machine parameter file • **Motor constant [Nm/A]:** Internal reference value that indicates the ratio of torque to current (= torque constant). • **Voltage constant:** Internal reference value that indicates the ratio of no-load voltage to rated speed. • **Field angle [°]:** Value converted to [°] for the corresponding entry from the MP file (MP2256.x).	

Symbol	Symbol	Symbol	Function
		**Power module** **ID Label**, see page 1297 tab: ■ **Plug and Play:** Information about whether the device is Plug and Play capable ■ **Device:** Designation of the device ■ **Model:** Designation of the device type ■ **Serial number:** Serial number of the device ■ **ID number:** HEIDENHAIN ID of the device	
		Power module **Status** tab: ■ Status information about various signals, see page 1296	
		**EnDat rotational speed encoder** Depending on the functions provided by the encoder, the submenus status, diagnosis and temperature are available.	
		**EnDat rotational speed encoder** Depending on the functions provided by the encoder, the submenus status, diagnosis and temperature are available.	
		**EnDat position encoder** Depending on the functions provided by the encoder, the submenus status, diagnosis and temperature are available.	
		**EnDat 2.2 position encoder** Depending on the functions provided by the encoder, the submenus status, diagnosis and temperature are available.	
			**Status** ■ Status information about various signals, see page 1295
			**Diagnosis** The function reserves are evaluated, which provide an exact overview of the current status of the encoder. The three values displayed inform the user at a glance of the encoder performance. In the yellow range, fault-free operation is no longer guaranteed. The encoder should be checked. If required, assistance can be provided by your HEIDENHAIN service agency.
			**Temperature** The temperature information provided by the temperature sensors in the encoder and the motor is displayed.

<table>
<tr><th>Symbol</th><th>Symbol</th><th>Symbol</th><th>Function</th></tr>
<tr><td rowspan="2"></td><td rowspan="2"></td><td colspan="2">Motor encoder test
(available only after the code number 688379 or 807667 has been entered)

Diagram tab:

After the test has been completed, the result is displayed in a diagram. Two green lines in the diagram mark the minimum and maximum height of the speed encoder signals. The red line shows the measured signals.</td></tr>
<tr><td colspan="2">Motor encoder test

Overview tab:

Begin the test by pressing the START MEASUREMENT button.

■ Status information about various signals, see page 1296

The plain text displayed in the lower part of the screen provides additional information about the test results.</td></tr>
<tr><td rowspan="2"></td><td rowspan="2"></td><td colspan="2">Position encoder test

Diagram tab:

After the test has been completed, the result is displayed in a diagram. Two green lines in the diagram mark the minimum and maximum height of the position encoder signals. The red line shows the measured signals.</td></tr>
<tr><td colspan="2">Position encoder test

Overview tab:

Begin the test by pressing the START MEASUREMENT button.

■ Status information about various signals, see page 1296

The plain text displayed in the lower part of the screen provides additional information about the test results.</td></tr>
<tr><td></td><td></td><td colspan="2">Drive test
(available only after the code number 688379 or 807667 has been entered)

Test tab:

Begin the test by pressing the START MEASUREMENT button.

■ Status information about various signals, see page 1297

The plain text displayed in the lower part of the screen provides additional information about the test results.</td></tr>
</table>

Meanings of the signals under "status / speed controller"

Signal	Meaning	Colors
`MC enabling marker`	The MC can accelerate the switch-off via this marker.	■ Gray: No information about the signal available ■ Green: Enable ■ Red: No enabling
`X150/X151 Drive enabling`	The signal shows the enabling status for the "X150/X151" switch-off. The signal is formed from the status of the inputs X150/X151 and the setting in MP2040.x.	■ Gray: No information about the signal available ■ Green: Enable: There is currently no switch-off via X150/X151 ■ Red: No enabling: The drive is currently switched off or locked via X150/X151
`Drive enabling from speed controller`	This signal is a group signal for all hardware enabling signals from the speed controller.	■ Gray: No information about the signal available ■ Green: Enable ■ Red: No enabling
`Drive enabled by software`	This signal is a group signal for all hardware enabling signals from the software.	■ Gray: No information about the signal available ■ Green: Enable ■ Red: No enabling
`Internal drive status`	Gating of all ready signals (external, internal and software)	■ Gray: No information about the signal available ■ Green: Enable ■ Red: No enabling
`Power module active (–SH2)`	The signal shows the status of the $\overline{SH2}$ line to the power module. The CC activates/deactivates this line for switching off the power module.	■ Gray: No information about the signal available ■ Green: Enable: The $\overline{SH2}$ signal is inactive ■ Red: No enabling: The $\overline{SH2}$ signal is active
`Current controller active`	The signal shows the status of the current controller. The current controller is either switched on (in control) or switched off.	■ Gray: No information about the signal available ■ Green: Enable: Current controller is on (in control) ■ Red: Do not enable: Current controller is off
`Speed controller is active`	The signal shows the status of the speed controller. The speed controller is either switched on (in control) or switched off.	■ Gray: No information about the signal available ■ Green: Enable: Speed controller is on (in control) ■ Red: Do not enable: Speed controller is off

Signal	Meaning	Colors
`Rotor position captured`	This signal gives information about determining the field angle: **Drive is not oriented:** ■ Motor with rotary encoder without Z1 track (incl. linear motors) before the first "Drive on" status ■ Non-aligned rotary encoder with EnDat interface (incl. linear motors), if the field angle has not yet been determined **Drive is roughly oriented:** ■ Motor with rotary encoder without Z1 track (incl. linear motors) after the first "Drive on" status ■ Motor with rotary encoder with Z1 track after it has been read **Drive is oriented:** ■ Motor with rotary encoder with Z1 track after traversing the reference mark ■ Aligned rotary encoder with EnDat interface immediately after switch-on ■ Non-aligned rotary encoder with EnDat interface immediately if the field angle has already been determined ■ Motor with rotary encoder without Z1 track after traversing the reference mark if the field angle has already been determined	■ Gray: No information about the signal available ■ Green: Field angle has been determined ■ Yellow: Field angle has been roughly determined ■ Dark gray: Field angle has not been determined
`Switching on speed controller`	This signal indicates whether the speed encoder has been switched on since the last time the **`Power interrupted`** message was active. Status changes are saved.	■ Gray: No information about the signal available ■ Dark gray: Speed controller has not been switched on yet (since power interruption) ■ Green: Speed controller is currently being switched on ■ Yellow: Speed controller was switched on once (this does not mean, however, that the speed controller is currently on)

Signal	Meaning	Colors
Switching off speed controller	This signal indicates whether the speed encoder has been switched off since the last time the **Power interrupted** message was active. Status changes are saved.	■ Gray: No information about the signal available ■ Dark gray: Speed controller has not been switched off yet (since power interruption) ■ Green: Speed controller is currently being switched off ■ Yellow: Speed controller was switched off once (this does not mean, however, that the speed controller is currently off)
Brake released	This signal shows the status of the motor brake signal on the PWM bus. This signal is led from the power module via a relay to the motor.	■ Gray: No information about the signal available ■ Green: Brake released ■ Red: Brake engaged
I2t warning	This signal shows the current and stored status of the I^2t monitoring.	■ Gray: No information about the signal available ■ Green: No I^2t warning up to now ■ Yellow: There has already (since switch-on) been an I^2t warning, but there is no current warning ■ Orange: There is a current I^2t warning
Torque ripple	The signal indicates the status of torque-ripple compensation.	■ Gray: No information about the signal available ■ Green: Compensation is active ■ Red: Compensation is not active
Acceleration feedforward control	The signal shows the status of acceleration feedforward control (MP1392 or 1391.1, depending on the selected mode of operation).	■ Gray: No information about the signal available ■ Green: Feedforward control is active ■ Red: Feedforward control is not active

Meanings of the signals under "status / position controller"

Signal	Meaning	Colors
`Position control loop closed`	Corresponds to PLC word W1024 for a specific axis. Information about whether the position control loop is closed or open.	■ Gray: No information about the signal available ■ Green: Position control loop closed ■ Red: Position control loop open
`Position-controlled`	The signal indicates whether the axis is operated with position feedback control or speed feedback control.	■ Gray: No information about the signal available ■ Green: Axis is operated with position feedback control ■ Red: Axis is operated with speed feedback control
`Nominal position value filter`	The signal indicates whether the nominal position value filters are active.	■ Gray: No information about the signal available ■ Green: Nominal position value filter is active ■ Red: Nominal position value filter is not active
`Position controller limit`	The signal indicates whether "eliminate following error" is active.	■ Gray: No information about the signal available ■ Green: Output limit is active ■ Red: Output limit is not active
`CC monitoring active`	The signal indicates whether following-error monitoring is active (can be deactivated by PLC, W1042).	■ Gray: No information about the signal available ■ Green: Following-error monitoring is active ■ Red: Following-error monitoring is not active
`Touch probe is active`	The signal indicates whether probing is active.	■ Gray: No information about the signal available ■ Green: Probing is active ■ Red: Probing is not active
`Reference pulse active`	The signal indicates whether the MC expects a reference pulse (host command).	■ Gray: No information about the signal available ■ Green: Expectation of reference pulse is active ■ Red: Expectation of reference pulse is not active

Signal	Meaning	Colors
`Velocity feedforward control`	The signal indicates whether the position controller is working with feedforward control or with following error. Acceleration feedforward control can be switched on/off cyclically by the MC. Corresponds to parameter MP 1392 or MP1391.0 in the MP file, depending on the current mode of operation.	■ Gray: No information about the signal available ■ Green: Operation with feedforward control ■ Red: Operation with following error

Meanings of the signals under "status/PLC"

Signal	Meaning	Colors
`Axis in position (PLC)`	If the axes have reached the positioning window after a movement, the status is shown in W1026.	■ Green: Axis in position ■ Yellow: Axis not in position
`Position control loop closed (PLC)`	Position control loop closed (W1040 inverted). By setting W1040 in the PLC, the position control loop is opened by the PLC program.	■ Green: Position control loop closed ■ Yellow: Position control loop open
`Axis enabled (PLC)`	W1024 shows if the position control loop is open or closed, and if the axis has been enabled.	■ Green: Axis not enabled ■ Yellow: Axis enabled
`Axis in motion (PLC)`	During axis movement, the NC sets the bits in W1026.	■ Green: Axis in motion ■ Yellow: Axis at standstill

Meanings of the signals under "EnDat/status"

Signal	Meaning	Colors
`Encoder model`	Designation of the encoder model	
`Device name`	Name of the encoder	
`Data width`	Number of bits of position value	
`Resolution`	Speed encoder Number of signal periods per revolution Linear encoder: Signal period in nanometers Value 0: The encoder is a pure serial encoder.	
`Distinguishable revolutions`	Information about multiturn encoders. Number of max. distinguishable revolutions.	
`Resolution of abs. track`	Speed encoder Number of measuring steps per revolution Linear encoder: Number of measuring steps per nanometer	
`Serial number`	Serial number of encoder	
`Absolute value`	Current absolute value	
`EnDat 2.2`	Information about whether the encoder is EnDat 2.2-compatible.	
`Alarms`	Evaluation of different types of error information from the encoder. If the encoder reports an encoder error, fault-free encoder function is no longer guaranteed. The encoder must be checked and replaced if necessary. If required, assistance can be provided by your HEIDENHAIN service agency.	■ Green: No error ■ Red: Encoder reports an error
`Warnings`	Evaluation of different types of status information from the encoder. If the encoder issues a warning, it means that the encoder still functions properly, but one of the functions is operating at its limits. The encoder must be checked and replaced if necessary. If required, assistance can be provided by your HEIDENHAIN service agency.	■ Green: No warning. ■ Yellow: Encoder issues a warning

Meanings of the signals under "power module / status"

Signal	Meaning	Colors
`Power module ready (LT-RDY)`	The power module is ready: ■ Safety relay is on ■ Main contactor is on ■ SH1 (MC) is "High" ■ No error from the power module	■ Gray: No information about the signal available ■ Green: Power module is ready ■ Red: Power module reports "not ready"
`Switch-off power module (IGBT)`	The signal shows that the IGBT in the power module has been switched off.	■ Gray: No information about the signal available ■ Green: No power module switch-off (IGBT) ■ Red: Power module switch-off (IGBT)
`Power module temperature`	The signal reports the status of the heat sink temperature in the power module: Either it is OK or too high.	■ Gray: No information about the signal available ■ Green: Temperature of power module OK ■ Red: Temperature of power module too high

Meanings of the signals under "speed controller / overview"

Information	Meaning	Colors
`Direction` (only available in motor encoder test)	Results of the test for the direction of rotation.	■ Green: Direction of counting and rotation are the same (OK) ■ Red: Direction of counting and rotation differ (error)
`Amplitude`	Result of the test of the speed encoders' signals	■ Green: Signal amplitude within the tolerance (OK) ■ Red: Signal amplitude outside the tolerance (error)

Automated testing of drives

Information	Meaning	Colors
`Phase 1, Phase 2, Phase 3`	Status of the phases in the motor and the supply line	■ Green: Phase is OK ■ Red: Phase error
`Leakage`	Information on a short circuit to ground in the motor, the power module or in the supply line	■ Green: No ground fault ■ Red: Ground fault detected
`Power module`	Status of the power module	■ Green: Power module OK ■ Red: Power module error
`Supply unit`	Status of the supply unit	■ Green: Supply unit OK ■ Red: Supply unit error
`Connection`	Status of the connection control ↔ power module	■ Green: Connection OK ■ Red: Connection error

Electronic ID label

HEIDENHAIN inverter components of type D, as well as HEIDENHAIN synchronous motors with absolute encoders with EnDat interface, are equipped with an electronic ID label.
The product name, the ID number and the serial number are saved in this ID label. These devices are automatically detected when the control is started.

▶ Load the displayed component to the corresponding machine parameter automatically with the **SELECT** soft key.

During every further control restart, the control checks whether the connected units with electronic ID label match the entries in MP2100.x or MP2200.x. If not, an error message might appear and the connected encoder must be transferred to the corresponding machine parameters by soft key.

In exceptional cases, the evaluation of the electronic ID label can be deactivated with MP7690.

MP7690 **Evaluation of the electronic ID labels**
Input: %xx
Bit 0 – HEIDENHAIN power modules
0: Active
1: Inactive
Bit 1 – HEIDENHAIN synchronous motors
0: Active
1: Inactive

8.1.31 Window Manager

The XFCE Window Manager for the graphic interface of the control was introduced with software version 340 49x-04. The XFCE Window Manager is a standard window manager for UNIX-based operating systems. It provides the following functions:

- Optional taskbar for switching between various applications (user interfaces).
- Additional desktop on which the machine tool builder's applications can be run.
- Controlling the focus between NC-software applications and those of the machine tool builder.
- The size and position of pop-up windows can be changed. It is also possible to close, minimize and restore the pop-up windows.

Basic configuration

The window manager is supplied with two default configurations (FULL and SIMPLE). The configuration of the window manager can be changed at any time by using the XFCE configuration dialog.

The SIMPLE configuration neither has a taskbar, nor a background image for the third desktop. As a result, the TNC software with window manager differs only very slightly from the NC software version without window manager. The FULL configuration has a taskbar, however.

HEIDENHAIN recommends using the taskbar of the window manager only on controls equipped with a mouse or a touch pad.

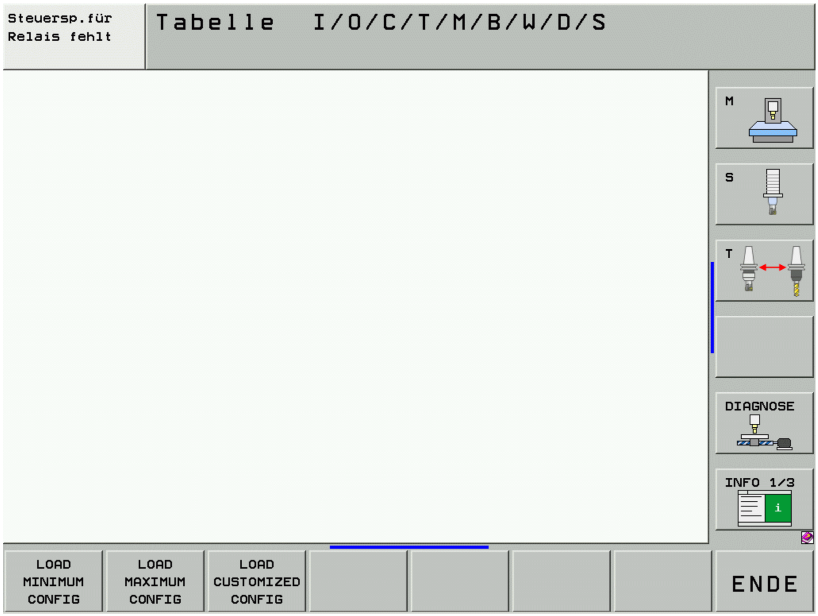
If a software version with the window manager is installed on a control for the first time, the **PLC:\WINDOWMANAGER** directory does not exist. It is created when the NC software is started up for the first time, and the SIMPLE configuration is active.

If you delete the **PLC:\WINDOWMANAGER** directory, the directory is created again during the next startup of the NC software and the SIMPLE configuration is active.

If TNCremoNT is used to transfer the PLC partition between the programming station and a single-processor or dual-processor version, the configuration of the Window Manager (**PLC:\WINDOWMANAGER** directory) is also transferred. The configuration is portable, except for the position of the XFCE taskbar (also called TNC taskbar).
On a dual-processor control, the TNC taskbar is moved to the opposite side of the screen if it displayed at the same position as the Windows taskbar, which is usually at the bottom of the screen.

If a window manager configuration exists, it is retained when the NC software is reinstalled (software update).

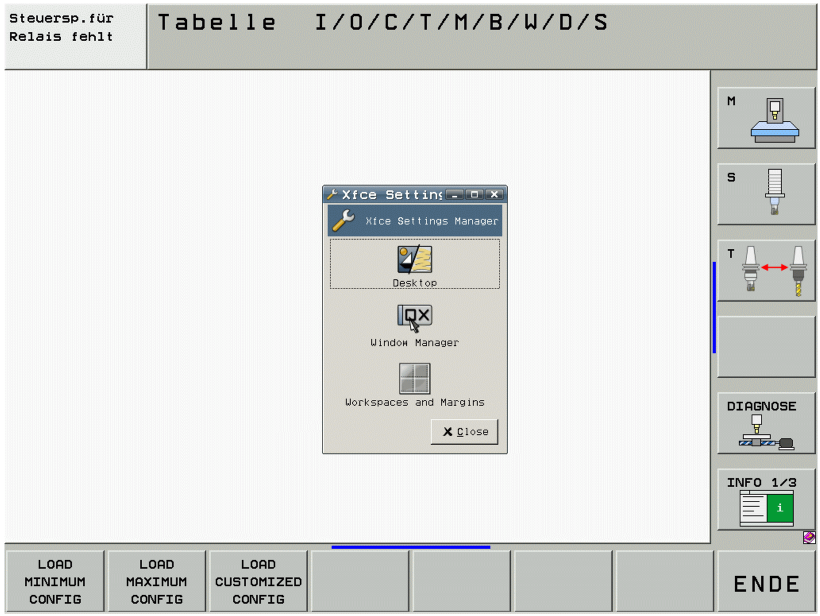
To call the configuration of the Window Manager, press the ADVANCED SETUP soft key and then the CONFIGURE WINDOW-MANAGER soft key.



- If you press the LOAD MINIMUM CONFIG soft key and confirm the following confirmation prompt, the current configuration of the window manager will be overwritten with the SIMPLE configuration.
- If you press the LOAD MAXIMUM CONFIG soft key and confirm the following confirmation prompt, the current configuration of the window manager will be overwritten with the FULL configuration.
- If you do not want to load any configuration, press the LOAD CUSTOMIZED CONFIG soft key. This starts the XFCE configuration dialog in which you can edit the current configuration.

Configuration with XFCE

If you want to create your own configuration containing the TNC taskbar, you should first activate the FULL configuration. If you do not want your configuration to contain the TNC taskbar, first select the SIMPLE configuration.
Do not start the XFCE configuration until you have activated the respective basic configuration.
The two supplied basic configurations SIMPLE and FULL are only changed temporarily when you configure the window manager using the LOAD CUSTOMIZED CONFIG soft key. The files containing the two basic configurations are stored in the control and remain unchanged. This means that you can call the basic configurations at any time by pressing the LOAD MINIMUM CONFIG soft key or the LOAD MAXIMUM CONFIG soft key.
Direct saving of your own configurations is not possible. However, you can save your own configurations by saving the complete **PLC:\WINDOWMANAGER** directory via TNCremoNT, for example. If you copy the **PLC:\WINDOWMANAGER** directory to other controls, the configuration saved in the directory will become active.



The XFCE configuration dialog provides the following configuration options:

- **Desktop – Appearance**
  The DESKTOP button and the Appearance tab of the XFCE configuration dialog enable you to define a background color and background images, which do only become effective on the 3rd desktop (machine tool builder) or the desktops after the 3rd one.
  The background image from OEM.SYS is shown on the first two desktops (Edit, Machine).

Note

Additional background images (i.e. images that are not supplied) should always be saved in the **PLC:\WINDOWMANGER\BACKDROPS** directory.
Saving the background images in this directory ensures that the path of the selected background image is adjusted to the configuration when backup/restore functions are executed between the programming station and dual-processor or single-processor controls.

- **Desktop – Behavior**
  This function is not yet supported.
  In the future, the DESKTOP button and the Behavior tab of the XFCE configuration dialog will enable you to change the size of the desktop icons.
- **Window Manager – Style**
  The Windows Manager button and the Style tab of the XFCE configuration dialog enable you to influence the layout of the windows. You can edit the font size of the window titles, the orientation of the window headers, the design of the window frames, etc.
- **Window Manager – Advanced**
  The Windows Manager button and the Advanced tab of the XFCE configuration dialog enable you to influence the behavior of the windows.

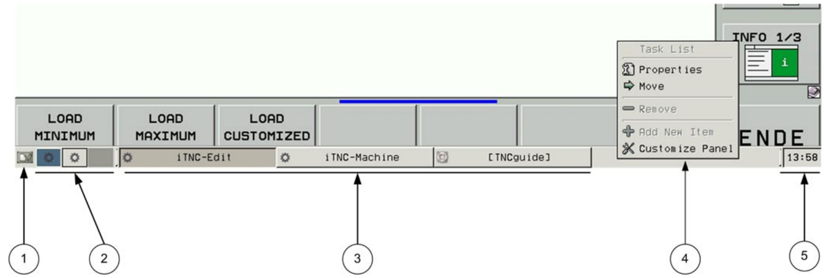
- **Workspaces and Margins**
  To define the number of possible desktops and their names, use the Workspaces and the Margin buttons of the XFCE configuration dialog.

Note

A minimum of two desktops (Edit and Machine) must be defined in Number of Workspaces.

Configuration of the TNC taskbar

The TNC taskbar appears when you move the mouse pointer to the position of the taskbar (standard setting: at bottom of screen). When the mouse pointer is located over the TNC taskbar, press the right mouse key to open the context menu for the configuration.



When the FULL configuration is active, the TNC taskbar contains the following element buttons (from left to right):

- 1: Show desktop button
- 2: Pager for switching between the desktops and for moving windows between the desktops with "drag and drop".
- 3: Task list (shows all open windows)
- 4: Context menu for configuring the TNC taskbar
- 5: Time

To configure the TNC taskbar, proceed as follows:

- Press the ADVANCED SETUP soft key in the PLC operating mode.
- Press the CONFIGURE WINDOW MANAGER soft key.
- Move the mouse pointer over the position of the TNC taskbar until it appears.
- Place the mouse pointer over the TNC taskbar and press the right mouse key. To edit the properties of an element in the taskbar, place the mouse pointer over the respective button in the taskbar.
  This opens the context menu for configuring the TNC taskbar and the selected element.

The context menu provides the following configuration options:

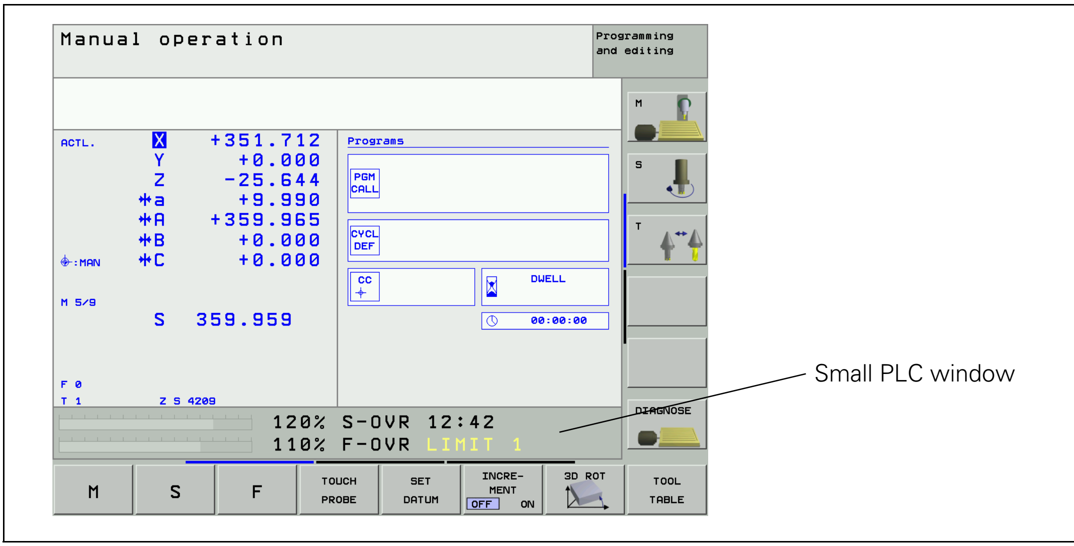
- **Properties**
  Change the properties of the selected element.
- **Move**
  Change the sequence of the buttons in the taskbar.
- **Customize Panel**
  Change the properties of the complete TNC taskbar.

8.2 PLC Window

8.2.1 Small PLC window

The small PLC window is shown in the following operating modes:

- **Manual Operation**
- **El. Handwheel**
- **Positioning with Manual Data Input**
- **Program Run, Single Block**
- **Program Run, Full Sequence**



Any ASCII text can be shown in two lines, each with 38 characters. In the left half of the line a bar diagram can be shown optionally or additionally.

- Specify the colors of the small PLC window in MP7370.x (see "Color settings" on page 1248).
- Configure the window display in the PLC program with Modules 9080 to 9083.
  - 9080: Clear small PLC window
  - 9081: Interrogate status of the small PLC window
  - 9082: Display a string in the small PLC window
  - 9083: Display a bar diagram in the small PLC window

Modules 9080, 9082 and 9083 must be called in a submit or spawn job.

Modules 9080, 9082 and 9083 are also in effect if the selected screen contains no PLC window (e.g. large graphic display) or the PLC window is in the background.

Do not interrupt processing of the modules through a CAN command!

Module 9080 Clear the small PLC window

With this module you can clear the contents of the small PLC window.

Call:
CM 9080

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Call was not in a submit or spawn job

Module 9081 Interrogating the status of the small PLC window

With this module you can ascertain whether a small PLC window is being displayed.

Call:
CM 9081
PL B/W/D <Status of the small PLC window>
Bit 0=1: A small window is in the selected screen (background or foreground)
Bit 1=1: Small PLC window in the foreground

Module 9082 Display a string in the small PLC window

The string is designated with a string number or is transferred as an immediate string and ends with the ASCII character <NUL>. It is shown in the small PLC window in line 0 or 1, with each character in the color given for it. Text strings with a length of max. 128 characters can be transferred to PLC module 9082 for display in the small PLC window. Depending on the size of the window and the screen being used, the NC automatically limits the text string to the maximum number of displayable characters. In the event of error, no string is shown.

Column	063
Line 0	
Line 1	

References to PLC dialogs or PLC error messages are executed:

- Entered dialog or error number not found:
  Is replaced by the ASCII character "@".
- Non-displayable character in the text (except string end):
  Is replaced by the ASCII character "^".

The character size is oriented to the size of the screen window or by the current operating mode, and cannot be influenced.

If the specified color number is zero, the text is shown in the same color as the character last shown. If the first character of a line is specified as zero, the color is undefined and can change from one display line to another.

Call:

```
PS      K/B/W/D   <Line number>
                  0 or 1
PS      K/B/W/D   <Column number>
                  0 to 63
PS      K/B/W/D   <Number of the color>
                  0 to 15
PS      K/B/W/D/S<String number or string>
CM      9082
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Call was not in a submit or spawn job or line less than zero or greater than 1 or Wrong string number or no end of strings or the last characters of the string can no longer be displayed in the screen window.

Module 9083 Showing a moving-bar diagram in the small PLC window

The moving-bar diagram is shown in the specified line with the specified length and colors.

The diagram can be limited to the left half of each line. In this case the ASCII text is limited to max. 19 characters of the right half.

Column	**0150**		**019**
Line 0			
Line 1			

The diagram comprises a rectangular frame in the maximum length and height of an ASCII character. A scale graduation is shown at the top after every ten units. The bar starts from the left-hand edge of the grid. The unused part of the grid is filled in with the background color.
If you define the maximum length > 150, the length is limited to 150. If the current length is > 150, the length is limited to the maximum length.

Color zero uses the background color. The background color of the PLC window can be used for the margin or scale graduation, for example, if they are not to be shown.

Call:

PS	K/B/W/D	<Line number> 0 or 1
PS	K/B/W/D	<Color for bar> 0 to 15
PS	K/B/W/D	<Color for margin and scale graduation> 0 to 15
PS	K/B/W/D	<Current length of the bar> 0 to 150
PS	K/B/W/D	<Maximum length of the bar> 0 to 150
CM	9083	

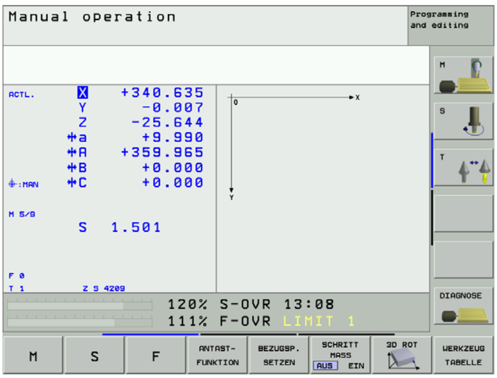
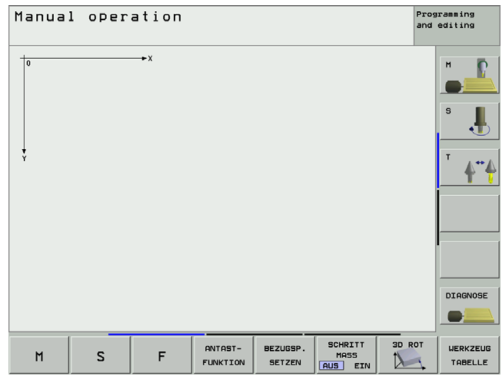
Error recognition:

Marker	**Value**	**Meaning**
M4203	0	No error
	1	Call was not in a spawn or submit job or line less than 0 or greater than 1

8.2.2 Large PLC window

Depending on the display mode, the large PLC window can be shown instead of the graphic/status window, or even over the entire screen. The PLC window can be combined with the PLC soft keys. See page 1320.

- Select the display mode with the screen management key or with Module 9202.

Mode 1		**Mode 2**	
			
SMALL	17 lines, 39 columns 0 = y16/x8	SMALL	27 lines, 79 columns 0 = y16/x8
MEDIUM	11 lines, 19 columns 0 = y24/x8	MEDIUM	18 lines, 39 columns 0 = y24/x8
LARGE	5 lines, 9 columns 0 = y48/x8	LARGE	9 lines, 19 columns 0 = y48/x8

- Define the character size with the special command **`charsize =`** (see "Special commands:" on page 1311). The specified position refers to the lower left corner of the first character.

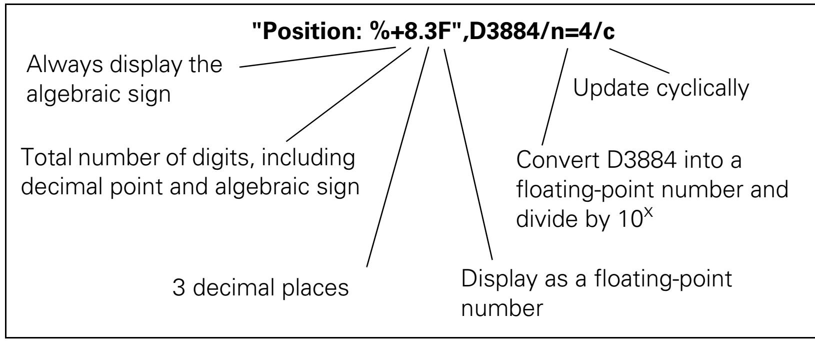
The content of the PLC window is defined in a screen mask – an ASCII file containing format instructions and special commands:

- Enter in Module 9210 the name of the screen mask to activate the PLC window, or use the functions of the soft-key project file for display (see "PLC Soft Keys" on page 1320).

Format instructions

Format instructions are enclosed in quotation marks (""). Variables are transferred as parameters. Symbolic operands can also be used.

Example:



Variable types specified in the format instruction can be written in lowercase or uppercase letters (e.g. `%D`). The variable types of the format instruction must agree with the specified variables.

Note

Integer variables in the iTNC have a length of 32 bits.

PLC variables can be displayed as a number with decimal places:

- With the variable switch `/n=x` you convert from integer to double.

Variable names:

Variable names	Meaning
`B<address>`	PLC bytes, integer
`W<address>`	PLC words, integer
`D<address>`	PLC double words, integer 0/1
`M<address>`	PLC markers, integer 0/1
`I<address>`	PLC inputs, integer 0/1
`O<address>`	PLC outputs, integer 0/1
`T<address>`	PLC timers, integer 0/1
`C<address>`	PLC counters, integer 0/1
`S<address>`	PLC strings, string [128]
`S#D<Number>`	PLC dialogs, string
`S#E<Number>`	PLC error texts, string
`TIME[0] to TIME[15]`	System time as in Module 9055, char
`AXISCHAR[Number]`	Code letters for NC axis, char
`MP<Number>`	Machine parameters, notation: **MP910.1** Input value: Decimal places: double Hex or binary: integer Text: char

Time		
HOUR	int	No. of hours from real-time clock
MIN	int	No. of minutes from real-time clock
SEC	int	No. of seconds from real-time clock
DAY	int	Day from real-time clock
MONTH	int	Month as no. from real-time clock
YEAR2	int	Two-digit year no. from real-time clock
YEAR4	int	Four-digit year from the real-time clock

Settings for the tool touch probe		
TT.RAD	double	Calibrated radius of TT
TT.CENTER [3]	double	Calibrated center of TT
TT.PNT1 [3]	double	Calibrated touch point 0 of TT
TT.PNT2 [3]	double	Calibrated touch point 1 of TT
TT.PNT3 [3]	double	Calibrated touch point 2 of TT
TT.PNT4 [3]	double	Calibrated touch point 3 of TT

Settings for RS-232		
`RS232.FEBAUD`	string	Baud rate FE on RS-232
`RS232.EXT1BAUD`	string	Baud rate EXT1 on RS-232
`RS232.EXT1BAUD`	string	Baud rate EXT2 on RS-232
`RS232.LSV2BAUD`	string	Baud rate LSV2 on RS-232
`RS232.MODE`	string	RS-232 mode

Settings for RS-422		
`RS422.FEBAUD`	string	Baud rate FE on RS-422
`RS422.EXT1BAUD`	string	Baud rate EXT1 on RS-422
`RS422.EXT2BAUD`	string	Baud rate EXT2 on RS-422
`RS422.LSV2BAUD`	string	Baud rate LSV2 on RS-422
`RS422.MODE`	string	RS-422 mode

Settings for simulation		
`SIMU.ENAPRESET`	string	Preset enable
`SIMU.ENALIMIT`	string	Limit switch enable
`SIMU.LIMITPL [5]`	double	Positive limit switch
`SIMU.LIMITMI [5]`	double	Negative limit switch
`SIMU.PRESET [5]`	double	Preset values

Settings of the machine		
`MACHINE.LIMIT1PL [5]`	double	Pos. limit switch group 1
`MACHINE.LIMIT1MI [5]`	double	Neg. limit switch group 1
`MACHINE.LIMIT2PL [5]`	double	Pos. limit switch group 2
`MACHINE.LIMIT2MI [5]`	double	Neg. limit switch group 2
`MACHINE.LIMIT3PL [5]`	double	Pos. limit switch group 3
`MACHINE.LIMIT3MI [5]`	double	Neg. limit switch group 3
`MACHINE.PRESET1 [5]`	double	Preset values 1
`MACHINE.PRESET2 [5]`	double	Preset values 2
`MACHINE.PRESET3 [5]`	double	Preset values 3
`MACHINE.POSINC [5]`	double	
`MACHINE.DRVOFFSET [5]`	double	
`MACHINE.HANDW_FACTOR [5]`	double	Handwheel subdivision factor

Settings for transformation of the machine reference system	
MATRANS.PRESX1_ABC [3]	double
MATRANS.PRESY1_ABC [3]	double
MATRANS.PRESZ1_ABC [3]	double
MATRANS.PRESAXIS1	string
MATRANS.PRESX2_ABC [3]	double
MATRANS.PRESY2_ABC [3]	double
MATRANS.PRESZ2_ABC [3]	double
MATRANS.PRESAXIS3	string
MATRANS.PRESZ3_ABC [3]	double
MATRANS.TRLPRES1 [6]	double
MATRANS.TRLPRES2 [6]	double
MATRANS.TRLPRES2 [6]	double
MATRANS.TRLPRES3 [6]	double
MATRANS.MANUAL	string
MATRANS.PGRMRUN	string
MATRANS.ANGLE [3]	double

Settings for the display	
DISPLAY.AXIS1	string
DISPLAY.AXIS2	string
DISPLAY.SCREEN	int
DISPLAY.FORMAT	string for **MM/INCH** conversion

Settings for the oscilloscope	
OSC.AXIS1	string
OSC.TIMEBASE	string
OSC.MODE	string
OSC.TRGCHAN	string
OSC.TRGTHRES	double
OSC.SLOPE	string
OSC.PRETRIG	string
OSC.FEED	double

Miscellaneous	
MISC.MDI	string
MISC.OUTPRECISION	string
MISC.TEACHINAXIS	int

Special characters:

\n
Newline: Shifts the cursor to the left edge of the window. At the same time, it moves downward by the preset distance defined with **LINEDIST.**

\f
Formfeed: Functions like **"\n"**. In addition, a page break is performed if the cursor moves past this special character. Otherwise the window is scrolled.

\xYY<SPACE>
Special character: YY is the hexadecimal number of the 8-bit ASCII code of the desired character, followed by a blank space, e.g. "\x23 " = "#". Keep in mind that certain characters could also be interpreted as system commands, e.g. "\25 " = "%" (see "iTNC character set" on page 1257).

Switches for variables:

/n=x
For B/W/D. The integer can be reformatted to a floating-point number with *x* decimal places (e.g. for displaying a 0.1-µm-format position in millimeters).

/mi
For B/W/D. Ensure that the inch conversion is active. The number can be converted to a value in inches.

/e
For B/W/D/M/S. Define the field length in the format string. The current contents of the variables are displayed and can be changed.

/i
For B/W/D/M/S. Define the field length in the format string. A value can be entered in the empty field.

/c
For B/W/D/M/S/TIME. Define the field length in the format string, otherwise the subsequent text may be overwritten if the field length varies due to a change in the numerical value. The field content is updated cyclically.

Special commands:

/*<Comment>*/
You can enter a comment between the asterisks.

MMINCH
Converts variables that contain an /mi switch (or a position) into inches. Select a default setting under Mode. Usual default setting: "No conversion"

POS=xpix, ypix
Writes the next text or graphic at the designated position. Default setting: Writing begins at the upper edge, then progresses line by line.
ypix = Distance in pixels from the upper edge of the current page
xpix = Distance in pixels from the left edge of the window

IPOS=xpix, ypix
Writes the next text or graphic at a position offset from the present position by the specified number of pixels.
xpix = Distance in pixels from the current X position
ypix = Distance in pixels from the current Y position

CPOS=column, line
Writes the next text or graphic at the designated position. The width of a column is calculated from the current character set. The height of a line is preset and can be changed with **LINEDIST.**
line = Line on the current page
column = Column from the left edge of the window

ICPOS=column, line
Writes the next text or graphic at a position offset from the present position by the specified number of lines and columns. The width of a column is calculated from the current character set. The height of a line is preset and can be changed with **LINEDIST.**
line = Distance in lines from the old line
column = Distance in spaces from the old space

LINEDIST=ypix Defines the line spacing. The default setting depends on the character size and is reset with every call of **CHARSIZE.**

COLOR=[f] or **COLOUR=[f]**
Sets the foreground color. Value range for f: 1 to 15
Default setting: Color 11
The colors are defined in MP7367.
(see "Color settings" on page 1248)

CURSOR=ON/OFF
Switches the inversion (highlighting) on and off.
Default setting: OFF

CHARSIZE=SMALL/MEDIUM/LARGE/AUTO
Defines the character size.
Split screen: **SMALL**
Large PLC window: **MEDIUM**
AUTO: Character size depends on the window size.
Default setting: **AUTO**
With every call of **CHARSIZE**, the value of **LINEDIST** is overwritten by a default setting that depends on the character set.

Default spacings:

Line	Column
SMALL	168 pixels
MEDIUM	2416 pixels
LARGE	4832 pixels

GRAPHICS= <filename> [,<Layer>] [/c]
Links a graphic into the window. The following graphic formats can be displayed:

- *.HE files (*.DXF files converted by PLCdesign)
- *.BMP
- *.BMX

With **POS, IPOS** or **ICPOS,** enter a position. For *.HE files the lower left corner is set to the current position; whereas for *.BMP and *.BMX files the upper left corner is set.

<filename>: File name with path or just the file name. In this case the path is completed according to MP7230.3 (language for Help files). If **%GraphicsResolution%** is entered in the path, then this text is replaced by the current screen resolution (1024x768 or 640x480).
<Layer> (optional): Layer of a BMX file to be displayed. If no entry is made, the basic image (layer 0) is displayed.
/c (optional): The graphic is cyclically refreshed. The cycle time is defined via **REFRESH =.** If it is a BMX file, the layers are displayed cyclically, beginning with 0 or **<Layer>**.

Screen	Max. resolution of *.BMP, *.BMX
BF 120 (position/program + PLC)	317 x 265 (16- or 24-bit color depth)
BF 120 (only PLC)	636 x 366 (16 or 24-bit color depth)
BF 150 (position/program + PLC)	442 x 431 (16 or 24-bit color depth)
BF 150 (only PLC)	886 x 593 (16 or 24-bit color depth)

Images that are too large are truncated.

Example: **GRAPHICS = PLC:\Images\%GraphicsResolution%\Test.BMX**
The file Test.BMX is searched for in the paths PLC:\Images\1024x768\ or PLC:\Images\640x480\.

TEXTFILE= <filename>
Links a text file into the PLC window. The text begins at the current position. Every additional line begins at the same X position, but offset downward by **LINEDIST.** The line break automatically adapts itself to the available space. Characters such as "Line Feed," "Carriage Return," "Horizontal Tab" and "Vertical Tab" are converted to spaces.
The backslash "\" is used as a special symbol. It is used to initiate the following functions:
"\n", "\N"
Insert manual line feed (end of paragraph)
"\f", "\F"
Insert page feed (division into more than one screen page).
"\\"
Shows the "\" character in the text.
<filename>
Contains a file name with path, or the file name only. In this case the path is completed according to MP7230.3 (language for Help files).

ERRQUE=n [/c] [/e] [/l] [/n] [/s]
Links a table with the messages waiting in the PLC error queue.

n: Number of table lines
/c: Table is updated cyclically.
/e: Paging in tables, message can be acknowledged with CE.
/l: Alternative to **/n.** Line number before the error text. Position in the error source is displayed.
/n: Alternative to **/l**. **/l** has priority.
Error number before the error text.
Line number of the .PET table is displayed.
/s: Three-digit status field with the following information:
C: CE possible
S: Message causes a stop
E: Message causes an EMERGENCY STOP
F: Resets the feed-rate enabling
0 to 2: Priority

REFRESH=n
Time interval in [ms]
All variables with the /c switch are checked and, if required, redisplayed.
Value range: 100 to 100 000 [ms]
Default setting: 400 ms

KBD
This command is needed only if relatively long texts are to be moved with the cursor keys. It assigns the keyboard to the PLC window as long as it is visible on the screen.
If the page limits were defined with **\f**, it is possible to scroll and to page up and down with the arrow keys.
If the mask contains elements with the **/e** or **/i** switch, the keyboard is automatically assigned to the PLC window. In this case the arrow keys jump from input field to input field.
Soft keys, screen switch-over keys, operating mode keys, special function keys (MODE, PGM-MGT, CALC) always remain assigned to the NC.

LINE=xpix, ypix
Draws a line from the current position to the designated position. The designated position is then assumed as the current position.
xpix = Distance in pixels from the left edge of the current page
ypix = Distance in pixels from the upper edge of the window

ILINE=xpix, ypix
Draws a line from the current position to a position that is offset by **xpix, ypix.**
xpix, ypix = Line lengths in x, y.
Then the current position is corrected by **xpix, ypix.**

LINESTYLE=SOLID/DASH/LDASH
Defines the line type for the **LINE/ILINE** command.
SOLID = Solid line
DASH = Dashed line (interrupted line)
LDASH = Dot-and-dash line
Default setting: **SOLID**
The width of the line is one pixel and cannot be changed.

FILE= <table name>
Opens a table for access with **tableread.**
You cannot open more than one table at a time. If the **FILE** command is called more than once, the previously opened table is closed. At the end of the mask the table is closed automatically.

tableread (line,column)
Reads field contents from the table that has been opened with **FILE=**. With the **/c** switch you can show the field contents of a table and update them cyclically.

Example:
```
CHARSIZE = SMALL;
LINESTYLE = SOLID;
FILE = TNC:\P_PLATZ.P;
COLOR=1;
"%s", tableread(0, "P-NR");
"%s", tableread(2, "P-NR");
```

Mathematical expressions for screen positions

If a numerical value is expected as a parameter for a special function, then a mathematical expression can be written in integer arithmetic.

The operators and priority rules used in the C programming language apply.

Available functions: +, –, *, /, %, &, |, ^.

The mathematical expression may have the following variables:

Variable	Meaning
PAGE	Number of the current page, beginning from zero
XPOS	X position of the cursor in pixels
YPOS	Y position of the cursor pixel
LINEDIST	Currently defined line spacing in pixels
ROWDIST	Currently defined character spacing, width of an ASCII character
XSIZE	Width of the screen window in pixels
YSIZE	Height of the screen window in pixels

Input fields

With the switches **/e** and **/i** you can assign input fields to the variables:

/e: shows the current value that can be overwritten.

/i: shows an empty field in which a new value can be entered.

In addition, both switches **/e** and **/i** can be given an identifier **xxx** (**/e = xxx**, **/i = xxx**), where **xxx** is a positive whole number. With Module 9211 you can then ascertain whether the cursor is located in this field.

The switch **/s = xxx** is used to created a field in which no entries can be made. By entering the identifier **xxx** it is possible to ascertain with Module 9211 whether the cursor is located in this field.

If the switches **/e**, **/i** or **/s** are used, the cursor keys function as jump commands from input field to input field. The current page is scrolled if necessary. Any text between the input fields might no longer be shown.

> Note
>
> Do not edit any text before the first input field or after the last.

The C command "printf" requires a format that defines the length of the numerical field:

- Save this format in the mask file. Otherwise the length of the input field depends on the coincidental content of the associated variable.

For the input function this format instruction is converted internally into a form suitable for the C command "scanf":

printf: %[flags][digits1][.[digits2]][1]conversion_char

scanf: %[digits1]   [size]conversion_char

> Note
>
> **Special characteristics**
>
> - %d, %e
>   The size information "1" can be omitted. Floating-point variables are of the double type and automatically add to this information.
> - %g
>   Do not use. Causes errors.
> - %i
>   Do not use. Any number entered with leading zeros would be interpreted as an octal number.
> - %u
>   Works correctly only in the definition range for the respective variables.
> - The size indicator **h** (short integer) of the "scanf" function cannot be written. All integer variables are automatically expanded to 32 bits for input and output.

You can enter data in the input field through the ASCII keyboard and the numerical keys.

The following keys have special functions:

Function	Meaning
CE	Clears the input field or the displayed error message.
ENT	Transfers the input value to the variable and sets the highlight to the next input field. If the input value is syntactically incorrect or exceeds the numerical range of the assigned variable, the error message ENTRY VALUE INCORRECT appears.
NOENT	Shows the original content of the field again and sets the highlight to the next input field.
–/+	If the input value begins with the algebraic sign – or +, the sign is switched.
<x	If the field was already edited, the last character of the entry is deleted. Otherwise the displayed value is assumed into the editing memory and the cursor is set to the end of the input value.

Opening or clearing screen mask for the PLC window

- Define the file names and path of the screen mask in one of the string memories or in an immediate string.
  If no path name is specified, the path for the language indicated in MP7230.3 (help files) is used.

You can also use the functions from the soft-key project file to display a large PLC window (see "PLC Soft Keys" on page 1320).

Module 9210 Open or clear screen mask for the PLC window
With this module you can activate or clear the display in the large PLC window.

Call:

PS	B/W/D/K/S<Clear no. of string memory/file name/PLC window>	
		–1: Clear PLC window
CM	9210	
PL	B/W/D	<Status/Error>
		0: Mask opened / mask cleared
		1: PLC window not yet ready again
		–1: Error

If a faulty mask file was activated, an error message appears in the PLC window.

Error message	Meaning
`COMMAND LIMITER ";" MISSING`	End of command not found
`UNKNOWN PARAMETER TOKEN`	Unknown keyword
`UNKNOWN COMMAND`	Unknown command
`STRING FORMAT ERROR`	Impermissible format instruction
`STRING NOT CLOSED`	String end missing
`TOKEN TOO LONG (>32 CHAR)`	Variable name is too long
`PARAMETER INDEX MISSING`	Index is missing Closing bracket "]" is missing
`SOURCE FILE NOT OPENED`	Source file is not opened
`TEMPORARY FILE NOT OPENED`	Temporary target file is not opened
`TOO FEW PARAMETERS`	Too few parameters for format instruction
`WRONG COMMAND PARAMETER`	Parameter does not fit the format
`WRONG PARAMETER SWITCH`	Incorrect switch

Module 9211 Status of the large PLC window

With this module you can interrogate the status of the large PLC window.

Number	Return code
0: Status	0: No screen mask activated 1: Screen mask was activated 2: Screen mask is being activated 3: Screen mask could not be activated
1: Horizontal size	0: No PLC window displayed > 0: Number of pixels
2: Vertical size	0: No PLC window displayed > 0: Number of pixels
3: Displayed page	Displayed page of the screen mask
4: Current field	0: No cursor or the cursor is not located in a field identified with `/s = xxx, /e = xxx` or `/i = xxx.` >0: Return of the value xxx of a field identified with `/s= xxx, /e = xxx` or `/i = xxx.`

Call:

```
PS    B/W/D/K   <Number>
CM    9211
PL    B/W/D     <Status information>
                –1: Error
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Invalid number of the status information

8.3 PLC Soft Keys

You can display your own soft keys on the iTNC and on the HR 420 through the PLC in all operating modes. You can easily define the soft-key projects with PLCdesignNT as of version 2.3 and MenuDesign (included in the PLCdesignNT package as of version 2.3)

When a PLC soft key is pressed the NC enters the soft-key number in W302 (horizontal soft-key row), W304 (vertical soft-key row) or W306 (handwheel soft keys). On the rising edge of the keystroke it enters the soft-key number; on the falling edge it enters –1. The PLC can enter –1 itself after recognizing the soft-key number.

8.3.1 Soft-key project file for screen

> **Note**
>
> The vertical soft keys can only be used with a BF 150.

The PLC soft-key structure is defined in a soft-key project file using various keywords. The number of submenus depends only on the iTNC memory.

In the OEM.SYS file, the PLC soft-key project file *.SPJ is entered through the keyword **SOFTKEYPROJECT =.** After acknowledgment of **Power interruption** the resource file with the same name and the extension .SYS is generated from this file. The results of this evaluation are stored in an ASCII file with the name **<Name of the soft-key project file>.SYS.LOG**.

The soft-key structure is displayed immediately. The PLC soft keys can be influenced with Modules 9205 to 9207. Module 9204 refreshes the PLC soft keys, which is necessary after Modules 9203 and 9207 are executed. Module 9208 determines the status information of the PLC soft keys.

In the standard setting the soft-key number is transferred to the PLC via W302/W304 after the NODE, BACK and ACTION soft keys have been pressed. When the key is released, –1 is confirmed. Direct operands can be coupled to soft keys in the project file or with Module 9206.
With Module 9205, you can also select a word address other than W302/W304.
BLANK soft keys are not reported to the PLC.

> **Note**
>
> The states of the assigned operands (STATUS markers or word) of the ACTION, PULSE, CHECK and RADIO soft keys and the display are checked cyclically. If these operands change, the display is adjusted accordingly. For example, if the marker of a CHECK soft key changes from 0 to 1, the display of the soft key is changed to "pressed."
> If the value of the word address of a soft-key group, in which a group code is saved, is changed by the PLC, then this value is checked. If the value is valid, then the display of the group is adjusted accordingly.

Miscellaneous keywords in the soft-key project file

Entry in *.SPJ	Meaning
`;`	Comment
`SKPATH`	Path of the soft-key graphic files
`SOFTKEY`	Soft-key definition. The name of the soft-key graphic file and the name of the soft key must be specified. The maximum permissible length of the soft-key name is 23 characters. First entry = soft-key number 0, Second entry = soft-key number 1, etc. With ACTION soft keys, the soft-key number is confirmed via W304 (unless changed by Module 9205) to the PLC. You can use the BMX and BMP formats. Properties: ■ Color depth: 16 or 24 bits ■ Soft key for BF 120: 76 x 48 ■ Soft key for BF 150: 120 x 74 You will find information on creating *.BMX files in the online help for PLCdesignNT.

Keywords in the soft-key menus

Entry in *.SPJ	Meaning
`;`	Comment
`#include`	An additional soft-key project file can be included. The name and path of this file must be entered.
Keywords for menu structuring:	
`SKMENU`	Beginning of the definition of a soft-key menu. The name of the menu must be specified. The soft keys are automatically assigned to the correct menu rows. Also note the additional parameters for this keyword on page 1324.
`ENDSKMENU`	End of the definition of a soft-key menu
`NODE`	Soft key jumps to a submenu. Is confirmed via W302/W304 to the PLC. The soft-key name and the name of the submenu must be indicated. Also note the additional parameters for this keyword on page 1324.
`BLANK`	Empty soft key. You can also specify a soft-key name.
`BACK`	Soft key jumps to a submenu. Is confirmed via W302/W304 to the PLC. The soft-key name and the name of the submenu must be indicated. Also note the additional parameters for this keyword on page 1324.

Entry in *.SPJ	Meaning
END	Closes an open pop-up menu. The soft key is reported to the PLC via W302/W304 for the duration that the key is pressed. Also note the additional parameters for this keyword on page 1324.
Keywords for function soft keys:	
ACTION	Function soft key. Is confirmed via W302/W304 to the PLC. The soft-key name must be indicated. Also note the additional parameters for this keyword on page 1324.
PULSE	The soft key is reported to the PLC via W302/W304 for the duration of the PLC cycle. A soft-key name must be indicated. Also note the additional parameters for this keyword on page 1324.
CHECK	A coupled marker is set the first time it is pressed, and is reset the next time. The soft key is reported to the PLC via W302/W304 for the duration that the key is pressed. A soft-key name must be indicated. Also note the additional parameters for this keyword on page 1324.
RADIO	From any group of these soft-key types, no more than one soft key can be pressed. Assigned status markers for the PLC are also set uniquely. This means that one status marker is always set for the PLC, but not more than one marker can be set at the same time (1-out-of-n rule). However, this does not apply to status markers that do not change within the PLC program (set or reset). If assigned status markers within the PLC program are changed, then the 1-out-of-n rule can be broken for one PLC cycle. It is corrected again during the next cycle. The soft key is reported to the PLC via W302/W304 for the duration that the key is pressed. It is also possible to define more than six RADIO soft keys to one group. A soft-key name must be indicated. Also note the additional parameters for this keyword on page 1324.
STATE	Multiple states can be managed for this soft key. The active state depends on the number of times the soft key was pressed. In order to display these states, the BMX file of the soft key must include the corresponding number of possible states, and a PLC word memory must be assigned. If a PLC bit memory is assigned, only two states can be managed. In the following example, a value is assigned to the PLC word W1000 via a BMX soft key with five possible states. Values from 0 to 4 are assigned here: **`SOFTKEY spindle_attr.bmx State_Softkey`** **`STATE State_Softkey STATUS:W1000 STATES:5`** Also note the additional parameters for this keyword on page 1324.

Entry in *.SPJ	Meaning
Global optional keywords:	
TREEFILE	Declares an alternate system-file name
DEFFILE	Declares an alternate definition-file name
HELPFILE	Declares an alternative help-file name for the context-sensitive help
MODOMAIN	Declares an alternative file name for language-sensitive text soft keys
MOPATH	Declares the location of the text files for language-sensitive text soft keys

Additional parameters for the keywords

Entry in *.SPJ	In connection with	Meaning
VROOT	SKMENU	The menu for the vertical soft-key row is defined in the header of the main menu.
HROOT	SKMENU	The menu for the horizontal soft-key row is defined in the header of the main menu.
EMODE	SKMENU	The menu for the programming modes is defined in the header of the main menu.
MMODE	SKMENU	The menu for the machine modes is defined in the header of the main menu.
ENABLE:<Marker>	NODE, BACK, END, ACTION, PULSE, CHECK, RADIO, STATE	Depending on the marker status, the soft key is either locked (marker = 0) or enabled (marker = 1). A locked marker is shown "inactive."
HIDE:<Marker>	NODE, BACK, END, ACTION, PULSE, CHECK, RADIO, STATE	The parameter **HIDE** assigns a marker to a soft key. If this marker receives the value 1, then that soft key is hidden and exchanged for an "empty" soft key.
STATUS:<marker or word>	NODE, BACK, END, ACTION, PULSE, CHECK, RADIO, STATE	An operand is assigned to the soft key (in addition to W302/W304). If a **marker** is indicated and the soft key is pressed, the marker is set. If a **word** is indicated, the soft key number is entered (index number from the *.sys file).
REPEATDELAY: <Delay time in ms> REPEATINTERVAL: <Interval in ms>	ACTION, PULSE	An "autorepeat function" can be realized for a soft key by combining these two parameters. The function of the soft key is performed after a delay **(REPEATDELAY)** at a specified interval **(REPEATINTERVAL).**
STATES:<Marker>	STATE	The **STATES** parameter is used to tell a soft key how many different states it can assume as the result of repeated pressing. The assigned soft-key image file must support the number of states (see the BMXdesign software from HEIDENHAIN for more information)
POPUPMENU:<Menu name>	NODE, ACTION, PULSE, CHECK, RADIO	The menu **Menu name** is shown in the other soft-key row (either vertical or horizontal).
CLOSEPOPUPMENU	NODE, BACK	A menu opened with the **POPUPMENU:<Menu name>** parameter is closed again.
LARGEWINDOW: <Mask file for PLC window>	NODE, ACTION, PULSE, CHECK, RADIO	A large PLC window with the given mask file is displayed across the entire screen.

Entry in *.SPJ	In connection with	Meaning
SMALLWINDOW: <Mask file for PLC window>	**NODE, ACTION, PULSE, CHECK, RADIO**	A large PLC window with the given mask file is opened instead of the graphics/status window. However, the user can extend it over the entire screen.
CLOSEPLCWINDOW	**NODE, ACTION, PULSE, CHECK, RADIO**	A large PLC window opened with the **LARGEWINDOW:** or SMALLWINDOW: parameter is closed again.
FirstInGroup	**ACTION, PULSE, CHECK, RADIO**	Defines a soft-key group. This keyword identifies the first soft key of the respective soft-key group. A group consists of at least two soft keys with the same name, which can even be collected over multiple soft-key rows (menus in the same level). The functionalities listed above are set depending on the soft-key keyword. **Evaluation via group code:** So that the operator entry can be evaluated via a group code, the **STATUS:<word>** parameter must be entered for the first soft-key in the group. No further **STATUS:<word>** definitions may be entered within this group. **ACTION**, **PULSE** and **CHECK** soft keys: The number of the soft key (within a soft-key group) is transferred bit-coded in the dual system in the given **word,** for example: 1st soft key (value = 1) 2nd soft key (value = 2) 3rd soft key (value = 4) 4th soft key (value = 8) ... e.g. 9th soft key (value = 256) For groups with **CHECK** soft keys, more than one soft key can be pressed at the same time. Here the respective value of the group code is summed, and the result is saved in the given **word.** **RADIO** soft keys: If this parameter is used in combination with a **RADIO** soft-key group, the value of the group code is saved as follows: 1st soft key (value = 0) 2nd soft key (value = 1) 3rd soft key (value = 2) 4th soft key (value = 3) …

The token **SOFTKEYPRESS** in **OEM.SYS** can be used to influence the display and behavior of the PLC soft key types CHECK and RADIO. If **SOFTKEYPRESS = ON** is entered, the soft-key graphics and the states of the PLC markers already change when the soft key is pressed. If this entry is missing from the OEM.SYS, the changes listed below are performed as previously. This means that all changes listed in the table are not performed until the soft key is released.
The behavior of the following types of soft keys can be influenced with **SOFTKEYPRESS = ON**:

SK type	Resulting behavior	
CHECK	If pressing the key causes the soft key to switch on,	
		the change to another soft-key graphics (if BMX file is available) is performed when the soft key is pressed.
		the change is performed in the PLC status marker when the soft key is pressed.
	If pressing the key causes the soft key to switch off,	
		the change to another soft-key graphics (if BMX file is available) is performed when the soft key is released.
		the change is performed in the PLC status marker when the soft key is released.
RADIO	the change of the active RADIO soft key is performed within the group when the soft key is pressed.	
	the change to another soft-key graphics (if BMX file is available) is performed when the soft key is pressed.	
	the change is performed in the PLC status marker when the soft key is pressed.	

The system parameters (see "Available system parameters:" on page 1623) for conditional compilation can also be used in the soft-key project file.

Example of a soft-key project file

Soft-key number (confirmed to the PLC):

```
     PLC SOFTKEY Project File - Version 1.0

     SKPATH 'PLC:\SK\1024x768\'

0    SOFTKEY 'BACK.BMX'       BACK_SK
1    SOFTKEY 'MAG.BMX'        MAG_SK
2    SOFTKEY 'MAG_CCW.BMX'    MAG_CCW_SK
3    SOFTKEY 'MAG_CW.BMX'     MAG_CW_SK
...  ...
```

Definition of the soft keys with names for graphics and soft keys

```
SKMENU RootMenuVM VROOT MMODE
NODE   MAG_SK      Magazine
BLANK
BLANK
BLANK
BLANK
BLANK
; Row 2
CHECK  DIAG_SK  POPUPMENU:Diag_Mnu
ACTION CHIP_BACK_SK
BLANK
BLANK
RADIO  SPI_OFF_SK
RADIO  SPI_ON_SK
ENDSKMENU
```

Main menu, vertical soft-key row, machining modes

```
SKMENU RootMenuHM HROOT MMODE
...
BLANK
ENDSKMENU
```

Main menu, horizontal soft-key row, machining modes

```
SKMENU Magazine
ACTION MAG_CCW_SK STATUS: MG_SOFTKEY_WZM_LINKS_DRE
ACTION MAG_CW_SK STATUS: MG_SOFTKEY_WZM_RECHTS_DRE
BACK BACK_SK
ENDSKMENU
```

Entry of an active marker

Submenu "Diagnosis"

```
SKMENU Diag_Mnu
ACTION DIAG_T_SK
BLANK
BLANK
BLANK
CHECK   MACH_SK     LARGEWINDOW:Machine.A
CHECK   MFUNC_SK    SMALLWINDOW:MFUNCT.A
ENDSKMENU
```

Display PLC window with mask file

```
SKMENU row_machine
ACTION  SK_NC_start  FIRSTINGROUP STATUS:NP_W310_GRO
ACTION   SK_NC_stop
ACTION  SK_axes_plus FIRSTINGROUP  STATUS:NP_W312_GR
ACTION   SK_axes_minus
...
```

Group with number saved in W310

Group with number saved in W312

		Set	Reset
W302	**Number of the horizontal PLC soft key that was pressed**	NC	NC
W304	**Number of the vertical PLC soft key that was pressed**	NC	NC

Module 9203 Activate PLC soft-key menu
Up to 340 420-05: with Module 9203, you activate a soft-key resource. The resource file *.SYS is entered.
The PLC soft-key project can be defined in such a way that it is effective only for specific modes of operation. This might be useful, for example, if you want to use different PLC soft-key projects for the machine and programming modes of operation.
The module supplies the resource handle for Modules 9204 to 9208.

Call:

```
PS    D          <Resource handle>
                 0 when it is called for the first time, otherwise
                 Resource handle from Module 9203
PS    B/W/D/K/S<String with path and name of resource file (*.SYS)>
                 (only necessary up to 340 420-05),
                    0 or "" is given (is no longer evaluated)
PS    B/W/D/K    <Number of PLC soft-key menu>
                 Number of the soft-key menu
                 Value < 0:
                    - Returns the current resource handle
                    - (up to 340 420-05: returns an error)
PS    B/W/D/K    <Mode>
                 1: Vertical PLC soft-key row
PS    B/W/D/K    <Mode of operation>
                 0: Programming modes of operation
                 1: Machine modes of operation
                 2: Programming and machine modes of operation
CM    9203
PL    D          <Resource handle>
                 0: Error code in W1022
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Resource-handle overflow, incorrect resource handle, incorrect mode, incorrect operating mode or number of PLC soft-key root menu negative.
	3	Incorrect string number or incorrect string
	20	Module was not called in a spawn job or submit job
	44	Error in the resource file

Module 9204 Update the PLC soft keys

If you want to restructure the PLC soft keys, you must call Module 9204. This is necessary each time you have called Modules 9203 and 9207.

Call:

PS	D	<Resource handle>
		Resource handle from Module 9203
CM	9204	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Incorrect resource handle
	20	Module was not called in a spawn job or submit job

Module 9205 Set the word for acknowledgment of PLC soft keys

Module 9205 can be used to define another word, in addition to W302/W304, in which the pressing of PLC soft keys is acknowledged. This can be done for the complete project file, individual PLC soft-key menus or individual PLC soft keys. Changes in individual PLC soft keys affect the entire project file.

Call:

PS	D	<Resource handle> Resource handle from Module 9203
PS	B/W/D/K	<Mode> 0: Complete project file 1: Individual menu 2: Individual PLC soft key
PS	B/W/D/K	<Number of PLC soft-key menu/Number of PLC soft key> If "complete project file": Non-functional, transfer 0
PS	D	<Reserved> 0 transferred
PS	B/W/D/K	<PLC word address for PLC soft keys> –1: W304
CM	9205	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Invalid PLC word address
	2	Incorrect resource handle or incorrect mode
	3	Invalid PLC label address
	20	Module was not called in a spawn job or submit job
	44	Error during setting of setup parameters

Module 9206 Change the settings of the PLC soft keys

With Module 9206, the settings of individual PLC soft keys in the PLC soft-key structure can be changed:

- PLC soft keys can be locked and unlocked. Locked PLC soft keys cannot be used.
- PLC soft keys can be coupled to new operands. This way the status of the PLC soft key is directly available in the PLC program.
- Couple PLC soft-keys to operands that
  - Unlock the PLC soft keys in a set state
  - Lock the PLC soft keys in a reset state

If a locked PLC soft key is pressed, it sets the marker M4577.

Call:

PS	D	<Resource handle> Resource handle from Module 9203
PS	B/W/D/K	<Soft-key number>
PS	B/W/D/K	<Function> 0: Lock soft key 1: Unlock soft key 2: Decouple soft key from assigned operand 3: Couple new operand to soft key 4: Decouple the soft-key unlocking/locking from the assigned operand 5: Couple the soft-key unlocking/locking with the operand
PS	B/W/D/K	<Operand address> Only for function 3 and 5, otherwise transfer 0
PS	B/W/D/K	<Operand type> Only for function 3 and 5, otherwise transfer 0 0: Marker M 1: Input I 2: Output O 3: Counter C 4: Timer T
CM	9206	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Incorrect resource handle or incorrect function
	20	Module was not called in a spawn job or submit job
	44	Error during setting of setup parameters

Module 9207 Replace PLC soft keys

With Module 9207, individual PLC soft-keys can be replaced by another PLC soft key. The change can be applied to the entire project file or only to an individual menu. If a soft key is to be replaced in the entire project file, the source menu is excepted from it so that the soft key can remain to make it possible to reverse the replacement.

Call:

PS	D	<Resource handle> Resource handle from Module 9203
PS	B/W/D/K	<PLC soft-key number source>
PS	B/W/D/K	<Number of PLC soft-key menu source>
PS	B/W/D/K	<PLC soft-key number target>
PS	B/W/D/K	<Number of soft-key menu target> –1: Entire resource file except source menu
PS	B/W/D/K	<Function> 0: Replace soft key
CM	9207	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Incorrect resource handle or incorrect function
	20	Module was not called in a spawn job or submit job
	44	Error during setting of setup parameters

Module 9208 Status information of the PLC soft keys

Call:

PS	D	<Resource handle> Resource handle from Module 9203
PS	B/W/D/K	<Function> 0: Number of the current soft-key menu
PS	B/W/D/K	<reserved> 0 transferred
CM	9208	
PL	D	<Status information>

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Incorrect resource handle or incorrect function
	20	Module was not called in a spawn job or submit job
	44	Error finding the status information

8.3.2 Soft-key project file for HR 420

Via the PLC, the iTNC can also manage freely-definable soft keys on the HR 420. The corresponding entries are made in the general soft-key resource file (*.spj) of the PLC project, in which the soft keys for the vertical and horizontal soft-key rows of the iTNC screen are defined.
It is possible to switch from the basic menu of the HR 420, via the **FCT** (Function) soft key, to a freely-definable submenu structure, each of which contains 5 soft keys, whose description and functions are specified in the soft-key resource file.

This menu is displayed in the fourth row of the HR 420. Four ASCII characters are available for each soft key. However, if each soft key uses all four characters, then there is no empty space between the soft-key designations. In addition, you can optionally have a menu title with 20 characters displayed in the third row of the HR 420 display for the main menu and each submenu. The scope of function of the soft keys on the handwheel is limited somewhat compared to the vertical and horizontal soft keys of the iTNC screen.

The following definitions are possible:

Entries for the menu definition and type of soft key in the HR 420 menu:

<table>
<tr><th>Entry</th><th>Parameter</th><th>Description</th></tr>
<tr><td>;</td><td></td><td>Comment</td></tr>
<tr><td rowspan="4">SKMENU <menu name>
ENDSKMENU</td><td></td><td>Beginning or end of the definition of a soft-key menu. The name of the menu must be given for SKMENU (e.g. "HRRootMenu"—see the example). The soft keys are aligned on the HR 420 in the sequence in which they appear in the file. Also note the additional parameters for this keyword.</td></tr>
<tr><td>HRROOT</td><td>Freely-definable root menu when called from the basic handwheel menu via the FCT (Function) soft key</td></tr>
<tr><td>HRMENU</td><td>Freely-definable submenu, called via the keyword NODE ...</td></tr>
<tr><td>TITLE:<name></td><td>Menu title: Freely definable text in the third line of the HR 420
The parameter TITLE: can also be surrounded by quotation marks. This permits blank spaces in the menu title.</td></tr>
<tr><td>NODE <submenu name></td><td colspan="2">Soft key jumps to a submenu. Is confirmed via W306 to the PLC. The soft-key name and the name of the submenu must be indicated.</td></tr>
<tr><td>BACK</td><td colspan="2">Soft key jumps to a submenu. Is confirmed via W306 to the PLC. The soft-key name and the name of the submenu must be indicated.</td></tr>
<tr><td>BLANK</td><td colspan="2">Empty soft key, is shown as "..". You can also specify a soft-key name.</td></tr>
<tr><td rowspan="2">ACTION <soft-key name></td><td></td><td>Function soft key. Is confirmed via W306 to the PLC. The soft-key name must be indicated.</td></tr>
<tr><td>STATUS:<marker or word></td><td>An operand is assigned to the soft key (in addition to W306). If a marker is indicated and the soft key is pressed, the marker is set. If a word is indicated, the soft key number is entered (index number in the *.sys file, e.g. Softkey.sys).</td></tr>
<tr><td>PULSE <soft-key name></td><td colspan="2">The soft key is reported to the PLC via W306 for the duration of the PLC cycle. A soft-key name must be indicated.</td></tr>
<tr><td colspan="3">The soft-key types RADIO, CHECK and STATE may not be used. Other parameters, such as ENABLE, HIDE, REPEAT, etc. are not yet available.</td></tr>
</table>

Additional keywords and parameters for the definition of soft keys in the HR 420 menu:

Keyword	Parameter	Description
TX	**#I<text>**	Language-neutral text for the description of a soft key (up to 4 ASCII characters)
	#A<ASCII value>	Input of an ASCII value as a possibility for displaying special characters. This value must be entered in decimal notation as three digits. Special characters include - 128 = Arrow up - 129 = Arrow down - 133 = Return symbol in the menu
	A combination of the parameters named above is possible, e.g. **#IUp#128** to display **Up**↑	

The following example shows the configuration of a soft-key menu for an HR 420. These settings must be fully integrated in the soft-key project file of the iTNC 530.

```
PLC SOFTKEY Project File - Version 1.0

SKPATH 'PLC:\SK\1024x768\'
...
TX      #IUp#A128       HR_SK1
TX      #IDn#A129       HR_SK2
TX      #ISta           HR_SK3
TX      #IPul           HR_SK4
TX      #IEND           HR_SK5
...

SKMENU  HRRootMenu  HRROOT "TITLE:Menu: Laser Head"
ACTION  HR_SK1      STATUS: MG_LASER_HEAD_UP
ACTION  HR_SK2      STATUS: MG_LASER_HEAD_DOWN
NODE    HR_SK3      HRSubMenu
ACTION  HR_SK4      STATUS: M1122
BACK    HR_SK5
ENDSKMENU


SKMENU  HRSubMenu   HRMENU
ACTION  HR_SK6
...
BLANK
ENDSKMENU
```

Definition of the soft keys with their description

Root menu of the handwheel menu

Submenu of the handwheel menu

8.3.3 Compatibility with TNC 426/TNC 430

With Module 9200 you can display entire soft-key rows. With Module 9201 you can show individual soft keys. With Module 9202 you can switch to the display with PLC soft keys and PLC windows. This module works like the screen management key.

Display/delete PLC soft-key row

- In the system file PLC:\PLCSOFTK.SYS, enter the names and path of the required soft-key files. With the sequence of your entries you specify the soft-key number: Line 0 = soft-key number 0, etc. One soft-key level can consist of up to four soft-key rows, i.e. 32 soft keys per level.
- When calling the module, indicate the row to be displayed first.
- With the transfer parameter, specify how the soft keys should be displayed:
  - After the screen management key is pressed, i.e. after the PLC window is selected.
  - In the current operating mode:
    In this case the NC soft keys are overwritten.
- Specify whether the NC soft keys should be overwritten or whether the PLC soft keys should be appended to the NC soft keys. If you append the PLC soft keys, a separate list is opened. Only one PLC soft-key row can be appended.

		Set	Reset
W302	**Number of the horizontal PLC soft key that was pressed**	NC	NC

Module 9200 Display/delete PLC soft-key row

The soft keys to be activated are specified in a constants field by their line numbers. If there is no PLCSOFTK.SYS file, or if the lines indicated in the constants field do not exist, no soft-key row is generated.

Can only be called from the sequential program.

Call:

PS	B/W/D/K/KF	<Select or delete soft-key address> 1: Delete soft-key level KF: Address of soft-key selection
PS	B/W/D/K	<Soft-key row> 0 to 3: Soft-key row to be displayed
PS	B/W/D/K	<Soft-key mode> 0: Soft-key row for displayed PLC window 1: Soft-key row in current operating mode 2: Append soft-key row to NC soft keys
CM	9200	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Incorrect transfer parameters (e.g. KF address not in address range of the PLC code)
	2	Line nr. < 0 (not –1) in the constants field
	24	Module was called in a spawn job or submit job
	25	More than 32 elements in the constants field

Display/delete PLC soft key

▶ Procedure for displaying/deleting a PLC soft key

Module 9201 Display/delete PLC soft key

If no PLCSOFTK.SYS file exists, or if the specified line does not exist, no soft key is generated. In an existing PLC soft-key level, the soft key is displayed/deleted at the specified position.

Can only be called from the sequential program.

Call:

PS	B/W/D/K	<Soft-key number (line no./delete)> 0: Line no. –1: Delete soft key
PS	B/W/D/K	<Position no.> 0 to 31
PS	B/W/D/K	<Soft-key mode> 0: Soft key for displayed PLC window 1: Soft key in current operating mode 2: Append soft key to NC soft key
CM	9201	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Transfer parameter out of value range
	2	Line nr. < –1
	24	Module was called in a submit job

Select/deselect PLC soft keys and PLC windows

Module 9202 Select/deselect PLC soft keys and PLC windows

With Module 9202 you activate the display with PLC windows or the PLC soft-key display. This module works like the screen management key.

Call:

PS	B/W/D/K	<Display mode> 0: PLC soft key/window deselected 1: Small PLC soft key/window deselected 2: Large PLC soft key/window deselected 3: Large PLC soft key/window selected while table editor is active
CM	9202	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Transfer parameter out of value range

8.4 Keystroke Simulation

HEIDENHAIN contouring controls have two control panels:

- iTNC keyboard unit
- The machine operating panel from the machine tool builder

These control panels are connected with the MC 42x(B,C) at connections X45 and X46.

The key code of the iTNC keyboard unit is evaluated directly by the NC.

PLC inputs and outputs for the machine operating panel are available on connection X46. You must evaluate the inputs and outputs in the PLC.

8.4.1 iTNC control panel

The key code of the iTNC keyboard unit is evaluated by the NC. The key code is displayed in W274 while a key is being pressed. (see "Codes for keystroke simulation" on page 1348).

If you press a disabled key, marker M4577 is also set.

The following modules can influence keys and soft keys:

- Module 9180: Simulation of NC keys
- Module 9181: Disabling of individual keys
- Module 9182: Re-enabling of individual keys
- Module 9183: Disabling groups of NC keys
- Module 9184: Re-enabling of groups of NC keys
- Module 9186: Calling a soft-key function
- Module 9187: Status of a soft-key function call
- Module 9188: Calling a soft-key function (as of 340 49x-04)

With MP4020 bit 9, specify whether a simulated key should only be transmitted to the NC, or also to an active PLC window.
With MP4020 bit 10, specify whether a disabled key should be disabled only for the active PLC window, or for the active PLC window and for the NC.

		Set	Reset
W274	**Code of the depressed key**	NC	NC
M4577	**Disabled key was pressed**	NC	PLC

MP4020 **PLC compatibility**

Input: Bit 9 – Behavior of a simulated key
0: Simulated key is transferred immediately to the NC
1: Simulated key is processed first by an active PLC window before being transferred to the NC
Bit 10 – Behavior of a disabled key
0: Locked key only works on the active PLC window
1: Locked key works on neither the active PLC window nor on the NC

Module 9180 Simulation of NC keys

With this module you can simulate the activation of NC keys and soft keys. You transfer the code of the desired key.

If you transfer the code value zero, the number of occupied elements in the keystroke queue is returned. In this case there is no keystroke simulation.

Call:
PS B/W/D/K <Key code>
CM 9180
PL B/W/D <Number of occupied elements / error status>
0: Key code was transferred, key queue is empty
1 to 16 : Key code was not yet simulated, max. 16 entries in the keystroke queue are possible
–1: For error see W1022

Error recognition:

Marker	Value	Meaning
M4203	0	NC key was simulated
	1	Error code in W1022
W1022	1	Transferred parameter > maximum value
	2	Transferred parameter invalid
	22	Keystroke queue overflow

Module 9181 Disable individual NC keys

With this module you can disable individual NC keys.

If you press a disabled key, marker M4577 is set.

Call:
PS B/W/D/K <Key code>
CM 9181
PL B/W/D <Error status>
0: NC key disabled
–1: For error see W1022

Error recognition:

Marker	Value	Meaning
M4203	0	NC key was disabled
	1	Error code in W1022
W1022	1	Transferred parameter > maximum value
	2	Transferred parameter invalid

Module 9182 Re-enabling individual NC keys
With this module you cancel the effect of Module 9181.

Call:
PS B/W/D/K <Key code>
CM 9182
PL B/W/D <Error status>
0: NC key enabled
–1: For error see W1022

Error recognition:

Marker	Value	Meaning
M4203	0	Disabling was cancelled
	1	Error code in W1022
W1022	1	Transferred parameter > maximum value
	2	Transferred parameter invalid

Module 9183 Disabling groups of NC keys
The key-group codes are:

- 0: All keys
- 1: ASCII
- 2: Horizontal soft keys, Page Up/Down
- 3: Cursor, ENT, NOENT, DEL, END, GOTO
- 4: Numbers, algebraic signs, decimal point, actual position capture
- 5: Operating modes
- 6: Block opening keys
- 7: Vertical soft keys, switch-over key

Call:
PS B/W/D/K <Key-group code>
CM 9183
PL B/W/D <Error status>
0: Group of NC keys disabled
–1: Transferred value> maximum value

Error recognition:

Marker	Value	Meaning
M4203	0	The group of NC keys was disabled
	1	Error code in W1022
W1022	2	Transferred parameter invalid

Module 9184 Re-enabling groups of NC keys

With this module you cancel the effect of Module 9183.

Call:

PS	B/W/D/K	<Key-group code>
CM	9184	
PL	B/W/D	<Error status> 0: Group of NC keys enabled –1: Transferred parameter > maximum value

Error recognition:

Marker	Value	Meaning
M4203	0	Disabling was cancelled
	1	Error code in W1022
W1022	2	Transferred parameter invalid

Module 9186 Call a soft-key function

With this module you can call certain soft-key functions in the machine operating modes.

Do not call a new function until the previous one is completed. You can interrogate this condition with Module 9187.

For a soft-key function to be simulated it must be displayed either in the foreground or background operating mode. Otherwise calling the module has no effect. Module 9187 reports the error.

Call:

PS	B/W/D/K	<Number of the soft-key function> 0: INTERNAL STOP 1: M output 2: S output 3: PROBE FUNCTION 4: PASS OVER REFERENCE MARK 5: RESTORE POSITION 6: INCREMENTAL JOG 7: Feed-rate limitation F MAX 8: MANUAL TRAVERSE
CM	9186	

Error recognition:

Marker	Value	Meaning
M4203	0	Soft-key function was called
	1	Error code in W1022
W1022	1	Parameter out of value range
	28	Previous call not ended

Module 9187 Status of a soft-key function call

Immediately after Module 9186 is called, the status 1 (soft-key function not yet completed) is set — regardless of whether the function can be run in the current operating mode. Module 9186 cannot be called again until status 0 or 2 is set. The error status 2 is erased if Module 9186 is called again or if power is switched on.

Call:

CM	9187	
PL	B/W/D	<Status> 0: Soft-key function completed or none called 1: Soft-key function not yet completed 2: Error: Soft-key function cannot be completed because soft key is not available or operating mode is incorrect

Module 9188 Call a soft-key function

Module 9188 enables you to simulate soft-key functions. The soft-key designation is used for programming.

Conditions:

- For a soft-key function to be simulated the soft key must be contained in the soft-key row that is displayed on the screen or in one of the soft-key rows that are in the same level but on the following screen pages.
- The parameter **Attribute** defines whether a function should be activated although the soft key is not displayed in the soft-key row on the screen, but is contained in one of the following soft-key rows.
- As opposed to module 9186, the status can be determined with error code W1022 as soon as the module is called.

The following soft keys can be simulated using the designators listed below:

Soft key	Soft-key designator
TOUCH PROBE	SK_NC_Manual_operation_TOUCH_PROBE
PRESET TABLE	SK_NC_Manual_operation_PRESET_TABLE
TOOL TABLE	SK_NC_TOOL_TABLE
INCRE- MENT OFF ON	SK_NC_Manual_operation_INCREMENT_OFF
INCRE- MENT OFF ON	SK_NC_Manual_operation_INCREMENT_ON
PASS OVER REFERENCE	SK_NC_Manual_operation_PASS_OVER_REFERENCE
SET DATUM	SK_NC_Manual_operation_SET_DATUM
TOOL USAGE TEST	SK_NC_Program_run_TOOL_USAGE_TEST
F MAX	SK_NC_Program_run_F_MAX
AUTOSTART	SK_NC_Program_run_AUTOSTART
AUTOSTART OFF ON	SK_NC_Program_run_AUTOSTART_ON

Soft key	Soft-key designator
AUTOSTART OFF ON	SK_NC_Program_run_AUTOSTART_OFF
ON OFF	SK_NC_Program_run_SLASHED_BLOCKS_NOT_ ACTIVE
ON OFF	SK_NC_Program_run_SLASHED_BLOCKS_ACTIVE
GLOBAL SETTINGS	SK_NC_Program_run_GLOBAL_SETTINGS
AFC SETTINGS	SK_NC_Program_run_AFC_ENTER_TABLES
TABLE EVALU- ATION	SK_NC_Program_run_AFC_RESULTS
TABLE SETTINGS	SK_NC_Program_run_AFC_SETTINGS
AFC OFF ON	SK_NC_Program_run_AFC_ON
AFC OFF ON	SK_NC_Program_run_AFC_OFF
SET STANDARD VALUES	SK_NC_Program_run_GLOBAL_SETTINGS_SET_ STANDARD_VALUES

Soft key	Soft-key designator
GLOBAL SETTINGS INACTIVE	SK_NC_Program_run_GLOBAL_SETTINGS_SET_INACTIVE
CANCEL CHANGE	SK_NC_Program_run_GLOBAL_SETTINGS_UNDO
INTERNAL STOP	SK_NC_Program_run_INTERNAL_STOP
RESTORE POSITION	SK_NC_Program_run_RESTORE_POSITION
MANUAL TRAVERSE	SK_NC_Program_run_MANUAL_TRAVERSE

Call:
PS B/W/D/K <Mode>
0 = Simulate the soft key using the soft-key designator
PS B/W/D/K/S<Soft-key designator>
PS B/W/D/K <Attribute>
Bit 0:
0 = Soft-key is only simulated if it is in the main row
1 = Soft key is simulated regardless of whether it is in the main or subordinate row
CM 9188

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	See error code
W1022	2	Invalid value for mode
	11	String as a soft-key designator is invalid
	20	Call was not in a submit or spawn job
	45	Internal operating system error
	61	Soft key is neither available in the main soft key row nor in the following soft-key rows
	62	Programmed soft-key name does not exist

Codes for keystroke simulation

Code	Key	Group
$00	No key	
$08	BACKSPACE	ASCII
$09	TAB	ASCII
$0A	RET	ASCII
$09	TAB	ASCII
$1B	ESC	ASCII
$20	SPACE	ASCII
$21	!	ASCII
$22	„	ASCII
$23	#	ASCII
$24	$	ASCII
$25	%	ASCII
$26	&	ASCII
$28	(	ASCII
$29	)	ASCII
$2A	*	ASCII
$2B	+	ASCII
$2C	,	ASCII
$2D	–	ASCII
$2E	. (ASCII DOT)	ASCII
$2F	/	ASCII
$30	0	Numbers
$31	1	Numbers
$32	2	Numbers
$33	3	Numbers
$34	4	Numbers
$35	5	Numbers
$36	6	Numbers
$37	7	Numbers
$38	8	Numbers
$39	9	Numbers
$3A	:	ASCII
$3B	;	ASCII
$3C	<	ASCII
$3D	=	ASCII
$3E	>	ASCII
$3F	?	ASCII
$41	A	ASCII
$42	B	ASCII
$43	C	ASCII
$44	D	ASCII

Code	Key	Group
$45	E	ASCII
$46	F	ASCII
$47	G	ASCII
$48	H	ASCII
$49	I	ASCII
$4A	J	ASCII
$4B	K	ASCII
$4C	L	ASCII
$4D	M	ASCII
$4E	N	ASCII
$4F	O	ASCII
$50	P	ASCII
$51	Q	ASCII
$52	R	ASCII
$53	S	ASCII
$54	T	ASCII
$55	U	ASCII
$56	V	ASCII
$57	W	ASCII
$58	X	ASCII
$59	Y	ASCII
$5A	Z	ASCII
$5E	^	ASCII
$160	Soft key 0	Vertical soft key
$161	Soft key 1	Vertical soft key
$162	Soft key 2	Vertical soft key
$163	Soft key 3	Vertical soft key
$164	Soft key 4	Vertical soft key
$165	Soft key 5	Vertical soft key
$17D	FNEXT (vertical)	Vertical soft key
$180	Soft key 0	Horiz. soft key
$181	Soft key 1	Horiz. soft key
$182	Soft key 2	Horiz. soft key
$183	Soft key 3	Horiz. soft key
$184	Soft key 4	Horiz. soft key
$185	Soft key 5	Horiz. soft key
$186	Soft key 6	Horiz. soft key
$187	Soft key 7	Horiz. soft key
$19C	FBACK	Horiz. soft key
$19D	FNEXT (horizontal)	Horiz. soft key
$19E	FNEXT-UP	Horiz. soft key
$1A0	C-UP	Cursor

Code	Key	Group
$1A1	C-DOWN	Cursor
$1A2	C-LEFT	Cursor
$1A3	C-RIGHT	Cursor
$1A8	ENTER	Cursor
$1A9	NO-ENTER	Cursor
$1AB	DEL	Cursor
$1AC	END BLOCK	Cursor
$1AD	GOTO	Cursor
$1AE	CE	
$1B0	X	
$1B1	Y	
$1B2	Z	
$1B3	IV	
$1B4	V	
$1B8	POLAR	
$1B9	INCREMENT	
$1BA	Q	
$1BB	ACTPOS	Numbers
$1BC	–	Numbers
$1BD	.	Numbers
$1C0	MANUAL	Operating mode
$1C1	TEACH-IN	Operating mode
$1C2	SINGLE	Operating mode
$1C3	AUTO	Operating mode
$1C4	EDIT	Operating mode
$1C5	HANDWHEEL	Operating mode
$1C6	TEST	Operating mode
$1C7	MOD	
$1CB	PGM MGT	
$1D0	PGM-CALL	Block opening
$1D1	TOOL DEF	Block opening
$1D2	TOOL CALL	Block opening
$1D3	CYCL DEF	Block opening
$1D4	CYCL CALL	Block opening
$1D5	LBL SET	Block opening
$1D6	LBL CALL	Block opening
$1D7	L	Block opening
$1D8	C	Block opening
$1D9	CR	Block opening
$1DA	CT	Block opening
$1DB	CC	Block opening
$1DC	RND	Block opening

Code	Key	Group
$1DD	CHF	Block opening
$1DE	FK	Block opening
$1DF	TOUCH-PROBE	Block opening
$1E0	STOP	Block opening
$1E1	APPR/DEP	Block opening
$1EA	DIA	
$1EB	FIG	
$1EC	Screen switch-over	
$1ED	HELP	
$1EE	INFO	
$1EF	CALC	
$1F0	NC START	

8.4.2 Machine operating panel

At connection X46 there are 25 PLC inputs (I128 to I152) and eight PLC outputs (O0 to O7) for evaluating the keys on the machine operating panel.

You can activate specific functions by linking the PLC inputs with the corresponding markers and words.

You can store the pressing of an axis-direction button:

- With MP7680 bit 0, enable the memory function.
- Use M4562 to save a depressed axis direction key. This means that the axis will move until there is an NC STOP.

If the LSV2 connection is active, the NC Start and NC Stop commands can be transmitted. PLC Marker M4230 functions like NC Start, and M4231 like NC Stop, unless the PLC markers are reset by the PLC program. The two markers for NC Start (M4564, M4230) are OR-gated in the PLC run-time system. The two markers for NC Stop (M4560, M4231) are AND-gated (if one marker = 0, then an NC Stop is triggered). After evaluating the states, the two markers of the LSV2 connection are set to their original state again (M4230=0, M4231=1).

MP7680 **Machine parameter with multiple function**
Input: Bit 0 – Memory function for axis-direction keys with M4562:
0: Not saved
1: Saved if M4562 is set

		Set	Reset
W1046	**Manual traverse in positive direction** Bits 0 to 13 correspond to axes 1 to 14: 0: Do not move axis 1: Move axis	PLC	PLC

		Set	Reset
W1048	**Manual traverse in negative direction** Bits 0 to 13 correspond to axes 1 to 14: 0: Do not move axis 1: Move axis	PLC	PLC

		Set	Reset
M4230	**NC start via LSV2**	NC	NC
M4231	**NC stop via LSV2**	NC	NC
M4560	**NC stop (0: stop)**	PLC	PLC
M4561	**Rapid traverse**	PLC	PLC
M4562	**Memory function for axis direction keys (MP7680 bit 0 = 1)**	PLC	PLC
M4564	**NC start**	PLC	PLC

8.4.3 Touchpad on USB port

In order to prevent accidental entries via the touchpad, you can lock it via the PLC with Module 9185.

This module is used to disable or enable entries via screen pointing devices that are attached to the USB port (X141/X142). The status of the locking can also be interrogated.

Module 9185 Touchpad status

Call:

PS	B/W/D/K	<Mode> 0: Enable touchpad (UNLOCK) 1: Lock touchpad (LOCK) 2: Status request
CM	9185	
PL	B/W/D	<Status> -1: Status not defined 0: Touchpad is enabled 1: Touchpad is locked

Error recognition:

Marker	Value	Meaning
M4203	0	Function was performed
	1	Error code in W1022
W1022	2	Invalid mode

8.5 Files

The iTNC enables you to edit various file types. File types are identified by an extension after the file name.

Disable soft keys for file types

With the **SELECT TYPE** soft key you can display a soft key for each file type:

- Select MP7224.0 to disable soft keys of specific file types.

Disabling file types for editing

Protected files cannot be edited or changed. They are displayed in the file overview with the color defined in MP7354.1 or MP7355.1, and if the EDIT ON OFF soft key is pressed, the **Protected file!** error message appears:

- Choose with MP7224.1 the file type that you want to protect.
- Choose with MP7224.2 the file type whose **EDIT ON OFF** soft key is to be disabled.

MP7224	**Lock specific file types**
Input:	Bit 0 – HEIDENHAIN programs *.H Bit 1 – ISO programs *.I Bit 2 – Tool tables *.T Bit 3 – Datum tables *.D Bit 4 – Pallet tables *.P Bit 5 – Text files *.A Bit 6 – Reserved Bit 7 – Point tables *.PNT 0: Do not disable 1: Disable
MP7224.0	Disable soft keys for file types
MP7224.1	Protect file types
MP7224.2	Disable the EDIT ON/OFF soft key

You can also set access levels with PLC Module 9285 to protect file types from being edited, see page 1158.

Selecting a file

If you are in the **Program Run, Single Block** or **Program Run, Full Sequence** operating modes, you can select a file via the PLC. W1018 returns the number of files opened by the PLC. W1020 returns the number of all open files.

- With Module 9290, transfer the name of the file to be selected.

Module 9290 Select a file
You can select a file in the **Program Run, Single Block** or **Program Run, Full Sequence** operating modes.

Call:
PS B/W/D/K/S<String number or file>
CM 9290

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid string was transferred
	8	Control is not in the Program Run, Single Block or Program Run, Full Sequence operating mode
	20	Module was not called in a spawn job or submit job
	29	Selected file is invalid or does not exist

Copying, renaming or deleting a file

Module 9248 is used to copy, rename and delete files. This module can be used to access all mounted drives.

Module 9248 Copying, renaming and deleting files
Any necessary file conversions (ASCII --> binary, binary --> ASCII) are performed automatically during the copy process.

Conditions:

- The file names must contain drive information (e.g. PLC:) and file types.
- The file types are used to determine whether a conversion is necessary.
- If a conversion must be performed, then the file types must be identical.
- The file types are used to determine whether renaming is permissible. If the file type remains the same during renaming, then there are no limitations. If the file type is changed, then renaming is only permitted between certain file types.
- Dependencies with other files are not considered when deleting files.

Call:

```
PS      B/WD/K/S <Name of the source file>
PS      B/W/D/K/S<Name of the target file>
                 Only with mode 0 or 1
PS      B/W/D/K  <Mode>
                 0: Copy
                 1: Rename
                 2: Delete
CM      9248
```

Error recognition:

Marker	Value	Meaning
M4203	0	Successful execution of module
	1	Error code in W1022
W1022	2	Module was called in an invalid mode setting
	3	Invalid source string or target string
	7	Error during file conversion, invalid source or target string, error during copying without file conversion, or source file does not exist
	20	Module was not called in a spawn or submit job
	29	Copying or renaming not allowed for the current file types
	36	Error during renaming or deleting. More information about the error can be requested with Module 9149, immediately after Module 9248 has been called. Or an error occurred during a copying process without conversion.

		Set	Reset
W1018	**Number of files opened by the PLC**	NC	NC
W1020	**Number of all open files**	NC	NC

8.5.1 Datum tables (*.D)

You can define up to 255 different datums in a datum table:

- In MP7226.1, define the size of the table.

With Cycle 7 **DATUM SHIFT**, you can enter the new datum with absolute coordinates or specify a line number from the datum table (see the User's Manual).

With Modules 9092 to 9094 you can use the PLC to read from and write to the current datum table (see "Tool and pocket number" on page 1513).

With **FN17: SYSWRITE** and **FN18: SYSREAD** you can read and overwrite values in the datum table (OEM cycles). See pages 1659 and 1671.

MP7226.1 **Size of the datum table**
Input: 0 to 255 [lines]

Reference for values in the datum table

The values from the datum table can be interpreted with respect to the workpiece datum or to the machine datum (MP960.x):

- Enter the datum in MP7475.

MP7475 **Reference for datum table**
Input: 0: Reference is workpiece datum
1: Reference is machine datum (MP960.x)

If MP7475 = 1 is programmed for NC software 340 422-x, and if Cycle 7 (DATUM SHIFT) is entered in the NC program, the error message **Use preset table!** appears.

8.5.2 Freely definable tables

You can adapt tables to suit your own applications:

- Define the number and names of the fields as prototypes.

You can interrogate and edit the entries through PLC modules or through the FN functions **FN26: TABOPEN, FN27: TABWRITE** and **FN28: TABREAD** (see the User's Manual).

With the exception of pallet tables and cutting data tables, freely definable tables are given the file name extension .TAB.

Creating a prototype

- Switch to PLC mode.
  (See "Selecting the PLC mode" on page9 – 1604).
- In the **PLC:\PROTO** directory, create a table with the extension .TAB.

If you have not yet defined a prototype, a standard prototype will be offered. If you have defined more than one prototype, a menu will appear when you create a table:

- Select an existing prototype and change the format by using the soft key **EDIT FORMAT**.

If you have selected a prototype, the structure commands of the individual columns are displayed:

- **NAME:** Heading of the column. Maximum 8 characters, no longer than WIDTH. Do not use any blanks.
- **TYPE:**
  - **N** = Numerical input (with "$"in hexadecimal and "%" in binary format)
  - **C** = Alphanumeric input
  - **X** = Date-stamp function; if the line in the table is changed, then depending on the length defined in **WIDTH,** the current time (**WIDTH**=8) or the current time and date (**WIDTH**=19) can be entered in the field. **hh:mm:ss dd.mm.yyyy** is the valid format here.
    The timestamp function is active in combination with write-accesses by the PLC (via module) and from NC programs (FN function TABWRITE, WRITE TO KINEMATIC).
  - **L** = Entry of an integer value (no decimal places); special type for quicker access for machining, such as for position requests.
- **WIDTH:** Width of the column. For **TYPE = N** it includes the algebraic sign, decimal point and decimal places.
- **DEC:** Number of decimal places; = 0 for hexadecimal or binary format input. Has no meaning for **TYPE = C.**
- **ENGLISH** to **RUSSIAN:** Language-specific messages that are shown in the dialog line during editing of the column. Maximum 32 characters per language. Dialog entry is optional.

- Press the "Insert line" soft key and enter your structure commands in the respective column.
- With the END key you exit the display of the structure definition. The table you have just created is displayed with the newly defined columns.

Note

A table can have a maximum of 30 columns and a maximum width of 500 characters.

It is also possible to show dialogs with the UTF8 character set in freely definable tables (*.TAB,*.P). This makes it possible to use all conversational languages available on the control.

Data transfer

Valid for tables with the file name extensions .TAB, .P and .CDT:

If a freely definable table is transferred through a data interface, in the externally saved file the structure definition is saved between the lines **#STRUCTBEGIN** and **#STRUCTEND.** The contents of the table are after the line **#STRUCTEND.**

Reading and editing table fields in the PLC

You can read and overwrite table fields in the PLC by using modules. You can give the user access to parts of tables for editing.

Note

The following modules must be called in a submit job or spawn job.

When entering the column names, pay attention to the case of the letters (whether they are small or capital).

Module 9245 Read a field from a table

This module is used to read data from any freely definable table. Module 9245 and the **tableread** function (PLC screen masks) can be used to read data, including strings, from system tables like the tool table and pocket table (*.T and *.TCH).
Open the desired table with the file name extension *.TAB, *.P, *.T or *.TCH with Module 9240, and not in the buffered mode. If an error occurs, the result is undefined.
The module provides the contents as a string.

Call:

```
PS    D           <File handle>
                  from Module 9240
PS    B/W/D/K     <Line>
                  0 to 65 535
PS    B/W/D/K/S<String number, column name>
PS    B/W/D/K/S<String number for the result>
CM    9245
```

Error recognition:

Marker	Value	Meaning
M4203	0	Field was read
	1	Error code in W1022
W1022	1	Line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	7	The table could not be read from
	20	Module was not called in a spawn job or submit job
	29	The opened file is not a valid table
	30	Field name does not exist in table

Module 9255 Reading a field from a table as an integer value

Open the table with the file name extension .TAB or .P with Module 9240, and not in the buffered mode. If an error occurs, the result is undefined.
The module provides the contents as an integer value.

Call:

PS	D	<File handle> from Module 9240
PS	B/W/D/K	<Line> 0 to 65 535
PS	B/W/D/K/S	<String number, column name>
CM	9255	
PL	B/W/D	<Result>

Error recognition:

Marker	Value	Meaning
M4203	0	Field was read
	1	Error code in W1022
W1022		See Module 9245

Module 9246 Writing to a field in a table

Open the table with the file name extension .TAB or .P with Module 9240, and not in the buffered mode.

The field defined by the column name and line number is overwritten.

If a line that does not yet exist is transferred, the file is filled with blank spaces up to the defined line.

The module transfers a string.

Call:

PS	D	<File handle> from Module 9240
PS	B/W/D/K	<Line> –1: Next vacant line 0 to 65 535
PS	B/W/D/K/S<String number, column name>	
PS	B/W/D/K/S<String number, contents to be written>	
CM	9246	

Error recognition:

Marker	Value	Meaning
M4203	0	Field was written to
	1	Error code in W1022
W1022	1	Line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	6	Table is write-protected
	7	Not a numerical field (Module 9256)
	11	The transferred value cannot be saved to the addressed field. Incorrect format
	20	NCMACRO.SYS does not exist
	29	The opened file is not a table with the extension .TAB or .P
	30	Column name not found

Module 9256 Writing an integer value to a field in a table
Open the table with the file name extension .TAB or .P with Module 9240, and not in the "buffered" mode.

The field defined by the column name and line number is overwritten.

This module can be used only for an integer. Values with decimal places are written without the decimal point.

If a line that does not yet exist is transferred, the file is filled with blank spaces up to the defined line.

Call:

PS	D	<File handle> from Module 9240
PS	B/W/D/K	<Line> –1: Next vacant line 0 to 65 535
PS	B/W/D/K/S	<String number, column name>
PS	B/W/D/K	<Numerical value to be written>
CM	9256	

Error recognition:

Marker	Value	Meaning
M4203	0	Field was written to
	1	Error code in W1022
W1022		See Module 9246

Module 9247 Search for a condition in a table
Open the table with Module 9240 not in the "buffered" mode.

The module searches for a field content that fulfills one or more conditions. The conditions are formulated with the commands of the System Query Language (SQL) data bank language. Pay attention to the case of the letters (whether they are small or capital) in the commands and column names.
If you indicate a starting line, the module can search for several suitable field entries.
Permissible SQL commands:

Command	Meaning
+,–,*,/	Arithmetic operators
NOT, AND, OR	Logical operators
<,>, <=, >=, ==, <>	Comparisons
LIKE ´abc´	Text comparison
LIKE ´_abc%´	Partial string
()	Parentheses
MIN(column name)	Minimal value from the column
MAX(column name)	Maximum value from the column

Example:
Search in a pallet table for the line with the NC program 1.H and the set datum X=–10.
String contents:
WHERE (PAL/PGM LIKE´PGM´) AND (NAME LIKE´1.H´) AND (X==–10)

Call:

PS	D	<File handle> from Module 9240
PS	B/W/D/K	<Starting line> 0 to 65 535
PS	B/W/D/K/S	<String number of condition or string with condition>
CM	9247	
PL	B/W/D	<Line that fulfills the condition> –1: Error code in W1022

Error recognition:

Marker	Value	Meaning
W1022	1	Start line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	7	Module could not be read from the table
	20	Module was not called in a spawn job or submit job
	29	Incorrect file format
	30	Column name not found
	31	Syntax error in the transferred condition
	32	No data record found that fulfills the condition

Start the PLC editor for tables

In the machining modes a table editor can be started:

- Specify the lines and columns that are to be displayed.

You can provide the PLC editor only with tables with the file extensions .TAB or .P. A temporary file with the name **SYS:\TEMP\PLCTABED.TAB** is saved.

With Modules 9240, 9241, 9245 and 9247 you can check this temporary file before you place the edited data into the original table with Module 9251.

- Enter the editable columns in the sequence in which they are to be displayed.
  Do not enter the line number! It is displayed automatically.
- Separate the individual columns by a space character. If you have transferred an empty string, all columns of the original table are displayed.
- Enter the first and last line to be displayed on the screen. Line numbering begins with zero. If you enter –1 as the last line, the table will be shown to its end. If you release all lines and columns for editing, you can choose:
  - Whether lines can be deleted and inserted
  - Whether the original table should be edited directly

If you edit the original table directly, you cannot cancel the changes with Module 9251.

If you do not edit the original table directly and the PLC program is recompiled while the PLC editor is open, the editor will be closed without transferring the changes to the original table.

If the END key or the END soft key is pressed while the PLC editor is opened, the NC sets M4159. The PLC editor is **not** closed by the NC. It must be closed by the PLC with Module 9251.

M4159 is reset when Module 9250 is called.

With Module 9035 you can request the active line in the PLC editor.

Module 9250 Start the PLC editor for tables

With this module you start a table editor in the machining modes (compare "Tool Tables").

Call only in a submit job or spawn job.

Call:

```
PS    B/W/D/K/S<String with table name>
                Complete path and name
PS    B/W/D/K/S<String with columns to be edited>
PS    B/W/D/K   <First line >
                [0 to 65 535]
PS    B/W/D/K   <Last line>
                [0 to 65 535]
PS    B/W/D/K   <Mode>
                Bit 0=1: Lines can be inserted and deleted
                (if all lines and columns are selected)
                Bit 1=1: Edit in the original file
                (if all lines and columns are selected)
                Bit 2=0: Shown as a table
                Bit 2=1: Shown as a form
                Bit 3=1: Write-protection of the file to be opened is
                   respected
CM    9250
```

Error recognition:

Marker	Value	Meaning
M4203	0	Editor was opened
	1	Error code in W1022
W1022	1	First and last line do not define a meaningful range, or incorrect value for mode was transferred
	3	Impermissible string number
	6	Write-protected file cannot be opened if write-protection is active (bit#3 = 1)
	7	The module could not read from the table or open the temporary file
	8	The module was called from an operating mode in which the table editor cannot be started.
	20	Module was not called in a spawn job or submit job
	28	PLC editor already open for another table
	29	The opened file is not a table (extension .TAB or .P)
	30	Column name not found
	36	Invalid file name or file type programmed

		Set	Reset
M4159	**PLC editor: END key or soft key pressed**	NC	NC/PLC

Module 9251 End the PLC editor for tables

With this module you end the PLC editor and specify whether the changes are to be put into the original table. The changed values are not checked automatically. Before calling Module 9251 you can read and check the temporary file in the PLC.

Call:

PS B/W/D/K <Mode>
0: Do not place changes into the original file
1: Place changes into the original file

CM 9251

Error recognition:

Marker	Value	Meaning
M4203	0	Editor was opened
	1	Error code in W1022
W1022	3	Incorrect value was transferred for mode
	6	Changes could not be saved in the original table
	20	Module was not called in a spawn job or submit job
	28	PLC editor had not been opened with Module 9250

Module 9252 Position the cursor in the PLC editor

With this module you place the cursor of the PLC editor on a specified line and in a specified column. The line is defined relative to the starting line of Module 9250. The designated column must be defined in Module 9250.

Call:

PS B/W/D/K/S<String with column name>
PS B/W/D/K <Line>
CM 9252

Error recognition:

Marker	Value	Meaning
M4203	0	Cursor was set
	1	Error code in W1022
W1022	1	Incorrect line number
	3	Incorrect string number
	20	Module was not called in a spawn job or submit job
	30	Incorrect column name
	35	PLC editor is not open (Module 9250)

8.5.3 PLC files

With modules you can create PLC files and read or write in them line-by-line. PLC files are in ASCII format and are used, for example, for saving data specific to the PLC.

> **Note**
>
> The following modules must be called only in a submit job or spawn job.

Module 9240 Open a file

You can open up to eight files simultaneously. They are accessed from the process in which they were opened (submit job or spawn job).

If you want to prevent the file from being opened by more than one process, use the "lock file" mode.

To ensure adequate speed, use the "buffered" mode to read from and write to ASCII files. In this mode a part of the file is buffered in the main memory. This mode is not permitted for tables.

When the process is ended (EM in the submit job or spawn job), all files opened for this process are closed.

After the file is opened, Module 9240 always transfers a "file handle." The file handle is a serial number that can be used to select this file again in other modules.

To append data to an existing file, set bit 0=1 (reading and writing) **and** bit 2=0 (record oriented).

Call:

```
PS   B/W/D/K    <Mode>
                Bit 0 = 0: Read only
                Bit 0 = 1: Read and write
                Bit 1 = 0: Do not lock file
                Bit 1 = 1: Lock file
                Bit 2 = 0: Record oriented (tables)
                Bit 2 = 1: Buffered (ASCII files)
                Bit 3 is only evaluated if bit 2 = 0:
                Bit 3 = 0: Do not create file
                Bit 3 = 1: Create file if it does not exist
PS   B/W/D/K/S<String with file name>
                Complete path and file name
CM   9240
PL   D          <File handle>
                Number for use in other modules
                –1: Error code in W1022
```

Error recognition:

Marker	Value	Meaning
M4203	0	File was opened
	1	Error code in W1022
W1022	1	Impermissible mode
	3	Incorrect string number
	7	File could not be opened
	20	Module was not called in a spawn or submit job

Module 9241 Close a file
With this module you close a file that has been opened with Module 9240. You must close the file in the process (submit job or spawn job) in which you opened it.

Call:
PS D <File handle>
Number from Module 9240
CM 9241

Error recognition:

Marker	Value	Meaning
M4203	0	File was closed
	1	Error code in W1022
W1022	2	Incorrect file handle
	20	Module was not called in a spawn or submit job

Module 9242 Positioning in a file

With this module you change the position of the cursor in a file opened with Module 9240. The new position is provided as result from Module 9242.
If the file was opened in the "record oriented" mode (tables), the cursor is positioned line by line.
If the file was opened in the "buffered" mode, the cursor is positioned character by character.
If you indicate a position before the beginning or after the end of the file, the cursor is positioned to the beginning or end of the file, respectively. The addressing of the new position is relative to the beginning or end of the file, or to the current position. You can interrogate the current position by transferring the position value zero relative to the current position.

Call only in a submit job or spawn job.

Call:

PS	D	<File handle> Number from Module 9240
PS	B/W/D/K	<Desired position>
PS	B/W/D/K	<Mode> 0: Position relative to the file beginning 1: Position relative to the current position 2: Position relative to the file end
CM	9242	
PL	B/W/D/K	<New position> –1: Error code in W1022

Error recognition:

Marker	Value	Meaning
M4203	0	Cursor was positioned
	1	Error code in W1022
W1022	1	Impermissible mode
	2	Incorrect file handle
	7	File system error
	20	Module was not called in a spawn job or submit job

Module 9243 Read from a file line by line

To read from a table, use Module 9245.

Open the file with Module 9240.

With Module 9243 you read line-by-line from an ASCII file.

The "buffered" mode provides faster access times. The result is saved in a string. The module reads up to the line break (LF); at most 126 characters.

Call:

PS	D	<File handle> Number from Module 9240
PS	B/W/D/K	<String number for the result>
CM	9243	
PL	B/W/D	<Number of read bytes> >0: Line has been read 0: File end has been reached –1: Error code in W1022

Error recognition:

Marker	Value	Meaning
M4203	0	Line was read
	1	Error code in W1022
W1022	2	Incorrect file handle
	3	Incorrect string number
	7	File system error
	20	Module was not called in a spawn job or submit job

Module 9244 Write to a file line by line

To write to a table, use Module 9246.

With Module 9244 you write line-by-line to an ASCII file.

Open the file with Module 9240.

If file is opened in "buffered" mode:

- Processing time is shorter.
- Files are saved to the hard disk only if more than 512 bytes are overwritten in several calls, or if the file is closed.
- The number of data specified in the transfer string is overwritten.

If file is opened in "record oriented" mode:

- Processing time is longer.
- The data is immediately saved to the hard disk.
- Exactly one line is overwritten. If there is a difference in length, the subsequent data is displaced by the difference.

Call:

PS	D	<File handle> Number from Module 9240
PS	B/W/D/K/S	<String number, source data>
CM	9244	
PL	B/W/D	<Number of written bytes (including LF)> –1: Error code in W1022

Error recognition:

Marker	Value	Meaning
M4203	0	Line was written
	1	Error code in W1022
W1022	2	Incorrect file handle
	3	Incorrect string number
	7	File system error
	20	Module was not called in a spawn job or submit job

8.6 DCM – Dynamic Collision Monitoring

8.6.1 DCM – monitoring the working space for collisions

General information

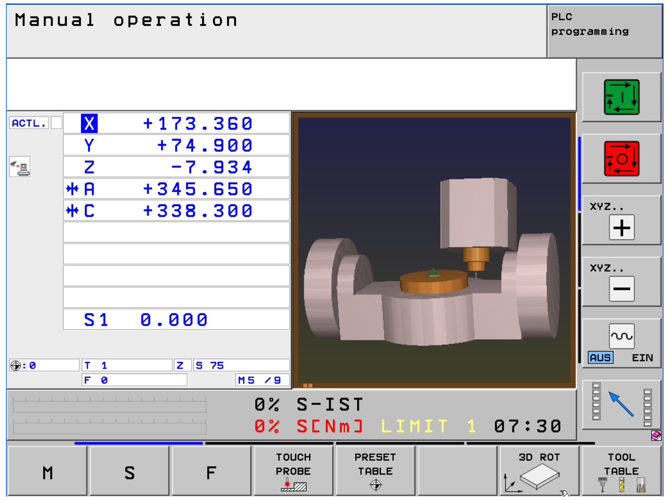
As of software version 340 490-02, the "DCM – Dynamic Collision Monitoring" function (option #40, Id. Nr. 526 452-01) can be used together with an MC 422B unit (or higher) to integrate collision monitoring for various machine elements via the kinematics tables.

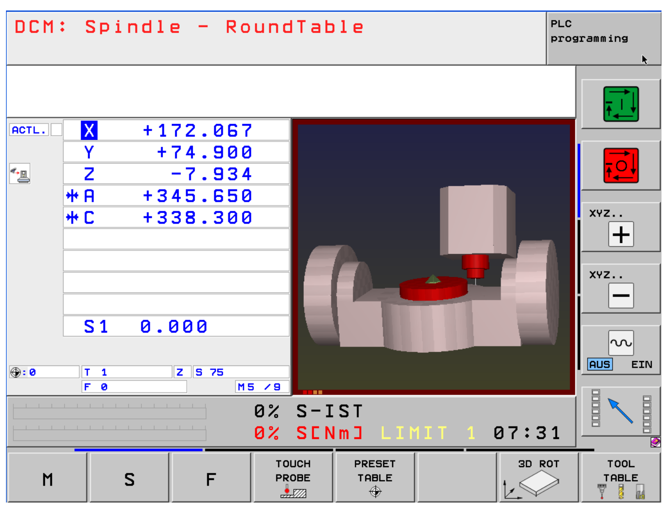
You use **DCM** to define objects (CMOs – "Collision Monitored Objects") within the kinematics description in relation to movable machine axes (X, Y, Z, A, B, C), as they appear on the machine as spatial objects or machine elements. These objects give a three-dimensional image of the machine.

These CMOs, including the **current tool**, an assigned, optional **tool-carrier kinematics description** and an optional **fixture** whose position and orientation have been ascertained, are then taken into account along with the motions of the machine. The active tool is automatically integrated at the tool reference point as a cylindrical object with the dimensions from the tool table. If there is a danger of collision between defined objects, or between the tool and defined objects, then the motions of the machine are stopped, the collision is avoided, and a collision warning or error message is output in the header of the iTNC530 screen. The machine's axes can then only be moved out of the danger zone by acknowledging the collision monitoring or by switching it off.

The Screen Layout key and the `PROGRAM + KINEMATIC` soft key enable you to display 3-D graphics of the collision objects defined in your machine in the Program Run operating modes. The viewer of the KinematicsDesign tool is used for this. However, the collision objects of the machine are only displayed and cannot be edited. A requirement for this to work is FCL 4 or higher and software option #40.

The following figures illustrate the early warning and message regarding a possible collision:





Function and operation

After the software option has been enabled, collision monitoring is permanently active in the **Manual, Positioning with Manual Input, Program Run Single Block, Program Run Full Sequence, smarT.NC** and **El. Handwheel** operating modes. You can switch collision monitoring off in the **Manual** operating mode in the "Collision Monitoring (DCM)" menu that appears when the **Collision** soft key is pressed. It is possible to activate/deactivate **DCM** separately here for the manual and the program-run modes of operation.

At the same time, individual objects can be deactivated at run-time (see "Activating and deactivating monitoring" on page 1388), or generally be excluded relative to other objects via the description table (see "Collision exclusions" on page 1385).

In the **Manual** and **El. Handwheel** operating modes, the iTNC reduces the speed of the traverse motions when objects monitored for collision come into proximity, in order to avoid a collision. This can result in sudden reductions or increases of the feed rate to the maximum permitted speed in the respective manual operating modes. If the objects come within a specified distance of each other, the control stops the machine's motions and outputs an error message. The control makes a distinction between two different zones, which are indicated visually.

- Early warning
  - No message, axes can still be moved
  - Brown frame around the graphical depiction of the kinematics; the affected objects monitored for collision are colored brown
  - Cause: The iTNC reduces the traversing speed in order to safely prevent a collision
  - Solution: Remove the cause of the danger, or retract the axes from the danger zone.
- Errors
  - Message example: **DCM: Spindle - RoundTable**
  - Meaning: Two objects monitored for collision are within approx. 4 mm of each other.
  - Cause: Distance is less than 4 mm [between 0 and 4 mm]
  - Solution: The axis can be moved again if the motion increases the distance between the collision objects (for example by pressing the axis-direction key for the opposite direction): First release the axis-direction key or stop the handwheel, and then press the appropriate axis-direction key or turn the handwheel in the appropriate direction. Acknowledging the error message with the **CE** key is then not necessary.

Machine parameter MP1294 can be set to attain higher traversing speeds in the **Manual Operation** and **Electronic Handwheel** operating modes when DCM is active. If you set MP1294 = 1, you can only move one axis in the **Manual Operation** and **El. Handwheel** operating modes at any one time. This simplifies the calculation of possible collisions and the axis can be moved at a higher feed rate.

MP1294	**Increased traversing speed for DCM by moving only one axis**
Input:	0: Function inactive 1: Function active

If you want to execute a specific motion that reduces the distance between the two collision objects, you must deactivate DCM collision monitoring for the manual operating mode.

- Press the **Collision** soft key.
- In the "Collision Monitoring (DCM)" menu, set it to **Inactive** for manual operation.
- Press the **End** soft key.
- Move the axes manually (axis keys) without collision monitoring.

When finished, reactivate collision monitoring.

- Press the **Collision** soft key.
- Set collision monitoring in the "Collision Monitoring (DCM)" menu to **Active** again.
- Press the **End** soft key.

In the **Positioning with Manual Data Input**, **Program Run Single Block**, **smarT.NC** and **Program Run Full Sequence** operating modes, the motion is considered block-by-block. For example, in the **Program Run, Full Sequence** mode, the NC program is stopped in the block in which the minimum safe distance calculated by the iTNC would exceeded. This safe distance is calculated by the iTNC 530 dynamically. It results from the requirement that the traverse motion can be stopped in time at any time, taking the feed rate and direction into account. As soon as a traverse motion would result in a distance 5 mm less than the calculated safe distance, the motion is stopped or not performed before the traverse block is executed, and an error message is output.

In an NC program with many small traverse blocks and high traverse speeds, this would mean, for example: If two machine components are on a collision course, then after **DCM** has braked and stopped the motion, the components would be between 0 mm and 5 mm from each other.

Attention

Collision monitoring is not active on a machine during reference run (incremental encoders). On machines with absolute path encoders, **DCM** is active as soon as the machine is switched on.

Attention

The following limitations and conditions must also be considered for reliable operation and correct application of DCM:

- **DCM** helps to minimize the danger of collision. However, certain constellations cannot be considered during operation. Please pay attention to the following information.
- **DCM** can only protect machine components from collision if they have been dimensioned correctly, and if their positions in the machine coordinate system have been entered correctly.
- Once **DCM** has been configured, all possible collisions must be tested in the Manual operating mode in order to detect faulty configurations.
- Collisions of defined machine components or the tool with the workpiece are not detected.
- As of software version 340 49x-04, "handwheel superpositioning" with M118 is possible with limited functionality in combination with collision monitoring, see page 1399.
- Please note that collision monitoring does not take any filters into account when calculating the path (e.g. nominal position value filters or angle tolerances for rotary axes in Cycle 32). The path deviation resulting from the nominal position value filters can be ignored. However, the machine manufacturer must calculate and define in MP1292 the deviations of the collision objects (CMOs) from the nominal path resulting from the angle tolerances for rotary axes (Cycle 32).
- **DCM** does not work in combination with the "Free Rotation" PLC function (page 8 – 1144), since this special function for the defined "free" rotary axis runs asynchronously to the system. For the work envelope this means that all CMOs moved with the "free" rotary axis cannot be monitored by **DCM.**
- In the cycles for "Tapping without Floating Tap Holder," you must note that **DCM** does not take into consideration the simple "tracking of the tool axis." In order to use **DCM** for these cycles, bit 4 of MP7160 must be set to 1 (activates the exact interpolation of the tool axis with the spindle).
- If a machine is operated in lag mode or in semi-feedforward mode, then it is assumed that the machine manufacturer takes the greatest possible path deviation into account and increases the size of the collision objects accordingly.
- Even if there are CMOs in the kinematics tables, there is no protection against collisions without software option #40 (DCM).

Attention

Furthermore, the following must be noted when editing kinematics or associated tables. Only if this is followed are changes to the kinematics detected by the collision monitoring:

- If kinematics or an associated table (e.g. CMO descriptions) are edited in an NC program or NC macro with **FN26:TABOPEN** and **FN27:TABWRITE,** then they must be closed after the changes are made. This occurs automatically at the end of a program or if **FN26:TABOPEN** is called again. Activation of the kinematics is permitted with **FN17:SYSWRITE ID290 NR1** here, since this FN17 function automatically closes all tables before activation. If a changed table is activated by the PLC via Module 9097 before it has been closed, the changes are not detected.
- If kinematics or an associated table are opened and edited by the PLC via Modules 9240 through 9246, they must be closed by the PLC after the changes have been made. This must be done before activation of the kinematics (e.g. via Module 9097), a tool call or a change of the tool data with M4538.
- If kinematics or an associated table are edited with the table editor, the editor must be exited with the END or PGM MGT keys. If the changed kinematics are activated (e.g. with PLC Module 9097 or **FN17:SYSWRITE ID290 NR1**) before the editor has been exited, the changes will not take effect.

Compensation of the angle tolerance

So that collision monitoring can function correctly when machining with Cycle 32 and angle tolerances, the maximum occurring angle tolerance must be limited with MP1290, and an oversize for collision objects must be specified with MP1292. The oversize (in MP1292) results from the maximum permissible angle allowance that the machine manufacturer must specify via MP1290 for Cycle 32, and from the dimensions of the swiveled or rotated machine components.

MP1290 and MP1292 are only effective if option #40 (DCM), M128 or TCPM Function and Cycle 32 with angle tolerance are active.

MP1290	**Maximum angle tolerance for DCM**
Input:	0.0000 to 3.0000 [°] Default: 3 [°]

(in combination with option #40)
In connection with DCM (Dynamic Collision Monitoring), the machine manufacturer must enter a default maximum permissible angle tolerance via MP1290 (usually 0.1°). An angle tolerance programmed with Cycle 32 is then limited to this value if collision monitoring is active, i.e. the maximum effective angle tolerance is the value from MP1290. If DCM is switched off (via soft key or by switching to kinematics without collision-object definitions), the value programmed in Cycle 32 is in effect again.
The angle entered in MP1290, in combination with the machine kinematics, is the basis for the allowance defined in MP1292 for collision-object calculations by the control.

MP1292	**Manual oversize for DCM**
Input:	0 to 1000 [mm] Default: 0 [mm]

(in combination with option #40)
For the DCM collision monitoring you enter here the necessary oversizes for the collision objects with a separate rotary axis filter.
The following should be considered for the oversizes:

- For the rotary axes, such as a rotary table, assume the largest possible radius during rotation (usually the table radius), or for a swivel head the distance from the tool tip (longest tool) to the most distant point of the swivel head.
- Use this information to calculate the non-considered offset of the rotary axis. Do so by using the maximum angle tolerance entered in MP1290 to calculate the offset in the following manner:

Example: Swivel head
With maximum tool length: 200 mm, head length: 480 mm,
angle tolerance: MP1290: 0.1 [°]

$s_{offset} = r \cdot \sin(MP1290)$

$s_{offset} = 680\ mm \cdot \sin(0.1°)$

$s_{offset} = 1.19\ mm$

Since the oversize entered in MP1292 is added to each length of the bodies in the calculation, the oversize to be entered in MP1292 must be halved:

$MP1292 = s_{offset} / 2$

MP1292 = 1.19 / 2

MP1292 = **1 [mm]** (rounded up from 0.595 mm)

Since the resolution in MP1292 must be entered as an integer value in millimeters, the values input must be rounded up.

> Note
>
> Please note that when operating with multiple rotary axes, the offsets can summate, and so you must add the oversizes together.
> Example:
> $Oversize_{Tilting\ table}$ = 0.595 [mm]
> $Oversize_{Rotary\ table}$ = 0.396 [mm]
> MP1292 = **1 [mm]** (rounded up from 0.991 mm)

Integration of CMOs in the description table

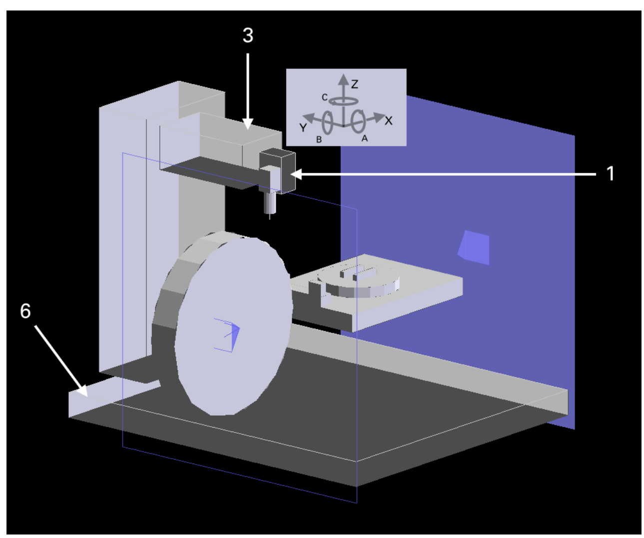
In order to use collision monitoring, the kinematics of the CMOs must be entered in the description table, see page 723.

The starting point for integrating the collision objects is always the tool reference point (e.g. spindle point) and the coordinate system (XYZ) valid for this point. It is then shifted via translations (e.g. X, Y or Z axes) and rotated or tilted via rotations (e.g. A, B or C axes). A collision object (CMO: cuboid, cylinder or plane) is inserted in the description table, taking the momentarily valid coordinate system into account.

The location at which the CMO is placed in the description table is important for the resulting machine and monitoring models. Please see the example below and the associated description table on the next page: The swivel head is defined (via **SUBFILE**) in line 1, and the B axis **(MachAxis B)** is defined in line 2. This means that when the B axis is traversed, all subsequent CMOs and of course the subsequent machine axes (including translations and transformations) will move in the B direction relative to previous CMOs.

In the lower example (machine figure), the portal (no. 3 in the figure) and all subsequent CMOs (no. 6 in the figure) rotate in the B axis direction relative to the double swivel head. The machine coordinate system shown in the figure below (illustrating the traverse directions) is valid.

In the description table, after the portal in line 3, the machine axes X and Y, followed by a machine floor **(CMO_FloorSection)** are defined. This means that when the X or Y axis is traversed, the machine floor **(CMO_FloorSection)** moves in X or Y direction relative to the previously defined portal **(CMO_Portal)**. The coordinate system shown is again valid.



NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST
0	TOOLFILE					
1	SUBFILE1					
2	MachAxis	B				
3	CMO				CMO_Portal	
4	MachAxis	X				
5	MachAxis	Y				
6	CMO				CMO_FloorSection	
7	Trans	X	470.0L92			
8	Trans	Y	-282.405			
9	Trans	Z	-900			
10	CMO			1	CMO_Cabin	
11	MachAxis	Z				
12	CMO				CMO_LiftTable	CMO_Cabin
13	MachAxis	C				
14	CMO				CMO_TurnTable	CMO_LiftTable
[END]						

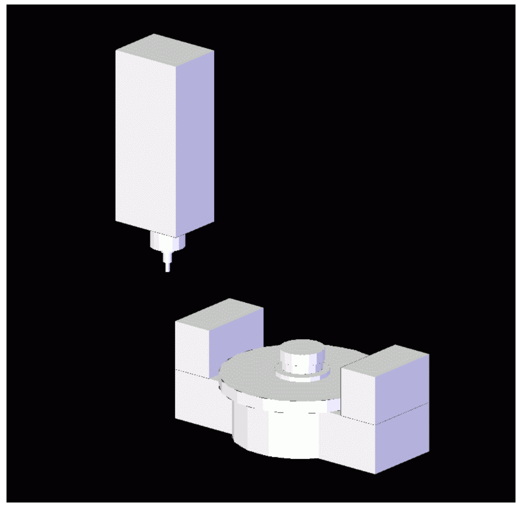
Optimizing the DCM performance

The calculation of the collision objects, as well as the monitoring of their relative positions, is very processor-intensive for the iTNC. This calculation may not consume too many resources, since they are essential to the actual functions.

In order to satisfy this demand, the number of collision objects must be defined such that up to and including NC software 340 49x-05, no more than 50 intersections need be calculated. Thanks to optimizations in software version 340 49x-06, the MC 422C main computer then permits up to 100 intersection calculations. The number of collision objects themselves must be defined in such a way that no more than the above mentioned number of intersection calculations are required. Otherwise increased block processing times could result from the greater calculation efforts. The maximum number of 50 or 100 intersection calculations apply to the complete collision monitoring model, including the monitoring of tool carrier kinematics and fixtures. You therefore need to save a sufficient number of intersection calculations when describing the actual machine so that the number of intersection calculations is not exceeded if you also want to monitor tool-carrier kinematics and fixtures. The number of collision objects defined (CYLINDERs and CUBOIDs in CMO files) is not definite, but rather the number of intersections calculated within the collision object model.

Deactivated collision objects (see "Activating and deactivating monitoring" on page 1388) and all definitions of excluded collisions (see "Collision exclusions" on page 1385) are not included in the calculations. It is therefore important that all necessary calculations of intersections be specified in advance.

Collision objects located in the same axis (CMOs between two "MachAxis…" machine-axis definitions), which due to physical realities can therefore not collide during traverse motions, are automatically **ignored** by the iTNC. This also applies if within two machine-axis definitions, collisions objects are separated by transformations **(TRANS [X,Y,Z,A,B,C]).** However, each of these objects can collide with the individual CMOs traversed in other axes.



In order to determine the number of possible intersections, proceed as follows:

- Ascertain the total number of collision objects defined (CYLINDERs and CUBOIDs) in the kinematics model with the most intersection calculations. Ascertain this number per axis (objects between two "MachAxis…" machine-axis definitions), and take into account any tool-carrier kinematics (TOOLFILE) which might be activated at run-time.
- Ascertain the number of possible intersections (collisions) by proceeding as shown in the following example (see figure above):

No.	Axis	No. of objects	Name
1	–	5	Tool, TOOLFILE and spindle with housing
2	X		
3	Y		
4	Z		
5	A		
6	–	3	Tilting table with shank
7	C		
8	–	4	Rotary table with chuck
9			

Calculation:

No. of objects1 = $5 \cdot (3 + 4) = 35$

No. of objects2 = $3 \cdot 4 = 12$

Intersections = No. of objects1 + No. of objects2 = **47**

Collision exclusions

Definition of collision exclusions

In order to avoid unnecessary calculations, collisions which cannot occur due to physical realities can be excluded from each other in the description table by using the **DONTTEST** column. This is useful since it saves processing time and avoids unnecessary reductions in the feed rate in the manual operating modes.

Two different methods are available

1. Exclusions via existing definition tables of collision-monitored objects

You can switch off monitoring relative to each other with existing definition tables by using the **DONTTEST** column. One or more objects can be defined in a CMO table. Collision monitoring is switched off for all objects defined in this table relative to the objects defined for them in the **FILE** column.

In the description table below, this means that, for example, in line 12 the objects defined in "CMO_LiftTable" will not be monitored for collision with the objects defined in "CMO_Cabin."

NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST
0	TOOLFILE					
1	SUBFILE1					
2	MachAxis	B				
3	CMO				CMO_Portal	
4	MachAxis	X				
5	MachAxis	Y				
6	CMO				CMO_FloorSection	
7	Trans	X	470.0L92			
8	Trans	Y	-282.405			
9	Trans	Z	-900			
10	CMO			1	CMO_Cabin	
11	MachAxis	Z				
12	CMO				CMO_LiftTable	CMO_Cabin
13	MachAxis	C				
14	CMO				CMO_TurnTable	CMO_LiftTable
[END]						

2. Exclusion table

DCM also makes it possible to exclude the collision of an entire machine component (definition table for CMOs) against objects from various other machine components. Here the indication of complete definition tables alone does not always suffice.

From the **DONTTEST** column it is possible to call exclusion tables consisting of only a **DOC** column.

In this column you can enter all names of the various collision-monitored objects (= names in the **DOC** column of the objects in the definition tables). Collision monitoring is switched off for all objects named in this table relative to the objects defined for them in the **FILE** column.

Note

Unique names must be assigned to different CMOs in the **DOC** (iTNC 530) or "Description" (KinematicsDesign) column.

Table **ExcludeDef1**

NR	DOC
0	Rotary table
1	Clamper front
2	Clamper rear
4	Cabin left
5	Cabin right
6	Tool[a]
[END]	

a. "Tool" is a name reserved for the tool by the system. See the special case below for the meaning.

In the description table below, this means that, for example, in line 3 the objects defined in **CMO_Portal** will not be monitored for collision with the objects defined in the name table **ExcludeDef1** above.

NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST
0	TOOLFILE					
1	SUBFILE1					
2	MachAxis	B				
3	CMO				CMO_Portal	ExcludeDef1
4	MachAxis	X				
5	MachAxis	Y				
6	CMO				CMO_FloorSection	
7	Trans	X	470.0L92			
8	Trans	Y	-282.405			
9	Trans	Z	-900			
10	CMO			1	CMO_Cabin	
11	MachAxis	Z				
12	CMO				CMO_LiftTable	
13	MachAxis	C				
14	CMO				CMO_TurnTable	
[END]						

Special case: Tool with the predefined CMO name "Tool"

In some cases it can be necessary to exclude the tool from monitoring for collision with certain machine components (e.g. tool and tool carrier relative to portal for B heads). For mechanical reasons a collision between these components is impossible, but the tool is also permanently monitored for collision with all objects. For example, if the distance between the tool and a machine component with which it cannot collide becomes too small, this could already lead to a reduced feed rate. In this case the tool can be excluded according to the "Exclusion table" method described above by using the "Tool" name to remove this possibility of collision from monitoring.

Activating and deactivating monitoring

Activating and deactivating monitoring

There are various methods for activating and deactivating collision monitoring.

- **Manually**
  In the **Manual** operating mode, press the **Collision** soft key to open the "Collision monitoring (DCM)" menu, and activate or deactivate DCM for the **Manual** and **Program Run** operating modes (active/inactive).
- **PLC module or FN17**
  - It is always possible to create two kinematics tables with the same kinematics description on the control for each application. One of these tables would not contain any descriptions of collision objects (CMOs). Depending on your needs, you can activate the kinematics with or without collision monitoring via the PLC (Module 9097) or with **FN17: SYSWRITE ID290 NR1.**
  - PLC Module 9063 is used to activate and deactivate collision monitoring (DCM) for the Program Run modes.
  - Bit 3 of Module 9221 is used to deactivate collision monitoring for PLC positioning movements. This means that it is possible to position an axis when DCM is active and the reference marks have not yet been traversed in all axes. A PLC positioning command can only be executed if collision monitoring is deactivated for all axes involved in the positioning movement. The deactivation of collision monitoring for the PLC positioning movement does not affect the status information provided by Module 9064.
- **WRITE TO KINEMATIC**
  At the same time, monitoring of these CMOs can be switched off by entering "1" in the **ON/OFF** column of the **active** description table. This can also be during program run in the active description table using the **WRITE TO KINEMATIC** function.

Example:
Deactivation of an active COM table so that collision monitoring is deactivated for a certain action, such as a tool change.

Table in the new kinematics description format before overwriting:

NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST
:						
1	CMO				CMO_Head	PLC:\Kinemat\...
2	MachAxis	B				
3	CMO				CMO_Portal	PLC:\Kinemat\...
4	MachAxis	X				
:						
[END]						

	Comment:
0 BEGIN PGM N545TCM MM	
:	
4 **WRITE TO KINEMATIC AT COLUMN "ON/OFF" CAPTURE "FILE" KEY "CMO_Portal" = 1**	Overwrite cell in the kinematics table
:	

Table after overwriting with collision monitoring switched off for the **CMO_Portal** object:

NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST
:						
1	CMO				CMO_Head	PLC:\Kinemat\...
2	MachAxis	B				
3	CMO			1	CMO_Portal	PLC:\Kinemat\...
4	MachAxis	X				
:						
[END]						

Note

Ensure that no matter which method was used to deactivate collision monitoring, that it is switched on again after the desired action (without collision monitoring) has finished.

DCM in the Test Run mode

As of software version 340 49x-05 it is possible to check a machining process for potential collisions in the **Test Run** operating mode in order to ensure that time-consuming machining processes are not canceled by collision messages. You specify the datum for the collision test performed in this operating mode in the "workpiece blank in working space" MOD function.

For more information on collision monitoring in **Test Run** mode, refer to the User's Manual for the iTNC 530.

Set MP7507 to define the operating modes (editing, simulation or program run modes of operation), in which the SELECT KINEMATICS soft key is to be displayed under MOD.

In order to select kinematics for the simulation:

- While in the Editing/Simulation operating mode, press the MOD key.
- Press the SELECT KINEMATICS soft key.

You can simulate NC programs that are adapted to a certain kinematics model or that switch the kinematics. The kinematics can also be displayed graphically in the simulation:

- Press the screen layout key.
- Press the **KINEMATIC** soft key.

In order to select kinematics for the real machine or for the machining operating modes:

- While in a machining operating mode, press the MOD key.
- Press the SELECT KINEMATICS soft key.

The previous method for selection, via the **KINEMATIC** code number, still exists. It activates the kinematics for both operating modes.

Further constraints:

- After the control has booted, the machine-parameter main file is in effect for the program run and editing/simulation modes of operation.
- The kinematics model selected for simulation (editing mode of operation) is saved in nonvolatile memory.
- PLC switching of the kinematics, individual machine parameters or machine parameter subfiles will only affect the program run modes of operation.
- Changes by the NC program or an NC macro will always affect the current operation mode, in which the NC program or NC macro is running.
- Selecting the machine parameter main file (manually and via PLC) affects all operating modes.

MP7507	**Selecting the kinematics for the operating mode**
Input:	%xxx Bit 0 0: Kinematics cannot be selected in Editing operating modes 1: Kinematics can be selected in Editing operating modes for simulation in Test Run mode Bit 1 0: Kinematics cannot be selected in Machining operating modes 1: Kinematics of the real machine can be selected in Machining operating modes Bit 2 0: 3D ROT soft key is available in Test Run mode 1: 3D ROT soft key is available in Test Run mode

Configuration of the definition table

Definition table for collision-monitored objects (CMO table)

The CMO table contains the descriptions of cuboids, cylinders or planes, which are included as collision-monitored objects at points in the transformation sequence.

It is possible to use uniform kinematics on machines with and without software option #40.

> Danger
>
> This means that an error message does not appear if DCM is not available. Even if there are CMOs in the kinematics tables, there is no protection against collisions without software option #40 (DCM).

A definition table for one or more machine objects can be included at the following points in the transformation chain:

- Description table (main kinematics)
- In **SUBFILE1, SUBFILE2** (partial kinematics via assignment table) and in a **TOOLFILE** (dynamic tool-carrier kinematics during tool change).

Creation of a CMO table:

- In program management, switch to the desired directory and enter the name of the description table, including the extension .TAB.
- Choose the table format with the KEY, X, Y, Z, AXIS, RADIUS, HEIGHT and DOC fields.
- With the aid of the formatting table for a CMO table shown below, enter the description of the collision-monitored object.

> Note
>
> Unique names must be assigned to different CMOs in the **DOC** (iTNC 530) or "Description" (KinematicsDesign) column.

Format of the CMO table:

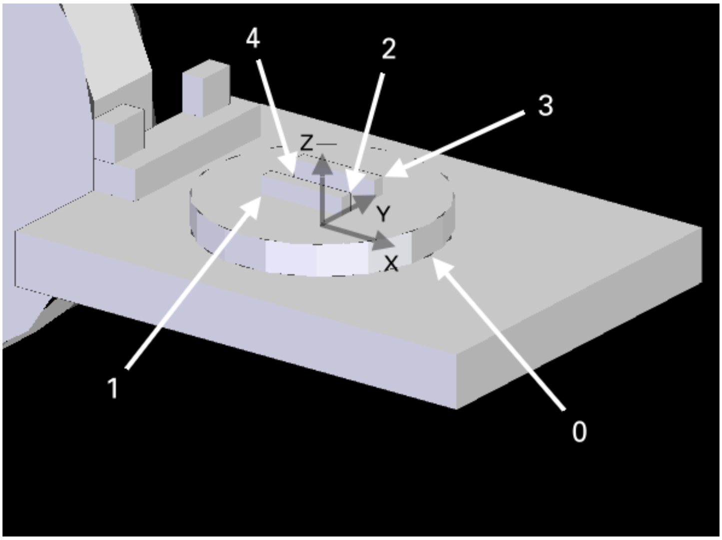
Column	Input	Description
NR	0, 1, 2 ...	Automatic line numbering
KEY	Cylinder	The following parameters for the **X, Y, Z, AXIS, RADIUS,** and **HEIGHT** columns define a cylindrical object, starting from the position of the previous translation, rotation or defined machine axis. Starting from this point, the cylinder is offset from **X, Y** and **Z,** and defined in the direction of the axis entered in **AXIS,** with radius **RADIUS** and height **HEIGHT.**
	Cuboid	The following parameters for the **X, Y** and **Z** columns define a rectangular object, starting from the position of the previous translation, rotation or defined machine axis. Starting from this point, the cuboid is defined from a minimum point in **X, Y** and **Z,** to a maximum point in the next line in **X, Y** and **Z,** (as with the BLK definition in conversational programming).
	LimitMin LimitMax	Here the columns **X, Y** and **Z** can be used to define limiting planes in the X, Y and Z directions.
AXIS	X, Y, Z	Entry of the axis designation for the direction of "Cylinder"-type objects in the **KEY** column
RADIUS	e.g. 40 [mm]	Entry of the radius for "Cylinder"-type collision-monitored objects in the **KEY** column
HEIGHT	e.g. 150 [mm]	Entry of the height for "Cylinder"-type collision-monitored objects in the **KEY** column
DOC	Name of the object[a]	Entry of a unique name for the object entered in the **KEY** column, so that in case of an imminent collision, a clear message regarding the danger of collision can be output in the header (e.g. **Tool <-> Rotary table**)
	TT_ENABLE_PROBING	During tool measurement with a TT, this CMO, which protects the TT, can automatically be excluded from collision monitoring. The CMO will then automatically be excluded during the execution of touch probe cycles TCH PROBE 30/31/32/33 or 480/481/482/483, see page 1398.

a. The "Tool" name is already assigned by the system, and may not be used twice.

Example of a CMO table ("CMO_TurnTable" object: no. 14 from the description table example):

NR	KEY	X	Y	Z	AXIS	RADIUS	HEIGHT	DOC
0	Cylinder	0	0	10	Z	270	70	Rotary table
1	Cuboid	-100	-70	70				Clamper front
2		100	-40	120				
3	Cuboid	-100	40	70				Clamper rear
4		100	70	120				
[END]								

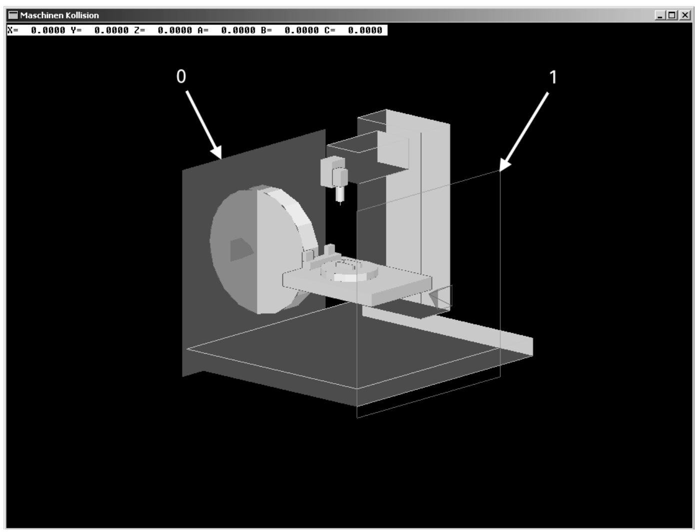
Representation of a collision-monitored object (CMO) with the CMO table above. The coordinate system shown indicates the transformed reference point, which is active in the description table. In this case it is the center point of the rotary table.



Example of a CMO table ("CMO_Cabin" object) as an example for limiting planes:

NR	KEY	X	Y	Z	AXIS	RADIUS	HEIGHT	DOC
0	LimitMin	-1200						Cabin left
1	LimitMax	-1000						Cabin right
[END]								

Representation of limiting planes as in the CMO table above ("CMO_Cabin") with two planes, which were defined in the X direction



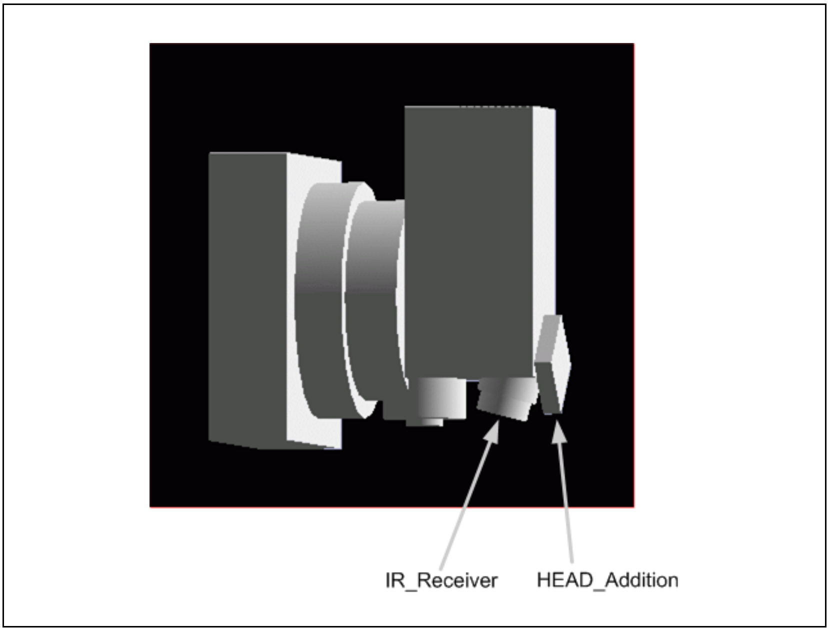
As of software version 340 49x-03 it is easier to describe collision monitored objects (CMO) for collision monitoring. It is now possible to insert the coordinate transformations directly into the object-description file of a CMO.

Shifts and rotations can be entered in a CMO definition table via the keyword **TRANS,** an axis (X, Y, Z, A, B or C) in the **AXIS** column and a value in the new **COORD** column. This functions like entering the description of a kinematics configuration. The coordinate transformations in a CMO definition file are only effective within the respective object-description file, but there it affects all subsequent objects.

This makes it possible in just one object-description file

- to describe rotated and unrotated objects,
- to describe multiple objects after a rotation that then all have the same position, and
- to shift the datum and then perform a rotation.

The following kinematics section will be used as an example to show the changes in the description of the CMOs.



The description of the kinematics with CMOs previously looked like this:

NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST
0	TOOLFILE					
1	CMO				HEAD	
2	Trans	Z	67			
3	Trans	Y	-105			
4	Trans	A	15			
5	CMO				IR_Receiver	
6	Trans	A	-15			
7	Trans	Y	105			
8	Trans	Z	-67			
9	Trans	Y	-140			
10	Trans	Z	70			
11	Trans	B	135			
12	CMO				HEAD_Addition	
13	Trans	B	-135			
14	Trans	Z	-70			
15	Trans	Y	140			
16	Trans	X	0.025			
17	Trans	Z	145			
18	MachAxis	B				
19	CMO				Z-Portal	
20	MachAxis	Z				
21	MachAxis	X				
..	...					
[END]						

The definition table for IR_Receiver:

NR	KEY	X	Y	Z	AXIS	RADIUS	HEIGHT	DOC
0	Cylinder	0	0	-35	Z	35	35	Connector IR_Receiver
1	Cylinder	0	0	-57	Z	30	22	IR_Receiver
[END]								

The definition table for HEAD_Addition:

NR	KEY	X	Y	Z	AXIS	RADIUS	HEIGHT	DOC
0	Cuboid	-40	-20	-40				HEAD Addition
1		40	00	40				
[END]								

The possibility of including the transformation of a CMO into the CMO's own definition table improves the clarity of the kinematics table. This also makes it easier to integrate CMOs, since there are no changes to the actual kinematics of a machine.

The description of the kinematics with CMOs can look like this as of 340 490-03:

NR	KEY	AXIS	COORD	ON/OFF	FILE	DONTTEST
0	TOOLFILE					
1	CMO				HEAD	
2	CMO				IR_Receiver	
3	CMO				HEAD_Addition	
4	Trans	X	0.025			
5	Trans	Z	145			
6	MachAxis	B				
7	CMO				Z-Portal	
8	MachAxis	Z				
9	MachAxis	X				
..	...					
[END]						

Transformations that were only contained in the kinematics tables for the description of CMOs are omitted completely and instead integrated in the CMO definition table.

The new definition table for IR_Receiver:

NR	KEY	X	Y	Z	AXIS	COORD	RADIUS	HEIGHT	DOC
0	Trans				Z	67			
1	Trans				Y	-105			
2	Trans				A	15			Position1
3	Cylinder	0	0	-35	Z		35	35	Connector IR_Receiver
4	Cylinder	0	0	-57	Z		30	22	IR_Receiver
[END]									

As an alternative, the **X, Y** and **Z** columns can be used for the description of transformations, but then no entry is permitted in the **AXIS** column. If there is an entry in the **AXIS** column, the **X, Y** and **Z** columns are ignored.

The new definition table for HEAD_Addition:

NR	KEY	X	Y	Z	AXIS	COORD	RADIUS	HEIGHT	DOC
1	Trans		-140	70					
2	Trans				B	135			
3	Cuboid	-40	-20	-40					HEAD Addition
4		40	00	40					
[END]									

Transformations in the CMO definition table must always occur before the definition of objects. These transformations only serve to describe the position and orientation of the CMOs, and do not need to be canceled in order to describe the further kinematics of the machine. If there is a transformation at the end of a definition table, the error message **Kinematic table defective** is output.

Excluding the TT for tool measurement

During tool measurement with a TT, the CMO protecting the TT can automatically be excluded from collision monitoring. This function is available as of software version 340 39x-04.
A CMO that has been identified correspondingly will then automatically be excluded during the execution of touch probe cycles TCH PROBE 30/31/32/33 or 480/481/482/483. The CMO for the touch probe is identified by entering **TT_ENABLE_PROBING** in the **DOC** column of the kinematics table.
The CMO identified in this way will be excluded from collision monitoring before the beginning of the touch probe cycle. At the end of the cycle or when the cycle is canceled, the CMO will automatically be included in collision monitoring again.

Another possibility for excluding an identified CMO from collision monitoring is to use the function **FN 17: SYSWRITE ID 990**. The function **FN 17: SYSWRITE ID 990 NR6 = 1** causes switching to input X13. The CMO is then excluded from collision monitoring for as long as X13 is active.
The following must be kept in mind:

- You can use **TT_ENABLE_PROBING** for only one CMO.
- If a touch probe cycle is canceled with the EXTERNAL STOP soft key, the tool (TT) should be retracted in the **Manual Operation** mode before pressing the INTERNAL STOP soft key. The INTERNAL STOP soft key results in the CMO being included in collision monitoring again, and the tool would then be immobilized by collision monitoring. DCM would trigger an error message reporting a distance < 2 mm. The tool (TT) can then not be retracted until collision monitoring has been switched off.
- After retraction from the measuring point, the tool (TT) must be located sufficiently far from the area for which protection is active again. After retraction, the distance must be > 5 mm. This safety clearance can be defined in MP6540.x.

Changing CMOs in NC programs

As of software version 340 49x-03, the NC function **WRITE TO KINEMATIC** can be used in order to adapt the position, orientation and size of CMOs to new situations even during execution of an NC program. It is now possible to search active CMO definition tables for keywords **(KEY)** and to overwrite individual cells in the table. When entering keywords in the **DOC** column, make sure that each keyword is only entered once.

> Note
>
> If you change the kinematics tables or CMO definition tables with **WRITE TO KINEMATIC**, then the values in the tables are overwritten. If you do not rescind the changes you made with **WRITE TO KINEMATIC** in the NC program, the new values remain in the table. The changes to the machine kinematics are then not just temporary.

Example for a change to a CMO definition table in an NC program:

	Comment:
0 BEGIN PGM N545TCM MM	
:	
5 WRITE TO KINEMATIC AT COLUMN "COORD" CAPTURE "DOC" KEY "Position1" = 25	Overwrite cell in the kinematics table
6 WRITE TO KINEMATIC AT COLUMN "HEIGHT" CAPTURE "DOC" KEY "IR_Receiver" = 45	Overwrite cell in the kinematics table
:	
43 WRITE TO KINEMATIC AT COLUMN "COORD" CAPTURE "DOC" KEY "Position1" = 15	Write original value to cell in table again
44 WRITE TO KINEMATIC AT COLUMN "HEIGHT" CAPTURE "DOC" KEY "IR_Receiver" = 22	Write original value to cell in table again
:	

DCM and handwheel superimpositioning

As of 340 49x-04, DCM can also monitor handwheel superimpositioning (via M118 or Global Program Settings) for collisions (FCL 4 function). During program run, however, the handwheel is not active.

Handwheel superimposition with active DCM is only possible when program run is stopped with **External Stop** or with the **Single block** operating mode. If an offset is defined for handwheel superimposition after the External Stop, this offset is considered by collision monitoring when the NC program run is continued. This behavior also applies to the virtual tool axis VT.

DCM with counter axes

As of software version 340 49x-03, the changes in position of manually-operated counter axes in the **Manual Operation** and **El. Handwheel** operating modes are detected by the collision monitoring. However, counter axes and controlled axes may not be moved simultaneously if collision monitoring is to be effective.

In the **Positioning with MDI, Program Run – Single Block, Program Run – Full Sequence** and **smarT.NC** machining modes, the counter axes may only be moved at standstill, M strobe, etc., in order to ensure protection by DCM against collisions.

The NC axes can then also be moved in the machining modes while being protected against collision.

Danger

If you position manually-operated counter axes during execution of an NC program or during an individual NC block (during calculation of path interpolations), the new positions of the counter axes cannot be detected or considered by DCM. There is no protection against collisions.

PLC modules

Module 9063 Deactivate and activate collision monitoring

With PLC module 9063, you can influence the status of DCM collision monitoring in the program run modes of operation.

Condition:

- During program run, collision monitoring can only be deactivated during an NC strobe.
- PLC module 9064 enables you to interrogate the status of collision monitoring for the current operating mode.
- It may occur that more than one PLC cycle is executed between activation/deactivation and feedback from the PLC module 9064.

Call:

```
PS   B/W/D/K  <Mode>
              0: Collision monitoring active/inactive during program run
PS   B/W/D/K  <Action>
              0: Deactivate collision monitoring
              1: Activate collision monitoring
CM   9063
```

Error recognition:

Marker	Value	Meaning
M4203	0	Function was performed correctly
	1	Error code in W1022
W1022	1	Invalid value for mode/action
	8	Call was during manual operation mode or during reference run
	21	Call was during program run without strobe signal
	45	Collision monitoring has already been programmed

Module 9064 Status information about collision monitoring

With Module 9064 you interrogate whether collision monitoring is active in the currently selected operating mode.

Call:

PS	B/W/D/K	<Mode> 0: Interrogation of DCM status in the current operating mode
CM	9064	
PL	B/W/D	<Status information> With Mode 0 0: Monitoring not active 1: Monitoring active

Error recognition:

Marker	Value	Meaning
M4203	0	Status ascertained
	1	Error code in W1022
W1022	1	Invalid value for mode

8.6.2 Fixture monitoring with DCM

DCM collision monitoring (software option #40) of the iTNC 530 also protects fixtures from collisions. For this purpose, modeled fixtures can be inserted in the kinematics model of a machine as of software version 340 49x-05.

HEIDENHAIN provides parameterized descriptions of standard fixtures. The FixtureWizard for PCs, which is available free of charge, enables you to use these standard fixtures for describing your own fixtures, and to transfer the description to the iTNC 530.

In the Manual Operation mode, the fixture management function is used to set up the fixtures in the machine's work envelope. An interactive menu allows you to use the integrated measuring cycles to measure and automatically transfer the fixture data, and to define the variable input values, such as the jaw distance of a vise.

The iTNC provides a test program for checking the defined fixtures. In the Program Run, Full Sequence mode the iTNC moves to defined test points and evaluates them. The result is displayed on the screen, or is available as a log file.

That chapter on collision monitoring describes the fundamentals of collision monitoring and the creation of collision-monitored objects (CMOs), see "DCM – monitoring the working space for collisions" on page 1373.
Kinematics descriptions that are used for collision monitoring may contain a maximum of 100 transformation entries (**Trans** and **MachAxis**). However, not all of the active CMOs must be considered. You only have to add the transformations of the CMO with the most transformations to the transformation entries of the actual machine kinematics.

Attention

Finally, the fixture model for collision monitoring, and the behavior of the complete collision model must be tested on the machine as usual.

Using DCM to protect fixtures from collisions requires essentially three steps:

- **1. Modeling fixtures with KinematicsDesign**
  HEIDENHAIN provides fixture templates such as vises or jaw chucks, which were created with the PC program KinematicsDesign, in a fixture template library. You can create additional fixture templates and provide them to your end users. The fixture templates have the file name extension *.cft.
- **2. Defining fixtures with FixtureWizard**
  With the FixtureWizard, the machine operator defines the exact dimensions of the fixture by entering parameter values in the fixture template. The FixtureWizard is available as a stand-alone PC tool and as a component of the iTNC fixture management feature. It generates a placeable fixture with concrete dimensions defined by you. Placeable fixtures have the file name extension *.cfx.
  For information on operating the FixtureWizard, refer to Chapter 11.3 – "Fixture Monitoring" of the User's Manual for the iTNC 530.
- **3. Placing the fixtures in the kinematics model on the iTNC 530 and testing them**
  In an interactive menu whose texts you create in KinematicsDesign, the TNC guides the user through the actual probing process. The probing process consists essentially of the performance of various probing functions on the fixture and entering variable quantities, such as the jaw gap of a vise. After you have placed the fixture, you can have the TNC create a measuring program as needed with which you can have the actual position of the placed fixture compared with the nominal position. If the deviations between the nominal and actual positions are too large, the TNC issues an error message.

The most important functions are described below by using a simple example fixture. Use the fixture templates from HEIDENHAIN as examples if you want to create more complex fixture templates. However, be sure not to change the original HEIDENHAIN templates.

Prerequisites

- The software option #40 (DCM - Dynamic Collision Monitoring) must be enabled.
- Your machine must have a 3-D touch probe for workpiece measurement. Otherwise you cannot place the fixture on the machine.
- `FIXTURE.FUNCTION = YES` must be entered in the OEM.SYS.
- Permissible placement points for the fixtures must be defined in the kinematics description , see page 1405.

Files

The following files are involved in fixture management:

- **Template for a fixture – *.cft**
  Variable template for a fixture. This template is created with KinematicsDesign. Fixture templates from HEIDENHAIN are in the directory **TNC:\system\fixture\JH**. Please create a separate directory for your own fixture templates. Use the FixtureWizard to edit the created *.cft file and to generate a *.cfx file from it.
- **Individual fixture – *.cfx**
  Fixture that can be placed in the machine kinematics, and have its position and orientation ascertained. It only contains variables of the **Manual input field** type that have to be defined when placing the fixture. Once a *.cfx file has been placed and the position and orientation ascertained, a test program can be generated from it, if necessary. The *.cfx files created with FixtureWizard are saved in the same default directory, from which the *.cft file was opened.
- **Test programs for fixtures – *.H**
  Once the fixture has been placed and its position and orientation ascertained (*.cfx), the CREATE TEST PROGRAM soft key becomes available if the fixture contains test points, see page 1424. Press the soft key to create an NC program, which automatically probes the test points in the fixture kinematics. In this way, you can check the position and orientation of the respective fixture.
  Test programs are identified by an index and are always saved in the directory **TNC:\system\Fixture\TpCheck_PGM**. The index is incremented automatically if several identical fixtures are mounted to the same placement point (e.g. vise.001.H, vise.002.H, vise.003.H).
  Please do not change or delete the test programs *.H. The fixture management needs all these files and usually deletes them automatically when the fixture is removed from the machine kinematics.
- **System files for fixture monitoring**
  Other files necessary for fixture monitoring are automatically saved in the directory PLC:\Fixture\. Please do not change or delete these files $tpcheck$.cfg, *.path, *.cfx and *.tab. The fixture management needs all these files and usually deletes them automatically when the fixture is removed from the machine kinematics.

Tools

The following programs are involved in fixture management:

- **KinematicsDesign**
  KinematicsDesign is a program for interactive creation of control kinematics, fixture templates, completed fixtures and tool carrier kinematics. The program is available for download from the HEIDENHAIN homepage, and is available on the iTNC 530 control.
- **FixtureWizard**
  With the FixtureWizard, you use the fixture templates and the templates for the tool carrier kinematics to generate executable kinematics files for the control. These kinematics files can be used for DCM collision monitoring of the control. The program is available for PCs as a part of KinematicsDesign. It is also available on the control.

Defining placement points

As OEM, you must define the location point or the location points of the machine kinematics at which a fixture can be placed. The keyword **Clamp** was introduced for this purpose. The keyword **Clamp** in the column **KEY** of the kinematics table identifies the locations at which fixtures can be placed. The entry in the corresponding **DOC** column is displayed as a dialog when the fixture is placed.

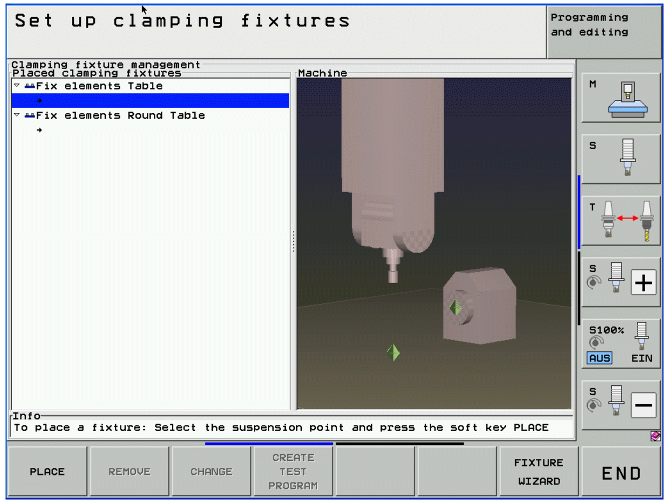
Also, you must specify a file name for a *.tab file in the column **FILE**. In this file, the control saves a path to a *.tab file in the directory **PLC:\Fixture**. After the fixture has been placed and its position and orientation ascertained, this file contains all data relevant for collision monitoring, if the user has assigned values to all open variables of the fixture.

You must also create the tab file for the entry in **FILE**. The file must be of the prototype **CLAMP.TAB**.

Example of two placement points in a kinematics table:

NR	KEY	AXIS	COORD	DOC	FILE
0	TOOLFILE			ToolCarrier	
1	CMO			Spindle	CMO\...
2	CMO			Head	CMO\...
3	Trans	X	+0	Transform X	
4	Trans	Z	+200	Transform Z	
5	MachAxis	B		Axis B	
6	CMO			Fork	CMO\...
7	Trans	X	+0	Transform X	
8	Trans	Y	+0	Transform Y	
9	MachAxis	C		Axis C	
10	CMO			CMO_SlideZ	CMO\...
11	MachAxis	Z		Axis Z	
12	MachAxis	Y		Axis Y	
13	MachAxis	X		Axis X	
14	MachBase			Base	
15	Trans	Z	-700	Transform Z	
16	CMO			Table	CMO\...
17	Clamp			Fixture 1	Fix1.tab
18	Trans	X	+500	Transform X	
19	Trans	Y	+0	Transform Y	
20	Trans	Z	+200	Transform Z	
21	CMO			RoundTableHouse	CMO\...
22	MachAxis	A		Axis A	
23	CMO			RoundTable	CMO\...
24	Clamp			Fixture 2	Fix2.tab
[END]					

The location points defined by you can be selected by the user after pressing the FIXTURE MANAGEMENT soft key in the 3rd soft-key row.



Several different or identical fixtures can be inserted at any location in the description. The automatic assignment of an index to every fixture is used to distinguish between identical fixtures that are inserted at the same location in the description. (e.g. vise.001, vise.002, vise.003)

Modeling fixtures

KinematicsDesign helps you to model the fixtures. You can define readily usable fixtures (file extension *.cfx), or you can create variable fixture templates (file extension *.cft).

For fixture templates (*.cft) you define the shape of the fixture by using variables for different dimensions of the fixture objects (CMOs). The fixtures are designed in the same way as the machine kinematics: CMOs, translations and transformations are used. Input parameters permit variable design of the fixture templates, and enable you to help the end user in the final definition, placement and testing of the fixture.

The *.cft files for fixture templates are in a new table format in ASCII. The tables are an extension of the familiar columns of the kinematics tables *.tab of the machine kinematics, which were specially adapted for the modeling of fixtures. The tables can be opened with Microsoft Excel if the semicolon is selected as separator.

Fixture templates created by HEIDENHAIN are available on the HEIDENHAIN homepage at www.heidenhain.de under:

- Services and Documentation > Software > Fixtures
- Select the desired fixture: You will see a ZIP file with the appropriate files.
- Download the ZIP file and unzip it.
- Copy the unzipped files to the TNC in the **TNC:\system\Fixture\JH** directory.

Save only fixture templates from HEIDENHAIN in the **TNC:\system\Fixture\JH** directory, because this directory will be overwritten during a software update. Fixtures or fixtures templates that were created or changed by you should be saved in a separate directory, such as **TNC:\system\Fixture\OEMfixtures**, or at least under their own names in order to avoid losing data during a software update.

The following parameters or columns with the appropriate input options are available for modeling fixtures:

Format of the CFT/CFX table:

Column	Input	Description
NR	0, 1, 2 ...	Automatic line numbering
KEY	■ CMO	Defines a monitorable collision body in the current kinematics sequence. This machine element is described by the file (table) referenced in the **FILE** column. Fixtures can also contain individual CMOs that have already been defined, see page 1423.
	■ Trans	Here a transformation of the indicated axis (**AXIS** column) in the current kinematics sequence by the value entered in the **COORD** column is performed. This can be a linear translation as well as a rotation about an axis.
	■ Cylinder	The following parameters for the **X, Y, Z, AXIS, RADIUS,** and **HEIGHT** columns define a cylindrical object, starting from the position of the previous translation, rotation or defined machine axis. Starting from this point, the cylinder is offset from X, Y and Z, and defined in the direction of the axis entered in **AXIS,** with radius (**RADIUS**) and height (**HEIGHT**).
	■ Cuboid	The following parameters for the **X**, **Y** and **Z** columns define a rectangular object, starting from the position of the previous translation, rotation or defined machine axis. Starting from this point, the cuboid is defined from a minimum point in X, Y and Z, to a maximum point in the next line in X, Y and Z, (as with the BLK definition in conversational programming).

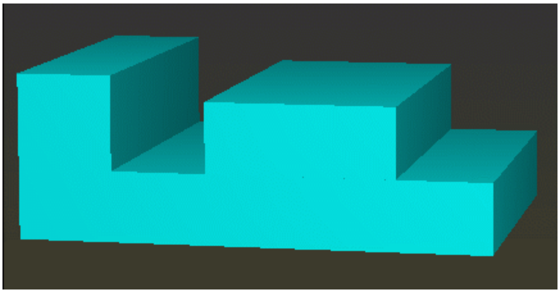
Column	Input	Description
	Only for fixtures (*.cft table):	
KEY	■ Var	Page 8 – 1419
	■ VarParam	Page 8 – 1419
	■ VarTp	Page 8 – 1419
	■ VarEntry	Page 8 – 1419
	■ TpRef	Page 8 – 1413
	■ TpPos	Page 8 – 1413
	■ TpAng	Page 8 – 1413
	■ TpRefAng	Page 8 – 1413
	■ TpCheck	Page 8 – 1413
AXIS	X, Y, Z, A, B, C ...	Entry of the axis designation for function given in the **KEY** column (valid for the **MachAxis**, **Trans** and **TPXxx** parameters in the **KEY** column).
RADIUS	e.g. 40 [mm]	Entry of the radius for "Cylinder"-type collision-monitored objects in the KEY column.
HEIGHT	e.g. 150 [mm]	Entry of the height for "Cylinder"-type collision-monitored objects in the KEY column.
COORD	e.g. 47.092 [mm] or 45.05 [°]	Entry of the transformation value for linear axes (X, Y, Z, ...) in mm or for rotary axes (A, B, C) in degrees. The units are not entered. The iTNC infers the units from the axis designation entered (valid for the **Trans** parameter in the **KEY** column).
FILE	Path and file name[a]	Entry of the path and file name of a definition table for a collision-monitored object (valid for the **CMO** parameter in the **KEY** column).
DOC	Designator	Entry of a designator

Column	Input	Description
Only for fixtures (*.cft/*.cfx table):		
X, Y, Z	Position value	Entry of a position shift in X, Y, Z for "Cylinder" or "Cuboid"-type objects under "Position" in KinematicsDesign. Entry of the position at which the fixture is to be probed during probing cycles.
XSIZE, YSIZE, ZSIZE	Dimensions	Dimensions of a cube within a fixture file
NAME	Name	Name of a variable or a probing cycle
FUNC	Function	Function for determining the value of a variable from a manual probing cycle, see page 1418
VALUE	Value	Value of a variable, see page 1419
RANGE	Value range	Permissible value range of a variable with manual input field. e.g. "1...10"; "100..."; "...5, 10...15, -20"
SUBTYPE	Format	Format of a variable with manual input field, see page 1419.
DEFINCH	Value in inches	This is the default value for the variable when "Inch" is set. Only for linear input, see page 1419.
PICTURE	Help graphic	File with help graphic for a variable or a variable subfixture, see page 1419.
LABELID	Label text	Plain language or text ID for a variable, see page 1419.

a. File paths in *.cft and *.cfx files must be entered relative to the active file.

Example

Example of modeling a simple fixture without variable, which consists of three individual bodies (CMOs).



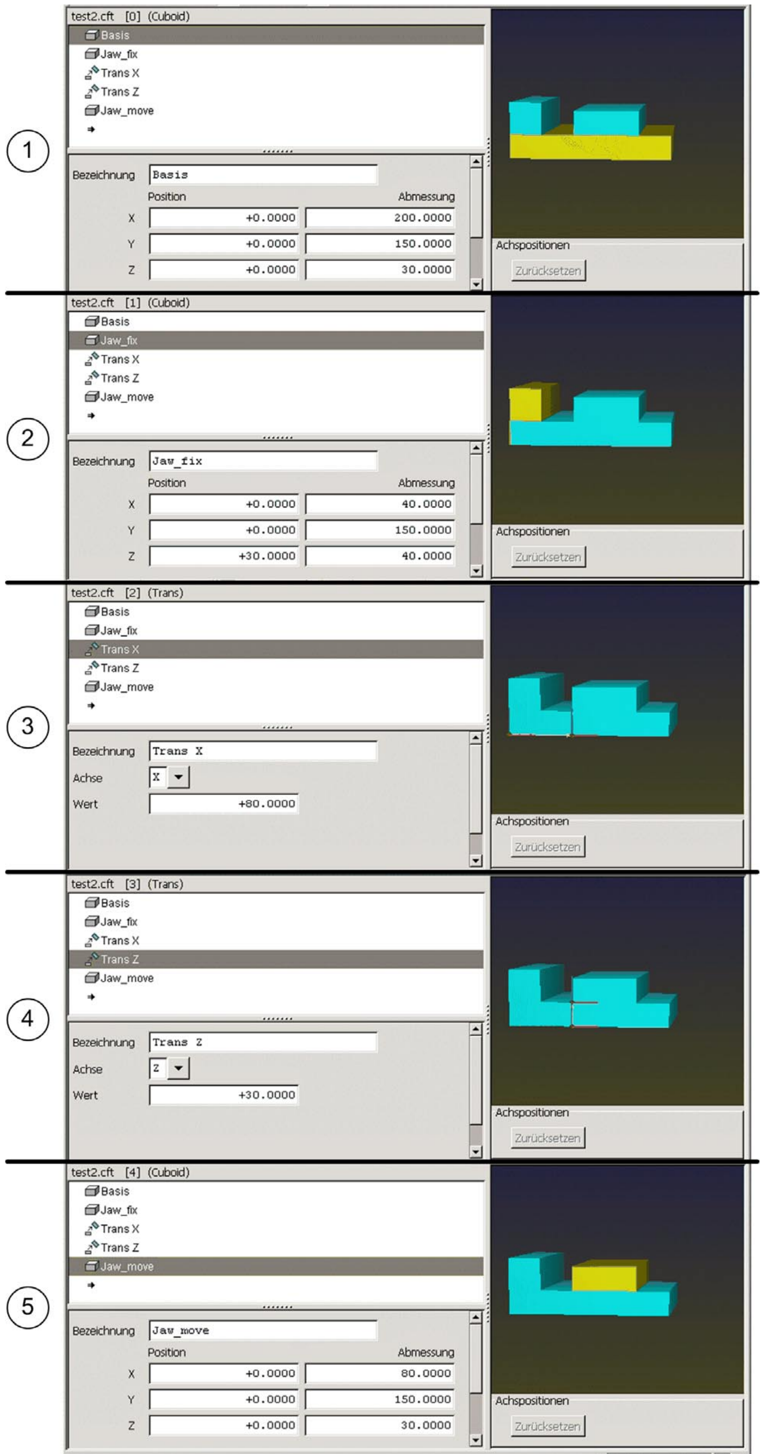
To model a new fixture, proceed as follows:

- Start the KinematicsDesign PC tool.
- Select **File / New...** to create a new file.
- Select:
  Fixture to model a permanent fixture without variable input parameters for the user (*.cfx file), or
  Fixture template to model a variable fixture (*.cft), which will be given its final shape with the FixtureWizard tool.
- Enter any file name for the fixture.
- Design the desired fixture by inserting collision-monitored objects (CMOs) and the required transformations between the individual objects. The generation of the simple fixture shown above is described below in steps 1 to 5.
- KinematicsDesign saves the fixture as a *.cfx or *.cft file, depending on the selection.

Fixture templates *.cft and individual fixtures *.cfx enable the user to enter variable input parameters. Keep the following differences in mind for modeling:

- Design a fixture in the form of a fixture template *.cft in such a way that it can be changed with the FixtureWizard. The user then uses the FixtureWizard to adapt the incomplete fixture template to the actual fixture. This gives the user the possibility to change, for example, the length of the fixture or the jaw height, depending on the application (i.e. the real fixture).
- Design a readily usable fixture *.cfx, and give the user the possibility to change parameters of the otherwise complete fixture, which might not be constant quantities (such as the jaw distance) when ascertaining the position and orientation of the fixture. You enter these changes without using the FixtureWizard. During the process of ascertaining the position and orientation, the user is guided through a menu, in which the editable parameters can be defined.

For information on operating the FixtureWizard, refer to Chapter 11.3 – "Fixture Monitoring" of the User's Manual for the iTNC 530.



When entering the positions of the collision bodies, the values should be entered in the **X**, **Y** and **Z** input fields under **Position** (as in the example in step 2). Additional transformations should not be used (as in this example using **Trans X** and **Trans Z** in steps 3 and 4). The additional transformations result in additional calculations during collision monitoring. This would unnecessarily require computing power.

Probe cycle

Along with the body of the fixture, you must also define the necessary probing points and probing cycles for the operator to clearly and unambiguously ascertain the position and orientation of the fixture. These points and cycles are defined in KinematicsDesign when you model the fixture. You can choose between various probing cycles (see format of the *.cft table).

Input possibilities for probing cycles:

Entry	Input	Description
Description (DOC)	Any text	Designation or name of the object, used in the fixture description.
Type (KEY)	■ TpRef	Probing cycle of type: Datum in plane Defines a manual probing cycle at this point in the fixture kinematics. The type prescribes which probing cycles are available for selection while ascertaining the position and orientation of the fixture.
	■ TpPos	Probing cycle of type: Datum in one axis Defines a manual probing cycle at this point in the fixture kinematics. The type prescribes which probing cycles are available for selection while ascertaining the position and orientation of the fixture.
	■ TpAng	Probing cycle of type: Basic rotation Defines a manual probing cycle at this point in the fixture kinematics. The type prescribes which probing cycles are available for selection while ascertaining the position and orientation of the fixture.
	■ TpRefAng	Probing cycle of type: Basic rotation with corner Defines a manual probing cycle at this point in the fixture kinematics. The type prescribes which probing cycles are available for selection while ascertaining the position and orientation of the fixture.
	■ TpCheck	Probing cycle of type: Test point Defines a manual probing cycle at this point in the fixture kinematics. These probing points are used when creating a test program, see page 1424.
Name (NAME)	Any text	Name for the probing cycle. This name can be used to access the results of the probing cycle.
Axis (AXIS)	Axis identifier	Indication of the main axis for the probing cycle.

Entry	Input	Description
X, Y, Z	Position value	Entry of the position at which the fixture is to be probed.
Label text (LABELID)	Any text	Entry of a plain-language name or text ID. Is displayed for the operator as an additional aid when placing the fixture.
Tilt angles (SPA, SPB, SPC)	Tilting angles in the coordinate system of the fixture	Only for probing cycles of type: Test point (TpCheck). Entry of a tilting angle (spatial) in the coordinate system of the fixture. The tilting angle must be used if a test point cannot be probed without tilting. In that case, the test program automatically tilts the coordinate system for this test point. The tilt angle is only active for this test point. It is automatically reset after probing.

Using the probing cycle, the operator places the fixture at the prescribed point of insertion **(CLAMP)** in the kinematics. Only once the exact position and orientation of the fixture have been ascertained can it then effectively be monitored for collisions.

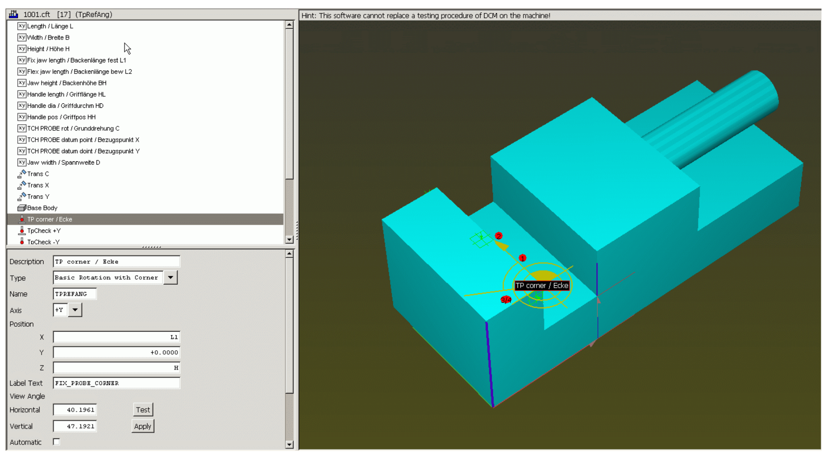
Insertion of a corresponding probing cycle necessitates additional amendments to the fixture kinematics. These amendments are necessary in order for the results of the probing cycle to be included in the fixture kinematics. For example, variable translations or rotations that shift the fixture in the kinematics, depending on the results from the probing cycle, must be inserted.

Proceed as follows to insert a probing cycle into the fixture kinematics:

- Insert the appropriate probing cycle at the corresponding location in the fixture kinematics. Specify the necessary values.
- At the beginning of the fixture kinematics, define variables of the type **manual probing cycle**. In the **Function** entry you specify which result from the probing cycle will be loaded in to the variable, see page 1418. The variable is addressed under the name given in the **Name** entry (e.g. TouchedX).
- After the variables, but before the actual fixture kinematics, insert variable transformations or rotations. These values are determined through the probing cycle, and are then available as variables (e.g. TouchedX).

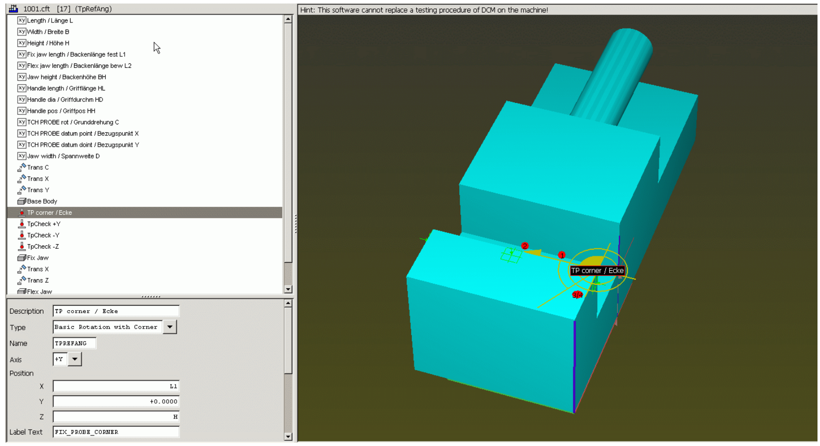
As of KinematicsDesign version 1.3 and NC software 340 49x-06, a touch-probe cycle is no longer necessary for defining a fixture. This, together with the already existing possibility of specifying more than one insertion point in a fixture definition, will make it possible to realize so-called integrated fixture systems.
An integrated fixture system with touch-probe cycles and more than one insertion point can be defined for these systems. The touch-probe cycle serves to define the position of the integrated fixture system. The insertion points serve to define the fixture positions if more than one fixture is used. The individual fixtures that can be installed in the integrated fixture system do not require individual separate touch-probe cycles. Their position and orientation must therefore be unambiguously defined by the integrated fixture system.

A view angle for the fixture can be entered manually for each touch-probe cycle. As before, you can also have KinematicsDesign select the view angle automatically.
Press the **Apply** button to save the current view of the fixture as view angle. Press the **Test** button to rotate the fixture to the position that is saved as view angle.

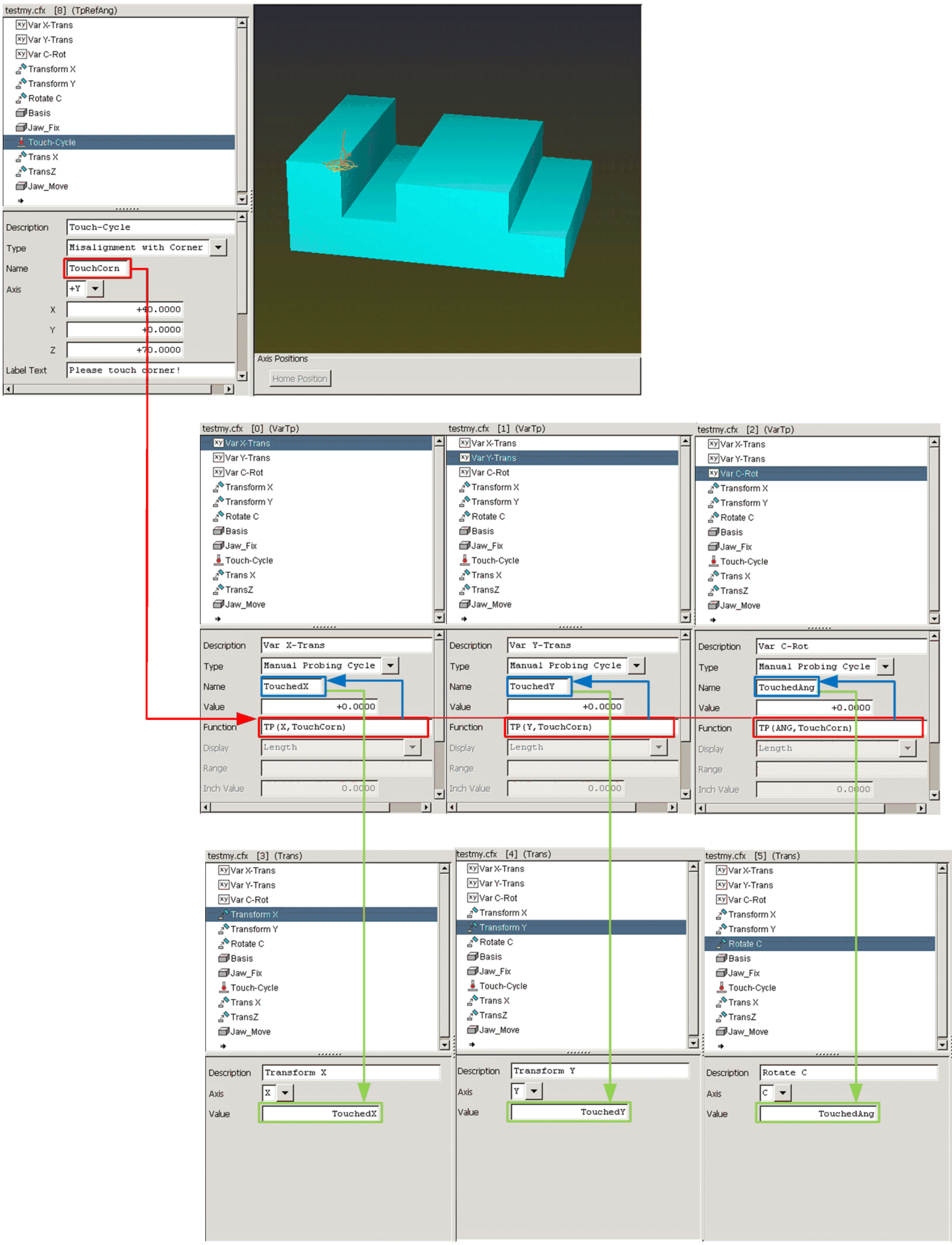


Entry	Input	Description
View angle (VIEWH, VIEWV)	View angle of the fixture	Manual definition of a horizontal and vertical view angle for the fixture. During this touch-probe cycle, the fixture is displayed to the operator at this angle.
View angle – Automatic (AUTOVIEW)	Automatic selection of the view angle	If this option is set, the view angle is calculated by KinematicsDesign and is used to display the fixture. Any manually entered view angles will be ignored.

The touch points required for a touch-probe cycle are automatically numbered to make the measurement of fixtures easier. The numbers inform the operator about the sequence in which the points have to be probed to define the position of the fixture. In the FixtureWizard, the touch points can be shown or hidden as desired.



The following overview illustrates the associations when data is exchanged between the probing cycle and variable transformations. In the example, a probing cycle with the name TouchCorn was inserted. The values from the probing cycle are loaded into the variables TouchedX, TouchedY and TouchedAng via TP functions. These variables, in turn, serve as values for the two transformations and the rotation. This way the exact position and orientation of the fixture is ascertained.



TP functions

You can use the TP functions to read out measured values from the manual probing cycles used in the placement of the fixture and the ascertainment of its position and orientation.

TP functions have the following syntax:
TP(<value>,<cycle name>, {cycle index})

Entry	Input	Description
<Value>	X, Y, Z, ANG	■ X: Principal axis value of a probing cycle ■ Y: Minor axis value of a probing cycle ■ Z: Value in direction of probing ■ ANG: Angular value for rotation in the probing plane
<cycle name>	Text	Name of the probing cycle whose value is to be read.
<cycle index>	Integer (1 to n)	It can happen that the same part is used more than once in the fixture kinematics. This part can include a probing cycle that has the same name even though it is used more than once. Use the <cycle index> parameter to choose between the two cycles. Example: The CMO "Jaw chuck" contains 3 CMOs with the name "Jaw". Each "Jaw" includes a probing cycle with the name "Rotation". If you want to read the value of the rotation on the third "Jaw", the function must be entered as: **TP(ANG,Rotation,3)**

Variables

For the flexible definition of fixtures, KinematicsDesign offers the possibility of creating variables for fixture kinematics. These are used for the placement of the fixture (ascertaining its position and orientation), or for variable length information, for example.

Input possibilities for variables:

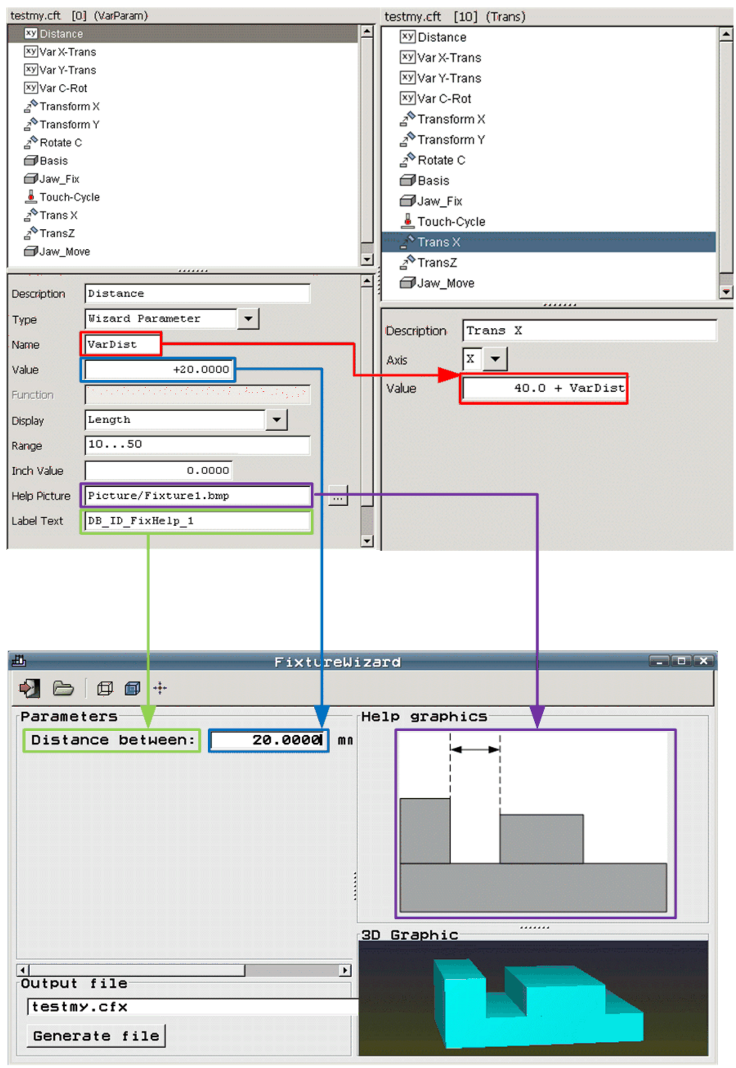
Entry	Input	Description
Description (DOC)	Any text	Object designation used in the fixture description.
Model (KEY)	■ VarEntry	Variable of the type: Manual input field Defines a number variable, whose value must be entered when placing the workpiece via the control's user interface. The value can be changed by the operator without FixtureWizard and without ascertaining the position and orientation of the workpiece again (e.g., the distance between jaws changes, but the vise remains at the same position on the table).
	■ VarTp	Variable of the type: Manual probing cycle Defines a number variable, whose value is determined by a probing cycle when ascertaining the position and orientation of the fixture.
	■ Var	Variable of the type: Miscellaneous variable Defines a miscellaneous variable that can be used with the fixture kinematics, e.g. for the buffering of values. Value assignment by the operator not possible.
	■ VarParam	Variable of the type: Wizard parameter Defines a number variable in which a value can be entered in the FixtureWizard (e.g. variable distance between jaws).
Name (NAME)	Any text	Name of the variable with which it can be addressed.
Value (VALUE)	Numerical value	Value of the variable as a number or a formula for calculation.
Function (FUNC)	Function	Function for determining the value of a variable from a manual probing cycle (only for VarTp).
Format (SUBTYPE)	Position value	Format of a variable with manual input field (only for VarEntry, VarParam): Length: Linear value in [mm] or [inches] Angle: Angular value in degrees [°] Integer: Numeration type, only permits whole numbers Bool: Logical value, active/inactive through check box

Entry	Input	Description
Value range (RANGE)	Numerical range	Permissible value range of a variable with manual input field. e.g. "1...10"; "100..."; "...5, 10...15, -20"
Value in inches (DEFINCH)	Numerical value	This is the default value for the variable when **Inch** is set. Only for linear information.
Help graphic (PICTURE)	Numerical value	Path to the help-graphic file for a variable or a variable subfixture.
Label text (LABELID)	Any text	Entry of a plain-language name or text ID. Is displayed for the operator as an additional aid when placing the fixture. You can use the PLCtext software tool to create the texts as BMX texts. The texts must be located in a **Text** subdirectory of the respective fixture.

Variable of the type **Wizard parameter** as example:

In this example fixture the distance between the jaws is to be variable, and the operator must be able to enter this information in the FixtureWizard. Proceed as follows:

- Insert a variable of the type **Wizard parameter**, and enter a designation for the variable under Name. Under Value range, enter the permissible input range for the variable.
- Enter a default value for the variable in the **Value** field.
- You also have the option of entering a help graphic and a label text, which aids the operator when entering the parameters in the FixtureWizard.
- In this example the distance between jaws is to be variable. Therefore, the transformation that determines the position of the rear jaw must be linked to the variable. In this example, the result of 40.0 + VarDist indicates the position of the rear jaw. The fixed value 40 determines the position of the rear jaw when VarDist = 0 in the example.



Mathematical functions

The final value does not need to be entered directly for any input fields for numbers (e.g. position, transformation, function). These fields can also have formulas entered in them, in order to calculate the value. These formulas may again contain variables.

The C syntax with the usual precedence rules (multiplication/division before addition/subtraction) apply to the formulas. Only standard parentheses () are permitted for parenthetical expressions. The upper/lower case of the keywords listed below is unimportant.

Operation	Keyword / operator
Basic operations	+ - * /
Exponent	^
Constants	Fixed decimal point and floating point numbers with a period as decimal sign and exponent "e", e.g.: 1.234e5
Variables	<Name of the variable>
Trigonometric functions	sin, cos, tan, asin, acos, atan sinh, cosh, tanh, arsinh, arcosh, artanh (For trigonometric functions such as sin, the angle must be given in radians. Use the rad function for this.)
Conversion between degrees and radians	deg, rad
Logarithmic functions	log, ln, exp
Square root	sqrt
Absolute value	abs
Value as a whole number (integer)	int
Algebraic sign function -1 for x < 0 0 for x = 0 1 for x > 0	sign
Circle constant pi	pi

CMO

In a fixture, you can use additional CMOs. These collision monitored objects must exists as files.

Input possibilities for CMOs:

Entry	Input	Description
Description (DOC)	Any text	Object designation used in the fixture description.
File (FILE)	Path and file name	Path and file name of the CMO that is to be used at this spot in the fixture kinematics.
Wizard CMO (FCMO)	Active/Not active via check box	When the Wizard CMO check box is activated, a variable CMO is defined at this position. The selection of the CMO actually used occurs later in the FixtureWizard. The FixtureWizard then offers the operator a browser in which the path and file name of the actual CMO can be entered (such as for selection between step jaws and square jaws).
Help graphic (PICTURE)	Numerical value	Path to the help-graphic file for a variable or a variable subfixture.
Label text (LABELID)	Any text	Entry of a plain-language name or text ID. Is displayed for the operator as an additional aid when placing the fixture. You can use the PLCtext software tool to create the texts as BMX texts. The texts must be located in a **Text** subdirectory of the respective fixture.

Test point

You use probing cycles of the type `Test point` to define probing points in the fixture kinematics, which are used after the fixture has been placed in order to verify the position and orientation of the fixture. If the fixture kinematics contains test points, then once the fixture has been placed and its position and orientation ascertained, the CREATE TEST PROGRAM soft key will be offered to the operator. It is used to create a test program that automatically probes all test points in order to check the position and orientation of the fixture.

Attention

HEIDENHAIN recommends assigning test points to all fixture kinematics. In this way, the operator can check the position and orientation of the fixture.

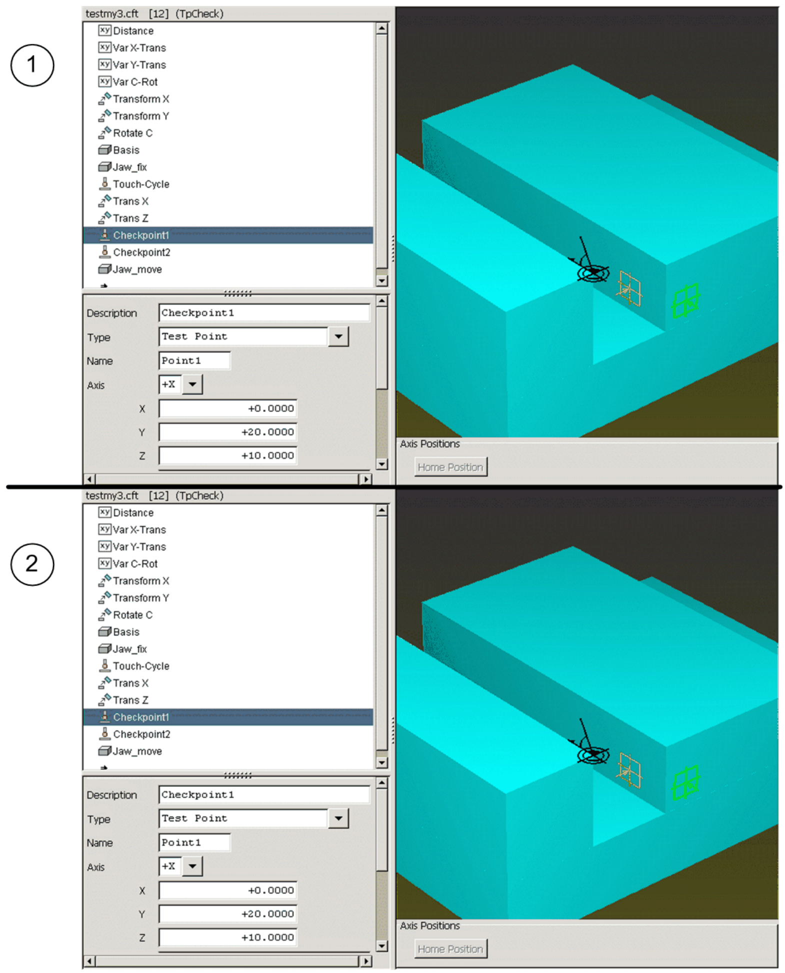
Input possibilities for variables:

Entry	Input	Description
Description (DOC)	Any text	Designation or name of the object, used in the fixture description.
Model (KEY)	■ Test point	Defines a number variable, whose value must be entered when placing the workpiece via the control's user interface. The value can be changed by the operator without FixtureWizard and without ascertaining the position and orientation of the workpiece again (e.g., the distance between jaws changes, but the vise remains at the same position on the table).
Name (NAME)	Any text	Name for the probing cycle. The entry of a name is mandatory for the creation of a test program. It is used internally by the test program.
Axis (AXIS)	Axis identifier	Indication of the main axis for the probing cycle.
X, Y, Z	Position value	Entry of the position at which the fixture is to be probed.
Label text (LABELID)	Any text	For test points without function.

Probing cycle of the type **Test point** as example:

Two test points are to be inserted for the sample fixture, so that the operator can verify both the position and orientation of the fixture, as well as the entry of the distance between jaws. Proceed as follows:

- Insert a probing cycle of the type **Test point**, and enter a designation for the test point under Name.
- Enter the probing direction for the respective test point in the **Axis** field.
- Enter the position of the test point under **X, Y, Z**.



For the procedure for verifying the information of the fixture whose position and orientation were previously ascertained with the TNC, refer to Chapter 11.3 – "Fixture Monitoring" of the User's Manual for the iTNC 530.

The result of the test program's verification is displayed on the screen, and is available as a log file (*.txt). The log file is automatically given the same name as the test program, and is also saved to the same directory.

8.6.3 Tool carrier kinematics and DCM

DCM collision monitoring (software option #40) of the iTNC 530 already protected tool carriers from collisions. As of software version 340 49x-05, the tool carriers to be monitored can be selected from a list in the same way as machine kinematics. If necessary, the operator can select the desired tool-carrier kinematics from the KINEMATIC column in the tool table TOOL.T.

The ASSIGN KINEMATICS soft key is available to the operator in the tool table for this. The list shows the operator all tool-carrier kinematics in *.TAB format that are in the **PLC:\Toolkinematics** directory, as well as all tool-carrier kinematics that are in *.CFX format from the **TNC:\system\Toolkinematics** directory

If the operator selects a *.CFX file from the TNC directory as tool-carrier kinematics for a tool, then when the TOOL CALL for this tool is performed, the file is converted to a *.TAB file in the background. This is then saved to the SYS partition and used for DCM collision monitoring.

> Attention
>
> If you change the carrier kinematics by editing the cfx file, you must re-assign the carrier kinematics description to a tool in the tool table, and execute a TOOL CALL. Not until the TOOL CALL does the TNC convert the cfx file into *.TAB format and then activate the corrected carrier kinematics description.

That chapter on collision monitoring describes the fundamentals of collision monitoring and the creation of collision-monitored objects (CMOs), see "DCM – monitoring the working space for collisions" on page 1373.

> Attention
>
> Finally, the tool-carrier model for collision monitoring and the behavior of the complete collision model must be tested on the machine as usual.

HEIDENHAIN makes tool-carrier kinematics for HEIDENHAIN touch probes available for downloading from its homepage. You will find the carrier kinematics descriptions under www.heidenhain.de > Services and Documentation > Software > Tool-carrier kinematics.

Below are the most important items for the protection of tool carriers by using the features of software version 340 49x-05 and DCM:

- **1. Modeling tool carriers with KinematicsDesign**
  HEIDENHAIN provides its own carrier kinematics descriptions, which were created with the PC program KinematicsDesign, for downloading. You can create additional tool-carrier kinematics descriptions, and provide them to your end users. Templates for carrier kinematics descriptions have the file name extension *.cft
- **2. Defining tool-carrier kinematics descriptions with the FixtureWizard**
  With the FixtureWizard, the machine operator defines the exact dimensions of the tool carrier by entering parameter values in the template. The FixtureWizard is available as a stand-alone PC tool and as a component of the iTNC. It generates a tool carrier with concrete dimensions defined by you. Usable tool-carrier kinematics descriptions have the file name extension *.cfx.
  For information on operating the FixtureWizard, refer to Chapter 11.3 – "Fixture Monitoring" of the User's Manual for the iTNC 530.

The most important functions are described below by using a simple example carrier. Use the tool-carrier templates from HEIDENHAIN as examples if you want to create more complex templates. However, be sure not to change the original HEIDENHAIN templates.

Prerequisites

- The software option #40 (DCM - Dynamic Collision Monitoring) must be enabled.
- Permissible placement points for the tool carriers must be defined in the kinematics description . Do so in the KEY column of the description table via the entry TOOLFILE.

Files

The following files are relevant for tool-carrier kinematics.

- **Template for a tool-carrier kinematics description – *.cft**
  Variable template for a tool-carrier kinematics description. This template is created with KinematicsDesign. Tool-carrier templates from HEIDENHAIN are in the directory `PLC:\Toolkinematics`. Please save your own tool-carrier templates with names differing from those of the HEIDENHAIN templates. Use the FixtureWizard to edit the created *.cft file and to generate a *.cfx file from it.
- **Individual tool-carrier kinematics – *.cfx**
  Tool carrier that can be placed directly in the machine kinematics. The *.cfx files created with FixtureWizard are saved in the same default directory, from which the *.cft file was opened.
- **Kinematics tables for collision monitoring – *.tab**
  If a TOOL CALL calls a tool for which a tool-carrier kinematics description is indicated in the tool table, then *.TAB files are automatically created during the TOOL CALL. These *.TAB files are then taken into account by DCM. Each *.CFX file is converted to a corresponding *.TAB file on the SYS partition. Therefore, all *.CFX files that are necessary for the description of a tool carrier must be on the control.

Tools

The following programs are involved in the creation of tool-carrier kinematics:

- **KinematicsDesign**
  KinematicsDesign is a program for interactive creation of control kinematics, fixture templates, completed fixtures and tool carrier kinematics. The program is available for download from the HEIDENHAIN homepage, and is available on the iTNC 530 control.
- **FixtureWizard**
  With the FixtureWizard, you use the fixture templates and the templates for the tool carrier kinematics to generate executable kinematics files for the control. These kinematics files can be used for DCM collision monitoring of the control. The program is available for PCs as a part of KinematicsDesign. It is also available on the control.

Modeling tool carriers

KinematicsDesign helps you to model the tool carriers. You can define readily usable tool carriers (file extension *.cfx), or you can create variable tool-carrier templates (file extension *.cft).

For tool-carrier templates (*.cft) you define the shape of the tool carrier by using variables for different dimensions of the tool-carrier objects (CMOs). The tool carriers are designed in the same way as the machine kinematics: CMOs, translations and transformations are used. Input parameters permit variable design of the tool-carrier templates, and enable you to help the end user in the final definition of the tool carrier.

The *.cft files for tool-carrier templates are in a new table format in ASCII. The tables are an extension of the familiar columns of the kinematics tables *.tab of the machine kinematics, which were specially adapted for the modeling of fixtures and tool carriers. The tables can be opened with Microsoft Excel if the semicolon is selected as separator.

When saving tool-carrier kinematics to the **TNC:\system\Toolkinematics** directory, only give them names differing from the HEIDENHAIN templates, since the HEIDENHAIN templates will be overwritten during a software update. Tool carriers or tool-carrier templates that were created or changed by you should be saved in a separate directory or under their own names, in order to avoid losing data during a software update.

The following parameters or columns with the appropriate input options are available for modeling tool carriers:

Format of the CFT/CFX table:

Column	Input	Description
NR	0, 1, 2 ...	Automatic line numbering
KEY	■ CMO	Defines a monitorable collision body in the current kinematics sequence. This machine element is described by the file (table) referenced in the **FILE** column.
	■ Trans	Here a transformation of the indicated axis (**AXIS** column) in the current kinematics sequence by the value entered in the **COORD** column is performed. This can be a linear translation as well as a rotation about an axis.
	■ Cylinder	The following parameters for the **X, Y, Z, AXIS, RADIUS,** and **HEIGHT** columns define a cylindrical object, starting from the position of the previous translation, rotation or defined machine axis. Starting from this point, the cylinder is offset from X, Y and Z, and defined in the direction of the axis entered in **AXIS,** with radius (**RADIUS**) and height (**HEIGHT**).
	■ Cuboid	The following parameters for the **X**, **Y** and **Z** columns define a rectangular object, starting from the position of the previous translation, rotation or defined machine axis. Starting from this point, the cuboid is defined from a minimum point in X, Y and Z, to a maximum point in the next line in X, Y and Z, (as with the BLK definition in conversational programming).
	Only for tool carriers (*.cft/*.cfx table):	
KEY	■ Var	Page 8 – 1431
	■ VarParam	Page 8 – 1431
AXIS	X, Y, Z, A, B, C ...	Entry of the axis designation for function given in the **KEY** column (valid for the **MachAxis**, **Trans** and **TPXxx** parameters in the **KEY** column).
RADIUS	E.g. 40 [mm]	Entry of the radius for "Cylinder"-type collision-monitored objects in the KEY column.
HEIGHT	E.g. 150 [mm]	Entry of the height for "Cylinder"-type collision-monitored objects in the KEY column.

Column	Input	Description
COORD	E.g. 47.092 [mm] or 45.05 [°]	Entry of the transformation value for linear axes (X, Y, Z, ...) in mm or for rotary axes (A, B, C) in degrees. The units are not entered. The iTNC infers the units from the axis designation entered (valid for the **Trans** parameter in the **KEY** column).
FILE	Path and file name[a]	Entry of the path and file name of a definition table for a collision-monitored object (valid for the **CMO** parameter in the **KEY** column).
DOC	Designator	Entry of a designator
X, Y, Z	Position value	Entry of a position shift in X, Y, Z for "Cylinder" or "Cuboid"-type objects under "Position" in KinematicsDesign.
XSIZE, YSIZE, ZSIZE	Dimensions	Dimensions of a cube within a tool-carrier kinematics description
NAME	Name	Name of a variable
VALUE	Value	Value of a variable, see page 1431
RANGE	Value range	Permissible value range of a variable with manual input field. e.g. "1...10"; "100..."; "...5, 10...15, -20"
SUBTYPE	Format	Format of a variable with manual input field, see page 1431.
DEFINCH	Value in inches	This is the default value for the variable when "Inch" is set. Only for linear input, see page 1431.
PICTURE	Help graphic	File with a help graphic for a variable, see page 1431
LABELID	Label text	Plain language or text ID for a variable, see page 1431.

a. File paths in *.cft and *.cfx files must be entered relative to the active file.

Variables

For the flexible definition of tool carriers, KinematicsDesign also offers the possibility of creating variables for tool-carrier kinematics. These are used for variable length information, for example.

Input possibilities for variables:

Entry	Input	Description
Description (DOC)	Any text	Object designation that is used in the tool-carrier description.
Model (KEY)	■ Var	Variable of the type: Miscellaneous variable Defines a miscellaneous variable that can be used with the kinematics, e.g. for the buffering of values. Value assignment by the operator not possible.
	■ VarParam	Variable of the type: Wizard parameter Defines a number variable in which a value can be entered in the FixtureWizard (e.g. variable carrier height).
Name (NAME)	Any text	Name of the variable with which it can be addressed.
Value (VALUE)	Numerical value	Value of the variable as a number or a formula for calculation.
Format (SUBTYPE)	Position value	Format of a variable with manual input field (only for VarParam): Length: Linear value in [mm] or [inches] Angle: Angular value in degrees [°] Integer: Numeration type, only permits whole numbers Bool: Logical value, active/inactive through check box
Value range (RANGE)	Numerical range	Permissible value range of a variable with manual input field. e.g. "1...10"; "100..."; "...5, 10...15, -20"
Value in inches (DEFINCH)	Numerical value	This is the default value for the variable when **Inch** is set. Only for linear input.
Help graphic (PICTURE)	Numerical value	Path to a help graphic for a variable.
Label text (LABELID)	Any text	Entry of a plain-language name or text ID. You can use the PLCtext software tool to create the texts as BMX texts. The texts must be located in a **Text** subdirectory of the respective fixture.

Mathematical functions

The final value does not need to be entered directly for any input fields for numbers (e.g. position, transformation). These fields can also have formulas entered in them, in order to calculate the value. These formulas may also contain variables.

The C syntax with the usual precedence rules (multiplication/division before addition/subtraction) apply to the formulas. Only standard parentheses () are permitted for parenthetical expressions. The upper/lower case of the keywords listed below is unimportant.

Operation	Keyword / operator
Basic operations	+ - * /
Exponent	^
Constants	Fixed decimal point and floating point numbers with a period as decimal sign and exponent "e", e.g.: 1.234e5
Variables	<Name of the variable>
Trigonometric functions	sin, cos, tan, asin, acos, atan sinh, cosh, tanh, arsinh, arcosh, artanh (For trigonometric functions such as sin, the angle must be given in radians. Use the rad function for this.)
Conversion between degrees and radians	deg, rad
Logarithmic functions	log, ln, exp
Square root	sqrt
Absolute value	abs
Value as a whole number (integer)	int
Algebraic sign function -1 for x < 0 0 for x = 0 1 for x > 0	sign
Circle constant pi	pi

CMO

In the tool-carrier kinematics, you need to define additional CMOs to describe the collision-monitored objects. These collision monitored objects must exists as files. As of 340 49x-05, the files for CMOs of the tool-carrier kinematics may also be in *.CFT format. Upon a TOOL CALL, the control automatically converts these files of the respective tool to the necessary *.TAB format.

In the definition of a CMO, shifts and rotations of subsequent collision objects can be entered via the keyword TRANS. This functions like entering the description of a kinematics configuration. However, the coordinate transformations in a CMO definition file are only effective within the respective object-description file.

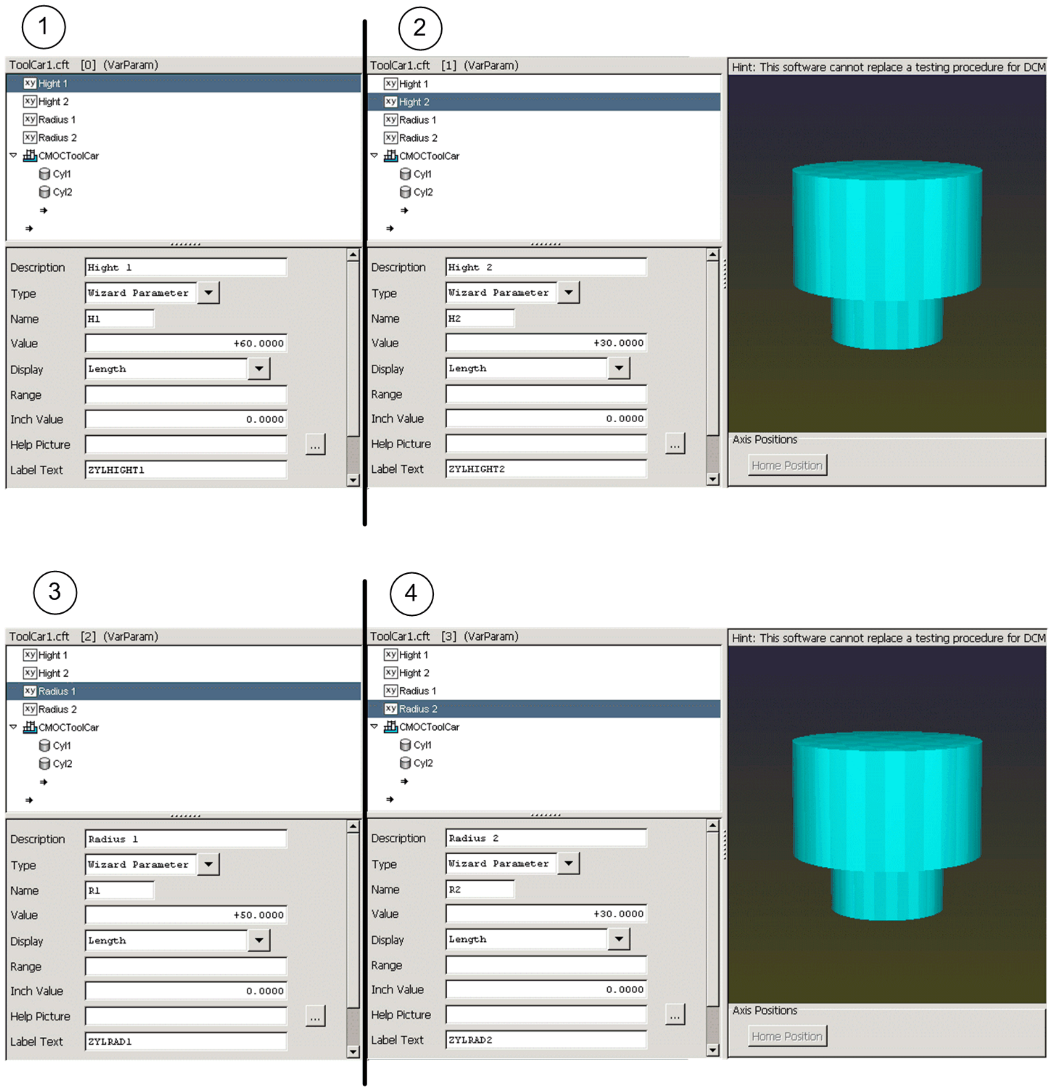
Input possibilities for CMOs:

Entry	Input	Description
Description (DOC)	Any text	Object designation that is used in the tool-carrier description.
File (FILE)	Path and file name	Path and file name of the CMO that is to be used at this spot in the tool-carrier kinematics.
Help graphic (PICTURE)	Numerical value	Path to the help-graphic file for a variable or a variable tool-carrier part.
Label text (LABELID)	Any text	Entry of a plain-language name or text ID. You can use the PLCtext software tool to create the texts as BMX texts. The texts must be located in a **Text** subdirectory of the respective fixture.

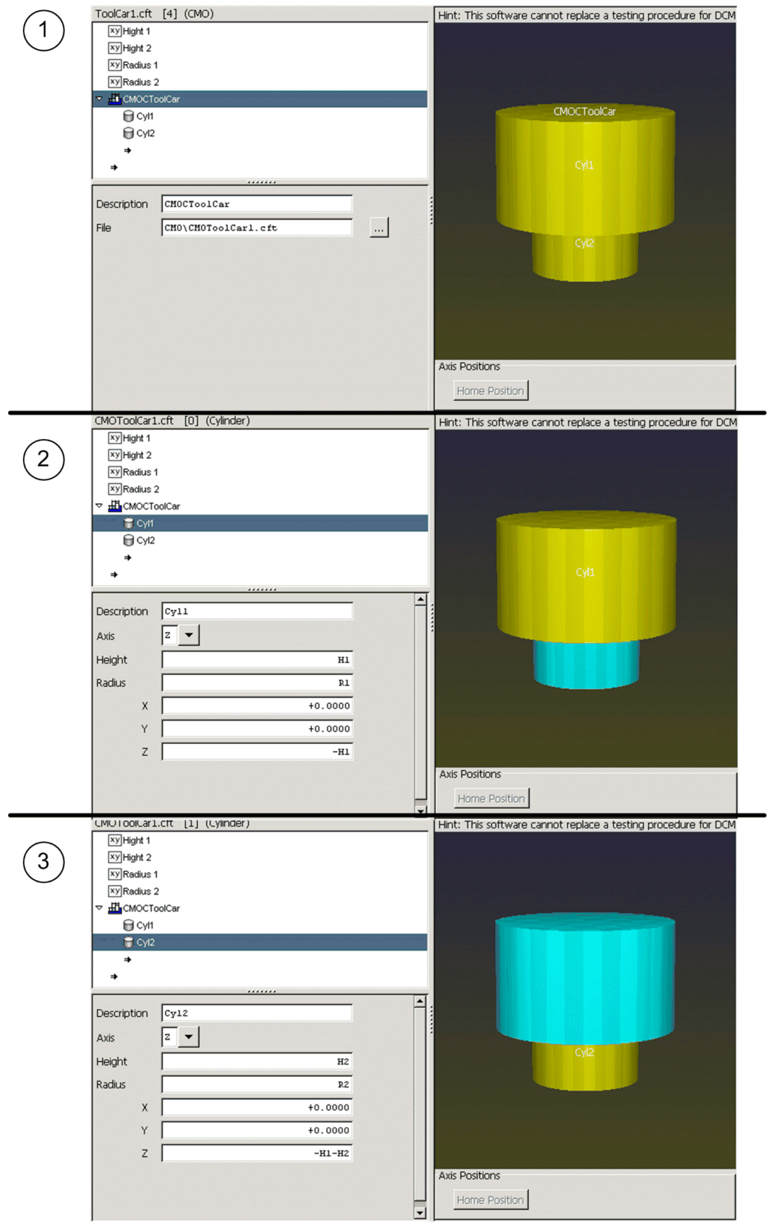
Example

Here is an example of how a simple tool-carrier kinematics description (*.cft template) is created with KinematicsDesign:

- Start the KinematicsDesign PC tool.
- Under **File / New...**, select **Tool-Carrier Template**.
- Assign any file name (the extension .cft is added automatically).
- You can then create your variable **tool-carrier template** or a **tool-carrier kinematics description**. In the example, first four variables are defined for the height and radius of the two cylinders. This means that the appearance of the tool carriers is variable. The height and radius of the two cylinders will later be specified with the FixtureWizard.



- In the example, a CMO is inserted after the variables have been defined. This CMO describes the appearance of the tool carrier, and is specified in a separate file (here CMOToolCar.cft). The cylinders contain the variables for the height and radius, which will later be specified with the FixtureWizard.



- However, the variable tool carrier *.cft cannot yet be used for collision monitoring. You must open the *.cft file with the Fixture Wizard, specify values for all variables, and then save it as a *.cfx file. You can do this on the PC or the control.
  This *.cfx file can then be used as a tool-carrier kinematics description.

The advantage of such a variable template is that you specify the basic shape. The operator can then use the FixtureWizard to define a multitude of tool carriers whose basic shape is the same. Only dimensions that were created as variables can be changed.

8.6.4 KinematicsDesign

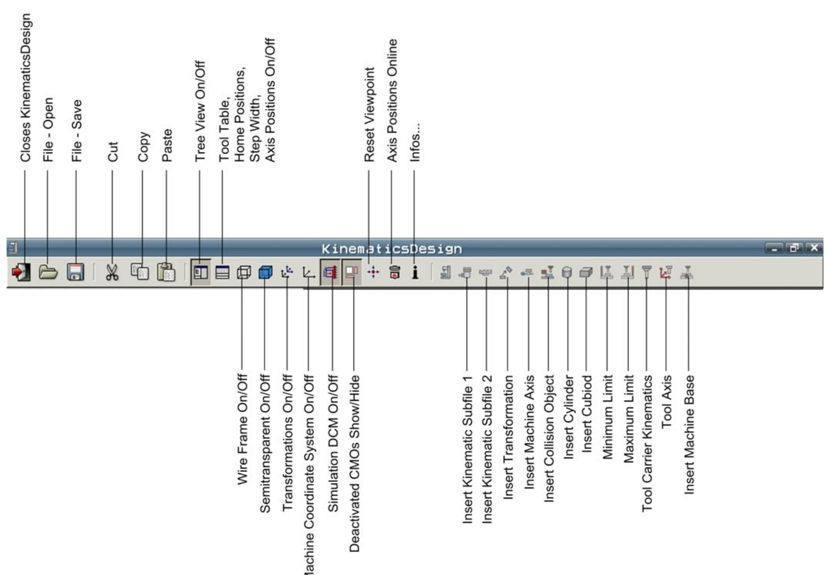
As of software version 340 49x-04, the KinematicsDesign PC tool is also available on the control. The KinematicsDesign version on the control supports you in optimizing, adjusting and changing the kinematics of your machine. The working space and the collision objects of the active kinematics can be displayed and edited. The version in the control contains almost all the functions of the PC version.

The command "File – Save as" was removed, because the active kinematics for editing is automatically saved when the KinematicsDesign version on the control is used.
All other functions of the PC version that you find under "Edit," "View," "Paste," etc. have been implemented in the control version by using appropriate icons.

HEIDENHAIN recommends using a mouse for operating KinematicsDesign.

The documentation of the KinematicsDesign PC tool is stored in the **PLC:\tncguide** directory on the control. The operation of the KinematicsDesign PC tool described in the documentation substantially corresponds to that of the KinematicsDesign version stored in the control.
To start the tool, switch to the PLC mode of operation. Press the **ADVANCED SETUP** soft key, and then the **KINEMATICS** soft key.

To switch between KinematicsDesign and the screen of the control, press the screen switchover key.
The corresponding icons provide you with the following functions:



Danger

- When you use the KinematicsDesign version on your control, you edit and change the active kinematics of your machine. The changes made immediately become effective in the active kinematics.
- The KinematicsDesign software is not a replacement for testing collision monitoring on the machine!

Requirements on the control:

- If no valid kinematics is active on the control, or if the entry **KINEMATIC =** is missing from the OEM.SYS, the error message **Kinematics table missing** appears, and KinematicsDesign is not started.
- If you have created the kinematics description only with the KinematicsDesign version on the control and without the KinematicsDesign PC tool, then no .ini file (KineDesign.ini) of KinematicsDesign is available in the control.
  In this case, it may be necessary to create the file manually. If the machine configuration does not differ from the standard configuration (positive Z axis points up, and the positive X axis to the right) assumed by KinematicsDesign, you do not need to create the KineDesign.ini file. Also, if the kinematics description was transferred to the control by the KinematicsDesign PC tool, the .ini file is created automatically in the control and does not need to be created manually.
  However, if the configuration of your machine differs from the assumed standard configuration, you must manually create the KineDesign.ini file on the control. The file must be stored in the same directory as the KINELIST.tab file. The KineDesign.ini file must be an ASCII file with the two following entries:
  UpwardAxis =
  RightwardAxis =

  Example of KineDesign.ini:
  UpwardAxis = –Z
  RightwardAxis = -X

8.7 Pallet Management

Configuring a pallet table

The pallet table is a "freely definable table":

- Define the prototype in the directory PLC:\PROTO with the file name extension .P.
- Create the prototype.
  (See "Freely definable tables" on page8 – 1358)

If you have more than one prototype with the file name extension .P, a menu for format selection will appear when you create a pallet table. Your PLC program must be adapted to the various formats.

The COPY SAMPLE FILES soft key copies prototypes for the pallet tables into the directory PLC:\PROTO.
The appropriate prototypes are offered when you create a new pallet table. If you do not want this to happen, delete unnecessary prototypes from the PLC:\PROTO directory. Then only existing prototypes are displayed.

- **PROTOTYP.P** = Standard prototype (**PAL/PGM, NAME, DATUM, X, Y, Z**)
- **PROTO_TO.P** = Prototype for tool-oriented machining
- **PROTOPR.P** = Standard prototype for preset tables (as of 340 422-01)
- **PRO_TOPR.P** = Prototype for tool-oriented machining with preset tables (as of 340 422-01)

Field names

The following types of fields are used in the pallet table:

- Mandatory fields: Values must be entered.
- Optional fields: Values can be entered. They have a fixed, predefined meaning for the NC.
- Freely definable fields: You can display additional fields. The names and meaning are defined as desired. The entries are for information, or you can interrogate and change them through the PLC.

Name	Type of machining	Meaning
PAL/PGM	Workpiece-oriented/tool-oriented	Mandatory field: Definition of the entry ■ **PAL** = Pallet ■ **PGM** = NC program ■ **FIX** = Fixture (only tool-oriented)
W-STATUS	Tool-oriented	Optional field: Machining status ■ **BLANK** = Workpiece blank ■ **ENDED** = Machining complete ■ **INCOMPLETE** = Machining not complete
METHOD	Tool-oriented	Mandatory field: Type of machining ■ **TO** = Tool-oriented ■ **WPO** = Workpiece-oriented ■ **CTO** = Tool-oriented for several entries

Name	Type of machining	Meaning
NAME	Workpiece-oriented/tool-oriented	Mandatory field: Name of the pallet or the NC program. NC program names without paths are searched for in the directory with the pallet file. Permit only decimal numbers, so that you can interrogate the pallet's name in the change macro with **FN18: SYSREAD**.
DATUM	Workpiece-oriented/tool-oriented	Optional field: Name of the datum table. Datum tables without paths are searched for in the directory with the pallet tables.
X, Y, Z, U, V, W, A, B, C	Workpiece-oriented/tool-oriented	Optional fields: Definition of the datum[a] In standard format only the columns **X, Y** and **Z** are used.
PRESET	Workpiece-oriented/tool-oriented	Optional field: Definition of the preset by entering a number from the preset table (as of 340 422-01)[a].
SP-X, SP-Y, SP-Z	Tool-oriented	Optional fields: Safe positions; with **FN18: SYSREAD,** these positions can be read in NC macros.
CTID	Tool-oriented	If—due to a tool change—an NC program must be stopped during tool-oriented machining, the iTNC enters a code. This code enables the iTNC to resume the machining process at the position where it has been stopped.
LOCATION	Workpiece-oriented/tool-oriented	Optional field (not used in standard format): Location of pallet. If the **LOCATION** column is used, an NC program can be run only if this column contains the entry **MA** (= pallet for the machine).
LOCK	Workpiece-oriented/tool-oriented	Optional field: Lines containing any entry in this column will not be run. If more than one program or pallet is to be run, the next permitted line is used. Unlocked lines in a locked pallet are also skipped.
PALPRES	Workpiece-oriented/tool-oriented	Optional field: Line number to be activated in the active pallet preset table.
Any names	Workpiece-oriented/tool-oriented	Freely definable

a. For pallet entries the values refer to the machine datum (MP960.x). For NC programs the values refer to the pallet reference point.

Executing a pallet table

The pallet table is selected and started with PGM MGT like a normal part program in the **Program Run, Single Block** and **Program Run, Full Sequence** operating modes.

Pallet entries **(PAL)** and fixture entries **(FIX)** result in a call for an NC macro. Program entries **(PGM)** are run like a PGM CALL.

- With MP7683 bit 6, specify if the NC program and the pallet table should appear simultaneously in the split screen, or if the active NC program or active pallet table should be shown individually.
- With MP7683 bits 0 to 2 and bit 8, specify the operating sequence following an NC start.
- With MP7683 bit 3, specify the operating sequence upon reaching the end of the pallet table.
- Determine with MP7683 bit 4 whether the current pallet table should be editable with the EDIT PALLET soft key in the **Program Run, Single Block** and **Program Run, Full Sequence** operating modes.
- Bit 9 of MP7683 allows you to select whether the EDIT PALLET soft key for editing the pallet table *.P is displayed in the Program Run modes of operation.

As soon as a pallet table is selected, M4160 is set.

Through the PLC you can graphically display the tool changer status in the PLC window and enable the user to control the tool changer through PLC soft keys.

You can provide the user with excerpts from the pallet table for editing (see "Freely definable tables" on page 1358).

With Module 9035 you can interrogate the active line of the pallet file, and with Modules 9090 or 9281 you select a certain line in the pallet table. Unlike Module 9090, a datum shift or datum setting can be executed immediately with Module 9281.

Example:

NR	PAL/PGM	W-STATUS	METHOD	NAME	DATUM	X	Y	Z	SP-X	SP-Y	SP-Z	CTID
0	PAL			120		0	0	0			150	
1	FIX										150	
2	PGM	BLANK	WPO	PART1.H								
3	PGM	BLANK	TO	PART2.H		120	120	0				
4	PAL			130	NULL1.D	0	10	15			150	
5	PGM	BLANK	TO	PART3.H		100	100	100				
6	PGM	BLANK	CTO	PART3B.H								
[END]												

Line 0:
The pallet with the name **120** is defined. The NC macro for changing the pallet is activated. The active datum equals the machine datum. A clearance height was programmed.

Line 1:
A fixture is defined and a clearance height is specified. The NC macro for changing the fixture is active.

Line 2:
The fixture holds an unmachined part to be machined with NC program **PART1.H** (workpiece-oriented machining).

Line 3:
The fixture holds a second unmachined part to be machined with NC program **PART2.H** (tool-oriented machining).
The active datum is offset from the pallet datum by the given values.

Line 4:
The pallet with the name **130** is defined. The NC macro for changing the pallet is activated. The active datum is offset from the machine datum by the given values. The datum table **NULL1.D** is active.

Lines 5 and 6:
The pallet holds two unmachined parts which are to be machined together in one setup with NC programs **PART3.H** and **PART3B.H** (tool-oriented machining). The active datum of the first part is offset from the pallet datum by the given values.

Pallet preset table

As of software version 340 49x-05, a preset table for managing pallet datums can be activated in addition to the preset table for managing workpiece datums. This makes it possible now to manage the pallet datums independently of the workpiece datums.

Pallet datums are an easy way to compensate mechanical differences between individual pallets.

To activate the pallet preset table, the token **PALLETPRESET = <path/file name>** must be entered in OEM.SYS. Any name can be assigned to the pallet preset file *.PR, e.g. PALLETPRESET = TNC:\Palletpreset.pr. Only one workpiece datum and one pallet datum can be active at the same time. Both datums are effective in sum.

The NC function FN17: SYSWRITE ID540 NR1 is also introduced for activating a pallet datum. FN 18:SYSREAD ID 540 NR1 is used to read the active line number. In addition, a pallet datum can be activated by entry of a line number of the active pallet preset table in the **PALPRES** column of the pallet table.

The iTNC displays the number of the active pallet preset in the additional status display.

For determining the pallet datums, there is an additional ENTRY IN PALLET PRES. TAB soft key in the manual probing functions with which you can also store the probing results in the pallet preset table.

Markers and machine parameters

Settings possibilities for markers and machine parameters

		Set	Reset
M4160	**Pallet table selected**	NC	NC

MP7683 **Executing pallet tables and NC programs**

Format: %xxxxx

Input: Bit 0 – PROGRAM RUN, SINGLE BLOCK operating mode:
0: During the start, a line of the NC program is run. The pallet change macro is executed completely.
1: During the start, a complete NC program is run.
Bit 1 – PROGRAM RUN, FULL SEQUENCE operating mode:
0: During the start, a complete NC program is run.
1: At the start all NC programs are executed up to next pallet.
Bit 2 – PROGRAM RUN, FULL SEQUENCE operating mode:
0: As defined in bit 1
1: All NC programs and pallets up to the end of the table are executed.
Bit 3 – When the end of the table is reached, the process begins again with the first line.
0: Function is not in effect
1: Function is effective (bit 2=1)
Bit 4 – Editing the active pallet table
0: Active pallet table cannot be edited.
1: The active pallet can be edited in the PROGRAM RUN, FULL SEQUENCE and PROGRAM RUN, SINGLE BLOCK modes.
Bit 6 – Display of pallet table and NC program
0: Both simultaneously in a split screen
1: Pallet table or NC program individually
Bit 7 – AUTOSTART function
0: AUTOSTART by the NC
1: AUTOSTART by the PLC
Bit 8 – Procedure for tool-oriented machining in the **`Program Run`** operating modes
0: NC start machines all workpieces on the pallet until the next tool change
1: NC start executes all NC programs until the end of the pallet
Bit 9 – Editing of pallet tables
0: EDIT PALLET soft key is not displayed
1: EDIT PALLET soft key is displayed

Module 9090 Select a line in the pallet table

With this module, you set the cursor on a particular line of the pallet table that you selected in the program run mode. If the iTNC is in another mode, the selection will be made when the control returns to the **Program Run, Single Block** or **Program Run, Full Sequence** operating mode.

The selection is possible only as long as no pallet file has been started.

Call only in a submit job or spawn job.

Call:

PS	B/W/D/K	<Line number in the pallet table>
CM	9090	
PL	B/W/D	<Error code> 0: No error, line was selected 1: Module was not called in a spawn or submit job 2: Call during running NC program 3: No pallet table selected in full sequence 4: Line does not exist

Module 9281 Select a line in the pallet table

With this module you set the cursor on a particular line of the pallet table that you selected in the **Program Run, Single Block** or **Program Run, Full Sequence** operating mode. Datum shift and datum setting can be run immediately. If the iTNC is in another mode, the selection will be made when the control returns to the **Program Run, Single Block** or **Program Run, Full Sequence** operating mode.

Call:

PS	B/W/D/K	<Line number in the pallet table>
PS	B/W/D/K	<Mode> Bit 0 – 0: Do not run datum shift or set the datum 1: Run the datum shift/set the datum immediately Bit 1 – Shift the datum/set the datum 0: Do not run if the line is disabled by an entry in the LOCK column 1: Run even if the line is disabled by an entry in the LOCK column
CM	9281	
PL	B/W/D	<Error code> 0: No error, line was selected 1: Module was not called in a spawn or submit job 2: Call during running program 3: No pallet table selected in full sequence 4: Line does not exist 5: Error during datum setting, in the datum table or pallet table

NC macro for changing the tool during tool-oriented machining

A special tool-change macro is required for tool-oriented pallet machining. This is defined through the keyword **TCTOOLMODE=** in NCMACRO.SYS.

This specific NC macro is called for tool oriented machining instead of the standard tool-change macro.

The macro must perform the following functions:

- Positioning to clearance height
- Execution of M146
- Tool change through **TOOL CALL**. The standard tool-change macro is called.

With **FN18: SYSREAD Qxxx = ID510 NR5** or **NR6 IDX<Axis>** you can find whether a clearance height has been programmed for an axis, and if so, the value specified for the clearance height in the NC macro.

With the M function M146 the current geometry information is saved in a temporary file. This information is required for continuing NC program run after an interruption due to a **TOOL CALL** during tool-oriented machining. In addition, a code is entered in the **CTID** column and the entry in **W-STATE** is changed to **INCOMPLETE**, if required.

A simple example of an NC macro for tool changing with tool-oriented machining:

```
0       BEGIN PGM TOchange MM
1       L Z–32 R0 FMAX M91
2       FN 18: SYSREAD Q1 = ID60 NR1
3       FN 18: SYSREAD Q2 = ID60 NR2
4       FN 18: SYSREAD Q3 = ID60 NR3
5       FN 18: SYSREAD Q4 = ID60 NR4
6       FN 18: SYSREAD Q5 = ID60 NR5
7       FN 18: SYSREAD Q7 = ID60 NR7
7       FN 18: SYSREAD Q8 = ID60 NR8
8       M146
9       TOOL CALL Q1 Z SQ3 DL+Q4 DR+Q5 DR2:+Q7
10      END PGM TOchange MM
```

NC macro for changing pallets and fixtures

- In NCMACRO.SYS, use the entry **PALETT=** to define the complete path and name of the NC macro that is to be called when a pallet entry **(PAL)** is run.
- In NCMACRO.SYS, use the entry **CLAMP=** to define the complete path and name of the NC macro that is to be called when a fixture entry **(FIX)** is run.

In these macros you can request the current line or pallet name with **FN18: SYSREAD Qxxx = ID510 NR1** or **NR2,** respectively.

This NC macro also can be started from the PLC with Module 9280.
To synchronize the current machine status and the look-ahead calculation with an NC macro call, see "NCMACRO.SYS" on page 1644.

A simple example of an NC macro for changing fixtures:

```
0       BEGIN PGM TOclamp MM
1       FN 18: SYSREAD Q1 = ID510 NR5 IDX1
2       FN 18: SYSREAD Q2 = ID510 NR5 IDX2
3       FN 18: SYSREAD Q3 = ID510 NR5 IDX3
4       FN 18: SYSREAD Q4 = ID510 NR5 IDX4
5       FN 18: SYSREAD Q5 = ID510 NR5 IDX5
6       FN 9: IF +Q3 EQU +0 GOTO LBL 1
7       FN 18: SYSREAD Q13 = ID510 NR6 IDX3
8       L Z+Q13 R0 FMAX
9       LBL 1
10      FN 9: IF +Q1 EQU +0 GOTO LBL 2
11      FN 18: SYSREAD Q11 = ID510 NR6 IDX1
12      L X+Q11 R0 FMAX
13      LBL 2
14      FN 9: IF +Q2 EQU +0 GOTO LBL 3
15      FN 18: SYSREAD Q12 = ID510 NR6 IDX2
16      L Y+Q12 R0 FMAX
17      LBL 3
18      FN 9: IF +Q4 EQU +0 GOTO LBL 4
19      FN 18: SYSREAD Q14 = ID510 NR6 IDX4
20      L C+Q14 R0 FMAX
21      LBL 4
22      FN 9: IF +Q5 EQU +0 GOTO LBL 5
23      FN 18: SYSREAD Q15 = ID510 NR6 IDX5
24      L B+Q15 R0 FMAX
25      LBL 5
26      END PGM TOclamp MM
```

Module 9280 Start the NC macro (Run pallet entry)
The NC macro must be defined in NCMACRO.SYS with the entry **`PALETT =`.**
It can only be activated if the control is in the **`Program Run, Single Block`** or **`Program Run, Full Sequence`** operating mode, a pallet table is selected, and no macro or NC program is running.

Call:
PS B/W/D/K <Pallet number>
PS B/W/D/K <Line number>
CM 9280

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	7	The file entered in the entry PALETT= does not exist.
	8	Control is not in the SINGLE BLOCK or FULL SEQUENCE mode
	20	Module was not called in a spawn job or submit job
	28	An NC program or NC macro is running.
	29	Selected file is invalid or does not exist
	30	There is no PALETT= entry in the NCMACRO.SYS file
	36	NCMACRO.SYS does not exist

NC macro at the end of an NC program

- In NCMACRO.SYS, use the entry **`PALEPILOG=`** to define the complete path and name of the NC macro that is to be called at the end of an NC program that was started from the pallet table.

To synchronize the current machine status and the look-ahead calculation with an NC macro call, see "NCMACRO.SYS" on page 1644.

8.8 Electronic Handwheel

The following handwheels can be connected with HEIDENHAIN controls:

- One panel-mounted HR 130 handwheel, or
- Three HR 150 panel-mounted handwheels via the HRA 110 handwheel adapter, or
- One HR 410 portable handwheel, or
- One HR 420 portable handwheel with display

For information on the operation of the electronic handwheel, see the User's Manual.

- In MP7640, enter the type of handwheel connected to the control. If you enter a value greater than zero and no handwheel is connected, the error message **HANDWHEEL?** appears.
- If you use more than one handwheel together with the HRA 110 handwheel adapter, enter in MP7650 for each axis the counting direction of the individual handwheels. If you use only one handwheel, enter the counting direction in bit 0.

Shock or vibrations can cause a slight motion at the handwheel and produce an unintentional axis movement. In this case:

- Enter a threshold sensitivity in MP7660.

With W1062 you can disable the handwheel pulses for specific axes, if more than one handwheel is used in connection with the HRA 110 handwheel adapter. If the marker M4576 is set, all axes are disabled. If it is reset, W1062 applies. If only one handwheel is used, a selectable axis can be disabled with W1062.

The subdivision factor specifies the traverse per handwheel revolution:

- For the HR 130 or HRA 110, choose the **HANDWHEEL** operating mode and enter a subdivision factor for each handwheel. To ensure that the rapid traverse rates specified in MP1010.x are not exceeded, the smallest possible input step is preset by the control.
- For the HR 130 or HR 410, with MP7641 you specify whether the subdivision factor is entered directly through the TNC keyboard or via PLC Module 9036.

Subdivision factor	Traverse distance per revolution [mm]	Effective beginning from rapid traverse: MP1010.x [mm/min]
0	20	12 000
1	10	6 000
2	5	3 000
3	2.5	1 500
4	1.25	750
5	0.625	80
6	0.312	80
7	0.156	80
8	0.078	80
9	0.039	80
10	0.019	80

MP7674.x can always be used to set a minimum axis-specific subdivision factor for the HR 130, HR 330, and HR 332. It can be used for the HR 410 if MP7641 bit 1 is not set (without detent encoder).

You can choose a larger input step for the traverse distance per rotation than that calculated by the NC:

- In MP7670.x, enter a subdivision factor.
- In MP7645.x, enter an initialization parameter for the handwheel.

The two machine parameters are evaluated by the HRA 110 or HR 410, but not by the HR 420.

MP7675.x has no meaning for the HR 130, HR 330, or HR 332. If MP7641 bit 1 is set (with detent encoder) for the HR 410 or HR 420 , then a maximum distance in [mm] per detent can be entered for each axis. If MP7641 bit 1 is not set (without detent encoder) for the HR 420 , then a maximum distance in [mm] per handwheel revolution can be entered for each axis.
These limitations serve to prevent the movement monitoring from becoming active because of abrupt motions when a large gear ratio is in effect on handwheels for axes with poor dynamics.

The VT axis (Virtual Tool Axis) can now be activated for handwheel superimposition in the active tool-axis direction by setting bit 4 of MP7641.

If bit 5 of MP7641 is set, the keys of the HR 420 (PLC markers 4661 to 4668) are reported to the PLC even if the HR 420 is not active. This enables the PLC to make the keys of the HR 420 (e.g. NC stop, spindle stop) become effective even if the handwheel is not selected.

Bit 6 of MP7641 is used to enable traversing of auxiliary axes with the HR 420. This makes it possible to select axes that are indicated with lowercase letters in MP100 in the operator menu of the HR 420 handwheel, and to move them with the handwheel.

The vertical soft keys (PLC soft keys) could continue to be used on the machine side when the HR 420 handwheel was active. As of 340 49x-05 these soft keys can only be used if bit 9 is set in MP7641.

Danger

MP7641 bit#9 may only be set if the vertical soft keys (PLC soft keys) do not have any functions that could pose danger to the operator (e.g. starting an axis movement or a spindle movement).

MP7640 **Handwheel**

Input:
0: No handwheel
1: Reserved
2: HR 130
3: Reserved
4: Reserved
5: Up to three HR 150 via HRA 110
6: HR 410
7 to 10: Reserved
11: HR420/HR520 without LED activation
12: In future for the HR 550 wireless handwheel
13: HR520 with LED activation

MP7641 **Handwheel settings**

Input:
Bit 0 – HR 410: Entry of subdivision factor
0: Through iTNC keyboard
1: Through PLC Module 9036
Bit 4 – Handwheel superimposition in the active tool-axis direction
0: Behavior as before
1: VT axis can be selected
Bit 5 – Inactive behavior of HR 420
0: Report the keys of the HR 420 to the PLC only when the HR is active
1: Report the keys of the HR 420 to the PLC even if the HR is not active
Bit 6 – Selecting and traversing auxiliary axes with HR 420
0: Traversing auxiliary axes not possible
1: Traversing auxiliary axes is possible
Bit 7 – Teach-In button on HR 550
0: By the NC
1: By PLC
Bit 8 – CTRL button on HR 550
0: By the NC
1: By the PLC
Bit 9 – Vertical soft keys when the handwheel is active
0: PLC soft keys are not active when HR is active
1: PLC soft keys are active when HR is active

MP7650 **Handwheel counting direction (for HRA 110: for each axis)**

Input:
Bit 0:
0: Negative counting direction
1: Positive counting direction
Only for HRA 110: Bits 0 to 13 represent axes 1 to 14
0: Negative counting direction
1: Positive counting direction

MP7660	**Sensitivity for electronic handwheel**
Input:	0 to 65 535 [increments]

MP7670	**Subdivision factor for handwheel**
Input:	0 to 10
MP7670.0	Subdivision factor for slow speed
MP7670.1	Subdivision factor for medium speed (only HR 410)
MP7670.2	Subdivision factor for fast speed (only HR 410)

MP7674.x	**Axis-specific subdivision factor for handwheel**
Input:	0: No limitation 1 to 10

MP7675.x	**Handwheel, axis-specific maximum path**
Input:	0: No limitation 0.0001 to 10.0000 [mm]

		Set	**Reset**
M4576	**Locking the handwheel**	PLC	PLC
W1062	**Lock the handwheel for specific axes**	PLC	PLC

Module 9036 Writing status information
Prerequisite: MP7641 = 1

The information to be overwritten is designated with a transferred number.

Handwheel subdivision factors are limited to a smallest possible value, depending on the rapid traverse rate of the respective axis.
CAUTION: No error message!

Number	Function	Value
0	Handwheel subdivision key X	0 to 10
1	Handwheel subdivision key Y	0 to 10
2	Handwheel subdivision key Z	0 to 10
3	Handwheel subdivision key IV (MP410.3)	0 to 10
4	Handwheel subdivision key V (MP410.4)	0 to 10
5	Handwheel subdivision of all axes	0 to 10
6	Select the handwheel axis (not for HRA 110)	0 to 8 Axes 1 to 9
7	Handwheel speed range ■ 0 = Slow speed ■ 1 = Medium speed ■ 2 = Rapid speed	0 to 3
10	See "Incremental Jog Positioning"	
11	Handwheel subdivision of axis 1	0 to 10
12	Handwheel subdivision of axis 2	0 to 10
13	Handwheel subdivision of axis 3	0 to 10
14	Handwheel subdivision of axis 4	0 to 10
15	Handwheel subdivision of axis 5	0 to 10
16	Handwheel subdivision of axis 6	0 to 10
17	Handwheel subdivision of axis 7	0 to 10
18	Handwheel subdivision of axis 8	0 to 10
19	Handwheel subdivision of axis 9	0 to 10

Call:

PS	B/W/D/K	<Number of the status information>
PS	B/W/D/K	<Value to be written>
CM	9036	
PL	B/W/D	<Error code> 0: Status written 1: Incorrect status code 2: Transferred value out of range 3: Input disabled

Error recognition:

Marker	Value	Meaning
M4203	0	Status information was written
	1	Error code in W1022
W1022	1	Transferred value out of range
	2	Incorrect number of the status information
	6	Input disabled

8.8.1 HR 130 panel-mounted handwheel

- Enter MP7640 = 2 (HR 130).

When the axis keys are pressed, the associated cursor and the handwheel symbol are displayed simultaneously.

8.8.2 HR 410 portable handwheel

- Enter MP7640 = 6 (HR 410).
- In MP7645, specify whether the keys on the handwheel are to be evaluated by the NC or PLC.

Evaluation of the keys by NC: MP7645.0 = 0

X		IV
Y		V
Z		
Slow feed rate	Medium feed rate	Fast feed rate
–		+
O109 I173	O110 I174	O111 I175

With the exception of the function keys A, B and C, all keys are evaluated by the NC.

- With MP7670.x, select the subdivision factors for slow, medium and fast speed.
- With MP7671.x, define the values for slow, medium and fast speed. The speed is entered as a percentage of the manual feed rate (MP1020.x).
- If MP7641 bit 1 is set, then with MP7672.x you can set the distance per handwheel step for the HR 410. Different distances per step can be set for three speed stages.

Evaluation of the keys via the PLC: MP7645.0 = 1

O96 I160		O97 I161
O98 I162		O99 I163
O100 I164		O103 I167
O104 I168	O105 I169	O106 I170
I171		I172
O109 I173	O110 I174	O111 I175

All keys are evaluated by the PLC.

Module 9036 sets the handwheel axis and handwheel subdivision.

With W766 you can influence the feed rate of the axis direction keys.

Window upon activation of the HR 420
When the HR 420 is activated, a pop-up window appears on the iTNC, indicating that the HR 420 has assumed control. This window is a special pop-up window. The text color is defined in MP7366.1, and the window background color in MP7350.

MP7645	**Initializing parameter for handwheel. If an HR 410 is installed, MP7645.0 has the following meaning:**
MP7645.0	Assignment of the handwheel keypad for HR 410
Input:	0: Evaluation of the keys by NC, including LEDs 1: Evaluation of the keys by PLC

MP7645.1–7 Have no function

MP7670 **Subdivision factor for handwheel**
Input: 0 to 10
MP7670.0 Subdivision factor for slow speed
MP7670.1 Subdivision factor for medium speed (only HR 410)
MP7670.2 Subdivision factor for fast speed (only HR 410)

MP7671 **Handwheel feed rate in the Handwheel operating mode with HR 410**
Input: 0 to 1000 [% of MP1020]
MP7671.0 Slow speed
MP7671.1 Medium speed (only HR 410)
MP7671.2 Fast speed (only HR 410)

MP7672 **HR 410, distance per handwheel step**
Input: 0.0000 to 1.0000 [mm]
MP7672.0 Slow speed
MP7672.1 Medium speed
MP7672.2 Fast speed

8.8.3 HR 420 portable handwheel

Settings

- Enter MP7640 = 11 (HR 420)
- In MP7641, specify whether you are using an HR 420 with or without detent, and whether the keys on the handwheel are to be evaluated by the NC or PLC.

Information about MP7641:

- Bit 2
  - Bit 2 = 0: If the HR 420 assumes control, then W1046 (Manual traverse in positive direction), W1048 (Manual traverse in negative direction), W1050 (Incremental jog positioning in positive direction), W1052 (Incremental jog positioning in negative direction) and M4561 (Rapid traverse) have no effect. Only the keys on the HR 420 are valid.
  - Bit 2 = 1: The direction keys and the rapid-traverse key of the HR 420 must be evaluated by the PLC. For safety reasons, once the HR 420 has assumed control, only the keys of the HR 420 (M4660) should be active.
- Bit 3
  - Bit 3 = 0: M4564 (NC start) has no affect once the HR 420 assumes control. Only the NC start from the HR 420 is valid. An NC stop is performed via the corresponding key on the HR420 or via M4560 (NC stop).
  - Bit 3 = 1: The NC start and NC stop keys of the HR 420 must be evaluated by the PLC. For safety reasons, NC start should only be activatable from the HR 420 once it has assumed control (M4660).
- Spindle start and spindle stop must always be conducted by the PLC.

All keys are evaluated by the NC. Certain keys are mapped to markers.

F1	F2	F3	F4	F5
X	Y	Z	IV	V
	↑	Handwheel active/ inactive	↓	
	– (M4667)	Rapid traverse (M4663)	+ (M4666)	
	Spindle Start (M4664)	Actual position capture	NC start (M4661)	
	Spindle Stop (M4665)	Ctrl (M4668)	NC stop (M4662)	

Activation and override potentiometers

If the HR 420 is activated (via the handwheel key on the HR), a small pop-up window appears on the screen of the iTNC and entries via the keys (keyboard and horizontal soft keys) are disabled. However, the override potentiometers of the keyboard remain active.

The activation of the HR 420 can be disabled by PLC. If you set the marker M4680, the message **`HR not allowed`** appears in the handwheel's display. If the marker is set and the user presses the handwheel activation key, the error message **`Wrong operating mode for handwheel`** appears.
If the marker M4680 is set when the handwheel is active, the marker has no effect.

If the override potentiometers of the handwheel are to become active, then this can only be done with the <Ctrl> + <handwheel symbol> key combination on the HR 420.

If this key combination is pressed, then a selection menu appears on the handwheel, in which the potentiometers to be activated must be selected.

- Display: **`Handwheel/ Operating panel`**
- Soft keys on the handwheel:
  - Soft key F1: **`HW`** (potentiometers on the handwheel become active)
  - Soft key F2: **`KBD`** (potentiometers on the operating panel become active)

If the override potentiometers on the handwheel are switched active, then the message **`Handwheel override active`** also appears in the pop-up window on the screen of the iTNC, and the handwheel can only be deactivated if the control of the override potentiometers is returned to the operating panel. If you try to deactivate the HR 420 while it has control over the override, then a window appears requesting that you first deactivate the handwheel’s control over the override.

Danger

Please note that the feed rate or spindle speed set becomes higher or lower depending on the potentiometer setting.

PLC Marker M4670 provides information on whether the override potentiometers of the HR 420 or of the machine operating panel are active.
M4670 – Potentiometer of HR 420 is active
M4670 = 1: Potentiometer of machine operating panel is active

The override factor of the active override potentiometer can also be influenced with M764 or W766.

When marker M4626 is set, the PLC locks all key inputs of the TE keyboard unit, including the soft keys, except for the CE and the screen switch-over key. The lock is only effective for the machine screen (iTNC - machine), the editing screen (iTNC - editing) is not affected by the lock. Furthermore, this disabling does not affect the keys on the MB machine operating panel.

This has the same effect as pressing the handwheel activation key on the HR 420. Setting the new marker M4626 automatically results in setting M4660 (HR 420 assumes control). If a corresponding message window informing you about the current status on the screen is desired, it must be displayed via the PLC program. Control must also be returned to the operating panel via the PLC program.

MP7641 Handwheel settings

Input: Bit 1 – HR 420: With detent positions
0: Without detent positions
1: With detent positions
Bit 2 – HR 420: Axis direction keys and rapid traverse
0: By the NC
1: By the PLC
Bit 3 – HR 420: NC Start / NC Stop
0: By the NC
1: By the PLC

		Set	Reset
M4626	**Disabling of keys on the TE**	PLC	PLC
M4660	**HR 420 assumes control**	NC	NC
M4661	**NC start on HR 420**	NC	NC
M4662	**NC stop on HR 420**	NC	NC
M4663	**Rapid traverse key on HR 420**	NC	NC
M4664	**Spindle start on HR 420**	NC	NC
M4665	**Spindle stop on HR 420**	NC	NC
M4666	**+ key on HR 420**	NC	NC
M4667	**– key on HR 420**	NC	NC
M4668	**CTRL key on HR 420**	NC	NC
M4670	**Potentiometer of HR 420 is active**	NC	NC
M4680	**Disable activation of the HR 420**	PLC	PLC

Free soft-key definition for HR 420

You will find information about the freely definable soft-key menu for the HR 420 under “Soft-key project file for screen" on page 1320.

8.8.4 HR 520 portable handwheel

With the HR 520, HEIDENHAIN is introducing the first handwheel of the new generation HR 5xx. Unlike its predecessor model HR 420, the new HR 520 features a 6th axis key, which can be configured, for example, for selecting the "virtual tool axis" (option 44). The HR 520 also offers 6 LEDs, which can be controlled by PLC markers as of NC software 34049x-06 (iTNC 530). For example, these LEDs can be used for optical confirmation of menu-driven switching processes on the machine. In addition, it is possible to scan all handwheel keys from the PLC and use them accordingly in the PLC program.

The handwheel can be fastened to the machine using the magnetic holding pads on the rear of the handwheel or an optionally available mount (ID 591 065-xx).

The HR 520 handwheel is available in versions with or without detent.

The HR 520 is connected with the same cables and adapters as the HR 420.

Power consumption of HR 520: 1 W

Settings

- Set MP7640 to 11 or 13, depending on whether you want to use the LEDs of the HR 520.
- In MP7641, specify whether you are using an HR 5x0 with or without detent, and whether the keys on the handwheel are to be evaluated by the NC or PLC.

All settings that can be defined in the machine parameters of the HR 420 can also be used for the HR 520. The PLC markers of the HR 420 also match those of the HR 520.

The six LEDs of the HR 5x0 can be controlled by the specified PLC markers.

F1	**F2**	**F3**	**F4**	**F5**
	X	**Y**	**Z**	
	IV	**V**	**VI**	
	↑	Handwhee l active/ inactive	↓	
	– (M4667)	Rapid traverse (M4663)	+ (M4666)	
	Spindle Start (M4664) LED (M4684)	Actual position capture LED (M4689)	NC start (M4661) LED on (M4681)	
	Spindle Stop (M4665) LED (M4685)	Ctrl (M4668) LED (M4688)	NC stop (M4662) LED (M4682)	

8.8.5 HR 150 panel-mounted handwheels with HRA 110 handwheel adapter

▶ Enter MP7640 = 5 (HR 150 via HRA 110)

You can use the step switch S1 to choose the subdivision factor (see "HRA 110 handwheel adapter" on page 408). For this purpose you must evaluate the inputs I160 to I167 of the switch in the PLC and activate the corresponding interpolation factor with Module 9036.

Axes X and Y are permanently assigned to the handwheel inputs X1 and X2. You can assign the third handwheel (input X3) to any other axis. All handwheel axes are indicated by the handwheel symbol.

▶ Take the designation for axes IV and V from MP410.x.

▶ In MP7645.2, specify how the axis for the third handwheel is selected.

- Selection by axis selection switch (switch S2, see MP7645.0)
- Selection permanently defined in MP7645.1

MP7645	**Initializing parameter for handwheel**	
MP7645.0	Assignment of a third handwheel via axis selector switch S2, when MP7645.2 = 0	
Input:	0:	Switch position 1 (at the left stop)3rd handwheel axis Z Position 23rd handwheel axis IV Position 33rd handwheel axis V
	1:	Switch position 1 (at the left stop)3rd handwheel axis X Switch position 23rd handwheel axis Y Position 33rd handwheel axis Z Position 43rd handwheel axis IV Position 53rd handwheel axis V
	2:	Position 33rd handwheel axis Z Position 43rd handwheel axis IV Position 53rd handwheel axis V
MP7645.1	Fixed assignment of a third handwheel if MP7645.2 = 1	
Input:	1: Axis X 2: Y axis 4: Z axis 8: Axis IV (MP410.3) 16: Axis V (MP410.4)	
MP7645.2	Assignment of a third handwheel by axis selection switch or MP7645.1	
Input:	0: Assignment by axis selection switch according to MP7645.0 1: Assignment by MP7645.1	
MP7645.3–7	Have no function	

Assignment of switch positions to PLC inputs

The tables below list the assignments of switch positions of S1 and S2 to the PLC inputs I160 to I175.

The two switches work with a 0 V logic circuit.

Example: If switch S1 is in position 3, input I162 is logically 0, and the inputs I160, I161, I163 to I167 are logically 1.

Step switch 1: Step switch for choosing the subdivision factor

Switch position	PLC input
1 (at the left stop)	I160
2	I161
3	I162
4	I163
5	I164
6	I165
7	I166
8 (at the right stop)	I167

Step switch 2: Axis selection switch

Switch position	PLC input
1 (at the left stop)	I168
2	I169
3	I170
4	I171
5	I172
6	I173
7	I174
8 (at the right stop)	I175

8.9 PLC Inputs/Outputs

The MC 42x(B,C) provides you with switching and analog inputs/outputs for the PLC. If the available number of I/O is not enough, you can add up to four PL 4xxB or PL 510 input/output units.

	MC 42x(B,C)			
	X9 (only MC422(B))	**X41**	**X42**	**X48**
Switching inputs 24 V–	–	–	56	–
Switching outputs 24 V–	–	31	–	–
Analog inputs 10 V–	–	–	–	3
Inputs for Pt 100 thermistors	–	–	–	3
Analog outputs 10 V–	12[a])	–	–	–
Control-is-ready output	–	2	–	–
Control-is-ready input	–	–	2	–

a. You need one analog output for each analog axis.

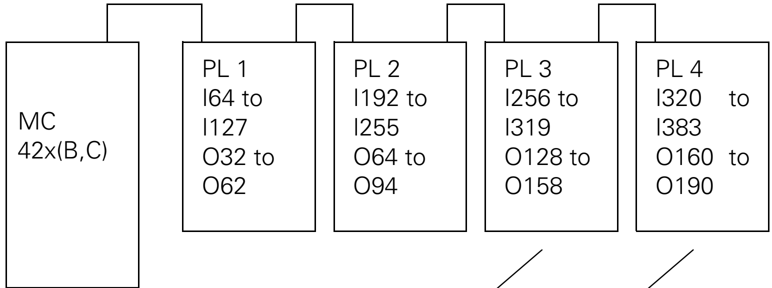
	PL 4xxB PLC input/output unit		**PL 510 PLC input/output unit**	
	PL 410 B	**PL 405 B**	**PLD 16-8**	**PLA 4-4**
Switching inputs 24 V–	64	32	16	–
Switching outputs 24 V–	31	15	8	–
Analog inputs 10 V–	(4)	–	–	4
Inputs for Pt 100 thermistors	(4)	–	–	4
Analog outputs 10 V–	–	–	–	–
Control-is-ready output	1	1	(1)	–
Control-is-ready input	–	–	–	–

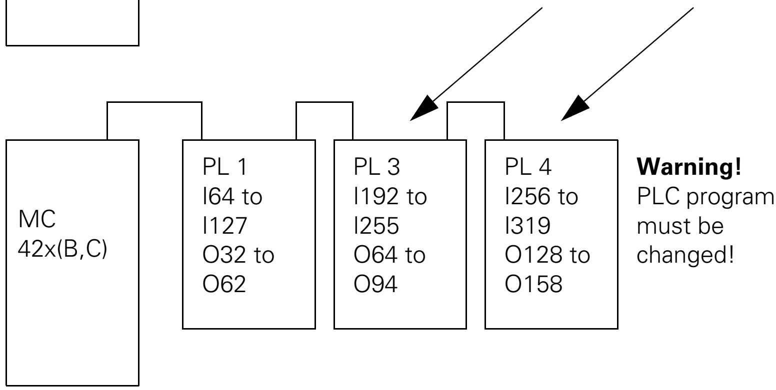
To interrogate and set the inputs and outputs of the PLC I/O units you need PLC modules.

PL assignment

Up to four PL PLC I/O units can be connected. The first PL is connected to the MC 42x(B,C), the second PL to the first PL, the third to the second, etc. The PLs are permanently assigned to specific inputs and outputs. If a PL is dropped, the assignment of the inputs and outputs to the PLs also changes. To avoid having to change your PLC, you can assign a logical PL (no. of the PL according to the assignment of I/O in the PLC program) to the physical PL (no. of the PL as seen by the MC 42x(B,C)).

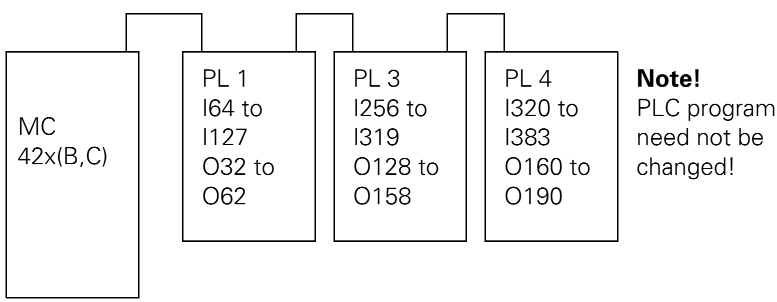
Example:





or

MP4030.0 = 0 : 1st physical PL = 1st logical PL
MP4030.1 = 2 : 2nd physical PL = 3rd logical PL
MP4030.2 = 3 : 3rd physical PL = 4th logical PL



In MP4031 you can enter the number of physically connected PLs in order to monitor the number of connected PLs (PL 4xxB, PL B510). When the control is started and every time the PLC is restarted, the entry in MP4031 is compared to the actual number of physical PLs. If the entry in MP4031 is not identical to the number of connected PLs, the PLC is not started and the error message **`PL: Configuration incorrect`** appears.

MP4030	**Assignment of physical to logical PL**
Input:	0: First logical PL 1: Second logical PL 2: Third logical PL 3: Fourth logical PL
MP4030.0	First physical PL
MP4030.1	Second physical PL
MP4030.2	Third physical PL
MP4030.3	Fourth physical PL

MP4031	**Monitoring of number of PLs**
Input:	–1: Previous behavior; monitoring not active 0 to 4: Number of PLs being monitored

Diagnosis of the PL

Module 9007 Diagnostic information of the PL 4xxB

Module 9007 can ascertain diagnostic information of the PL 4xxB. To save computing time, refrain from repeatedly calling this module.

Call:

PS	B/W/D/K	<Number of the logical PL 4xxB>
PS	B/W/D/K	<Number of the information> 0: Readiness 1: Supply voltage Bit 0: Logic voltages 24 V and 5 V Bits 1 to 4: 24 V for X11 to X14 Bit 5: 24 V for analog inputs 2: Analog inputs used 3: Total number of PLs on this MC 42x(B,C) 4: Further PLs on this PL 5: PL is a PL 410B
CM	9007	
PL	B/W/D/K	<Diagnostic information> 0: Not available 1: Available 0 to 4: Number of PLs (PL 4xxB and PL 510)

Error recognition:

Marker	Value	Meaning
M4203	0	Diagnostic information was read
	1	Error code in W1022
W1022	1	Invalid code
	2	Invalid PL
	24	Module was called in a spawn or submit job
	51	Function not possible or not a PL 4xxB

Module 9137 Diagnostic information of the PL 510

Module 9007 can ascertain diagnostic information of the PL 4xxB. To save computing time, refrain from repeatedly calling this module.

Call:

PS	B/W/D/K	<Number of the PLB 510 basic module (0 to 3)>
PS	B/W/D/K	<Number of the slot (0 to 3)>
PS	B/W/D/K	<Number of the information> 0: Possible mode of operation 1: Active mode of operation 2: Basic module model 3: Reserved 4: Reserved 5: Status of the basic module 6: Module model in the slot 7: Reserved 8: Reserved 9: Status of the module in the slot 10: Logical status of the outputs of a PLD 16-8 11: Short-circuit of the outputs of a PLD 16-8 12: Number of connected PL 510
CM	9137	
PL	W/D	<Diagnosis information> Information no. 0: 0: "PL 510" operating mode not possible (only "PL 4xxB" operating mode, without new functions of the PL 510) 1: "PL 510" operating mode possible Information no. 1: 0: "PL 4xxB" operating mode active (without new functions of the PL 510) 1: "PL 510" operating mode active Information no. 2: 0: No PLB 510 1: PLB 510 Information no. 5: Bit 0 = 1: Power supply of the PLB 510 is OK Bits 1 to 15: Reserved Information no. 6: 0: No module in slot 1: Reserved 2: PLD 16-8 in slot 3: PLA 4-4 in slot Information no. 9: PLD 16-8: Bit 0 = 1: Power supply outputs 0 to 3 are OK Bit 1 = 1: Power supply outputs 4 to 7 are OK Bit 2 = 1: Short circuit at an output Bit 3 = 1: At least one output idle (< 300 mA) Bits 4 to 6: No meaning Bit 7 = 1: Output 7 is a programmable output (otherwise "control is ready") Bits 8 to 31: No meaning PLA 4-4: Bit 0 = 1: Power supply of the inputs is OK Bits 1 to 31: No meaning

Information no. 10:
Bit 0: Status of output 0 (PLD 16-8)
to
Bit 7: Status of output 7 (PLD 16-8)
Information no. 11:
Bit 0: Short circuit at output 0 (PLD 16-8)
to
Bit 7: Short circuit at output 7 (PLD 16-8)
Bit 8: No load (< 300 mA) at output 0 (PLD 16-8)
to
Bit 15: No load (< 300 mA) at output 7 (PLD 16-8)

Error recognition:

Marker	Value	Meaning
M4203	0	Diagnostic information was read
	1	Error code in W1022
W1022	1	Invalid code
	2	Invalid basic module number or slot number
	24	Module was called in a spawn or submit job
	51	Function not possible or not a PL 510

Module 9139 Reset short-circuit monitoring of the outputs on the PLD 16-8

The short circuit of an output of the PLD 16-8 is indicated by an LED, and the output is reset. Short-circuit monitoring remains in place, and must therefore be reset with Module 9139.
To save computing time, refrain from continuously calling this module.

Call:
PS B/W/D/K <Function>
0: Reserved
1: Reserved
2: Reset short-circuit monitoring
CM 9139

Error recognition:

Marker	Value	Meaning
M4203	0	Short-circuit monitoring was reset
	1	Error code in W1022
W1022	1	Invalid function
	24	Module was called in a spawn or submit job
	51	Function not possible or no PL 510

8.9.1 24 V– switching input/outputs

In PLC addresses you can find the current states of the switching inputs and outputs.

For the current states of the inputs/outputs of the PLC:

- Read all inputs with Module 9002,
- or only certain inputs with Module 9008.
- Update all outputs with Module 9005,
- or only certain outputs with Module 9009.

With Module 9004 you can evaluate the rising or falling edge of the PLC inputs.

> **Note**
>
> Before the PLC program is converted, the PLC outputs are reset. In addition, the memory of the PLC outputs is reset.
> During a loss of power (power fail), the control tries to reset the PLC outputs.

If all PLC outputs are switched off (e.g., during PLC program compilation or due to a PLC run-time error), the outputs that can not be switched off by an emergency stop can be switched off delayed by 250 ms.

If all PLC outputs are switched off (e.g., during PLC program compilation or due to a PLC run-time error), the outputs defined in MP4060.x can be switched off delayed by the time defined in MP4061.x.
The delay only affects outputs that cannot be switched off by emergency stop, since for the outputs that can be shut off by an emergency stop, the 24-V supply is shut off immediately. In X44 you define which outputs are to be switched off via an emergency stop (see "X44: PLC supply voltage" on page 366).

With MP4043 you can delay the shutdown of the control (after pressing the SHUTDOWN soft key) to enable the PLC to execute final actions.
When you press the SHUTDOWN soft key, the control starts shutting down, PLC marker M4179 is set and the delay time entered in MP4043 starts to elapse. The PLC program can execute final actions during the delay time. If the PLC marker M4179 is reset by the PLC program, the SHUTDOWN process will be continued before the time set in MP4043 has elapsed.
The SHUTDOWN process is automatically continued at the latest after the expiration of the time in MP4043.
If the following conditions apply, however, it cannot be guaranteed that final PLC actions (e.g. data backup of non-volatile memory ranges of the PLC) are executed:

- If no PLC program is active at the time the control is shut down, (e.g. PLC run time error has occurred, power interruption status), the shutdown time is not delayed.
- If a PLC run time error occurs during shutdown, the delay is canceled.

MP4060.0-3 Outputs that are to be switched off with the delay from MP4061.x when all outputs are switched off

Input: 0 to 30
-1: Do not switch off output with delay

MP4061.0-3 Delay time for switching off the outputs in MP4060.x

Input: 0 to 5.000 [s]

MP4043 Delay during shutdown for the PLC to execute final actions

Input: 1 to 60 [s]
0 = No delay

Up until NC software 340 422-08 and 340 480-08, the switch-off delay is defined in the following machine parameters:

MP4044 Switch off outputs that cannot be switched off by emergency stop after 250-ms delay

Input: %xxxxxxxx
Bits 0 to 7 correspond to O16 to O23
0: Do not switch off output with delay
1: Switch off output with delay

MP4045 Switch off outputs that cannot be switched off by emergency stop after 250-ms delay

Input: %xxxxxxx
Bits 0 to 6 correspond to O24 to O30
0: Do not switch off output with delay
1: Switch off output with delay

Module 9002 Reading all inputs of a PLC input/output unit

In PLC addresses you can read the current states of the PLC input/output unit. The memory contents remain unchanged until you call this module or Module 9008. The module does not recognize whether a PLC input/output unit is actually connected.

For the PL 510, inputs of empty slots are not read.

The program can be called only in the cyclic PLC program.

Call:
PS B/W/D/K <Number of the PL>
0: First PLC input/output unit
1: Second PLC input/output unit
2: Third PLC input/output unit
3: Fourth PLC input/output unit
CM 9002

Error recognition:

Marker	Value	Meaning
M4203	0	Inputs were read
	1	Error code in W1022
W1022	2	Invalid PL number
	24	Module was called in a spawn or submit job

Module 9008 Read certain inputs of a PL 4xxB

In PLC addresses you can read the current states of the PL 4xxB.
The memory contents remain unchanged until you call this module or Module 9002. The module recognizes whether a PLC input/output unit is actually connected.

The program can be called only in the cyclic PLC program.

Call:

PS	B/W/D/K	<Number of the PL> 0: First PL 4xxB 1: Second PLC input/output unit 2: Third PLC input/output unit 3: Fourth PLC input/output unit
PS	D/K	<Bit 0 to 31 = outputs 0 to 31>
PS	D/K	<Bits 0 to 31 = inputs 32 to 63>
CM	9008	

Error recognition:

Marker	Value	Meaning
M4203	0	Inputs were read
	1	Error code in W1022
W1022	2	Invalid PL number or PL not connected
	24	Module was called in a spawn or submit job
	51	Function not possible or no PL 510

Module 9005 Update all outputs of a PLC input/output unit

Module 9005 overwrites the outputs of the PLC input/output unit with the values from the PLC addresses. The outputs are set or reset immediately at the time of module execution and remain in their state until they are set or reset again by this module or Module 9009. The module does not recognize whether a PLC input/output unit is actually connected.

For the PL 510, the outputs of empty slots are not overwritten with values from the PLC addresses.

The program can be called only in the cyclic PLC program.

Call:

PS	B/W/D/K	<Number of the PL> 0: First PLC input/output unit 1: Second PLC input/output unit 2: Third PLC input/output unit 3: Fourth PLC input/output unit
CM	9005	

Error recognition:

Marker	Value	Meaning
M4203	0	Outputs were set
	1	Error code in W1022
W1022	2	Invalid PL number
	24	Module was called in a spawn or submit job

Module 9009 Update certain outputs of a PL 4xxB

Module 9009 overwrites certain outputs of the PL 4xxB with the values from the PLC addresses. The outputs are set or reset immediately at the time of module execution and remain in their state until they are set or reset again by this module or Module 9005. The module recognizes whether a PL 4xxB is actually connected.

The program can be called only in the cyclic PLC program.

Call:

PS	B/W/D/K	<Number of the PL>
		0: First PLC input/output unit
		1: Second PLC input/output unit
		2: Third PLC input/output unit
		3: Fourth PLC input/output unit
PS	D/K	<Bit 0...31 = outputs 0...31>
CM	9009	

Error recognition:

Marker	Value	Meaning
M4203	0	Outputs were set
	1	Error code in W1022
W1022	2	Invalid PL number or PL not connected
	24	Module was called in a spawn or submit job
	51	Function not possible or no PL 510

Module 9004 Edges of PLC inputs

With this module you set, upon falling or rising edges of the PLC inputs, specified end markers or bits in the specified byte range. Changes in the inputs are recognized only if a change also occurs in the PLC addresses (see Module 9002).

Ensure that the specified edge markers or edge bytes are in an unoccupied area. The edge bytes are written beginning with the least significant bit. Superfluous bits are erased.

Call:

PS	B/W/D/K	<Number of the first PLC input>
PS	B/W/D/K	<Number of the first edge marker or edge byte>
PS	B/W/D/K	<Number of PLC inputs>
PS	B/W/D/K	<Edge evaluation>
		0: Rising edge. Entry in edge marker
		1: Falling edge. Entry in edge marker
		2: Rising edge. Entry in edge byte
		3: Falling edge. Entry in edge byte
CM	9004	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Invalid transfer parameter

8.9.2 Analog inputs

Socket X48 of the MC 42x(B,C) provides ±10-V– analog inputs and analog inputs for Pt 100 temperature resistors (see "Analog Input" on page 387).

There is also a version of the PL 4xxB with additional analog inputs, and the PL 510 can be fitted with PLA 4-4 analog modules (see "Overview of Components" on page 250).

Only voltages of up to ±10 V– are permissible at the analog inputs.

The temperatures measured by the Pt 100 thermistors are saved in the PLC words W486 to W490, and the values of the analog inputs are saved in the PLC words W480 to W484.

Read the current states of the inputs with Module 9003.

		Set	Reset
W480-484	**Analog input at X48 [0.1 V]** For inputs 1 to 3	NC	NC
W486 - 490	**Temperature input at X48 [0.5 °C]** For inputs 1 to 3	NC	NC

Module 9003 Read the analog input of the MC and of the PL 4xxB

Module 9003 reads the current value of the specified analog input, regardless of whether it is actually connected.

Value range of ±10 V– input:
–10 to +10, at a resolution of 10 mV
–100 to +100, at a resolution of 100 mV

Value range at Pt 100 input:
0 to 200, at a resolution of 0.5 °C
0 to 1000, at a resolution of 0.1 °C

The module can be called only in the cyclic PLC program.

Call:

PS	B/W/D/K	<Number of analog input>
		0 to 3: ±10 V inputs X15 to X18 on first PL 4xxB
		4 to 7: Pt100 inputs X19 to X22 on first PL 4xxB
		8 to 11: ±10 V inputs X15 to X18 on second PL 4xxB
		12 to 15: Pt100 inputs X19 to X22 on second PL 4xxB
		16 to 19: ±10 V inputs X15 to X18 on third PL 4xxB
		20 to 23: Pt100 inputs X19 to X22 on third PL 4xxB
		24 to 27: ±10 V inputs X15 to X18 on fourth PL 4xxB
		28 to 31: Pt100 inputs X19 to X22 on fourth PL 4xxB
		32 to 63: Reserved
		64 to 66: ±10 V input on connection X48
		67 to 69: Pt 100 input on connection X48
CM	9003	
PL	W/D	<Analog value>
		Nr. 0 to 31: Natural number with the unit 0.1 V or 0.5 °C
		No. 64 to 69: Natural number with the unit 0.01 V or 0.1 °C

Error recognition:

Marker	Value	Meaning
M4203	0	Input was read
	1	Error code in W1022
W1022	2	Invalid PL number or invalid analog input number
	24	Module was called in a spawn job or submit job
	51	Function not possible or no PL 510

Module 9138 Read analog input of the PL 510

Module 9138 reads the current value of the given analog input of the PL 510.

Value range of ±10 V– input: –1000 to +1000, at a resolution of 10 mV
Value range of Pt 100 input:0 to 10000, at a resolution of 0.01 °C

To save computing time, refrain from repeatedly calling this module. The module can be called only in the cyclic PLC program.

Call:

```
PS    B/W/D/K   <Number of the PLB 510 basic module (0 to 3)>
PS    B/W/D/K   <Number of the slot (0 to 3)>
PS    B/W/D/K   <Number of the analog input (0 to 7)>
CM    9138
PL    B/W       <Analog value>
                Analog inputs 0 to 3: Natural number in steps of 0.01 V
                Analog inputs 4 to 7: Natural number in steps of 0.01 °C
```

Error recognition:

Marker	Value	Meaning
M4203	0	Input was read
	1	Error code in W1022
W1022	2	Invalid basic module number or slot number
	24	Module was called in a spawn job or submit job
	51	Function not possible or not a PL 510 or PLA 4-4 analog module

In the standard setting, the values of the Pt 100 inputs are taken over with a change rate of 1 K/s. The disadvantage here is that for large changes in temperature it can take a long time until the correct temperature reading is attained. For example, it would take 30 seconds to correctly read a temperature change of 30 K. An advantage of this, however, is a low sensitivity to disturbance: the temperature display will not jump back and forth between two values:

- If you wish to work with a change rate of 1 K/s, set MP4020 bit 7 = 0.
- If you wish to accept the values of the Pt 100 inputs immediately, set MP4020 bit 7 = 1.

MP4020 PLC compatibility

Format: %xxxxxxxx
Input: Bit 7: Transferring the values of the Pt 100 inputs
0: Accept values at a change rate of 1 K/s
1: Accept results immediately

8.9.3 Analog outputs

You can drive analog outputs 1 to 12 at connections X8 and X9.

Note

Every analog axis or analog spindle needs an analog output. These outputs are no longer available to the PLC.

Module 9130 Output of an analog voltage
With this module you place an analog voltage on an analog output. The voltage is output with a slight delay after the end of the PLC scan.

Call the module only once for each output per PLC scan!

Format: 1 mV
Voltages greater than +10 V or less than –10 V are limited to the respective maximum value.

Call:

PS	B/W/D/K	<Number of the analog output>
		1 to 6: Analog outputs 1 to 6 (X8)
		7 to 13: Analog outputs 7 to 13 (X9)
PS	B/W/D/K	<Analog voltage in mV>
CM	9130	

Error recognition:

Marker	Value	Meaning
M4203	0	Analog voltage was output
	1	Error code in W1022
W1022	1	Invalid analog output
	2	Disabled analog output

8.10 Incremental Jog Positioning

- The "incremental jog positioning" function is switched on and off with the **INCREMENT OFF/ON** soft key.
- To position with incremental jog, press the direction keys (W1046/W1048).

You can interrogate the current state with Marker M4579.

You can limit the jog increment with Module 9036.

You can ascertain the current jog increment with Module 9035.

You can switch the incremental jog function on and off through the PLC with Module 9186.

		Set	Reset
M4579	**INCREMENT OFF/ON soft key**	NC	NC

Module 9036 Write status information
The information to be overwritten is designated with a transferred number.

Number of the status information	Function	Value
0 to 7	See "Handwheel"	
10	Jog increment limiting	0.0001 to 10 mm: Jog increment limiting –1; < –2; > 50: Cancellation of jog increment limitation and activation of the jog increment entered last –2: Cancellation of the jog increment limitation and activation of the minimum from the jog increment entered last and the last limitation
11 to 19	See "Handwheel"	

Call:
PS B/W/D/K <Number of the status information>
PS B/W7D/K <Value to be written>
CM 9036
PL B/W/D <Error code>
0: Status written
1: Incorrect status code
2: Transferred value is out of input range
3: Input disabled

Error recognition:

Marker	Value	Meaning
M4203	0	Status information was written
	1	Error code in W1022
W1022	1	Transferred value out of range
	2	Incorrect number of the status information
	6	Input disabled

Module 9035 Read status information

Call:

```
PS   B/W/D/K  <26>
CM   9035
PL   B/W/D    <Jog increment>
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid
	20	Call was not in a submit or spawn job

8.11 Hirth coupling

Hirth coupling describes a type of clamping of rotary axes and swivel heads. Finely splined disks mesh together in order to create a rigid connection.

During datum setting, the NC rounds off according to the grid spacing from MP430.x:

▶ Configure the exact positioning in the Hirth grid as PLC positioning.

After positioning an axis with Hirth coupling, Module 9148 can use the nominal position value of the axis as its actual position value. This affects the actual-value display and other internal calculations, such as for the transformation chain for tilting axes.

MANUAL operating mode

As soon as an axis direction key is pressed, the NC resets the corresponding bit in W1026 (axis in position).

▶ As soon as the axis-in-position bit is set again, you check the nominal position with the Hirth grid and derive from it a PLC positioning command to the next grid point.

ELECTRONIC HANDWHEEL operating mode

For the current handwheel axis, the corresponding bit is reset in W1026 (axis in position).

As soon as you select another handwheel axis, "axis in position" is set for the previous axis.

The Hirth axis can be positioned with the handwheel:

▶ Check the actual position with the Hirth grid and derive from it a PLC positioning to the next grid point.

Controlled positioning

The positions of the Hirth axis must be programmed in the grid:

▶ Check the positions in the PLC during program run.

▶ As soon as "axis in position" is reset, check the target position with the Hirth grid.

- If the target position is not in the Hirth grid, output a PLC error message.

MP420.x Hirth coupling
Input: 0: No Hirth coupling
1: Hirth coupling

MP430.x Prescribed increment for Hirth coupling
Input: 0.0000 to 30.0000 [°]

Module 9148 Use nominal value as actual value

With Module 9148 you can use the nominal value as actual value for selected axes when the position loop is open. This makes it possible to use the nominal value for certain internal functions such as the actual value display and calculations such as transformation chains of tilting axes.

If you switch from using the actual value to the nominal value with Module 9148, then the actual value is also no longer available in the PLC via Module 9041.

Call:

PS	B/W/D/K	<Axis> Bits 0 to 13 represent axes 1 to 14
PS	B/W/D/K	<Mode> 0: Use nominal value as actual value
CM	9148	

Error recognition:

Marker	Value	Meaning
M4203	0	Nominal value used as actual value
	1	Error code in W1022
W1022	1	Invalid mode
	2	Invalid axes
	24	Module was called in a spawn job or submit job

8.12 Datum Shift

With the datum shift function you can offset the defined datum point.

The same initial position must apply for the description of the machine's geometry (see "Tilting Axes" on page 705) and for the datum shift.

You can activate the datum shift during an M/S/T/Q strobe.

Datum shift with D528 to D544

- In D528 to D544, enter for each axis the distance by which the datum is to be shifted, or use Module 9230. For axes 6 to 9, use only Module 9230.
- Activate the datum shift with M4132. After the datum shift the NC resets M4132.

The offset is calculated into the position display — the display now shows the position values according to the shifted coordinate system.

Example:
Actual value display for X axis without datum shift = 50
Shift value in D528 = +20
M4132 is set, i.e. the offset is active
New actual value display X = +70 (the old datum receives the value 20).

		Set	Reset
D528 - 544	**Datum shift for axis 1 to 5**	PLC	PLC
M4132	**Activate datum shift from D528 to D544, or call Module 9230**	PLC	NC

Module 9230 Datum shift

With this module you transfer the axis and the amount by which the datum is to be shifted.

M4132 is set when Module 9230 is called. After execution of the datum shift, the NC resets M4132.

Call:
PS B/W/D/K <Axes bit-encoded>
PS B/W/D/K <Shift [0.1 µm]>
CM 9230

Error recognition:

Marker	Value	Meaning
M4203	0	No errors
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing strobe or control is active
	24	Module was called in a spawn or submit job

8.13 Touch Probe

The following touch probes can be connected:

- TS 120, TS 220: Touch-trigger probe with cable connection for workpiece setup and measuring during machining
- TS 440, TS 640: Touch-trigger probe with infrared transmission for workpiece setup and measurement during machining
- TS 444: Battery-free touch-trigger probe with infrared transmission for workpiece setup and measurement during machining
- TT 130: Touch-trigger probe for tool measurement

▶ Specify in MP6010 which touch probe is connected.

▶ Make sure that the spindle is locked during the measuring process.

With **FN18: SYSREAD** you can read the current touch probe data.

MP6010	**Selection of the touch probe**
Input:	0: Touch probe with cable transmission (TS 120, TS 220) 1: Touch probe with infrared transmission (TS 632) 2: Touch probe with infrared transmission (TS 440, TS 640) 3: Battery-free touch probe with infrared transmission (TS 444)

8.13.1 Using the touch probes

TS 440,TS 444, TS 640

The TS 440, TS 444 and TS 640 touch probes are activated on the rising edge of the starting signal and deactivated on the falling edge. For safety reasons, the touch probes are automatically switched off after 30 minutes of no activity (no stylus deflection).

HEIDENHAIN recommends:

▶ Set M4056 as soon as the touch probe is in the spindle. This way the touch probe is switched on via the rising edge of the start signal.

▶ Reset M4056 in order to switch the touch probe off. The touch probe is switched off via the falling edge of the start signal.

As long as M4056 is set, the NC checks the ready signal of the touch probe, and automatically triggers it anew as soon as it switches itself off. This ensures that the touch probe is switched on as long as it is within the working space.

If it is not possible to keep the touch probe ready, positioning motions in the **Manual** and **Handwheel** modes of operation are interrupted with the **Probe system not ready** error message. After acknowledging the error message with the CE key, the touch probe can be freely traversed for 60 seconds in the **Manual** mode of operation. The error message then appears again.

> Attention
>
> Touch-probe monitoring is not active as long as the touch probe does not report that it is ready.

If, in an exceptional case, the touch probe must be triggered anew during a probing cycle, the **Probe system not ready** error message appears, and the cycle must be restarted from the beginning.

> Note
>
> The iTNC 530 always emits a start signal when beginning a touch probe cycle, meaning Modules 9135 and 9136 do not need to be used for HEIDENHAIN touch probes.

If M4056 is set and the stylus is deflected, the NC stops the machine in all operating modes. The maximum feed rate is limited to the value specified in MP6150. If M4056 is set and the touch probe does not provide a ready signal, the feed-rate enabling is reset.

If you do not set M4056, the control detects a deflection of the stylus only if a probing function has been started.

Before the probing process is started, the NC sets M4055. Before executing the function, the NC waits until you reset M4055. This allows you to take a break, for example, to clean the measured object with compressed air before starting the probing process.

M4051 is set if the stylus is deflected before the probe block has been started. If it is, the probing block start is delayed by 1 second

Software 340 49x-04 introduces marker M4054 for the battery-free TS 444 (MP6010 = 3) touch probe.
The marker is set when the touch probe is ready for operation (M4050) and the supply voltage for the battery-free touch probe is too low. If no touch probe is connected, the value of marker M4054 is undefined. Power is supplied to the battery-free touch probe by compressed air driving an integrated air turbine generator. The capacitor in the touch probe stores energy for approx. two minutes of continuous operation. Evaluation of marker M4054 makes it possible to purposefully control the supply of compressed air.

The NC takes over control of the probing process. Certain conditions are indicated in M4050 to M4054.

		Set	Reset
M4050	**Touch probe not ready, ready signal is missing**	NC	NC
M4051	**Stylus deflected before start of probing cycle**	NC	NC
M4052	**Stylus is deflected, probing process is completed**	NC	PLC
M4053	**Probing process has been completed or canceled**	NC	NC
M4054	**Battery voltage too low (battery warning at touch probe connection); evaluated only during the probing process**[a]	NC	NC
M4055	**Enable the probing process**	NC	PLC
M4056	**NC stop in all operating modes if stylus is deflected**	PLC	PLC
M4057	**Touch probe cycles active (FN17: ID990 NR2)**	NC	NC

a. Is not supported as of 340 422-03 and 340 480-03, since the warning by the NC suffices
Is supported for the battery-free TS 444 (MP6010 = 3) touch probe as of 340 49x-04.

Non-HEIDENHAIN touch probes

Module 9135 Switching on the 3-D touch probe
With Module 9135 you can switch on or retrigger certain 3-D touch probes. If the touch probe is already switched on, the module call has no effect. If M4056 is set and the touch probe does not provide a ready signal, the feed-rate enabling (M4563) is reset.

Call:
CM 9135

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error in module run

Module 9136 Switching the touch probe on/off
Module 9136 switches a touch probe at X12 on or off once. If the touch probe does not supply a ready signal, and if M4056 is set (NC stop for deflected touch probe in all operating modes), the feed-rate enable is reset.
After Module 9136 is called the first time, the NC does not output any more on/off signals, meaning that the touch probe is now solely controlled by the PLC program.

Call:
PS B/W/D/K <Touch probe state>
0: Switch off touch probe
1: Switch on touch probe
CM 9136

Error recognition:

Marker	Value	Meaning
M4203	0	Touch probe on or off
	1	Error code in W1022
W1022	1	Invalid touch probe state

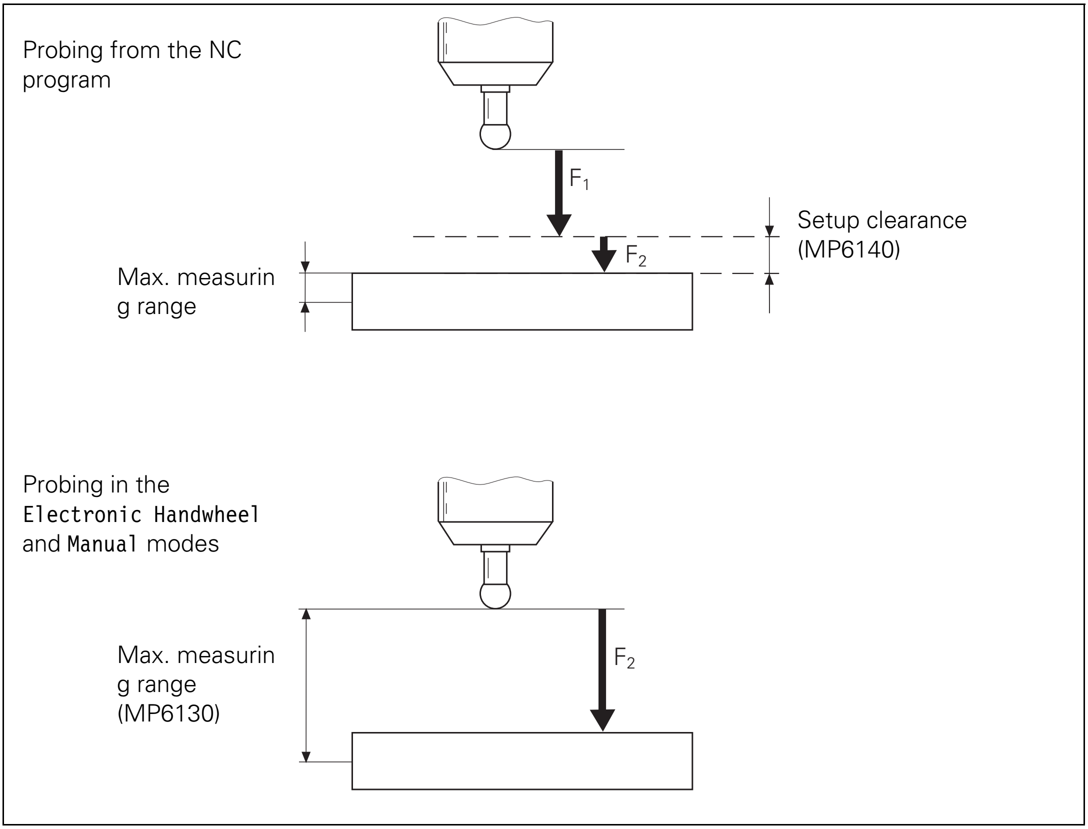
8.13.2 Touch probe cycles

The probing cycles are available in the **Manual** and **Electronic Handwheel** modes and in the NC program (see the Touch Probe Cycles User's Manual).

- With the machine parameters, adjust the touch probe to the measuring conditions.
- With MP6165, you can specify if during manual measurement and in the touch probe cycles 0 and 1 the touch probe with infrared transmission is oriented so that it is always deflected in the same direction.
- With MP6166 you can set whether the probing directions is transformed in the rotated plane in the manual measuring cycles and when basic rotation is active.

Note

Please note that MP6166 is not in effect for the calibration cycles and the cycles for determining the basic rotation.



F1 = rapid traverse during probing from the NC program:
MP6150 for triggering touch probe

F2 = probing feed rate:
MP6120 for triggering touch probe

If the maximum measuring range (MP6130) is exceeded, the error message **Touch point inaccessible** appears.

MP6140 and MP6150 have no meaning in the **Manual** and **Electronic Handwheel** operating modes.

MP6120 **Probing feed rate (triggering touch probe)**
Input: 1 to 10 000 [mm/min]

MP6130 **Maximum measuring range**
Input: 0.001 to 99 999.9999 [mm]

MP6140 **Setup clearance above measuring point**
Input: 0.001 to 99 999.9999 [mm]

MP6150 **Rapid traverse in probing cycle**
Input: 10 to 20 000 [mm/min]

MP6151 **Pre-positioning in probing cycle with rapid traverse**
Input: 0: Pre-position with speed from MP6150
1: Pre-positioning at rapid traverse

MP6165 **Orient the probe before approaching with Cycle 0 or 1, or with manual probing**
Input: 0: Probe is not oriented before each probing
1: Probe is oriented and always deflected in the same direction

MP6166 **Probing direction of the touch probe with consideration of an active basic rotation**
Input: 0: Inactive (default)
1: Active

Special case: Horizontal / vertical swivel head

If you are using a **horizontal/vertical swivel head,** the compensation values of the touch probe must be accounted for in different axes:

- With Module 9153, switch the touch probe axis in order to correctly account for the compensation values.

Module 9153 Switching the touch probe axis
Specify a new touch probe axis (axis 0, 1 or 2) for manual measurement. A new touch probe axis can be specified only if MP7490 bit 2 = 1.

Call:
PS B/W/D/K <Axis number 0 to 2>
CM 9153

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis number
	20	Module was not called in a spawn or submit job

Calibration data

The iTNC can save the calibration data for up to three touch probes at once:

- Set MP7490 bit 2 = 1.
- Use the traverse range switching function to activate the current data with M4574/M4575.
- Specify with MP7411 bit 0 whether a probing block is to use the tool data (length, radius, axis) from the last `TOOL CALL` block or from the calibrated data of the touch probe. If MP7411 bit 0 = 1, you can use soft keys to take the effective length and effective radius over into the tool table. Also, the tool name and tool number of the touch probe are then displayed in manual touch probe cycles.
- In the `Manual` and `Electronic Handwheel` operating modes, enter the tool number in the menu for touch probe calibration.

		Set	Reset
M4574	**Select the traverse range (with M4575)**	PLC	PLC
M4575	**Select the traverse range (with M4574)**	PLC	PLC

MP7411 **Tool data in the touch probe block**
Input: Bit 0 –
0: Use the calibrated data of the touch probe
1: Use the current tool data from the last TOOL CALL

MP7490 **Functions for traverse ranges**
Format: %xxxx
Input: Bit 2 – Calibration data: touch probe for workpiece measurement:
0: One set of calibration data for all traverse ranges
1: Every traverse range has its own set of calibration data

Multiple probe calibration data blocks

You can use the tool table to manage several blocks of touch probe calibration data. Use the tool table columns `CAL-OF1` (touch probe center offset in the reference axis), `CAL-OF2` (touch probe center offset in the minor axis) and `CAL-ANG` (spindle angle when calibrating). In the standard setting, these columns are hidden. They can be shown, however, with MP7266.28, MP7266.29 and MP7266.30. The current touch probe calibration data can be viewed and edited in the calibration menu for manual measurement:

- With MP7411 bit 1, activate the probe calibration management function in the tool table. If bit 1 = 1, bit 0 has no function.

MP7411 **Tool data in the touch probe block**
Input: Bit 1 –
0: Only one set of touch probe calibration data
1: Use the tool table to manage more than one set of touch probe calibration data

Probing from OEM cycles

- With **FN17:SYSWRITE ID990 NR1** adjust the approach behavior. If the input value = 0, the setup clearance from MP6140 and the effective radius are accounted for. If the input value > 0, the workpiece is approached as if the effective radius and setup clearance were zero. This function can be used, for example, for measuring small holes.

If you are using a TS 440 or TS 640 infrared touch probe, then before starting the probing cycle, you should orient the touch probe to the position at which it was calibrated. Define an M function for automatically orienting the probe to this position before probing. If the ready signal of the touch probe is already available, the touch probe is not oriented.

- In MP6161, enter the number of the M function.
- Enter the orientation angle in MP6162.
- In MP6163, enter a minimum angle difference for orienting with the defined M function..

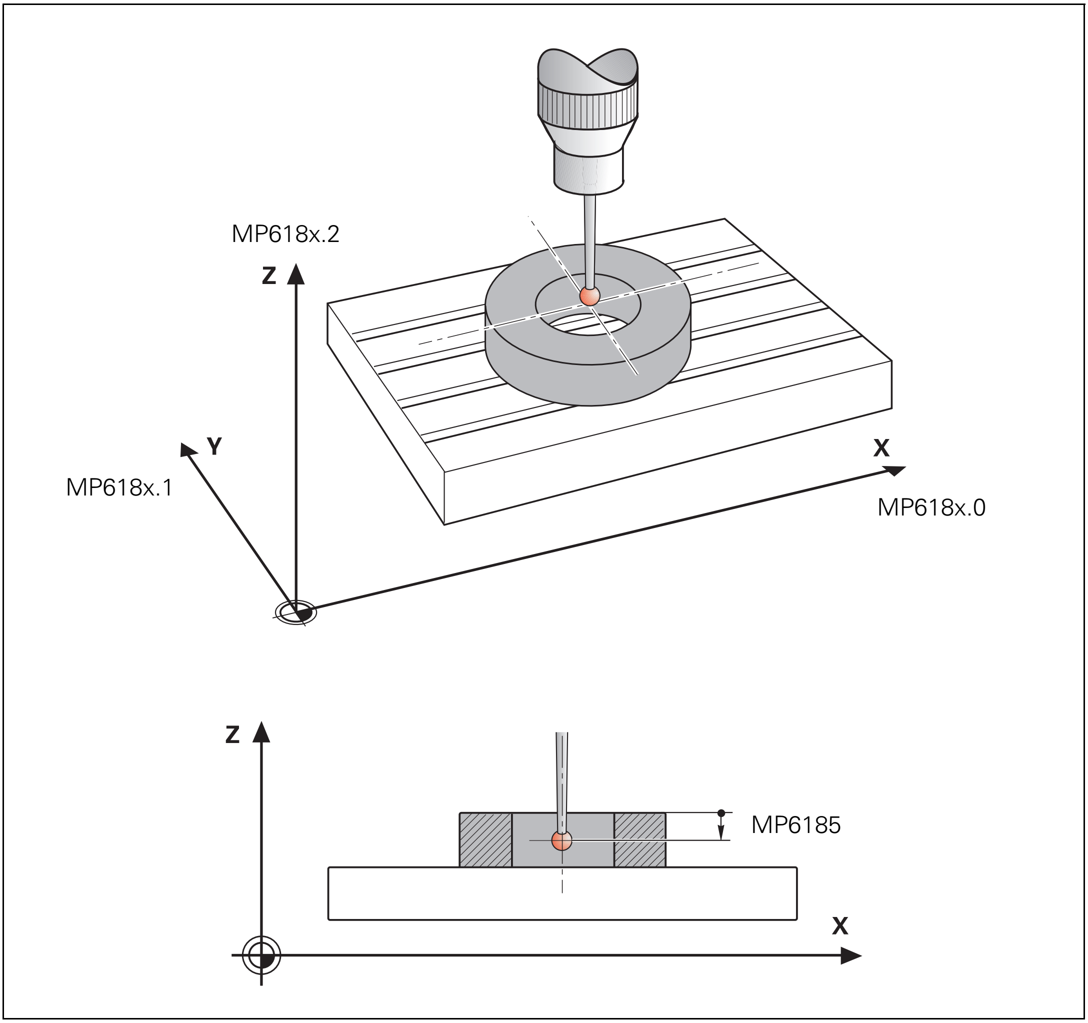
Orient the touch probe if MP6163 < (current spindle angle – MP6162)

MP6161	**M function for orienting the touch probe before every measuring process**
Input:	–1: Spindle orientation directly through NC 0: Function inactive 1 to 999: Number of the M function
MP6162	**Orientation angle**
Input:	0 to 359.9999 [°]
MP6163	**Minimum difference between the current spindle angle and MP6162 before executing an oriented spindle stop**
Input:	0 to 3.0000 [°]

Calibration

To calibrate the touch probe from within the NC program:

- In MP618x.0 and MP618x.1, enter the approximate position of the ring gauge center.
- In MP618x.2, enter the surface of the ring gauge with respect to the spindle nose. Be sure to consider the length of the touch probe or of the tool.
- In MP6185, enter the distance of the probing point below the ring's top surface.



If you probe from opposite orientations during calibration, the control stores the spindle orientation position during calibration (calculation of center offset for X and Y). You can probe at any spindle angle at a later date and the control will consider the current spindle angle and compensate for the center offset accordingly. Therefore, you do not need to orient the spindle to a specific position for probing.

The center offset is then automatically compensated during all probing processes (see the User's Manual):

- With MP6160, select whether the spindle should be oriented in a 180° rotation directly through the NC or through the PLC.
  For spindle orientation by the NC you must reset M4012.
  For spindle orientation by the PLC you must enter the number of the M function in MP6160. The respective position is transferred as in the "oriented spindle stop" cycle.

In the **Manual Operation** and **Electronic Handwheel** operating modes, the rotation is activated after a soft key is pressed.

Special case: tilting axes

The actual position of the spindle position encoder can vary with tilted axes. It depends on the machine's mechanical design. Since the iTNC uses the actual position of the spindle as its reference when compensating the eccentricity, it would be necessary to recalibrate the touch probe for each new tilt in position:

- In D760, enter the current offset with respect to the initial position.
- Calibrate the touch probe in the initial position.

The iTNC compensates the entered offset when compensating the eccentricity. In the initial position, D760 must equal 0.

		Set	Reset
D760	**Offset in tilting axes touch probe center offset [1/10 000°]**	PLC	PLC
M4012	**Open the spindle control loop**	PLC	PLC

MP6160	**M function for probing from opposite directions**
Input:	–1: Spindle orientation directly by NC 0: Function inactive 1 to 999: Number of the M function for spindle orientation through PLC
MP6180	**Coordinates of the ring gauge center for Probing Cycle 2 with respect to the machine datum (traverse range 1)**
Input:	0 to +99 999.9999 [mm]
MP6180.0	X coordinate
MP6180.1	Y coordinate
MP6180.2	Z coordinate
MP6181	**Coordinates of the ring gauge center for Probing Cycle 2 with respect to the machine datum (traverse range 2)**
Input:	0 to +99 999.9999 [mm]
MP6181.0	X coordinate
MP6181.1	Y coordinate
MP6181.2	Z coordinate

MP6182	**Coordinate of the ring gauge center for Probing Cycle 2 with respect to the machine datum (traverse range 3)**
Input:	0 to +99 999.9999 [mm]
MP6182.0	X coordinate
MP6182.1	Y coordinate
MP6182.2	Z coordinate

MP6185	**Distance of probing point below ring top surface during calibration**
Input:	+0.001 to +99 999.9999 [mm]

Measuring tolerance

In the touch probe cycles for NC programs for automatic workpiece measurement you can enter limit values and use them for tolerance monitoring.

The following markers are set by the NC. You can evaluate them through the PLC:

- M4065: All workpiece dimensions are OK
- M4066: Workpiece must be remachined
- M4067: Workpiece to be scrapped

When probing from the NC program you can repeat measurements as desired in order to increase measurement precision:

- ▶ In MP6170 enter the number of measurements to be performed per probing process.
- ▶ Enter in MP6171 a value by which the measurement result may differ.

The mean value is formed from the measurement results. If the individual results of measurement differ by more than the tolerance defined in MP6171, an error message is output. This function can be used to detect whether a measurement has been influenced, for example, by chips.

		Set	Reset
M4065	**Workpiece dimensions are OK**	NC	PLC
M4066	**Workpiece must be reworked**	NC	PLC
M4067	**Workpiece is scrap**	NC	PLC

MP6170	**Number of measurements in a programmed measurement (touch probe block)**
Input:	1 to 3
MP6171	**Confidence range for programmed measurement (MP6170 > 1)**
Input:	0.002 to 0.999 [mm]

8.13.3 Measurement log in manual touch probe cycles

For every manual touch probe cycle there is one print mask per language. Standard print masks are saved on the control's hard disk before it is shipped from the factory.

With the print masks, the output format of the measurement results is defined in the %TCHPRNT.A file:

- In the MOD menu "RS232/RS422 Setup" in the **PRINT** line, define the path for the %TCHPRNT.A file:

If the path name begins with **RS232:** or **RS422:** the measurement results are transferred over the data interface.

If no path is entered, the file is saved in the root directory TNC:\.

- Start the output of the measurement data with the **PRINT** soft key in the manual probe cycle.

If you do not wish to use the standard print masks, you can create you own print masks:

- Save these masks in the language-specific paths on the PLC partition (see "Conversational language" on page 1262).

File names of the individual print masks

- Calibration for length, triggering touch probe: TSLCAL.A
- Calibration for radius, triggering touch probe: TSRCAL.A
- Basic rotation: ROT_2PTS.A
- Point measuring: DAT_SURF.A
- Corner as datum: DAT_CORN.A
- Circle center as datum: DAT_CC.A
- Basic rotation over 2 holes: ROT_2HLS.A
- Datum over 4 holes: DAT_IS4H.A
- Circle center over 3 holes as datum: DAT_CC3H.A
- Calibration for length, measuring touch probe: TBLCAL.A
- Calibration of measuring touch probe: TMCAL.A

Format of the print masks

For the text lines of the print masks:

- Lines of text must be put into quotation marks.
- Each line must be concluded with a semicolon.
- Format instructions can be given in the C programming language.
- Variables of the format instructions must be separated by commas and placed after the text string.
- Special control commands:
  - **MM** and **INCH:** Switch the display to mm or inches. The commands affect only number types that allow an inch representation.
  - **mm_display:** The following values are displayed only if under MOD **Change M/INCH** is set to **MM.**
  - **inch_display:** The following values are displayed only if under MOD **Change MM/INCH** is set to **INCH.**
  - **all_display:** The following values are displayed regardless of the setting in **MM/INCH** under MOD.

Variable names:

Time management		
Name	**Format type**	**Description**
HOUR	int	No. of hours from real-time clock
MIN	int	No. of minutes from real-time clock
SEC	int	No. of seconds from real-time clock
DAY	int	Day from real-time clock
MONTH	int	Month as no. from real-time clock
STR_MONTH	string	Month as string abbr. from real-time clock
YEAR2	int	Two-digit year no. from real-time clock
YEAR4	int	Four-digit year from the real-time clock

Settings of the manual measuring cycles		
Name	**Format type**	**Description**
TCH.AXIS	string	Selected probe axis
TCH.PLANEROT	double	Basic rotation angle
TS.RAD	double	Calibrated probe radius
TS.LEN	double	Calibrated stylus length
TS.OFF1	double	Calibrated center offset in reference axis
TS.OFF2	double	Calibrated center offset in minor axis
TS.RINGRAD	double	Radius of calibration ring

Results or input from the manual measuring cycles		
Name	**Format type**	**Description**
BZ	double	Reference point
BEZA	string	String reference point axis

Reference point at corner, circle, 4 holes, 3 holes on a circle		
Name	**Format type**	**Description**
BZ_HA	double	Reference point in reference axis
BZ_NA	double	Reference point in minor axis
LKALBEZ	double	Reference point entered with calibrated probe length
HA	string	Reference-axis character
NA	string	Minor-axis character
TA	string	Probe axis character

Calculated straight lines from straight-line probing		
Name	**Format type**	**Description**
GE_HA[2]	double	Straight-line axis section of reference axis
GE_NA[2]	double	Straight-line axis section of minor axis
GE_WI[2]	double	Straight-line angle

Calculated radii from circle probing		
Name	**Format type**	**Description**
RAD[8]	double	8 radii

Calculated center points from circle probing		
Name	**Format type**	**Description**
MP_HA[8]	double	Reference axis of center points
MP_NA[8]	double	Minor axis of center points

Accumulated touch points from probes		
Name	**Format type**	**Description**
AP_HA[32]	double	Touch points in reference axis
AP_NA[32]	double	Touch points in minor axis
AP_TA[32]	double	Touch points in probe axis

Example

```
"Touch probe calibration";
"___________________";
%02.2d-%02.2d-%4d:%02.2d:%02.2d "Time",DAY,MONTH,YEAR4,HOUR,MIN,SEC;
Probe axis:"%s",TA;
Probe radius: "%4.3lf" TS.RAD;
Probe length: "%4.3lf" TS.LEN;
Ring radius: "%4.3lf",TS.RINGRAD;
Center offset in reference axis: "%4.3lf" TS.OFF1;
Center offset in minor axis: "%4.3lf" TS.OFF2;
```

8.13.4 Measurement log in the touch probe cycles for probing from the NC program

For every touch probe cycle for probing from the NC program there is a print mask for all languages. For the HEIDENHAIN touch probe cycles a print mask is saved for every cycle on the hard disk. This print mask cannot be changed. However, you can provide an OEM touch probe cycle with a print mask of your own.

Unlike the print masks for the manual touch probe cycles, for the touch probe cycles for probing from the NC program you only need one print mask. The individual text blocks are distinguished through language code words. The text block that is defined in MP7230.0 is always output.
Otherwise the syntax of the print masks is identical.

Conversational language	Language code word
English	L_ENGLISH
German	L_GERMAN
Czech	L_CZECH
French	L_FRENCH
Italian	L_ITALIAN
Spanish	L_SPANISH
Portuguese	L_PORTUGUE
Swedish	L_SWEDISH
Danish	L_DANISH
Finnish	L_FINNISH
Dutch	L_DUTCH
Polish	L_POLISH
Hungarian	L_HUNGARIA
Russian	L_RUSSIAN
Language neutral texts	L_ALL

Example Here you see the print mask of Cycle 421 for English and German.

```
L_ENGLISH;
"-----------------------------------------------------------------------
-----"; "************ Measuring Log for Probing Cycle 421: Hole
Measuring **********";
"Date: %02.2d-%02.2d-%4d",DAY,MONTH,YEAR4;
"Time: %2d:%02.2d:%02.2d",HOUR,MIN,SEC;
"Measuring program: %S",CALL_PATH;
"-----------------------------------------------------------------------
-------";
" ";
"Nominal values:   Center in 1st axis: %6.4LF", Q273;
"                  Center in 2nd axis: %6.4LF", Q274;
"                  Diameter: %6.4LF", Q262;
"";
"-----------------------------------------------------------------------
--------";
"";
"Given limit values:Maximum dimension for center in 1st axis: %6.4LF",
Q31;
"                  Minimum dimension for center in 1st axis: %6.4LF",
Q32;
"";
"                  Maximum dimension for center in 2nd axis: %6.4LF",
Q33;
"                  Minimum dimension for center in 2nd axis: %6.4LF",
Q34;
"";
"                  Maximum dimension for hole: %6.4LF", Q275;
"                  Maximum dimension for hole: %6.4LF", Q276;
"";
"*********************************************************************
********";
"";
"Actual values:    Center in 1st axis: %6.4LF", Q151;
"                  Center in 2nd axis: %6.4LF", Q152;
"                  Diameter: %6.4LF", Q153;
"";
"-----------------------------------------------------------------------
--------";
"";
"Deviations: Center in 1st axis: %6.4LF", Q161;
"            Center in 2nd axis: %6.4LF", Q162;
"            Diameter: %6.4LF", Q163;
"";
"*********************************************************************
********";
"";
"Further measuring results: measuring height: %6.4LF", Q261;
"";
"**************************** End of Measuring Log
****************************";
L_GERMAN;
"-----------------------------------------------------------------------
--------"; "************** Messprotokoll Antastzyklus 421 Bohrung
```

```
messen ***************";
"Datum: %02.2d-%02.2d-%4d",DAY,MONTH,YEAR4;
"Uhrzeit: %2d:%02.2d:%02.2d",HOUR,MIN,SEC;
"Messprogramm: %S",CALL_PATH;
"-------------------------------------------------------------------
--------";
"";
"Sollwerte:  Mitte Hauptachse: %6.4LF", Q273;
"            Mitte Nebenachse: %6.4LF", Q274;
"            Durchmesser  : %6.4LF", Q262;
"";
"-------------------------------------------------------------------
--------";
"";
"Vorgegebene Grenzwerte: Größtmaß Mitte Hauptachse: %6.4LF", Q31;
"                        Kleinstmaß Mitte Hauptachse: %6.4LF", Q32;
"";
"                        Größtmaß Mitte Nebenachse: %6.4LF", Q33;
"                        Kleinstmaß Mitte Nebenachse: %6.4LF", Q34;
"";
"                        Größtmaß Bohrung: %6.4LF", Q275;
"                        Kleinstmaß: %6.4LF", Q276;
"";
"*******************************************************************
********";
"";
"Istwerte:   Mitte Hauptachse: %6.4LF", Q151;
"            Mitte Nebenachse: %6.4LF", Q152;
"            Durchmesser        : %6.4LF", Q153;
"";
"-------------------------------------------------------------------
--------";
"";
"Abweichungen:Mitte Hauptachse: %6.4LF", Q161;
"            Mitte Nebenachse: %6.4LF", Q162;
"            Durchmesser        : %6.4LF", Q163;
"";
"*******************************************************************
********";
"";
"Weitere Messergebnisse: Messhöhe : %6.4LF", Q261;
"";
"**************************** Messprotokoll-Ende
****************************";
```

8.13.5 Tool measurement

With the HEIDENHAIN table touch probes (TT) you can measure and inspect tools. HEIDENHAIN provides standard cycles for automatic tool measurement and calibration of the TT (see the Touch Probe Cycles User's Manual).

Technical prerequisites

You need:

- TT (TT 130/TT 140)
- Central tool file TOOL.T must be active (via machine parameter)

The iTNC can save the calibration data for up to three touch probes at once:

- Use the traverse range switching function to activate the current data with M4574/M4575.
- Set MP7490 bit 3 to save three separate sets of calibration data.

MP7490	**Functions for traverse ranges**
Format:	%xxxx
Input:	Bit 3 – Calibration data: touch probe for tool measurement: 0: One set of calibration data for all traverse ranges 1: Every traverse range has its own set of calibration data

Standard measuring cycles

The TT must be mounted and interfaced:

- With MP6500 bit 0, enable the cycles for tool measurement.

MP6500	**Tool measurement with TT table touch probe**
Format:	%xxxxxxxxxxxx
Input:	Bit 0 – 0: Cycles for tool measurement disabled 1: Cycles for tool measurement not disabled

Note

In the standard measuring cycles for tool measurement, the PLC program may command a gear shift during output of the spindle speed without interrupting the cycle.

Tool radius and tool length measurement

- With MP6500 bits 1 and 2, specify whether tool radius and tool length measurements are allowed and whether individual teeth are to be measured.
- Specify in MP6500 bit 14 if tool measurement with stationary spindle is to be carried out for tools with the value 0 in the "number of teeth" column **(CUT.)** in the tool table. This can be necessary for tools with diamond teeth, for example.

MP6500	**Tool measurement with TT table touch probe**
Format:	%xxxxxxxxxxxxxxx
Input:	Bit 1 – 0: Tool radius measurement allowed Tool length measurement with rotating spindle 1: Tool radius measurement and individual tooth measurement disabled Bit 2 – 0: Tool length measurement with rotating spindle (bit 1=1) 1: Tool length measurement with rotating spindle, only if a tool radius offset (**TT: R-OFFS**) has been entered in the tool table Bit 14 – Tool measurement with number of teeth = 0 0: Tool measurement with rotating spindle 1: Tool measurement with stationary spindle

Oriented spindle stop

Spindle orientation must be active for individual tooth measurement, otherwise the tool radius measurement is subject to error:

- Define with MP6500 bit 3 whether the tool is measured with or without spindle orientation. If MP6500 bit 3 = 1 (without spindle orientation), each entry in MP6560 that does not equal 0 is not considered.
- With MP6560, specify whether the spindle is to be oriented directly via NC or through the PLC.
  - For spindle orientation directly by NC:
    Reset M4012.
  - For spindle orientation by PLC:
    Enter the number of the M function in MP6560.

The respective positions are transferred as in the "oriented spindle stop" cycle. M4017 is set during every spindle orientation.

MP6500	**Tool measurement with TT table touch probe**
Input:	Bit 3 – 0: Tool measurement with spindle orientation 1: Tool measurement without spindle orientation. Individual tooth measurement not possible. Tool radius measurement possibly faulty.
MP6560	**M function for spindle orientation during individual tooth measurement**
Input:	–1: Spindle orientation directly by NC 0: Function inactive 1 to 999: Number of the M function for spindle orientation by PLC

Probing direction

- In MP6505.x, define the probing direction for tool radius measurement.

MP6505 Probing direction for tool radius measurement for 3 traverse ranges

Input: 0: Positive probing direction in the angle reference axis (0° axis)
1: Positive probing direction in the +90° axis
2: Negative probing direction in the angle reference axis (0° axis)
3: Negative probing direction in the +90° axis

MP6505.0 Traverse range 1
MP6505.1 Traverse range 2
MP6505.2 Traverse range 3

Offset of probe contact to the tool

- In MP6530.x enter the distance from the tool end to the top of the probe contact during tool radius measurement.
- In the **L-OFFS** field of the tool table, enter an additional tool-specific offset.

MP6530 Distance from the tool end to the top of the probe contact during tool radius measurement for 3 traverse ranges

Input: 0.001 to 99.9999 [mm]

MP6530.0 Traverse range 1
MP6530.1 Traverse range 2
MP6530.2 Traverse range 3

Safety zone

After a cycle for tool measurement starts, the tool automatically moves at the feed rate defined in MP6550 from the clearance height defined in the cycle to the limit of the safety zone.

- In MP6540.x, define a safety zone around the probe contact of the TT table touch probe.
- In MP6550, define the feed rate at which the border of the safety zone is approached.

MP6540 Safety zone around the probe contact of the TT table touch probe for pre-positioning

Input: 0.001 to 99 999.9999 [mm]

MP6540.0 Safety clearance in tool axis direction
MP6540.1 Safety clearance in the plane perpendicular to the tool axis

MP6550 Rapid traverse in probing cycle for TT table touch probe

Input: 10 to 1 000 000 [mm/min]

Probe contact

- In MP6531.x, enter the diameter (disk) or the edge length (cube) for the probe contact.
- In MP6580.x, MP6581.x and MP6582.x, enter the coordinates of the probe contact center with respect to the machine datum. After calibration, the NC internally stores the exact center of the probe contact.
- If a PLC datum shift should be included in the tool measurement, set MP6500 bit 12 = 1.

For a cube it suffices to probe from one direction:

- Set MP6500 bit 8 = 1.
- With MP6500 bit 9, specify whether the basic rotation of the cube is measured automatically or whether it should be aligned to the axes mechanically. During automatic measurement, the edge of the touch probe is probed twice and the basic rotation is calculated. All subsequent probing is done automatically at a right angle to the touch probe edge.
- With MP6500 bit 10, select how to pre-position to the starting point. If bit 10 = 1, bit 9 must equal 0.

MP6500	**Tool measurement with TT table touch probe**
Format:	%xxxxxxxxxxxxx
Input:	Bit 7 – Reserved Bit 8 – Probing routine 0: Probe contact is probed from several directions 1: Probe contact is probed from one direction Bit 9 – Automatic measurement of the direction of the probe contact basic rotation (bit 8 = 1) 0: Basic rotation is not measured 1: Basic rotation of the probe element is automatically measured Bit 10 – Probing routine (bit 8 = 1) 0: Pre-positioning to starting point in all three principal axes 1: Pre-positioning to starting point in the tool axis and in the axis of the probing direction (MP6505) (bit 9 = 0) Bit 12 – Inclusion of the PLC datum shift 0: Do not include the PLC datum shift 1: Include the PLC datum shift

MP6531	**Diameter or edge length of the TT table touch probe stylus contact for 3 traverse ranges**
Input:	0.001 to 99.9999 [mm]
MP6531.0	Traverse range 1
MP6531.1	Traverse range 2
MP6531.2	Traverse range 3
MP6580.0-2	**Coordinates of the TT table touch probe stylus contact center with respect to the machine datum (traverse range 1)**
Input:	–99 999.9999 to +99 999.9999 [mm]
MP6581.0-2	**Coordinates of the TT table touch probe stylus contact center with respect to the machine datum (traverse range 2)**
Input:	–99 999.9999 to +99 999.9999 [mm]
MP6582.0-2	**Coordinates of the TT table touch probe stylus contact center with respect to the machine datum (traverse range 3)**
Input:	–99 999.9999 to +99 999.9999 [mm]

Probing feed rate and spindle speed

The probing feed rate from MP6520 is used for tool measurement with a nonrotating tool.

The iTNC automatically calculates the probing feed rate and the spindle speed for tool measurement with rotating tool. The speed is calculated from the maximum permissible surface cutting speed (MP6570) and the tool radius in the tool table:

- Enter MP6500 bit 4 = 0.
- In MP6572, enter the maximum permissible speed.
- In MP6570, enter the maximum permissible surface speed of the tool edge.

The control calculates the speed from the following formula:

$$n = \frac{MP6570}{2 \cdot \pi \cdot r \cdot 10^{-3}}$$

n: Speed [min^{-1}]
MP6570 = Maximum permissible surface speed of the tool edge [m/min]
r: Tool radius [mm]

High frequency spindles often cannot function at speeds under 1000 min^{-1}:

- In this case enter MP6500 bit 4 = 1, in order to always use the lowest possible speed for that spindle. This is automatically calculated by the TNC. MP6570 and MP6572 then are without function.

The probing feed rate is calculated from the revolutions per minute and the measuring tolerance defined in MP6510.0.

- In MP6510.0, enter the maximum permissible measuring error, the "measuring tolerance."

v = measuring tolerance · n

v: Probing feed rate [m/min]
Measuring tolerance: Measuring tolerance [mm] from MP6510.0 depending on MP6507
n: Speed [min^{-1}]

- With MP6507, specify the type of calculation of the probing feed rate.

MP6507=0: Calculation of the probing feed rate with constant tolerance

The measuring tolerance remains constant, regardless of the tool radius. For large tools, however, the probing feed rate becomes so small that it falls below the smallest programmable increment and becomes zero. The smaller the maximum surface cutting speed and the measuring tolerance, the sooner this effect begins.

MP6507=1: Calculation of the probing feed rate with variable tolerance

The measuring tolerance changes depending on the tool radius. A probing feed rate results even for large tool radii.

The measuring tolerance is changed according to the following table:

Tool radius	Measuring tolerance
Up to 30 mm	MP6510.0
30 mm to 60 mm	2 · MP6510.0
60 mm to 90 mm	3 · MP6510.0
90 mm to 120 mm	4 · MP6510.0

MP6507=2: Constant probing feed rate

The probing feed rate remains the same, regardless of the tool radius. The absolute measuring error grows proportionally with the size of the tool radius.

$$\text{Measuringtolerance} = \frac{r}{5\,[\text{mm}]} \cdot \text{MP6510.0}$$

r: Tool radius [mm]
MP6510.0: Max. permissible measuring error [mm]

$$v = \frac{\text{MP6570} \cdot \text{MP6510}}{2 \cdot \pi \cdot 10^{-3}}$$

v: Probing feed rate [m/min]
MP6570: Maximum permissible surface speed of the tool edge [m/min]

MP6500 **Tool measurement with TT table touch probe**
Format: %xxxxxxxxxxxxx
Input: Bit 4 –
0: Automatically determine speed
1: Always use minimum spindle speed

MP6507 **Calculation of the probing feed rate**
Input: 0: Calculation of the probing feed rate with constant tolerance
1: Calculation of the probing feed rate with variable tolerance
2: Constant probing feed rate

MP6520 **Probing feed rate for tool measurement with non-rotating tool**
Input: 1 to 10 000 [mm/min]

MP6570 **Max. permissible surface cutting speed at the tooth edge**
Input: 1.0000 to 129.0000 [m/min]

MP6572 **Maximum permissible speed during tool measurement**
Input: 1 to 1000 [min^{-1}]
0: 1000 [min^{-1}]

Individual tooth measurement

The TNC attempts to maintain the tolerance from MP6510.0 during the tooth search for individual tooth measurement. At the same time MP6510.0 is used to calculate the probing feed rate.
If the tolerance from MP6510.0 cannot be maintained during the tooth search, e.g. due to the missing spindle accuracy, the TNC attempts to maintain the tolerance from MP6510.1. If this also cannot be maintained, the error message `Tolerance in MP6510 too small` appears.

- Enter the first maximum measuring error in MP6510.0.
- Enter the second maximum measuring error in MP6510.1.

MP6510	**Permissible measuring error for tool measurement with rotating tool**
Input:	0.002 to 0.999 [mm]
MP6510.0	First measurement error
MP6510.1	Second measurement error

Monitoring of the rotary axes and secondary linear axes

To ensure that the rotary axes and the secondary linear axes are always in a defined position during the tool measuring cycles:

- In MP6585, enter the axes to be monitored.
- In MP6586.x, enter the reference coordinate at which the axis should be located during the tool measuring cycles.

If, during activated monitoring, the nominal position does not match the position from MP6586.x, an error message is displayed.

MP6585	**Monitoring the position of the rotary and additional linear axes during the tool measurement cycles**
Format:	%xxxxxx
Input:	0: Axis is not monitored 1: Axis is monitored Bit 0 – A axis Bit 1 – B axis Bit 2 – C axis Bit 3 – U axis Bit 4 – V axis Bit 5 – W axis

MP6586	**Ref. coordinate for monitoring the position of the rotary and additional linear axes during the tool measurement cycles**
Input:	–99 999.9999 to +99 999.9999 [mm] or [°]
MP6586.0	A axis
MP6586.1	B axis
MP6586.2	C axis
MP6586.3	U axis
MP6586.4	V axis
MP6586.5	W axis

Tool measurement in a tilted coordinate system

- If the tool is to be measured in a tilted position other than that in which the tool touch probe was calibrated, set MP6500 bit 13 = 1.

MP6500	**Tool measurement with TT table touch probe**
Format:	%xxxxxxxxxxxxx
Input:	Bit 13 0: Tool is measured in the tilt position in which the tool touch probe was also calibrated 1: Tool is measured in another tilt position

> Attention
>
> If the tool is not measured in the same tilt position as that in which the tool touch probe was calibrated, ensure that the tool is perpendicular to the contact plate!

Tool breakage

- With MP6500 bits 5 and 6, specify whether the NC program should stop when the breakage tolerance is exceeded. M4063 is always set when the breakage tolerance is exceeded.
- With bit 11, specify whether the result of "tool checking" measurement is to be entered in the tool table.

MP6500	**Tool measurement with TT table touch probe**
Format:	%xxxxxxxxxxxxx
Input:	Bit 5 – NC stop during "tool checking" 0: The NC program is not stopped when the breakage tolerance is exceeded 1: If the breakage tolerance is exceeded, the NC program is stopped and the error message "Tool broken" is displayed. Bit 6 – NC stop during "tool measurement" 0: The NC program is not stopped when the breakage tolerance is exceeded. 1: If the breakage tolerance is exceeded, the NC program is stopped and the error message "Touch point inaccessible" is displayed. Bit 11 – "Tool checking" and changing in the tool table 0: After "tool checking" the tool table is changed 1: After "tool checking" the tool table is not changed

M functions before and after tool measurement

Machine parameters MP6562.0 and MP6562.1 can be used to run M functions before and after a tool measurement cycle (TT cycle). MP6562.0 activates an M function before the beginning of a cycle, MP6562.1 after the end of a cycle. You need to program M-function macros in an NC block to be able to run them.

If the M-function macros are used to make changes that are not automatically undone at an internal stop, you must ensure that the machines are restored to a consistent state (e.g. through a suitable cancel macro).

MP6562	**M function before and after tool measurement**
Input:	0 to 999 [M function macro] –1: Function inactive

Markers in the PLC

M4060 is set if a cycle for tool measurement is started.

M4061 displays whether a cycle was activated for tool measurement or for tool checking.

M4062 and M4063 are set if during tool checking one of the entered tolerances was exceeded. The tool is locked.

The markers M4050, M4051, M4052, M4053, M4055 and M4056 function as in the standard cycles. You must enable the cycles for tool measurement with M4055. For spindle orientation directly by the NC (MP6560 = –1), you must reset M4012.

		Set	Reset
M4060	**Cycle for tool measurement started**	NC	NC
M4061	**0: Measure the tool** **1: Check the tool**	NC	NC
M4062	**0: Wear tolerance not exceeded** **1: Wear tolerance exceeded**	NC	NC/PLC
M4063	**0: Breakage tolerance not exceeded** **1: Breakage tolerance exceeded**	NC	NC/PLC

8.14 Special Functions for Laser Cutting Machines

You can activate special functions to interface the iTNC to laser cutting machines and water jet machines.

8.14.1 Analog voltage output

If you do not need the analog output S for the spindle, you can define other functions for this output:

- With MP3011, select the function of analog output S. If MP3010 > 3, MP3011 has no effect.

MP3011	**Function of analog output S, if MP3010 < 3**
Input:	0: No special function 1: Voltage is proportional to the current contouring feed rate, depending on MP3012 2: Voltage is defined as through Module 9130 3: Voltage is defined through M functions (M200 to M204)

Voltage proportional to the contouring feed rate, MP3011 = 1

A voltage proportional to the current contouring feed rate is output:

- In MP3012, enter the feed rate achieved when a 10-V analog voltage is output.

MP3012	**Feed rate from output of an analog voltage of 10 V, MP3011 = 1**
Input:	0 to 300 000 [mm/min]

Voltage from the PLC, MP3011 = 2

The voltage that you have defined with Module 9130 is output.

Definition of the voltage through M functions, MP3011 = 3

The voltage to be output is defined through M functions M200 to M204:

- Set MP3011 = 3, otherwise the M functions described above are not available.

The M functions are executed synchronously to the positioning blocks and are effective at the beginning of the positioning blocks.

Direct output of the programmed voltage: M200 V...

The iTNC outputs the value after M200 V... as a voltage.

Input: 0 to 9.999 [V]
Duration of effect: M200 V... is effective until a new voltage is output with M200 to M204.

Voltage output varies with the distance: M201 V...

The iTNC outputs the voltage as a function of the traversed distance. Starting from the active voltage, the iTNC increases or decreases the voltage linearly to the value programmed behind M201 V.

Input: 0 to 9.999 [V]
Duration of effect: M200 V... is effective until a new voltage is output with M200 to M204.

Voltage output varies with the velocity: M202 FNR

The iTNC outputs the voltage as a function of the velocity:

- In MP3013.x and MP3014.x, define up to three characteristic curves in a table.

In the table, certain analog voltages are assigned to certain feed rates:

- With M202 FNR select the curve in which the iTNC finds the voltage to be output.

Input: 1 to 3
Duration of effect: M202 FNR is effective until a new voltage is output with M200 to M204.

You can enter up to four kink points per curve in the table. The values to be distributed are interpolated linearly between the kink points. The first kink point must start with the input value zero. For the following kink points of the curve the input values must rise steadily. The iTNC detects the beginning of a new curve from the input value zero.

Example:

Velocity		Voltage		Curve
MP3013.0	0	MP3014.0	0	1
MP3013.1	25	MP3014.1	0	
MP3013.2	500	MP3014.2	4.5	
MP3013.3	1000	MP3014.3	9.999	
MP3013.4	0	MP3014.4	0	2
MP3013.5	10 000	MP3014.5	9.999	
MP3013.6	0	MP3014.6	0	3
MP3013.7	50	MP3014.7	0.5	
MP3013.8	300	MP3014.8	1.5	
MP3013.9	5000	MP3014.9	9.999	
MP3013.10	0	MP3014.10	0	Not used
MP3013.11	0	MP3014.11	0	

MP3013.x Characteristic curve kink points (velocity) for output of the analog voltage with M202
Input: 10 to 300 000 [mm/min]

MP3014.x Characteristic curve kink points (voltage) for output of the analog voltage with M202
Input: 0.000 to 9.999 [V]

Voltage output varies with the time (time-dependent ramp): M203 V... TIME...

The iTNC outputs the voltage as a function of the time. Starting from the active voltage, the iTNC increases or decreases the voltage linearly in the time programmed behind TIME to the value programmed behind V.

Input: Voltage V: 0 to 9.999 [V]
TIME: 0 to 1.999 [sec]
Duration of effect: M203 V... TIME... is effective until a new voltage is output with M200 to M204.

Voltage output varies with the time (time-dependent pulse): M204 V... TIME...

The iTNC outputs the value programmed after V... as a pulse. The duration of the pulse is specified with "TIME...."

Input: Voltage V: 0 to 9.999 [V]
TIME: 0 to 1.999 [sec]
Duration of effect: M204 V... TIME... is effective until a new voltage is output with M200 to M204.

8.14.2 Graphic simulation (without TOOL CALL)

Graphic simulation is also available on machines that operate without tool definition (e.g., water jet and laser cutting machines):

- In MP7315, specify the tool radius for the graphic simulation.
- In MP7316, define the penetration depth of the simulated tool.
- Use M functions to mark the program sections to be simulated and define the functions in MP7317.x.

Furthermore, by entering in MP7312 as small a value as possible between 1 and 8, you should be able to detect an acceleration of the graphic simulation (the 3-D graphics) in the Test Run operating mode. However, HEIDENHAIN instead recommends entering the real values of the corresponding tooth lengths in the LCUTS column of the tool table.

MP7312	**Limitation of the tooth length LCUTS**
Input:	0 = No limitation, infinitely long tooth length > 0: Tooth length = 2 * tool radius * MP7312
MP7315	**Tool radius for graphic simulation without TOOL CALL**
Input:	0.0000 to 99 999.9999 [mm]
MP7316	**Penetration depth of the tool**
Input:	0.0000 to 99 999.9999 [mm]
MP7317	**M function for graphic simulation**
MP7317.0	Beginning of graphic simulation
Input:	0 to 88
MP7317.1	Interruption of graphic simulation
Input:	0 to 88

8.14.3 Program stop for M functions and TOOL CALL S

TOOL CALL S means a TOOL CALL in which only one spindle speed was programmed.

For TOOL CALL S and also in the **PROGRAM RUN, FULL SEQUENCE** and **PROGRAM RUN, SINGLE BLOCK** modes, the output of an M function interrupts the program run until you confirm execution with M4092.

However, on applications such as laser cutting machines, the program should not be interrupted:

- With MP7440 bit 2 and MP3030 bit 0, specify whether the program run should be interrupted.
  If you deselect the program stop, you must not perform the following functions during output:
  - PLC positioning
  - Datum shift
  - Oriented spindle stop
  - Limit switch range switchover

> Attention
>
> Do not use this function on milling machines and boring mills!

MP3030 **Behavior of the spindle**
Input: Bit 0 –
0: Axis stop for TOOL CALL S...
1: No axis stop for TOOL CALL S...

MP7440 **Output of M functions**
Format: %xxxxx
Input: Bit 2 – Program stop with M functions:
0: Program stop until acknowledgment of the M function
1: No program stop, no waiting for confirmation

8.15 Tool Changer

You control the tool changer through PLC outputs.

If the tool changer is to be driven by controlled axes, use PLC axes. See page 669. You can also control the tool changer through proximity switches:

- Save the information about the tool in the tool table and the information about the tool changer in the pocket table.

Tool management (replacement tool, tool life, etc.) is handled by the NC. Markers and words provide you with the information necessary for driving the tool changer.

8.15.1 Tool and pocket number

You can edit the tool table in the machining modes of operation:

- Ensure that the tool table and pocket table are neither locked nor protected via MP7224.x. See page 1355.
- Press the TOOL TABLE soft key.

From the tool table you can call the pocket table (see the User's Manual):

- Ensure that the POCKET TABLE soft key is not hidden by MP7263 bit 0.
- Press the POCKET TABLE soft key.

The current tool table is TOOL.T, and the pocket table is TOOL_P.TCH. Both files are saved in the root directory TNC:\.

Definition of the tool and pocket table:

- In MP7266.x, specify the fields of the tool table that are to be displayed and the sequence in which they appear.
- In MP7267.x, specify the fields of the pocket table that are to be displayed and the sequence in which they appear.
- Ensure that the tool table and pocket table are neither locked nor protected via MP7224.x. See page 1355.
- In MP7260, specify the number of tools in the tool table.
  - If MP7260 = 0, no tool table is used (TOOL.T does not exist). In this case, you must program the tool length and radius in the NC program with a `TOOL DEF` block (see the User's Manual). There is no automatic tool management.
- If you are only using one tool magazine, set the number of pockets in MP7261.0 and enter MP7261.1-7 = 0. If you use multiple tool magazines, see "Managing multiple tool magazines" on page 1536.
  - If MP7261.0-7 = 0, no pocket table is generated.

With Modules 9092, 9350 (9093), 9351 (9094) and 9096 you can read the tool and pocket tables and overwrite them.

If an input field is open in the editor at the time the modules are called, this field is closed automatically.

The status display shows the current tool data.

With MP7263 bit 1 you configure the output of the column in the pocket table during backup and during conversion from binary format to ASCII.
By setting MP7263 bit 2, the "Edit ON/OFF" soft key can be hidden when the pocket table is displayed. This makes it possible to prevent manual editing of the pocket table.
MP7263 bit 3 can be used to hide the RESET POCKET TABLE and RESET COLUMN T soft keys.

MP7260 to MP7267 can also be overwritten by the PLC or the LSV2 protocol.

The bits #4, #5 and #6 of machine parameter MP7263 are used to define settings for deleting tools and their index entries. These settings apply to tools in the pocket table.

As of software version 340 49x-05, during control start-up the prototype pocket table **PLC:\PROTO\PROTOTYP.TCH** is used when a pocket table is created. The previous standard pocket table is only created if no prototype pocket table exists.

If bit #12 is set in MP7682, the error message "Tool radius too large" is suppressed if R2 > R for a tool in the tool table. This might be necessary if barrel cutters are used. Please keep in mind that the cutter form compensation with LN blocks does not work with this cutter shape.

MP7260 **Number of tools in the tool table**
Input: 0 to 30 000

MP7261.0-7 **Number of pockets in the tool magazine 1 to 4**
Input: 0 to 9999

MP7263 **Pocket table**
Format: %xxxxxxx
Input: Bit 0 –
0: POCKET TABLE soft key is shown
1: POCKET TABLE soft key is hidden
Bit 1 – Output of the pocket table for file functions
0: Output only the displayed columns
1: Output all columns
Bit 2 – Show the "Edit ON/OFF" soft key in the pocket table
0: Display soft key
1: Do not display soft key
Bit 3 – Display the RESET POCKET TABLE and RESET COLUMN T soft keys in the pocket table
0: Display soft key
1: Do not display soft key
Bit 4 - Deletion possible for a tool that is in the pocket table. Deletion must be confirmed.
0: Deletion impossible
1: Deletion possible (with confirmation)
Bit 5 – Deletion of a tool possible even without confirmation (if bit #4 = 1)
0: Deletion not possible without confirmation
1: Deletion possible without confirmation
Bit 6 – Deletion of index entries of a tool behaves like deletion of a tool. The settings of bit#4 and #5 also apply to the index entries if bit#6 is set.
0: Deletion always impossible
1: Deletion possible depending on settings in bits #4 and #5

MP7266 **Elements of the tool table**
Input: 0 = no display
1 to 99 = position in the tool table

MP7682 **MP with multiple function**
Input: Bit 12 – Error message "Tool radius too large" if R2 > R
0 = Error message displayed
1 = Error message is not displayed

MP	Meaning	Column name	Column width
MP7266.0	16-character alphanumeric tool name	NAME	16
MP7266.1	Tool length	L	11
MP7266.2	Tool radius	R	11
MP7266.3	Tool radius 2 for toroidal cutter	R2	11
MP7266.4	Oversize in tool length	DL	8
MP7266.5	Oversize in tool radius	DR	8
MP7266.6	Oversize in tool radius 2	DR2	8
MP7266.7	Locked tool?	TL	2
MP7266.8	Replacement tool	RT	3
MP7266.9	Maximum tool age (M4543)	TIME1	5
MP7266.10	Maximum tool age TOOL CALL	TIME2	5
MP7266.11	Current tool age	CUR.TIME	8
MP7266.12	Comment on the tool	DOC	16
MP7266.13	Number of tool teeth	CUT	4
MP7266.14	Wear tolerance for tool length	LTOL	6
MP7266.15	Wear tolerance for tool radius	RTOL	6
MP7266.16	Cutting direction of the tool	DIRECT	7
MP7266.17	Additional information for PLC (Module 9093)	PLC	9
MP7266.18	Tool offset: length	TT: LOFFS	11
MP7266.19	Tool offset: radius	TT: ROFFS	11
MP7266.20	Breakage tolerance for tool length	LBREAK	6
MP7266.21	Breakage tolerance for tool radius	RBREAK	6
MP7266.22	Tooth length	LCUTS	11
MP7266.23	Plunge angle	ANGLE	7

MP	Meaning	Column name	Column width
MP7266.24	Tool type ■ BOR: Boring tool ■ BCKBOR: Back-boring tool ■ CENT: NC spot drill / center drill ■ CSINK: Countersinking tool ■ DRILL: Drilling tool ■ MILL: Milling cutter ■ MILL_R: Rough cutter ■ MILL_F: Finishing cutter ■ MILL_RF: Rough and finishing cutter ■ MILL_FD: Floor finishing cutter ■ MILL_FS: Side finishing cutter ■ MILL_FACE: Face-milling cutter ■ REAM: Reamer ■ TAP: Tapping tool ■ GF: Thread miller ■ GSF: Thread miller with chamfer ■ EP: Thread miller for single threads ■ WSP: Thread miller with indexable insert ■ BGF: Thriller ■ ZBGF: Circular thread miller ■ TSINK: Piloted counterbore	TYPE	10

MP	Meaning	Column name	Column width
MP7266.25	Tool material	TMAT	16
MP7266.26	Cutting data table	CDT	16
MP7266.27	PLC value	PLC-VAL	11
MP7266.28	Probe center misalignment in ref. axis	CAL-OF1	11
MP7266.29	Probe center misalignment in minor axis	CAL-OF2	11
MP7266.30	Spindle angle during calibration	CAL-ANG	8
MP7266.31	Tool type for pocket table	PTYP	2
MP7266.32	Maximum shaft speed [rpm]	NMAX	6
MP7266.33	Retract tool	LIFTOFF	1
MP7266.34	PLC value Input range: –99999.9999 to +99999.9999	P1	11
MP7266.35		P2	
MP7266.36		P3	
MP7266.37	Additional kinematics description for tool and tool carrier	KINEMATIC	16
MP7266.38	Point angle for DRILL and CSINK	T-ANGLE	9
MP7266.39	Thread pitch for TAP	PITCH	10
MP7266.40	Control strategy name for AFC (Adaptive Feed Control)	AFC	10
MP7266.41	Tool value or tool radius R2	R2TOL	6
MP7266.42	Compensation value table for 3DToolComp	DR2TABLE	16
MP7266.43	Time stamp during tool changing	LAST_USE	16

MP7267	**Elements of the pocket table**
Input:	0: No display 1 to 99: Position in the pocket table
MP7267.0	Tool number (T)
MP7267.1	Special tool (ST)
MP7267.2	Fixed pocket (F)
MP7267.3	Locked pocket (L)
MP7267.4	PLC status (PLC)
MP7267.5	Tool name (TNAME)
MP7267.6	Comment on the tool (DOC)
MP7267.7	Tool type for pocket table (PTYP)
MP7267.8	Value 1 (P1)
MP7267.9	Value 2 (P2)
MP7267.10	Value 3 (P3)
MP7267.11	Value 4 (P4)
MP7267.12	Value 5 (P5)
MP7267.13	Reserve pocket (RSV)
MP7267.14	Pocket above locked (LOCKED_ABOVE)
MP7267.15	Pocket below locked (LOCKED_BELOW)
MP7267.16	Pocket at left locked (LOCKED_LEFT)
MP7267.17	Pocket at right locked (LOCKED_RIGHT)
MP7267.18	S1 value (P6)
MP7267.19	S2 value (P7)

Note

The **TNAME** (tool name) column contains the name of the tool from the tool table and therefore cannot be edited. For indexed tools, the name of the tool is entered with the index 0.

Module 9092 Search for an entry in the tables selected for execution (.T/.D/.TCH)

Prerequisite for table: M status must be set.

The entry or value sought is given as a natural number, shifted by the number of decimal places that can be entered.

As return code the function replies with the number of the line in which the value was found.

It is possible, for example, to look for the vacant pocket (corresponds to T0) in the pocket table.

If you wish to look for more occurrences of the same value, you must enter the line number of the last occurrence plus one as the starting line.

Call:

PS	B/W/D/K	<File type> 0: *.T file (tool table) 1: *.D file (datum table) 2: *.TCH file (pocket table)
PS	B/W/D/K	<Element value>
PS	B/W/D/K	<Element number> ***.T file** 0: Tool length (L) 1: Tool radius (R) 2: Reserved 3: Replacement tool (RT); (–1= not defined) 4: Reserved 5: TIME 1 6: TIME 2 7: CURRENT TIME 8: Tool radius 2 (R2) 9: Oversize for tool length (DL) 10: Oversize for tool radius (DR) 11: Oversize for tool radius 2 (DR2) 12: Tool locked (TL); (0: No, 1: Yes) 13: Number of the tool teeth (CUT) 14: Wear tolerance for tool length (LTOL) 15: Wear tolerance for tool radius (RTOL) 16: Cutting direction of the tool (DIRECT); (0:+; 1: –) 17: PLC status (PLC) 18: Tool offset for tool length (TT:LOFFS) 19: Tool offset for radius (TT:ROFFS); ($7FFF FFFF = R) 20: Breakage tolerance for tool length (LBREAK) 21: Breakage tolerance for tool radius (RBREAK) 22: Tooth length (LCUTS) 23: Plunge angle (ANGLE) 24: Tool number 25: Tool index 26: PLC value (PLC-VAL) 27: Probe center offset in reference axis (CAL-OF1) 28: Probe center offset in minor axis (CAL-OF1) 29: Spindle angle during calibration (CAL-ANG) 30: Tool type for pocket table (PTYP) 31: Maximum shaft speed [rpm] (NMAX) 32: Retract tool (LIFTOFF) 33: Value for PLC (P1)

34: Value for PLC (P2)
35: Value for PLC (P3)
36: Point angle for tool types DRILL and CSINK (T-ANGLE)
37: Thread pitch for tool type TAP (PITCH)
38: Wear tolerance for tool radius 2 (R2TOL)

***.D file:**

0: Shift in axis 1 ($7FFF FFFF = –)
1: Shift in axis 2 ($7FFF FFFF = –)
2: Shift in axis 3 ($7FFF FFFF = –)
3: Shift in axis 4 ($7FFF FFFF = –)
4: Shift in axis 5 ($7FFF FFFF = –)
5: Shift in axis 6 ($7FFF FFFF = –)
6: Shift in axis 7 ($7FFF FFFF = –)
7: Shift in axis 8 ($7FFF FFFF = –)
8: Shift in axis 9 ($7FFF FFFF = –)

***.TCH file:**

0: Tool number (T);
(–1, if no tool is entered)
1: Special tool (ST);
(0: no, 1 = yes)
2: Fixed pocket (F);
(0: no, 1 = yes)
3: Locked pocket (L);
(0: no, 1 = yes)
4: PLC status (PLC)
5: Tool type for pocket table (PTYP)
6: Reserve pocket (RSV)
7: Value 1 (P1)
8: Value 2 (P2)
9: Value 3 (P3)
10: Value 4 (P4)
11: Value 5 (P5)

PS B/W/D/K <Line number for beginning of search>
CM 9092
PL B/W/D <Line number (in case of error –1)>
PL B/W/D <Error number>
0: No error. Element was found.
1: Call was not in a submit or spawn job
2: File type does not exist
3: No file of the entered type was found with M status
4: Line number not in file
5: Incorrect element number
6: Element value not found

Error recognition:

Marker	Value	Meaning
M4203	0	No errors
	1	See above for errors

Module 9093 Read data from tables selected for program run (.T/.D/.TCH)

Prerequisite for table: M status must be set.

You transfer the line number (i.e. tool number for *.T, vector number for *.D or pocket number for *.TCH) and the number of the element to be read.

The value is given as a natural number, shifted by the number of decimal places that can be entered.

The module must be called in a submit job or spawn job.

Call:

```
PS    B/W/D/K   <File type (see Module 9092)>
PS    B/W/D/K   <Line number>
PS    B/W/D/K   <Element number (see Module 9092)>
CM    9093
PS    B/W/D     <Element value>
PL    B/W/D     <Error number>
                0: No error
                1: Call was not in a submit job
                2: File type does not exist
                3: No file of the entered type was found with M status
                4: Line number not in file
                5: Incorrect element number
```

Error recognition:

Marker	Value	Meaning
M4203	0	No errors
	1	See above for errors

Module 9094 Write data into a tool and datum table

Prerequisite for table: M status must be set.

You transfer the line number and the element number of the element to be overwritten.

The value is given as a natural number, shifted by the number of decimal places that can be entered.

The execution of Module 9094 reinitializes the geometry.

The module must be called in a submit job or spawn job.

```
Call:
PS    B/W/D/K   <File type (see Module 9092)>
PS    B/W/D/K   <Line number>
PS    B/W/D/K   <Element number (see Module 9092)>
PS    B/W/D/K   <Element value>
CM    9094
PL    B/W/D     <Error number>
                0: No error. Element was written.
                1: Call was not in a submit or spawn job
                2: File type does not exist
                3: No file of the entered type was found with M status
                4: Line number not in file
                5: Incorrect element number
                6: Element value is outside the permissible range
```

Error recognition:

Marker	Value	Meaning
M4203	0	No errors
	1	See above for errors

Module 9096 Delete a line from the tool table

You remove a line from the tool table and cancel any link with a replacement tool.

The module must be called in a submit job or spawn job.

Call:

PS	B/W/D/K	<Tool number / pocket number>
PS	B/W/D/K	<Mode>
		Bit 0: Delete entries in pocket table
		0: Pocket table remains unchanged
		1: Tool number in pocket table is deleted
		Bit 1: Tool or pocket number
		0: Transferred value = tool number
		1: Transferred value = pocket number
CM	9096	

Error recognition:

Marker	Value	Meaning
M4203	0	Line was deleted
	1	Error code in W1022
W1022	2	Invalid pocket or tool number
	21	Module was not called in a spawn or submit job
	24	File error

Module 9350 Read data from the tool table

Module 9350 reads the contents of a cell in the tool table with the status **M.** The value is read as an integer value.

Call:

PS	B/W/D/K	<Tool number>
PS	B/W/D/K	<Tool index>
		≤ 0: Main entry
PS	B/W/D/K	<Element number (see Module 9092)>
CM	9350	
PL	B/W/D	<Element value>
PL	B/W/D	<Error number>
		0: No error, element value was read
		1: Module was not called in a spawn or submit job
		2: File type does not exist
		3: No tool table with status **M**
		4: Line number does not exist
		5: Incorrect element number

Error recognition:

Marker	Value	Meaning
M4203	0	Element value read
	1	Error code in W1022
W1022	2	Incorrect element number
	7	Line number does not exist
	20	Module was not called in a spawn job or submit job
	36	No tool table with status M

Module 9351 Write data to tool table

Module 9351 writes the contents of a cell in the tool table with the status `M`. The value must be given as an integer value.

Call:
PS B/W/D/K <Tool number>
PS B/W/D/K <Tool index>
–1: Write all indexes of a tool
PS B/W/D/K <Element number>
See Module 9350
PS B/W/D/K <Element value>
CM 9351
PL B/W/D <Error number>
0: No error, element value was written
1: Module was not called in a spawn or submit job
2: File type does not exist
3: No tool table with status `M`
4: Line number does not exist
5: Incorrect element number
6: Element value is out of range
7: Error while writing to the file

Error recognition:

Marker	Value	Meaning
M4203	0	Element value written
	1	Error code in W1022
W1022	2	Incorrect element number
	7	Line number does not exist
	20	Module was not called in a spawn job or submit job
	36	No tool table with status M

Definition of the tool magazine using magazine rules

You can usually place more than one tool type in a tool magazine. Depending on the tool, however, surrounding pockets may have to be locked. The ASCII file *.TCR contains magazine rules for such definitions.

- In OEM.SYS, use the keyword **TCHRULES =** to enter the name and path of the ASCII file *.TCR.

or

- Use Module 9343 to compile the ASCII file with the magazine rules.
- Create the file *.TCR with the following keywords.

Keyword	Meaning
[magazine]a	All the following rules apply to the tool magazine **a.** Example: **[magazine]4**
[search]a = a b ...	Definition of the search sequence for the tool type **a.** The tool type named first (here **a**) is searched for first, then the next tool type (separated by a space) (here **b**). You may only enter tool types in whose pockets tools of the type **a** may be placed! Example: **[search]1 = 1 2**
[tooltype]a	All the following rules apply to the tool type **a.** 99 tool types can be specified. Example: **[tooltype]2**
[place]a = bx cx ...	Description of the pocket **a**. Define the pockets (here **b** and **c**) that are affected by pocket **a** when the current tool type (keyword **[tooltype]a**) is placed there. Immediately after the pocket number the identifier **x** follows, indicating which area of the pocket is affected (**r** = right area, **l** = left area, **b** = bottom area, **a** = top area) Example: [place]21 = 20r 22l

Note

As of NC software 340 422-03 and 340 480-03, you must define the search sequence for each magazine separately, not just globally at the beginning of the *.TCR file. If the search sequence is to be defined the same for each magazine, then it must be repeated after each magazine definition.

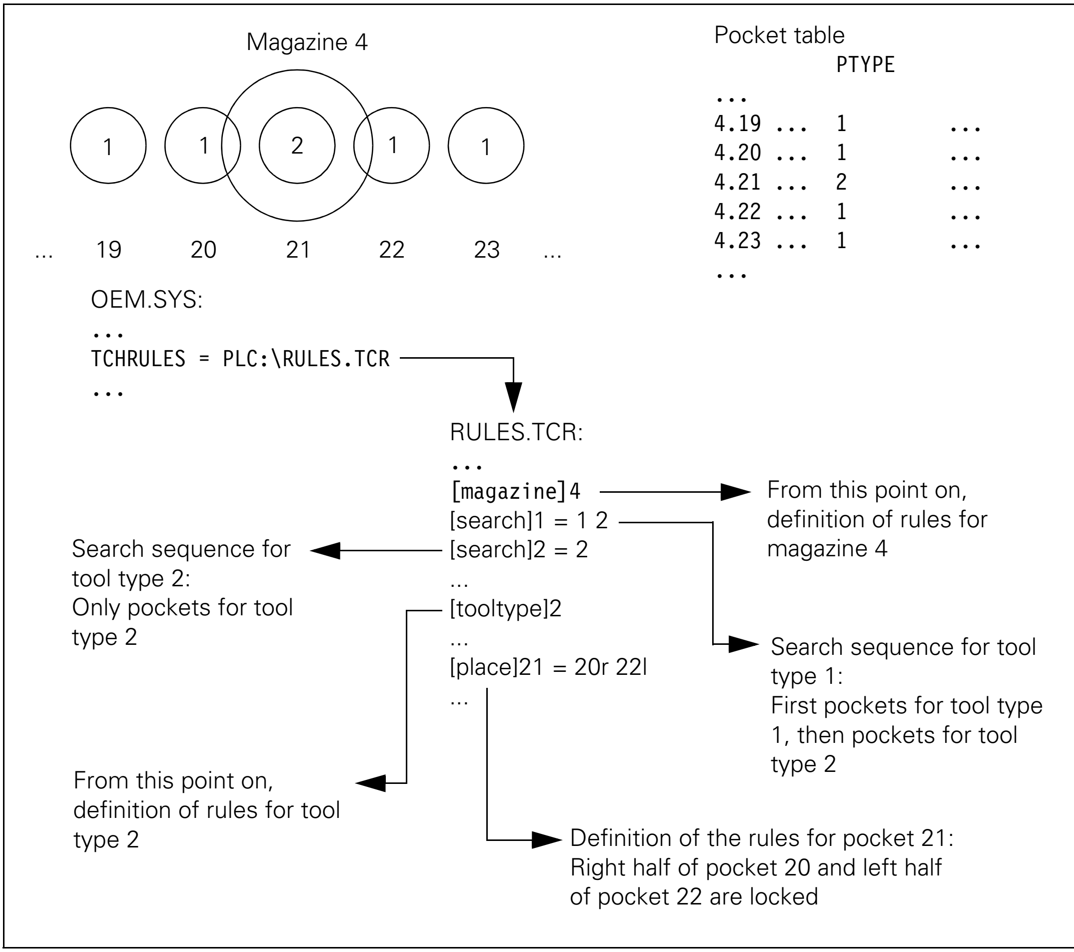
The following columns in the pocket table are used for magazine rules:

- **PTYP:** Tool type
- **RSV:** Reserved pocket
- **P1** to **P5:** Values 1 to 5 for evaluation in the PLC (e.g., axis positions of a pocket in the box magazine). Module 9304 copies the values to a word memory.
- **LOCKED_ABOVE:** Pocket above is locked
- **LOCKED_BELOW:** Pocket below is locked
- **LOCKED_LEFT:** Pocket to the left is locked
- **LOCKED_RIGHT:** Pocket to the right is locked

With **FN18: SYSREAD ID51** the cells of the pocket table can be read.

Module 9340 searches a magazine for vacant, reserved or unavailable pockets.
Module 9342 uses a tool number to determine the magazine number and pocket number.
Module 9341 processes the pockets depending on the magazine rules. Pockets can be reserved, released and made unavailable.
Module 9216 is used to display a selection list in a pop-up window for placing tools into magazines and for removing them. The selection list is created by the NC at run time, and contains tools with and without pocket assignment as well as empty pockets. The user selects an entry from the selection list with the arrow keys, and Module 9216 reports the selection to the PLC for further processing.

Example for a description of the tool magazine using magazine rules:



Module 9216 Pop-up window with tool selection list

Module 9216 opens a pop-up window in which the arrow keys are used to make a selection for the tool management. The selection list is created by the NC at the run time for the module. The module responds with the tool or pocket number for further processing.

- Entries using the iTNC keyboard are registered by the pop-up window.
- The pop-up window is only shown in the machining modes.
- If another pop-up window is active, this window is placed in the background. After the pop-up window with the selection list is closed, this other pop-up window is returned to the foreground.
- If the pop-up window with the selection list is active, and another pop-up window is opened, any keystrokes on the iTNC keyboard will be registered by the second pop-up window, not by the selection list.
- The module should be called in its own spawn process, since the module does not return until the pop-up window is closed, and would therefore block all subsequent submit jobs.
- The window title is displayed in the language set in MP7230.0.

Call:

PS	B/W/D/K	<Mode> 0: Tools in tool table not in the magazine 1: Tools in tool table in the magazine 2: Empty pockets in the magazine 3: Tools with reserved pockets in the magazine
PS	B/W/D/K	<Magazine number> Only for mode 2
PS	B/W/D/K	<Tool number> Only for mode 2 (determining the tool type)
CM	9216	
PL	B/W/D	<Tool or pocket number> –1: Error code in W1022 –2: General error –3: Selection list closed without selection –4: Menu file not available –5: Pop-up window cannot be opened –6: Selection window already active –7: Selection window not available –8: Menu file without selection list

Error recognition:

Marker	Value	Meaning
M4203	0	Selection complete
	1	Error code in W1022
W1022	1	Line in the pocket table could not be found
	2	Invalid magazine number
	3	Invalid mode
	4	Invalid tool number or type
	6	Tool number is already contained in the pocket table
	20	Module was not called in a spawn job or submit job
	36	File error in the tool or pocket table
	45	Module execution canceled, see return value for error

Module 9304 Copying columns P1 to P5 to the pocket table
Module 9304 transfers the contents of columns P1 to P5 from the pocket table to the defined double-word address.

Call:
PS B/W/D/K <Magazine number>
PS B/W/D/K <Pocket number>
PS B/W/D/K <Double-word address>
CM 9304

Error recognition:

Marker	Value	Meaning
M4203	0	Columns copied
	1	Error code in W1022
W1022	1	Invalid pocket number
	2	Invalid magazine number
	4	Invalid double-word address
	20	Module was not called in a spawn job or submit job
	36	File error in pocket table

Module 9340 Searching for a pocket depending on magazine rules

Module 9340 searches a tool magazine for vacant, locked or unavailable pockets. The search for free pockets is according to the magazine rules.

Call:

```
PS    B/W/D/K   <Magazine number>
PS    B/W/D/K   <Pocket number for starting the search>
PS    B/W/D/K   <Tool number or type>
PS    B/W/D/K   <Mode>
                Bit 0=0: Transfer tool number
                Bit 0=0: Transfer tool type
                Bit 1=1: Search for a vacant pocket
                   (depending on magazine rules)
                Bit 2=1: Search for a reserved pocket
                Bit 3=1: Search for an unavailable pocket
CM    9340
PL    B/W/D/K   <Pocket number>
                –1: Error code in W1022
                –2: No free pocket or tool not found
```

Error recognition:

Marker	Value	Meaning
M4203	0	Search for pocket completed
	1	Error code in W1022
W1022	1	Invalid pocket number
	2	Invalid magazine number
	3	Invalid mode
	4	Invalid tool number or type
	20	Module was not called in a spawn job or submit job
	36	File error in the tool or pocket table
	45	Module execution canceled, see return value for error
	55	Pocket table could not be locked

Module 9341 Editing a pocket table depending on magazine rules
Module 9341 reserves, releases, or makes pockets unavailable in the pocket table, in accordance with the magazine rules.
The module affects the columns **RSV, LOCKED_ABOVE, LOCKED_BELOW, LOCKED_LEFT,** and **LOCKED_RIGHT.** Therefore these columns may not be changed manually nor by the PLC program.

Call:
PS B/W/D/K <Magazine number>
PS B/W/D/K <Pocket number>
PS B/W/D/K <Tool number>
PS B/W/D/K <Mode>
0: Release pocket (depending on magazine and tool number)
1: Release pocket (depending on magazine and pocket number)
2: Reserve pocket (depending on magazine, pocket and tool number)
3: Make pocket unavailable (depending on magazine and pocket number)
4: Reserve pocket if previously unavailable (depending on magazine and pocket number)
CM 9341

Error recognition:

Marker	Value	Meaning
M4203	0	Pocket table edited
	1	Error code in W1022
W1022	1	Invalid pocket number
	2	Invalid magazine number
	3	Invalid mode
	4	Invalid tool number
	6	Reservation not possible
	7	Magazine rules not compiled or not present
	20	Module was not called in a spawn job or submit job
	36	File error in pocket table
	45	Module cancellation, error evaluation using return value
	55	Pocket table could not be locked

Module 9342 Find magazine and pocket number

Module 9342 determines the magazine and pocket number from the tool number. The module takes the **RSV** column of the pocket table into account if magazine rules are in effect. If the module is used to find reserved pockets, it returns the first reserved pocket with ascending magazine number. However, further pockets can be reserved. In this case the search must be repeated with another "start magazine for the search."

Call:

PS	B/W/D/K	<Tool number>
PS	B/W/D/K	<Mode> 0: Look for occupied pocket 1: Look for reserved pocket
PS	B/W/D/K	<Start magazine for the search>
CM	9342	
PL	B/W/D/K	<Magazine number> –1: Magazine could not be found
PL	B/W/D/K	<Pocket number> –1: Pocket could not be found

Error recognition:

Marker	Value	Meaning
M4203	0	Magazine and pocket number found
	1	Error code in W1022
W1022	1	Invalid mode
	2	Invalid start magazine for the search
	20	Module was not called in a spawn or submit job
	30	Tool not found
	36	File error in pocket table

Module 9343 Compilation and activation of magazine rules

Module 9343 is used to compile and activate magazine rules (*.TCR), independent of the entry **TCHRULES =** in OEM.SYS. If the entry exists in OEM.SYS, the magazine rules are overwritten when Module 9343 is called. If an error occurs during compilation, the PLC program is stopped. The magazine rules must be activated during the first run of the PLC program or before the first call of Modules 934x.

Call:
PS B/W/D/K/S<Path and file name of the magazine rules>
CM 9343

Error recognition:

Marker	Value	Meaning
M4203	0	Magazine rules have been compiled and activated
	1	Error code in W1022
W1022	11	Invalid string programmed
	20	Module was not called in a spawn job or submit job
	38	Error during compilation

Pocket exchange in the pocket table

To switch the pockets of two tools in the pocket table:

- Lock the pocket table with Module 9300.
- Switch the pockets with Module 9305.
- Release the pocket table with Module 9300.

Module 9300 Locking/releasing the pocket table

Module 9300 locks the pocket table for pocket switching with Modules 9305, 9306 and also 9301, 9302 or 934x, and then releases it again. The module can also be called while an NC program is running.

Call:

PS	B/W/D/K	<Lock/release pocket table> 0: Release the pocket table 1: Lock the pocket table
CM	9300	
PL	B/W/D	<Error> 0: Pocket table locked/released 1: Pocket table could not be locked 2: Pocket table could not be released 3: Transfer parameter invalid 4: Module was not called in a spawn job or submit job 5: Not used 6: Code 0: Pocket table already released Code 1: Pocket table already locked

Error recognition:

Marker	Value	Meaning
M4203	0	Pocket table locked/released
	1	Error code in W1022
W1022	2	Invalid parameter for locking/releasing the pocket table
	6	Pocket table was already locked/released
	20	Module was not called in a spawn or submit job
	21	Module was called during an NC program run

Module 9305 Tool exchange in the pocket table

Module 9305 is used to exchange the tools in the pocket table. The module can be called while an NC program is running. Only column T (tool number) is changed. All other columns remain unchanged. The pocket table must be locked with Module 9300 before switching the pockets, and then it must be released again.

Call:

```
PS      B/W/D/K   <Original pocket>
PS      B/W/D/K   <New pocket>
CM      9305
```

Error recognition:

Marker	Value	Meaning
M4203	0	Pocket has been exchanged
	1	Error code in W1022
W1022	2	Invalid parameter
	6	Magazine management using magazine rules is active
	20	Module was not called in a spawn job or submit job
	21	Module was called during an NC program run without any locking in place
	30	No valid tool in the original pocket

Managing multiple tool magazines

Up to eight different tool magazines can be managed in the pocket table. In the pocket table the tool magazines are listed from 1 to 8, i.e., tool magazine 1 with tool 1 to <MP7261.0> is in first position. Immediately thereafter, tool magazine 2 appears with tool 1 to <MP7261.1>, then tool magazine 3, etc.

- Enter the number of pockets in tool magazines 1 to 8 in MP7261.0 to MP7261.7.

The current tool magazine number is saved in W268.
Module 9302 searches for an open pocket in a tool magazine, and Module 9306 switches tools between the tool magazines.

Module 9301 determines the number of the entry in the pocket table. The number of the entry depends on the tool magazine and pocket numbers.

- Enter this number in the modules which cannot accept tool magazine numbers (e.g. Modules 9092, 9093, 9094).

		Set	Reset
W268	**Tool magazine number** **–1: External tool** **0: Tool in the spindle** **1 to 8: Number of the tool magazine**	NC	NC

Search sequence of the tool magazines

Sequence for searching in tool magazines for the tool to be called. To call a tool through tool names, it is possible in machine parameter 7484.0 to .7 to specify a sequence (index from MP7261, or -1 cancel) for searching in tool magazines. If several tools are available in the magazine and a tool usage list is available, the tool with the shortest sufficient service life is chosen. If the tool is not found in any given magazine, the first tool with sufficient service life from the TOOL.T file is used.

MP7484	**Search sequence in tool magazines**
Input:	0 to 7 [index from MP7261] –1 = Cancel

Module 9301 Find the number of an entry in the pocket table

Module 9301 determines the number of an entry in the pocket table. This number is necessary for the modules in which no tool magazine numbers can be entered.

Call:

```
PS    B/W/D/K   <Tool magazine number>
PS    B/W/D/K   <Pocket number>
CM    9301
PL    B/W/D     <Number of the entry in the pocket table>
                –1: M4203 = 1
```

Error recognition:

Marker	Value	Meaning
M4203	0	Number of the entry was found
	1	Error code in W1022
W1022	1	Invalid tool magazine number
	2	Invalid pocket number
	20	Module was not called in a spawn job or submit job

Module 9302 Search for a vacant pocket in the tool magazine

Module 9302 searches for a vacant pocket in a tool magazine.

Call:

```
PS    B/W/D/K   <Tool magazine number>
PS    B/W/D/K   <Pocket at which the search is to be started>
CM    9302
PL    B/W/D     <Number of the vacant pocket>
                –1: No vacant pocket available
```

Error recognition:

Marker	Value	Meaning
M4203	0	Search completed
	1	Error code in W1022
W1022	1	Invalid pocket number
	2	Invalid tool magazine number
	20	Module was not called in a spawn or submit job
	36	Error in file handling

Module 9306 Exchange tools between tool magazines

With Module 9306, tools are exchanged between tool magazines. The pocket table must be locked with Module 9300 before calling this module, and then it must be released again. In the original and new entry only the tool number is changed. Pocket-specific data remains unchanged. The module must be called at standstill or during a strobe output. However, it can be called while an NC program is running.

Call:

```
PS      B/W/D/K   <Original tool magazine>
PS      B/W/D/K   <Original pocket>
PS      B/W/D/K   <New tool magazine>
PS      B/W/D/K   <New pocket>
CM      9306
```

Error recognition:

Marker	Value	Meaning
M4203	0	Pockets exchanged successfully
	1	Error code in W1022
W1022	1	Invalid pocket number
	2	Invalid tool magazine number
	6	Magazine management using magazine rules is active
	20	Module was not called in a spawn job or submit job
	21	Module was called during an NC program run without any locking in place
	30	No valid tool in the original pocket
	36	Error in file handling

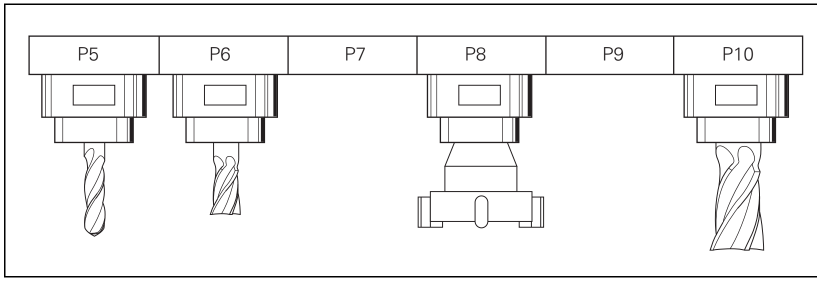
Special tools

In the pocket table:

- In the column **ST** you define tools as special tools.

For oversized special tools:

- Leave a pocket free in the tool magazine on both sides of the pocket (see illustration).
- In the column **L** you lock pockets that are to remain empty.
- With M4541, block the variable tool-pocket coding for special tools.



As soon as M4541 is set, all special tools are returned to their original pocket in spite of the "variable tool-pocket coding" function.

With the column **F** (fixed pocket) you can define this function selectively for individual tools.

		Set	Reset
M4541	**Special tool in original pocket in spite of variable pocket coding**	PLC	PLC

Tool life, replacement tool

You can enter two tool life values (**TIME1** and **TIME2**) and one replacement tool (**RT**) for each tool in the tool table.

For the **TOOL CALL:**

- **CUR.TIME** (current tool age) > **TIME2:** Pocket or tool number (MP7480) of the replacement tool and a T strobe M4073 are output and M4525 is set.
- **CUR.TIME** (current tool age) > **TIME2** > 0 and no replacement tool is defined: After expiration of the time, the error message **Max. tool age expired** is displayed for this tool, and M4546 and M4525 are set.
- **CUR.TIME** (current tool age) > **TIME1:** The NC sets M4543 and M4525.

You decide in the PLC what should happen when M4543 or M4546 is set (e.g. display a PLC error message).

With M101, you activate the automatic insertion of the replacement tool after expiration of the tool life (**TIME1** or **TIME2**). With M102, you deactivate the insertion. The tool is not changed immediately after expiration of the tool life, but rather it varies depending on the processor load. The tool change is transmitted delayed by at least one block and by no more than one minute.
In order to also be able to activate the automatic insertion of the replacement tool with TCPM, you must program a retraction with M140 in the tool change macro. After the tool change, the tool moves with an approach logic to the pre-compensated position and then returns to the contour.
To synchronize the current machine status and the look-ahead calculation with an NC macro call, see "NCMACRO.SYS" on page 1644.

> **Note**
>
> In standard NC programs (NC block with **RR, RL** or **R0**), the same radius must be defined for the replacement tool as for the original tool.

No radius compensation is given in NC blocks with surface-normal vectors.
One delta value for tool length and radius **(DR** and **DL)** can be entered for each tool in the tool table. These delta values are taken into account by the iTNC.

If the radius of the replacement tool differs from the original tool, you must define this in the **DR** column. The delta value must always be negative. If you enter a positive delta value, the error message **Tool radius too large** appears.
You can suppress this error message with the M function M107, and reactivate it with M108.

You can select whether the tool length is given with respect to the south pole or the ball center of a spherical cutter:

- With MP7680, select whether the tool radius **(R2)** should be taken into account for the calculation of the tool length.

The current tool age is calculated in the **Program Run, Single Block** and **Program Run, Full Sequence** operating modes if the following conditions are fulfilled.

- Spindle ON
- No F MAX
- F enable
- Control-in-operation symbol is on

After program interruption with "internal stop," M02, M30 or **END PGM,** the tool age counter is stopped.

The tool age counter does not run in the **Manual Operation**, **Electronic Handwheel** and **Positioning with MDI** operating modes.

The user can reset the current tool age by entering zero.

		Set	Reset
M4543	**Tool life 1 expired (TIME1 in the tool table)**	NC	NC/PLC
M4546	**Tool life 2 expired (TIME2 in the tool table)**	NC	NC/PLC

MP7680 **Machine parameter with multiple function**

Input: Bit 6 – Tool length in blocks with normal vectors:
0: Without R2 from tool table (south pole)
1: With R2 from tool table (center of sphere)

Indexed tools

You can also work with indexed tools in the tool table, e.g., when you use a stepped drill with more than one length compensation value. For indexed tools, the tool number is given an index (e.g. 1.1).

- In MP7262, enter the maximum tool index number.

The index number of the programmed tool is saved in W266.

If you are working with indexed tools and wish to use Modules 9092, 9093 or 9094, you must first find the line number of the tool, since these modules will need it. As an alternative you can use Modules 9350 and 9351; the tool number and tool index can be transferred in these modules.

- Use Module 9091 to determine the line number of a tool in the tool table.

MP7262 **Maximum tool index number for indexed tools**
Input: 0 to 9
MP7262 can also be overwritten by the PLC and the LSV2 protocol.

		Set	Reset
W266	**Index number of a programmed indexed tool**	NC	NC

Module 9091 Find the line number of a tool in the tool table

Call:

```
PS   B/W/D/K   <Tool number>
PS   B/W/D/K   <Tool index>
CM   9091
PL   B/W/D     <Line number>
```

Error recognition:

Marker	Value	Meaning
M4203	0	Line number was found
	1	Error code in W1022
W1022	2	Invalid value for tool number or tool index number
	20	Module was not called in a spawn or submit job
	29	Tool table (TOOL.T) not found
	30	Tool number not found
	32	Tool index number not found

8.15.2 Tool-usage test

When testing an NC program in the **Test Run** operating mode (calculate machining time: active) or via an LSV2 command, a tool-usage file (*.T.DEP) can be created automatically. It contains all required tools (number, index, name, radius), the machining times (at 100% override) and their program calls.

- Enable the function for generating the tool-usage file with MP7246 bit 2=1.

In the **Program Run, Single Block** and **Program Run, Full Sequence** operating modes, press the **TOOL USAGE TEST** soft key to compare the data in the tool usage file with the data in the tool table. If the tool-usage file is not current or does not exist, the error message **Generate tool usage file!** appears. Otherwise a popup window with the results of the comparison appears. The value entered for **TIME2** in the tool table must be at least 10% greater than the time required.

In MP7485 a percentage value (0 to 100, default 10) can be entered for application to the usage time found in the tool usage list. This makes it possible to influence the selection if the selection of a tool depends on its usage time. This machine parameter is also effective with the TOOL USAGE TEST soft key in the Program Run, Full Sequence operating mode. Up to now, a constant value of 10% was applied.

If MP7246 bit 4 is set, the calculation of the tool usage times is written to an ASCII file. This file contains the run time and the absolute end time of each NC block listed by NC block numbers. The name of the ASCII file is derived from the name of the NC program (e.g. NCPROG.H) and has the extension *.POS.DEP (e.g. NCPROG.H.POS.DEP).

MP7246	**Machine parameter with multiple function**
Input:	Bit 2 – Tool usage file 0: Do not generate 1: Generate Bit 4 – ASCII file for machining time per NC block 0: Do not create ASCII file for machining time per NC block 1: Create ASCII file for machining time per NC block
MP7485	**Add usage time**
Input:	0 to 100 [%] Default setting: 10

Tool usage test for pallet tables

The **TOOL USAGE TEST** soft key is also available for pallet tables.

- If a line with an NC program is active, the test is performed only for the NC program in question. The tool-usage file must first have been created in the **Test Run** operating mode.
- If a line with a pallet entry is active, the test is performed for the complete pallet table. The tool-usage files of the NC programs called must first have been created in the **Test Run** operating mode. One tool-usage file is created for the entire pallet table.
- With Module 9282, the tool-usage test for a pallet table can be performed by the PLC. One tool-usage file is created for the entire pallet table.

Module 9282 Tool usage test for pallet table

Module 9282 allows you to check the tools used in a pallet table. The pallet file must be selected in the **Program Run, Single Block** or **Program Run, Full Sequence** operating mode. The tool usage file for the pallet table and the test result file in ASCII format are created. The test result file contains the results from the comparison of the tool usage file with the tool table.
If a pallet call is given as the line number, all subordinate machining operations are checked. If a program call is given, only the tool usage file of the NC program is checked.

Call:
PS B/W/D/K <Line number of the pallet table>
PS B/W/D/K/S<Name of the test result file (*.P.T.DEP)>
PS B/W/D/K/S<Name of the tool usage file (*.T.DEP)>
CM 9282
PL B/W/D <Result>
0: No error
1: Tool usage file of an NC program of the pallet table not available or no longer up-to-date
2: Tool life not sufficient
3: Required tool not available
4: Radius of required tool is incorrect
5: Tool is not in magazine
6: Tool usage file of an NC program of pallet table cannot be opened
7: Test result file could not be created
8: Test result file cannot be written to
9: Tool usage file of an NC program of pallet table cannot be read
10: No memory for creating the tool usage file

Error recognition:

Marker	Value	Meaning
M4203	0	Tool usage test has been performed
	1	Error code in W1022
W1022	11	Invalid string programmed
	20	Module was not called in a spawn job or submit job

Structure of the tool-usage file

The columns of the tool-usage file have the following meanings:

Column	Description
NR	Line number
T	Tool number and index combined. (e.g. '14.4' for tool 14 with index 4).
TOKEN	**TOOL**: Usage time per **TOOL CALL;** the entries are listed in chronological order **TTOTAL**: Total usage time of a tool **STOTAL**: Call of a subprogram (including cycles). The entries are listed in chronological order.
TNR	Tool number (–1: no tool inserted yet)
IDX	Tool index
NAME	Tool name
TIME	Tool-usage time in seconds Time duration that the tool was used to machine a workpiece (spindle on and not at rapid traverse!)
WTIME	Tool-selection time in seconds Time duration that a tool was selected during the program. It also includes the times where the tool was moved without M3/ M4, as well as times where it was moved at **FMAX.**
RAD	**Tool radius R** + **Oversize for tool radius DR;** Data from the tool table in 1/10 µm
BLOCK	Line number with **TOOL CALL**
PATH	**Token** = **TOOL:** Active program or sub program **Token** = **STOTAL:** File name of the subprogram or cycle

Line	Description
TIMETOTAL	Sum of all entries in the **TIME** and **WTIME** columns. In the other columns this line contains the value 0.
TOOLFILE	The **PATH** column contains the tool table that was active when the tool-usage file was created.

Example of a tool-usage file:

```
NR    TOKENTNRIDXNAMETIMERADBLOCKPATH
0     TOOL+2 +0MILLER2+144+60000+4TNC:\Stefan\Test\TOOLING.H
1     TOOL+3 +0DRILLER1+7+30000+9TNC:\Stefan\Test\TOOLING.H
2     TOOL+4 +0DRILLER2+100+40000+11TNC:\Stefan\Test\TOOLING.H
3     TOOL+2 +0MILLER2+1588+60000+14TNC:\Stefan\Test\TOOLING.H
4     TTOTAL+3+0DRILLER1+7+30000+0
5     TTOTAL+4+0DRILLER2+100+40000+0
6     TTOTAL+2+0MILLER2+1732+60000+0
7     STOTAL+0+0+0  +0+0   SYS:\JHCYC\NC\252.CYC
8     STOTAL+0+0+0  +0+0   SYS:\JHCYC\NC\INIT25X.CYC
9     STOTAL+0+0+0  +0+0   SYS:\JHCYC\NC\PSRD25X.CYC
10    STOTAL+0+0+0  +0+0   SYS:\JHCYC\NC\MILL25X.CYC
11    STOTAL+0+0+0  +0+0   TNC:\Stefan\Test\DRILL.H
12    STOTAL+0+0+0  +0+0   SYS:\JHCYC\NC\200.CYC
13    STOTAL+0+0+0  +0+0   TNC:\Stefan\Test\DRILL2.H
14    STOTAL+0+0+0  +0+0   SYS:\JHCYC\NC\205.CYC
15    STOTAL+0+0+0  +0+0   SYS:\JHCYC\NC\251.CYC
[END]
```

8.15.3 Automatic calculation of cutting data

The optimum spindle speed and the corresponding contouring feed rate is calculated from the values entered in cutting tables for the tool and workpiece material.

For cutters, the cutting data table lists the cutting speed and the feed rate per tooth. For drills it lists the feed rate per revolution.

$$S = \frac{v_c \cdot 1000}{d \cdot \pi}$$

S: Spindle speed [min^{-1}]

v_c: Cutting speed [m/min]

d: Tool diameter [mm]

Milling cutter: $F = f_z \cdot S$

F: Feed rate [mm/min]

f_z: Feed rate per tooth [mm]

z: Number of teeth

Drill: $F = f_u \cdot S$

f_u: Feed rate per revolution [mm]

Tool table

- In the **CDT** column of the tool table, enter the name of the cutting data table that is to be used for that tool.
- In the TYP column define the type of tool:
  - **DRILL** = Drilling tool
  - **TAP** = Tapping tool
  - **MILL** = Milling cutter
- Enter the following values in the table:
  - Tool radius **R**
  - Tool material **TMAT**
  - For cutter: Number of teeth **CUT.**

The tool types are defined in the file PLC:\TTYP.TAB.

If you edit this file, you must use the command **TTYP =** to enter the new name and path in the system file OEM.SYS.

Cutting data table

The cutting data for specific tools are available from the tool manufacturer.

Cutting data tables have the file name extension .CDT.

Each line in the cutting data table contains the data for a specific combination of workpiece and tool material. For milling cutters you can enter up to four cutting speeds with the corresponding feed rates per teeth. In the tables of the manufacturers these data are specified for different infeeds and for climb and up-cut milling. For drills you enter a cutting speed with the corresponding feed rate per revolution.

A standard cutting data table is saved in the root directory of the iTNC (TNC:\).
You can add as many cutting data tables as desired.

If you change the standard cutting data table, you must copy the changed table to another path. Otherwise your changes will be overwritten with HEIDENHAIN standard data during the next software update:

- In the system file TNC.SYS, use the code word **PCDT =** to enter the path in which your cutting data tables are saved.

Material tables

The workpiece materials used are defined in the table WMAT.TAB, the tool materials in the table TMAT.TAB.

Standard tables are in the root directory of the TNC (TNC:\).

You can arbitrarily expand and change all tables.

If you change the tables, you must copy them to another path. Otherwise your changes will be overwritten with HEIDENHAIN standard data during the next software update:

- In the system file TNC.SYS, use the code words **TMAT =** and **WMAT =** to enter the path and file names of your tables.

In the material data tables:

- In the **Name** column, enter a brief name for the material (e.g. **HSS**).
- Enter additional information on the material in **DOC** column.

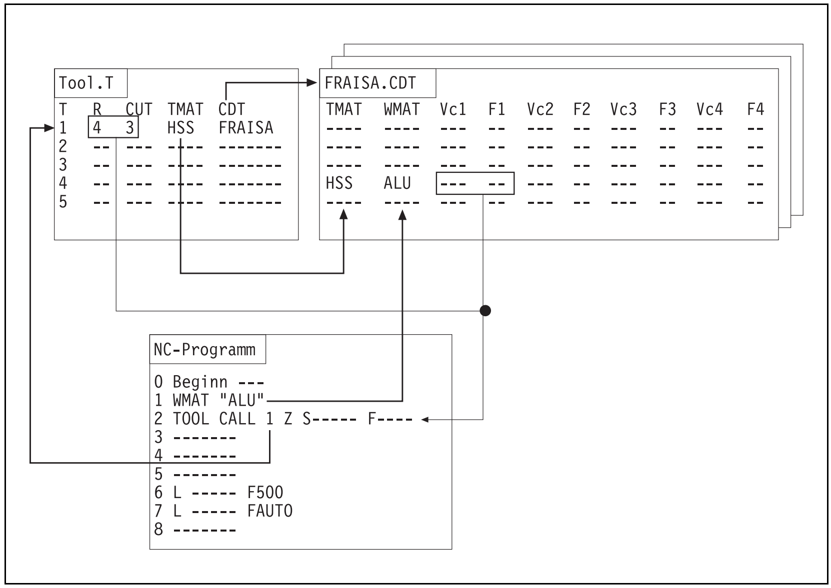
Calculation of cutting data

- Define the workpiece material in the NC program with the **WMAT** soft key.

The **TOOL CALL** block provides soft keys for automatic acceptance of various speeds (S1 to S4) and for the selection of the feed rate (F1 to F4).

If you enter the spindle speed manually, this value is taken into account in the calculation of the feed rate. You cannot, however, enter F for calculation of S. If you enter the feed rate manually, the entered value applies until you program another feed rate. With the **F AUTO** soft key you can again activate the feed rate from the TOOL CALL block.

Principle



8.15.4 Automatic tool recognition

Automatic tool identification is possible with the Balluff tool identification system (BIS).

Please contact HEIDENHAIN for further information.

8.15.5 Controlling the tool changer

You program the control of the tool changer in the PLC.
This includes:

- Positioning of the changing arm and carousel
- Tool change sequence

The NC handles the tool management. This includes:

- Tool life
- Pocket assignment
- Evaluation of the **TOOL DEF** blocks
- Evaluation of the **TOOL CALL** blocks

The NC and PLC communicate through markers and words.

For execution of the **TOOL CALL** block, the NC takes the tool geometry data from the tool table:

- Activate with M4538 the geometry of the tool defined in W264. With this marker you make sure that the current tool geometry is always active even if the tool change sequence is cancelled.
  CAUTION: Activate only together with an M/S/T/Q strobe or if M4176 = 0 (control-in-operation display is lit or blinking)!

With the **TOOL DEF** block you can pre-position the tool changer:

- After a tool has been changed, program the next tool with **TOOL DEF.**
- Evaluate the tool and pocket number and pre-position the tool changer to the follow-up tool via a PLC positioning.
- In MP7682 bit 6 you specify the behavior of the control when a **TOOL DEF** strobe is current.

Bit 6 = 1: The control always waits for an acknowledgment of the **TOOL DEF** strobe by the PLC before continuing with the NC program.

Bit 6 = 0: Continuation of the NC program depends on the NC program. If the **TOOL DEF** is within a contiguous contour, then an acknowledgment of the **TOOL DEF** strobe is **not** waited for. Depending on the type of tool changer, this behavior can lead to problems:
Example:

NC Program	Description
... L IX+50 F500 **... TOOL DEF 1** **... L IX+50 F500**	The **TOOL DEF** block is **within** a contiguous contour. The acknowledgment of the strobe is **not** waited for.
... **... TOOL DEF 1** **... L IX+50 F500**	The **TOOL DEF** block is **not** within a contiguous contour. The acknowledgment of the strobe **is** waited for.

MP7682 **Machine parameter with multiple function**
Input: Bit 6 – Behavior with TOOL DEF strobe
0: Depending on the NC program, the TOOL DEF strobe must be acknowledged by the PLC (TOOLDEF within a contiguous contour)
1: TOOL DEF strobe must always be acknowledged by the PLC

Calling an NC program with TOOL CALL

With the NC block TOOL CALL you can call an NC program of your own definition:

- With the keyword **`TC = <path name>\<file name>`** in the PLC:\NCMACRO.SYS file, define the name of the NC program to be called.

To synchronize the current machine status and the look-ahead calculation with an NC macro call, see "NCMACRO.SYS" on page 1644.
The tool geometry is not taken over then. You must program a TOOL CALL at another place to update the tool data.

NC functions that must be reset at the beginning of a tool-change macro:

The tool-change macro requires non-radius-compensated movements with M91 (coordinates refer to the machine datum) or M92 (coordinates refer to a position defined by the machine manufacturer):

- M103 (Reduce feed rate during plunging to factor F)
- M112 (Insert rounding radius between nontangential straight lines)
- M114 (Automatic correction of machine geometry when machining with tilting axes)
- M118 (Superimpose handwheel positioning during program run)
- M124 (Ignore points when machining non-radius compensated straight line blocks)
- M128 (Retain position of tool tip when positioning tilting axes) or **`FUNCTION TCPM`**
- Cycle19 **`(WORKING PLANE)`** or **`PLANE`**
- Possibly M126 (Permit zero crossover on 360° rotary axes), if rotary axes are moved and their traverse ranges permit multiple paths.
- Possibly M136 (Feed rate F in millimeters per spindle revolution), if feed rates other than **`FMAX`** are used.
- Possibly M144 (Compensating the machine’s kinematics configuration for ACTUAL/NOMINAL positions at end of block), if this function is activated via MP7502.
- Possibly Cycle 32 **`(TOLERANCE),`** if a certain tolerance (MP1096.x) is required in the tool-change macro.

Program example: Positioning to the tool change position:

- Preferably file the program in the PLC partition so that it cannot be changed by the end user.

The tool data in the current program are not active. They must be requested with **FN18: SYSREAD** (see "Data transfer NC > NC program (FN18: SYSREAD)" on page 1659).

- In the called program, enter a **TOOL CALL** so that the tool data becomes active and a T strobe is transferred to the PLC.

With **FN17: SYSWRITE** you can overwrite the software limit switch for the tool-change position. If you use **FN18: SYSREAD** to call the programmed position after the **TOOL CALL,** you can program a continuous positioning movement of the spindle from the tool magazine to the next position.

With **FN20: WAIT FOR** you can delay execution of the NC program until the entered condition is fulfilled. These conditions can be comparisons of a PLC variable with a constant (see "Interrogate PLC operands in the NC program (FN20: WAIT FOR)" on page 1674).

With **FN17: SYSWRITE ID420 NR0 IDX0 = 0,** all coordinate transformations (e.g. cycles 7, 8, 10, 11, 19) performed in the tool-change program become globally effective. Without this block, they remain locally effective (only in the tool-change program).

With **FN18: SYSREAD ID61 NR0 IDX<tool number>** you find the corresponding tool-change sequence. This information is reported to the PLC at the same time with the pocket and magazine number. You can also find this information with Module 9035.

To ensure that during a block scan the tool-change program is not run until the end of the scan, you must enter the instruction **NCMACRO = TC** in the MGROUPS.SYS file (also see "Returning to the contour" on page 1225). If no NC program is specified in the NCMACRO.SYS file, the **TOOL CALL** is executed without calling the tool-changing program.

For test purposes, the tool-change program can be called from the TNC partition. In this case, the program call is handled as **PGM CALL,** i.e. defined values such as Q parameters and feed rate remain globally effective.
If the tool-change program is called from the PLC partition, the tool-change program is handled as a cycle call, i.e. defined values remain only locally effective.

Note

In contrast to the TNC 426/TNC 430, the iTNC 530 handles the tool number and tool index as two separate parameters. Therefore, with the iTNC 530, the tool number must be read with **FN18: SYSREAD Q1 = ID60 NR1** and the tool index with **FN18: SYSREAD Q7 = ID60 NR8.** Both values are then transferred in **TOOL CALL Q1 .Q7.**

```
0 BEGIN PGM TOOLCALL MM
1 * - ? DATUM SHIFT PLC OFF/ON
2 * - 3D ROTATION OFF/ON
3 * - READ TOOL CALL DATA
4 FN18: SYSREAD Q1 = ID60 NR1 IDX0 ; Tool number
5 FN18: SYSREAD Q2 = ID60 NR2 IDX0 ; Tool axis
6 FN18: SYSREAD Q3 = ID60 NR3 IDX0 ; Spindle speed
7 FN18: SYSREAD Q4 = ID60 NR4 IDX0 ; Oversize in tool length DL
8 FN18: SYSREAD Q5 = ID60 NR5 IDX0 ; Oversize in tool radius DR
9 FN18: SYSREAD Q6 = ID60 NR7 IDX0 ; Oversize in tool radius DR2
10 FN18: SYSREAD Q7 = ID60 NR8 IDX0 ; Tool index
11 * - T-NEW POSITION AXIS 7 (FKA)
12 FN 18: SYSREAD Q6 = ID2000 NR70 IDX98; Read W98
13 * - POSITION AXIS 8 (FK)
14 FN 18: SYSREAD Q7 = ID2000 NR80 IDX98; Read D98
15 * - TC Z SAFETY CLEARANCE
16 L Z+0 R0 F MAX M91
17 * - ALIGN SPINDLE M21
18 M21
19 FN 18: SYSREAD Q8 = ID1000 NR4210 IDX11 ; Read MP4210.11
20 FN 18: SYSREAD Q9 = ID1000 NR4210 IDX12 ; Read MP4210.12
21 FN 18: SYSREAD Q10 = ID230 NR3 IDX2 ; Pos. software limit switch Y
22 * - OPEN TRAVERSE RANGE Y
23 Q9 = Q6 + 0.5
24 FN 17: SYSWRITE ID230 NR3 IDX2 = +Q9
25 * - TC Z APPRAOCH AREA
26 L Y+Q6 R0 F MAX M91
27 L Z+Q7 R0 F MAX M91
28 * - TOOL MACRO ACTIVE ?
29 FN 20: WAIT FOR SYNC M1999==1
30 * - WRITE TOOL CALL DATA ->PLC
31 FN 9: IF +Q2 EQU +0 GOTO LBL 11
32 FN 9: IF +Q2 EQU +1 GOTO LBL 12
33 FN 9: IF +Q2 EQU +2 GOTO LBL 13
34 FN 9: IF +Q2 EQU +3 GOTO LBL 14
35 FN 9: IF +Q2 EQU +4 GOTO LBL 15
36 FN 9: IF +Q2 EQU +5 GOTO LBL 16
37 TOOL CALL Q1 SQ3 DL+Q4 DR+Q5
38 FN 9: If +0 EQU +0 GOTO LBL 17
39 LBL 11
40 TOOL CALL Q1 .Q7 X SQ3 DL+Q4 DR+Q5 DR2:+Q6
41 FN 9: If +0 EQU +0 GOTO LBL 17
42 LBL 12
43 TOOL CALL Q1 .Q7 Y SQ3 DL+Q4 DR+Q5 DR2:+Q6
44 FN 9: If +0 EQU +0 GOTO LBL 17
45 LBL 13
46 TOOL CALL Q1 .Q7 Z SQ3 DL+Q4 DR+Q5 DR2:+Q6
47 FN 9: If +0 EQU +0 GOTO LBL 17
48 LBL 14
49 TOOL CALL Q1 .Q7 U SQ3 DL+Q4 DR+Q5 DR2:+Q6
50 FN 9: If +0 EQU +0 GOTO LBL 17
51 LBL 15
52 TOOL CALL Q1 .Q7 V SQ3 DL+Q4 DR+Q5 DR2:+Q6
53 FN 9: If +0 EQU +0 GOTO LBL 17
54 LBL 16
55 TOOL CALL Q1 .Q7 W SQ3 DL+Q4 DR+Q5 DR2:+Q6
```

```
56 LBL 17
57 * - Z LEAVE TC AREA
58 L Z+0 R0 F MAX M91
59 L Y+0 R0 F MAX M91
60 * - CLOSE TRAVERSE RANGE Y
61 FN 17: SYSWRITE ID230 NR3 IDX2 = +Q8
62 END PGM TOOLCALL MM
```

Variable and fixed pocket coding

If you work with **one magazine,** you must specify the type of pocket coding for this magazine:

- Set MP7482 = %0000.
- Specify with MP7480.x whether the tool or pocket number is to be transferred to the PLC:
  - Variable pocket coding: Pocket number must be transferred. Set MP7480.x = 3 or 4.
  - Fixed pocket coding: Working with the tool number is preferred. Set MP7480.x = 1 or 2.

If you work with **more than one magazine,** you must specify the type of pocket coding for each magazine individually:

- Set MP7480.x to 3 or 4 for variable pocket coding.
- Define in MP7482 the type of pocket coding for each magazine.

Depending on the setting of MP7480.x, the NC transfers either only the number of the programmed tool to word W264 or the tool and pocket number to W262 and W264.
The NC sets M4073 **(TOOL CALL)** or M4074 **(TOOL DEF)**. The strobe markers are not reset until you have set M4093 **(TOOL CALL)** or M4094 **(TOOL DEF** after the tool or pocket number, respectively, have been processed. After you have reset the strobe marker, the NC program is resumed (only with **TOOL CALL**).
If a **TOOL CALL** block is followed by the output of a T strobe and G strobe, then M4547 is set by the output of the T strobe and reset by output of the G strobe. If there is no output of either the T or G strobe, M4547 is not set.

If the tool number zero is processed, the NC sets marker M4521. The marker is not reset until there is a **TOOL CALL** for another tool.

With MP7483 you specify if the tool name or tool number or both can be used for **TOOL CALL** or **TOOL DEF**. For example, if a tool number is used, an NC error message will be output if only tool names are allowed (MP7483 = 1).

Example of how the FN18 functions for tool name and number can be used in combination with MP7483. In this example the Q-parameter number Q330 could have been transferred as a string or a number from a cycle. The following example can be used for further processing:

`...`		
`FN 9:`	`IF +Q330 EQU +0 GOTO LBL 131`	
`FN 18:`	`SYSREAD Q35 = ID1000 NR7483`	Interrogate MP7483 whether tool name and number are permitted
`FN 18:`	`SYSREAD Q30 = ID30 NR52 IDX330`	Interrogate if the tool name is programmed
`FN 9:`	`IF +Q30 EQU +0 GOTO LBL 45`	No tool name programmed
`FN 10:`	`IF +Q35 NE +2 GOTO LBL 46`	Tool name programmed
`FN 14:`	`ERROR = 1094`	Tool name is not permitted
`LBL 46`		
`FN 18:`	`SYSREAD Q330 = ID990 NR10 IDX330`	Determine the tool number for the tool name
`FN 10:`	`IF +Q330 NE -1 GOTO LBL 34`	
`FN 14:`	`ERROR = 1092`	No tool number could be found
`LBL 45`		
`FN 10:`	`IF +Q35 NE +1 GOTO LBL 34`	
`FN 9:`	`IF +Q330 EQU +0 GOTO LBL 34`	Tool number = 0 always permitted
`FN 14:`	`ERROR = 1093`	Tool number not permitted
`LBL 34`		
`FN 18:`	`SYSREAD Q0 = ID50 NR22 IDX330`	
`...`		

MP7480	**Output of the tool or pocket number**
MP7480.0	With a TOOL CALL block
Input:	0: No output 1: Tool number output only when tool number changes 2: Tool number output for every TOOL CALL block 3: Pocket number and tool number output only when tool number changes 4: Pocket number and tool number output for every TOOL CALL block 5: Pocket number and tool number output only when tool number changes. Pocket table is not changed. 6: Pocket number and tool number output for every TOOL CALL block. Pocket table is not changed.
MP7480.1	With a TOOL DEF block
Input:	0: No output 1: Tool number output only when tool number changes 2: Tool number output for every TOOL DEF block 3: Pocket number and tool number output only when tool number changes 4: Pocket number and tool number output for every TOOL DEF block

MP7482	**Pocket coding of the tool magazine**
Format:	%xxxx 0: Variable pocket coding 1: Fixed pocket coding
Input:	Bit 0: Magazine 1 Bit 1: Magazine 2 Bit 2: Magazine 3 Bit 3: Magazine 4

MP7483 **Tool name/number for TOOL CALL / TOOL DEF**
Input: 0: Names and numbers are permitted (as before)
1: Only names are permitted
2: Only numbers are permitted

		Set	Reset
W262	**Tool pocket number**	NC	NC
W264	**Tool number**	NC	NC
M4073	**Strobe signal T code (P code) with TOOL CALL**	NC	NC
M4074	**Strobe signal T code (P code) with TOOL DEF**	NC	NC
M4093	**Acknowledgment of T code (P code) with TOOL CALL**	PLC	PLC
M4094	**Acknowledgment of T code (P code) with TOOL DEF**	PLC	PLC
M4521	**Tool number zero programmed**	NC	NC
M4538	**Geometry of the tool from W264**	PLC	NC
M4547	**T and G strobes with TOOL CALL**	NC	NC

Output of the tool number with fixed pocket coding

For fixed pocket coding of tools you must evaluate the tool number:

▶ With MP7480.x, specify when the tool number is to be transferred.

- For every **TOOL CALL** or **TOOL DEF** block: MP7480.x = 2
- When the tool number changes: MP7480.x = 1
  During execution of a **TOOL CALL** or **TOOL DEF** block the tool number is saved in W264. W262 is not used.
- For MP7480.x = 5 or 6: The pocket number is saved in W262. The assignment of tool and pocket number in the pocket table does not change.

Output of the pocket number with variable pocket coding

With variable pocket coding (MP7480.x = 3 or 4) the pocket number of the called tool is transferred to the PLC and the assignment of tool and pocket number is changed in the pocket table. The current tool number is additionally saved in W264. The NC takes over responsibility for variable pocket management.

If you have set M4542, the assignment of tool and pocket numbers in the pocket table does not change, although variable pocket coding was selected. You set this marker, for example, during a block scan (except if MP7681 bit 1=1).

▶ In MP7261, enter the number of tools with pocket number. The maximum input value is the number of the pockets in the tool magazine.

In the tool table you can define more tools than can be held by the tool magazine (MP7260 > MP7261). If a tool number is programmed for which no pocket was defined, during a **TOOL CALL** the pocket number –1 (W262) is transferred and M4523 is set.

During programming of **TOOL DEF** the tool and pocket numbers are transferred. A **TOOL DEF** for a manual tool has no relevance for the PLC.

Define in the column **F** a fixed pocket. If a fixed pocket has been defined for a tool, it will be returned to its original pocket in spite of the variable pocket coding.

		Set	Reset
M4520	**Another T code (P code) follows with TOOL CALL** 0: Normal tool follows a normal tool (N → N) Manual tool follows a manual tool (M → M) Special tool follows a special tool (S → S), if M4541 = 0 1: Special tool follows a manual tool (M → S), if M4541 = 1 Special tool follows a special tool (S → S), if M4541 = 1 Manual tool follows a special tool (S → M) Manual tool follows a normal tool (N → M) Normal tool follows a manual tool (M → N) Normal tool follows a special tool (S → N) see M4540	NC	NC
M4522	**Tool with pocket number programmed is in effect with MP7480.0 = 3 or 4 and TOOL CALL**	NC	NC
M4523	**Tool without pocket number programmed is in effect with MP7480.0 = 3 or 4 and TOOL CALL**	NC	NC
M4524	**Special tool called, TOOL CALL**	NC	NC
M4525	**TOOL CALL after expiration of tool life** 1: TOOL CALL after expiration of tool life	NC	NC
M4540	**Sequence of tool number or pocket number transfer (M4520 = 1)** ■ 0: First the number for the old tool, then the number for the new tool (single changing arm) ■ 1: First the number for the new tool, then the number for the old tool (double changing arm)	PLC	PLC
M4541	**Special tool in original pocket in spite of variable pocket coding**	PLC	PLC
M4542	**Do not update pocket number in pocket table**	PLC	PLC

A variety of tool types can be called from the machining program. The abbreviations below are defined for the following examples:

- **N:** Tool for which one pocket is defined in the tool table (**N**ormal)
- **M:** Tool for which no pocket number is defined in the tool table. You must change the tool manually (**M**anual).
- **S: S**pecial tool, definition in the tool table

There are nine possible combinations in the tool-change sequence. For some sequences it is necessary during **TOOL CALL** to output two pocket or tool numbers in sequence. You can see in M4520 if another tool or pocket number is transferred. The sequence of transfers for tool and pocket numbers can be defined in two manners:

- Define the sequence in MP7481.x. In this case set M4540 = 0.
- Define the sequence in M4540. In this case set MP7481.x = %00000000.

You must evaluate and acknowledge both pocket or tool numbers. As an alternative, all of this information can be found with Module 9035.

MP7481	**Sequence for new and returned tool when changing tools**
Format:	%xxxxxxxx 0: First, output the pocket of the tool to be returned 1: First, output the pocket of the new tool
Input:	Bit 0: New tool from magazine 1 Bit 1: New tool from magazine 2 Bit 2: New tool from magazine 3 Bit 3: New tool from magazine 4 Bit 4: New tool from magazine 5 Bit 5: New tool from magazine 6 Bit 6: New tool from magazine 7 Bit 7: New tool from magazine 8
MP7481.0	Tool to be returned to magazine 1
MP7481.1	Tool to be returned to magazine 2
MP7481.2	Tool to be returned to magazine 3
MP7481.3	Tool to be returned to magazine 4
MP7481.4	Tool to be returned to magazine 5
MP7481.5	Tool to be returned to magazine 6
MP7481.6	Tool to be returned to magazine 7
MP7481.7	Tool to be returned to magazine 8

Module 9035 Read status information

Call:

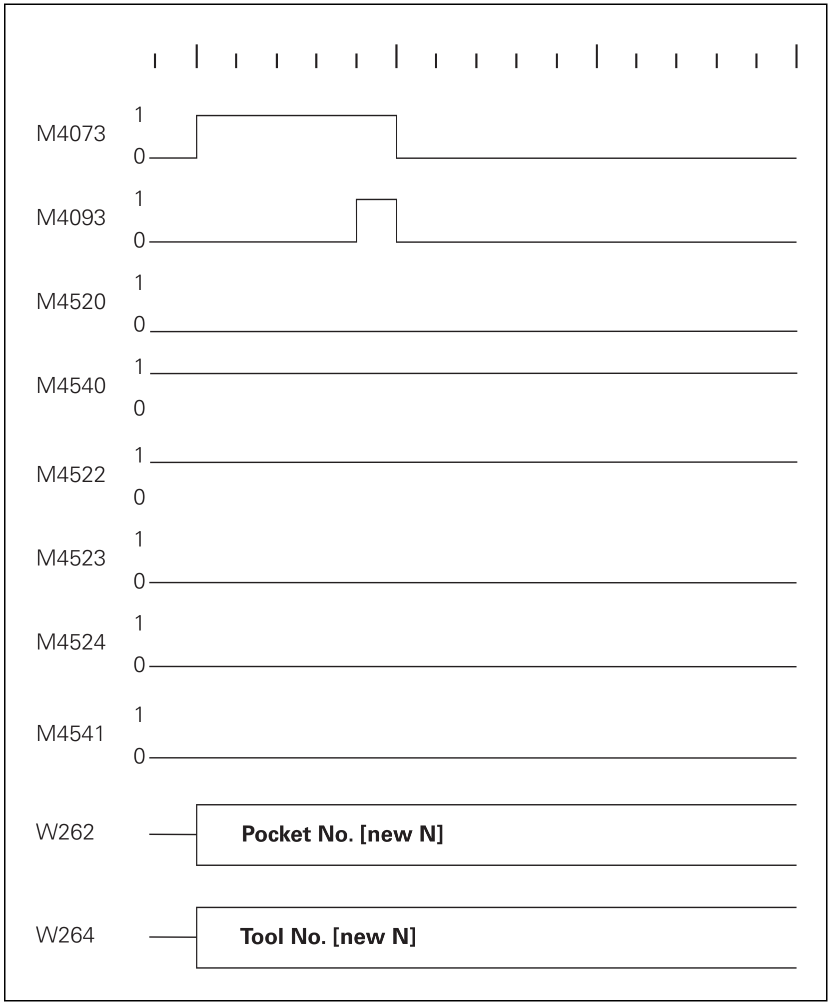
PS	B/W/D/K	<Number of the status information> 50: Tool change sequence (see **`FN18: SYSREAD ID61 NR0`**) 51: Pocket number for reserve 52: Magazine number for reserve 53: Pocket number for insertion 54: Magazine number for insertion
CM	9035	
PL	B/W/D	<Status information>

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid
	20	Call was not in a submit or spawn job

N → N: Normal tool follows a normal tool

The pocket number and the tool number of the called tool are transferred.



S → N: Normal tool follows a special tool

With this change sequence, two pocket numbers or two tool numbers must be transferred in succession. M4520 indicates that another **TOOL CALL** strobe (M4073) will follow:

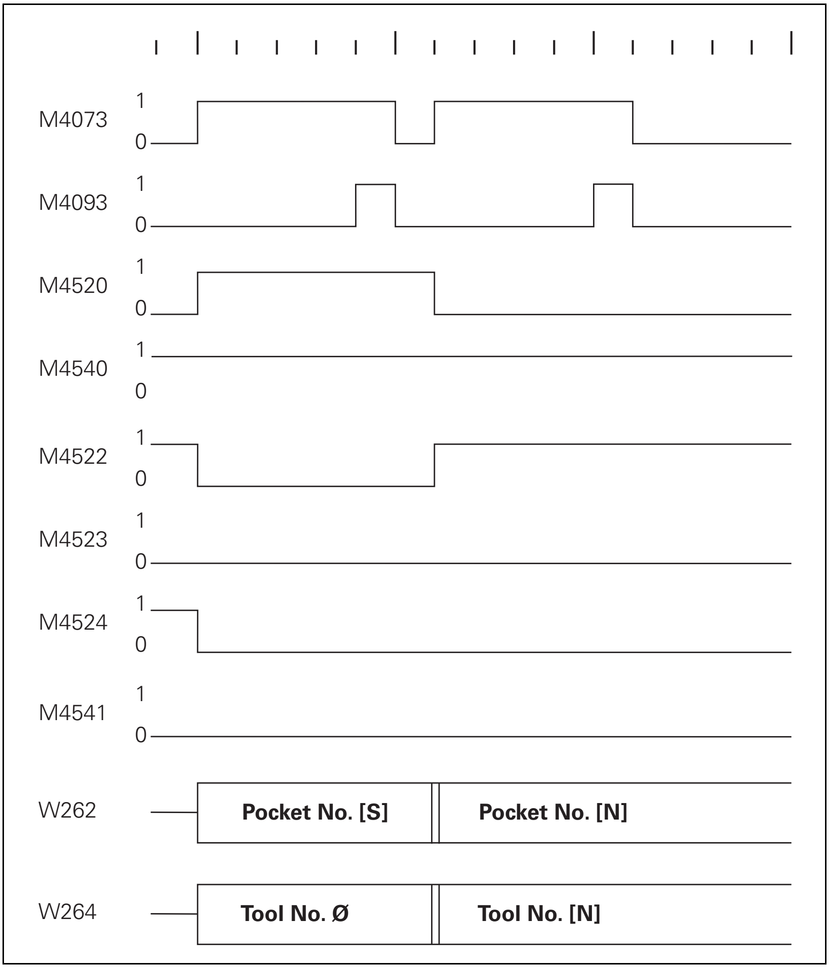
- With M4540 or MP7481.x specify the sequence in which the pocket numbers are transferred, depending on whether single or double changing arm.

S → N, Single changing arm, M4540 = 0 or MP7481.x, bit x = 0

First the pocket number of the old tool and the tool number zero are transferred.

Zero means clear the spindle!

- Clear the spindle and acknowledge with M4093.
  Then the pocket and tool numbers of the new tool are transferred.

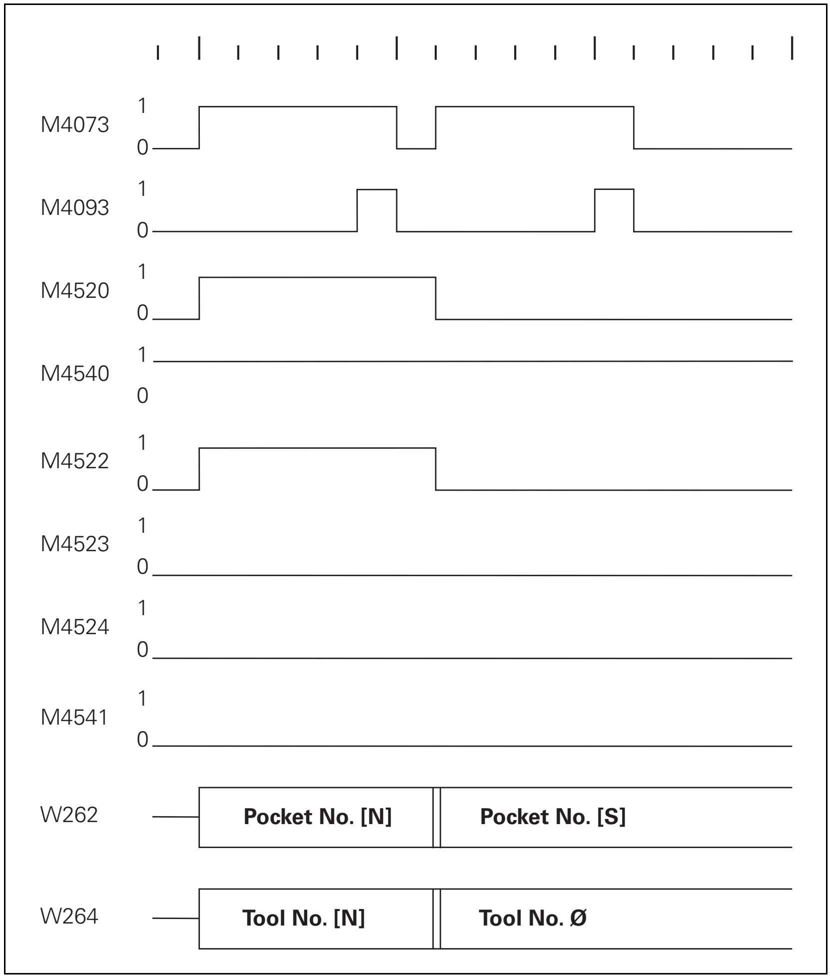


S → N, Double changing arm, M4540 = 1 or MP7481.x, bit x = 1

First the pocket and tool numbers of the new tool are transferred.

- Acknowledge with M4093.
  Then the pocket tool number of the old tool and the tool number zero are transferred.

Zero means clear the spindle!

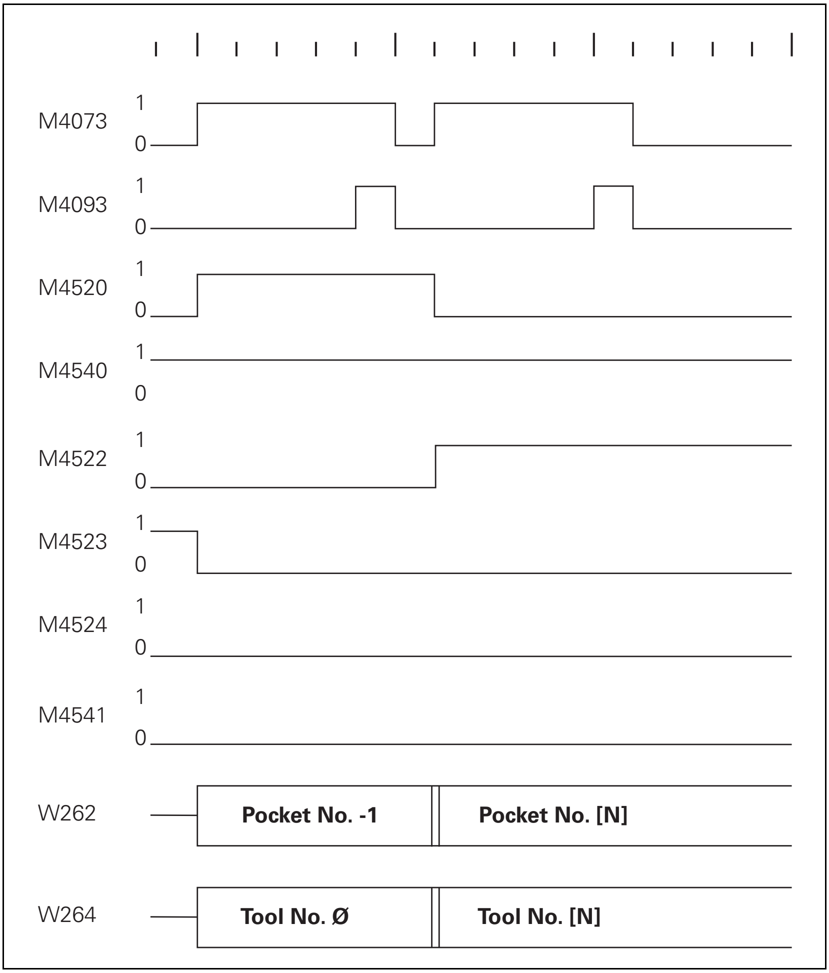


M → N:
Normal tool follows a manual tool

With this change sequence, two pocket numbers or two tool numbers must be transferred in succession. M4520 indicates that another `TOOL CALL` strobe (M4073) will follow. Regardless of M4540 or MP7481.x, the pocket number –1 and tool number zero are transferred first.

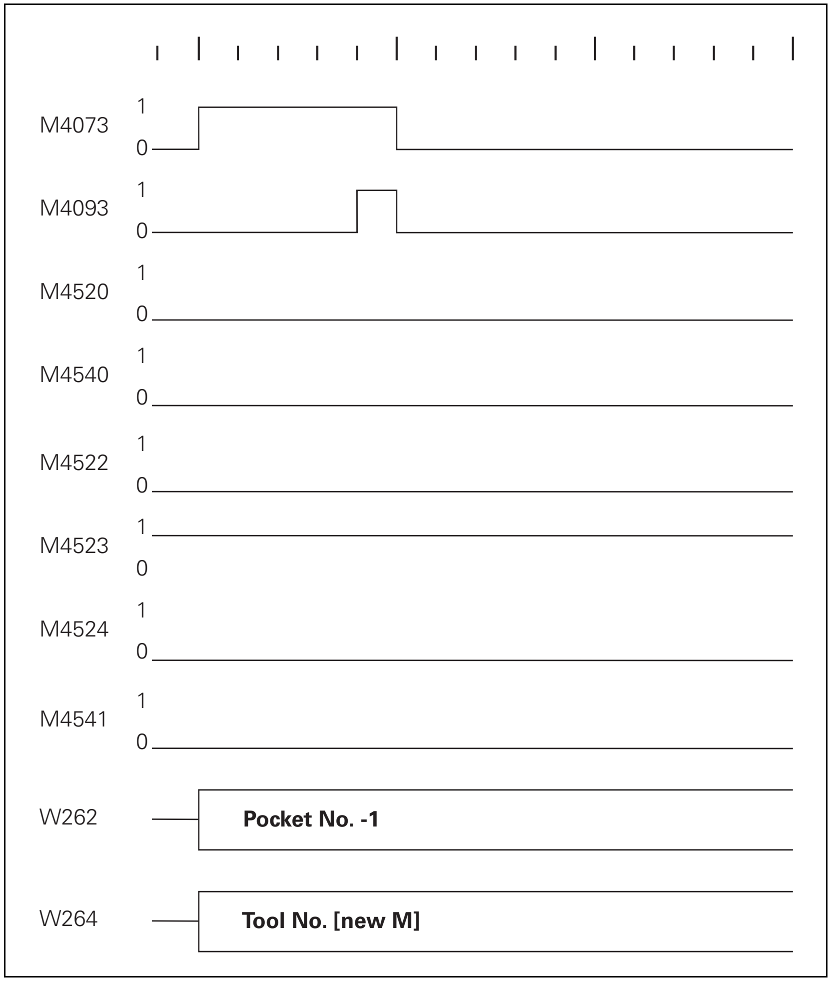
Zero means clear the spindle!
Pocket number –1 means: no pocket in the tool magazine!

- Acknowledge with M4093.
  Then the pocket number and tool number of the new, called tool are transferred.



M → M: Manual tool follows a manual tool

Pocket number –1 means: no pocket in the tool magazine!



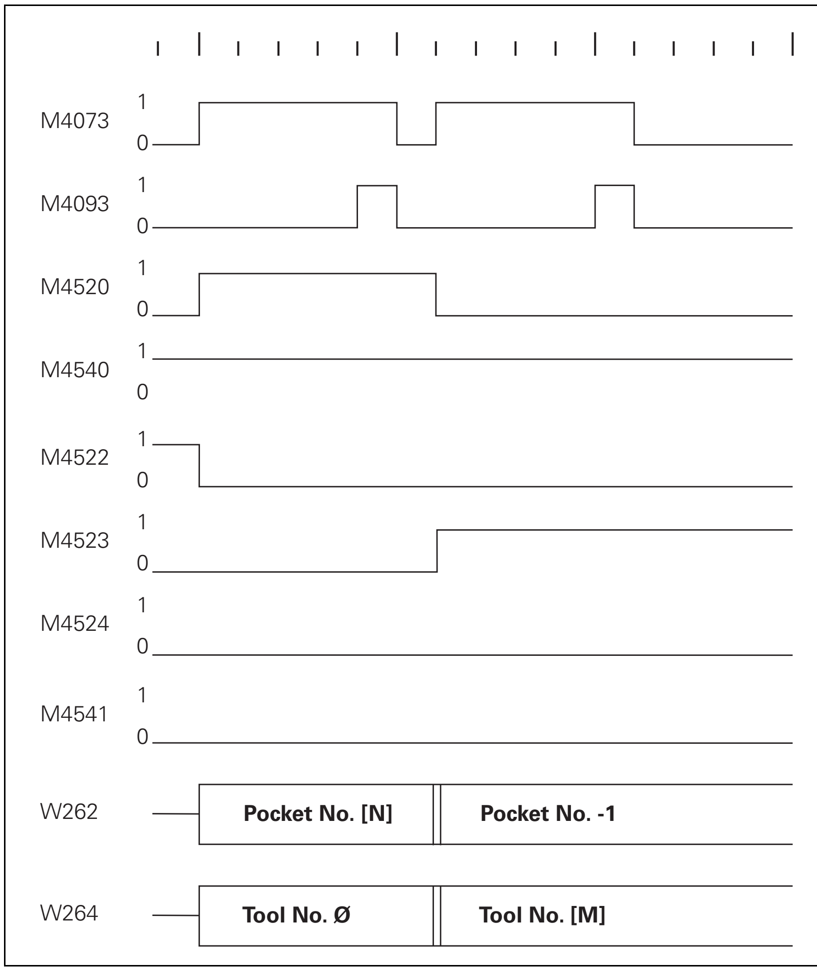
N → M: Manual tool follows a normal tool

With this change sequence, two pocket numbers or two tool numbers must be transferred in succession. M4520 indicates that another **TOOL CALL** strobe (M4073) will follow. Regardless of M4540 or MP7481.x, the pocket number of the old tool and tool number zero are transferred first.

Zero means clear the spindle!

- Acknowledge with M4093. Then the pocket number –1 and tool number of the new, called tool are transferred.

Pocket number –1 means: no pocket in the tool magazine!



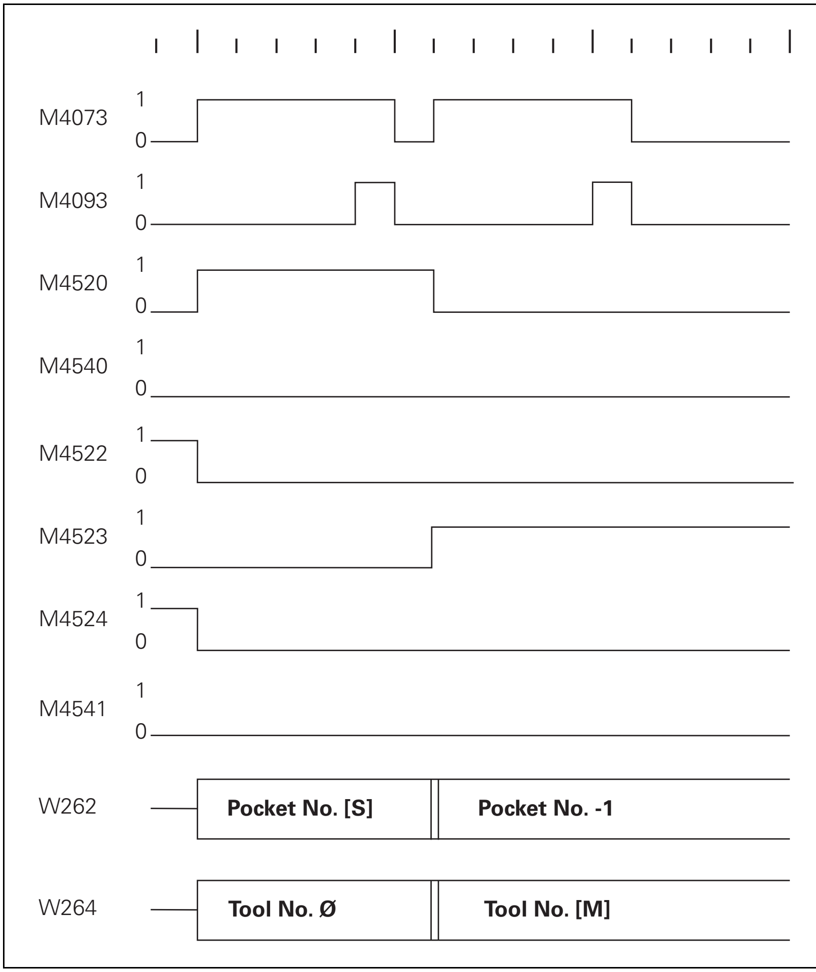
S → M: Manual tool follows a special tool

With this change sequence, two pocket numbers or two tool numbers must be transferred in succession. M4520 indicates that another **TOOL CALL** strobe (M4073) will follow. Regardless of M4540 or MP7481.x, the pocket number of the old tool and tool number zero are transferred first.

Zero means clear the spindle!

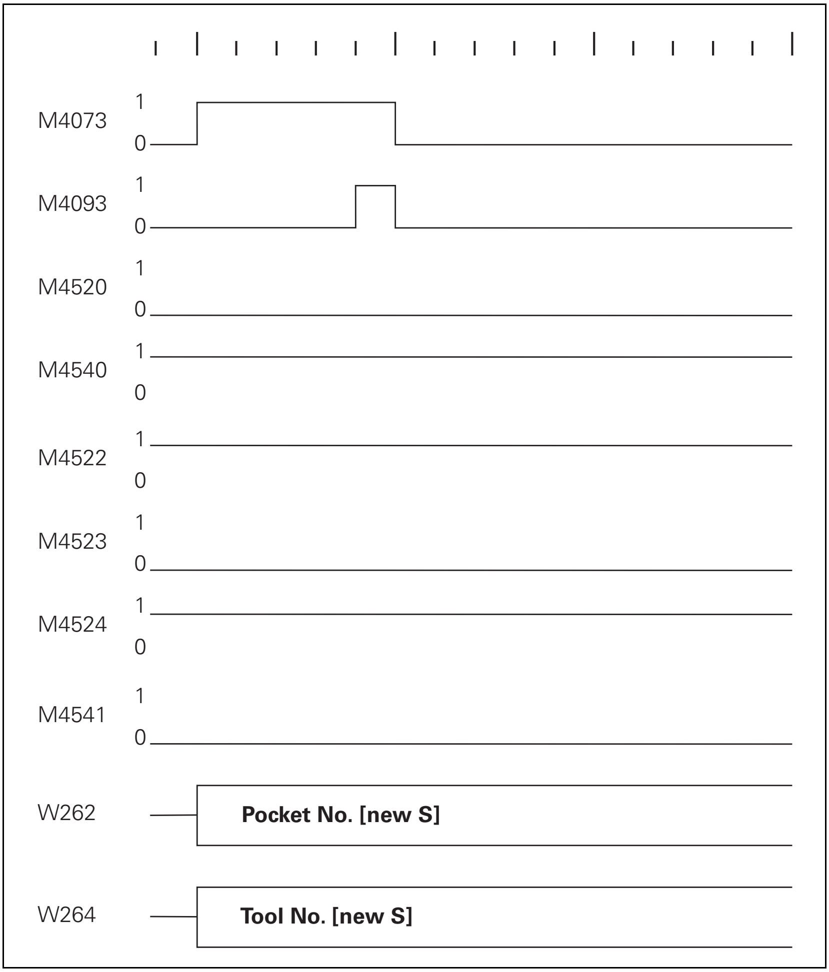
- Acknowledge with M4093.
  Then the pocket number –1 and tool number of the new, called tool are transferred.

Pocket number –1 means: no pocket in the tool magazine!



S → S: Special tool follows a special tool

- With M4541 or the column **F** in the pocket table, specify whether the special tool should be returned to the original pocket in spite of variable pocket coding.
  - No, M4541 = 0
    The same logic program applies for single and double changer arms.
  - Yes, M4541 = 1
    Single and double changer arms have different sequences of pocket number transfer.

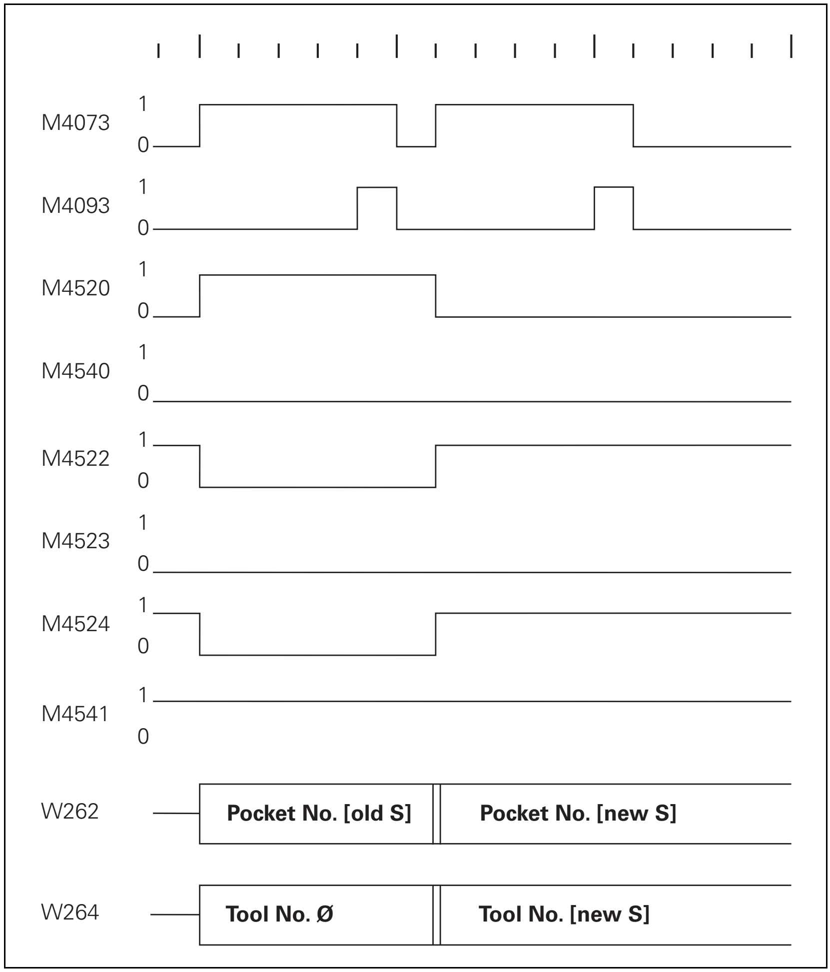


S → S, Single changing arm, M4540 = 0 or MP7481.x, bit x = 0

First the pocket number of the old tool and the tool number zero are transferred.

Zero means clear the spindle!

- Acknowledge with M4093.
  Then the pocket number and tool number of the new tool are transferred.

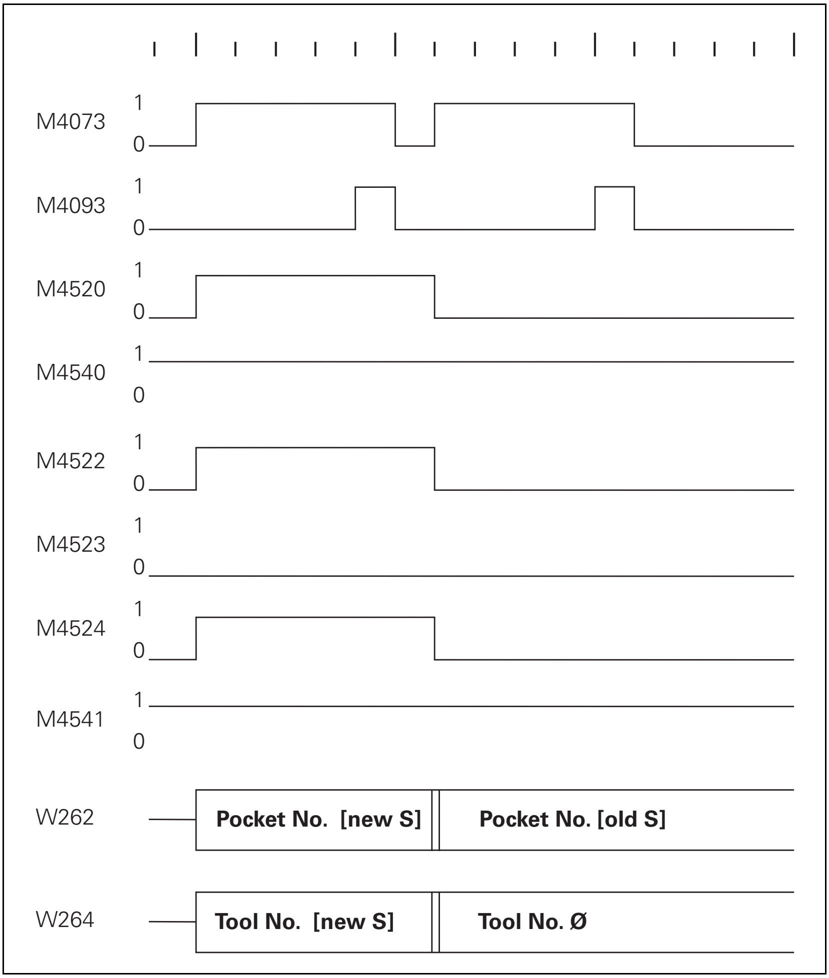


S → S, Double changing arm, M4540 = 1 or MP7481.x, bit x = 1

First the pocket number and tool number of the new tool are transferred.

- Acknowledge with M4093.
  Then the pocket tool number of the old tool and the tool number zero are transferred.

Zero means clear the spindle!



N → S: Special tool follows a normal tool

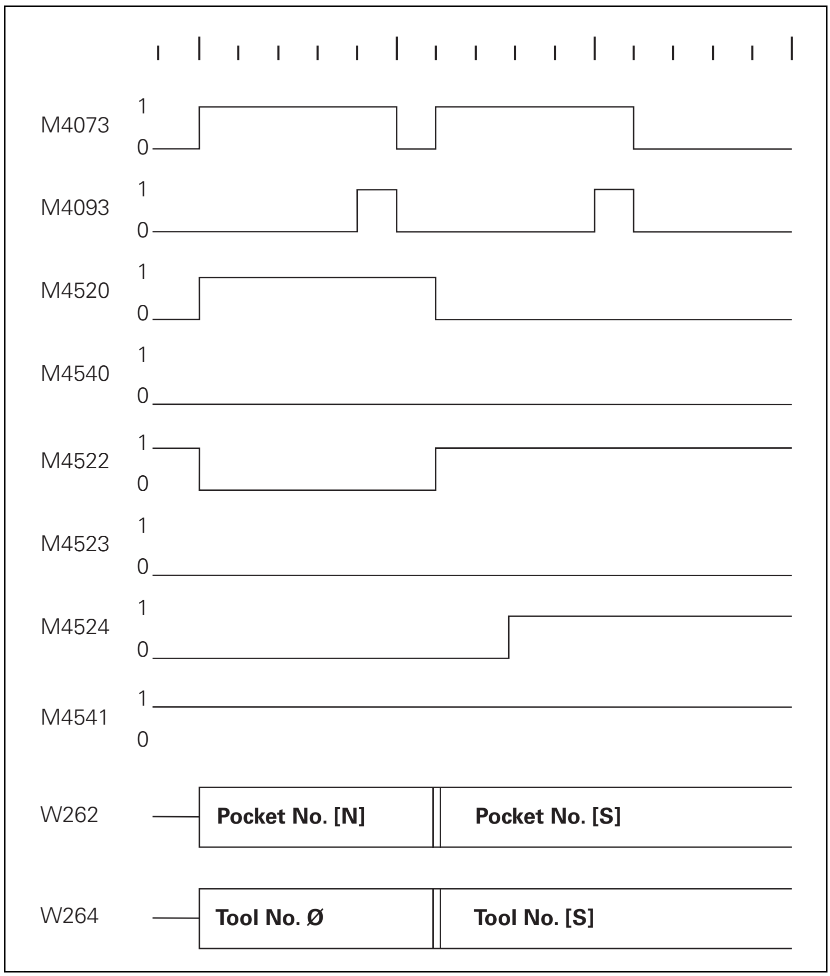
With this change sequence, two pocket numbers or two tool numbers must be transferred in succession. M4520 indicates that another **TOOL CALL** strobe (M4073) will follow. Regardless of M4541, there is a different sequence for the pocket number transfer for single and double-arm changers (M4540 or MP7481.x).

N → S, Single changing arm, M4540 = 0 or MP7481.x, bit x = 0

First the pocket number of the old tool and the tool number zero are transferred.

Zero means clear the spindle!

- Acknowledge with M4093.
  Then the pocket number and tool number of the new tool are transferred.

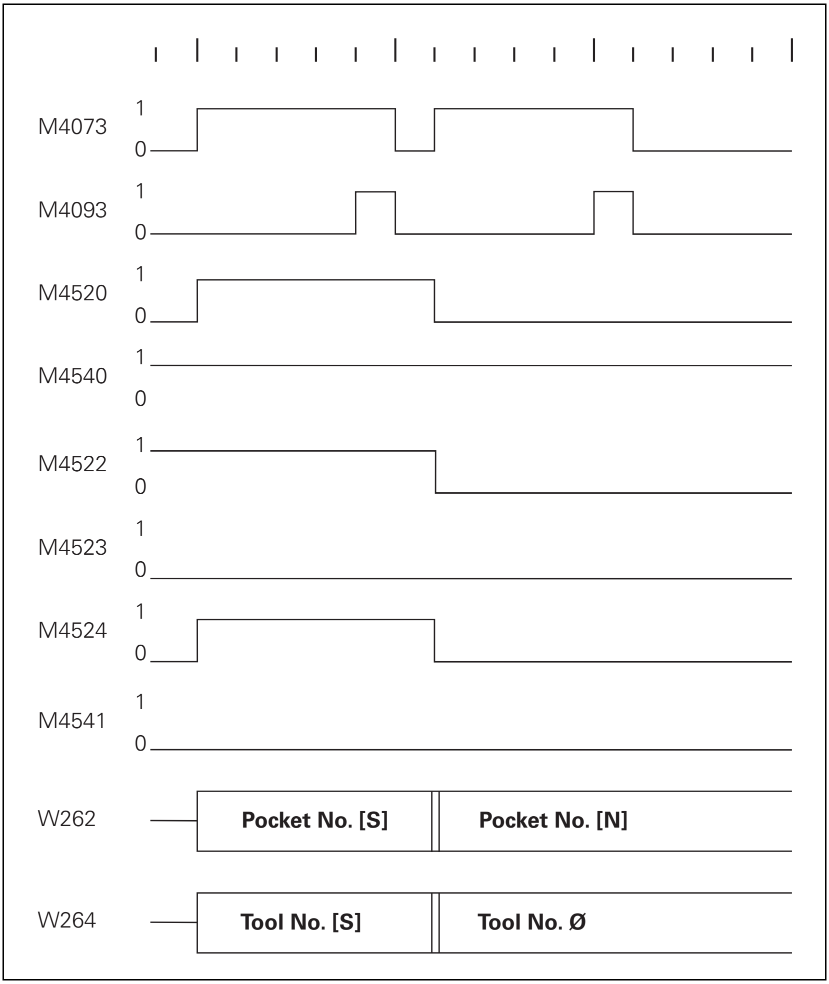


N → S, Double changing arm, M4540 = 1 or MP7481.x, bit x = 1

First the pocket and tool numbers of the new tool are transferred.

- Acknowledge with M4093.
  Then the pocket tool number of the old tool and the tool number zero are transferred.

Zero means clear the spindle!

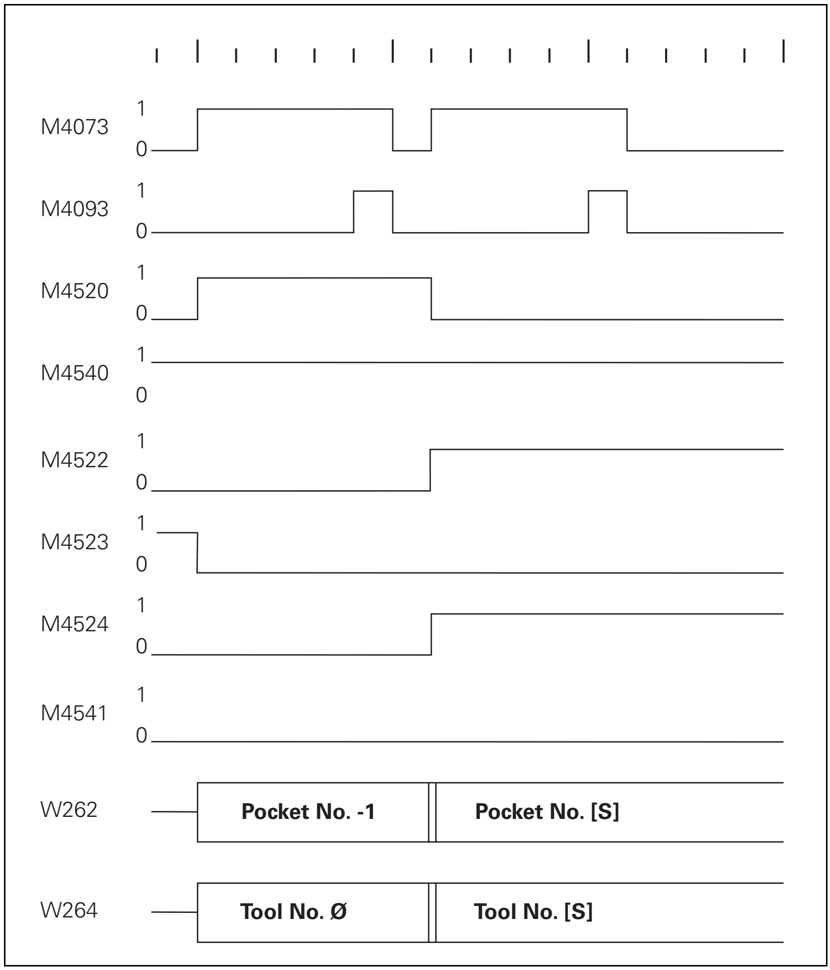


M → S:
Special tool follows a manual tool

With this change sequence, two pocket numbers or two tool numbers must be transferred in succession. M4520 indicates that another **TOOL CALL** strobe (M4073) will follow. Regardless of M4541 and M4540 or MP7481.x, the pocket number –1 and tool number zero are transferred first.

Tool number zero means clear the spindle!
Pocket number –1 means: no pocket in the tool magazine!

- Acknowledge with M4093.
  Then the pocket number and tool number of the new, called tool are transferred.



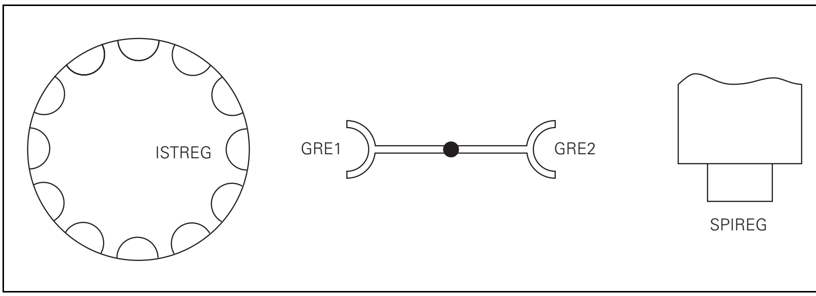
8.15.6 PLC Programming Example

Tool changer and basic flowcharts of the associated PLC program

▶ Create the PLC program with the PLC development software PLCdesign.

The tool changer treated here has the following features:

- Up to 254 tools
- Variable pocket coding, MP7480.x = 4
- Special tools allowed
- Providing the next tool with **TOOL DEF**
- Tool change with **TOOL CALL**
- Tools can be manually changed without pocket number definition in the tool table
- Double changing arm
- Special tools variable, MP4541 = 0



The following variables are used in the basic flowchart:

- ISTREG = The pocket number at the tool change position of the tool magazine
- GRE1 = Pocket number of tool in changing arm facing tool magazine
- GRE2 = Pocket number of the tool in the arm facing the spindle
- SPIREG = Pocket number of the tool in the spindle

		Set	Reset
W262	**Tool pocket number**	NC	NC
W264	**Tool number**	NC	NC
M4073	**Strobe signal T code (P code) with TOOL CALL**	NC	NC
M4074	**Strobe signal T code (P code) with TOOL DEF**	NC	NC
M4093	**Acknowledgment of T code (P code) with TOOL CALL**	NC	NC
M4094	**Acknowledgment of T code (P code) with TOOL DEF**	NC	NC
M4520	**Additional T code (P code) follows with TOOL CALL**	NC	NC
M4524	**Special tool called (TOOL CALL)**	NC	NC
M4540	**Sequence of the tool numbers or pocket number transfer, M4520 = 1**	PLC	PLC
M4541	**Special tool to original pocket in spite of variable pocket coding**	PLC	PLC

Machines parameter that are used:

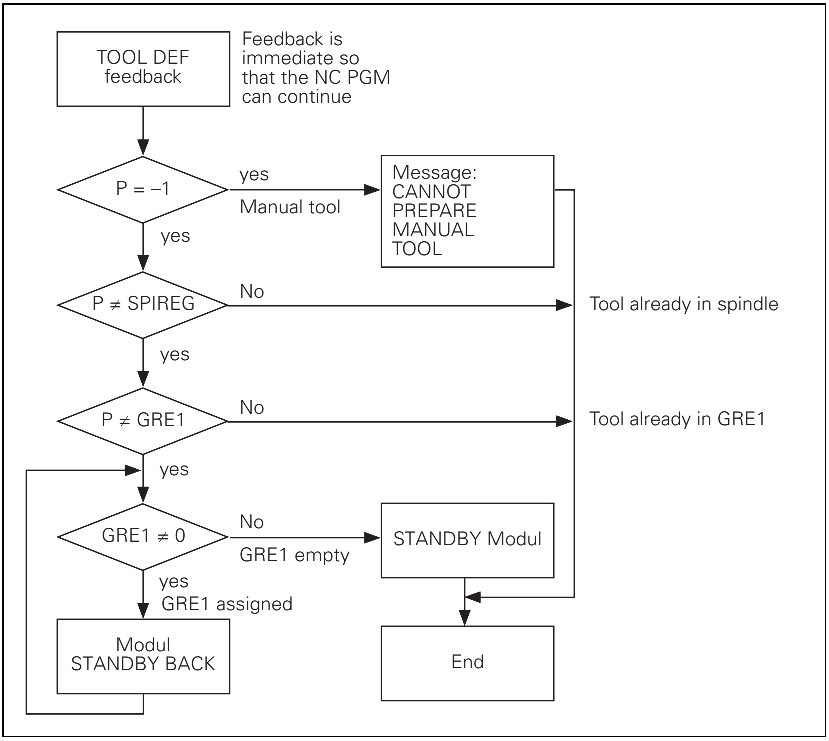
Machine parameters	Meaning
MP7260 = 90	Number of tools in the tool table
MP7261 = 12	Number of the pockets in the tool magazine
MP7480.0 = 4	Pocket number and tool number output for every TOOL CALL block.
MP7480.1 = 4	Pocket number and tool number output for every TOOL DEF block

The flowchart for this tool changer is divided into the following modules or subprograms:

Module	Meaning
TOOL DEF	Search for tool and load in GRE1
TOOL CALL	Automatic tool change
STANDBY	Search for tool and load in GRE1
STANDBY BACK	Return tool from GRE1 to the magazine
MANUAL TOOL IN	Manual tool follows a normal or special tool
MANUAL TOOL OUT	Normal or special tool follows a manual tool
MANUAL IN/OUT	Manual tool follows a manual tool
INSERT	Replace old tool with new tool
COMPUTE SHORTEST DIRECTION	
COMPARE P CODE WITH ISTREG	
COMPARE GRE1 WITH ISTREG	

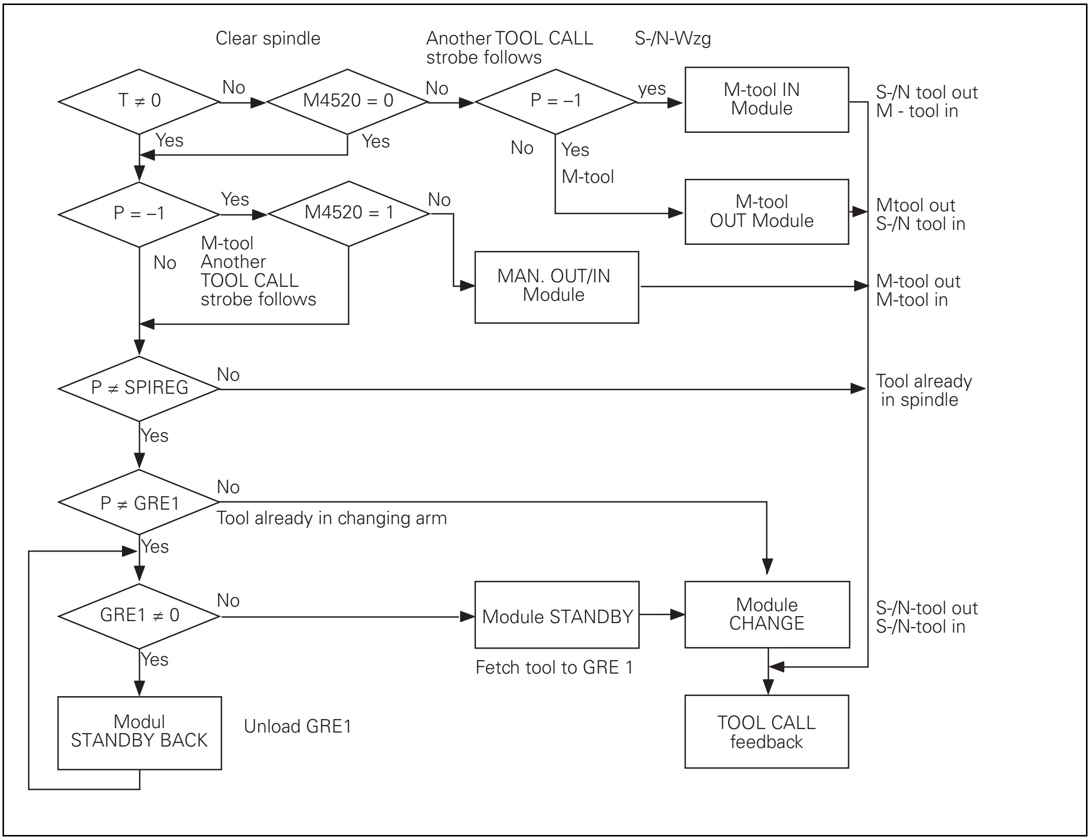
TOOL DEF program module

Search for tool and load in GRE1



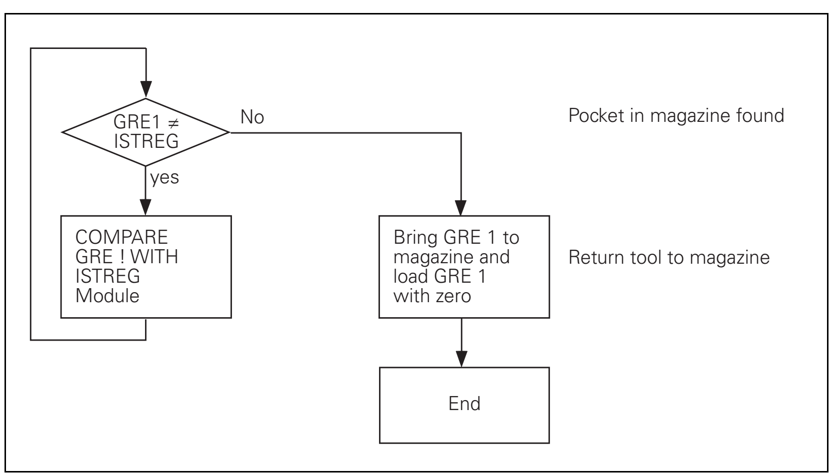
TOOL CALL program module

Automatic tool change, main program



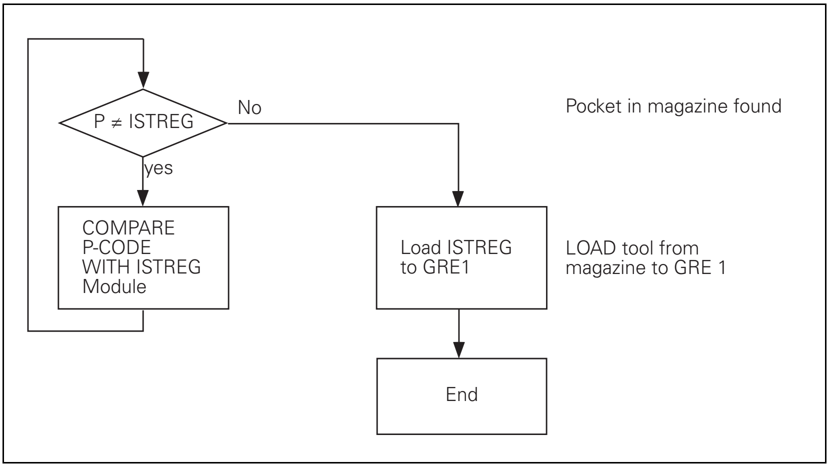
STANDBY program module

Search for tool and load in GRE1



STANDBY BACK program module

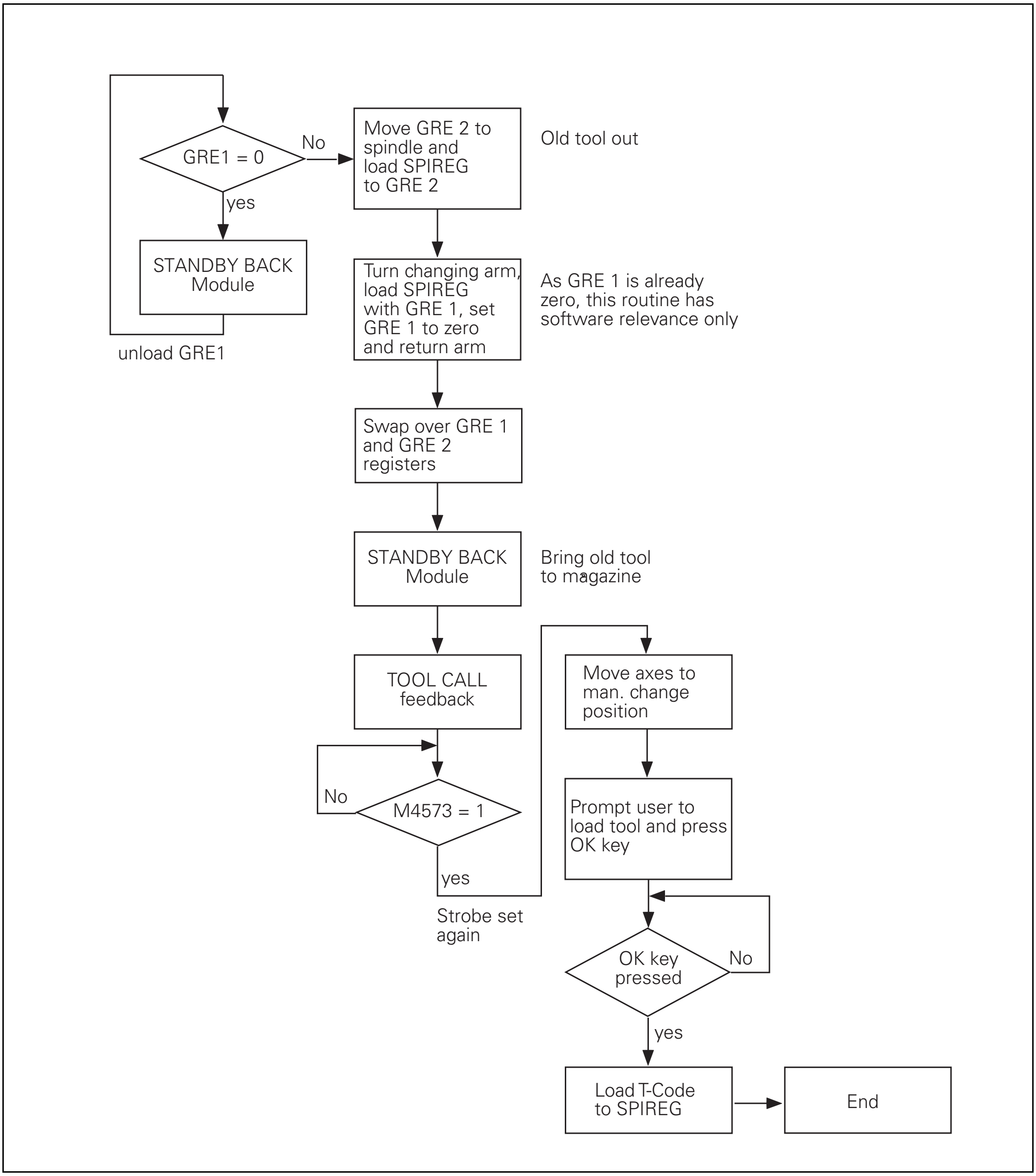
Return tool from GRE1 to the tool magazine



MANUAL TOOL IN program module

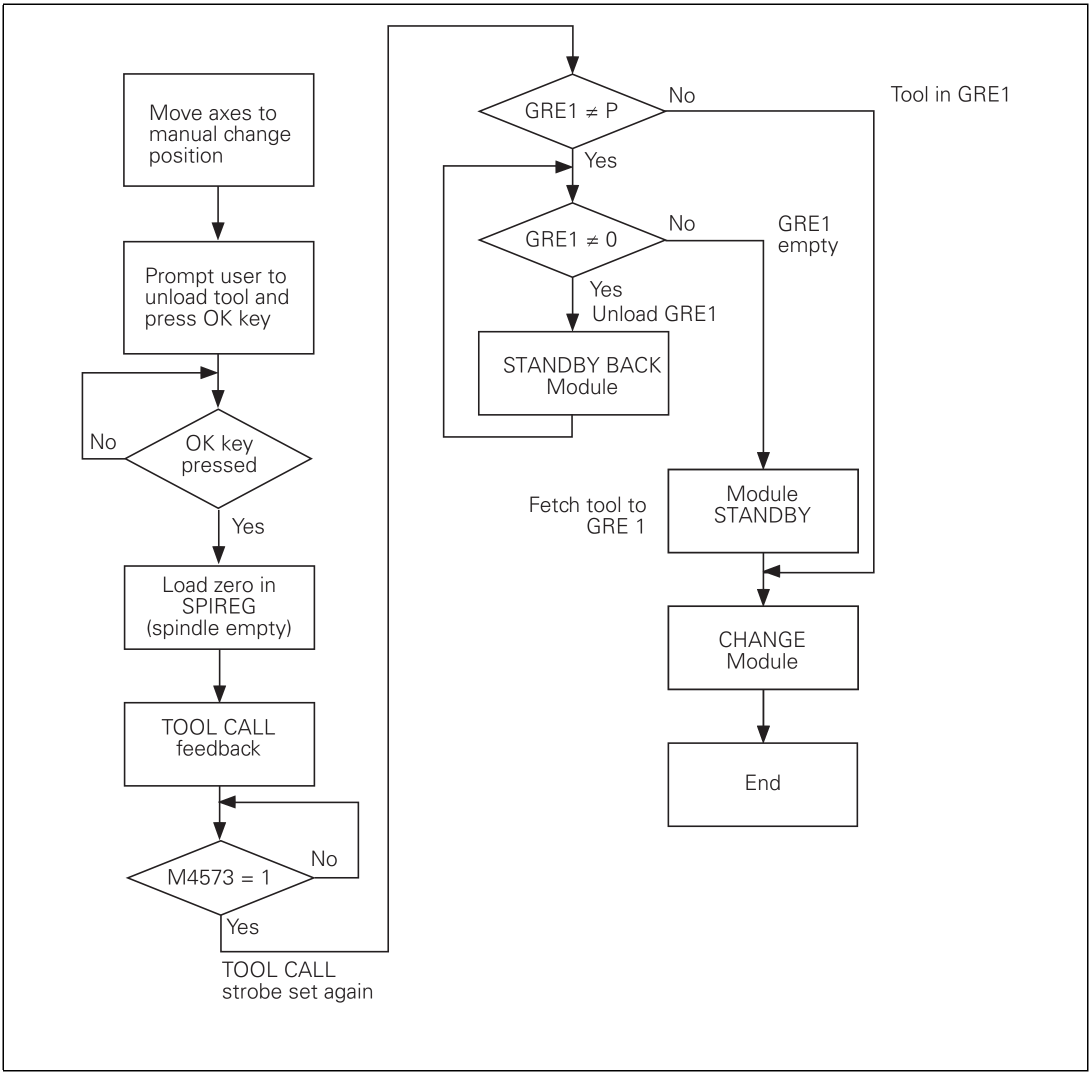
N → M or S → M:

Manual tool follows a normal or special tool The old tool is placed in the tool magazine and the user is prompted to insert a manual tool (which is not in the tool magazine).



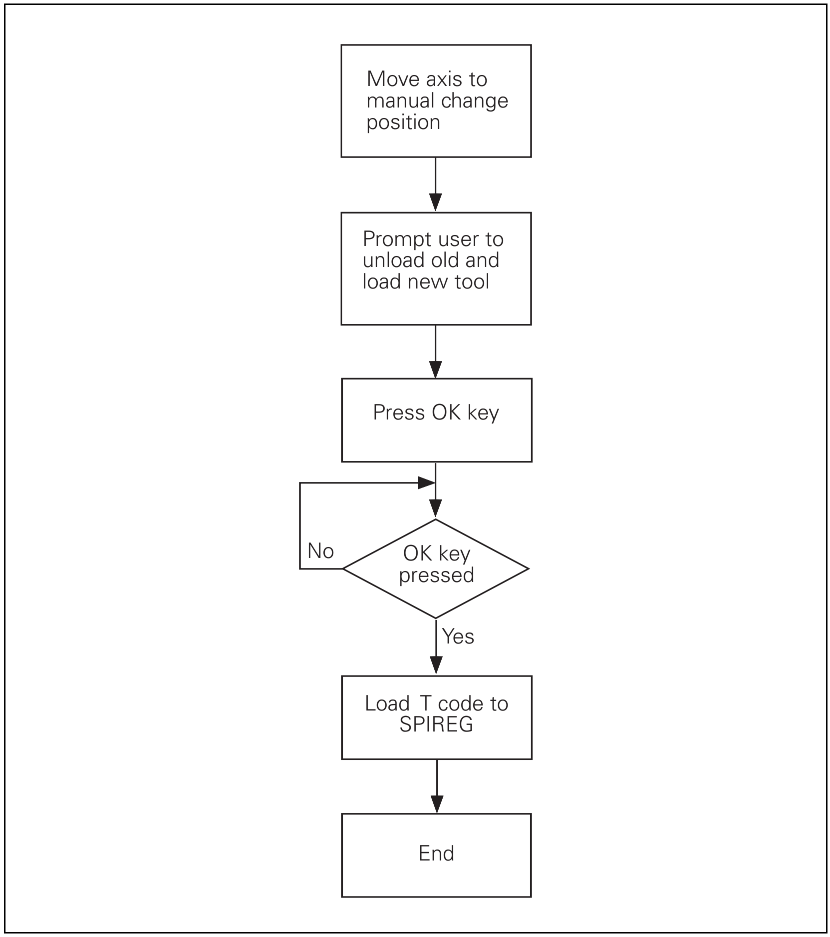
MANUAL TOOL OUT program module

M → N or M → S:
Normal or special tool follows a manual tool The operator is prompted to empty the spindle manually, since there is no room in the tool magazine for the current tool. The called tool is automatically inserted.



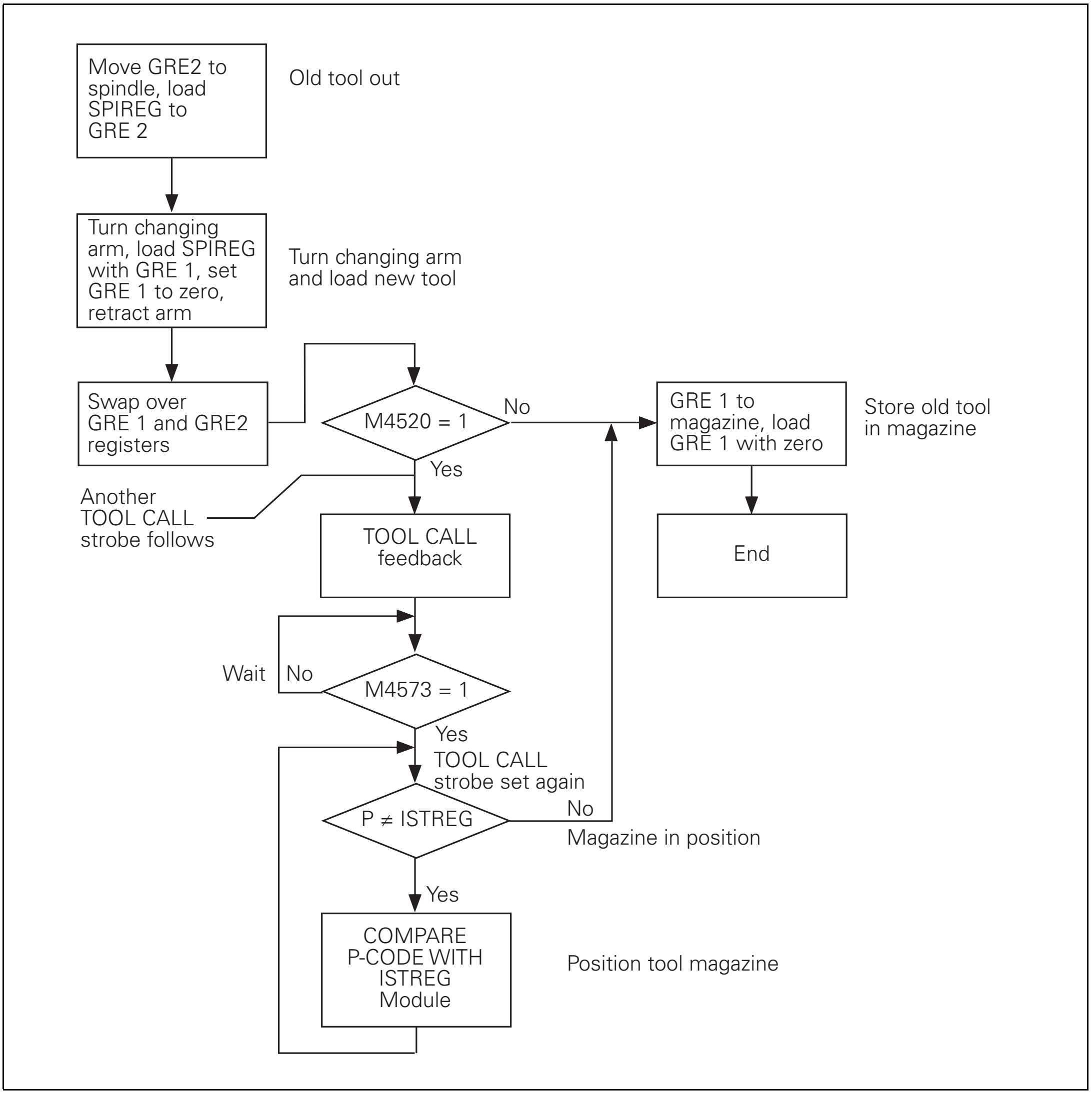
MANUAL TOOL OUT/IN program module

M → M:
Manual tool follows a manual tool. The user is prompted to remove the tool from the spindle manually and insert the new tool, since there is not room for the tools in the tool magazine.



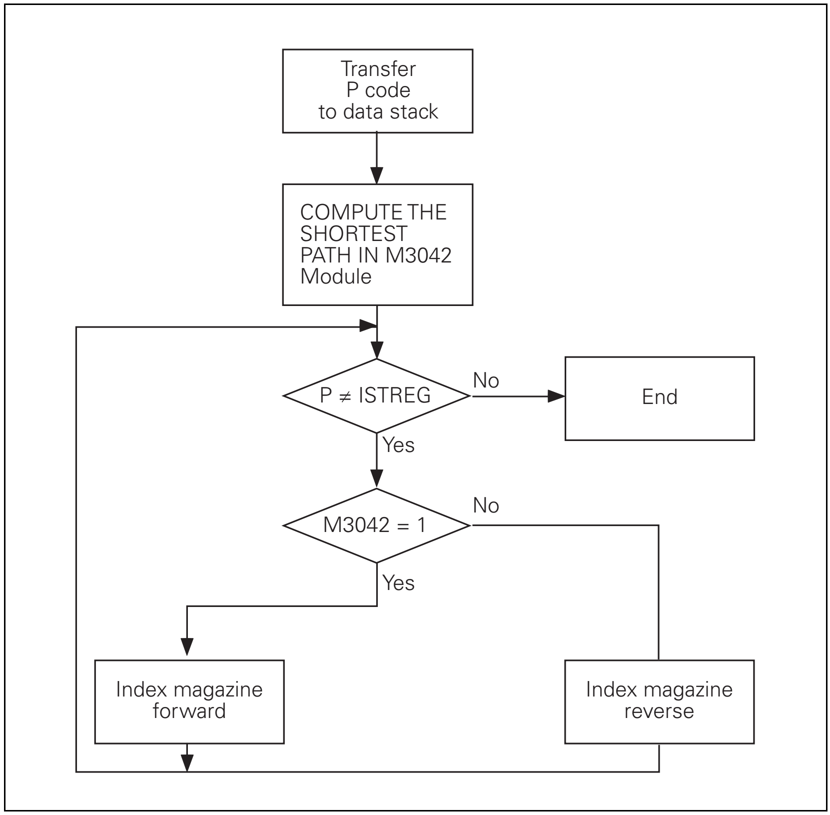
INSERT program module

The spindle is emptied and the new tool is automatically inserted. The program takes into account whether the tool should be returned to its original pocket in the tool magazine (e.g. special tool).



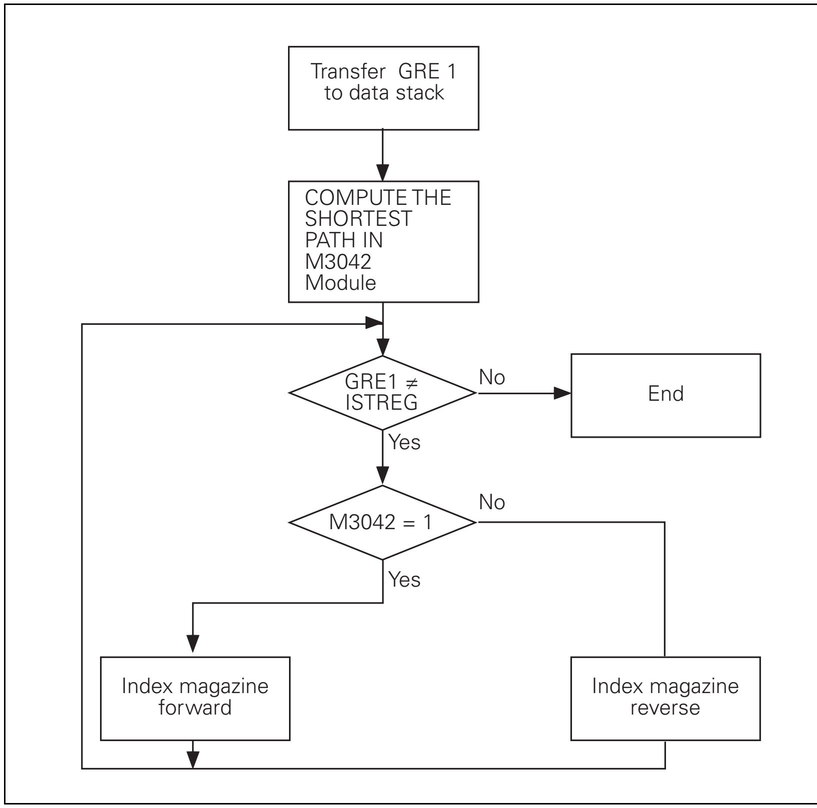
COMPARE P CODE WITH ISTREG program module

The tool magazine is positioned in the shortest direction to the desired pocket number.



COMPARE GRE1 WITH ISTREG program module

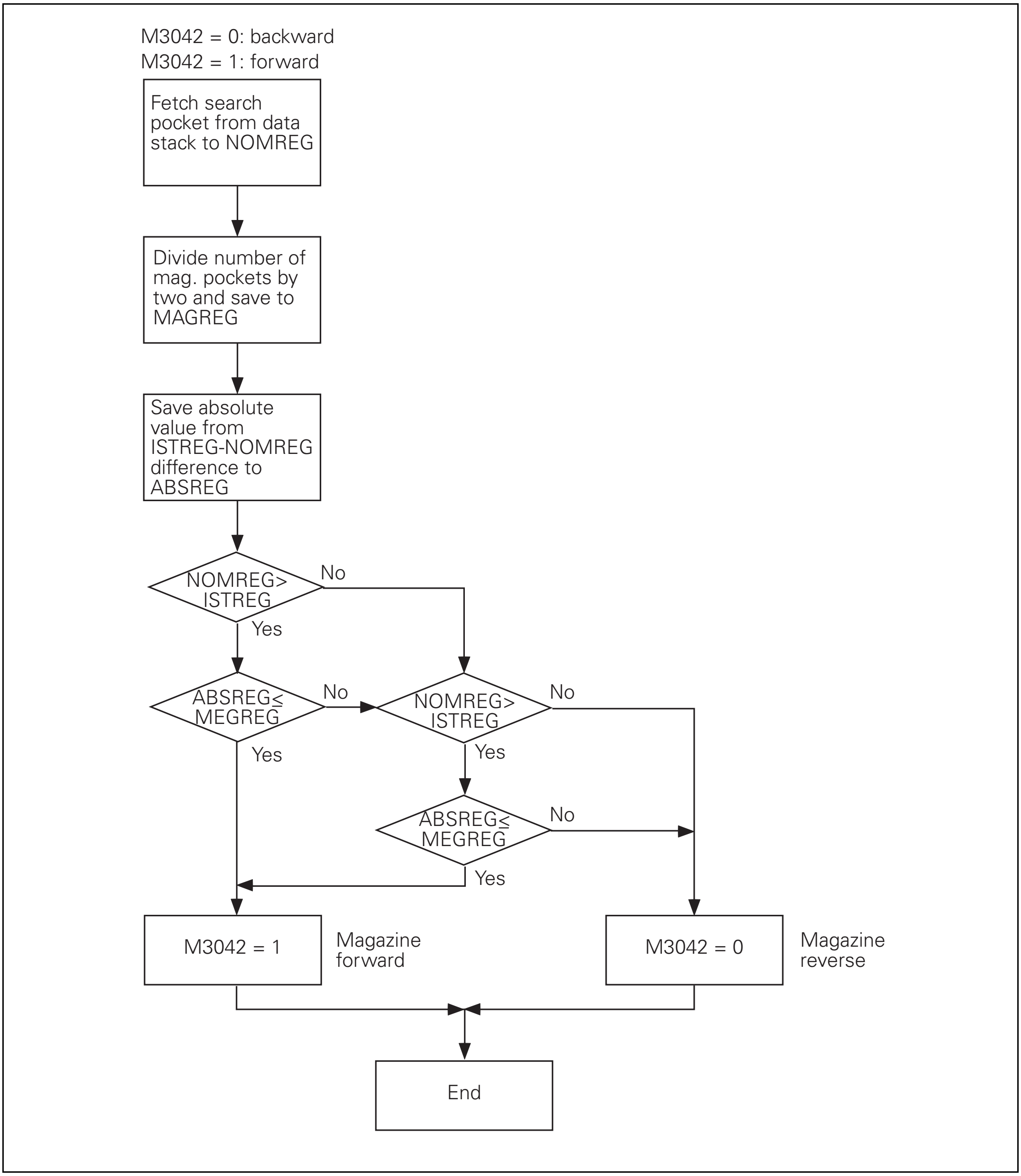
The tool magazine is positioned in the shortest direction to the pocket number that is located in GRE1.



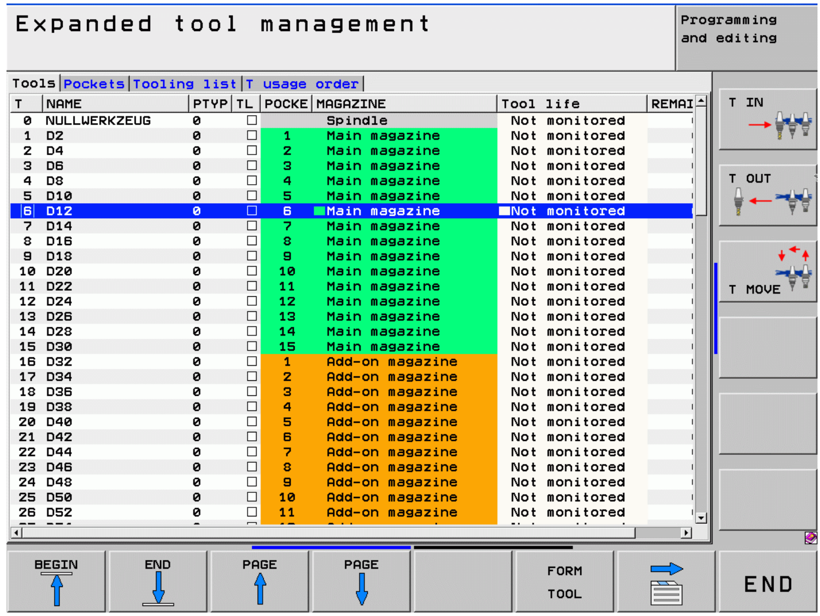
COMPUTE SHORTEST DIRECTION program module

The PLC ascertains the direction of tool-magazine movement for the shortest traverse distance to the desired pocket number.

The direction is saved in M3042:
M3042= 0: Backward
M3042= 1: Forward



8.15.7 Enhanced tool management



With the enhanced tool management, you can provide many functions with regard to tool handling. Examples:

- Easily readable and, if you desired, adaptable representation of the tool data in tables and forms
- Any description of the individual tool data in the new table view
- Mixed representation of data from the tool table and the pocket table
- Fast sorting of all tool data by mouse
- Use of graphic aids, e.g. color coding of fields for different tool or magazine status
- List of all available tools of the current NC program
- Program-specific usage sequence of all tools

The new tool management function is included in the HEIDENHAIN standard features and can be activated any time. However, the previously known tables are active after software 340 49x-06 has been installed.

The enhanced tool management function additionally includes the "tooling list" table and the "T usage order" table as software option #93 Extended Tool Management.

- Tooling list
  List of all tools in the NC program that is selected in the Program Run mode (only if you have already created a tool usage file)
- T usage order
  List of the sequence of all tools that are inserted in the program selected in the Program Run mode (only if you have already created a tool usage file)

You activate the enhanced tool management function through the PYTHON.SYS system file. The PLC:\PYTHON.SYS file links soft-key functions with a Python script, among other things. You will find a more detailed description of the PYTHON.SYS file in the "Python in the iTNC" supplement to the Technical Manual on the HEIDENHAIN FileBase.

The standard PYTHON.SYS provides three options (A, B, C). Each of them activates a different variant of the new tool management function:

- Variant A
  The new tool management function is started/activated directly through the TOOL TABLE soft key
  This always activates the new tool management. The previous tables (tool table, pocket table) are no longer available to the user.
- Variant B
  The new tool management function is started directly through the TOOL TABLE soft key after a reset/activation
  Same as variant A, but the tables of the new tool management function are already loaded in the background when the **Power interrupted** message is acknowledged. The time required for opening the new tables after pressing the TOOL TABLE soft key is reduced, especially for large tool tables and pocket tables.
- Variant C
  The new tool management function is started through the TOOL MANAGEMENT soft key from the tool table, pocket table and the Program Run, Full Sequence mode of operation after a reset/activation
  The known soft keys TOOL TABLE and POCKET TABLE can continue to be used to activate the previous tables. The new tool management function can be started by pressing the TOOL MANAGEMENT soft key in the third soft-key row that appears after pressing the TOOL TABLE, POCKET TABLE soft key in the **Program Run, Full Sequence** mode of operation.

You do not need to activate software option #46 (Python OEM Process) if you want to use one of the above-mentioned standard features.

Note

If the enhanced tool management function is active, a TNC backup can only be restored before the **Power interrupted** message has been acknowledged.
Later the tool table and the pocket table are constantly exchanging data with the Python scripts and are therefore locked for the restore process.

Also, the tool management function can be configured to meet your requirements. The following rule files and configuration files are used for configuration:

- OEMSettings.py
- OEMRuleList.py
- OEMRules.py
- OEMEntryList.py
- OEMEntries.py

The new tool management function is configured using the columns from the previous tool table and pocket table. These remain available in the background. The tool management function mainly changes the user interface. In addition, columns with new contents can be created for the new tool management function by linking the data of the previous table columns (e.g. by calculating new values).

OEMSettings.py

Configuration file for OEM-specific enhanced tool management

In this file, you specify the OEM-specific files and settings. In this way, you can also configure only individual parts of the enhanced tool management function on your own.

You need to save the OEMSettings.py in the directory **`PLC:\Python\ETMC\gui_jh`**. If the file exists, the TNC evaluates the entries of this file. In this file, all OEM-specific entries are marked as comments by default, and are therefore not active. If you activate individual entries, the TNC will use this OEM-specific configuration.

The paths indicated in OEMSettings.py are relative paths to the file itself. You should not change any paths or file names to ensure that the complete project is easily understandable if servicing becomes necessary. Therefore, you must save the OEM-specific files with the paths and files names indicated in OEMSettings.py.

OEM-specific files offer you the possibility of defining the following default settings:

- Rule-based configuration of table columns and tables of the tool table and pocket table. If software option #93 is active, this also applies to the tooling list and T usage order.
- Language-sensitive files for the texts used
- Rule-based configuration of fillable forms
- Soft-key row of magazine management
- Context menu for magazine management

OEMRuleList.py

Rule-based configuration file for tables

Definition of the following views in the rule-based configuration file OEMRuleList.py:

- Tool table
- Pocket table
- Tooling list (software option #93)
- T usage order (software option #93)

The respective views (tables) are defined in the OEMRuleList.py file by a sequence of column configurations (saved in OEMRules.py). The individual tables are configured by specifying the column names. The column names used must match those used in the OEMRules.py file for the definition of the columns.

The individual tables have a permanent identifier (key) that is used to assign the column list to the table:

- **rule_t**: Identifier for tool table
- **rule_p**: Identifier for pocket table
- **rule_u**: Identifier for tooling list
- **rule_s**: Identifier for T usage order

The columns are displayed in the configured view in exactly the sequence in which they were defined.

The structure of the list of desired columns must match that of a Python list. Example of configuring a tool table:

```
from OEMRules import *
rule_t = [
          rule_tt
          ,rule_pocket
          ,rule_time
          ,rule_rtime
          ,rule_mag
          ,rule_tplcbit0
     ,
     ]
range_s = '19'
```

This example results in a tool table that consists of the six specified columns. The structure of the columns is defined in the OEMRules.py file.

HEIDENHAIN recommends creating tables with a maximum of 1500 lines. Basically, more entries are possible, but this affects computing performance and may lead to unnecessary waiting times when accessing tables.

The entry **range_s = '<number>'** specifies the number of precalculated tool calls for the "T usage order" table. Remember: The larger the number of values to be calculated, the higher the load on the control. Look-ahead calculation is possible for up to 50 tools. If you enter a fixed value in '<number>', the number of lines displayed in the table will be exactly the number of lines specified. If you use a fixed value in NC programs with many tool calls, not all of the tool calls of the program might be displayed. This can be avoided by entering a '+' sign in addition to the number, e.g. '30+'. In this way, you can ensure that at least all of the tools (tool calls) programmed in the NC program will be displayed in the "T usage order" table. Otherwise, the entered number of precalculated tool calls will be displayed (here: 30).

OEMRules.py

Rule-based configuration file for columns

Rules are used to define the individual columns in the configuration file OEMRules.py. Using rules, you can use or process the columns of the known tool or pocket table and the two tool usage lists, and display them in the new tool management.

Every column defined by the configuration can be used in multiple views if the possible tables links allow it.

The data of the existing views (tables) are accessed in the same way as table data are usually accessed from Python. The following tokens are currently possible for tool management:

- **TOOL** – active tool table
- **TOOL_P** – active pocket table
- **GEO\channel\0\toolPrep\toolingList** – list of all available tools of the currently selected NC program (software option #93)
- **GEO\channel\0\toolPrep\toolSequence** – tool usage order in the currently selected NC program (software option #93)

Configuration

In general, the JH Python functions make it possible to read the tables in the control line by line. A string called the ID string must be entered for identifying each individual line. An ID string must identify the path to the table and the desired data contents (also see the supplement to the Technical Manual: Python in HEIDENHAIN controls).

Examples:

- The line identified by "T=5" is read from the "TOOL" tool table:
  \\TABLE\\TOOL\\T\\5*
- The length "L" entered in the line identified by "T=5" is read from the "TOOL" tool table:
  \\TABLE\\TOOL\\T\\5\\L
- The complete line identified by "P=1.12" is read from the "TOOL_P" pocket table:
  \\TABLE\\TOOL_P\\P\\1.12*

An asterisk ("*****") at the end of the ID string means that the complete line is read. But you can also specify a certain column, e.g. "L". Then only the cell value of the desired line and column will be returned.

Tool management views

The enhanced tool management function makes it possible to display the previous tables of the control in a different view. The described functions and rules enable you to "build" the new enhanced tool-management tables. The respective views of the new tool-management tables (tools, pockets, tooling list, T usage order, etc.) are each based on an existing table.

- The "Tools" view is based on the table \\TABLE\\TOOL\\
- The "Pockets" view is based on the table \\TABLE\\TOOL_P\\
- The "Tooling list" view is based on the table \\GEO\\channel\\0\\toolPrep\\toolingList
- The "T usage list" view is based on the table \\GEO\\channel\\0\\toolPrep\\toolSequence

This means that the previously known tool table provides the basic information for displaying the "Tool" view, for example. Therefore, the number of lines in the "Tools" view is identical to the number of lines in the tool table. All lines of the tool table are read in order to display the "Tools" view. In the "Tools" view, one line is created for every line read, and the table cells in the view are displayed according to the column rules.

Column rules

A view (table) of the enhanced tool management function is defined column by column by specifying column rules. The column rules are summarized in the OEMRuleList.py file and define the configuration of a complete view (table). In the OEMRules.py file, however, individual columns are defined, which can be used (more than once) in different views. A column rule ultimately results in the value for a table cell of the view. The value of a table cell is defined by entering a data path in the form of an ID string in the respective column definition. Because a column rule does not contain any information about a certain line, a placeholder must be used as a line identifier. A question mark "?" is normally used as a placeholder.

Example:

```
rule_tt = {'value':(('\\TABLE\\TOOL\\T\\?\\T','='),)}
```

Meaning:
This rule results in the value of column 'T' from the tool table. If the "Tools" view is active, the 'T' value from the respective line of the tool table is used in the corresponding table cell of the view, for example. If a view other than "Tools" is active, the value to be entered in the table cell must first be determined by linking the tables.

For more examples of column rules, see page 1596.

Linking tables

You can specify different source data paths in the configuration of a view (e.g. "Tools") in order to display data from different source tables in a view.

Example:
rule_tt = t{'value':(('\\TABLE\\TOOL\\T\\?\\T','='),)}
rule_tp = t{'value':(('\\TABLE\\TOOL_P\\P\\?\\P','='),)}

The tables are linked in the OEMRuleList.py file:
rule_t = [rule_tt, rule_tp]

Meaning:
Two columns are displayed in the "Tools" view:
1st column 'T' from the tool table and
2nd pocket number 'P' of the tool from the pocket table (if present)

In this case, both source tables must contain the column 'T' for the tables to be linked. Also, the value in column 'T' must have the same meaning in both source tables (e.g. tool number). Tables that are not related to each other cannot be linked to each other.

The placeholder **'?'** in the rule "rule_tp" has a completely different meaning from the "Tools" point of view than it has from the "Pockets" point of view:

- rule_tp when used from a "Tools" point of view:
  Show pocket number from the pocket table for the tool currently being read from the tool table
- rule_tp when used from a "Pockets" point of view:
  Show pocket number of the line currently being read from the pocket table

Possible table links

Every view only allows certain links, because the related column values are not always unique.

- Valid data paths for the "Tools" view
  - **\\TABLE\\TOOL** [basic information]
  - **\\TABLE\\TOOL_P**
- Valid data paths for the "Pockets" view
  - **\\TABLE\\TOOL_P** [basic information]
  - **\\TABLE\\TOOL**
- Valid data paths for the "Tooling list" view
  - **\\GEO\\channel\\0\\toolPrep\\toolingList** [basic information]
  - **\\TABLE\\TOOL**
  - **\\TABLE\\TOOL_P**
- Valid data paths for the "T usage list " view
  - **\\GEO\\channel\\0\\toolPrep\\toolSequence** [basic information]
  - **\\TABLE\\TOOL**
  - **\\TABLE\\TOOL_P**

If you are in the Tools" or "Pockets" view, you cannot create a link to the "Tooling list" and "T usage order" tables, because there is no unique value in a common column (here 'T'). The same applies to linking values of the "Tooling list" to values of the "T usage order."

The prerequisite for linking, i.e. the relationship between the "Tools" view and the "Pockets" view, is the unique column 'T' (tool number). However, a tool can occur more than once in the "Tooling list" and "T usage order" views (e.g. the tool is called several times in an NC program).

Column definition: The description of a column of the view (table) consists of multiple configuration settings. The structure of a column description must correspond to that of a Python dictionary:

Entry	Description
'name':	Column identifier that is displayed as the column title (e.g. 'T')
'type':	Data type of values in the column ('float' for numbers, 'str' for string, 'check' for check box). The 'float' and 'str' column types differ only in the type of sorting used (sorting by string or numerical value). The 'check' column type generates a column with check boxes.
'sort':	Conditions for sorting ■ '.blanks_on_top', blank cells at the beginning of the table ■ '.blanks_on_bottom', blank cells at the end of the table ■ '.number_before_letter', numerical values before strings ■ '.letter_before_number', strings before numerical values The numerical values and strings are each sorted separately. The settings for blank cells and number/string sorting can be combined.
'align':	Text alignment in the column ('left'/'right'/'center')
'gettext':	Translate or do not translate the cell content ('value') with the gettext function ('true'/'false')
'value':	Value to be used as cell content. Further rules can be used to adapt the value. The rules are executed sequentially one after the other, thereby enabling you to nest the rules. The last applicable rule determines the displayed value.
optional **'background':**	Color to be used for the background of the cell Rules can be used to define a background color for the cell.
optional **'background-pix':**	A semi-transparent window is shown in the currently selected line to make the line better visible. ('true'/'false')
optional **'condition':**	The value will be displayed or the cell remains blank, depending on a condition. If the result of the condition is 'false', the value will not be displayed and the cell remains blank.

A column definition is evaluated for every read line of the table that provides the basic information. Linking the line to be evaluated to the column definition to be used therefore results in exactly one cell value of the view. This also means that the computing time required for displaying a view increases significantly with the number of columns defined.

The system automatically tries to translate the **'name'** entries by means of the 'gettext' function. For the cell contents (**'value'**), however, the rule **'gettext'='true'** or **'false'** must be used to activate translation.

The language-sensitive *.po files for the gettext function can be created through PLCtext and must be located in the directory **PLC:\LANGUAGE\Python**. For more information about this, refer to the Python texts help topic in the PLCtext help.

If the 'gettext' function does not find a translation for a bookmark into the respective language, the bookmark will automatically be displayed. HEIDENHAIN therefore recommends using the English text for the bookmark. This results in the English text being displayed if no translation is found. You can use the PLC-Text tool for PCs to create the texts for the 'gettext' function.

When defining a column, use the 'type' entry to define the type of column. The 'check' column type generates a column with check boxes.

To show a check box, you must define a rule that returns the result **'true'** for "activated" and **'false'** for "not activated." Alternatively, you can also define a bitmask function using the '&' operator. This function will return **'true'** for "bit is set" and **'false'** for "bit is not set." Another possibility is the definition of a **'true-false'** rule. For this type of rule, the value that activates or does not activate a cell is entered directly.

Example:
('\\{cell}', 'true-false', '-1', '0')
'-1' activates the cell, '0' means "not activated"

To include values of different columns in calculations or to link them with each other, the following logical operations are available:

Operand	Description
<	Less than
<=	Less than or equal to
==	Comparison for "equal to"
!=	Comparison for "not equal to"
>=	Greater than or equal to
>	Greater than
&	Logical AND gating (masking)
=	Accept the value

Operand	Description
+	Addition
–	Subtraction
*	Multiplication
/	Division
*.	Only integer places
.*	Only decimal places
M	The magazine the tool is located in
.P	The pocket in the magazine the tool is located in
TNR	Returns the T number if the tool number is programmed; returns an empty string '' if the tool name is programmed
NOT	Returns the string 'true' if the value is not set; otherwise, 'false' is returned

All of these operations return a character string that is displayed as cell value in the view. More than one rule operation can be defined for the **`'value'`** configuration setting of a column. All conditions defined are evaluated sequentially, and the last valid value determined is loaded as cell value into the view.

Special columns

The check-box column plays a special role in tabular views. A check box is shown active/inactive if:

- The value calculated from the rule operation is **`'true'`** or **`'false'`** (comparison rules except **`'true-false'`** and **`'&'`**).
  Example: **`( '\\TABLE\\TOOL\\T\\?\\TL', '<>', '0', 'true', 'false' )`**
  -> 'TL' <> '0' means "activated"
  'TL' = '0' means "not activated"
- A value that is permanently defined as **`'true'`** or **`'false'`** can be determined (comparison rule **`'true-false'`**).
  Example: **`( '\\TABLE\\TOOL\\T\\?\\TL', 'true-false', '-1', '0' )`**
  -> 'TL' = '-1' means "activated"
  'TL' = '0' means "not activated"
- Masking bits to a numerical value of "bit is set" or "bit is not set" results in (comparison rule **`'&'`**)
  Example: **`( '\\TABLE\\TOOL\\T\\?\\PLC', '&', '1' )`**
  -> Masking bits to the integer value 1 and comparing it to the 'PLC' byte results in:
  Logical '1'. This means "activated."
  Logical '0'. This means "not activated."

Examples of OEMRules.py

Examples of configuring a table column:

The rules are executed sequentially one after the other, thereby enabling you to nest the rules. The last applicable rule determines the displayed value.

1. Example
The column **rule_tt** is filled with the values from column T of the tool table (**'='**):

```
rule_tt = {
          'name'     : 'T'
         ,'type'     : 'float'
         ,'sort'     : 'letter_before_number'
         ,'align'    : 'center'
         ,'get' : 'false'
         ,'value'    : (
                        ('\\TABLE\\TOOL\\T\\?\\T', '=')
          )
     }
```

2. Example
The column **rule_pocket** is filled with the values from column P of the pocket table (**'='**). However, only the pocket number is entered. If the pocket number is equal to 0 (**'==', '0', 'DB_Spindle'**), then the text from **'DB-SPINDLE'** will be displayed.

```
rule_pocket = {
   'name'     : 'DB_POCKET'
   ,'type'    : 'St'
   ,'sort'    : 'number_before_letter,blanks_on_bottom'
   ,'align'   : 'left'
   ,'get' : 'true'
          ,'value'   : (
        (('\\TABLE\\TOOL_P\\P\\?\\P', '.P'), '=')
       ,(('\\TABLE\\TOOL_P\\P\\?\\P', '.P'), '==', '0', 'DB_SPINDLE')
        )
       }
```

3. Example

The column **rule_time** is filled with the values resulting from the subtraction of the values from the CUR.TIME column from the values of the TIME2 column. If the resulting value is less than or equal to 5, then **'DB_TIME_1'** will be displayed. Otherwise, **'DB_TIME_2'** will be displayed. If the calculated value is greater than 10, then **'DB_TIME_3'** will be displayed.

The following rule applies to **'background'**: If the calculated value is less than 5, then the background of the cell will be red. Otherwise, it will be yellow. If the calculated value is greater than 10, then the background of the cell will be green.

```
rule_time = {
   'name'     : 'DB_TIME_AV'
   ,'type'    : 'St'
   ,'get' : 'true'
           ,'value'   : (
         (
('\\TABLE\\TOOL\\T\\?\\TIME2', '-', '\\TABLE\\TOOL\\T\\?\\CUR.TIME')
, '<=', '5', 'DB_TIME_1', 'DB_TIME_2'),
         (
('\\TABLE\\TOOL\\T\\?\\TIME2', '-', '\\TABLE\\TOOL\\T\\?\\CUR.TIME')
, '>', '10', 'DB_TIME_3'),
        )
   ,'background':(
         (
('\\TABLE\\TOOL\\T\\?\\TIME2', '-', '\\TABLE\\TOOL\\T\\?\\CUR.TIME')
, '<=', '5', 'red', 'yellow'),
         (
('\\TABLE\\TOOL\\T\\?\\TIME2', '-', '\\TABLE\\TOOL\\T\\?\\CUR.TIME')
, '>', '10', 'green')
         )

              ,'background-pix' : 'true'
       }
```

4. Example

The column **rule_PLC** is composed of check boxes. A check box is shown as being set if the AND operation performed on the value from the PLC column and the decimal number "1" results in logical "1". The decimal number is converted to the corresponding binary number for comparison.

```
rule_tplcbit0 = {
           'name'     : 'PLC_Bit'
           ,'type'    : 'check'
           ,'get' : 'false'
           ,'value'   : (
                          ('\\TABLE\\TOOL\\T\\?\\PLC', '&', '1')
            )
        }
```

OEMEntries.py

Configuration file for data transfer to the Tool Data form

You can define the value to be shown in every user interface element in the input form. The display of values (TNC data) is only supported for certain user interface elements:

- **`Entry`**: Input field
- **`Checkbutton`**: Check button
- **`ComboBoxEntry`**: Input field with list box

All table values contained in the TNC table, which provides the basic information, can be used as calculation values. Table values from related tables can also be displayed, provided that this is compatible with the table links (see Possible table links of OEMRules.py).

Definition of a data element
A data element shows the contents of a user interface element and is defined by several configuration settings:

Entry	Description
`'widget':`	Type of widget. Widgets of the 'Entry' type are supported.
`'value':`	Value to be used as cell content. Values can only be assigned using the "equal to" operator (**`'='`**)
optional **`'edit':`**	As an option, you can define in the OEMEntries.py whether the user can: edit the entry (**`'edit':('false','=')`**), or cannot edit the entry (**`'edit':('true','=' )`**). If no information is defined for **`'edit'`**, editing is possible. Further rules can be used to create a conditional edit function. The rules are executed sequentially one after the other, thereby enabling you to nest the rules.
optional **`'range':`**	As an option, you can enter permissible input values for the cell contents. The following entries are possible: Numerical input range **`'range':('1 to 99')`** or permissible ASCII characters **`'range:('ABC123')`**. This requires the user to enter a value in the field, which is defined by **`'range'`** as permissible value. All other entries are not accepted.

Special configuration settings
If you want to display an input field with a list box, you need an additional parameter to define the available selection.

- option
  Column of any desired table, all of whose values are made available in the list box of the input field.
  Example of displaying values in a list box:
  `option : ( '\\TABLE\\TTYP\\NR\\?\\NAME' )`
  –> All of the table values contained in the 'NAME' column of the **`'TTYP'`** table are now available in the list box.

Values and operators

Values from the available table cells can be used as calculation values for element rules. Constraints for **'value'**:

- Values can be edited only if the displayed value is from the table that provides the basic information.
- Only the operators '=', '&' and 'true-false' are allowed, because only these operators can ensure that a value is unique. For example, if a sum of two table values is shown in a cell, after an editing process the rule algorithm cannot determine which of the two original table values should be edited.

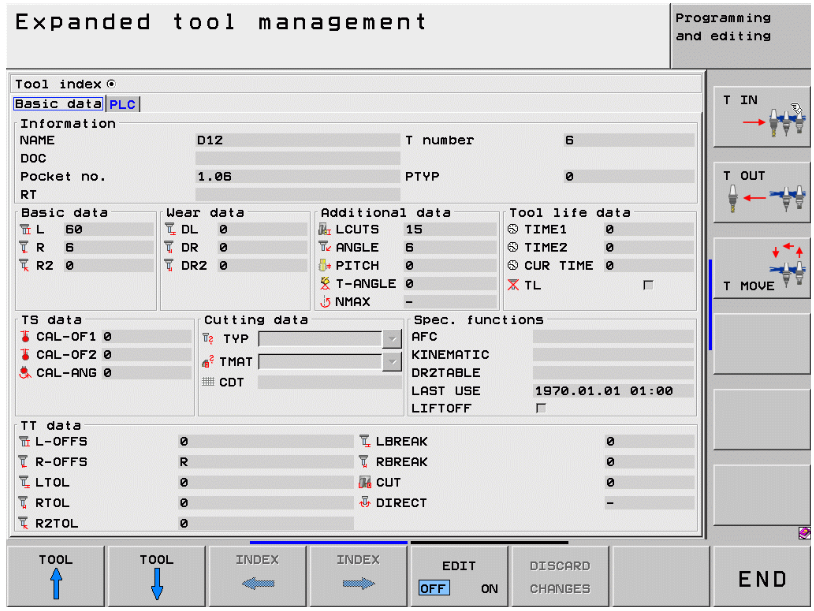
A valid element rule for **'edit'** must return the result **'true'** or **'false'** (for a more detailed description of rules, see "Logical operations" in OEMRules.py)

Example of a OEMEntries.py file:

The rules are executed sequentially one after the other, thereby enabling you to nest the rules. The last applicable rule determines the displayed value.

```
# Tool table

entry_tool_name = {
           'widget'  : 'Entry'
           ,'value'  : ('\\TABLE\\TOOL\\T\\?\\NAME', '=')
           #,'edit'  : ('\\TABLE\\TOOL\\T\\?\\NAME', '!=','','true')
     ,
     }
entry_tool_t = {
           'widget'  : 'Entry'
           ,'value'  : ('\\TABLE\\TOOL\\T\\?\\T', '=')
           ,'edit'  : ('false', '=')
     ,
     }
entry_tool_doc = {
           'widget'  : 'Entry'
           ,'value'  : ('\\TABLE\\TOOL\\T\\?\\DOC', '=')
     ,
     }
entry_toolp_p = {
           'widget'  : 'Entry'
           ,'value'  : ('\\TABLE\\TOOL_P\\P\\?\\P', '=')
           ,'edit'  : ('false', '=')
     ,
     }
checkbutton_tool_tl = {
           'widget'  : 'CheckButton'
           ,'value'  : ('\\TABLE\\TOOL_P\\P\\?\\P', '=')
     ,
     }
comboboxentry_tool_typ = {
           'widget'  : 'ComboBox'
           ,'value'  : ('\\TABLE\\TOOL\\T\\?\\TYP', '=')
     ,
     }
```

OEMEntryList.py **Configuration file for Tool Data form**



In the OEMEntryList.py file, you create the connection between the user interface elements of a data form, such as the tool form, and the data to be displayed (defined in OEMEntries.py). A data form displays a data block or part of a data block from the current view. Additionally, you can define the user interface element that is to have the input focus when the form opens.

Creating a data form

The Glade User Interface Designer (Glade is a free visual programming environment [Rapid Application Development] for creating GTK+ user interfaces graphically and interactively) is used to create a data form for a certain view. The display of TNC data is only supported for certain user interface elements:

- **`gtk.Entry()`**: Input field
- **`gtk.CheckButton()`**: Check button
- **`gtk.ComboBoxEntry()`**: Input field with list box

Assigning data to a user interface element

You must create a connection between the user interface element and the data element to be used so that you can define the data to be displayed in a certain user interface element. This connection is defined using the name of the user interface element. When editing the data form, you can use the "Properties dialog" in "Glade" to define a name for a certain user interface element (also called 'widget'). This name will be assigned to a data element in the OEMEntryList.py file.

Each form definition must be saved under a certain key. The following forms are currently supported:

- **entry_t**: "Tools" data form
- **entry_p**: "Pockets" data form

Special configuration settings:

- **entry_t_focus_element**:
  User interface element from **'entry_t'**, which has the input focus when the "Tools" data form opens.
- **entry_p_focus_element**:
  User interface element from **'entry_p'**, which has the input focus when the "Pockets" data form opens.

Example of a OEMEntryList.py:

The structure of the keys of the form definitions must be correspond to that of a Python dictionary, i.e. a list or sequence of keys<-->value pairs. The key to be specified is the name of the respective widget. The value to be specified is the name of the rule from OEMEntries.py, which defines the data contents of the widget, e.g. the value in the rule identifier **entry_tool_name** is transferred to the widget named **entry_TOOL_NAME**.
For the form concerned, the two entries **entry_x_focus_element** define the widget (user interface element) that has the focus when the form opens.

```
entry_t_focus_element = 'entry_TOOL_NAME'
entry_t = {
           'entry_TOOL_NAME'          : entry_tool_name
           ,'entry_TOOL_T'            : entry_tool_t
           ,'entry_TOOL_DOC'          : entry_tool_doc
           ,'entry_TOOLP_P'           : entry_toolp_p
           ,'checkbutton_TOOL_TL'     : checkbutton_tool_tl
           ,'comboboxentry_TOOL_TYP'  : comboboxentry_tool_typ
       ...
      }

entry_p_focus_element = 'entry_TOOLP_DOC'
entry_p = {
           'entry_TOOL_NAME'          : entry_tool2_name
           ,'entry_TOOL_T'            : entry_toolp_t
           ,'entry_TOOLP_DOC'         : entry_toolp_doc
           ,'entry_TOOLP_P'           : entry_toolp_p
           ,'checkbutton_TOOLP_ST'    : checkbutton_toolp_st
       ...
      }
```

9 PLC Programming

9.1 PLC Functions

The integrated PLC of the iTNC contains its own text editor for creating the list of statements for the PLC program. You enter PLC commands and comments using the keyboard. It's easier, however to create your PLC programs on a PC with the PLC compiler software PLCdesignNT. If you want to use PLCdesignNT, contact HEIDENHAIN.

The iTNC supports you with the COMPILE function, which and checks the PLC program for logical errors, and the TRACE, TABLE and WATCH LIST functions, with which you can check the status of the operands.

Due to symbol information used in common by the main program and the soft-key file, a specific sequence must be followed in order to avoid conflicts when compiling the PLC program.

- Soft-key project file
- Main program of the PLC
- Soft-key project file (2nd compilation procedure for checking the symbol information)

The process memory works with a compiled PLC program up to a size of 512 KB. Every 10.8 ms—the PLC cycle time—the iTNC begins a new PLC scan, i.e. every 10.8 ms the inputs are reread and the outputs are reset. The PLC cycle time can be set with MP7600.1 and ascertained with Module 9196.

Module 9196 Find the PLC cycle time
The PLC cycle time is determined in µs.

Call:
CM 9196
PL D <PLC cycle time in µs>

MP7600.1 Only CC 422: PLC cycle time
= MP7600.1 · Position controller cycle time
= MP7600.0 · MP7600.1 · 0.6 ms
Input: 1 to 20 (recommended input value: 6)

Note

An exact PLC cycle time is only available after the emergency stop test.

9.1.1 Selecting the PLC mode

To select the PLC mode:

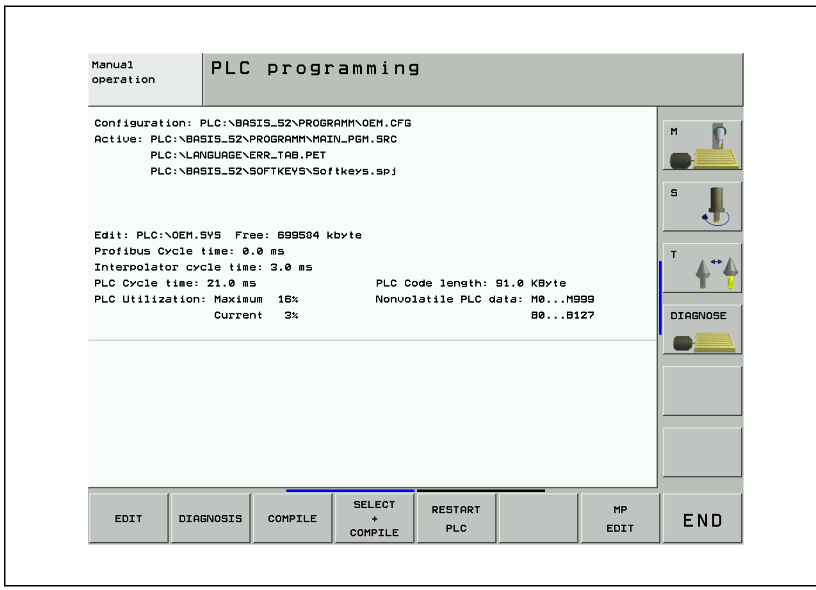
- Select the **`Programming and Editing`** operating mode.
- Press the MOD key.
- Enter the code number 807667 and confirm your entry with the ENT key, or if you already entered the code number, press the PLC EDIT soft key.

Exit PLC mode:

- Press the END soft key or the END key.

9.1.2 PLC main menu

After you have entered the code number (or pressed the PLC EDIT soft key) the iTNC displays the PLC main menu:



Configuration:

Active PLC configuration file. The machine-specific configurations are set in this file.

Active:
Files of the integrated PLC in the process memory. These include the PLC program, PLC error table, soft-key project file and possibly the Profibus configuration file or configuration file for source-code programming.
At start-up, the iTNC automatically compiles the files defined in OEM.SYS. The files only become active after they have been compiled.

Edit:
Name of the file located in RAM memory.

Free:

The available memory in the PLC partition (PLC:\) is shown in kilobytes. Additionally, the PLC partition is checked to see if there are at least 10 MB of memory available. If the available memory is less than 10 MB, the **PLC partition: Not enough memory** error message is output.

Profibus cycle time:

Currently needed cycle time for requests and write-processes of the Profibus system.

Interpolator cycle time:

Current interpolator cycle time.

PLC cycle time:

Current PLC cycle time, set via MP7600.0 and MP7600.1.

PLC utilization Maximum:

Maximum run time of the PLC program
The PLC processing time (time for a PLC cycle) is given as a percentage of the maximum time: 100% is the equivalent of a run time of 1 ms at a cycle time of 21 ms. Use the following formula to calculate the run time t_{run} [ms] in dependence of the PLC cycle time t_{PLC} [ms] and the processing time $t_{calc.}$ [%]:

$$t_{run} = \frac{t_{PLC} \cdot t_{calc.}}{21}$$

If the maximum run time of the sequential program is exceeded, the iTNC displays the blinking error message **PLC: time out.**

PLC utilization Currently:
The time taken for the latest PLC scan in %.

PLC code length:
Length of the compiled sequential program in KB. Maximum value: 512 KB.

PLC functions of the main menu

From the PLC main menu you can use soft keys to access the following PLC functions:

Soft key	Function
EDIT	Edit the file located in RAM memory
DIAGNOSIS	Call diagnostic functions, see page 1276
COMPILE	Compile files registered in OEM. SYS, see page 1619
SELECT + COMPILE	Select and compile files to be compiled, see page 1619
RESTART PLC	Stop and restart the PLC program (M4173 is supported)
MP EDIT	Display a list of machine parameters
WATCH LIST	Display the statuses of selected operands in tabular format, see page 1608
I/O-FORCE LIST	Set inputs and outputs. The PLC program is ignored, see page 1611
TABLE	Display the logical states of the PLC operands, see page 1613
LOGIC DIAGRAM	Show the logic diagram, see page 1617
TRACE IN-CODE	Display the TRACE function, see page 1615
PROCESS MONITOR	Display the process monitor, see page 1766
OSCI	Activate the integrated oscilloscope, see page 987
END	End PLC programming

9.1.3 File management

File management in the PLC mode is largely the same as in the **Programming and Editing** mode of operation (see the iTNC 530 User's Manual). If you press the PGM MGT key while in the PLC mode, the iTNC displays the PLC partition as well, at the upper left next to the TNC partition.

Differences from file management of NC part programs

File types displayed by the iTNC when you press the SELECT TYPE soft key:

Soft key	Function
SHOW ALL	Show all files
.PLC FILES	Show only PLC programs (*.PLC)
SHOW .A	Show only ASCII files (*.A)
.HLP FILES	Show only help files (*.HLP)
.SYS FILES	Show only system files (*.SYS)
.COM FILES	Show only compensation value tables (*.COM)
.CMA FILES	Show only tables with compensation value assignments (*.CMA)
.PET FILES	Show only PLC error tables (*.PET)
.SRC FILES	Show only PLC source files (*.SRC)
.SPJ FILES	Show only soft-key project files (*.SPJ)
END	Return to previous menu

9.1.4 The WATCH LIST function

With the WATCH LIST function you can create a table with dynamic display of the states of the selected operands.

Meaning of the columns in the WATCH LIST:

- **MODULE:** **<Global>** for global symbolic operands or path with the name of the *.SRC file in which the operand is defined
- **SYMBOL:** Symbolic address of the operand
- **ADDR:** Absolute address of the operand
- **VALUE:** Content of the operand
- **COMMENT:** Comment for the operand

Soft keys within the WATCH LIST function:

Soft key	Function
BEGIN ↑	Jump to the beginning of the WATCH LIST
END ↓	Jump to the end of the WATCH LIST
PAGE ↑	Scroll back one page in the WATCH LIST
PAGE ↓	Scroll forward one page in the WATCH LIST
BEGIN LINE ←	Jump to the beginning of the current line
END LINE →	Jump to the end of the current line
HEX ↕ DEZIMAL	Show contents of operands as decimals or hexadecimals
INSERT LINE	Insert a new line above the current line
DELETE LINE	Delete current line

Soft key	Function
SYMBOL LIST	Display a selection list with all symbolic operands used in the active PLC program
BEGIN	Jump to the beginning of the selection list
END	Jump to the end of the selection list
PAGE	Scroll back one page in the selection list
PAGE	Scroll forward one page in the selection list
FIND	Search the selection list for a specific text
SELECT	Load selected operand into the WATCH LIST
END	Close the pop-up window
ORDER	If the cursor is in the **SYMBOL** or **ADDR** column, then this soft key becomes visible, and the column can be sorted alphabetically ascending or descending.
ADD TO LOGIC DIAGRAM	Load selected operands into the logic diagram, see page 1617
FIND	Search text in the WATCH LIST (first the **SYMBOL** column and then the **ADDR** column is searched)
ADD TO I/O-FORCE LIST	Add selected input or output into the I/O Force List, See "The I/O-FORCE LIST" on page 1611
TABLE	Display the logical states of the PLC operands, see page 1613
COLUMN SEQUENCE	Select columns to display in the WATCH LIST and their order
EXECUTE	Assume column settings of the WATCH LIST
END	Close selection window without applying new settings
END	Return to previous menu

Display of symbolic operands in the WATCH LIST

- Press the WATCH LIST soft key to open the WATCH LIST menu.
- Press the SYMBOL LIST soft key to open a list box with all local and global operands used in the PLC program.
- Select the desired operand with the arrow keys and load it with the SELECT soft key or with the ENT key.
- Press the END soft key to close the list box.

> Note
>
> Operands can only be selected with the SYMBOL LIST soft key if you are working with the *.SRC source files of the PLC program on the control. Otherwise the error message **`Selection list is empty`** appears.

Display of operands in the WATCH LIST

- Press the WATCH LIST soft key to open the WATCH LIST menu.
- Press the ZEILE EINFÜGEN soft key.
- In the **`ADDR`** column, enter the absolute address of the operand, i.e. W1022.
- Press the ENT key.

Add operands to the I/O FORCE LIST

- Press the WATCH LIST soft key to open the WATCH LIST menu.
- Press soft key ADD TO I/O-FORCE LIST.

The operand is written to the I/O-FORCE LIST, and a message confirming the transfer of the operand to the I/O-FORCE LIST appears.

Internal process of the WATCH LIST function

`DEBUGPATH = PLC:\DEBUG` is automatically entered in OEM.SYS. This is the working directory for the WATCH LIST function.

If you are working with the source files on the control, a *.WLC file is generated from the *.MAP file when compiling the PLC program. This *.WLC files has the same name as the PLC main program, and contains all local and global symbolic operands. The *.WLC file is saved in the working directory mentioned above for the WATCH LIST function.

The selection window is used to select the symbolic operands from the *.WLC file for the WATCH LIST. The first time operands are selected and a WATCH LIST is created, the file TEMP.WLT is created and saved in the working directory. The entry in OEM.SYS is expanded to **`DEBUGPATH = PLC:\DEBUG\TEMP.WLT.`** This ensures that when the WATCH LIST function is next called, the most recent WATCH LIST will be active.

If there is more than one *.WLT file in the working directory, the desired file can be chosen with PGM MGT. Selecting a new *.WLT file also changes the **`DEBUGPATH =`** entry in OEM.SYS.

9.1.5 The I/O-FORCE LIST

With the I/O-Force List you create a table in which the status of selected **inputs and outputs** is written based on the process image.

Danger

The I/O-FORCE LIST can overrule safety-relevant monitoring operations as well as any inputs and outputs in the PLC program! Activation of the I/O-FORCE LIST can therefore only be done with extreme care. Also make sure that hanging axes are supported.

Meaning of the columns in I/O-FORCE LIST:

- **SYMBOL:** Symbolic address of the operand
- **ADDR:** Absolute address of the operand
- **VALUE:** Content of the operand
- **COMMENT:** Comment for the operand

Adding operands to the I/O-FORCE LIST

- Press the WATCH LIST or TABLE soft key in the **PLC Programming** operating mode.
- **With WATCH LIST:** In the WATCH LIST, select the line with the desired operand (input or output). You may also select the operand via the SYMBOL LIST first.
- **With TABLE:** In the TABLE, select the desired operand via soft key and the keyboard arrow keys.
- Press the ADD TO I/O-FORCE LIST soft key.

The operand is written to the I/O-FORCE LIST, and a message confirming the transfer of the operand to the I/O-FORCE LIST appears.

Note

When using the I/O-FORCE LIST, remember that:

- If an operand is entered and the I/O-FORCE LIST is activated via soft key, then this operand is marked in color in the TABLE, and the **SET** and **RESET** soft keys have no meaning.
- The green error message **I/O-Force is active** is output in the **PLC Programming** mode of operation if the I/O-FORCE LIST is active.

Soft keys in the I/O-Force List:

Soft key	Function
BEGIN	Jump to the beginning of the I/O-Force List
END	Jump to the end of the I/O-Force List
PAGE	Scroll one page backward in the I/O-Force List
PAGE	Scroll one page forward in the I/O-Force List
INSERT LINE	Insert a new line above the current line
DELETE LINE	Delete current line
I/O-FORCE LIST OFF ON	Switch on or off the I/O-Force List. When this soft key is ON, the values entered in VALUE are written via the corresponding inputs or outputs. The states of each input are not evaluated. In addition, the outputs set by the PLC program are not sent to the hardware.
ORDER	If the cursor is in the **SYMBOL** or **ADDR** column, then this soft key becomes visible, and the column can be sorted alphabetically ascending or descending.
END	Return to previous menu

Multiple I/O-Force Lists possible

It is possible to use different lists for the I/O Force function in the PLC operating mode. If an I/O Force List is open, press the PGM MGT key to switch to an existing list or create a new list. Create a new I/O Force List by entering a new file name with the extension .FLT. The selected list is identified by the automatically generated **FORCELISTPATH = PLC:*.FLT** entry in OEM.SYS.

9.1.6 The TABLE function

From the main menu, choose the TABLE soft key to select the table of the PLC memory in order to show its states dynamically on the screen. To select a certain operand, use the cursor keys or the GOTO key. Press the END key to return to the main menu.

Within the TABLE function, you can navigate with the appropriate soft keys, or with the m, i, o, c, t, b, w, d and s keyboard keys for each operand type. The b, c, and d keys are an exception in the B, W, D and HEX views, since they are needed for the entry of hexadecimal values.

In the TABLE function the first two symbols (in alphabetical order) are displayed in the table if in the symbol definition more than one symbol is assigned to the PLC operand through the /c switch.

Soft key	Function
SET	Set the selected operand
RESET	Reset the selected operand
MARKER	Show a list of the markers
INPUT	Show a list of the inputs
OUTPUT	Show a list of the outputs
COUNTER	Show a list of the counters
TIMER	Show a list of the timers
BYTE	Show a list of the bytes
WORD	Show a list of the words
DOUBLE	Show a list of the double words
STRING	List of strings (only the first 70 characters). Overwriting is not possible.
HEX ↕ DEZIMAL	Show contents of operands as decimals or hexadecimals
SAVE M/B/W/D	Save states of selectable operand areas in an ASCII file. Areas of more than one operand can be saved, e.g. M0 to M100, W100 to W118.
RESTORE M/B/W/D	Display saved ASCII file with states of operands
ADD TO WATCH LIST	Load selected operands into the WATCH LIST, see page 1608
END	Return to previous menu

9.1.7 The TRACE function

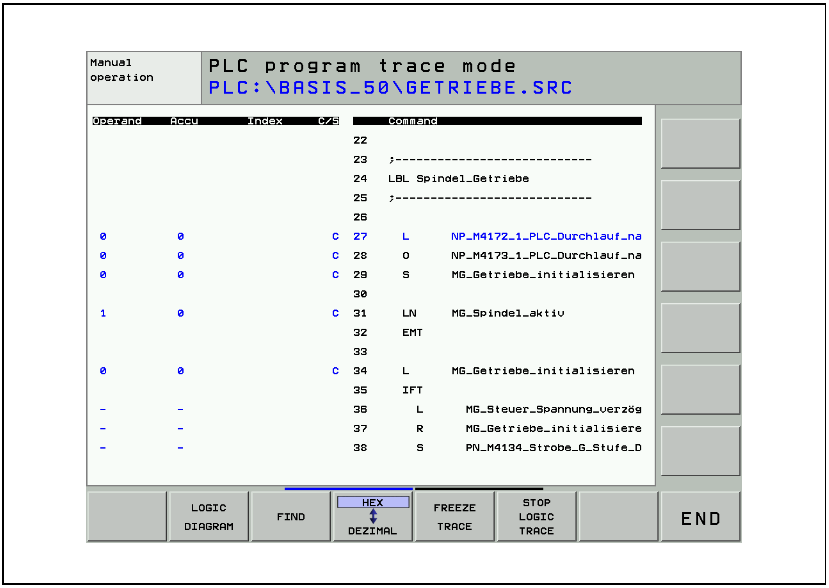
The TRACE function enables you to:

- Control the logical states of markers, inputs, outputs, timers, and counters
- Check the content of bytes, words and double words

From the PLC main menu, press the TRACE IN-CODE soft key to select the TRACE function. The iTNC then shows:

- The statement list (STL) of the selected PLC program
- For every program line, the content of the operand and the accumulator in HEX or decimal code (selectable by soft key)

The iTNC identifies every cyclically executed command with a **C**. With the arrow keys or the GOTO function you can select the program section that the iTNC should display on the screen.
The PLC program to be displayed is chosen with PGM MGT, and must be the currently active main program or a file integrated with USES.



Soft keys within the TRACE function:

Soft key	Function
LOGIC DIAGRAM	Show the logic diagram (, see page 1617)
HEX ⇕ DEZIMAL	Show operand or accumulator contents in hexadecimal or decimal format
START TRACE	Stop dynamic display of the operand content, the accumulator content, and the logic diagram with STOP; continuously update again with START.
FREEZE TRACE	
START LOGIC TRACE	Start/End the trace
STOP LOGIC TRACE	
ADD TO WATCH LIST	Load selected operands into the WATCH LIST, see page 1608
BEGIN ↑	Jump to the beginning of the STL
END ↓	Jump to the end of the STL
PAGE ↑	Scroll back one page in the STL
PAGE ↓	Scroll forward one page in the STL
FIND	Find text in the STL
END	Return to previous menu

9.1.8 The logic diagram

Soft keys within the LOGIC DIAGRAM function:

Soft key	Submenu	Function
SELECT M/I/O/T/C	Select **M**arkers/**I**nputs/**O**utputs/**T**imers/**C**ounters for a logic diagram, trigger logic and recording time.	
	LOGIC DIAGRAM	Return to the recording interface
	SELECT TRIGGER LOGIC	Selecting the trigger logic: Here you specify if recording begins when a change in the signal state (0 or 1) first occurs **(OR)**, or only after all trigger conditions have occurred once **(AND)**.
	SELECT SCAN TIME	Selecting the recording time. Here you specify how long the signal states are recorded from the defined trigger time point. 4 different times are available, depending on the PLC cycle time (a total of 2048 PLC cycles are recorded).
TRACE IN-CODE		Display the TRACE function (, see page 1615)
SAVE TRACE BUFFER		Save current logic diagram in an ASCII file (*.A)
RESTORE TRACE BUFFER		Show the saved logic diagram
START TRACE		Stop dynamic display of the operand content, the accumulator content, and the logic diagram with STOP; continuously update again with START.
FREEZE TRACE		
START LOGIC TRACE		Start/End the trace
STOP LOGIC TRACE		
END		Return to previous menu

With the LOGIC DIAGRAM function you can graphically display the logical states of up to 16 operands (M/I/O/T/C) at once. 2048 PLC cycles are recorded.

The operands to be shown must be saved in a table that you create with the SELECT M/I/O/T/C soft key. The iTNC asks per dialog for the individual positions in the table. To delete the lines, simply press DEL.

For each operand you can enter one trigger condition. The iTNC records from the trigger time for as long as it was instructed to. A total of 2048 PLC cycles is always recorded, and depending on the setting, 128, 256, 512 or 1024 PLC cycles from the trigger time. The appropriate cycles before the trigger time are also shown.

The following trigger conditions are possible:

- 1: Record if operand is logically 1 (trigger on positive edge).
- 0: Record if operand is logically 0 (trigger on negative edge).
- No trigger condition: Only record in combination with other triggers.

If you do not need a trigger condition, you can enter this in the **Trigger** column with the **CE** key. The iTNC records the states of this operand in parallel with the triggered operands.

To start recording:

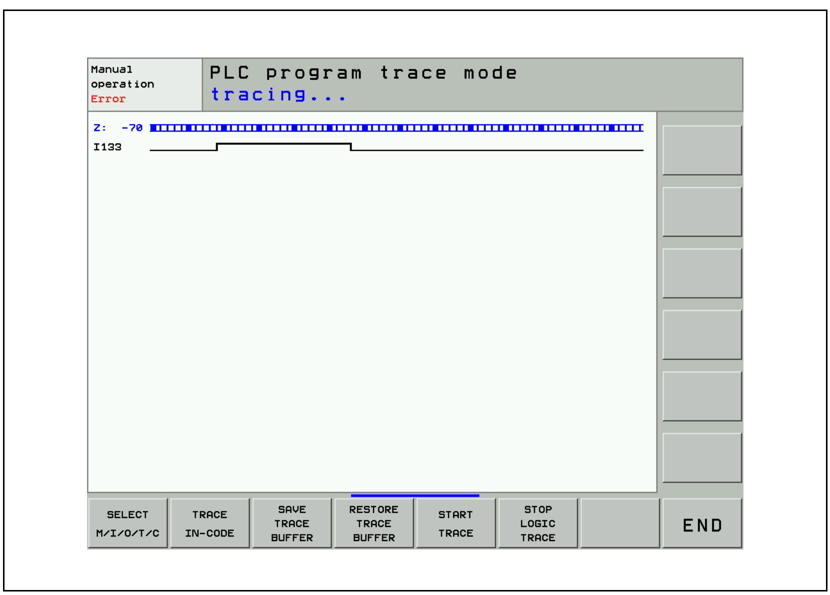
▶ Press the START LOGIC TRACE soft key.

To stop recording:

▶ Press the STOP LOGIC TRACE soft key, or the iTNC terminates recording automatically as soon as the trigger event occurs.

The "PCTR" indicator blinks in the status window as long as the iTNC is recording logical states. As soon as recording ends, you can use the arrow keys to select the desired area in the TRACE buffer.

Example of logic diagram:



9.1.9 The COMPILE function

Compiling a completed PLC program transfers it to the process memory where it can then become active. The name of the compiled program then appears in the main menu next to **PGM IN EXEC.MEM.**

The compiled PLC program is saved in the control, i.e. after confirmation of **power interruption,** the PLC program need not be compiled. It is compiled only if one of the source files has changed. A binary PLC program can be created with PLCdesignNT for test purposes and transferred to the control. This does not change the entry in OEM.SYS.

Soft keys within the COMPILE function:

Soft key	Function
COMPILE ALL	Compile current PLC program, current PLC error table, and current soft-key project file (**PLCMAIN=, PLCERRTAB=,** and **SOFTKEYPROJECT=** entries in OEM.SYS)
COMPILE PLC-MAIN PROGRAM	Compile only the current PLC program (**PLCMAIN=** entry in OEM.SYS)
COMPILE PLC-ERRORTAB.	Compile only the current PLC error table (**PLCERRTAB=** entry in OEM.SYS) (, see page 1165)
COMPILE SOFTKEY-PROJECT	Compile only the current soft-key project file (entry **SOFTKEYPROJECT=** in OEM.SYS), see page 1320
END	Return to PLC main menu

Soft keys within the SELECT + COMPILE function:

Soft key	Function
SELECT COMPILER CONFIG.	Select and compile the configuration file for the compilation of the source code, see page 1621
SELECT PLC-MAIN PROGRAM	Select and compile a PLC program
SELECT PLC-ERRORTAB.	Select and compile a PLC error table, see page 1165
SELECT SOFTKEY-PROJECT	Select and compile a soft-key project file, see page 1320
SELECT MAGAZINE RULES	Select and compile a magazine rule file, see page 1526
END	Return to PLC main menu

For example, to compile any PLC program:

- Press the SELECT + COMPILE soft key: the iTNC displays the soft keys for the SELECT + COMPILE function.
- Press the SELECT PLC-MAIN PROGRAM soft key: the iTNC opens the program manager.
- Use the arrow keys to select the PLC program to be compiled.
- Press the ENT key.

The name and path of the compiled PLC program are entered with **PLCMAIN=** in OEM.SYS.
Press the COMPILE PLC-MAIN PROGRAM soft key if you only want to compile the PLC program from this entry.

PLC Marker M4188 is set at the beginning of each compilation process in the PLC project. Marker M4188 is reset as soon as the initialization of all components involved in the compilation process has been completed and the PLC has been started.

		Set	Reset
M4188	**Compilation process of the PLC project active**	NC	NC

9.2 Conditional Compilation

Depending on the machine parameters, a PLC program can be conditionally compiled on the iTNC. This allows you to select and deselect machine options by entering the options in machine parameters. Therefore, only one PLC program is necessary for all variants of machine options.

You use the PLC configuration file *.cfg also to influence the memory allocation of the control and change the configuration data of the PLC. Enter certain keywords to perform changes. After compilation by the PLC, the NC detects the changed settings, and the control must be restarted.

The values you enter may only be greater than the default values.

Available keywords:

Keyword	Description	Example
DEFINE	Configuration definitions for controlling the conditional compilation, soft-key menu generation and cycle-project configuration	
REMBYTEMIN	Start address of the bytes, words or double words whose data remains stored after a power interruption (remanence). Default value: 0	REMBYTEMIN = 0
REMBYTEMAX	End address of the bytes, words or double words whose data remains stored after a power interruption (remanence). The range defined by REMBYTEMIN and REMBYTEMAX may not be larger than 1024 bytes. Default value: –1 = Deactivated	REMBYTEMAX = 200

Keyword	Description		Example
REMMARKERMIN	Start address of the markers whose data remains stored after a power interruption (remanence). Default value: 0		REMMARKERMIN = 0
REMMARKERMAX	End address of the bytes, words or double words whose data remains stored after a power interruption (remanence). The range defined by REMMARKERMIN and REMMARKERMAX may not consist of more than 2048 markers. Default value: –1 = Deactivated		REMMARKERMAX = 150
MARKERS	A total of 100 000 bytes is available for all keywords, timers, counters and strings combined	Number of markers available. Default value: 10000	MARKERS = 15000
BYTES		Size in bytes for the byte/word/double word memory. Default value: 10000	BYTES = 20000
INPUTS		Number of input markers available. Default value: 384	INPUTS = 450
OUTPUTS		Number of output markers available. Default value: 192	OUTPUTS = 250
INPUTBYTES		Size in bytes for the byte/word/double word memory range used by the Profibus inputs. Default value: 1000	
OUTPUTBYTES		Size in bytes for the byte/word/double word memory range used by the Profibus outputs. Default value: 1000	

Along with the keywords there are also system parameters. System parameters are identified by a preceding and following $ character. Depending on which options have been set, they are defined via the compiler and can be used for conditional compilation. For more information about this, please refer to the PLCdesignNT help.

Available system parameters:

System parameters	Meaning
For interrogation with #ifdef:	
$MULTIFILE$	Multifile option is set
SYM_LABELS	Symbolic labels option is set
$<nc model>$	Current control model, e.g. **#ifdef $TNC407$**
$<nc variante>$	Current NC ID number, e.g. **#ifdef 340490_05**
$<nc basisvar.>$	Current NC basic ID number, e.g. **#ifdef 340490**
For interrogation with #if:	
$<VARIANT>$	Current NC ID number as numerical value, e.g. **#if $VARIANT$ >= 34049005**
$<VARIANTBASE>$	Current NC basic ID number as numerical value, e.g. **#if $VARIANTBASE$ = 340490**
$<VARIANTINDEX>$	Current NC ID number index as numerical value, e.g. **#if $VARIANT$ >= 05**
$<VERSIONBASE>$	Current compiler basic version as numerical value
$<VERSIONINDEX>$	Current compiler index as numerical value

The environmental variable **%GraphicsResolution%** in the PLC configuration file is also available for conditional compilation of the PLC program. The variable is set to the value 1024x768 or 1280x1024, depending on the screen resolution. The content of the environmental variable can be evaluated and used for conditional compilation.

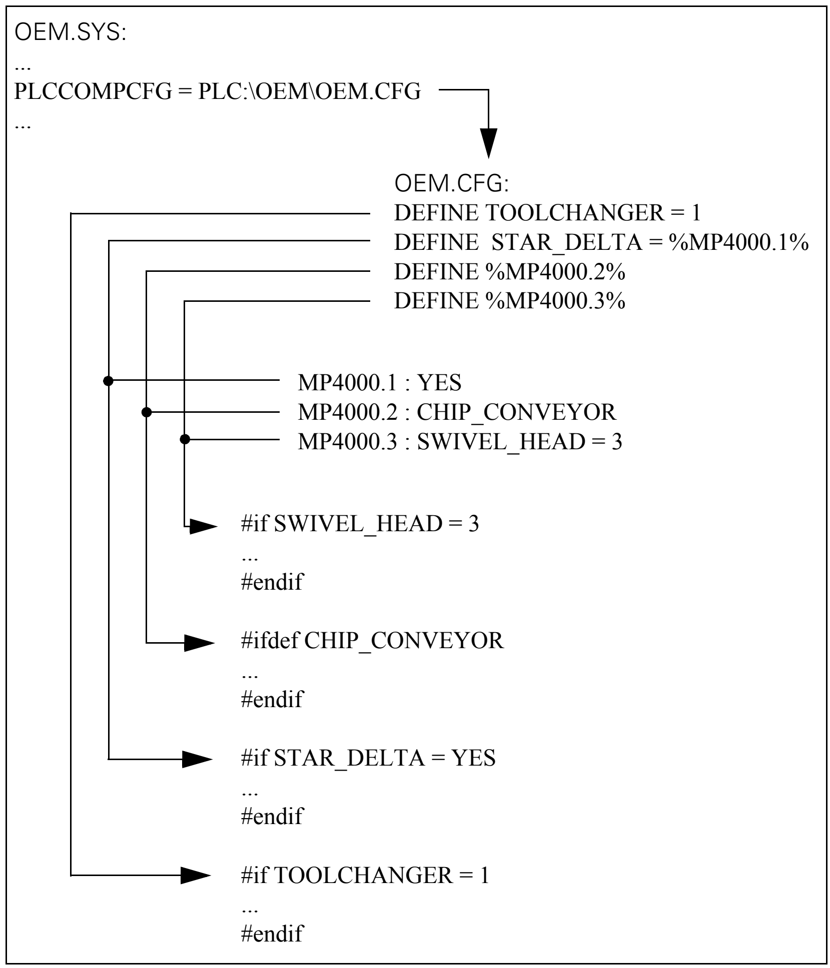
Example:

```
define resolution=%GraphicsResolution%
...
# if resolution=1024x768
...
#endif
...
# if resolution=1024x768
...
#endif
```

Example of a config file:

- Enter the commands for the conditional compilation in the PLC program.
- Create the config file.
- In OEM.SYS, enter **PLCCOMPCFG =** followed by the path for the config file.
- Enter the machine options in the machine parameters MP4000.x.
- Reset the iTNC or recompile the PLC program.

Example:



MP4000.0-63 Options for the conditional compilation of the PLC program

9.3 Hard-Disk Organization

The hard disk of the iTNC is divided into three partitions:

TNC partition

User-specific data such as NC programs, tool tables, datum tables, and pallet tables.

PLC partition

Your OEM-specific data such as system files, PLC programs, machine parameters, help files, PLC dialogs, PLC error tables, compensation value tables and OEM cycles. The PLC partition is visible only after you have entered the code number 807667.

As a machine tool builder, you are concerned primarily with the PLC partition.

SYS partition

System-specific files such as system files, NC dialogs, HEIDENHAIN cycles, etc. The SYS partition is not visible and cannot be selected.

Attention

Alterations to the system partition can impair proper function of the iTNC!

Size of the partitions

The following partition sizes were valid until approximately August 2002:

Partition	Contents	Size
SYS	System files	2 GB
PLC	OEM files	2 GB
TNC	User files	Remaining memory on hard disk (at least 2 GB)

The following sizes are valid for the partitions on hard disks for the iTNC 530 that were delivered from August 2002 up to NC software 340 422-08.

Partition	Contents	Size
SYS	System files	1 GB
PLC	OEM files	1 GB
TNC	User files	4 GB

The following sizes are valid for the partitions on hard disks for the iTNC 530 that were delivered with NC software 340 422-09 to 340 422-12.

Partition	Contents	Size
SYS	System files	1 GB
PLC	OEM files	1 GB
TNC	User files	25 GB

The following sizes are valid for the partitions on hard disks for the iTNC 530 that were delivered with NC software 340 490-01 or higher.

Partition	Contents	Size
SYS	System files	2 GB (for multiple software version in packed format)
PLC	OEM files	1 GB
TNC	User files	25 GB

For partition sizes of the iTNC 530 with Windows 2000 (delivered starting from NC software 340 480-09), See "Hard disk" on page 1848.

Directory structure

HEIDENHAIN recommends creating the following directory structure in the PLC partition:

PLC:\	System files *.SYS
BASIS_33	PLC programs *.PLC (main program and modules)
CORRECT	Compensation value tables *.CMA and *.COM
CYCLE	OEM cycles
JH	Machine parameter description, CycleDesign files
JHSAMPLE	Standard PLC error table *.PET
KINEMAT	Kinematics tables
LANGUAGE	PLC dialogs and error messages *.A; Help files *.HLP
LOGO	OEM logo
MFUNCT	M-function macros
MP	Machine parameter files, motor tables
NC_MACRO	NC macros
NET	Network settings
PROTO	Prototypes for tables
SOFTKEYS	Pictures for PLC soft keys

Note

In the PLC and SYS partition, a maximum of 512 entries each can be stored in the root directory, otherwise an error message appears.

Checking the file system

(Only for single-processor version)
The TEST FILE SYSTEM soft key was introduced in the second soft-key row in the Programming and Editing mode under MOD. It can be used to start the program **dosfsck** in order to check and automatically repair the file system of the PLC and TNC partition of the control's hard disk.

Please ensure that no NC program is being run, and save open files if necessary. When the **dosfsck** program is started, the NC software will be terminated and an EMERGENCY STOP will be triggered as soon as you confirm the associated prompt with the YES soft key. The outputs of the corresponding program **dosfsck** are shown in a window and must be acknowledged with the <Enter> key or OK button. The control then boots again. The outputs are also written to the log. The SYS: partition cannot be checked, but it is checked each time the control is switched on.

9.3.1 Encrypted PLC partition (PLCE:)

As of software version 340 49x-04, the machine manufacturer can create an additional PLC partition, which is automatically encrypted. The **PLCE:** partition enables the machine manufacturer to protect his files and data from unauthorized access.

The encrypted PLC partition **PLCE:** can only be accessed with a password that the OEM chooses. The password is stored in encrypted form either in the OEM.SYS file or in the SIK.

Note

HEIDENHAIN recommends:

- Storing the password in the SIK, because it provides a higher degree of security than the OEM.SYS.
- Using the PLCE partition generally only for specific files that are to be protected from unauthorized access.
- Against using the PLCE partition for system files (such as MP files, NCMACRO.SYS). Please keep in mind that HEIDENHAIN cannot access the data in the encrypted partition, even if servicing becomes necessary.

When creating the PLCE: partition, you must specify the desired memory size (max. 100 MB) of the encrypted partition. Because the PLCE partition saved in the **PLC:\PLCE.BIN** file is a part of the PLC partition, the memory available on the PLC partition is reduced accordingly.

Danger

- When you lose the password for the **PLCE:** partition, the data is permanently lost.
- Deleting the **PLC:\PLCE.BIN** binary file leads to loss of the data on **PLCE:**
- HEIDENHAIN does not have any possibility to access the data on the encrypted **PLCE:** partition. This also applies when servicing becomes necessary.
- The data of the encrypted **PLCE:** partition cannot be written to the service file.
- HEIDENHAIN provides the encrypted PLC partition (PLCE) to its customers. This makes it possible to have all data automatically be encrypted that are copied to the PLCE partition. However, HEIDENHAIN points out that no data encryption method offers 100% protection of the data, especially against access, damage or destruction by unauthorized persons. Therefore, HEIDENHAIN assumes no liability of any kind if the data stored on the PLCE partition is lost, damaged or used by unauthorized persons, nor for any resulting damages.

If the password is stored in the control, the NC software can access the data and files in the **PLCE:** partition at any time. The control can therefore run on the PLCE partition in the same way as on the standard PLC partition.

The NC software automatically searches the PLC partition for files that it needs during run time. If the search is not successful and a PLCE partition exists, the search is continued in the PLCE partition.

This does not apply to the OEM.SYS file itself, or to data and files that are called by using absolute paths (e.g. in PLC modules, OEM.SYS). The OEM.SYS file must always be stored in the PLC partition of the control. If data or files are called by using an absolute path, the NC software only searches the given path for the data or files. The search is not automatically continued in the other partition.

Note

Do not store the same files both in the PLC partition and the PLCE partition.

All data or files on the PLCE partition are automatically encrypted and are therefore protected from unauthorized access. The only exception is the active machine parameter file. If the active MP file is stored in the PLCE partition, it can nevertheless be edited when you switch to the **Machine Parameter Programming** mode of operation.

Backup/Restore and PLCE

The control operator does not need to know the password to back up or restore the PLCE data. The encrypted content remains protected. When restoring a backup on a control, you must make sure that the password on the control (in the SIK or OEM.SYS) is identical to the password that was used for creating the **PLCE.BIN** file or the PLCE partition.

To enable automatic updating of the PLCE partition through a restore process or software update, you must manually separate the PLCE partition before the restore process by using **Unmount.**

You must keep the following in mind to ensure that the PLCE partition is successfully overwritten with the **PLC:\PLCE.BIN** file during a restore process or a software update:

- The password for the PLCE partition is stored in the SIK or OEM.SYS in the control.
- The password in the control matches the password in PLCE.BIN.
- The PLCE partition has been separated (**Unmount**).
- The PLCE partition offers sufficient storage space for the new data.

Software update and PLCE

A software update can either contain the **PLC:\PLCE.BIN** file or the **PLC:_mpupdate\plce.zip** file.

If the software update contains a **PLC:\PLCE.BIN** file, the requirements described above under "Backup/Restore and PLCE" apply as well.

If the software update contains a **PLC:_mpupdate\plce.zip** file, the following requirements apply:

The .zip file must be encrypted with the same password as the **PLCE:** drive. When the control is started up the next time, this file is unpacked and the corresponding files are copied to **PLCE:**. This .zip file can be created using PLCdesignNT with version 2.5 or later.

> Danger
>
> Please note that the encryption algorithm of this .zip file is less secure than encrypting the PLCE partition on the hard disk of the control.

You must keep the following in mind to make sure that the data is transferred to the PLCE partition (with the **PLC:_mpupdate\plce.zip** file) during a software update:

- The password for the PLCE partition is stored in the SIK or OEM.SYS in the control.
- The password in the control matches the password in plce.zip.
- The PLCE partition exists and is mounted (**Mount**).
- The PLCE partition offers sufficient storage space for the new data.

Automatic binary-to-ASCII conversion is not possible for the encrypted **PLCE** partition during a software update. This means that you must manually convert binary-coded files on the PLCE partition to ASCII format before a software update. After the software update, the files must be reconverted back to the binary format. To convert the files manually, proceed as follows:

- Switch to the Machine Parameter Programming mode of operation.
- Press the MOD key and then the UPDATE DATA soft key.
- Use the CONVERT BIN -> ASC soft key to convert the files to ASCII format before a software update. Use the CONVERT ASC -> BIN soft key to convert the files back to binary format after the software update.

In order to avoid converting the files manually, you could also load the binary-coded files onto the PLCE partition after a software update.

PLC project

A plce.sys file can be stored in the PLC:_mpupdate directory. In this file, you can save the information for creating an encrypted PLCE: drive in the following form:
CREATE -s=<size> -p=<Password> [-oemsys] [-force]
This enables the PLC project to create the PLCE partition automatically. You no longer need to create the PLCE partition manually before loading the PLC project. Where:

- <size>
  Size of the drive to be created (in megabytes between 1 MB to 100 MB)
- <Password>
  Encryption password, this password can also be in encrypted form: For example, if a password for an encrypted drive has been created manually, saved in OEM.SYS and then copied from OEM.SYS. The password must be between 12 and 21 characters long.
- [-oemsys]
  If this option is set, the password will be saved in OEM.SYS. If the option is not set, it will be saved in the SIK.
- [-force]
  If this option is set, an existing encrypted drive will be overwritten. If the option is not set, it will not be overwritten.
- Examples:
  CREATE -s=1 -p=VeryLongPassword -oemsys
  CREATE -s=80 -p=ShortPassword -force

ENCCYC. ZIPNAME =

The possibility of machining with or without preset tables in OEM cycles can also be used with the encrypted **PLCE** partition. A separate directory is created on the PLCE partition for each cycle project (PLCE:\OEMCYC_ZIP\, PLCE:\OEMCY2_ZIP\, etc.), just as is done on the PLC partition. In the OEM.SYS, enter after the code word the name of the *.ZIP file to be unpacked. For example, this is for the
PLCE partition: **ENCCYC.ZIPNAME = ABC.ZIP**, **ENCCY2.ZIPNAME = DEF.ZIP**, etc. The *.ZIP files contain all information for the cycles, including the directory structure. When the control is started up, the appropriate *.ZIP files are unpacked in the folders. The documentation for CycleDesign contains more detailed information.

PC tools and PLCE

HEIDENHAIN offers the following PC tools for working with the data of the encrypted partition: PLCdesignNT, PLCtext, CycleDesign and TNCremoNT.

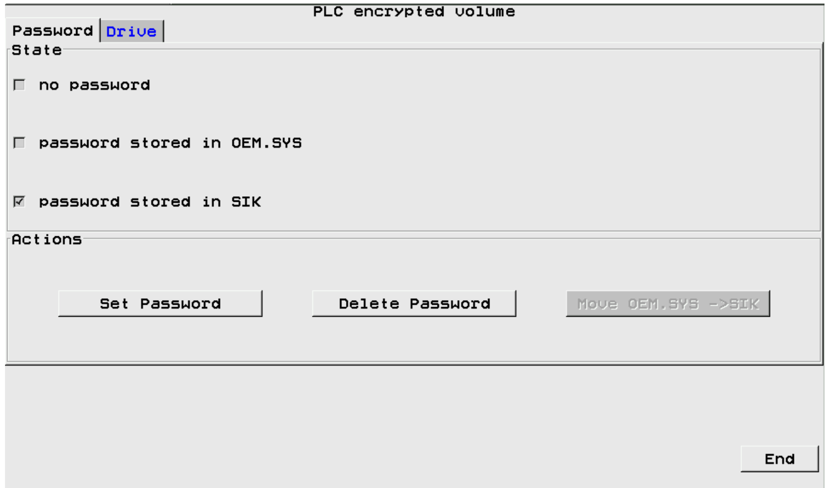
The PLCE partition cannot be accessed until the correct password has been entered.

> Note
>
> At present it is not possible to create an encrypted **PLCE:** partition with the iTNC 530 programming station.

Possible settings

In the PLC mode of operation, press the ADVANCED SETUP soft key and then the ENCRYPTED DRIVE soft key to call the settings of the encrypted partition.

The following information and options are available on the **Password** tab:



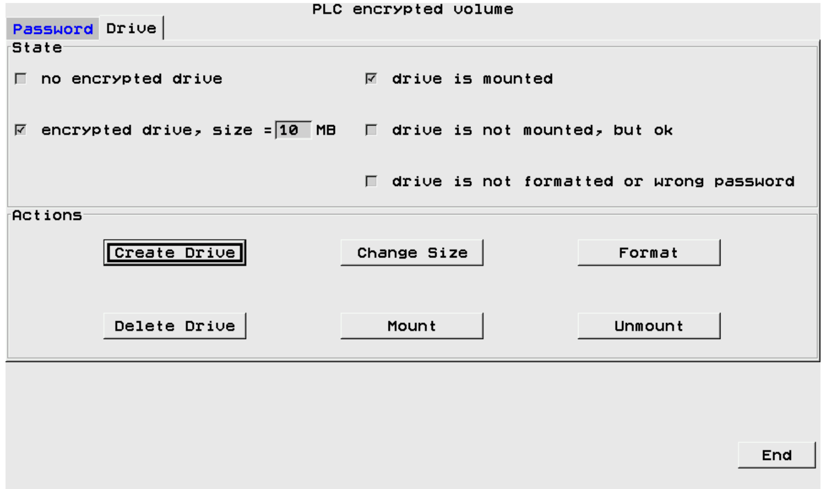
Password – State:

Display	Meaning
No password	Active if you have not yet assigned a password to the **PLCE:** partition.
Password stored in OEM.SYS	Active if the password for the **PLCE:** partition is saved in the OEM.SYS file.
Password stored in SIK	Active if the password for the **PLCE:** partition is saved in the SIK.

Password – Actions:

Button	Meaning
Set password	Assign a password for access to the **PLCE:** partition.
Delete password	Delete the password for access to the **PLCE:** partition.
Move OEM.SYS -> SIK	Move the password for access to the **PLCE:** partition from the OEM.SYS file to the SIK.

The following information and options are available on the **Drive** tab:



Drive – State:

Display	Meaning
No encrypted drive	Active if no **PLCE:** partition has been created yet.
Encrypted drive, size = xxx MB	Active if **PLCE:** partition has been created. The size of the partition is indicated in [MB].
Drive is mounted	Active if **PLCE:** partition is mounted.
Drive is not mounted, but OK	Active if **PLCE:** partition is available, but not mounted.
Drive is not formatted or wrong password	Active if the **PLCE:** partition cannot be used because it is formatted incorrectly, or because the password is incorrect.

Drive – Actions:

Button	Meaning
Create Drive	The encrypted **PLCE:** partition is created and the size of the partition is indicated in [MB]. Only available if no **PLCE:** partition has been created yet. Maximum size: 100 MB Minimum size: 1 MB
Delete Drive	Delete the encrypted **PLCE:** partition. Only available if the **PLCE:** partition is not mounted.
Change Size	Change the size in [MB] of the **PLCE:** partition. Only available if the **PLCE:** partition is not mounted.
Mount	Mount the **PLCE:** partition. Only available if the **PLCE:** partition is not mounted.
Format	Format the **PLCE:** partition. Only available if the **PLCE:** partition is not mounted.
Unmount	Unmount the **PLCE:** partition. Only available if the **PLCE:** partition is mounted.

Create and mount the PLCE partition

1. To assign a password to the encrypted partition, proceed as follows:

▶ While in the **Programming and Editing** operating mode, press the MOD key.

▶ Enter the code number 807 667 to switch to the PLC operating mode.

▶ Press the ADVANCED SETUP soft key.

▶ Press the ENCRYPTED DRIVE soft key.

▶ Press the **Set Password** button under **Actions**.

- In **New Password:** you enter a valid password for access to the encrypted partition.
  The password must be between 12 and 21 characters long. You can use numbers and special characters, but do not use any umlauts. The password is case sensitive.
- Repeat the selected password in **Verification:**
- In **Store Password in** you select whether the password is saved in the SIK or in the OEM.SYS.
- Confirm your entry with the **OK** button.

Note

- When entering the password, keep in mind that the password is case sensitive.
- HEIDENHAIN recommends storing the password in the SIK.

The control confirms the password by displaying the message **Password has been set**, and under **State** it shows where the password is stored.

2. To create an encrypted partition, proceed as follows:

▶ Select the **Drive** tab.

▶ Press the **Create Drive** button under **Actions**.

- In **Size:**, you enter the desired size for the encrypted partition in [MB].
- Confirm your entry with the **OK** button.

The control confirms the creation of the **PLCE:** partition by displaying the message **Encrypted drive has been created** and changes the status under **State** to indicate that the partition has been created.
The right column beneath **State** shows that the partition exists, but that it is not mounted.

3. To mount the encrypted partition, proceed as follows:

- The newly created **PLCE:** partition must first be formatted:
  To perform the formatting process, press the **Format** button and then press the **Apply** button.

> Danger
>
> All data stored on the PLCE: partition are lost during the formatting process.

- To mount the encrypted partition, press the **Mount** button and then press the **Apply** button.

The control confirms that the **PLCE:** partition has been mounted by displaying the message **Encrypted drive has been mounted** and changes the status under **State** to indicate that the partition has been mounted.

Access to PLCE: by the user

The control operator cannot access the mounted **PLCE:** partition until he has entered the correct password for the PLCE partition.

- While in the **Programming and Editing** operating mode, press the MOD key.
- Enter the password for the encrypted **PLCE:** partition.

If you then press the PGM MGT key, the iTNC displays the **PLCE:** partition in the directory structure on the left side of the screen.

If the PLCE partition is displayed in the **File Management,** files and data can be handled in the same way as in all other partitions (e.g. move, delete, rename). Encryption is automatic.

9.4 System Files

9.4.1 OEM.SYS

In the OEM.SYS file you must enter code words to call certain functions. After the code word, and separated by an equal sign **=** you enter the directory in which the files for these functions are to be found, as well as the file names themselves.

You must make your entries in the OEM.SYS file either manually or with Module 9271 (MPFILE and PLCMAIN can only be entered manually). Module 9271 overwrites the content of existing code words and inserts non-existent code words at the end of the OEM.SYS file.

With Module 9270 you can read all entries in the OEM.SYS file.

OEM.SYS is reevaluated during activation of the **machine-parameter programming** operating mode and before downloading a machine parameter file. In connection with TNCremoNT 2.2, during the restoring of a backup, first OEM.SYS is transferred and also evaluated before transferring the machine parameter file.

The following code words are defined (in alphabetical sequence):

AXISNUMBER =

Number of the indexes of the machine parameters (except MP2xxx.y) in the machine-parameter file.

Input example: **AXISNUMBER = 6**

DEBUGPATH =

Path for the most recently active *.wlt file for the WATCH LIST function. Other *.WLT files can also be saved in this folder. The folder is also used as the working directory for the WATCH LIST function. When you select a new *.wlt file, the iTNC automatically enters that name in OEM.SYS.

Input example: **DEBUGPATH = PLC:\DEBUG\TEMP.WLT**

FNERRFIX =

FN14: ERROR = gives additional information about fixing an error. The text and information about the cause of the error (**FNERRREASON =**) are shown after pressing the HELP key. The setup of the file must correspond to that of the file for PLC error messages (**PLCERRFIX =**).

Input example: **FNERRFIX = FN14-FIX.A**

FNERROR =

ASCII file (*.A) that contains the error messages for **FN14: ERROR =** (0 to 299) in lines 1 to 300. The file must be located under PLC:\LANGUAGE\<language>, where <language> depends on MP7230.3.

Input example: **FNERROR = FN14-ERR.A**

FNERRREASON =

FN14: ERROR = gives additional information about the cause of an error. The text and information about fixing the error (**FNERRFIX =**) are shown after pressing the HELP key. The setup of the file must correspond to that of the file for PLC error messages (**PLCERRREASON =**).

Input example: **FNERRREASON = FN14-REASON.A**

KINEMATIC = Path for the assignment table of the tilting-axis geometry description.

Input example: **KINEMATIC = PLC:\KINELIST.TAB**

LOCKEDOPTIONS = You can use the **LOCKEDOPTIONS =** keyword to select software options that will not be enabled with the General Key. <option> is the number of the software option to be disabled. This way you can disable those software options for which your machine is not prepared.

Example entry: **LOCKEDOPTIONS = 44,45**

LOGO = Path for customer-specific company logo during control power-up.

Input example: **LOGO = PLC:\LOGO\OEM-LOGO.BMP**

LOGOSP = Path for a customer-specific company logo shown when the control is started, instead of the note regarding an installed service pack.

Input example: **LOGOSP = PLC:\LOGO\SP-LOGO.BMP**

LSV2TIME0 = Timeout for block reception (STX to ETX).

LSV2TIME1 = Timeout for acknowledging ENQ or check sum.

LSV2TIME2 = Timeout during transmission of DLE 0, DLE 1 or NAK until reception of a valid character.

MODEHELP = Path for help texts and machine commands.

Input example: **MODEHELP = PLC:\LANGUAGE\GERMAN\OPTIMIER.HLP**

MPFILE = (Mandatory entry): Path for the active MP file. If you have loaded an MP file editor and you exit the editor, the iTNC automatically enters this MP file in the OEM.SYS file!

Input example: **MPFILE = PLC:\MP\NC530V02.MP**

MPFRAGMENT <Value> = Up to 10 (<value> from 0 to 9) machine parameter subfiles can be defined. They can be activated with **FN17: SYSWRITE ID1020 NR1 = <value>**. Changes to the spindle machine parameters (MP3xxx or MP13xxx) are only active after an S output. All other machine parameters are active immediately. The changes also remain in effect if a new NC program is selected, but not if the control is restarted.

Input example: **MPFRAGMENT0 = PLC:\MP\FINISH.MP**

> **Note**
>
> Please note that **MPFRAGMENTx** and **MPFRAGMENTFILE...** of course cannot be used as keywords at the same time.

MPFRAGMENTFILE =<path/file name>

By entering **MPFRAGMENTFILE = <path\file name>,** you specify a file containing the paths to machine-parameter subfiles, which can be activated via FN17. The value entered in the **FN17 SYSWRITE ID 1020 NR1 = <line number>** function then corresponds to the MP subfile reference in this file.
This makes it possible to enter any number of MP subfiles (until now only 10 files **MPFRAGMENT0-9 = <path>**).
Changes to spindle machine parameters (MP3xxx or MP13xxx) are only active after an S output. All other machine parameters are active immediately.
The changes also remain in effect if a new NC program is selected, but not if the control is restarted.

Input example: **MPFRAGMENTFILE = PLC:\MP\MPFILELIST.MP**

Note

Please note that **MPFRAGMENTx** and **MPFRAGMENTFILE...** of course cannot be used as keywords at the same time.

MPLOCKFILE =

Path of a machine parameter subfile. If there are differences between this file and the current machine parameter file, an error message appears and the value from this subfile is offered for acceptance.

Input example: **MPLOCKFILE = PLC:\MP\340420.MPL**

MPPASSWORD =

Code number for calling the machine parameter file (instead of 95148).

Input example: **MPPASSWORD = MP**

Note

Do not enter a code number that has already been defined by HEIDENHAIN!

NUMBERMP4111 =

Number of required timers > 96. The corresponding number of machine parameters MP4111.96 to MP4111.x is created.

Input example: **NUMBERMP4111 = 10**
(Machine parameters MP4111.96 to MP4111.105 are created.)

NUMBERMP4230 =

Sets a number in the PLC for Module 9032. The corresponding number of machine parameters MP4230.x is created. The maximum input value is 99; no entry or an invalid entry defines 32 indexes.

Input example: **NUMBERMP4230 = 40**
(Machine parameters MP4230.0 to MP4230.40 are created.)

OEMCYC. ZIPNAME = The possibility of machining with or without preset tables can also be used in OEM cycles. A separate directory is created on the PLC partition for each cycle project (PLC:\OEMCYC_ZIP\, PLC:\OEMCY2_ZIP\, etc.). In OEM.SYS, enter after the keyword the name of the *.ZIP file to be unpacked, e.g. **`OEMCYC.ZIPNAME = ABC.ZIP`**, **`OEMCY2.ZIPNAME = DEF.ZIP`**, etc. The *.ZIP files contain all information for the cycles, including the directory structure. When the control is started up, the appropriate *.ZIP files are unpacked in the folders. The documentation for CycleDesign contains more detailed information.

PALETTPRESET = Activate pallet preset table in order to manage reference points for pallets, see page 1441

Input example: **`PALLETPRESET = TNC:\Palletpreset.pr`**

PLCCOMPCFG = Configuration file for conditional compiling.

Input example: **`PLCCOMPCFG = PLC:\OEM\OEM.CFG`**

PLCDIALOG = Name for text file with PLC dialogs; the path for the text file is permanently defined.

Input example: **`PLCDIALOG = DIALOG.A`**

PLCERRFIX = Path for "Corrective action" help text.

Input example: **`PLCERRFIX = FIX.A`**

PLCERROR = Name for text file with PLC error messages; the path for the text file is permanently defined.

Input example: **`PLCERROR = PLC_ERR.A`**

PLCERRREASON = Path for "Cause of error" help text.

Input example: **`PLCERRREASON = REASON.A`**

PLCERRTAB = (Mandatory entry for PLC error messages): Path for PLC error message table. If you compile a PLC program, the iTNC automatically enters it in the OEM.SYS file.

Input example: **`PLCERRTAB = PLC:\ PLC_PGM \ERR_TAB.PET`**

PLCEVENTS = Path for event list (SPAWN command).

Input example: **`PLCEVENTS = PLC:\EVENTS.PEV`**

PLCMAIN = (Mandatory entry): Path for the active PLC program. If you compile a PLC program, the iTNC automatically enters it in the OEM.SYS file.

Input example: **`PLCMAIN = PLC:\PLC_PGM\MAIN_530.PLC`**

PLCPASSWORD = Code number for calling the PLC mode (instead of 807667).

Input example: **`PLCPASSWORD = 123456789`**

> Note
>
> Do not enter a code number that has already been defined by HEIDENHAIN!

PLCPWM = Path for PLC program for commissioning of digital axes

Input example: **`PLCPWM = PLC:\IB_PGM\IB530.PLC`**

PLCSOFTVERS = (Mandatory entry): iTNC displays PLC software version when the MOD key is pressed.

Input example: **`PLCSOFTVERS = BASIS--33-03`**

PR.LINESLOCKED = Lines in the preset table that are to be write-protected, see page 1195.

Input example: **`PR.LINESLOCKED = 1,4-8,22`**

PRESETTABLE = You can deactivate datum management via the preset table, see page 1195 (only possible through software version 340 49x-02).

Input example: **`PRESETTABLE = OFF`**

PWM PARAMETER = Number of the indexes of machine parameters MP2xxx.y (for the current and speed controller) in the machine-parameter file (up to 30).

Input example: **`PWMPARAMETER = 6`**

REMOTE. LOCKSOFTKEY VISIBLE = Display **`External access ON/OFF`** soft key.

Input example: **`REMOTE.LOCKSOFTKEYVISIBLE = YES`**

REMOTE. PLCPASSWORD FORCED = Setup, machine backup and full backup only with the password from **`PLCPASSWORD =`**

Input example: **`REMOTE.PLCPASSWORDFORCED = YES`**

REMOTE. PLCPASSWORD NEEDED = Access to the PLC partition using the LSV2 protocol only with the password from **`PLCPASSWORD =`**

Input example: **`REMOTE.PLCPASSWORDNEEDED = YES`**

REMOTE. TIMEKEEPIDLE = Network connections for LSV2 applications are cyclically checked. If the remote station does no longer respond (e.g. because the connecting cable was disconnected), the LSV2 connection will automatically be terminated. In the default setting, the connection is checked four times within a ten-second cycle. If the remote station does not respond after the fourth attempt, the connection is automatically terminated.
If you do not want a cycle of ten seconds, you can increase the cycle time to a maximum of 6000 seconds with the **`REMOTE.TIMEKEEPIDLE`** token.

Input example: **`REMOTE.TIMEKEEPIDLE = 250`**

RUNTCH = Indicate a specific pocket table. If this token exists in OEM.SYS, files of the type *.tch are not displayed in the file manager and in the machining modes.

SERVICE. REQUEST. CONTENT =	Only for remote diagnosis with the TeleService PC software: Path of a text file containing the content of the UDP service request package, with which the machine can be identified uniquely. The iTNC reports this content to the TSAgent when the SERVICE soft key is pressed. With short texts it can also be entered directly in quotation marks. If this entry is missing, then the default value PLC:\REQUEST.SYS is used automatically. Input example: `SERVICE.REQUEST.CONTENT = PLC:\REQUEST.SYS`
SERVICE. REQUEST.HOST =	Only for remote diagnosis with the TeleService PC software: IP address of the PC that receives the service request. Input example: `SERVICE.REQUEST.HOST = 160.1.180.99`
SERVICE. REQUEST.PERIOD =	Only for remote diagnosis with the TeleService PC software: Repeat time in seconds for the UDP packets. If this entry is missing, then the default value 10 is used automatically. Input example: `SERVICE.REQUEST.PERIOD = 10`
SERVICE. REQUEST.PORT =	Only for remote diagnosis with the TeleService PC software: Port for the UDP packets. If this entry is missing, then the default value 19001 is used automatically. Input example: `SERVICE.REQUEST.PORT = 19001`
SERVICE. REQUEST. TIMEOUT =	Only for remote diagnosis with the TeleService PC software: Timeout checking interval in minutes—if there has been no data transfer over the Ethernet since the last timeout check, transmission of the service requests is terminated. If this entry is missing, then the default value 15 is used automatically. Input example: `SERVICE.REQUEST.TIMEOUT = 15`
SOFTKEY PROJECT =	Path for PLC soft-key project file *.spj with the structure of the vertical PLC soft key. Input example: `SOFTKEYPROJECT = PLC:\SOFTKEY.SPJ`
SOFTKEYPRESS =	Display and behavior of the CHECK and RADIO types of soft keys. If `SOFTKEYPRESS = ON` is entered, the soft-key graphics and the states of the PLC markers already change when the soft key is pressed, see page 1324. Input example: `SOFTKEYPRESS = ON`
TABCMA =	Path for compensation value tables for axis error compensation (see "Nonlinear axis error compensation" on page 691). Input example: `TABCMA = PLC:\AXIS_COR\CORRECT.CMA`

TCHRULES =

Path for the *.TCR definition file containing magazine rules for tool magazines.

Input example: **TCHRULES = PLC:\RULES.TCR**

TNCOPT.LOCK-SOFTKEYVISIBLE =

After pressing the MOD key, the **TNCOPT ON/OFF** soft key is displayed in order to enable commissioning on the control via TNCopt, see page 1023.

Input example: **TNCOPT.LOCKSOFTKEYVISIBLE = YES**

TTYP =

Path and file name for list of the tool types.

Module 9270: Reading a code word

With Module 9270 you can read an entry from the OEM.SYS file.

Call:
PS B/W/D/K/S<String with code word>
PS B/W/D/K <String number for result>
CM 9270

Error recognition:

Marker	Value	Meaning
M4203	0	Interface enabled
	1	Error. See W1022.
W1022	3	Not a valid string for code word or result
	12	String for code word is too long
	20	Module was not called in a spawn job or submit job
	30	Code word was not found

Module 9271: Writing a code word

With Module 9271 you can write an entry into the OEM.SYS file.

Call:
PS B/W/D/K/S<String with code word>
PS B/W/D/K <String number for result>
CM 9271

Error recognition:

Marker	Value	Meaning
M4203	0	Interface enabled
	1	Error. See W1022.
W1022	3	Entry was written
	6	PLCMAIN or MPFILE was transferred
	12	String for code word is too long
	30	Module was not called in a spawn job or submit job

9.4.2 NCMACRO.SYS

The NC macros are defined in this file. Certain NC macros are predefined. You can also define new NC macros (See "Module 9291 Calling an NC macro" on page 1645).

The following NC macros are predefined:

- TC = <Name of the tool change macro>
- PALETT = <Name of the pallet change macro>
- CLAMP = <Name of the NC macro for changing the fixture (FIX)>
- RUNCANCEL = <Name of the macro called when an NC program is cancelled>
- RESETINIT = <Name of the macro called when **traverse reference point** is left>
  If the macro is not performed completely,
  - you cannot switch to the program run operating modes,
  - the error message **Machine not initialized** appears, and
  - the soft key **INIT** appears. The soft key can be used to restart the macro.
- STARTUPCANCEL = <Name of the macro called when mid-program startup is not completed with **Restore machine status**>
- TCSIMU = <Name of the tool-change macro that is called in the Test Run mode for a tool change. As with the TOOL CALL macro for the machine operating modes, a TOOL CALL block must be executed within a macro in order to activate the new tool>

Example entry: TC=PLC:\NC_MACRO\TOOLCALL.H

In order to increase the speed with which NC macros are executed, limit-switch monitoring can be turned off with **FN17: SYSWRITE ID230 NR5.** At the end of an NC macro the limit switch monitoring is always switched on. For transferring Q parameters between NC programs and NC macros, see page 1648.

PGM CALL, including NC macros, and **CYCL CALL** (for cycles greater than 68) are calculated automatically with the look-ahead function and run without exact stop. At the beginning and end of the called program or cycle, it can happen that a missing synchronization between machine status and look-ahead calculation may lead to problems.
Example:
A **TOOL CALL** is run in look-ahead calculation. In this **TOOL CALL** a PLC function is needed (e.g. opening the tool changer gate). The tool is automatically changed on the machine. During this time the look-ahead calculation reaches another **TOOL CALL.** Since the PLC function has been fulfilled (the tool changer gate is open), the look-ahead calculation is continued. After the first tool change has been completed, the PLC function is no longer fulfilled (the tool changer gate is closed). The second **TOOL CALL** would be executed if the PLC function were not fulfilled (the tool changer gate is closed).
The function **FN20: WAIT FOR SYNC** provides a remedy for this problem. If this function is programmed at the beginning of an NC program (NC macro) or cycle, in the look-ahead calculation the **PGM CALL** (NC macro call) or **CYCLE CALL** is not executed until the calling program has actually reached the **PGM CALL** (NC macro call) or **CYCL CALL.**
Look-ahead is halted for **FN20: WAIT FOR SYNC,** no matter at what point this block is programmed.

Module 9291 Calling an NC macro

With Module 9291, you can call an NC macro in any operating mode. They are executed like cycles, without block display. The control-in-operation symbol is displayed while the macro is being executed. No macros can be activated if there is currently an **`External emergency stop`** error message.
The predefined code words of the NCMACRO.SYS file and the code words defined by the user can be transferred. They only need to be entered in NCMACRO.SYS to be defined. To prevent name conflicts with future HEIDENHAIN code words, your code words should begin with the character "P$" or with the name of the company.

PLC Module 9291 starts an NC macro with the same mechanisms as program selection and starting in the user interface, i.e. with the selection of the NC macro the same modal settings are reset as with PGM MGT and Enter. This means that the continuation of an already started NC program must be prevented after Module 9291 has been called.

Call:
PS B/W/D/K/S<Keyword>
CM 9291

Error recognition:

Marker	Value	Meaning
M4203	0	NC macro was executed
	1	Error code in W1022
W1022	2	NCMACRO.SYS does not exist, code word does not exist, or invalid string
	7	Macro cannot be executed.
	8	External emergency stop is active
	20	Module was not called in a spawn job or submit job
	28	NC program or other macro is already running
	29	The file given under the keyword is not an NC program (*.H or *.I)
	36	The file given under the keyword does not exist

9.4.3 MGROUPS.SYS

In the system files PLC:\MGROUPS.SYS and PLC:\MSPLIT.SYS, you define the M functions to be output after a block scan (See "Returning to the contour" on page 1225).

9.4.4 MSPLIT.SYS

M functions that are effective in several groups are divided in the MSPLIT.SYS file into function components (See "Returning to the contour" on page 1225).

9.4.5 PLCSOFTK.SYS

Path for the file names of the PLC soft-key pictures. (See "PLC Soft Keys" on page1320.)

9.4.6 CYCLE.SYS

Definition of the soft-key structure, if you have integrated OEM cycles. This file is created automatically by the PC software CycleDesign (see OEMCYC directory).

9.4.7 TNC.SYS

The end user can define certain paths and functions in the TNC:\TNC.SYS file:

- TMAT = <Path for list of tool materials>
- WMAT = <Path for list of workpiece materials>
- PCDT = <Path for cutting data tables>
- REMOTE.TNCPASSWORD = <Password for LSV2 access>
- REMOTE.TNCPRIVATEPATH = <Path to be protected by the password>
- REMOTE.PERMISSION = <computer>
  External access to the control via LSV2 can be limited to specified computers.
  If the entry **`REMOTE.PERMISSION = <computer>;<computer>`** exists in the TNC.SYS file, then only the computers listed will be granted access to the control. The entry can be either the names or IP addresses of computers, and must be separated by semicolons. Access to the control is refused if the REMOTE.PERMISSION entry exists in the TNC.SYS file but the computer is not listed. If the REMOTE.PERMISSION entry does not exist, then every computer can access the control.

9.5 Data Transfer NC > PLC, PLC > NC

Information is exchanged between PLC and NC by markers, bytes, words and double words. The function of the individual markers, bytes, words and double words is fixed.

The transfer of certain data to the PLC is controlled by strobes:

- M codes
- S codes
- T codes
- G codes
- Q codes

Example:

If an M function is output, the NC sets the strobe signal M4072. After evaluating the M function, you must first set the acknowledgement marker M4092 and then reset it. The PLC must then reset the acknowledgment marker, otherwise no further strobes can be sent by the NC.

Note

- PLC messages that would terminate an NC strobe are only released once all PLC functions that are only permitted during the strobe have been completed.
- If there is an emergency stop, current NC strobes are automatically acknowledged by the NC, and in this case should not be acknowledged additionally by the PLC.

9.5.1 Data transfer of NC program > PLC (FN19: PLC =)

With the Q-parameter function **FN19: PLC =** you can transfer two values from an NC program to the PLC. The iTNC stores the transferred values as integer values of the form 1/10 000 in the double words D280 and D284. M4570 defines the unit of measure of both numerical values. During transfer, the marker M4075 is set by the NC. The PLC must acknowledge the transfer by setting marker M4095.

		Set	Reset
M4075	**Transfer active with FN19**	NC	NC
M4095	**Acknowledgment of transfer with FN19**	PLC	PLC
M4570	**Unit of measure for transfer with FN19** 0: mm 1: inches	NC	NC
D280	**First numerical value from FN19**	NC	NC
D284	**Second numerical value from FN19**	NC	NC

9.5.2 Data transfer PLC > NC program (Q-parameters)

Data transfer from the PLC to the NC program goes through Q parameters Q100 to Q107, i.e., from the PLC you can overwrite Q parameters Q100 to Q107:

- In double word D528, enter the numerical value to be transferred.
- In word W516, define the target parameter (0=Q100, 7=Q107).
- Activate transfer with strobe marker M4131.
- The iTNC transfers the values with the next strobe.

		Set	Reset
M4131	**Activation of Q-parameter transfer to the NC; data from D258, Q number from W516**	PLC	NC
D528	**Double word with multiple function, here data for transfer from PLC to NC**	PLC	PLC
W516	**Q no. 0-7 for numerical data transfer PLC → NC**	PLC	PLC

Q parameters in OEM cycles are only effective locally if the NC program of the OEM cycle is saved in the *.CYC format, otherwise (*.H) the Q parameters are effective globally.

Cycle 12 (PGM CALL) acts like **`CALL PGM,`** meaning the Q parameters are effective globally.

Q parameter	Effectiveness/Meaning
Q0 to Q99	Freely applicable parameters, as long as no overlapping with SL cycles can occur, globally effective for all programs stored in the TNC memory
Q100 to Q199	Parameters for special TNC functions
Q200 to Q1199	Parameters that are primarily used for cycles, globally effective for all programs stored in the TNC memory
Q1199 to Q1399	Parameters that should primarily be used for OEM cycles, and are globally effective for all programs stored in the TNC memory. This may require coordination with third-party suppliers.
Q1400 to Q1499	Reserved for OEM cycles (CALL-active) Parameters that are primarily used for call-active OEM cycles, and are globally effective for all programs that are stored in the TNC memory.
Q1500 to Q1599	Reserved for OEM cycles (DEF-active) Parameters that are primarily used for DEF-active OEM cycles, globally effective for all programs that are stored in the TNC memory
Q1600 to Q1999	Freely applicable parameters, globally effective for all programs stored in the TNC memory
QL0 to QL499	Freely usable QL parameters, only effective locally (within a program)
QR0 to QR499	Freely usable QR parameters that are nonvolatile, i.e. they remain in effect even after a power interruption

QS parameters (the S stands for string) are also available on the TNC and enable you to process texts. In principle, the same ranges are available for QS parameters as for Q parameters (see table above).

Note

- Some Q and QS parameters are always assigned the same data by the TNC. For example, Q108 is always assigned the current tool radius (please also note the list of preassigned Q parameters).
- If you are using the parameters Q60 to Q99 in encoded OEM cycles, define via MP7251 whether the parameters are only to be used locally in the OEM cycles (.CYC file), or may be used globally.
- With MP7300 you specify whether the TNC should reset Q parameters at the end of the program, or if the values should be saved. Make sure that this setting does not have any effect on Q-parameter programs!
- Note that for the QS parameters the QS100 to QS199 range is reserved for internal texts.

MP7251 **Number of global Q parameters starting from Q99 (up to Q60) that are transferred from the OEM cycle to the calling program.**

Input: 0 to 40

9.5.3 Data transfer NC program > NC (FN17: SYSWRITE)

You can use the FN17: SYSWRITE function particularly for OEM cycles if you wish to overwrite certain NC data, e.g., an active basic rotation, from the NC program. A group number, a system data number, and an index specify the particular item of system data that you write: FN17: SYSWRITE IDxxxx NRxxxx IDXxxxx = Qxxx or numerical value; comment. In the NC program you cannot define function FN17 (soft keys: Q-parameter programming, special functions) until you have entered the code number 555 343. After a control reset the code number must be entered again if you want to program FN17. The iTNC provides the following functions:

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Switch the spindle				
	20	13	–	0 = Spindle 1 1 = Spindle 2
		18	–	Second axis-specific rapid traverse
Data from the tool table				
	50	1	Tool no.	Tool length L
		2	Tool no.	Tool radius R
		3	Tool no.	Tool radius R2
		4	Tool no.	Oversize in tool length DL
		5	Tool no.	Oversize in tool radius DR
		6	Tool no.	Oversize for tool radius DR2
		7	Tool no.	Tool locked TL 0 = not locked, 1 = locked
		8	Tool no.	Number of replacement tool RT
		9	Tool no.	Maximum tool age TIME1
		10	Tool no.	Maximum tool age TIME2
		11	Tool no.	Current tool age CUR. TIME
		12	Tool no.	PLC status
		13	Tool no.	Maximum tooth length LCUTS
		14	Tool no.	Maximum plunge angle ANGLE
		15	Tool no.	TT: Number of teeth CUT
		16	Tool no.	TT: Wear tolerance in length LTOL
		17	Tool no.	TT: Wear tolerance in radius RTOL
		18	Tool no.	TT: Direction of rotation DIRECT 0 = positive, –1 = negative
		19	Tool no.	TT: Offset in plane R-OFFS R = 99999.9999
		20	Tool no.	TT: Offset in length L-OFFS
		21	Tool no.	TT: Break tolerance for length LBREAK
		22	Tool no.	TT: Break tolerance in radius RBREAK
		23	Tool no.	PLC value
		24	Tool no.	Center misalignment in reference axis CAL-OF1
		25	Tool no.	Probe center offset in minor axis CAL-OF2
		26	Tool no.	Spindle angle for calibration CAL-ANG

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
		27	Tool no.	Tool type for pocket table
		28	Tool no.	Maximum speed NMAX
		29	Tool no.	PLC value P1
		30	Tool no.	PLC value P2
		31	Tool no.	PLC value P3
		32	Tool no.	Point angle for DRILL and CSINK T-ANGLE
		33	Tool no.	Thread pitch for TAP PITCH
Edit multiple-task tool				
	53	See FN 17-ID50	Tool no.	Completely edit the data of a multiple-task tool for all indexes. Behaves like FN17-ID50, only that here the same value is written to all indexes.
		7	–	Completely disable all indexes of the current multiple-task tool: `FN 17: SYSWRITE ID 53 NR7 = 1.0`
Coordinate transformations				
	210	1	–	Basic rotation (manual)
		3	–	Active mirror axis Bits 0 to 2 and 6 to 8: Axes X, Y, Z and U, V, W
		6	–	Tilt working plane in Program Run mode (0 = inactive, –1 = active)
		7	–	Tilt working plane in Manual mode (0 = inactive, –1 = active)
Exchange tool axis				
	212	–	–	0: Tool axis Z 1: Tool axis X 2: Tool axis Y 3: Tool axis from TOOL CALL

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Traverse range				
	230	2	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Negative software limit switches
		3	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Positive software limit switches
		4	Number of axes whose software limit switches are to be overwritten	Number of the first of several consecutive Q parameters 1st Q: Neg. limit switch in 1st axis 2nd Q: Pos. limit switch in 1st axis 3rd Q: Neg. limit switch in 2nd axis etc.
		5	–	Limit-switch monitoring (1 = off, 0 = on)
Swivel axes				
	290	1	–	Description of swivel-axis geometry
		5	–	0: Temperature compensation not active in the kinematics table 1: Temperature compensation active in the kinematics table

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
TS touch-trigger probe				
	350	10	–	Tool axis
		11	–	Effective radius
		12	–	Effective length
		13	–	Calibration ring radius
		14	1	Center offset (reference axis)
			2	Center offset (minor axis)
		15	–	Direction of center offset
TT touch probe for tool measurement				
	350	20	1	Center in axis 1
			2	Center in axis 2
			3	Center in axis 3
		21	–	Effective radius
		22	1	Probing position 1 in axis X
			2	Probing position 1 in axis Y
			3	Probing position 1 in axis Z
		23	1	Probing position 2 in axis X
			2	Probing position 2 in axis Y
			3	Probing position 2 in axis Z
		24	1	Probing position 3 in axis X
			2	Probing position 3 in axis Y
			3	Probing position 3 in axis Z
		25	1	Probing position 4 in axis X
			2	Probing position 4 in axis Y
			3	Probing position 4 in axis Z
Coordinate transformations				
	420	0	0	0 = Globally effective
Write values into active datum table				
	500	Line	Column	Depends on MP7475
	501	Line	Column	

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Write values into the active preset table				
	502	Line	Column	Write value with conversion of the currently active coordinate system into the preset table
	503	Line	Column	Write value into the preset table without conversion
	504	Line	Column	Write basic rotation into the preset table
	530	1	–	Activate preset
Pallet preset table				
	540	1	–	Active line in the pallet preset table
Save machine statuses				
	590	1	1 to 9	"Variables" 1 to 9 can only be read with **FN18: SYSREAD ID590 NR1.** Deletion depends on MP7300.
		2	1 to 9 (X, Y, Z, A, B, C, U, V, W)	The "variables" for ID621. Independent of MP7300.
		3	1 to 9 (X, Y, Z, A, B, C, U, V, W)	The "variables" for ID621. Independent of a power failure.
	621	0	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Stop measurement of the dynamic load and save the result.
Velocity semifeedforward control				
	600	1	Axis	Factor for velocity semifeedforward
		2	0 or NO ENT	Use factor from MP1396.x

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Adaptive Feed Control (AFC)				
	622	0	0.0	0: AFC not active 1: AFC active
		0	1.0	Time in seconds after which the teach-in cuts of the AFC are automatically ended. 0: Function not active
		0	3.0	Definition of reference power in [%] of the rated power output of the spindle
		0	4.0	Monitoring of tool wear: Definition of a monitoring limit in [%] of the reference power PLC Marker 4510 is set if the monitoring value is exceeded.
		0	5.0	Spindle load monitoring (cut-based): Definition of a monitoring limit in [%] of the reference power An NC stop is triggered if the monitoring value is exceeded.
Spindle load monitoring independent of cutting				
	622	Filter constant: 1 to 60 ms	6.0	Spindle load monitoring (independent of cutting): Definition of monitoring limit in [%] of the rated power output of the spindle An NC stop is triggered if the monitoring value is exceeded.
Touch-probe cycles				
	990	1	–	Approach behavior: 0 = standard behavior 1 = effective radius, set-up clearance is zero
		2	–	0.0= Touch probe monitoring off, M4057 not used 1.0 = Touch probe monitoring on, M4057 not used 2.0 = Touch probe monitoring off, M4057 used 3.0 = Touch probe monitoring on, M4057 used
		3	–	Place probe data of the manual probing cycles into the tool table
		6	–	Touch probe cycle 3 0.0 = input X12 1.0 = input X13

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Coordinate transformation				
	990	4	1	Transformation from the coordinate system in Manual mode to the active coordinate system (e.g. rotated, shifted).
			2	Transformation of the active coordinate system (e.g. rotated, shifted) into the coordinate system in Manual mode.
		5	5	Enquiry whether axis is transformed to another axis in the untilted coordinate system through a swivel motion. The number of the first of two successive Q parameters must be transferred. The parameter contains the axis to be enquired (0= X, 1 = Y, 2 = Z). The second Q parameter supplies the respective image (0 = X, 1 = Y, 2 = Z, –1 = axis is not imaged).
		8	–	Spindle orientation including the angle
Changing machine parameters in the run-time memory				
	1000	MP number	MP index	Value of the machine parameter. The change remains in effect until the control is rebooted or the MP file or MP subfile is selected again. The unit must be the same as the unit of the MP in the MP file. For linear entries the values are always written in [mm].
Activate machine parameter subfile				
	1020	1	<Value depends on type of reference in OEM.SYS>	The machine-parameter subfiles defined in OEM.SYS via **MPFRAGMENT<Wert> =** (0 to 9) can be activated, or if **MPFRAGMENTFILE = <path/file name>** given in OEM.SYS is used, you can enter the line with the corresponding machine-parameter subfile (0 to n).

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
PLC data				
PLC data Block transfer of up to 8 variables possible Example, "Writing Q parameters to PLC markers": **`FN 17: SYSWRITE ID2000 NR10 IDX880 = BLOCK Q1620 - Q1627`**				
	2000	10	Marker no.	PLC marker
		60	Byte no.	PLC byte
		70	Word no.	PLC word
		80	Double-word no.	PLC double word
The function **`FN 17: SYSWRITE ID 2001`** is identical to the function FN 17: SYSWRITE ID 2000, but it makes it possible to exchange data between the NC program and the PLC without synchronization. However, there are many constraints on using the function FN 17: SYSWRITE ID 2001, which are difficult to estimate. This new FN function (FN 17: SYSWRITE ID 2001) may only be used after consultation with HEIDENHAIN.				
	2001	10	Marker no.	PLC marker
		60	Byte no.	PLC byte
		70	Word no.	PLC word
		80	Double-word no.	PLC double word
		90	Output marker no.	Profibus operand output marker
		100	Output byte no.	Profibus operand output byte
		110	Output word no.	Profibus operand output word
		120	Output double word no.	Profibus operand output double word
Set PLC markers for tapping				
	2020	1	–	Set PLC markers 4030 and 4031 for tapping.

9.5.4 Data transfer NC > NC program (FN18: SYSREAD)

The FN18: SYSREAD function is particularly useful for OEM cycles if you wish to have access from the part program to certain NC data, such as active tool compensation. A group number, a system data number and an index specify the particular item of system data that you read:
FN18: SYSREAD Qxxx = IDxxxx NRxxxx IDXxxxx (xxxx: Q parameter or numerical value); comment

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Program information				
	10	1	–	mm = 0, inch = 1
		2	–	Overlap factor for pocket milling
		3	–	Number of the active fixed cycle
		4	–	Number of the last DEF-active OEM cycle
Machine status				
	20	1	–	Active tool number
		2	–	Prepared tool number
		3	–	Active tool axis 0 = X 6 = U 1 = Y 7 = V 2 = Z 8 = W
		4	–	Programmed spindle speed
		5	–	Active spindle status –1 = spindle status undefined 0 = M3 active 1 = M4 active 2 = M5 active after M3 3 = M5 active after M4
		8	–	Active coolant status 0 = off, 1 = on
		9	–	Active feed rate
		11	–	Index of active tool
		13	–	Number of the active spindle (0 or 1)
		15	Number of logical axis	Assignment of the logical axes and geometrical axes (0 = X, 1 = Y, 2 = Z, 3 = A, 4 = B, 5 = C, 6 = U, 7 = V, 8 = W)
		17	–	Current range of traverse

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Cycle parameters				
	30	1	–	Setup clearance
		2	–	Hole depth / milling depth
		3	–	Infeed depth
		4	–	Feed rate for plunging
		5	–	First side length of pocket
		6	–	Second side length of pocket
		7	–	First side length of slot
		8	–	Second side length of slot
		9	–	Radius of circular pocket
		10	–	Feed rate for milling
		11	–	Rotational direction of the milling path
		12	–	Dwell time
		13	–	Thread pitch
		14	–	Finishing allowance
		15	–	Roughing angle
		52	<no.>	Interrogation whether Q-parameter number **<no.>** contains a string or numerical value: 0 = Tool number 1 = Tool name
Data from the tool table				
	50	1	Tool no.	Tool length L
		2	Tool no.	Tool radius R
		3	Tool no.	Tool radius R2
		4	Tool no.	Oversize in tool length DL
		5	Tool no.	Oversize in tool radius DR
		6	Tool no.	Oversize for tool radius DR2
		7	Tool no.	Tool locked TL 0 = not locked, 1 = locked
		8	Tool no.	Number of replacement tool RT
		9	Tool no.	Maximum tool age TIME1
		10	Tool no.	Maximum tool age TIME2
		11	Tool no.	Current tool age CUR. TIME
		12	Tool no.	PLC status
		13	Tool no.	Maximum tooth length LCUTS
		14	Tool no.	Maximum plunge angle ANGLE
		15	Tool no.	TT: Number of teeth CUT

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
		16	Tool no.	TT: Wear tolerance in length LTOL
		17	Tool no.	TT: Wear tolerance in radius RTOL
		18	Tool no.	TT: Direction of rotation DIRECT 0 = positive, –1 = negative
		19	Tool no.	TT: Offset in plane R-OFFS R = 99999.9999
		20	Tool no.	TT: Offset in length L-OFFS
		21	Tool no.	TT: Break tolerance in length LBREAK
		22	Tool no.	TT: Breakage tolerance in radius RBREAK
		23	Tool no.	PLC value
		24	Tool no.	Center misalignment in reference axis CAL-OF1
		25	Tool no.	Probe center offset in minor axis CAL-OF2
		26	Tool no.	Spindle angle for calibration CAL-ANG
		27	Tool no.	Tool type for pocket table
		28	Tool no.	Maximum speed NMAX
		29	Tool no.	PLC value P1
		30	Tool no.	PLC value P2
		31	Tool no.	PLC value P3
		32	Tool no.	Point angle for DRILL and CSINK T-ANGLE
		33	Tool no.	Thread pitch for TAP PITCH
Data from the pocket table				
	51	1	Pocket number	Tool number
		2	Pocket number	0 = No special tool 1 = Special tool
		3	Pocket number	0 = No fixed pocket 1 = Fixed pocket
		4	Pocket number	0 = Pocket not locked 1 = Locked pocket
		5	Pocket number	PLC status
		6	Pocket number	Tool type
		7 to 11	Pocket number	P1 to P5
		12	Pocket number	0 = not a reserved pocket 1 = reserved pocket
		13	Pocket number	Pocket above is locked
		14	Pocket number	Pocket below is locked
		15	Pocket number	Pocket to the left is locked
		16	Pocket number	Pocket to the right is locked

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Tool pocket				
	52	1	Tool number	Pocket number P
		2	Tool number	Tool magazine number
File information				
	56	1	–	Number of lines of the selected tool table
		2		Number of lines of the selected datum table
		3	No. of the 1st of 9 consecutive Q parameters for axes X, Y, Z, A, B, C, U, V, W	Number of axes programmed in the selected datum table (the function is identical to **FN18: SYSREAD ID990 NR3**)
		4	–	Number of rows in a freely definable table that has been opened with **TABOPEN=**. The function returns the value –1.0 if no table is open at the time of reading.
Values programmed in TOOL CALL				
	60	1	–	Tool number T (the TNC 426/430 reads a tool index as a decimal character)
		2	–	Active tool axis 0 = X 6 = U 1 = Y 7 = V 2 = Z 8 = W
		3	–	Spindle speed S
		4	–	Oversize in tool length DL
		5	–	Oversize in tool radius DR
		6	–	Automatic TOOL CALL 0 = yes, 1 = no
		7	–	Oversize for tool radius DR2
		8	–	Tool index

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Tool-change sequence				
	61	0	Tool no.	–1: Sequence cannot be ascertained 0: Tool already in the spindle 1: Manual tool → manual tool 2: Normal tool → manual tool 3: Special tool → manual tool 4: T0 → manual tool 5: Manual tool → normal tool 6: Normal tool → normal tool 7: Special tool → normal tool 8: T0 → normal tool 9: Manual tool → special tool 10: Normal tool → special tool 11: Special tool → special tool 12: T0 → special tool 13: Manual tool → T0 14: Normal tool → T0 15: Special tool → T0
Position programmed after TOOL CALL				
	70	1	–	1 = Valid position
		2	1	Position in axis X
			2	Position in axis Y
			3	Position in axis Z
		3	–	Feed rate (–1 = no feed rate programmed)
Tool compensation				
	200	1	–	Active radius (including oversizes) with algebraic sign
		2	–	Active length (including oversizes)

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Coordinate transformations				
	210	1	–	Basic rotation (manual)
		2	–	Programmed rotation
		3	–	Active mirror axis Bits 0 to 2 and 6 to 8: Axes X, Y, Z and U, V, W
		4	1	Active scaling factor in X
			2	Active scaling factor in Y
			3	Active scaling factor in Z
			7	Active scaling factor in U
			8	Active scaling factor in V
			9	Active scaling factor in W
		5	1	3-D ROT A
			2	3-D ROT B
			3	3-D ROT C
		6	–	Tilt working plane in Program Run mode (0 = inactive, –1 = active)
		7	–	Tilt working plane in Manual mode (0 = inactive, –1 = active)
		8	–	Angle of misalignment between spindle and tilted coordinate system
	214	8	–	Tolerance programmed with cycle 32 or MP1096
	220	2	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Current datum shift
		3	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Difference between reference point and datum point
		4	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Current PLC datum shift
Traverse range				
	230	2	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Negative software limit switches
		3	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Positive software limit switches

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Nominal position in the REF system				
	240	1	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Current NOMINAL position of the axis in the REF system
		8	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Current ACTUAL position of the axis in the REF system
Current position in the active coordinate system				
	270	1	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Note: Does not apply to axes that are only displayed. Solution only possible via PLC or **`FN 20: WAIT FOR SYNC`** **`FN 18: SYSREAD Q<xxx> = ID2000 ...`**
M128 active				
	280	1	–	–1 = M128 (TCPM) active, 0 = M128 (TCPM) not active
		2	–	Feed rate programmed with M128
Tilting axes				
	290	1	–	Rows of the active kinematics table
		2	Number of the bit	Values of the individual bits of the active MP7500 (kinematics table or machine parameters)
M144 active				
	310	144	–	–1 = M144 active 0 = M144 not active
		116	–	<> 0 = M116 active 0 = M116 not active
		128	–	<> 0 = M128 active 0 = M128 inactive
System time on the control				
	320	1	0	Current system time on the control
TS touch-trigger probe				
	350	10	–	Tool axis
		11	–	Effective radius
		12	–	Effective length
		13	–	Calibration ring radius
		14	1	Center offset (reference axis)
			2	Center offset (minor axis)
		15	–	Direction of center offset with respect to spindle 0°

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
TT touch probe for tool measurement				
	350	20	1	Center in axis 1
			2	Center in axis 2
			3	Center in axis 3
		21	–	Effective radius
		22	1	Probing position 1 in axis X
			2	Probing position 1 in axis Y
			3	Probing position 1 in axis Z
		23	1	Probing position 2 in axis X
			2	Probing position 2 in axis Y
			3	Probing position 2 in axis Z
		24	1	Probing position 3 in axis X
			2	Probing position 3 in axis Y
			3	Probing position 3 in axis Z
		25	1	Probing position 4 in axis X
			2	Probing position 4 in axis Y
			3	Probing position 4 in axis Z
Datum from touch probe cycle				
	360	1	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Last datum of a manual touch probe cycle, or last touch point from Cycle 0 without compensation of stylus length, but with compensation of stylus radius (workpiece coordinate system)
		2	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Last reference point of a manual touch probe cycle, or last touch point from Cycle 0 without stylus length or stylus radius compensation (machine coordinate system)
		3	–	Measurement result of touch probe cycles 0 and 1 without probe radius or length compensation
	370	6	–	Measurement result on the screen
Read values from active datum table				
	500	Line	Column	Read ACTUAL values (MP7475 = 0)
	501	Line	Column	Read REF values (MP7475 = 1)
	505	1	–	0 = No datum table selected 1 = Datum table selected

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Read values from the active preset table				
	502	Line	Column	Read value with conversion in the currently active coordinate system from the preset table
	503	Line	Column	Read value from the preset table without conversion to the currently active coordinate system
	504	Line	–	Read basic rotation from the preset table
Values from the active pallet table				
	510	1	–	Active row (-1 = No pallet table active)
		2	–	Pallet number from column **Name**
		3	–	Current row of the pallet table
		4	–	Last line of the NC program of the current pallet
		5	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Tool-oriented machining 0 = Safety height not programmed 1 = Safety height programmed
		6	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Programmed clearance height in pallet table for tool-oriented machining
Preset table				
	530	1	–	Active row
		2	Row in the preset table	0 = Row is not write-protected ≠ 0 = Row is write-protected
Pallet preset table				
	540	1	–	Active row

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Read machine statuses				
	590	1	1 to 9	"Variables" 1 to 9 can only be written with **FN17: SYSWRITE ID590 NR1.** Deletion depends on MP7300.
	590	2	1 to 9 (X, Y, Z, A, B, C, U, V, W)	The "variables" for ID621. Independent of MP7300.
	590	3	1 to 9 (X, Y, Z, A, B, C, U, V, W)	The "variables" for ID621. Independent of a power failure.
	621	0	1 to 9 (X, Y, Z, A, B, C, U, V, W)	Start measurement of the dynamic load.
	630	0	1 to 175	You can explicitly determine whether the SIK option given in **IDX** is set or not 1 = Option is enabled 0 = Option is not enabled
	630	1	–	It can be determined whether a Feature Content Level (for upgrade functions) is set, and which one. –1 = No FCL is set <No.> = FCL that is set
	630	2	–	Serial number of the SIK -1 = No valid SIK in the system

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
Touch probe cycles				
	990	1	–	Approach behavior 0 = Standard behavior 1 = Effective radius, set-up clearance is zero
		2	10	0.0 = Execution not in block scan 1.0 = Execution in block scan –1.0 = Invalid index
			16	0.0 = Execution not in Automatic operating mode 1.0 = Execution in Automatic operating mode –1.0 = Invalid index
		4	–	0.0 = Spindle or 1.0 = Touch probe deflected
		10	<no.>	Tool number that belongs to the tool name of Q parameter **IDX <no.>**
Coordinate transformation				
		3	No. of the 1st of 9 consecutive Q parameters for axes X, Y, Z, A, B, C, U, V, W	Number of axes that are programmed in the selected datum table
		8	–	Current spindle angle
Machine parameters				
	1000	MP number	MP index	Value of the machine parameter (even with MP1054.x and MP7530.x, if they don't have a formula). As of software 340 49x-05, the entered value is returned even if MPs are not active.
		410	3 or 4	ASCII value of the axis designation from MP410.3 or MP410.4
	1010	MP number	MP index	0 = MP does not exist 1 = MP exists

Group name	Group number ID....	System data number NR....	System data index IDX....	System datum
PLC data (block transfer of up to 8 variables possible) Example "Reading PLC markers to Q parameters": **`FN 18: SYSREAD BLOCK Q1620 - Q1627 = ID2000 NR10 IDX880`**				
	2000	10	Marker no.	PLC marker
		20	Input no.	PLC input
		30	Output no.	PLC output
		40	Counter no.	PLC counter
		50	Timer no.	PLC timer
		60	Byte no.	PLC byte
		70	Word no.	PLC word
		80	Double-word no.	PLC double word
		90	Input byte no.	Profibus operand input byte
		100	Input word no.	Profibus operand input word
		110	Input double word no.	Profibus operand input double word
		120	Output byte no.	Profibus operand output byte
		130	Output word no.	Profibus operand output word
		140	Output double word no.	Profibus operand output double word

9.5.5 Data transfer of machine parameters > PLC

Up to 192 machine parameters are reserved for data transfer to the PLC. The iTNC saves the contents of MP4210.x, MP4220.x and MP4310.x in PLC words. You must call the contents of MP4230.x and MP4231.x by using Module 9032. In these machine parameters you can save, for example, values for PLC positioning or datum shifts, feed rates for PLC positioning or coding for the release of certain PLC functions. The number of indexes for MP4230.x can be increased to a maximum of 99 with the `NUMBERMP4230 =` entry in OEM.SYS. No entry or an invalid entry defines 32 indexes for MP4230.x. You must evaluate the transferred numerical values in your PLC program. The iTNC internally rounds input values less than 0.001 mm (or °) to 0.001 mm (or °).

		Set	Reset
D768	**Value from MP4210.0**	NC	NC
D772	**Value from MP4210.1**	NC	NC
D776	**Value from MP4210.2**	NC	NC
D780	**Value from MP4210.3**	NC	NC
D784	**Value from MP4210.4**	NC	NC
D788	**Value from MP4210.5**	NC	NC
D792	**Value from MP4210.6**	NC	NC
D796	**Value from MP4210.7**	NC	NC
D800	**Value from MP4210.8**	NC	NC
D804	**Value from MP4210.9**	NC	NC
D808	**Value from MP4210.10**	NC	NC
D812	**Value from MP4210.11**	NC	NC
D816	**Value from MP4210.12**	NC	NC
D820	**Value from MP4210.13**	NC	NC
D824	**Value from MP4210.14**	NC	NC
D828	**Value from MP4210.15**	NC	NC
D832	**Value from MP4210.16**	NC	NC
D836	**Value from MP4210.17**	NC	NC
D840	**Value from MP4210.18**	NC	NC
D844	**Value from MP4210.19**	NC	NC
D848	**Value from MP4210.20**	NC	NC
D852	**Value from MP4210.21**	NC	NC
D856	**Value from MP4210.22**	NC	NC
D860	**Value from MP4210.23**	NC	NC
D864	**Value from MP4210.24**	NC	NC
D868	**Value from MP4210.25**	NC	NC
D872	**Value from MP4210.26**	NC	NC
D876	**Value from MP4210.27**	NC	NC

		Set	Reset
D880	**Value from MP4210.28**	NC	NC
D884	**Value from MP4210.29**	NC	NC
D888	**Value from MP4210.30**	NC	NC
D892	**Value from MP4210.31**	NC	NC
D896	**Value from MP4210.32**	NC	NC
D900	**Value from MP4210.33**	NC	NC
D904	**Value from MP4210.34**	NC	NC
D908	**Value from MP4210.35**	NC	NC
D912	**Value from MP4210.36**	NC	NC
D916	**Value from MP4210.37**	NC	NC
D920	**Value from MP4210.38**	NC	NC
D924	**Value from MP4210.39**	NC	NC
D928	**Value from MP4210.40**	NC	NC
D932	**Value from MP4210.41**	NC	NC
D936	**Value from MP4210.42**	NC	NC
D940	**Value from MP4210.43**	NC	NC
D944	**Value from MP4210.44**	NC	NC
D948	**Value from MP4210.45**	NC	NC
D952	**Value from MP4210.46**	NC	NC
D956	**Value from MP4210.47**	NC	NC
W960	**Value from MP4220.0**	NC	NC
W962	**Value from MP4220.1**	NC	NC
W964	**Value from MP4220.2**	NC	NC
W966	**Value from MP4220.3**	NC	NC
W968	**Value from MP4220.4**	NC	NC
W976	**Value from MP4310.0**	NC	NC
W978	**Value from MP4310.1**	NC	NC
W980	**Value from MP4310.2**	NC	NC
W982	**Value from MP4310.3**	NC	NC
W984	**Value from MP4310.4**	NC	NC
W986	**Value from MP4310.5**	NC	NC
W988	**Value from MP4310.6**	NC	NC
W990	**Value from MP4310.7**	NC	NC
W992	**Value from MP4310.8**	NC	NC
W994	**Value from MP4310.9**	NC	NC
M4300 - M4315	**Value from MP4310.0**	NC	NC
M4316 - M4331	**Value from MP4310.1**	NC	NC
M4332 - M4347	**Value from MP4310.2**	NC	NC
M4348 - M4363	**Value from MP4310.3**	NC	NC
M4364 - M4379	**Value from MP4310.4**	NC	NC
M4380 - M4395	**Value from MP4310.5**	NC	NC
M4396 - M4411	**Value from MP4310.6**	NC	NC
M4412 - M4427	**Value from MP4310.7**	NC	NC
M4428 - M4443	**Value from MP4310.8**	NC	NC
M4444 - M4459	**Value from MP4310.9**	NC	NC

MP4210.0-47 Setting a number in the PLC (D768 to D956)
Input: –99 999.9999 to +99 999.9999

MP4220.0-4 Setting a number in the PLC (W960 to W968)
Input: 10 to 30 000

MP4230.0-31 Setting a number in the PLC (Module 9032)
The number of indexes can be increased via an entry in OEM.SYS.
Input: –99 999.9999 to +99 999.9999

MP4231.0-31 Setting a number in the PLC (Module 9032)
Input: –99 999.9999 to +99 999.9999

MP4310.0-9 Setting a number in the PLC (W976 to W994, M4300 to M4459)
Format: Number, $xxxx [Hex], %xxxxxxxxxxxxxxxx [Bin]
Input: 0 to 65 535

Module 9032 Read machine parameters

With this module you can read the value of the given machine parameter from the active machine parameter file. The input value is transferred as a natural number with the decimal point shifted by the number of possible decimal places.

Only the value from the editable machine parameter file is read, not any value modified in the run-time memory by PLC Module 9031.

For non-indexed machine parameters, zero must be transferred as the index.

Call only in a submit job.

Call:

```
PS    B/W/D/K          <MP number>
PS    B/W/D/K          <MP index>
CM    9032
PL    B/W/D       <MP value / Error code>
                  1: MP number does not exist
                  2: No separator (:)
                  3: MP value out of range
                  4: MP not found in file
                  5: No MP file found
                  6: Call was not in a submit job.
                  7: MP is of the "string" type
                  8: No system memory
```

9.5.6 Interrogate PLC operands in the NC program (FN20: WAIT FOR)

With **FN20: WAIT FOR** you can interrupt the NC program until the condition programmed in the FN20 block is fulfilled. These conditions can be comparisons of a PLC operand with a constant. Permitted PLC operands: M, B, W, D, T, C, I, O and their symbolic operands

Operator	Function
==	Equal to
!= or <>	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

If you enter no condition, the interruption will continue until the operand = 0.

Examples:

FN20: WAIT FOR I10==1
Continue the NC program if PLC input I10 is set.

FN20: WAIT FOR I10
Continue the NC program if PLC input I10 equals zero.

FN20: WAIT FOR B3000>250
Continue the NC program, if the content of B3000 is greater than 250.

FN20: WAIT FOR is processed in look-ahead, i.e. if a synchronization with real time is necessary, then **FN20: WAIT FOR SYNC** must be programmed in the preceding NC block. Look-ahead is then stopped, and **FN20: WAIT FOR** is not performed until this block is actually reached in the NC program.

Examples with symbolic operands:

FN 20: WAIT FOR KEYCODE == 3 instead of
FN 20: WAIT FOR W274 == 3

Relational operands can also be symbolic. Example:
FN 20: WAIT FOR MARKER_A == MARKER_B

9.6 Operands

9.6.1 Overview of operands

Operand	Abbreviation	Address range
Marker	M (marker)	**M0 to M9999** **M0 to M999** are free; they are deleted only after entering the code number 531210, not during a reset (nonvolatile range); the range can be reduced in the *.CFG file of the PLC compiler. **M1000 to M3999** free, are deleted upon reset **M4000 to M5999** reserved for NC/PLC interface (M4800 to M4999 are deleted before the first run of the PLC program, e.g. after compilation or restarting). **M6000 to M9999** are free; they are deleted during reset.
Input	I (input)	**I0 to I31** (MC 42x(B,C)) **I128 to I152** (machine operating panel) **I64 to I127** (first PL) **I192 to I255** (second PL) **I256 to I319** (third PL) **I320 to I383** (fourth PL)
Output	O (output)	**O0 to O30** (MC 42x(B,C)) **O0 to O7** (via machine operating panel) **O32 to O62** (first PL) **O64 to O94** (second PL) **O128 to O158** (third PL) **O160 to O190** (fourth PL)
Counter	C (counter)	Set counter: **C0 to C47** Counter contents: **C48 to C95** Counter pulse release: **C96 to C143**
Timer	T (timer)	Timer start: **T0 to T47** Timer is running: **T48 to T95 and T96 to T999**
Byte Word Double word	B (byte) W (word) D (double word)	**B0 to B9999** (8 bits) **B0 to B255** are free; depending on the definition in the *.CFG file of the PLC compiler, the defined range is deleted only after entering the code number 531210, not during reset (nonvolatile range). If no range is defined in the *.CFG file, B0 to B127 is the nonvolatile range. **B256 to B2047** are reserved for NC/PLC interface. **B2048 to B9999** are free. They are deleted by a reset.
Constant	K	**–2 147 483 647 to +2 147 483 647**
String	S	**S0 to S99**

9.6.2 Operand addressing (byte, word and double word)

The memory for operands B (8 bits), W (16 bits), and D (32 bits) is only 8 bits wide. Since the operands can be 8, 16 or 32 bits wide, an overlap of the memory areas will occur, which you must take into account when addressing the memory.

Double word	Word	Byte	Memory	Word address	Double-word address
D0	W2	B3	8 bits	High byte	Highest byte
		B2	8 bits	Low byte	
	W0	B1	8 bits	High byte	
		B0	8 bits	Low byte	Lowest byte
D4	W6	B7	8 bits	High byte	
		B6	8 bits	Low byte	
	W4	B5	8 bits		
		B4	8 bits		
• • •	• • •	• • •	• • •	• • •	• • •
D9996	W9998	B9999	8 bits	High byte	Highest byte
		B9998	8 bits	Low byte	
	W9996	B9997	8 bits	High byte	
		B9996	8 bits	Low byte	Lowest byte

For byte addressing, every address is accessible; for word addressing, every second address; and for double word addressing, every fourth from 0 to 9996. The address parameter indicates the low byte of the word address (W) and the lowest byte of the double-word address (D).

Markers, timers and counters are addressed with the corresponding code letters M, T or C followed by the operand number (e.g. M500, T7, C18).

9.6.3 Timers

The PLC has 952 timers, which you control through special markers with the symbol T. You define the run time of the timers T0 to T47 in MP4110.x, and the run time of timers T96 to T999 in MP4111.x. MP4111.x is defined by entering the keyword **NUMBERMP4111 =** followed by the required number of timers in the OEM.SYS file. The unit of time (input value 1 in MP4110.x and MP4111.x) is seconds.

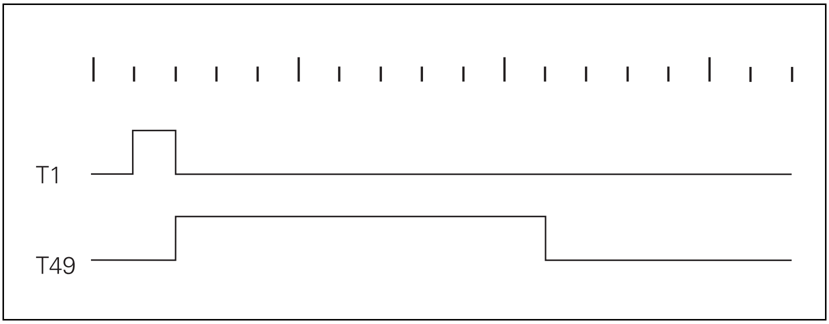
You can start the first 48 timers by setting one of the timers T0 to T47 for at most one PLC scan (otherwise the iTNC restarts the timer with the negative edge for each additional scan). The iTNC reserves the timer with the duration defined in MP4110.x, and sets the markers T48 to T95 (timer is running) until the defined duration has expired.

You can also set and start timers T0 to T47 with Module 9006. Timers T96 to T999 can only be started through Module 9006.

Cyclic timers (> T96) can be defined and started with Module 9197. They are reset for one PLC cycle, and are then restarted automatically.

Example:

Start of timer 1
Run time in MP4110.1 = 9 (PLC cycles)



Start timer	Timer is running	Machine parameter
T0	T48	MP4110.0
T1	T49	MP4110.1
T2	T50	MP4110.2
T3	T51	MP4110.3
T4	T52	MP4110.4
T5	T53	MP4110.5
T6	T54	MP4110.6
T7	T55	MP4110.7
T8	T56	MP4110.8
T9	T57	MP4110.9
T10	T58	MP4110.10
T11	T59	MP4110.11

Start timer	Timer is running	Machine parameter
T12	T60	MP4110.12
T13	T61	MP4110.13
T14	T62	MP4110.14
T15	T63	MP4110.15
T16	T64	MP4110.16
T17	T65	MP4110.17
T18	T66	MP4110.18
T19	T67	MP4110.19
T20	T68	MP4110.20
T21	T69	MP4110.21
T22	T70	MP4110.22
T23	T71	MP4110.23
T24	T72	MP4110.24
T25	T73	MP4110.25
T26	T74	MP4110.26
T27	T75	MP4110.27
T28	T76	MP4110.28
T29	T77	MP4110.29
T30	T78	MP4110.30
T31	T79	MP4110.31
T32	T80	MP4110.32
T33	T81	MP4110.33
T34	T82	MP4110.34
T35	T83	MP4110.35
T36	T84	MP4110.36
T37	T85	MP4110.37
T38	T86	MP4110.38
T39	T87	MP4110.39
T40	T88	MP4110.40
T41	T89	MP4110.41
T42	T90	MP4110.42
T43	T91	MP4110.43
T44	T92	MP4110.44
T45	T93	MP4110.45
T46	T94	MP4110.46
T47	T95	MP4110.47

MP4110.0-47 Run time PLC timer T0 to T47

Input: 0 to 1 000 000.000 [s]

MP4111.96-x Run time PLC timer T96 to x (defined in OEM.SYS)

Input: 0 to 1 000 000.000 [s]

Module 9006:Set and start PLC timer

Use Module 9006 to define the run time for a PLC timer and start the timer. Constraints:

- If during a PLC scan a timer from T0 to T47 is set in the PLC program, and the same timer is activated through Module 9006, then the direct activation through T0 to T47 has priority regardless of whether the module is called before or after setting T0 to T47.
- Immediately after the module call, one of the markers T48 to T96 is set. T0 to T47 are not set.
- The iTNC rounds the actual run time to integral PLC cycle times.
- Cancel run time: Reset timers T48 to T999.

Call:

PS	B/W/D/K	<timer number>
		Input value: 0 to 999
PS	B/W/D/K	<Run time>
		0 to 1 000 000 000 [ms]
		–1: Run time from MP4110.x or MP4111.x
CM	9006	

Error recognition:

Marker	Value	Meaning
M4203	0	Timer started
	1	Error. See W1022.
W1022	1	Invalid timer number or excessive run time
	2	Timer already assigned for cyclic timer
	3	Timer is started as cyclic timer (Module 9197)

Module 9197 Start cyclic timer

Module 9197 can define and start a timer > T96 as cyclic timer. After expiration of the defined time, the timer is reset for a PLC cycle and afterwards is automatically restarted.

- Stop timer: Transfer run time 0
- The iTNC rounds the actual run time to integral PLC cycle times.

Call:

PS	B/W/D/K	<Timer number>
		96 to 999
PS	B/W/D/K	<Run time>
		0 to 1 000 000 000 [ms]
		–1: Run time from MP4111.x
CM	9197	

Error recognition:

Marker	Value	Meaning
M4203	0	Timer started
	1	Error. See W1022.
W1022	1	Excessive run time
	3	Invalid timer number

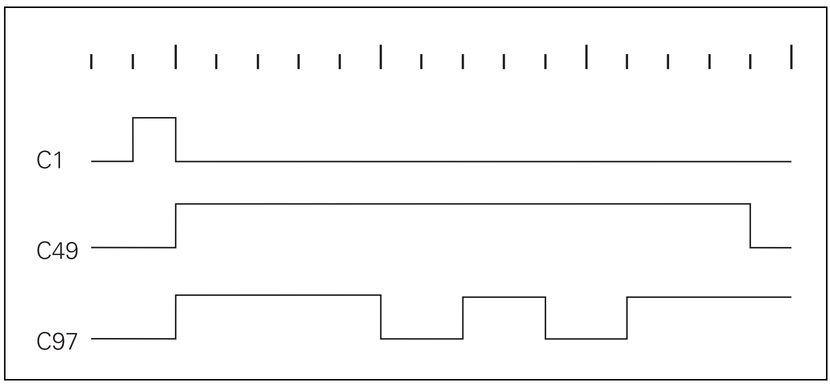
9.6.4 Counters

The PLC has 48 counters, which you control through special markers with the symbol C. After you have set a marker from the C0 to C47 range, the iTNC loads the counter with the value that is saved in machine parameter MP4120.x. The marker range C48 to C95 indicates whether the counter has expired. With markers C96 to C144 you can start and stop the counter.

MP4020 bit 11 defines whether the counter is defined in PLC cycles or seconds. In this way, the counters can also be used as timers. With this definition of counters in PLC cycles, the decimal places are not evaluated by MP4120.x.

Example:

Logic diagram for counter C1
Preset value in MP4120.1 = 10 (PLC cycles or seconds)



Set the counter	Counter is running	Start counter	Machine parameter
C0	C48	C96	MP4120.0
C1	C49	C97	MP4120.1
C2	C50	C98	MP4120.2
C3	C51	C99	MP4120.3
C4	C52	C100	MP4120.4
C5	C53	C101	MP4120.5
C6	C54	C102	MP4120.6
C7	C55	C103	MP4120.7
C8	C56	C104	MP4120.8
C9	C57	C105	MP4120.9
C10	C58	C106	MP4120.10
C11	C59	C107	MP4120.11
C12	C60	C108	MP4120.12
C13	C61	C109	MP4120.13
C14	C62	C110	MP4120.14

Set the counter	Counter is running	Start counter	Machine parameter
C15	C63	C111	MP4120.15
C16	C64	C112	MP4120.16
C17	C65	C113	MP4120.17
C18	C66	C114	MP4120.18
C19	C67	C115	MP4120.19
C20	C68	C116	MP4120.20
C21	C69	C117	MP4120.21
C22	C70	C118	MP4120.22
C23	C71	C119	MP4120.23
C24	C72	C120	MP4120.24
C25	C73	C121	MP4120.25
C26	C74	C122	MP4120.26
C27	C75	C123	MP4120.27
C28	C76	C124	MP4120.28
C29	C77	C125	MP4120.29
C30	C78	C126	MP4120.30
C31	C79	C127	MP4120.31
C32	C80	C128	MP4120.32
C33	C81	C129	MP4120.33
C34	C82	C130	MP4120.34
C35	C83	C131	MP4120.35
C36	C84	C132	MP4120.36
C37	C85	C133	MP4120.37
C38	C86	C134	MP4120.38
C39	C87	C135	MP4120.39
C40	C88	C136	MP4120.40
C41	C89	C137	MP4120.41
C42	C90	C138	MP4120.42
C43	C91	C139	MP4120.43
C44	C92	C140	MP4120.44
C45	C93	C141	MP4120.45
C46	C94	C142	MP4120.46
C47	C95	C143	MP4120.47

MP4120.0-47 Preset value for PLC counters

Input: 0 to 1 000 000.000 [s or PLC cycles, depending on MP4020, bit 11]

9.6.5 Fast PLC inputs

With MP4130 you can define PLC inputs that are not interrogated within the PLC cycle, but rather in the control loop cycle. Markers M4590 to M4593 show the current state of the fast PLC inputs.

You must activate the **fast PLC inputs** in the PLC program with W522 bit 2 to bit 5.

For the iTNC to identify with certainty a signal change, the signal duration at the fast PLC input must last a minimum of 4 ms.

MP4130	**Numbers of fast PLC inputs**
Input:	0 to 20 000 [no. of the PLC input] -1: Function inactive
MP4130.2	Fast PLC input sets marker M4590
MP4130.3	Fast PLC input sets marker M4591
MP4130.4	Fast PLC input sets marker M4592
MP4130.5	Fast PLC input sets marker M4593

MP4131.2-5	**Activation criterion for fast PLC inputs**
Input:	0: Activate at LOW level 1: Activate at HIGH level

		Set	**Reset**
W522	**Activate the high-speed PLC inputs** Bit 2: Fast PLC input defined in MP4130.2 Bit 3: Fast PLC input defined in MP4130.3 Bit 4: Fast PLC input defined in MP4130.4 Bit 5: Fast PLC input defined in MP4130.5	PLC	PLC

		Set	**Reset**
M4590	**Status fast PLC input from MP4130.2**	NC	PLC
M4591	**Status fast PLC input from MP4130.3**	NC	PLC
M4592	**Status fast PLC input from MP4130.4**	NC	PLC
M4593	**Status fast PLC input from MP4130.5**	NC	PLC

Attention

Only the PLC inputs of the MC 42x(B,C) can be defined as fast PLC inputs, and not the inputs on a PL.

9.7 Program Creation

Attention

HEIDENHAIN would like to point out the following information for the processing of signals in the PLC:

- For all PLC markers, the status 0 (not set) must be used as the safe status. Only this way can the safety of a machine be ensured if the power fails.
- The following must be kept in mind for non-volatile markers:
  Non-volatile markers, bytes, words and double words can also be deleted (e.g. if the buffer battery is empty). In order to detect this state, we recommend setting a flag marker in the non-volatile memory. Set this marker once, at a time when it is ensured that all signals in the PLC are valid.
  This marker should be used as a reference in the PLC program. As long as this marker is set, the non-volatile markers have not been deleted (e.g. by an empty buffer battery or an error in the PLC). If this flag marker is deleted, all non-volatile markers are invalid. An appropriate safety reaction with an error message must occur if this happens.

In order to attain an optimum level of safety for your machine and the operator, HEIDENHAIN recommends the combined use of these safety measures.

In connection with this, a possibility for outputting the cause of the last PLC run-time error was introduced. PLC word W1002 can be used to determine the last PLC run-time error that led to the stop of PLC program execution.

If a PLC run-time error occurs, the generated error code is saved in W1002 and simultaneously in the non-volatile memory of the control. When a new PLC run-time error occurs, W1002 and the value in the non-volatile memory are overwritten.
If the control is restarted or a PLC program start is triggered in some other manner (by Compile, Restart PLC or acknowledgment of PLC run-time errors), the value in W1002 regenerates itself via the information in the non-volatile memory. Once the value has been entered in W1002 again, this information is deleted from the non-volatile memory. This way the last PLC run-time error is always available in W1002, even after the PLC is restarted. In addition, the error code can already be seen in the PLC table as soon as the **Power interrupted** status after the control has been restarted.

The error code in W1022 can be output and evaluated in the PLC program. The following applies to the evaluation of the error code, which should be performed during the first PLC run-through:

- W1002 = 0: No PLC run-time error occurred before the last PLC program start
- W1002 = 1: Error information in W1002 is invalid (e.g. because of an error in the hardware). If necessary: Implement safety reaction.
- W1002 ≥ 50: PLC run-time error with number code (see the PLCdesignNT help for the error name). If necessary: Implement safety reaction.

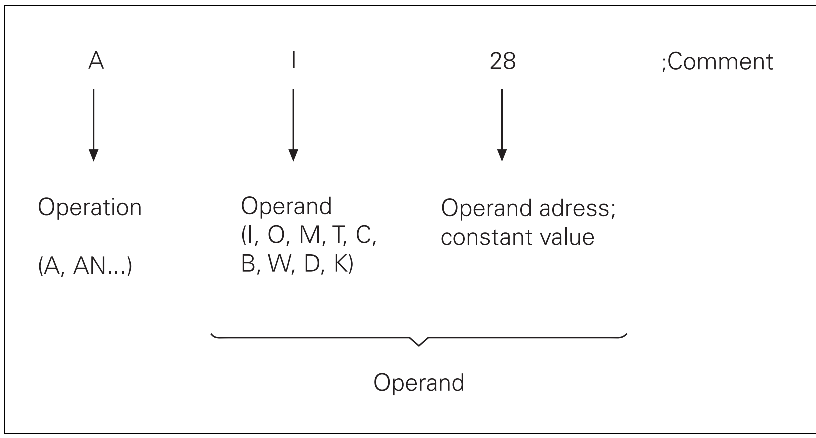
9.7.1 ASCII editor

With the integrated editor you can create the PLC program and all other necessary files right at the control through the ASCII keyboard. You will find a comprehensive description of the editor including its soft keys in the User's Manual of the control.

9.7.2 Program format

Command

A command is the smallest unit of a PLC program. It consists of the operation part and the operand part.



The operation describes the function to be executed. It defines how the operand is to be processed by the iTNC. The operand shows what is to be operated with. It consists of the operand abbreviation and a parameter (address). With the PLC commands you can combine (gate), delete and load register and memory contents, both with bit and word processing. For word processing, you can address memory contents with a length of 8 bits (byte), 16 bits (word) or 32 bits (double word).

9.7.3 Program structure

To make it easier to maintain and expand your PLC program, you should give it a modular structure. Modular means that you write a separate program module for each function. You can then call the individual modules from the main program. You should interrogate improper functioning of the machine in the PLC program and indicate such malfunctions on the screen with plain-language error messages.

Module 9019 Size of the processing stack

To debug functions you can use Module 9019 to interrogate the contents of the processing stack. The function answers with the number of the bytes that lie on the processing stack of the PLC at the moment. If the processing stack is empty, the iTNC returns the value zero. A byte, word, double word or string occupies four bytes on the stack; a marker, input, output, timer or counter occupies two bytes.

Call:

CM	9019	
PL	B/W/D	<Number of bytes on processing stack>

9.7.4 Logical names for files

Instead of a permanent file names, you can also enter logical names in PLC modules, such as **PS S">OEM.PLCMAIN"**.

Syntax: **>Group.name**

Name of the entry

Group name (NCPATH or OEM)

Identifier for logical name

Examples

>NCPATH.NCEDIT:
The iTNC transfers the complete name and path of the file that is currently selected in the editing mode.

>OEM.PLCMAIN:
The iTNC transfers the complete name and path of the PLC program that was entered in the OEM.SYS file with the command **PLCMAIN.**

List of the logical names:

Group	Entry	Meaning
NCPATH		
	PLCEDIT	Selected file in the PLC Programming mode
	NCEDIT	Selected file in the Programming and Editing mode
	RUNPGM	Selected file in the Program Run mode
	RUNDATUM	Selected datum table in the Program Run mode
	SIMPGM	Selected file in the Test Run mode
	SIMDATUM	Selected datum table in the Program Test mode
	SIMTOOL	Selected tool table in the Program Test mode
	RUNBRKPGM	Target file in the block scan in the Program Run mode
	SIMBRKPGM	Target file in the block scan in the Program Test mode
	MDIPGM	Selected file in the Positioning with Manual Data Input operating mode
	TCHPATH	Selected datum table for manual probing
OEM		
	TABCMA	Active compensation table
	MODEHELP	Active help file
	PLCMAIN	Active PLC main program
	PLCPWM	Active PLC commissioning program for digital axes
	PLCEVENTS	Active event list for spawn command
	PLCERRTAB	Active PLC error message list (PET)
	WMAT	Active tool material file
	TMAT	Active workpiece material file
	MPFILE	Active machine parameter list
	Your own entry	In the OEM.SYS file you can indicate the desired file names with path behind your own entry. For example: HUGO=TNC:\HUGO\TEST.H

9.8 Command Set

9.8.1 Overview

The following table provides an overview of all commands explained in this chapter:

Group of functions	Syntax	Function
Loading and saving commands		
	L	Load
	LN	Load NOT
	L–	Load two’s complement
	LB	Load BYTE
	LW	Load WORD
	LD	Load DOUBLE WORD
	=	Assignment
	B=	Assign BYTE
	W=	Assign WORD
	D=	Assign DOUBLE WORD
	=N	Assign NOT
	=–	Assign two’s complement
Setting commands		
	S	Set
	R	Reset
	SN	Set NOT
	RN	Reset NOT
Logical operations		
	A	And
	AN	And NOT
	O	or
	ON	Or NOT
	XO	Exclusive OR
	XON	Exclusive OR NOT
Arithmetical commands		
	+	Addition
	–	Subtraction
	x	Multiplication
	/	Division
	MOD	Remainder

Group of functions	Syntax	Function
Increment		
	INC	Increment operand
	INCW	Increment word accumulator
	INCX	Increment index register
Decrement		
	DEC	Decrement operand
	DECW	Decrement word accumulator
	DECX	Decrement index register
Comparisons		
	==	Equal to
	<	Less than
	>	Greater than
	<=	Less than or equal to
	>=	Greater than or equal to
	<>	Not equal to
Parenthetical expression in logical operations		
	A[ ]	And [ ]
	AN[ ]	And NOT [ ]
	O[ ]	Or [ ]
	ON[ ]	Or NOT [ ]
	XO[ ]	Exclusive OR [ ]
	XON[ ]	Exclusive OR NOT [ ]
Parenthetical expressions with arithmetical instructions		
	+[ ]	Addition [ ]
	–[ ]	Subtraction [ ]
	x[ ]	Multiplication [ ]
	/[ ]	Division [ ]
	MOD[ ]	Remainder [ ]
Parenthetical expressions in comparisons		
	==[ ]	Equal to [ ]
	<[ ]	Less than [ ]
	>[ ]	Greater than [ ]
	<=[ ]	Less than or equal to [ ]
	>=[ ]	Greater than or equal to [ ]
	<>[ ]	Not equal to [ ]

Group of functions	Syntax	Function
Shifting commands		
	<<	Shift left
	>>	Shift right
Bit commands		
	BS	Bit set
	BC	Bit clear
	BT	Bit test
Stack operations		
	PS	Push data onto the data stack
	PL	Pull data from the data stack
	PSL	Push logic accumulator onto the data stack
	PSW	Push word accumulator onto the data stack
	PLL	Pull logic accumulator from the data stack
	PLW	Pull word accumulator from the data stack
Jump commands		
	JP	Unconditional jump
	JPT	Jump if logic accumulator = 1
	JPF	Jump if logic accumulator = 0
	CM	Call module
	CMT	Call module if logic accumulator = 1
	CMF	Call module if logic accumulator = 0
	EM	End of module, program end
	EMT	End of module if logic accumulator = 1
	EMF	End of module if logic accumulator = 0
	LBL	Label

9.8.2 LOAD (L)

Logic processing with the LOAD command

Syntax: L (LOAD)

Operands: M, I, O, T, C

Action:

Load the value of the addressed operand into the logic accumulator. Always use the L command at the beginning of a logic chain in order to be able to gate the operand in the following program sequence.

Example:

Gate the inputs I4 and I5 with AND, and assign the result to output O2.
Initial state:
Input I4 = 1
Input I5 = 0
Output O2 = ?

Function	STL	Logic accumulator	Operand content
Load the operand content into the logic accumulator.	L I4	Logic accumulator = 1	
Gate the content of the logic accumulator and input I5 with AND.	A I5		0
Assign the gating result to output O2.	= O2		0

Word processing with the LOAD command

Syntax: L (LOAD)

Operands: B, W, D, K

Action:

Load the value of the addressed operand, or of a constant, into the word accumulator. If necessary, the accumulator is supplemented with the correct algebraic sign. In contrast to logical operations, you must always begin a sequence of word gating operations with an L command. You cannot replace the L command with a logical gating instruction.

Example:

Gate a constant and byte B5 with AND, and assign the result to byte B8.
Initial state:
Constant 54 = 36 (hex)
Byte B5 = 2A (hex)
Output B8 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K+54	36	
Gate the contents of word accumulator and byte B5 with AND.	A B5		2A
Assign the gating result to byte B8.	= B8		22

9.8.3 LOAD NOT (LN)

Logic processing with the LOAD NOT command

Syntax: LN (LOAD NOT)

Operands: M, I, O, T, C

Action:

Load the one's complement of the addressed operand into the logic accumulator. Always use the L command at the beginning of a logic chain in order to be able to gate the operand in the following program sequence.

Example:

Gate the inverted logical state of inputs I4 and I5 with AND, and assign the result to output O2.
Initial state:
Input I4 = 0
Input I5 = 1
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the inverted operand content into the logic accumulator.	LN I4	0	
Gate the content of the logic accumulator and input I5 with AND.	A I5		1
Assign the gating result to output O2.	= O2		1

Word processing with the LOAD NOT command

Syntax: LN (LOAD NOT)

Operands: B, W, D, K

Action:

Load the complement of the addressed operand, or of a constant, into the word accumulator. If necessary, the accumulator is supplemented with the correct algebraic sign. In contrast to logical operations, you must always begin a sequence of word gating operations with an L command. You cannot replace the L command with a logical gating instruction.

Example:

Gate the complement of byte B6 and byte B5 with AND, and assign the result to byte B8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = B6 (hex)
Byte B8 = ?

Function	STL	Accumulator content	Operand content
Invert byte B6, and load into the word accu.	LN B6	2A	
Gate the contents of word accumulator and byte B5 with AND.	A B5		B6
Assign the gating result to byte B8.	= B8		22

9.8.4 LOAD TWO'S COMPLEMENT (L–)

Syntax: L– (LOAD MINUS)

Operands: B, W, D, K

Action:

Load the two's complement of the addressed operand, or of a constant, into the word accumulator. If necessary, the iTNC fills the accumulator with the correct algebraic sign. The two's complement allows negative numbers to be stored, i.e., a number loaded with the L command appears in the accumulator with an inverted sign. This command can be used only with word processing.

Example:

Negate the content of byte B5 and then add it to the content of byte B6.
Assign the result to byte B8.
Initial state:
Byte B5 = 15 (dec)
Byte B6 = 20 (dec)
Byte B8 = ?

Function	STL	Accumulator content	Operand content
Load byte B5 into the word accumulator, invert the algebraic sign.	L– B5	–15	+15
Add the contents of the word accumulator and byte B6.	+B6	+5	+20
Assign the gating result to byte B8.	= B8	+5	+5

9.8.5 LOAD BYTE (LB)

Syntax: LB (LOAD BYTE)

Operands: M, I, O, T, C

Action:

Copy 8 markers, inputs, outputs, timers or counters with ascending numbering into the word accumulator. Each operand occupies one bit in the accumulator. The iTNC saves the entered operand address in the accumulator as LSB, the entered address +1 as LSB +1 etc. The last (8th) Operand is now the MSB! If necessary, the iTNC fills the accumulator with the correct algebraic sign.

Example:

A pure-binary coded value is read through inputs I3 to I10 and saved in byte B8 in order to process it later.
Initial state:

Input	I3	= 1	Input	I7	= 0
Input	I4	= 1	Input	I8	= 1
Input	I5	= 1	Input	I9	= 1
Input	I6	= 0	Input	I10	= 0

Function	STL	Accumulator content	Operand content							
		7 6 5 4 3 2 1 0	I10	I9	I8	I7	I6	I5	I4	I3
Load inputs I3 to I10 into the accumulator (bit 0 to bit 7).	LB I3	0 1 1 0 0 1 1 1	0	1	1	0	0	1	1	1
			7	6	5	4	3	2	1	0
Assign accumulator contents to byte 8.	= B8	0 1 1 0 0 1 1 1	0	1	1	0	0	1	1	1

9.8.6 LOAD WORD (LW)

Syntax: LW (LOAD WORD)

Operands: M, I, O, T, C

Action:

Copy 16 markers, inputs, outputs, timer or counters with ascending numbering into the word accumulator. Each operand occupies one bit in the accumulator. The iTNC saves the entered operand address in the accumulator as LSB, the entered address +1 as LSB +1 etc. The last (16th) Operand is now the MSB! If necessary, the iTNC fills the accumulator with the correct algebraic sign.

Example:

See example command LB. Use command LW in the same way as LB. However, the iTNC processes 16 operands.

9.8.7 LOAD DOUBLE WORD (LD)

Syntax: LD (LOAD DOUBLE WORD)

Operands: M, I, O, T, C

Action:

Copy 32 markers, inputs, outputs, timers or counters with ascending numbering into the word accumulator. Each operand occupies one bit in the accumulator. The iTNC saves the entered operand address in the accumulator as LSB, the entered address +1 as LSB +1 etc. The last (32nd) Operand is now the MSB! If necessary, the iTNC fills the accumulator with the correct algebraic sign.

Example:

See example command LB. Use command LD in the same way as LB. However, the iTNC processes 32 operands.

9.8.8 ASSIGN (=)

Logic processing with the ASSIGN command

Syntax: = (STORE)

Operands: M, I, O, T, C

Action:

Assign the content of the logic accumulator to the addressed operand. Use the = command only at the end of a sequence of logical gating operations in order to transfer a gating result to a logic operand. This command can be used several times in succession (see example).

Example:

Gate the inputs I4 and I5 with AND, and assign the result to outputs O2 and O5.
Initial state:
Input I4 = 1
Input I5 = 0
Output O2 = ?
Output O5 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of the logic accumulator and input I5 with AND.	A I5	0	0
Assign the gating result to output O2.	= O2	0	0
Assign the gating result to output O5.	= O5	0	0

Word processing with the ASSIGN command

Syntax: = (STORE)

Operands: B, W, D

Action:

Assign the content of the word accumulator to the addressed operand. Unlike bit processing, in word processing you can also use the = command within a sequence of word-gating operations. This command can be used several times in succession.

Example:

Gate a constant and byte B5 with AND, and assign the result to byte B8 and byte B10.
Initial state:

Constant	54	= 36 (hex)
Byte	B5	= 2A (hex)
Byte	B8	= ?
Byte	B10	= ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K+54	36	
Assign the contents of the word accumulator to byte B8.	= B8	36	36
Gate the contents of word accumulator and byte B5 with AND.	A B5	22	2A
Assign the gating result to byte B8.	= B8	22	22
Assign the gating result to byte B10.	= B10	22	22

9.8.9 ASSIGN BYTE (B=)

Syntax: B= (STORE BYTE)

Operands: M, I, O, T, C

Action:

Assign 8 bits from the word accumulator to markers, inputs, outputs, timers or counters with ascending numbering. Every bit occupies an operand. The iTNC assigns the LSB in the accumulator to the operand address specified in the command, the specified address +1 as LSB +1 etc. The last (8th) operand is assigned the MSB.

Example:

See example of command W=. Use command B= in the same way as W=. However, the iTNC processes 8 operands.

9.8.10 ASSIGN WORD (W=)

Syntax: W= (STORE WORD)

Operands: M, I, O, T, C

Action:

Assign 16 bits from the word accumulator to markers, inputs, outputs, timers or counters with ascending numbering. Every bit occupies an operand. The iTNC assigns the LSB in the accumulator to the operand address specified in the command, the specified address +1 as LSB +1 etc. The last (16th) operand is assigned the MSB.

Example:

Transfer a certain bit pattern, located in word W8, to the output addresses O1 to O16.
Initial state:
Word W8 = 36FF (hex)

Function	STL	Accumulator content	Operand content
Load content of word W8 into the word accumulator.	L W8	36FF	
			O16 ... O1
Assign accumulator content to outputs O1 to O16.	W= O1	36FF	0 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1

9.8.11 ASSIGN DOUBLE WORD (D=)

Syntax: D= (STORE DOUBLE WORD)

Operands: M, I, O, T, C

Action:

Assign 32 bits from the word accumulator to markers, inputs, outputs, timers or counters with ascending numbering. Every bit occupies an operand. The iTNC assigns the LSB in the accumulator to the operand address specified in the command, the specified address +1 as LSB +1 etc. The last (32nd) operand is assigned the MSB.

Example:

See example of command W=. Use command D= in the same way as W=. However, the iTNC processes 32 operands.

9.8.12 ASSIGN NOT (=N)

Logic processing

Syntax: =N (STORE NOT)

Operands: M, I, O, T, C

Action:

Assign the complement of the logic accumulator to the addressed operand. For an example, see the ASSIGN (=) command.

Word processing

Syntax: =N (STORE NOT)

Operands: B, W, D

Action:

Assign the complement of the word accumulator to the addressed operand. For an example, see the ASSIGN (=) command.

9.8.13 ASSIGN TWO’S COMPLEMENT (=–)

Syntax: =– (STORE MINUS)

Operands: B, W, D

Action:

Assign the TWO’S COMPLEMENT of the word accumulator to the addressed operand. For an example, see the ASSIGN (=) command.

9.8.14 SET (S)

Syntax: S (SET)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 1, then set the addressed operand to 1, otherwise do not change it. Use the S command at the end of a sequence of logical gating operations in order to influence an operand, depending on the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 1, then set output O2 and marker M500.
Initial state:
Input I4 = 1
Input I5 = 0
Output O2 = ?
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of the logic accumulator and input I5 with OR.	O I5	1	0
Since the result of the operation is 1, set output O2.	S O2	1	1
Since the result of the operation is 1, set marker M500.	S M500	1	1

9.8.15 RESET (R)

Syntax: R (RESET)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 1, then set the addressed operand to 0, otherwise do not change it. Use the R command at the end of a sequence of logical gating operations in order to influence an operand, depending on the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 1, then reset output O2 and marker M500.
Initial state:
Input I4 = 1
Input I5 = 0
Output O2 = ?
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of the logic accumulator and input I5 with OR.	O I5	1	0
Since the result of the operation is 1, reset output O2.	R O2	1	0
Since the result of the operation is 1, reset marker M500.	R M500	1	0

9.8.16 SET NOT (SN)

Syntax: SN (SET NOT)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 0, then set the addressed operand to 1, otherwise do not change it. Use the SN command at the end of a sequence of logical gating operations in order to influence an operand independently from the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 0, then set output O2 and marker M500.
Initial state:

Input	I4	= 0
Input	I5	= 0
Output	O2	= ?
Marker	M500	= ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	0	0
Gate the content of the logic accumulator and input I5 with OR.	O I5	0	0
Since the result of the operation is 0, set output O2.	SN O2	0	1
Since the result of the operation is 0, set marker M500.	SN M500	0	1

9.8.17 RESET NOT (RN)

Syntax: RN (RESET NOT)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 0, then set the addressed operand to 0, otherwise do not change it. Use the RN command at the end of a sequence of logical gating operations in order to influence an operand independently from the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 0, then reset output O2 and marker M500.
Initial state:
Input I4 = 0
Input I5 = 0
Output O2 = ?
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	0	0
Gate the content of the logic accumulator and input I5 with OR.	O I5	0	0
Since the result of the operation is 0, reset output O2.	RN O2	0	0
Since the result of the operation is 0, reset marker M500.	RN M500	0	0

9.8.18 AND (A)

Logic processing with the AND command

Syntax: A (AND)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an L command, i.e., the logical state of the operand is loaded into the logic accumulator. This is to ensure compatibility with the TNC 355, which does not have the special L command. In PLC programs, a sequence of logical gating operations should always be started with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with AND. The iTNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with AND, and assign the result to output O2.
Initial state:
Input I4 = 1
Input I5 = 0
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of the logic accumulator and input I5 with AND.	A I5	0	1
Assign the gating result to output O2.	= O2	0	0

Word processing with the AND command

Syntax: A (AND)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with AND. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The iTNC saves the result of the operation in the word accumulator.

Example:

Gate the contents of byte B5 and byte B6 with AND, and assign the result to byte B8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = 36 (hex)
Byte B8 = ?

Function	STL	Accumulator content	Operand content
Load byte B6 into the word accumulator.	L B6	2A	2A
Gate the contents of word accumulator and byte B5 with AND.	A B5	22	36
Assign the gating result to byte B8.	= B8	22	22

9.8.19 AND NOT (AN)

Logic processing with the AND NOT command

Syntax: AN (AND NOT)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an LN command, i.e., the logical state of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with AND NOT. The iTNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with AND NOT, and assign the result to output O2.
Initial state:
Input I4 = 1
Input I5 = 1
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of logic accumulator and input I5 with AND NOT.	AN I5	0	1
Assign the gating result to output O2.	= O2	0	0

Word processing with the AND NOT command

Syntax: AN (AND NOT)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with AND NOT. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The iTNC saves the result of the operation in the word accumulator.

Example:

Gate the content of words W4 and W6 with AND NOT, and assign the result to word W8.
Initial state:
Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load W6 into the word accumulator.	L W6	3C36	3C36
Gate the contents of word accumulator and word W4 with AND NOT.	AN W4	814	36AA
Assign the gating result to word W8.	= W8	814	814

9.8.20 OR (O)

Logic processing with the OR command

Syntax: O (OR)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an L command, i.e., the logical state of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with OR. The iTNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with OR, and assign the result to output O2.
Initial state:
Input I4 = 0
Input I5 = 1
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	0	0
Gate the content of the logic accumulator and input I5 with OR.	O I5	1	1
Assign the gating result to output O2.	= O2	1	1

Word processing with the OR command

Syntax: O (OR)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with OR. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The iTNC saves the result of the operation in the word accumulator.

Example:

Gate the content of byte B5 and byte B6 with OR, and assign the result to word W8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load byte B6 into the word accumulator.	L B6	36	36
Gate the contents of the word accumulator and byte B5 with OR.	O B5	3E	2A
Assign the gating result to word W8.	= W8	3E	3E

9.8.21 OR NOT (ON)

Logic processing with the OR NOT command

Syntax: ON (OR NOT)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an LN command, i.e., the complement of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with OR NOT. The iTNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with OR NOT, and assign the result to output O2.
Initial state:
Input I4 = 0
Input I5 = 0
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	0	0
Gate the content of logic accumulator and input I5 with OR NOT.	ON I5	1	0
Assign the gating result to output O2.	= O2	1	1

Word processing with the OR NOT command

Syntax: ON (OR NOT)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with OR NOT. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The iTNC saves the result of the operation in the word accumulator.

Example:

Gate the content of words W4 and W6 with OR NOT, and assign the result to word W8.
Initial state:
Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load W6 into the word accumulator.	L W6	3C36	3C36
Gate the contents of word accumulator and word W4 with OR NOT.	ON W4	FD77	36AA
Assign the gating result to word W8.	= W8	FD77	FD77

9.8.22 EXCLUSIVE OR (XO)

Logic processing with the EXCLUSIVE OR command

Syntax: XO (EXCLUSIVE OR)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an L command, i.e., the logical state of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with EXCLUSIVE OR. The iTNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with EXCLUSIVE OR, and assign the result to output O2.

Initial state:
Input I4 = 1
Input I5 = 1
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of logic accumulator and input I5 with EXCLUSIVE OR.	XO I5	0	1
Assign the gating result to output O2.	= O2	0	0

Word processing with the EXCLUSIVE OR command

Syntax: XO (EXCLUSIVE OR)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with EXCLUSIVE OR. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The iTNC saves the result of the operation in the word accumulator.

Example:

Gate the contents of byte B5 and byte B6 with EXCLUSIVE OR, and assign the result to word W8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load byte B6 into the word accumulator.	L B6	36	36
Gate the contents of the word accumulator and byte B5 with EXCLUSIVE OR.	XO B5	1C	2A
Assign the gating result to word W8.	= W8	1C	1C

9.8.23 EXCLUSIVE OR NOT (XON)

Logic processing with the EXCLUSIVE OR NOT command

Syntax: XON (EXCLUSIVE OR NOT)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an LN command, i.e., the logical state of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with EXCLUSIVE OR NOT. The iTNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and marker M500 with EXCLUSIVE OR NOT, and assign the result to output O2.
Initial state:
Input I4 = 0
Marker M500 = 0
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L M500	0	0
Gate the content of logic accumulator and input I4 with EXCLUSIVE OR NOT.	XON I4	1	0
Assign the gating result to output O2.	= O2	1	1

Word processing with the EXCLUSIVE OR NOT command

Syntax: XON (EXCLUSIVE OR NOT)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with EXCLUSIVE OR NOT. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The iTNC saves the result of the operation in the word accumulator.

Example:

Gate the content of words W4 and W6 with EXCLUSIVE OR NOT, and assign the result to word W8.
Initial state:
Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load W6 into the word accumulator.	L W6	3C36	3C36
Gate the contents of word accumulator and word W4 with EXCLUSIVE OR NOT.	XON W4	F563	36AA
Assign the gating result to word W8.	= W8	F563	F563

9.8.24 ADDITION (+)

Syntax: + (PLUS)

Operands: B, W, D, K

Action:

The iTNC extends the operand to the width of the accumulator (32 bits) and then adds the content of the operand to the content of the word accumulator. The result of the operation is stored in the word accumulator where you can process it further.

Example:

Add the constant and the number saved in word W6, then assign the result to double word D8.
Initial state:
Constant = 10 000 (dec)
Word W6 = 200 (dec)
Double word D8 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K10000	10000	
Add the content of the word accumulator and word W6.	+ W6	10200	200
Assign the result to double word D8.	= D8	10200	10200

9.8.25 SUBTRACTION (–)

Syntax: – (MINUS)

Operands: B, W, D, K

Action:

The iTNC extends the operand to the width of the accumulator (32 bits) and then subtracts the content of the operand from the content of the word accumulator. The result of the operation is stored in the word accumulator where you can process it further.

Example:

Subtract the number saved in word W6 from the constant, and then assign the result to double word D8.
Initial state:
Constant = 10 000 (dec)
Word W6 = 200 (dec)
Double word D8 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K10000	10000	
Subtract word W6 from the content of the word accumulator.	– W6	9800	9800
Assign the result to double word D8.	= D8	9800	9800

9.8.26 MULTIPLICATION (X)

Syntax: x (MULTIPLY)

Operands: B, W, D, K

Action:

The iTNC extends the operand to the width of the accumulator (32 bits) and then multiplies the content of the operand with the content of the word accumulator. The result of the operation is stored in the word accumulator where you can process it further. If the iTNC cannot execute the multiplication correctly, it then sets marker M4200, otherwise it resets it.

Example:

Multiply the constant and the number saved in word W6, then assign the result to double word D8.
Initial state:
Constant = 100 (dec)
Word W6 = 20 (dec)
Double word D8 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K100	100	
Multiply the content of the word accumulator with word W6.	x W6	2000	20
Assign the result to double word D8.	= D8	2000	2000

		Set	Reset
M4200	**Overflow during multiplication**	NC	PLC

9.8.27 DIVISION (/)

Syntax: / (DIVIDE)

Operands: B, W, D, K

Action:

The iTNC extends the operand to the width of the accumulator (32 bits) and then divides the content of the word accumulator by the content of the operand. The result of the operation is stored in the word accumulator where you can process it further. If the iTNC cannot execute the division correctly, it then sets marker M4201, otherwise it resets it.

Example:

Divide the constant by the number saved in word W6, then assign the result to double word D8.
Initial state:

Constant		= 100 (dec)
Word	W6	= 20 (dec)
Double word	D8	= ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K100	100	
Divide the content of the word accumulator by word W6	/ W6	5	20
Assign the result to double word D8.	= D8	5	5

		Set	Reset
M4201	**Division by 0**	NC	PLC

9.8.28 REMAINDER (MOD)

Syntax: MOD (MODULO)

Operands: B, W, D, K

Action:

The iTNC extends the operand to the width of the accumulator (32 bits) and then calculates the remainder resulting from the division of the content of the word accumulator by the content of the operand. The remainder is stored in the word accumulator where you can process it further. If the iTNC cannot execute the MOD command correctly, it sets marker M4202, otherwise it resets it.

Example:

Divide the number saved in word W6 by the constant, then calculate the REMAINDER and assign the result to double word D8.
Initial state:

Word	W6	= 50 (dec)
Constant		= 15 (dec)
Double word	D8	= ?

Function	STL	Accumulator content	Operand content
Load W6 into the word accumulator.	L W6	50	50
Divide the content of the word accumulator by a constant, then save the integral REMAINDER in the word accumulator.	MOD K15	11	15
Assign the REMAINDER to double word D8.	= D8	11	11

		Set	Reset
M4202	**Incorrectly executed modulo**	NC	PLC

9.8.29 INCREMENT (INC)

INCREMENT operand

Syntax: INC (INCREMENT)

Operands: B, W, D

Action:

Increase the content of the addressed operand by one.

INCREMENT word accumulator

Syntax: INCW (INCREMENT WORD)

Operands: None

Action:

Increase the content of the word accumulator by one.

INCREMENT index register

Syntax: INCX (INCREMENT INDEX)

Operands: None

Action:

Increase the content of the index register by one.

9.8.30 DECREMENT (DEC)

DECREMENT operand

Syntax: DEC (DECREMENT)

Operands: B, W, D

Action:

Decrease the content of the addressed operand by one.

DECREMENT word accumulator

Syntax: DECW (DECREMENT WORD)

Operands: None

Action:

Decrease the content of the word accumulator by one.

DECREMENT index register

Syntax: DECX (DECREMENT INDEX)

Operands: None

Action:

Decrease the content of the index register by one.

9.8.31 EQUAL TO (==)

Syntax: == (EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator and the operand are equal, the condition is true and the iTNC sets the logic accumulator to 1. If they are not equal, the logic accumulator is set to 0. The comparison takes place over the number of bits corresponding to the operand:
where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M300 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Compare the content of the word accumulator with the operand content D8; if not equal, set the logic accumulator to 0.	== D8	0	15000
Assign the result to marker M500.	= M500	0	0

9.8.32 LESS THAN (<)

Syntax: < (LESS THAN)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator is less than the operand, the condition is true and the iTNC sets the logic accumulator to 1. If the word accumulator is greater than or equal to the operand, it sets the logic accumulator to 0. The comparison takes place over the number of bits in the operand:
where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator < operand; if not, set logic accumulator to 0.	< D8	0	15000
Assign the result to marker M500.	= M500	0	0

9.8.33 GREATER THAN (>)

Syntax: > (GREATER THAN)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator is greater than the operand, the condition is true and the iTNC sets the logic accumulator to 1. If the word accumulator is less than or equal to the operand, it sets the logic accumulator to 0. The comparison takes place over the number of bits in the operand: where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator > operand; if so, set logic accumulator to 1.	> D8	1	15000
Assign the result to marker M500.	= M500	1	1

9.8.34 LESS THAN OR EQUAL TO (<=)

Syntax: <= (LESS EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator is less than or equal to the operand, the condition is true and the iTNC sets the logic accumulator to 1. If the word accumulator is greater than the operand, it sets the logic accumulator to 0. The comparison takes place over the number of bits in the operand:
where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator <= operand; if not, set logic accumulator to 0.	<= D8	0	15000
Assign the result to marker M500.	= M500	0	0

9.8.35 GREATER THAN OR EQUAL TO (>=)

Syntax: >= (GREATER EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator is greater than or equal to the operand, the condition is true and the iTNC sets the logic accumulator to 1. If the word accumulator is less than the operand, it sets the logic accumulator to 0. The comparison takes place over the number of bits in the operand: where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator >= operand; if so, set logic accumulator to 1.	>= D8	1	15000
Assign the result to marker M500.	= M500	1	1

9.8.36 NOT EQUAL (<>)

Syntax: <> (NOT EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator and the operand are not equal, the condition is true and the iTNC sets the logic accumulator to 1. If they are equal, the logic accumulator is set to 0. The comparison takes place over the number of bits corresponding to the operand:
where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator <> operand; if so, set logic accumulator to 1.	<> D8	1	15000
Assign the result to marker M500.	= M500	1	1

9.8.37 AND [ ] (A[ ])

Syntax: A[ ] (AND [ ])

Operands: None

Action:

By using parentheses you can change the sequence of processing logic commands in a statement list. The opening-parenthesis command puts the content of the accumulator onto the program stack. If you address the logic accumulator in the last command before an opening-parenthesis statement, the iTNC puts the content of the logic accumulator onto the program stack. If you address the word accumulator, the iTNC saves the content of the word accumulator. With the closing-parenthesis command, the iTNC gates the buffered value from the program stack with the content of the logic accumulator or word accumulator, depending on which you have addressed before the opening-parenthesis statement. The iTNC assigns the result of the gating operation to the corresponding accumulator. Maximum nesting depth: 16 parentheses.

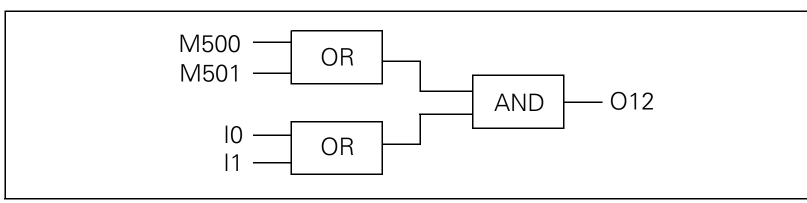
Please note:

The sequence of function is the same for word processing, however the iTNC writes the complete word accumulator onto the program stack.

Example:

Example for the commands AND [ ], AND NOT [ ], OR [ ], OR NOT [ ], EXCLUSIVE OR [ ], EXCLUSIVE OR NOT [ ]:

Use parentheses to develop a statement list in accordance with the following logic circuit diagram:



Initial state:

Marker	M500	= 0
Input	I0	= 0
Marker	M501	= 1
Input	I1	= 1
Output	O12	= ?

Function	STL	Accumulator content	Operand content
Load marker M500 into the logic accumulator.	L M500	0	0
Gate logic accumulator with marker M501.	O M501	1	1
Opening parenthesis: Buffer the accumulator content onto the program stack.	A[		
Load the state of input I0 into the logic accumulator.	L I0	0	0
Gate the logic accumulator with the state of input I1.	O I1	1	1
Closing parenthesis: Gate the accumulator content with the program stack (A[, O[...).	]		
Assign the result of the total operation to output O12.	= O12	1	1

9.8.38 AND NOT [ ] (AN[ ])

Syntax: AN[ ] (AND NOT [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

9.8.39 OR [ ] (O[ ])

Syntax: O[ ] (OR [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

9.8.40 OR NOT [ ] (ON[ ])

Syntax: ON[ ] (OR NOT [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

9.8.41 EXCLUSIVE OR [ ] (XO[ ])

Syntax: XO[ ] (EXCL: OR [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

9.8.42 EXCLUSIVE OR NOT [ ] (XON[ ])

Syntax: XON[ ] (EXCL: OR NOT [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

9.8.43 ADDITION [ ] (+[ ])

Syntax: +[ ] (PLUS[ ])

Operands: None

Action:

Use parentheses together with arithmetical commands **only** for word processing. By using parentheses you can change the sequence of processing in a statement list. The opening-parenthesis command puts the content of the word accumulator onto the program stack. This clears the accumulator for calculation of intermediate results. The closing-parenthesis command gates the buffered value from the program stack with the content of the word accumulator. The iTNC saves the result in the accumulator again. Maximum nesting depth: 16 parentheses. If an error occurs during calculation, the iTNC sets the marker M4201.

Example:

Example for the commands ADD [ ], SUBTRACT [ ], MULTIPLY [ ], DIVIDE [ ], REMAINDER [ ].

Divide a constant by double word D36, add the result to double word D12, and assign the result to double word D100.
Initial state:
Constant = 1000 (dec)
Double word D12 = 15000 (dec)
Double word D36 = 100 (dec)
Double word D100 = ?

Function	STL	Accumulator content	Operand content
Load the double word D12 into the word accumulator.	L D12	15000	15000
Opening parenthesis: Buffer the accumulator content onto the program stack.	+[		
Load the constant K1000 into the word accu.	L K1000	1000	
Divide the word accumulator by the content of the double word D36.	/ D36	10	100
Closing parenthesis: Gate the accumulator content with the program stack (+[, –[.....).	]		
Assign the result of the total operation to double word D100.	= D100	15010	15010

		Set	Reset
M4200	**Overflow during multiplication**	NC	PLC
M4201	**Division by 0**	NC	PLC
M4202	**Incorrectly executed modulo**	NC	PLC
M4203	**Error status for PLC module**	NC	NC/PLC

9.8.44 SUBTRACT [ ] (–[ ])

Syntax: –[ ] (MINUS –[ ])

Operands: None

Action:

See example of command ADD [ ]

9.8.45 MULTIPLY [ ] (x[ ])

Syntax: x[ ] (MULTIPLY [ ])

Operands: None

Action:

See example of command ADD [ ]

9.8.46 DIVIDE [ ] (/[ ])

Syntax: /[ ] (DIVIDE [ ])

Operands: None

Action:

See example of command ADD [ ]

9.8.47 REMAINDER [ ] (MOD[ ])

Syntax: MOD[ ] (MODULO [ ])

Operands: None

Action:

See example of command ADD [ ]

9.8.48 EQUAL TO [ ] (==[ ])

Syntax: ==[ ] (EQUAL[ ])

Operands: None

Action:

By using parentheses you can change the sequence of processing comparative commands in a statement list. The opening-parenthesis command puts the content of the word accumulator onto the program stack. This clears the accumulator for calculation of intermediate results.

The closing-parenthesis command gates the buffered value from the program stack with the content of the word accumulator. The iTNC saves the result in the accumulator again. Maximum nesting depth: 16 parentheses.

Comparative commands cause a direct transition from word to logical processing. If the specified comparative condition is true, the iTNC sets the logic accumulator to 1; if the condition is not fulfilled, it sets it to 0.

Example:

Multiply a constant with double word D36, compare the result with double word D12, and assign the result to output O15.
Initial state:
Constant = 1000 (dec)
Double word D12 = 15000 (dec)
Double word D36 = 10 (dec)
Output O15 = ?

Function	STL	Accumulator content	Operand content
Load the double word D12 into the word accumulator.	L D12	15000	15000
Opening parenthesis: Buffer the accumulator content onto the program stack.	== [		
Load the constant into the word accumulator.	L K1000	1000	
Multiply the content of the word accumulator with double word D36.	x D36	10000	10
Closing parenthesis: Gate the accumulator content with the program stack (==[, >=[...); if condition not fulfilled, set logic accumulator to 0.	]		
Assign the result to output O15.	= O15	0	0

9.8.49 LESS THAN [ ] (<[ ])

Syntax: <[ ] (LESS THAN [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

9.8.50 GREATER THAN [ ] (>[ ])

Syntax: >[ ] (GREATER THAN [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

9.8.51 LESS THAN OR EQUAL TO [ ] (<=[ ])

Syntax: <=[ ] (LESS EQUAL [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

9.8.52 GREATER THAN OR EQUAL TO [ ] (>=[ ])

Syntax: >=[ ] (GREATER EQUAL [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

9.8.53 NOT EQUAL [ ] (<>[ ])

Syntax: <>[ ] (NOT EQUAL [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

9.8.54 SHIFT LEFT (<<)

Syntax: << (SHIFT LEFT)

Operands: B, W, D, K

Action:

A SHIFT LEFT statement multiplies the content of the word accumulator by two. This is done by simply shifting the bits in the accumulator by one place to the left. The result must lie in the range of -2 147 483 648 to +2 147 483 647, otherwise the accumulator contains an undefined value. You define the number of shifts through the operand. The iTNC fills the right end of the accumulator with zeros.

This statement is one of the arithmetic commands because it includes the sign bit. For this reason, and to save time, you should not use this command to isolate bits.

Example:

Shift the content of double word D8 four times to the left, then assign it to double word D12.
Initial state:
Double word D8 = 3E 80 (hex)
Double word D12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E80	3E80
Shift the content of the word accumulator to the left by the number of bits that are specified in the operand.	<< K+1	7D00	
	<< K+1	FA00	
	<< K+1	1F400	
	<< K+1	3E800	
Assign the result to double word D12.	= D12	3E800	3E800

Instead of using the << K+1 command four times, simply use the << K+4 command.

9.8.55 SHIFT RIGHT (>>)

Syntax: >> (SHIFT RIGHT)

Operands: B, W, D, K

Action:

A SHIFT RIGHT statement divides the content of the word accumulator by two. This is done by simply shifting the bits by one place to the right. You define the number of shifts through the operand. The bits that the iTNC shifts to the right out of the accumulator are then lost. The iTNC extends the left side of the accumulator with the correct sign.

This statement is one of the arithmetic commands because it includes the sign bit. For this reason, and to save time, you should not use this command to isolate bits.

Example:

Shift the content of double word D8 four times to the right, then assign it to double word 12.
Initial state:
Double word D8 = 3E 80 (hex)
Double word D12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E80	3E80
Shift the content of the word accumulator to the right by the number of bits that are specified in the operand.	>> K+1	1F40	
	>> K+1	FA0	
	>> K+1	7D0	
	>> K+1	3E8	
Assign the result to double word D12.	= D12	3E8	3E8

Instead of using the >> K+1 command four times, simply use the >> K+4 command.

9.8.56 BIT SET (BS)

Syntax: BS (BIT SET)

Operands: B, W, D, K, X

Action:

With the BIT SET command you can set each bit in the accumulator to 1. The corresponding bits are selected (addressed) by the content of the specified operand or by a constant. As to the bit numbering, bit 0 = LSB and bit 31 = MSB. For operand contents greater than 32, the iTNC uses the operand value modulo 32, i.e. the integral remainder of the result of the operand value divided by 32.

Example:

Load double word D8 into the accumulator, set bit 0 of the accumulator to 1, and save the result in double word D12.
Initial state:
Double word D8 = 3E 80 (hex)
Double word D12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E80	3E80
Set the bit specified in the operand to 1.	BS K+0	3E81	
Assign the result to double word D12.	= D12	3E81	3E81

9.8.57 BIT CLEAR (BC)

Syntax: BC (BIT CLEAR)

Operands: B, W, D, K, X

Action:

With the BIT CLEAR command you can set each bit in the accumulator to 0. The corresponding bits are selected (addressed) by the content of the specified operand or by a constant. As to the bit numbering, bit 0 = LSB and bit 31 = MSB. For operand contents greater than 32, the iTNC uses the operand value modulo 32, i.e. the integral remainder of the result of the operand value divided by 32.

Example:

Load double word D8 into the accumulator, set bit 0 of the accumulator to 0, and save the result in double word D12.
Initial state:
Double word D8 = 3E 81 (hex)
Double word D12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E81	3E81
Set the bit specified in the operand to 0.	BC K+0	3E80	
Assign the result to double word D12.	= D12	3E80	3E80

9.8.58 BIT TEST (BT)

Syntax: BT (BIT TEST)

Operands: B, W, D, K, X

Action:

With the BIT TEST command, you can interrogate the status of each bit in the accumulator. With the BT command there is a direct transition from word to logic processing, i.e. the iTNC checks the state of a bit in the word accumulator and then sets the logic accumulator. If the interrogated bit = 1, the iTNC sets the logic accumulator to 1; otherwise it sets it to 0. The corresponding bits are selected (addressed) by the content of the specified operand or by a constant. As to the bit numbering, bit 0 = LSB and bit 31 = MSB. For operand contents greater than 32, the iTNC uses the operand value modulo 32, i.e. the integral remainder of the result of the operand value divided by 32.

Example:

Load the double word D8 into the accumulator and assign the logical state of bit 0 to output O12.
Initial state:
Double word D8 = 3E 81 (hex)
Output O12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E81	3E81
Check the state of the bit specified in the operand.	BT K+0	1	
Assign the result to output O12.	= O12	1	1

9.8.59 PUSH DATA ONTO THE DATA STACK (PS)

Logic processing with the PS command

Syntax: PS (PUSH)

Operands: M, I, O, T, C

Action:

The PS command enables you to buffer data. The iTNC loads the addressed operand onto the data stack. Because the data stack has a width of 32 bits, you must write to it with a minimum width of one word. The iTNC copies the operand value into bit 7 of the data stack’s current address. The vacant bits of the occupied memory remain undefined or unused. In the event of a stack overflow, the iTNC outputs an error message.

Data stack [bit]														
31 ... 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x ...	x	x	x	x	x	x	L	x	x	x	x	x	x	x

Example:

See PSW command.

Word processing with the PS command

Syntax: PS (PUSH)

Operands: B, W, D, K

Action:

The PS command enables you to buffer data. The iTNC copies the addressed operand value into the current address of the data stack. During the word processing, the iTNC copies two words per PS command onto the data stack and extends the operand—in accordance with the MSB—with the correct algebraic sign. In the event of a stack overflow, the iTNC displays an error message.

Data stack for byte, word, double word and constant [bit]			
31 24	23 16	15 8	7 0
x x x x x x x x	x x x x x x x x	x x x x x x x x	B B B B B B B B
x x x x x x x x	x x x x x x x x	W W W W W W W W	W W W W W W W W
D D D D D D D D	D D D D D D D D	D D D D D D D D	D D D D D D D D
K K K K K K K K	K K K K K K K K	K K K K K K K K	K K K K K K K K

Example:

See PSW command.

9.8.60 PULL DATA FROM THE DATA STACK (PL)

Logic processing with the PL command

Syntax: PL (PULL)

Operands: M, I, O, T, C

Action:

The PL command is the counterpart to the PS command. Data that has been buffered with the PUSH command can be taken from the data stack by using the PULL command. The iTNC copies bit 7 of the data stack's current address into the addressed operand. If the stack is empty, the iTNC displays an error message.

Example:

See PSW command.

Word processing with the PL command

Syntax: PL (PULL)

Operands: B, W, D, K

Action:

The PL command is the counterpart to the PS command. Data that has been buffered with the PUSH command can be taken from the data stack by using the PULL command. During word processing, the iTNC copies with the PL command two words of the current data stack address into the addressed memory area. If the stack is empty, the iTNC displays an error message.

Example:

See PSW command.

9.8.61 PUSH LOGIC ACCUMULATOR ONTO THE DATA STACK (PSL)

Syntax: PSL (PUSH LOGICACCU)

Operands: None

Action:

The PSL command enables you to buffer the logic accumulator. With the PSL command, the iTNC copies the logic accumulator onto the data stack. Because the data stack has a width of 32 bits, you must write to it with a minimum width of one word. The iTNC copies the operand value into bit 7 of the data stack's current address. The vacant bits of the occupied memory remain undefined or unused. In the event of a stack overflow, the iTNC outputs an error message.

Data stack [bit]														
31 ... 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x ...	x	x	x	x	x	x	L	x	x	x	x	x	x	x

Example:

See PSW command.

9.8.62 PUSH WORD ACCUMULATOR ONTO THE DATA STACK (PSW)

Syntax: PSW (PUSH WORDACCU)

Operands: None

Action:

The PSW command enables you to buffer the word accumulator. With the PSW command, the iTNC copies the word accumulator onto the data stack. The content of the word accumulator (32 bits) occupies two words on the data stack. In the event of a stack overflow, the iTNC displays an error message.

Example:

Since the sequence is the same for all stack operations, this example also applies to the commands PS, PL, PSW, PLL, PLW. The difference between the individual operations lies merely in the transferred data width.

Call Module 15 at a certain place in the program. After returning to the main program, restore the original accumulator content. Accumulator content before the module call: 1A 44 3E 18

Function	STL	Accumulator content	Data stack
Buffer the word accumulator in the data stack	PSW	1A443E18	1A443E18
Call subroutine 15	CM 15		
Restore data stack into word accumulator.	PLW	1A443E18	1A443E18

9.8.63 PULL LOGIC ACCUMULATOR FROM THE DATA STACK (PLL)

Syntax: PLL (PULL LOGICACCU)

Operands: None

Action:

The PLL command is the counterpart to the PSL command. Data that has been buffered with the PUSH command can be restored from the data stack by using the PULL command. The iTNC copies bit 7 of the data stack's current address into the logic accumulator. If the stack is empty, the iTNC displays an error message.

Example:

See PSW command.

9.8.64 PULL WORD ACCUMULATOR FROM THE DATA STACK (PLW)

Syntax: PLW (PULL WORDACCU)

Operands: None

Action:

The PLW command is the counterpart to the PSW command. Data that has been buffered with the PUSH command can be restored from the data stack by using the PULL command. During word processing, the iTNC copies with the PLW command two words of the current data stack address into the word accumulator. If the stack is empty, the iTNC displays an error message.

Example:

See PSW command.

9.8.65 UNCONDITIONAL JUMP (JP)

Syntax: JP (JUMP)

Operands: Label (LBL)

Action:

After a JP command, the iTNC jumps to the label that you have entered and resumes the program from there. The JP command interrupts a logic sequence.

Example:

See JPT command.

9.8.66 JUMP IF LOGIC ACCUMULATOR = 1 (JPT)

Syntax: JPT (JUMP IF TRUE)

Operands: Label (LBL)

Action:

The JPT command is a conditional jump command. If the logic accumulator = 1, the iTNC resumes the program at the label that you have entered. If the logic accumulator = 0, the iTNC does not jump. The JPT command interrupts a logic sequence.

Example:

This example also applies to the JP and JPF commands.

Depending on the state of the input I5, skip a certain program section.
Initial state:
Input I5 = 1

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I5	1	1
If logic accumulator =1, jump to LBL 10.	JPT 10	1	
Skip the function.	L I3		
Skip the function.	O M500		
Skip the function.	= 020		
Label	LBL 10		
Resume the program run.	L M100	0	0

9.8.67 JUMP IF LOGIC ACCUMULATOR = 0 (JPF)

Syntax: JPT (JUMP IF FALSE)

Operands: Label (LBL)

Action:

The JPF command is a conditional jump command. If the logic accumulator = 0, the iTNC resumes the program at the label that you have entered. If the logic accumulator = 1, the iTNC does not jump. The JPF command interrupts a logic sequence.

Example:

See JPT command.

9.8.68 CALL MODULE (CM)

Syntax: CM (CALL MODULE)

Operands: Label (LBL)

Action:

After a CM command, the iTNC calls the module that begins at the label that you have entered. Modules are independent subroutines that must be ended with the command EM. You can call modules as often as you wish from different places in your program. The CM command interrupts a logic sequence.

Example:

See CMF command.

9.8.69 CALL MODULE IF LOGIC ACCUMULATOR = 1 (CMT)

Syntax: CMT (CALL MODULE IF TRUE)

Operands: Label (LBL)

Action:

The CMT command is a conditional module call. If the logic accumulator = 1, the iTNC calls the module that begins at the label that you have entered. If the logic accumulator = 0, the iTNC does not call the module. The CMT command interrupts a logic sequence.

Example:

See CMF command.

9.8.70 CALL MODULE IF LOGIC ACCUMULATOR = 0 (CMF)

Syntax: CMF (CALL MODULE IF FALSE)

Operands: Label (LBL)

Action:

The CMF command is a conditional module call. If the logic accumulator = 0, the iTNC calls the module that begins at the label that you have entered. If the logic accumulator = 1, the iTNC does not call the module. The CMF command interrupts a logic sequence.

Example:

This example also applies to the CM and CMT commands.

Depending on the state of the input I5, call the Module 10. Initial state:
Input I5 = 0

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I5	0	0
If logic accumulator =0, jump to LBL 10.	CMF 10	0	
Resume main program after module execution.	L M100	1	1
	⋮		
End of main program	EM		
Label: Beginning of module.	LBL 10		
Statement in the module.	L I3	0	0
Statement in the module.	O M500	1	1
Statement in the module.	= O20	1	1
End of module, resume the main program with the command L M100.	EM		

9.8.71 END OF MODULE, END OF PROGRAM (EM)

Syntax: EM (END OF MODULE)

Operands: None

Action:

You must end each program or subroutine (module) with the EM command. An EM command at the end or within a module causes a return jump to the module call (CM, CMT, CMF). The iTNC then resumes the program with the instruction that follows the module call. The iTNC interprets the EM command as program end. The iTNC can reach the subsequent program instructions only through a jump instruction.

9.8.72 END OF MODULE IF LOGIC ACCUMULATOR = 1 (EMT)

Syntax: EMT (END OF MODULE IF TRUE)

Operands: None

Action:

An EMT command causes a return jump to the module call (CM, CMT, CMF) only if the logic accumulator = 1.

9.8.73 END OF MODULE IF LOGIC ACCUMULATOR = 0 (EMF)

Syntax: EMF (END OF MODULE IF FALSE)

Operands: None

Action:

An EMF command causes a return jump to the module call (CM, CMT, CMF) only if the logic accumulator = 0.

9.8.74 LABEL (LBL)

Syntax: LBL (LABEL)

Operands: ASCII name; maximum length: 32 characters

Action:

The label defines a program location as an entry point for the JP and CM commands. You can define up to 1000 jump labels per file. The ASCII name of the label may be up to 32 characters long. However, the iTNC evaluates only the first 16 characters.

For importing global labels, see EXTERN statement.

9.9 INDEX Register (X Register)

You can use the index register for:

- Data transfer
- Buffering results
- Indexed addressing of operands

The index register is 32 bits wide.

You can use the X register anywhere in the program. The iTNC does not check whether the current content is valid. Exception: During indexed write accessing the iTNC checks whether the amount of available address space is exceeded.

Example: = B100[X]

If the permitted addressable storage is exceeded, the iTNC issues a blinking error message: **`PLC: index range incorrect.`** Acknowledge the error message by pressing the END key. After restarting the iTNC you must not acknowledge the **`POWER INTERRUPTED`** message. Switch to the PLC editor, where you will be shown the error line.

> Note
>
> At the beginning of the PLC cycle the iTNC sets the index register to 0. Assign the index register a defined value before using it in your program.

The following addresses are valid:

- Mn[X]
- In[X]
- On[X]
- Cn[X]
- Tn[X]Operand number = n+X
- Bn[X]Operand number = n+X
- Wn[X]Operand number = n+2*X
- Dn[X]Operand number = n+4*X
- BTX Content of index register = operand
- BCX Content of index register = operand
- BSX Content of index register = operand
- Sn[X]String number = n+X
- S#Dn[X]Dialog text number = n+X
- S#En[X]Error text number = n+X
- S#An[X]ASCII code +X
- Sn^XSubstring from X-th character of the n-th string

The types "S", K, and K$ cannot be indexed.

> Note
>
> If you address S#Dn[X] or S#En[X], the iTNC loads the sequence <SUB>Dnnn or <SUB>Ennn in the string accumulator, where nnn is the modified string number.

Commands for operating the index register

The following commands are available for exchanging data between the word accumulator and index register, or between the stack and index register:

- LX (Load index to accu)Index register – word accumulator
- =X (Store accumulator to index)Word accumulator – index register
- PSX (Push index register)Index register – stack
- PLX (Pull index register)Stack – index register
- INCX(Increment index register)
- DECX(Decrement index register)

Error and dialog files

Starting with software version 05, indexed access to error and dialog files is possible via the operands B/W/D.

In the example, a copy of S1 ("TEST" is loaded into S2, and S3 then contains the string "TEST", but only starting from the 3rd character ("ST"). Also, indexed access to error and dialog files is shown:

```
#define Ofs             D100
#define Start           D104
#define offset_dialog   D108
#define offset_error    D112

L      S#"TEST"
=      S1

L      K1
=      Ofs

L      K2
=      Start
=X

L      S0[Ofs]
=      S2

L      S2^Start
=      S3

L      K5
=      offset_dialog
L      S#D10[offset_dialog]
=      S0[offset_dialog]

L      K6
=      offset_error
L      S#E10[offset_error]
=      S0[offset_error]

EM
```

9.10 Commands for String Processing

String processing enables you to use the PLC program to generate and manipulate any texts. Use Module 9082 to display these texts in the PLC window of the screen, and delete them with Module 9080. The iTNC features one string accumulator and 100 string memories (S0 to S99), in each of which you can save up to 128 characters:

String accumulator (characters)	
1	128
x x x x x x x x x x x x x x	x x x x x x x x x x x x x x

String memory (characters)		
	1	128
S0	x x x x x x x x x x x x x x	x x x x x x x x x x x x x x
...	x x x x x x x x x x x x x x	x x x x x x x x x x x x x x

As of software version 340 94x-05, you can also use operands for indexed access to string operands in the PLC program. Until now, the index register had to be used (for examples, see page 1750).

Example

String accumulator (characters)	
1	128
K U E H L M I T T E L E I N	

String accumulator and string memory are volatile, which means that they are erased by the iTNC when power is switched off. The operand "S" is available for string processing. You can use the operand "S" with different arguments.

Operand declaration

The "S" operands are to be used only for string processing. You can target the following addresses with the various arguments:

- Addressing string memory: After the operand designation, specify the number of the desired memory (S0 to S99).
- Address part of a string: Use the address Sn^X (see INDEX Register). The iTNC addresses the substring beginning with the X-th character of the specified string.
- Immediate string: You can also enter a string directly in the PLC program. The text string, which may contain a maximum of 128 characters, must be indicated by quotation marks.
  Example: **`"Coolant 1 on"`**
- Texts from the PLC error message file or from the PLC dialog file: By specifying the line number you can read texts from the active error message file or dialog file: **`PLC-ERROR: S#Exx xx`**: Line number from the PLC error message file (0 to 2047)
  `PLC-DIALOG: S#Dxx xx:` Line number from the PLC dialog file (0 to 2047).
  Enter the string #Exx or #Dxx in the argument <arg> of the string command. The iTNC then saves a 5-byte-long string <SUB> E0xx or <SUB> D0xx (<SUB> = ASCII <SUB>) in the accumulator. Instead of this string, the iTNC reads the line xx of the active error message or dialog file on the screen.
- Enter an ASCII character in the string. Define the ASCII character through its code: S#Axxx

Logical comparisons during string processing

Use the following procedure to compare two strings, depending on the argument:

- Compare the string memory or immediate string, then the iTNC checks both strings character by character. After the first character that does not fulfill the condition of comparison, the control resets the logic accumulator. Then the iTNC does not check the remaining characters. During a comparison, the iTNC always uses the significance of the characters from the ASCII table. This results, for example, in:
  A < B
  AA > A
- If you have entered PLC error messages or PLC dialog texts in the argument, the iTNC compares the position in the error-message file or dialog file (0 to 2047), but not the actual text as in an immediate string.

The processing times depend on the length of the strings.

9.10.1 LOAD STRING (L)

Syntax: L (LOAD)

Operands: S <arg>

Action:

Load the string accumulator. The string that the iTNC is to load is selected through the argument <arg> after the operand designation. See also "Operand declaration."

Example:

See OVWR command.

9.10.2 ADD STRING (+)

Syntax: + (PLUS)

Operands: S <arg>

Action:

Append another string to a string in the string accumulator. The string that the iTNC is to load is selected through the argument <arg> after the operand designation. See also "Operand declaration." The resulting string must not be longer than 128 characters.

Example:

See OVWR command.

9.10.3 STORE STRING (=)

Syntax: = (STORE)

Operands: S <arg>

Action:

Assign the content of the string accumulator to the string memory. The memory into which the iTNC is to copy the string is selected through the argument <arg> after the operand designation. Permissible arguments: 0 to 99 (String memory S0 to S99). See also "Operand declaration."

Example:

See OVWR command.

9.10.4 OVERWRITE STRING (OVWR)

Syntax: OVWR (OVERWRITE)

Operands: S <arg>

Action:

Save the string from the string accumulator in a string memory. This command differs from the = command in that the iTNC does not transfer the "string end" character along with it. In this way you can overwrite the beginning of a string that is already in the string memory. The memory into which the iTNC is to copy the string is selected through the argument <arg> after the operand designation. Permissible arguments: 0 to 99 (string memory S0 to S99). See also "Operand declaration."

Example:

This example also applies to the string commands L, + and =.

Add a string from the string memory S0 to an immediate string. The result is to overwrite the contents of string memory S1. Initial state:
Immediate string = **HYDRAULICS**
String memory S0 = **OIL**
String memory S1 = **COOLANT MISSING**

String memory (characters)		
	1	128
S0	O I L	
S1	C O O L A N T M I S S I N G	
...	...	

Function	STL	String accumulator (characters)	
		1	128
Load the immediate string into the string accumulator	L S "HYDRAUL."	O I L	
Add content of string memory S0 to string accumulator.	+ S0	H Y D R A U L . O I L	
Overwrite content of string memory S1 with content of string accumulator.	OVWR S1	H Y D R A U L . O I L	

Final status:

String memory (characters)		
	1	128
S0	O I L	
S1	H Y D R A U L . O I L M I S S I N G	
...	...	

9.10.5 EQUAL TO command for string processing (==)

Syntax: == (EQUAL)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator and the operand are equal, the condition is true and the iTNC sets the logic accumulator to 1. If they are not equal, the iTNC sets the logic accumulator is set to 0.

Example:

See command <>.

9.10.6 LESS THAN command for string processing (<)

Syntax: < (LESS THAN)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is less than the operand, the condition is true and the iTNC sets the logic accumulator to 1. If the string accumulator is greater than or equal to the operand, it sets the logic accumulator to 0.

Example:

See command <>.

9.10.7 GREATER THAN command for string processing (>)

Syntax: > (GREATER THAN)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is greater than the operand, the condition is true and the iTNC sets the logic accumulator to 1. If the string accumulator is less than or equal to the operand, it sets the logic accumulator to 0.

Example:

See command <>.

9.10.8 LESS THAN OR EQUAL TO command for string processing (<=)

Syntax: <= (LESS EQUAL)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is less than or equal to the operand, the condition is true and the iTNC sets the logic accumulator to 1. If the string accumulator is greater than the operand, it sets the logic accumulator to 0.

Example:

See command <>.

9.10.9 GREATER THAN OR EQUAL TO command for string processing (>=)

Syntax: >= (GREATER EQUAL)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is greater than or equal to the operand, the condition is true and the iTNC sets the logic accumulator to 1. If the string accumulator is less than the operand, it sets the logic accumulator to 0.

Example:

See command <>.

9.10.10 NOT EQUAL command for string processing (<>)

Syntax: <> (NOT EQUAL)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is not equal to the operand, the condition is true and the iTNC sets the logic accumulator to 1. If the string accumulator is equal to the operand, it sets the logic accumulator to 0.

Example:

This example of string processing also applies to the commands =, <, >, <=, >=, <>.

Compare the immediate string with the content of the string memory S0. Depending on the result, call Module 50.
Initial state:
String memory S0 = SPINDLE 2
Immediate string = SPINDLE 1

String memory (characters)	
	1 128
S0	S P I N D L E 2
...	...

Function	**STL**	**String accu. (characters), or logic accu.**
		1 128
Load the immediate string into the string accumulator	L S "SPINDLE 1"	S P I N D L E 1
Gate the content of string memory S0 with content of string accumulator (=, <, >, >=, ...)	<> S0	S P I N D L E 2
If the condition is fulfilled, set logic accumulator to 1 and call the module.	CMT 50	Logic accumulator = 1

9.10.11 Modules for string processing

Module 9070 Copy a number from a string

The iTNC searches a selectable string memory, (S0 to S99), for a numerical value. When the numerical value is first found, the iTNC copies it as a string into another selectable string memory. The iTNC does not check whether a conflict arises between the source and target string. It may overwrite the source string (even then, however, the function of the module is ensured). The iTNC recognizes unsigned and signed numbers, with and without decimal places. Both the period and comma are permitted as decimal point. The iTNC returns the position (in characters) of the first character after the found number in the string memory to be searched.

Call:

PS	K/B/W/D	<Address of the string memory to be searched>
PS	K/B/W/D	<Address of the string memory for the found number>
CM	9070	
PL	B/W/D	<Offset end of numerical string in the searched string memory>

Error recognition:

Marker	Value	Meaning
M4203	0	Number copied
	1	Error. See W1022.
W1022	2	Incorrect address of the source or target string
	11	No number, no string end, or number string has a length of more than 79 characters

Example

```
L S"X-POS.:123"
= S0
PS K+0
PS K+1
CM 9070
PL W520
```

String memory (characters)		Data stack [bit]
	1 ... 10 ... 128	
S0	X - P O S . : 1 2 3	
S1	1 2 3	10
...	...	

Module 9071 Find the string length
The iTNC ascertains the length of the string in a selectable string memory (S0 to S99).

Call:
PS K/B/W/D/S<String no. or string>
CM 9071
PL B/W/D <Length of the string>

Error recognition:

Marker	Value	Meaning
M4203	0	String length found
	1	Error. See W1022.
W1022	2	Invalid immediate strings, address of the source or target string is out of range (S0 to S99), string memory was searched but no string end was found

Module 9072 Copy a byte block into a string
Module 9072 copies a byte block into a string. The module does not check whether the byte block consists of valid ASCII characters. The content of the string may not be correctly displayed (e.g. due to special characters or end-of-string identification). An end-of-string identifier is set automatically after the last copied byte. The entire programmed byte block is always copied, regardless of any end-of-string identifiers within the byte block.

Call:
PS B/W/D/K <Start byte>
PS B/W/D/K <Length of the byte block>
PS B/W/D/K <String number>
CM 9072

Error recognition:

Marker	Value	Meaning
M4203	0	Byte block copied into string
	1	Error code in W1022
W1022	1	Invalid length of the programmed byte block (max. 127 characters)
	3	Invalid address of the start byte
	4	Invalid sum of the start byte and the length of the byte block
	11	Invalid string

Module 9073 Copy a string into a byte block

Module 9073 copies a string into a byte block. The module does not check whether the string consists of valid ASCII characters. The module always copies over the entire programmed length of the byte block, regardless of any end-of-string identifier in the byte block.

Call:

PS	B/W/D/K	<Target byte>
PS	B/W/D/K	<Length of the byte block>
PS	B/W/D/K	<String number>
CM	9073	

Error recognition:

Marker	Value	Meaning
M4203	0	String was copied into byte block
	1	Error code in W1022
W1022	1	Invalid target byte
	2	Invalid length of the programmed byte block (max. 127 characters)
	4	Invalid sum of target byte and length of the byte block
	11	Invalid string

9.11 Submit Programs

Submit programs are subroutines that the PLC submits to the NC for processing. This allows you to solve problems that are very processor-intensive, require program loops, or must wait for external results. It is a prerequisite that these programs are not restricted to a definite time frame. Depending on the processor utilization, the iTNC provides a certain computing power for a submit program. You start submit programs from the PLC program. They can access all data memory areas (M/B/W/D) as the main program can. To prevent problems, ensure that data processed by the PLC program is clearly separated from data processed by the submit program. You can place up to eight submit programs in a queue (submit queue). Each submit program receives an "identifier" (a number between 1 and 255, assigned by the NC), which the iTNC enters in the word accumulator. With this identifier and the REPLY function you can then interrogate whether the program is in the queue, is being processed, or has already been processed. The iTNC processes the submit programs in the sequence in which they were entered in the queue. If errors occur during execution of the submit program, the NC sets the following markers:

		Set	**Reset**
M4200	**Overflow during multiplication**	NC	PLC
M4201	**Division by 0**	NC	PLC
M4202	**Incorrectly executed modulo**	NC	PLC
M4203	**Error status for PLC module**	NC	NC/PLC
M4204	**Reserved for errors that the PLC programmer would like to catch**	NC	NC

The iTNC lists these markers separately in the submit job. This means that the same markers can be edited simultaneously in the PLC run program without changing the original markers. No exact times can be stated for the commands for managing the submit queue.

9.11.1 Calling the submit program (SUBM)

Syntax: SUBM (SUBMIT)

Operands: Label (LBL)

Action:

Assign an identifier (1 to 255) to a labeled subroutine and put it in the queue. At the same time, the iTNC writes the assigned number in the word accumulator. If programs are already entered in the submit queue, the iTNC does not run the addressed program until the programs before it are finished. A submission to the queue may only take place from a PLC program. A SUBM command in a submit program is not possible.

If there is no room in the queue, or if you program the SUBM command in a submit program (nesting), the iTNC assigns the value "0" to the word accumulator.

Example:

See CAN command.

9.11.2 Interrogating the status of a submit program (RPLY)

Syntax: RPLY (REPLY)

Operands: B/W

Action:

Interrogate the status of the submit program with the specified identifier. You must have already stored the identifier in a byte or word when you call the submit program. With the RPLY command and the defined memory address (byte or word containing the identifier) the iTNC transfers one of the following processing states to the word accumulator:

- Word accumulator = 0: Program complete/not in the queue
- Word accumulator = 1: Program running
- Word accumulator = 2: Program in the queue

Example:

See CAN command.

9.11.3 Canceling a submit program (CAN)

Syntax: CAN (CANCEL)

Operands: B/W

Action:

Cancel a submit program with the specified identifier during processing, or remove it from the queue. You must have already stored the identifier in a byte or word when you call the submit program. After you have canceled the program, the iTNC immediately starts the next submit program from the queue. The following PLC modules cannot be canceled at just any location with CANCEL:

- PLC module for access to screen (908X)
- PLC module for reading NC files (909X)

For these modules, you must check with the RPLY command whether the CAN command may be executed.

Example:

This example also applies to the SUBM and RPLY commands.

Depending on input I10, submit the subroutine with the label LBL 300 to the NC for processing. In addition, check the processing status of the subroutine in the main program with the RPLY command, and cancel it with the CAN command, depending on input I11.

Function	STL
Load the content of input I10 into the logic accumulator	L I10
If logic accumulator =0, jump to LBL 100.	JPF 100
Interrogate the status of the submit program and load it into the word accumulator.	RPLY B128
If the word accumulator is not equal to 0, i.e., the submit program has already been transferred to the NC for processing, set the logic accumulator to 1.	<> K+0
If logic accumulator =1, jump to LBL 100.	JPT 100
Call submit program 300.	SUBM 300
Save the identifier of the submit program in byte 128.	= B128
Label	LBL 100
Load the state of input I11 into the logic accumulator	L I11
If logic accumulator =0, jump to LBL 110 (skip the program cancellation).	JPF 110
Cancel the submit program.	CAN B128
Label	LBL 110
	⋮
End of main program	EM
Label: Beginning of the submit program	LBL 300
	⋮
End of the submit program	EM

Always insert submit programs, like any module, at the end of the main program. In this case, the content of the submit program could be a display in the PLC window that is realizable through permanently assigned PLC modules.

9.12 Cooperative Multitasking

You can run several processes in the PLC with cooperative multitasking. Unlike genuine multitasking, with cooperative multitasking information and tasks are exchanged only at places that you define. Cooperative multitasking permits up to eight parallel PLC processes and the submit queue. In a program that you have started with SUBM, you can use commands for changing tasks and controlling events (Module 926x). You should additionally insert a task change between the individual jobs in the submit queue, so that the iTNC can execute parallel processes by the end of a job at the latest. The cyclic PLC main program does not participate in cooperative multitasking, but interrupts a submit job and the parallel processes at whatever their current stage is.

9.12.1 Starting a parallel process (SPAWN)

Syntax: SPAWN <label>

Operands: D

Action:

In the specified double word, the iTNC returns the identifier (, see page 1761). If no process could be started, the iTNC returns the value –1. You can call the spawn command only in a submit job or in another spawn process (maximum of 16 parallel processes are permitted). If a process ends with EM, the iTNC removes it from the memory to provide space.

9.12.2 Control of events

The parallel processes can make events available to one another. This saves processing time otherwise spent in the constant interrogating of operating states by the individual processes. A special feature of event control is the waiting period, during which the process can "sleep" for a programmed time. With this function you can repeat program sections in a slow time grid, for example for display or monitoring functions.

List of events

In the OEM.SYS file, enter the command **PLCEVENTS=** to enter the complete name of an ASCII file in which you define the events. The entries in the event file are listed line by line with the following syntax:
<Job name> ; <condition> ; <event mask>; [comment]

Event	Function
<Job name>	This name is identical with the label specified with the spawn command. The iTNC evaluates only the first 16 characters.
<Condition>	Logical expression in accordance with the C language convention, identical with the syntax used in function FN20 (See "Data transfer PLC > NC program (Q-parameters)" on page 1648). Operand: M/I/O/T/C/B/W/D with a number that is permissible for this type Condition: == Equal to != or <> Not equal to < Less than > Greater than <= Less than or equal to >= Greater than or equal to If you enter no condition, the interruption will continue until the operand = 0.
<Event mask>	Hexadecimally coded mask of events that are triggered if the condition is fulfilled. The constraints defined in Module 9260 apply for bits 16 to 31.

Example

Entry in the OEM.SYS file:

PLCEVENTS=PLC:\EXAMPLE.PEV

Content of the file PLC:\EXAMPLE.PEV:

JOB_1;I5==1;$0010; Event $0010 to process JOB_1, if I5==1
JOB_1;B20==5;$0004; Event $0004 to process JOB_1, if B20==5
AUXJOB;W6 <10;$0100; Event $0100 to process AUXJOB, if W6 <10

The iTNC triggers an event if a particular condition is met after one run of the cyclic PLC program and if this condition was not met after the previous run of the cyclic PLC program (edge formation). The number of events of this type that can be activated simultaneously is limited to 15.

If you produce a PLC process with the spawn command, the iTNC searches the event file for entries for this process. It places all relevant entries in a list that is run after every cycle of the cyclic PLC program. If a PLC process terminates itself, or if you terminate the process by recompiling the PLC program, the iTNC then deletes all entries in the list.

The iTNC does not monitor the entries in the event file. This means that syntactically incorrect entries of incorrect job names do not result in an error message.

The iTNC issues a blinking error message if

- A non-existent event file is listed in the OEM.SYS file (when the first spawn command is executed).
- Due to the number of entries in the event file, more events need to be monitored than the run-time list permits. Maximum number of entries in the run-time list: 15

Process monitor

In the PLC programming mode you can use the PROCESS MONITOR soft key to open a status screen in which the iTNC displays all parallel processes, including the process for the submit queue. In a time interval, which can be set with the "+" and "–" soft keys, the iTNC displays

- the name of the process (**TASKNAME**)
- the current status of the process (**STATE**)
  - executable (**SCHED**)
  - running (**RUN+**)
  - waiting for event (**EVWAIT**)
  - waiting for time period (**TMWAIT**)
  - AND-gating of the bits in the event mask (**AND**)
  - OR-gating of the bits in the event mask (**OR**)
- the event mask (**EVMASK**)
- the PLC module letting the process wait (**MOD**)
- how often the process has changed contexts in the last time interval (**SCHED**).
- how much CPU time the processor has used from the defined time interval (**CPU(ms)**). The iTNC also shows the distribution of CPU time in a bar chart (**RATIO**).

Module 9260 Receiving events and waiting for events

Call the module only in a submit job or spawn job. The module enables a spawn job or submit job to interrogate or wait for the occurrence of one or more events. At the same time, the module triggers a change in context.

Markers 4200 to 4202 and 4204 have undefined changes after the module call. If you transfer the value zero for the event mask, the iTNC returns all set events without deleting them. Otherwise, in a call with a waiting period, the iTNC returns all the requested events and deletes them. For a call without a waiting period, the iTNC returns and deletes the events only if the condition is met.

As of software version 340 49x-05, symbolic operands of the PLC program can be used in PLC Module 9260 if events were defined in a file and the token PLCEVENTS = in the OEM.SYS.

Example:
Token in OEM.SYS: **`PLCEVENTS = PLC:\EVENTS.PEV`**
Entry in the file EVENTS.PEV: **`SPAWN_PROC1; KEYCODE==5; $0008`**
Condition: PLC variable **`KEYCODE`** corresponds to W274
The event 8 will be sent to the spawn process SPWAWN_PROC1 after pressing the Ctrl-E (= 5) key.

If the events are OR-gated, the iTNC returns and deletes only the set events. You can specify the events to be deleted by calling without a waiting period and with an OR gate.

Event bits 16 to 31 are reserved for the operating system:

- Bit 16: BREAK, cancels a function. Setting and reading is permitted. If you transmit this event, the iTNC cancels access to interfaces and the network!
- Bit 17: Reserved, do not use
- Bit 18: Reserved, do not use
- Bit 19: QUIT, acknowledgment of a request. Use this bit only in the immediate context of a request.
- Bit 20 to bit 31: Reserved, do not use

Call:

```
PS    B/W/D/K   <Wait>
                0 = Do not wait
                –1 = Wait
PS    B/W/D/K   <AND/OR>
                0 = OR-gated, otherwise AND-gated
PS    B/W/D/K   <Event mask>
                0 = Available events
CM    9260
PS    B/W/D/K   <Events>
                Read events
```

Error recognition:

Marker	Value	Meaning
M4203	0	Event has been read
	1	Error code in W1022
W1022	2	Incorrect transfer value for <Wait> parameter
	20	Module was not called in a spawn job or submit job

Module 9261 Sending events

With this module you can send events to a spawn or submit job and then interrogate them with Module 9260. You can call the module in the cyclic program section, in submit jobs and in spawn processes. The iTNC addresses the receiver through the identifier that the spawn command has returned. The submit queue is addressed through the identifier $80000000 (not through the identifier returned by the SUBM command!). The iTNC always assigns the events that you send to the submit queue to the job that is running at the time of arrival. If they are not read by this job, they remain for the next one. If you wish the receiver process to start immediately, after Module 9261 you must also call Module 9262 to enable a change of context.

Event bits 16 to 31 are reserved for the operating system (see Module 9260).

Call:

PS	D/K	<Identifier> Identifier from the spawn command of the receiver K$80000000 = submit queue
PS	B/W/D/K	<Events> Events to be triggered, bit encoded
CM	9261	

Error recognition:

Marker	Value	Meaning
M4203	0	Event has been sent
	1	Error code in W1022
W1022	30	Incorrect identifier

Module 9262 Context change between spawn processes

You can call Module 9262 only in a submit job or spawn job. The module switches the context to another PLC process or submit queue if such a process exists and is not waiting for an event or for the expiration of a dwell time. Markers 4200 to 4202 and 4204 have undefined changes after the module call.

Call:
CM 9262

Error recognition:

Marker	Value	Meaning
M4203	0	Context was changed
	1	Error code in W1022
W1022	30	Module was not called in a spawn or submit job

Module 9263 Interrupting a spawn process for a defined time

You can call Module 9263 only in a submit job or spawn job. The module interrupts the calling process for at least the specified time. If other processes or the submit queue are ready to run, the iTNC changes the context to one of these processes. Markers 4200 to 4202 and 4204 have undefined changes after the module call. The waiting period is interpreted as an unsigned number, so that negative values result in very long waiting periods.

Call:
PS B/W/D/K <Waiting period in ms>
CM 9263

Error recognition:

Marker	Value	Meaning
M4203	0	Delay is active
	1	Error code in W1022
W1022	30	Module was not called in a spawn or submit job

9.13 Constants Field (KF)

You can use the constants-field data type to access one of several constants, defined in tabular form, depending on the value of the index register X. You address it with KF <Name>[X], where <Name> is a label indicating the beginning of the constants field. Constants fields must be introduced with the label KFIELD <Name>. This is followed by any quantity (other than zero) of constants and the end label ENDK. Constants fields can only be programmed where the program has previously been concluded with an EM or JP statement. The name of constants fields corresponds to the rules for naming labels.

Addressing

Types of addresses:

- L KF <Name> [X], with X ≥ 0:
  The iTNC transfers the value of the constant defined by X in the constants field <Name>.
- L KF <Name> [X], with X = –1:
  The iTNC transfers the length of the constants field <Name>.
- L KF <Name>:
  The iTNC transfers the absolute address of the constants field <Name>. This is only worthwhile in conjunction with modules (e.g. Module 9200). You can also use this addressing in a constants field.

Example:

Function	STL
Access value field with X = [0 to 3].	L KF VAL_FIELD [X]
Assign one of the constants to word W0.	= W0
End of main program	EM
Define the constants field. Constant to be loaded with X = 0 Constant to be loaded with X = 3 End of the constants field.	KFIELD VAL_FIELD K+10 K+1 K$ABC K–100000 ENDK

The iTNC checks the access to constants fields in the same way as the writing access for indexed operands. X can assume only positive values from 0 to <Length of constants field –1>.

9.14 Program Structures

To design an easily understandable program, divide it into program sequences. Use labels (LBL) as well as conditional and unconditional jumps. If you use structured statements, the compiler creates the labels and jump commands. Remember that using these labels and jump commands reduces the number of available labels accordingly. You can nest structured instructions in up to 16 levels. It is not possible to share levels.

Example:

Correct program structure	Incorrect program structure
IFT	**IFT**
⋮	⋮
WHILEF	**WHILEF**
⋮	⋮
ENDW	**ENDI**
⋮	⋮
ENDI	**ENDW**

The statements IFT, IFF, WHILET, WHILEF, ENDW, UNTILT and UNTILF require a valid gating result in the logic accumulator. They conclude the sequence of gating operations. The statements ELSE, ENDI and REPEAT require that all previous operations sequences have been concluded.

9.14.1 IF ... ELSE ... ENDI structure

The IF ... ELSE ... ENDI structure permits the alternative processing of two program branches depending on the value in the logic accumulator. The ELSE branch is not mandatory. The following commands are available:

- IFT (IF LOGIC ACCU TRUE):
  Following code only if logic accumulator = 1
- IFF (IF LOGIC ACCU FALSE):
  Following code only if logic accumulator = 0
- ELSE (ELSE):
  Following code only if IF is not fulfilled
- ENDI (END OF IF STRUCTURE):
  End of the IF structure

Function	STL
Load input I0 into the logic accumulator	L I0
Run the following code if logic accumulator = 1	IFT
Program code for I0 = 1	⋮
Run the following code if logic accumulator = 0; command can be omitted	ELSE
Program code for I0 = 0, can be omitted	⋮
End of the conditional processing	ENDI

9.14.2 REPEAT ... UNTIL structure

The REPEAT ... UNTIL structure repeats a program sequence until a condition is fulfilled. Under no circumstances can you wait with this structure in the cyclic PLC program for the occurrence of an external event! The following commands are available:

- REPEAT (REPEAT):
  Repeat the program sequence from here.
- UNTILT (UNTIL TRUE):
  Repeat the sequence until the logic accumulator = 1.
- UNTILF (UNTIL FALSE):
  Repeat the sequence until the logic accumulator = 0.

The iTNC runs a REPEAT ... UNTIL loop at least once!

Function	STL
Assign the content of the logic accumulator to marker 100; conclusion of the previous commands	= M100
Repeat the following program code	REPEAT
Program code to be run	⋮
Load the index register	L X
Check the index register	>= K100
Repeat until X >= 100	UNTILT

9.14.3 WHILE ... ENDW structure

The WHILE ... ENDW structure repeats a program sequence if a condition is fulfilled. Under no circumstances can you wait with this structure in the cyclic PLC program for the occurrence of an external event! The following commands are available:

- WHILET (WHILE TRUE):
  Run the sequence if logic accumulator = 1.
- WHILEF (WHILE FALSE):
  Run the sequence if logic accumulator = 0.
- ENDW (END WHILE):
  End of the program sequence, return to the beginning

The iTNC runs a WHILE ... ENDW loop only if at the beginning the WHILE condition is fulfilled. Before the ENDW statement you must reproduce the condition for execution. For the WHILE ... ENDW structure the iTNC generates two internal labels. The condition can also be produced in a way different from before the WHILE statement!

Function	STL
	⋮
Load marker 100 into the logic accumulator; create condition for 1st WHILE scan.	L M100
Run the following code if logic accumulator = 1	WHILET
Program code for logic accumulator = 1	⋮
Produce the condition of repeated execution: Load marker 101 in the logic accumulator and gate the content of marker M102 with AND.	L M101 A M102
Jump back to the WHILE request	ENDW

9.14.4 CASE branch

Indexed module call (CASE)

Syntax: CASE (CASE OF)

Operands: B/W

Action:

Selects a certain subroutine from a list of module calls (CM). These CM commands must follow the CASE statement immediately and are numbered internally in ascending order from 0 to a maximum of 127. The content of the operand (B, W) addresses the desired module. Subsequent entries in the jump table (CM) must have addresses at least four bytes higher than the previous entry.

Example:

See ENDC command.

End of indexed module call (ENDC)

Syntax: ENDC (ENDCASE)

Operands: None

Action:

Use the ENDC command in connection with the CASE command. It must come immediately after the list of CM commands.

Example:

Function	STL
Case command and operand; the internal address of the desired module must be saved in the operand	CASE B150
Call module if operand content = 0 Internal addressing from 0 to max. 127	CM 100
Call module if operand content = 1	CM 200
Call module if operand content = 2	CM 201
Call module if operand content = 3	CM 202
Call module if operand content = 4	CM 203
Call module if operand content = 5	CM 204
Call module if operand content = 6	CM 300
End of the CASE statement	ENDC

9.15 Linking Files

You can store the source code of the PLC program in several files. To manage these files, use the following commands:

- USES
- GLOBAL
- EXTERN

These statements must be located at the beginning of your PLC program—i.e., before the first PLC command. With the USES statement you link another file into the program. The GLOBAL statement supplies a label from its own file for an entry that can be used by all other files. The EXTERN statement provides a label that is defined in another file and is identified there with GLOBAL. You can then call this label from the active file. You can dramatically improve the transparency of your program by dividing your source code by function into individual groups and then save these groups in individual files. The number of labels is not limited. You can link up to 256 files to one program. The total size is only limited by the available memory. If the memory is exceeded the error message **System memory overflow** appears. With multiple files, the main program in the directory must have the "M" status flag. This can be done in RAM by using the PLC program function "COMPILE" once and choosing the main program in the file window.

9.15.1 USES STATEMENT (USES)

Syntax: USES <file name>

Operands: None

Action:

You can use the USES statement in the main program to link other files. Files that are linked with USES can themselves also use the statement to link further files. It is also permissible to use the USES statement to link a single file to several other files. The code for this file is generated only once. The USES statement requires a file name as an argument. The USES statement only links a file; it does not run the file's program code. The USES statement cannot be compared with a CM statement. The linked files must therefore contain individual modules that you can then call with the CM statement.

Example:

USES PLCMOD1
USES EPRUPG
USES RAMPLC

Linking of files:

Function	**STL**
Main program	PLCMAIN.PLC
Link the file for spindle control.	USES SPINDLE.PLC
Link the file for tool change.	USES TCHANGE.PLC
Program code	⋮

Function	**STL**
File for spindle control	SPINDLE.PLC
Integrate file with general subroutines.	USES PLCUPG.PLC
Program code	⋮

Function	**STL**
File for tool change	TCHANGE.PLC
Integrate file with general subroutines.	USES PLCUPG.PLC
Program code	⋮

Function	**STL**
File with general subroutines	PLCUPG.PLC
Program code	⋮

9.15.2 GLOBAL statement (GLOBAL)

Syntax: GLOBAL <Label, declaration beyond the file boundary>

Operands: None

Action:

There is no limit to the number of labels in each file linked with USES. To enable a module that was defined in a file to be called from another file, you must declare the module to be global. This is done by entering the GLOBAL statement at the beginning of the file. You can set labels globally only if they are defined with LBL (and not with KFIELD!) later on in the program.

The main program must not contain any GLOBAL definitions. A single label cannot be declared global by more than one module. However, a name that is declared global in file A can be used again locally in file B. The number of labels is not limited.

9.15.3 EXTERN STATEMENT (EXTERN)

Syntax: EXTERN <Label, a module from another file can now be called with a CM command>

Operands: None

Action:

To enable a label in one file to access modules that other files have declared as GLOBAL, you must declare the label with EXTERN. You must write the EXTERN statement at the beginning of the file. In the program code you can then jump to this label with the commands CM, CMT and CMF.

The following functions are not permitted with external labels:

- JP, JPF, JPT
- Access to a constants field
- Linking a CM statement in a CASE branch

The name of the external label cannot be used again as a local label in the same file.

9.16 PLC Modules

A number of PLC modules are available for PLC functions that are very difficult or even impossible to perform with PLC commands alone. You will find descriptions of these modules under the corresponding functions. (See "Overview of Modules" on page625.)

If the iTNC does not execute a module correctly, it sets M4203 and transmits an error code in W1022. You can then evaluate M4203 and W1022 in order to display an error message. If desired, the errors of PLC modules can be logged in a special log book (See "Logs" on page 1265).

		Set	Reset
M4203	**Error status for PLC module**	NC	NC/PLC
W1016	**PLC module that was last processed erroneously**	NC	NC
W1022	**Error status of the last called PLC module**	NC	NC

9.16.1 Markers, bytes, words, and double words

Module 9000/9001 Copy in the marker or word range
Modules 9000 (markers) and 9001 (byte/word/double) copy a block with a certain number of markers or bytes, beginning with the start address, to the specified target address. For module 9001 the length should always be defined in bytes.

The iTNC copies sequentially, beginning with the first memory cell. Therefore the function is not ensured if the source block and the target block overlap and the source block begins at a lower address than the target block. In this case the iTNC overwrites the overlapping part of the source block before the copying process.

Call:

```
PS    B/W/D/K   <Number of the 1st marker in source block>
PS    B/W/D/K   <Number of the 1st marker in target block>
PS    B/W/D/K   <Length of block in markers>
CM    9000

PS    B/W/D/K   <Number of the 1st word in source block>
PS    B/W/D/K   <Number of the 1st word in target block>
PS    B/W/D/K   <Length of block in markers>
CM    9001
```

Error recognition:

Marker	Value	Meaning
M4203	0	Markers, bytes, words, or double words were copied
	1	Error code in W1022
W1022	1	Operand address invalid
	2	Address too high or block too long
	4	Programmed source or target block too long

Module 9010/9011/9012 Read in the word range

From the specified location in the word memory the iTNC reads a byte, word or double word and returns it as an output quantity to the stack. Indexed reading is possible by specifying a variable as designation of the memory location.

Call:

```
PS   B/W/D/K  <Address of the byte to be read>
CM   9010     ; READ BYTE
PL   B        <Target address for byte that was read>

PS   B/W/D/K  <Address of the word to be read>
CM   9011     ; READ WORD
PL   B        <Target address for word that was read>

PS   B/W/D/K  <Address of the double word to be read>
CM   9012     ; READ DOUBLE WORD
PL   B        <Target address for double word that was read>
```

Error recognition:

Marker	Value	Meaning
M4203	0	Byte was read
	1	Error code in W1022
W1022	3	Invalid address was programmed
	5	Module 9011: Specified address is not a word address Module 9012: Specified address is not a double word address

Example of Module 9010

Initial state:
Byte B10 = 35 (address)
Byte B35 = 80 (byte to be read)
Byte B100 = ?

Function	STL	Accumulator content (dec)	Data stack (dec)
Save the address (B10) of the byte to be read from the word accumulator in the data stack	PS B10	35	35
Read byte B35 and save in the data stack	CM 9010		80
Save data stack in byte B100.	PL B100	80	80

Module 9020/9021/9022 Write in the word range

The iTNC writes the given byte, word or double word to the defined location in the word memory. Indexed writing is possible by specifying a variable as designation of the memory location.

Call:

```
PS    B/W/D/K  <Address of the byte to be written>
PS    B/W/D/K  <Byte to be written>
CM    9020     ; WRITE BYTE TO ADDRESS

PS    B/W/D/K  <Address of the word to be written>
PS    B/W/D/K  <Word to be written>
CM    9021     ; WRITE WORD TO ADDRESS

PS    B/W/D/K  <Address of the double word to be written>
PS    B/W/D/K  <Double word to be written>
CM    9022     ; WRITE DOUBLE WORD TO ADDRESS
```

Error recognition:

Marker	Value	Meaning
M4203	0	Byte was written
	1	Error code in W1022
W1022	3	Invalid address was programmed
	5	Module 9021: Specified address is not a word address Module 9022: Specified address is not a double word address

Example of Module 9020

Initial state:
Byte B10 = 35 (address)
Byte B100 = 120 (byte to be written)
Byte B35 = ?

Function	STL	Accumulator content (dec)	Data stack (dec)
Save the address (B10) of the byte to be written from the word accumulator in the data stack	PS B10	35	35
Save byte B100 from the word accumulator in the data stack	PS B100	120	120
Write data stack to byte B35	CM 9020	120	

9.16.2 Number conversion

Module 9050 Conversion from binary → ASCII

Module 9050 converts a binary numerical value consisting of a mantissa and exponent to base 10 into an ASCII-coded decimal number and saves it as a string in the specified address. The exponent refers to the least significant place of the number. The iTNC detects a negative number when the mantissa corresponds to a negative number in the notation as a two's complement. The iTNC sets an algebraic sign only before negative numbers. The control does not convert trailing zeros after the decimal point or leading zeros before the decimal point. The iTNC writes the string left-aligned in the string address that you specify.

Constraints:
The decimal character is defined by Machine Parameter MP7280 as a comma (MP7280 = 0) or a period (MP7280 = 1).

Call:

```
PS    B/W/D/K   <Mantissa of the number to be converted>
PS    B/W/D/K   <Exponent to base 10 of the value>
PS    B/W/D/K   <String address in which the iTNC saves the ASCII-coded
                decimal number>
CM    9050
```

Error recognition:

Marker	Value	Meaning
M4203	0	Number was converted
	1	See W1022 for error code
W1022	2	Invalid string address or invalid exponent

Module 9051 Conversion from binary → ASCII

Module 9051 converts a binary numerical value into an ASCII-coded decimal number in the specified format and saves it as a string in the specified address. The number is interpreted as a two's complement. For algebraically unsigned notation, the iTNC converts the absolute amount of the number without putting a sign before the string. For algebraically signed notation, the iTNC sets an algebraic sign (+ or –) in front of the string in any event. For notation in inches, the number is divided by 25.4 before conversion. If the number has more decimal places than the total that you have specified for the number of places before and after the decimal point, then the iTNC omits the most highly significant decimal places. In right-aligned notation leading zeros before the decimal point are replaced by blanks; in left-aligned notation they are suppressed. Trailing zeroes after the decimal point are always converted.

Constraints:
The decimal character is defined by Machine Parameter MP7280 as a comma (MP7280 = 0) or a period (MP7280 = 1).

Call:

PS	B/W/D/K	<Numerical value to be converted>
PS	B/W/D/K	<Display modes, bit-encoded> Bit 1/0: Format 00: Sign and number left-aligned 01: Sign left-aligned, number right-aligned 10: Sign and number right-aligned 11: Not permitted Bit 2: Display converted to INCH Bit 3: Display with sign
PS	B/W/D/K	<Number of places after the decimal point>
PS	B/W/D/K	<Number of places before the decimal point>
PS	B/W/D/K	<String address in which the iTNC saves the ASCII-coded decimal number>
CM	9051	

Error recognition:

Marker	Value	Meaning
M4203	0	Number was converted
	1	Error code in W1022
W1022	2	Invalid string address, invalid display mode or invalid number of places before or after the decimal point

Module 9052 Conversion from ASCII → binary

Module 9052 converts an ASCII-coded decimal number (possibly with decimal places) into a signed number and an exponent to the base of 10. You must assign the ASCII-coded decimal number to one of the string memories. If the number has no algebraic sign, the iTNC interprets it as a positive number and accepts both a point and a comma as decimal character. If the full extent of the mantissa cannot be represented in a double word, then the last places are omitted and the exponent is corrected accordingly. If possible, the iTNC adjusts the exponent so that it corresponds to the ASCII notation.

Call:

PS	B/W/D/K	<String address in which the ASCII-coded decimal number is saved>
CM	9052	
PL	B/W/D	<Numerical value>
PL	B/W/D	<Exponent to the base of 10 of a value>

Error recognition:

Marker	Value	Meaning
M4203	0	Number was converted
	1	See W1022 for error code
W1022	2	Invalid string address or string contains none or too many characters

Module 9053 Conversion from binary → ASCII/hexadecimal

Module 9053 converts blocks of binary values from the word-marker range into a string of ASCII-coded hexadecimal numbers. The iTNC reads the specified number of bytes from the word address that you have specified and converts it to a hexadecimally coded ASCII string. Each byte produces two characters in the string memory.

Call:

PS	B/W/D/K	<Word address from which the binary values are saved>
PS	B/W/D/K	<String address in which the iTNC saves the hexadecimal numbers>
PS	B/W/D/K	<Number of data bytes>
CM	9053	

Error recognition:

Marker	Value	Meaning
M4203	0	Number was converted
	1	See W1022 for error code
W1022	1	Too many data bytes
	2	Invalid string address
	4	Invalid word address

Module 9054 Conversion from ASCII/hexadecimal → Binary

Module 9054 converts strings of ASCII-coded hexadecimal values into a block of binary values in the word-marker range. The string in the specified string memory is interpreted as a sequence of ASCII-coded hexadecimal numbers and converted into a block of corresponding binary bytes. Two ASCII characters produce one binary byte. The iTNC saves the binary block beginning at the specified address in the word-marker range.

Call:

PS B/W/D/K <String address in which the hexadecimal value is saved>
PS B/W/D/K <Word address from which the iTNC saves the binary values>
CM 9054

Error recognition:

Marker	Value	Meaning
M4203	0	Number was converted
	1	See W1022 for error code
W1022	2	Invalid string address
	11	Invalid word address
		Odd number of characters in the string or a character that cannot be interpreted as hexadecimal

Example

Initial state:
SO = ”63”
BO = 99

Function	STL	String accu. (characters), data stack [bits]
Push string address S0 onto the data stack.	PS K+0	63
Push word address B0 onto the data stack.	PS B0	99
Conversion of the two ASCII characters 6 and 3 into the binary number 99.	CM 9054	01100011

9.17 Python

Starting with software 340 49x-04, the "Python OEM Process" option (software option #46) provides you with an effective tool for using an object-oriented high-level programming language in the control (PLC).

"Python OEM Process" can be used for machine functions and complex calculations, as well as to display special information. It is especially useful for the implementation of user-specific or machine-specific solutions, regardless of whether special algorithms or interfaces for special functions have to be created.

The applications created can be displayed by the PLC in the familiar PLC windows or in separate freely definable windows that can be expanded to the TNC's full screen size.

The system requires at least 512 MB of RAM. The RAM of the MC can be checked in the **herosdiagnose.txt** file. The RAM of the MC is listed in [kB] under the **Total Memory heading.**
Proceed as follows to create the **herosdiagnose.txt file:**

- While in the Programming and Editing operating mode, press the MOD key.
- Press the DIAGNOSIS soft key, and then the HEROS DIAGNOSIS soft key.

A separate documentation exists for Python OEM Process, which you can download free of charge from the HEIDENHAIN FileBase under "PC software > Python" together with simple examples and additional PC tools for debugging and for the development of user interfaces.

With software version 340 49x-05, a new and simpler JH library of the HEIDENHAIN functions for Python will be introduced. It simplifies working with windows in the user interface of the control, and provides many new functions. These improvements, however, make the JH library 2.0 incompatible with the JH library 1.0, which was introduced with software version 340 49x-04. Python scripts created for JH library 1.0 are not executable with the active library 2.0.

However, library 1.0 is still available in software version 340 49x-05. A request at the beginning of the Python script enables you to switch between the JH library versions.
If the Python script does not contain any request, JH library version 1.0 will be used automatically. This way, existing Python scripts can be run on software version 340 49x-05 without the need for modification.

Please refer to the Python documentation for software 340 49x-05. It is also available on the HEIDENHAIN FileBase under "PC Software > Python".

10 Data Interfaces

10.1 Introduction

In addition to their Central Processing Unit (CPU), computer systems usually include various peripheral devices.

A CPU is, for example:

- PC
- Control

Peripheral devices include:

- Printers
- Scanners
- External storage devices, such as hard disks, floppy-disk drives or USB memory sticks
- Other computer systems

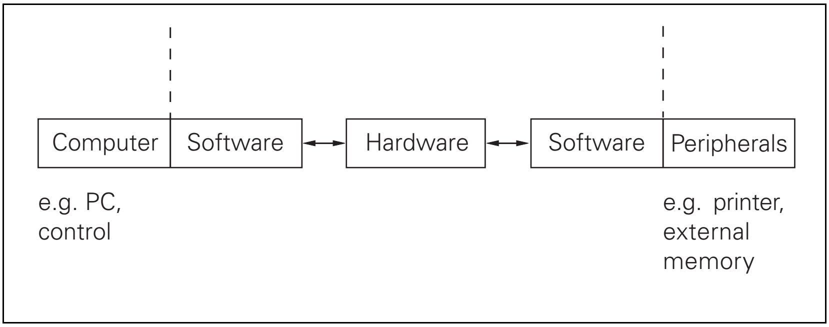
A data interface makes it possible for the CPU and its peripheral devices to communicate.

The interfaces, which consist of physical links between the computer system and the peripherals, need a transmission line and appropriate software in order to transfer data between the individual units.

Standard interfaces include:

- Ethernet
- USB 1.1
- RS-232-C/V.24 or
- RS-422/V.11

The relationship between hardware and software, which fully defines an interface, is illustrated by the following diagram:



The hardware in the diagram covers all the physical components, such as

- Circuit design
- Pin layout
- Electrical characteristics

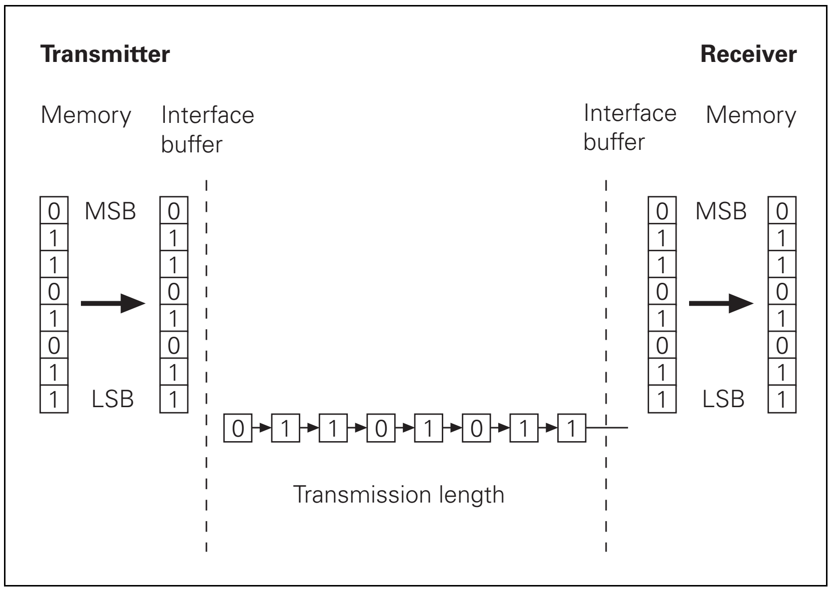
The software is the operating software, which includes, for example, the drivers for the output modules.

10.1.1 Principles of data transfer

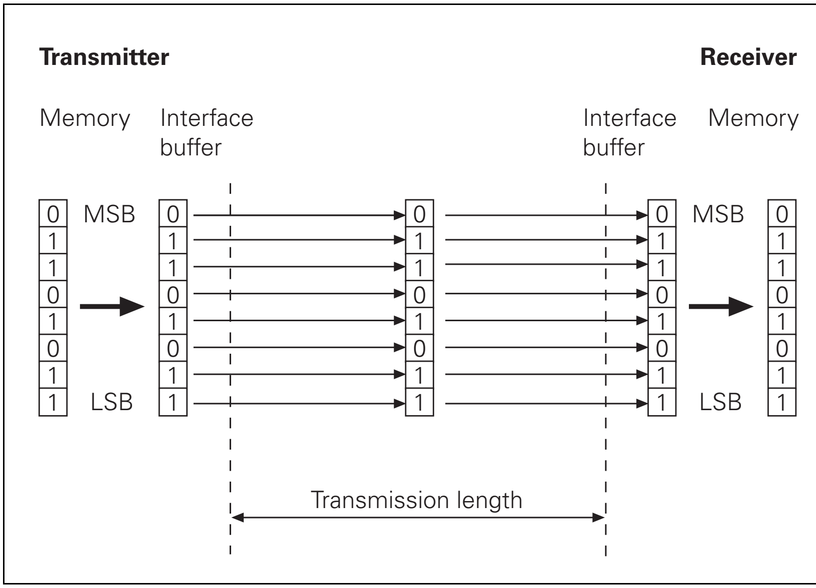
The term "data" is used to describe all of the information that the computer is capable of collecting and processing.

Serial/Parallel

Data can be transmitted in either serial or parallel format.
Basically, data is coded in the computer system, e.g. as bytes (8 bits), and supplied to the interface in parallel.
In the case of serial data transmission, the parallel information from the computer system has to be converted into a serial data flow by a USART (Universal Synchronous/Asynchronous Receiver/Transmitter).
The receiver accepts the serial data flow and converts it back again into parallel information:



For parallel data transmission, the interface needs line drivers, e.g. a 36-pin ribbon cable, instead of the USART illustrated above.



Advantages of serial data transmission:

- Economical
- Ideal for covering long distances

Disadvantages of serial data transmission:

- Slow

Advantages of parallel data transmission:

- Fast

Disadvantages of parallel data transmission:

- Somewhat more expensive

Data coding

A common code for data transfer is the ASCII code (American Standard Code for Information Interchange), which codes all characters with seven bits. This means that, in all, $2^7 = 128$ characters are coded.

Example:

The letter “z” is represented by the following combination of bits:

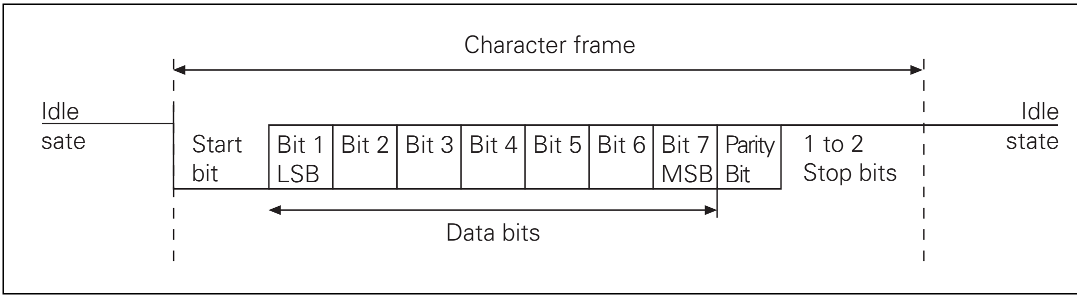
1 1 1 1 0 1 0 = 122 dec = 7A hex

When the letter “z” is transmitted via a serial interface, the appropriate characters are sent one after the other.

Synchronization

A synchronization process ensures that the receiver correctly detects the first character of a transmission.

With an asynchronous character frame, the transmission of a data word can begin at any time, starting from an idle state.



A start bit is transmitted before each data word. If the first bit of the data word had the same value as the idle state, the receiver would not notice any difference from the idle state.

After the start bit has been sent, the data word is transmitted, bit by bit, starting with the LSB (Least Significant Bit). The MSB (Most Significant Bit) of the data word is followed by the so-called parity bit in order to detect transmission errors.

The character frame is concluded by one or two stop bits. The stop bits enable the receiver to recognize the transmitter again before the start of the next character.

Synchronization is repeated before each character and applies for one character frame.

Transmission reliability: Parity bit

With an asynchronous character frame, transmission errors can be detected by using the parity bit.

The parity bit can take three different forms.

- No parity check (no parity): Error detection is dispensed with.
- Even parity:
  The transmitter counts bits with a value of one. If the number is odd, the parity bit is set to one, otherwise it is cleared to zero. The sum of set data bits and the parity bit is therefore always even. Upon receiving a word, the receiver counts all of the set bits, including the parity bit. If the count is odd, there is a transmission error and the data word must be repeated, or an error message will be displayed.
- Odd parity:
  The parity bit is so chosen by the transmitter that the total number of all the set bits is odd. An error will thus be detected if the receiver observes an even number of set bits in its evaluation.

Example

The letter "z" corresponds to the bit sequence: 1 1 1 1 0 1 0

Parity bit

- with even parity = 1
- With odd parity = 0

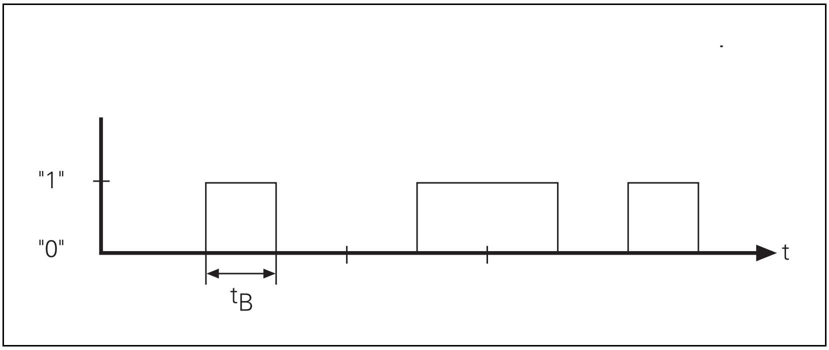
Data transfer rate

The data transfer rate is given in bits per second.

Common transfer rates are:

110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps

The time taken to transmit one bit (t_B) can be calculated from the transfer rate:



$$t_B = \frac{1}{\text{data transfer rate (bps/s)}}$$

For example, a transfer rate of 19 200 bps will have a bit duration of t_B = 52.083 µs.

$$t_B = \frac{1}{19200\ \text{(bps/s)}} = 52{,}083\ \mu s$$

The number of characters transmitted per second can be calculated from the transfer rate and the transmission format:

$$\text{characters per second} = \frac{\text{data transfer rate (bps/s)}}{\text{number of bit per characters}}$$

Example:

With a transmission format of one start bit, 7 data bits, two stop bits and a data transfer rate of 300 bps, exactly 30 characters per second will be transmitted.

$$\text{characters per second} = \frac{300\ \text{(bps/s)}}{1 + 7 + 2} = 30$$

10.1.2 Data transfer check: handshaking

By handshaking, two devices control data transfer between them. A distinction is made between software handshaking and hardware handshaking.

You can choose either of the two procedures:

Hardware handshaking

Data transfer is controlled by electrical signals. Information, such as Clear to Send (CTS), Request to Send (RTS), "Start transmission" and "Stop transmission", is passed on by the hardware.

Example:

When a computer is to transmit a character, it checks the CTS signal line to see whether it is active (ON). If it is, the character is transmitted.

Hardware handshaking requires:

- The data lines TxD and RxD (transmitted and received data)
- The RTS control line (switch transmitting unit on)
- The CTS signal line (Clear to Send)
- A ground connection

Software handshaking

Control of data transfer is achieved by control characters transmitted via the data line.

Example: XON/XOFF method with the RS-232-C/V.24 interface

The meaning XON is assigned to control character DC1 and the meaning XOFF to control character DC3. Before transmitting a character, the computer checks whether the receiving unit is transmitting the XOFF character. If it is, the computer delays transmission until it receives the character XON. XON indicates that the connected unit is ready to receive further characters.

Software handshaking requires:

- The data lines TxD and RxD (transmitted and received data)
- A ground connection

10.2 The Ethernet Interface of the iTNC

General information

You can integrate the iTNC into your plant's intranet or use a transposed cable to connect directly with a PC. The data transfer rate is dependent on the amount of traffic at the time on the net. For information on the pin layout: See "X26: Ethernet interface RJ45-port" on page 400.

The iTNC requires an NFS server (Network File System) or a Windows PC (SMB = Server Message Block) as the remote station. It must operate according to the TCP/IP protocol principle.

OSI 7-layer model		iTNC
7	Application layer	NFS, SMB
6	Presentation layer	
5	Communications layer	
4	Transport layer	TCP protocol
3	Network layer	IP protocol
2	Data link layer	Ethernet card
1	Physical layer	

Before networking, the TNC must be properly configured. Please discuss the required settings with your network supervisor.

SMB signing

HEIDENHAIN does not support SMB signing, which is a further development from Microsoft as part of the CIFS (Common Internet File System). It is a method for authentication, and is intended to thwart a "Man in the middle" attack. In such an attack, computer C intrudes into a connection between computers A and B. It then forces computer A to send all packages intended for B to C instead, where the packages are modified before being passed on to B.

The CIFS is not included with the HEIDENHAIN SMB driver. HEIDENHAIN supports SMB, in the version familiar from Windows 2000, but not SMB signing.

X26: Ethernet interface RJ45 port (100BaseT)

Maximum cable length:
Unshielded: 100 m
Shielded: 400 m

Network topology: Star configuration

This means a hub serves as a central node that establishes the connection to the other participants.

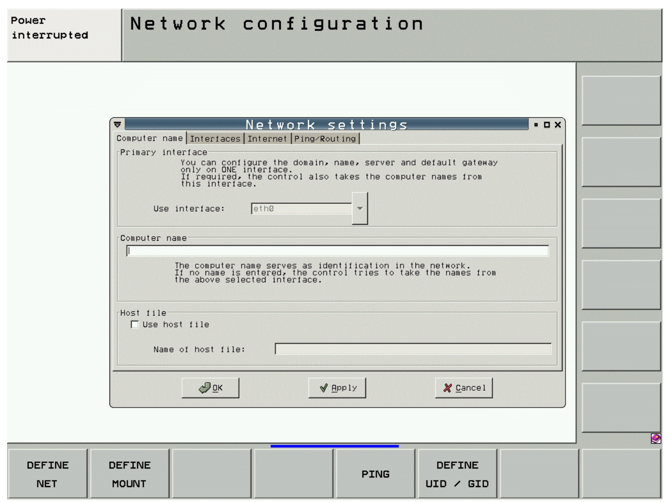
10.3 Connecting the iTNC to the Network

10.3.1 Settings at the iTNC

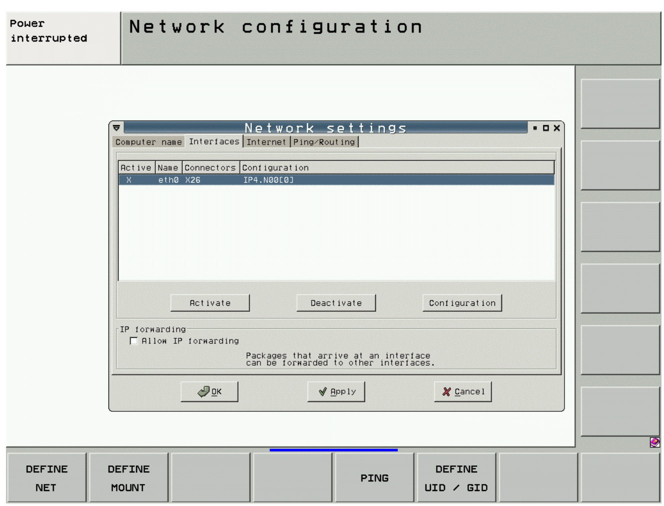
For the network settings of the iTNC 530 with Windows 2000: See "Network Settings" on page 1856.

- Press the MOD key in the **Programming and Editing** operating mode and enter the code number NET123. Use the soft keys to select the required network options:

Soft key	Meaning
DEFINE NET	Settings on the iTNC 530 for networking. The network settings for connecting the iTNC 530 to a network were completely revised for software 340 49x-05, and are divided into several pages of a window, which can be selected by tabs.

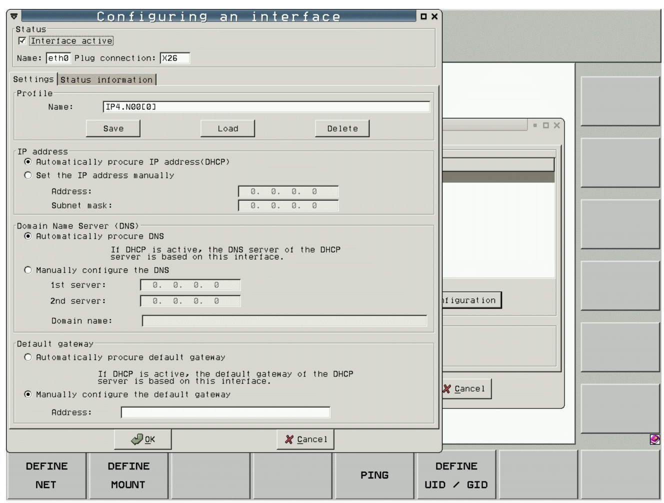


Tab	Option	Meaning
`Computer name`	Settings on the iTNC 530 for networking. The name of the control as well as other data can be defined.	
	`Primary interface`	Name of the Ethernet interface to be integrated in your company network. Only active if a second, optional Ethernet interface (MC 62xx) is available on the control hardware.
	`Computer name`	The iTNC registers itself in the network with this name, which you can also use to reach it (with TNCremoNT, for example). If you use a host-name server, you must enter the “Fully Qualified Host Name” here. If you do not enter a name here, the iTNC uses the so-called null authentication. If null authentication is used, the entries under UID, GID, DCM and FCM are ignored.
	`Host file`	Only required for special applications: Name of a file in which the assignments of IP addresses to computer names are defined. Then you can enter the names instead of the IP addresses in the network settings. Example: **`DOMAIN = NET.A`** File NET.A: **`PC1 160.1.180.20`** **`PC2 160.1.180.21`** **`...`**



Tab	Option	Meaning
Interfaces	List of the active Ethernet interfaces with information as to whether the interface is active, the name, the connector and the IP configuration ■ **Activate** button: Activate the selected interface (the letter **X** appears in the **Active** column) ■ **Deactivate** button: Deactivate the selected interface (a dash (–) appears in the **Active** column) ■ **Configuration** button: Open the Configuration menu	
	IP forwarding	This function must be inactive by default. Only activate this function if external access to the second, optional Ethernet interface of the control (available on the MC 62xx) via the TNC is necessary for diagnostic purposes.

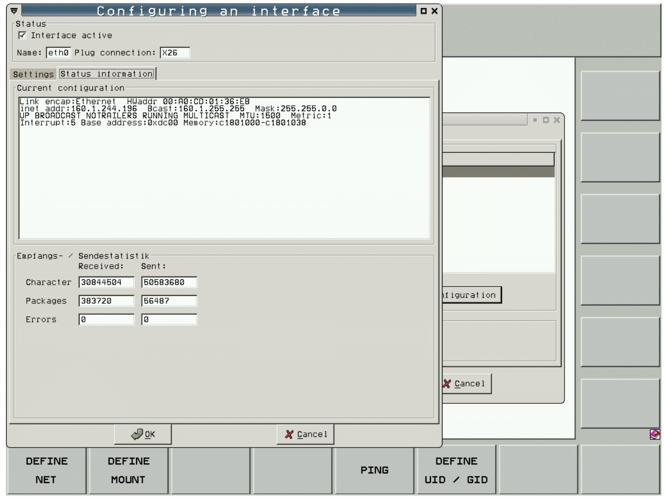
▶ Press the **Configuration** button below the interface list to open the Configuration menu:



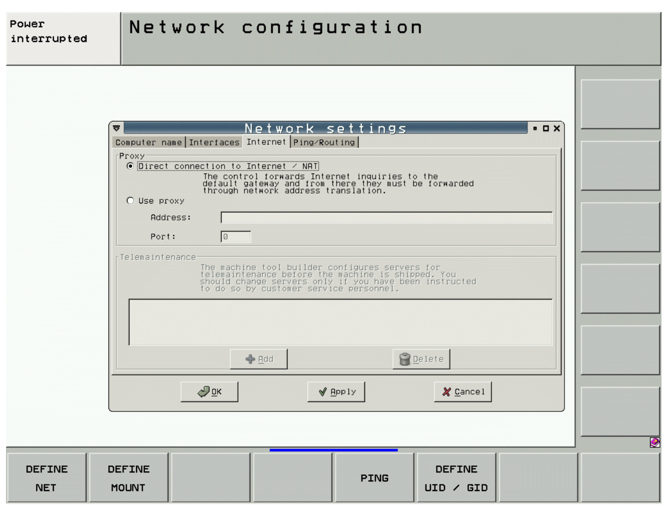
Option	Meaning
Status	■ **Interface active:** Status of the selected Ethernet interface. ■ **Name:** Name of the interface you are currently configuring. ■ **Plug connection:** Number of the plug connection of this interface on the logic unit of the control.
Profile	Here you can create or select a profile in which all settings shown in this window are stored. HEIDENHAIN provides two standard profiles: ■ **LAN-DHCP:** Settings for the standard TNC Ethernet interface, should work in a standard company network. ■ **MachineNet:** Settings for the second, optional Ethernet interface; for configuration of the machine network (available on the MC 62xx). Press the corresponding buttons to save, load and delete profiles.
IP address	■ **Automatically procure IP address:** The control is to procure the IP address automatically from the DHCP server. ■ **Set the IP address manually:** Manually define the IP address and subnet mask. Enter as four decimal numbers separated by points (dotted-decimal notation, for example: **160.1.180.20** and **255.255.0.0**). Your network specialist determines both addresses of the iTNC.
Domain Name Server (DNS)	■ **Automatically procure DNS:** The control is to automatically procure the IP address of the DNS server. ■ **Manually configure the DNS:** Manually enter the IP addresses of the servers and the domain name. Your network specialist determines both entries.
Default gateway	■ **Automatically procure default gateway:** The control is to automatically procure the default gateway. ■ **Manually configure the default gateway:** Manually enter the IP address of the default gateway. Your network specialist determines the address.

▶ Press the **OK** button to confirm the settings, or press the **Cancel** button to reject the settings.

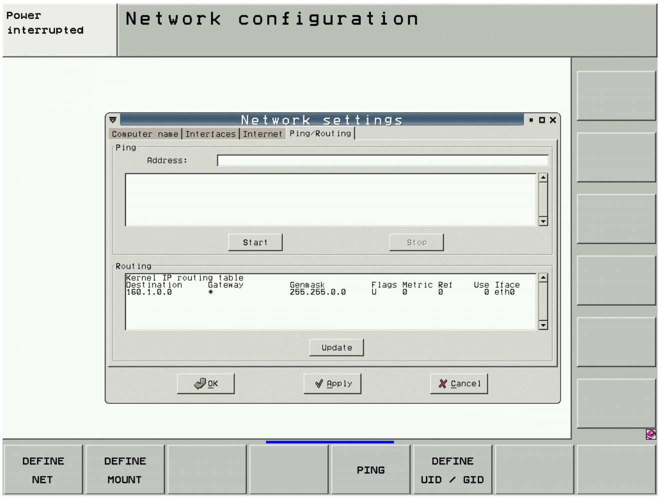
▶ Select the **Status Information** tab in the configuration menu of the interface:



Option	Meaning
`Current configuration`	Information for your network specialist about the current configuration of the Ethernet interface.
`Reception/ Transmission statistics`	Information for your network specialist about the received and transmitted characters, packets and errors.



Tab	Option	Meaning
Internet	The Internet tab does not have any function yet in software 340 49x-05.	



Tab	Option	Meaning
Ping/ Routing	Entry of ping and routing settings:	
	Ping	In the **Address:** field, enter the IP number for which you want to check the network connection. Input: Four numerical values separated by points, e.g. **160.1.180.20**. As an alternative, you can enter the name of the computer whose connection you want to check. ■ **Start** button: Start the test. The control shows status information in the Ping field. ■ **Stop** button: Conclude the test.
	Routing	For network specialists: Status information of the operating system for the current routing process. ■ **Update** button: Update routing.

Configuring an interface

To log your control onto a standard network, proceed as follows:

- Enter the code number NET123 in the PROGRAMMING AND EDITING operating mode.
- Press the DEFINE NET soft key
- In the **Computer name** field, enter the name under which the control is to identify itself in the network.
- Select the **Interfaces** tab. Select the name of the interface **(eth0),** connector **X26** and press the **Configuration** button.
- In the **Settings** of the interface under **Profile**, press the **Load** button.
- Select the profile **LAN-DHCP**, and press **OK** to confirm your selection.
- All other settings can be left at "Automatically procure", unless required otherwise for your network.
- To leave the **Settings**, press the **OK** button.
- Then press the **OK** button again to confirm your changes in the **network settings**.
- Then restart the control with the **Restart** button.

Soft key	Option	Meaning
DEFINE MOUNT	Definition of the devices in the network that can be addressed from the iTNC. For each device you define a separate line in the table.	
	MOUNTDEVICE	■ Connecting via NFS: Device name to be mounted: This is formed from the network address of the device, a colon, and the name of the directory. Enter the network address as four decimal numbers separated by periods (dotted-decimal notation). Pay attention to capitalization when entering the path name. ■ Connecting individual Windows computers: Enter the network name and share name of the computer, e.g. **//PC1791NT/C**
	MOUNTPOINT	Device name: The device name entered here is displayed at the TNC in the program management for the mounted network, e.g. **WORLD:** The name must end with a colon.
	FILESYSTEMTYPE	File system type: ■ **nfs:** Network File System ■ **smb:** Windows network
	OPTIONS (for **FILESYSTEMTYPE = nfs**)	Options that concern the file system type **nfs:** Options are entered without spaces, separated only by commas. Pay attention to capitalization. Options: **rsize:** Packet size in bytes for data reception. Input range 512 to 8192 **wsize:** Packet size in bytes for data transmission. Input range 512 to 8192 **timeo:** Time in tenths of a second after which the iTNC repeats a Remote Procedure Call not answered by the server. Input range 0 to 100000. If there is no entry, the standard value 7 is used. Use higher values only if the iTNC must communicate with the server over more than one router. Your network specialist determines this value. **soft:** The Remote Procedure Call is repeated until the NFS server answers. If **soft** is entered, it is not repeated.
	OPTIONS (for **FILESYSTEMTYPE = smb**)	Options that concern the file system type **smb:** Options are entered without spaces, separated only by commas. Pay attention to capitalization. Options: **ip=**: IP address of the Windows PC to which the iTNC is to be connected **username:** User name with which the iTNC should log in **workgroup:** Workgroup under which the iTNC should log in **password**: Password with which the iTNC should log in (up to 80 characters)
	AM	Auto mount (yes = **1**, no = **0**): Here you define whether during power-on the iTNC automatically mounts the network. Devices that are not automatically mounted can be mounted anytime in the program management.

<table>
<tr><th>Soft key</th><th>Option</th><th>Meaning</th></tr>
<tr><td>PING</td><td colspan="2">If a ping is sent, the receiver sends it back to the sender. Thus a ping can be used to check whether a connection to a particular remote station is possible. The address is entered as four decimal numbers separated by points (dotted-decimal notation).
After the ping has been sent, one of the following messages appears:
HOST RESPOND: Data package was received again
TIMEOUT: Data packages were not sent back within a certain period of time
CAN NOT ROUTE TNC: TNC could not send data package to the receiver</td></tr>
<tr><td rowspan="4">DEFINE UID / GID</td><td>TNC USER ID</td><td>Definition of which user identification the end user uses to access files in the network. Your network specialist determines this value.</td></tr>
<tr><td>OEM USER ID</td><td>Definition of which user identification the machine tool builder uses to access files in the network. Your network specialist determines this value.</td></tr>
<tr><td>TNC GROUP ID</td><td>Definition of which group identification is used to access files in the network. The group identification is the same for end users and machine tool builders. Your network specialist determines this value.</td></tr>
<tr><td>UID for mount</td><td>Defines the user identification (UID) for the log-on procedure.
USER: The login is with the USER identification.
ROOT: The login is with the identification of the ROOT user, value = 0.</td></tr>
</table>

Note

You might be able to omit the entries **username**, **workgroup** and **password** in the column **OPTIONS** for Windows 95 and Windows 98 networks.

You can encoder the password defined under **OPTIONS** with the **ENCRYPT PASSWORD** soft key.

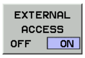
10.4 Protection Against Data Tampering

Due to the possibility of networking the iTNC 530 and accessing it remotely, protection from data tampering became necessary.

The following protection mechanisms are integrated:

- General disabling of data access by soft key
- Restricting access to the PLC partition
- Restricting access to parts of the TNC partition
  This function was added for the end user. Please see the notes in the User's Manual.

General disabling of data access

Soft key	Meaning
	This soft key disables access to the control using the LSV2 protocol, via both the serial and the Ethernet interfaces. It can be displayed in any operating mode with the MOD key.

The soft key is normally not shown. To display the soft key:

- Enter the keyword **`REMOTE.LOCKSOFTKEYVISIBLE = YES`** in OEM.SYS.

Restricting access to the PLC partition

In the standard setting, the PLC partition can be accessed via the LSV2 protocol using the code number 807667. To permit this access only with the code number defined in OEM.SYS under **`PLCPASSWORD =`** (no longer with 807667):

- Enter the keyword **`REMOTE.PLCPASSWORDNEEDED = YES`** in OEM.SYS.

Code number 807667 is used during machine backup and full backup with the LSV2 protocol to access the PLC partition. To permit this access only with the code number defined in OEM.SYS under **`PLCPASSWORD =`** (no longer with 807667):

- Enter the keyword **`REMOTE.PLCPASSWORDFORCED = YES`** in OEM.SYS.

10.5 The USB Interface of the iTNC (USB 1.1)

The USB interface is a standard serial interface.
(USB = Universal Serial Bus)

USB 1.1 provides a maximum data transfer rate of 12 Mbps.

Various USB devices, such as mouse, touchpad, external hard disks, and USB memory sticks, can be connected to the iTNC via the USB interface (X141, X142).

Note

If USB components require more than 0.5 A, a separate power supply becomes necessary for these components. One possibility is the USB hub from HEIDENHAIN (368 735-01).

The USB interface features "hot-plug capability." This means that you can connect USB devices to the USB interface and remove them, without having to shut down and then restart the control.

Transmission distance without hub: Up to 6 meters

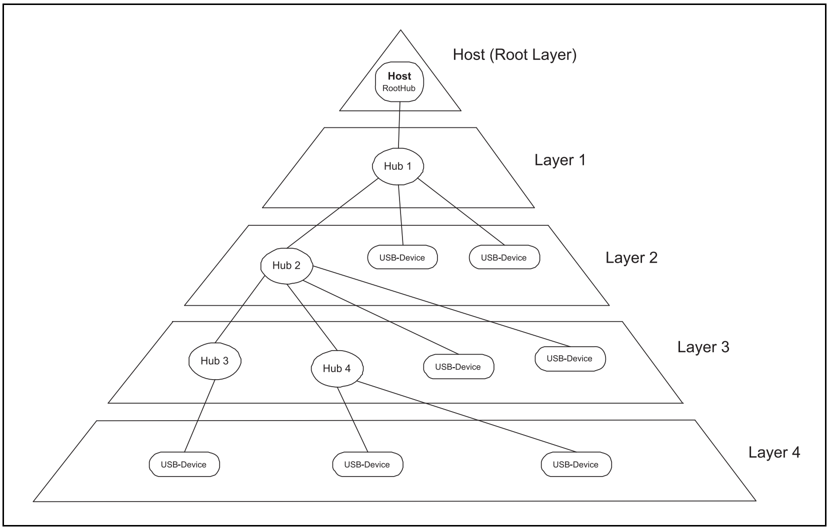
Note

For greater transmission distances, you must use a USB hub after every six meters in order to amplify the signal. You can use more than one hub for one transmission distance. USB cables with a length of up to 36 meters (with 6 integrated USB hubs) are available from HEIDENHAIN.

Design

The USB interface connects the USB peripheral devices with the USB host. The topology of a USB connection may consist of several levels arranged in a star configuration. Every level consists of a USB hub to which other USB devices or hubs are connected in a star configuration. A maximum of 127 USB devices can be connected to a USB host in this way.

Topology of the USB interface:



Functionality and signal designations

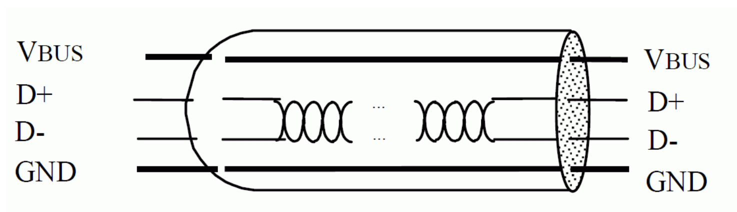
USB uses packet-based communication over two differential data lines. This reduces radiation and increases transmission reliability. USB provides significantly higher data transfer rates than the parallel / Centronics, and serial / RS-232, RS-422 external interfaces:

- USB full speed of up to 12 Mbps
- USB low speed of up to 1.5 Mbps

Conventional interfaces, such as the RS-232, are more suitable for time-critical applications because they are not based on packets which reduce the transfer rate (in case of packets with only a few bytes) or delay transmission (when collecting bytes for filling a packet).

Only four wires are required in a USB cable. Two for a power supply of 5 V (with max. 500 mA / 2.5 W) and two for data transmission.

Structure of the USB cable:



USB data carriers on the control (Upgrade function – FCL2)

The USB interface of the control allows for convenient and fast exchange of data. You can connect USB block devices, such as memory sticks, hard disks, CD-ROM drives, to your control via the USB interface without having to reboot the system. The data media can be accessed immediately after connection.

The control supports the following USB block devices:

- Floppy disk drives with FAT/VFAT file system or ISO9660
- Memory sticks with FAT/VFAT file system or ISO9660
- Hard disks with FAT/VFAT file system or ISO9660
- CD-ROM drives with FAT/VFAT file system or ISO9660

The control does not support USB devices with other file systems (e.g. NTFS). If you try to connect such devices, the control will issue an error message.

Note

- It should basically be possible to connect all USB block devices with the above-mentioned file systems to the control. If you nevertheless encounter problems, please contact HEIDENHAIN.
- Please note that the “Automatic Update” function on the control with USB data carriers is independent of the Feature Content Level.

Connecting and disconnecting USB data carriers (upgrade function)

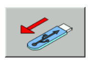
There are two soft keys in the iTNC's file manager for connecting and removing USB data carriers (hard disks, memory sticks, etc.):

Soft key	Meaning
	Connect a USB data carrier to the control
	Disconnect a USB data carrier from the control

These appear after the MORE FUNCTIONS soft key is pressed, if the USB device was selected in the file manager. If the USB device is not selected, then the iTNC only shows the soft key for connection.

Attention

In order to remove a USB data carrier, you must always press the MORE FUNCTIONS and soft keys.



Data can be lost if the USB data carrier is removed from the control before the appropriate soft key is pressed.

USB data carriers tested by HEIDENHAIN (upgrade function)

A variety of USB storage media from different manufacturers is available on the market. Numerous USB devices are supported by the control, but you should test them for proper functioning on the control before using them. It may happen in isolated cases that a USB device is not supported by the control.

Drive name and user group for USB device

As of software version 340 49x-04 you can define the drive name and the authorized user group for USB devices. This requires that **USBMOUNTS = <path\file>** is entered in OEM.SYS. This entry refers to a file containing one or more lines with the following syntax:
<Volume>,<Mountpoint:>,<User>

Example of entry in the file that is referred to, e.g. Stickconfig.a:
KINGSTON,PLCSTICK:,OEM

If the control detects a USB device during startup (only if it was switched on with the main switch), which has a VFAT partition with one of the specified volume names, the device is mounted and displayed with the name specified in <Mountpoint:>, and not with the name **USBn:**.
If the device also has other partitions, the other partitions are ignored except if they are also mounted with a volume name.
If a USB device with one of the indicated volume names is connected after the control has started up, the device is mounted with the name **USBn:** as usual.
The new functionality does not become effective until the control is switched off and then back on again.
The described function is not available on dual-processor controls with the MC 422B hardware. On dual-processor controls with the MC 422C, the new function only works for USB devices that are connected to a real-time processor (X143).
The following settings can be defined in any ASCII file (e.g. Stickconfig.a):

- <Volume>
  Volume name of a USB device formatted with VFAT. If the name contains blank spaces, the complete name must be enclosed between quotation marks (" "). Also, the name must always be given in capital letters.
- <Mountpoint>
  Drive name under which the USB device is to be mounted (e.g. PLCSTICK:). The last character of the name must be a colon. The name can have up to 8 characters (without colon).
- <User>
  User group for which the USB device is mounted and displayed. The following user groups are possible:
  - TNC:
    The USB device can be used by the machine operator.
  - OEM:
    The USB device can only be used by the OEM (code number 807667, PLC program, NC cycles).
  - SYS:
    The USB device can only be used by the HeROS system.

Connecting more than one USB device with the same volume name may cause problems. When the control is started up, the first detected USB device with the new drive name is mounted. As a result, it may occur that each time the control is started up, another device with the same volume name is mounted.

HEIDENHAIN therefore recommends using unique volume names that cannot be confused (e.g. "JHK247G4"). On a PC with Windows operating system, the volume name can be defined in the properties of the USB device.

10.6 iTNC Serial Data Interfaces

10.6.1 General information

The iTNC features one of each of the interfaces:

- RS-232-C/V.24 and
- RS-422/V.11

The two interfaces differ in the design of their hardware with regard to signal lines, signal levels and pin layout. The data format and transmission protocol are the same.

The two interfaces can be operated in parallel.

Three transmission protocols are available:

- Standard communications protocol
- Transmission protocol with Block Check Character (BCC)
- LSV2 transmission protocol

> **Note**
>
> No connection to the serial interface of the iTNC can be established if the file manager on the iTNC is open.

10.6.2 RS-232-C/V.24 interface

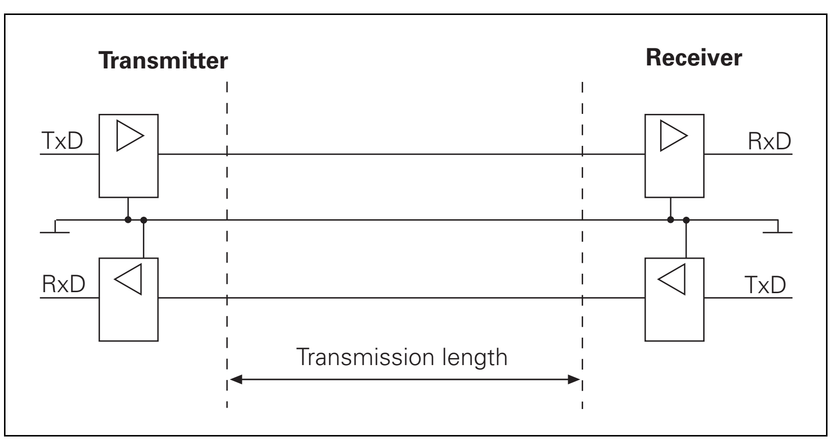
RS-232-C/V.24 is the designation for a serial interface. Data transfer is executed asynchronously, with a start bit before each character and one or two stop bits after each character.

Transmission distance: up to 20 m

Hardware

The physical connection between two RS-232-C/V.24 interfaces is an asymmetrical line, i.e. the common ground connection between transmitter and receiver is used as a return wire.

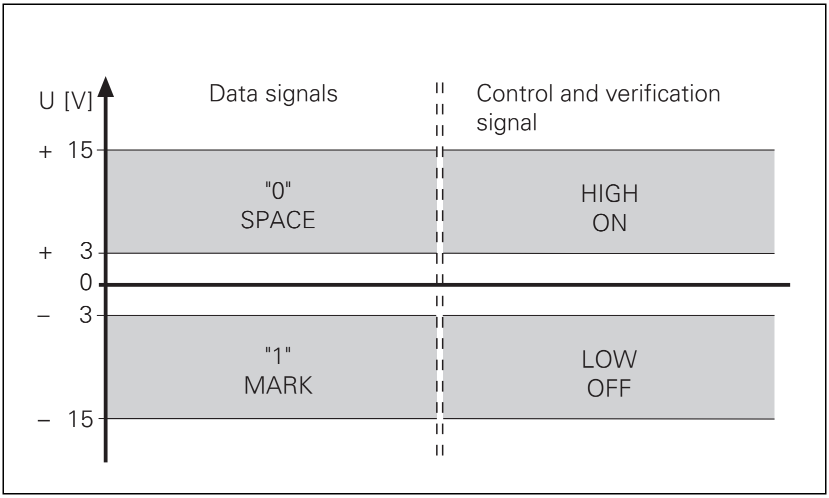
Physical connections:



Signal levels

The levels of the individual signal lines differ:

- Data lines: The data signals are defined as being logical zero (SPACE) over the range +3 V to +15 V and logical one (MARK) over the range –3 V to –15 V.
- Control and verification lines: These signals are defined as being ON (High) over the range +3 V to +15 V and as OFF (Low) over the range –3 V to –15 V.



Note

For all signals: The voltage range from –3 V to +3 V cannot be evaluated.

Signal designation

One must differentiate between the following types of lines and their signals:

- Data lines:
  - TxD Transmitted data
  - RxD Received data
- Control and signal lines:
  - DCD (Data Carrier Detect):
    Received signal level. The receiver signals that the information it has received lies within the defined level. The DCD signal is not used by the iTNC. The iTNC delivers no signal from this pin.
  - DTR (Data Terminal Ready):
    iTNC ready / not ready for service (e.g. the receiving buffer is full, the signal DTR indicates "LOW").
  - DSR (Data Set Ready):
    Peripheral device ready / not ready for service.
  - RTS (Request to Send):
    Switch transmission unit on. iTNC wishes to transmit data.
  - CTS (Clear to Send):
    Readiness for transmission. The peripheral wishes to transmit data.
- Ground conductors (lines for power supply):
  - Chassis GND:
    Housing connection
  - Signal GND:
    0 V lines for all signals

Pin layout

Keep in mind that there might be a difference between the pin layout of the MC 42x(B) and the adapter block.

10.6.3 RS-422/V.11 interface

RS-422/V.11 is a standard serial interface.

It is suitable for data transfer rates up to 10 Mbps.

The interface module of the iTNC can transmit data at up to 115 200 bps.

Transmission distance: over 1 kilometer

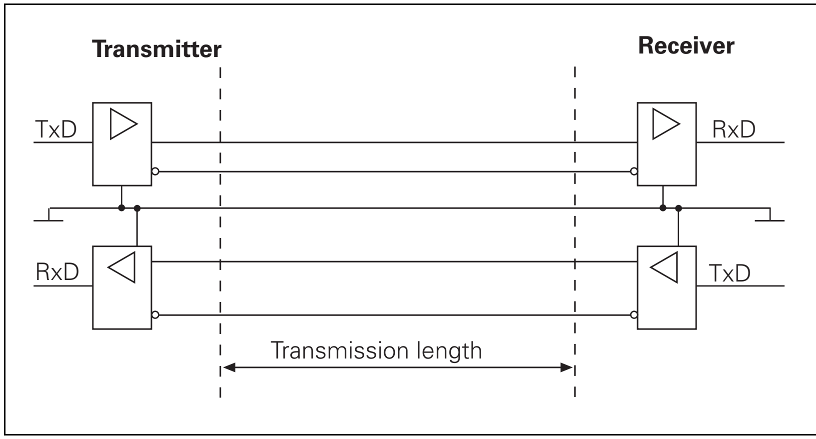
Hardware

The interface works symmetrically, using two signal lines. At the receiver, the difference in voltage of the two lines is evaluated.

Advantage:

- Longer lines can be used
- Higher data transfer rate

Physical connections:



Signal levels

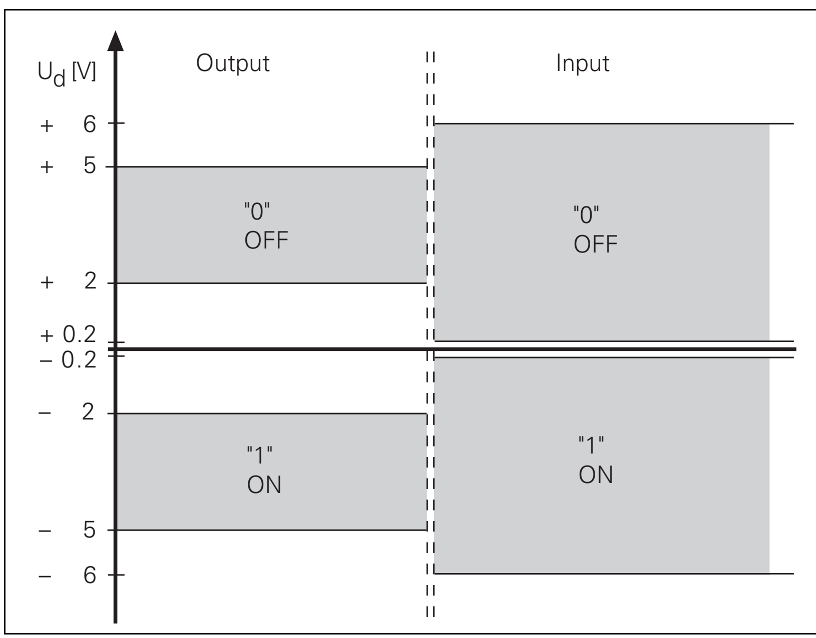
The signals are both transmitted and received as differential voltage.

A positive differential voltage corresponds to logical zero (OFF).

A negative differential voltage corresponds to logical one (ON).

V_{dmin} = 2 V and V_{dmax} = 5 V

The control unit detects the differential voltages between U_{dmin} = 0.2 and U_{dmax} = 6 V as a logically defined level.



Signal designation

The following signals are transmitted as differential signals:

Signals	Signal designation	
Data signals	TxD, $\overline{\text{TxD}}$	RxD, $\overline{\text{RxD}}$
Control and message signals	RTS	CTS
	DSR	DTR

The protective ground connects the transmitter and receiver housings.

GND is the differential voltage reference conductor.

These signals perform the same functions as those on the RS-232-C/V.24 interface.

Pin layout

The MC 42x(B) and adapter block have the same pin layout.

10.7 Configuration of Interfaces

10.7.1 Control characters

Overview of control characters specific to HEIDENHAIN

Character	Description	Description
SOH	Start of Header	Identifies the beginning of the data transfer header. The character string contains the program number and information about the type of program and the transfer mode.
STX	Start of Text	Identifies the beginning of a program block.
ETB	End of Text Block	Terminates a data transfer block. The character that follows (BCC) is used for data checking.
DC1	XON	Starts the transfer of data.
DC3	XOFF	Stops the transfer of data.
ETX	End of Text	Transmitted at the end of a program.
EOT	End of Transmission	Terminates the data transfer and establishes the idle state. This character is transmitted by the iTNC at the end of a program input and to the external device in the event of an error.
ACK	Acknowledgment	Transmitted by the receiver when a data block has been transferred without error.
NAK	Negative Acknowledgment	Transmitted by the receiver when a data block has been transferred with an error. The transmitter must re-transmit the data block.

10.7.2 Selection of interfaces and operating modes

To disable or enable either of the RS-232-C/V.24 and RS-422/V.11 interfaces:

- Select MP5000.

If at least one of the interfaces is enabled, you can select the following settings:

- Call the MOD functions.
- Choose the desired operating mode from the table.

For the following external devices	Choose the operating mode
HEIDENHAIN floppy disk unit: ■ FE 401 B ■ FE 401 from program no. 230 626-03	FE1
■ HEIDENHAIN FE 401 floppy disk unit up to program no. 230 626-02 ■ PC with HEIDENHAIN TNC.EXE data transfer software	FE2
PC with HEIDENHAIN software TNCremo	LSV2
Non-HEIDENHAIN devices such as printer, punch or PC with other data transfer software	EXT1 and EXT2

MP5000 **Disable data interfaces**
Input: 0: No interface disabled
1: RS-232-C/V.24 interface disabled
2: RS-422/V.11 interface disabled
3: RS-232-C/V.24 and RS-422/V.11 interfaces disabled

Communication between iTNCs

- Set both iTNCs to LSV2 protocol. The control from which you start the data transmission is the master.

The PLC provides you with access to the data interfaces (EXT3/EXT4).

10.7.3 Configuration of interfaces

To configure data format and the type of handshake in the EXT1/EXT2/EXT3/EXT4 operating modes (EXT3/EXT4 only for the PLC):

▶ Select machine parameter MP5020.x.

Data bits

With **bit 0** you determine whether transmission is to be with seven or eight data bits. Transmission with seven bits is normally used; for printer interfacing eight bits are required.

Block Check Character (BCC)

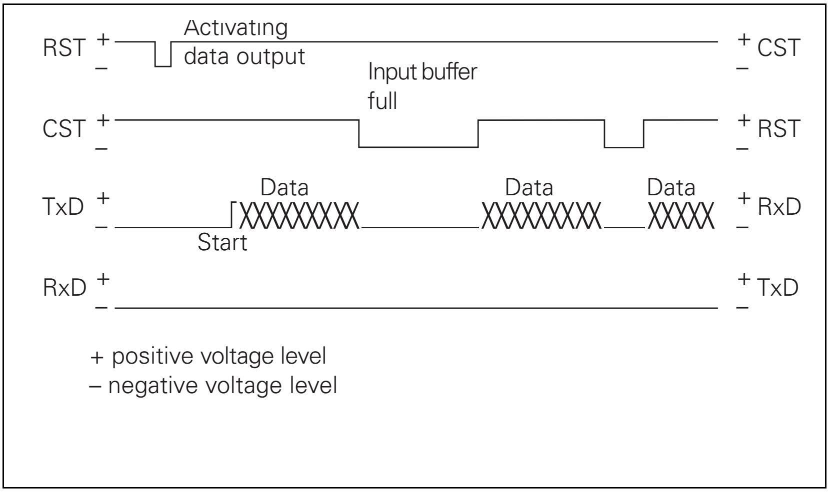
With **bit 1** you can ensure that the BCC is not interpreted as a control character.

On the iTNC, numbers less than $20 are defined as control characters. If calculation of the BCC produces a number less than $20, then a blank space is sent in addition immediately before <ETB>. The BCC will consequently always be greater than $20 and cannot therefore be interpreted as a control character.

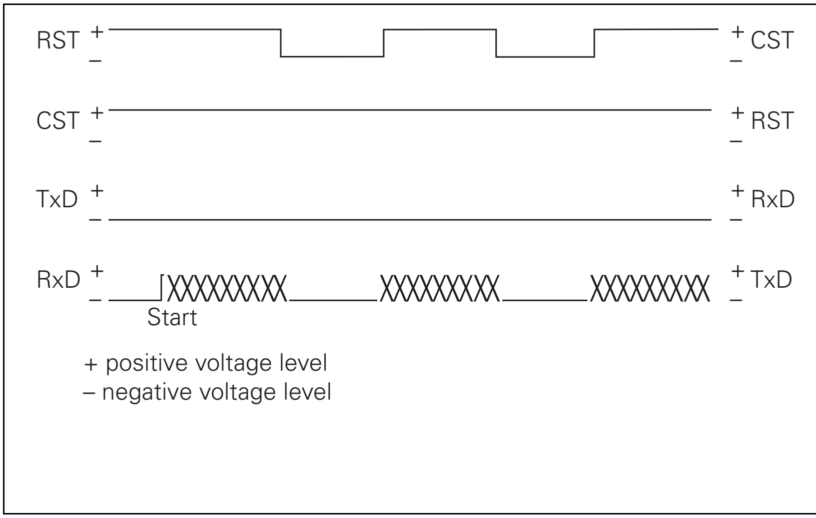
Hardware handshaking

Bit 2 can be set to determine whether the iTNC stops transfer from an external device by sending an RTS signal.

- Data output from iTNC to EXT
  When the receiving buffer is full, the external device resets the RTS signal. The iTNC detects that the peripheral unit receiving buffer is full at its CTS input:



- Data input from EXT to iTNC
  When the receiving buffer is full, the iTNC removes the RTS signal. This is detected by the peripheral device at its CTS input:



The DTR and DSR signals indicate the operational status of the iTNC and peripheral device:

- DTR: Interrogated by peripheral; it is logical one if iTNC is ready for service.
- DSR: Interrogated by iTNC.
  - HIGH level means: external data input/output ready.
  - LOW level means: external data input/output not ready.

Software handshaking

With **bit 3** you determine whether the iTNC stops transfer from an external device with control character <DC3>. Transfer is then resumed with character <DC1>. (XON/XOFF method)

If transfer is stopped with the control character <DC3>, up to three more characters can be stored; any further incoming characters are lost. Software handshaking is normally recommended when interfaces are connected to an external device.

Note

The iTNC reacts both to hardware and software handshakes, regardless of the setting in MP5020.x.

If no transmission stop is set in MP5020.x, the iTNC stops the peripheral unit with the software handshake.

If a transmission stop by RTS and by DC3 is active simultaneously, the iTNC stops transfer with the hardware handshake.

Character parity

Bits 4 and **5** determine the type of parity check.

Stop bits

Bits 6 and **7** determine the number of stop bits sent at the end of a character.

Testing the connection

If PLC Modules 910x transmit characters via the RS 232-/RS422 interface, the control no longer checks if a connection is available if **bit 10** is set to 1 in MP5020.x.
Bit 8 and **bit 9** in MP5020.x are reserved and currently have no function.

MP5020	**Configuration of the data interface**
Format:	%xxxxxxxx
Input:	Bit 0: 0 = 7 data bits, 1 = 8 data bits
	Bit 1: 0 = any BCC, 1 = BCC not control character
	Bit 2: 0 = transmission stop by RTS not active, 1 = active
	Bit 3: 0 = transmission stop by DC3 not active, 1 = active
	Bit 4: 0 = character parity even, 1 = odd
	Bit 5: 0 = character parity not desired, 1 = desired
	Bit 6 = 0, Bit 7 = 0: Length of the stop: 1.5 bits
	Bit 6 = 1, Bit 7 = 0: Length of the stop: 2 bits
	Bit 6 = 0, Bit 7 = 1: Length of the stop: 1 bit
	Bit 6 = 1, Bit 7 = 1: Length of the stop: 1 bit
	Bit 8 = Reserved
	Bit 9 = Reserved
	Bit 10: 0 = Check for missing connection active
	Bit 10: 1 = Check for missing connection not active
MP5020.0	Operating mode EXT1
MP5020.1	Operating mode EXT2
MP5020.2	Operating mode EXT3 (PLC)
MP5020.3	Operating mode EXT4 (PLC)

Communication protocol

The communication protocol for operating modes EXT1/EXT2/EXT3/EXT4 is defined with MP5030:

MP5030	**Communications protocol**
Input:	0 = standard data transfer protocol 1 = blockwise transfer 2 = without protocol (only for MP5030.2)
MP5030.0	Operating mode EXT1
MP5030.1	Operating mode EXT2
MP5030.2	Operating mode EXT3 (PLC)
MP5030.3	Operating mode EXT4 (PLC)

Example

The printer NEC P7 PLUS is to be configured with operating mode EXT1.

The parameters listed are preset by the printer (see the operating manual of the printer concerned):

- Serial interface
- 8 data bits
- Even character parity
- XON/XOFF protocol (software handshake)
- 9600 bps

The following settings are made at the iTNC:

- Select MP5000 = 0.
- Select MP5020.0 = %10101001.
- Select MP5030.0 = 0.

Machine parameters	Effect
MP5000 = 0	No interface inhibited
MP5020.0 = %10101001	Bit 0: 8 data bits Bit 1: any BCC character Bit 2: transmission stop by RTS not active Bit 3: transmission stop by DC3 active Bit 4: character parity even Bit 5: character parity desired Bit 6/7: 1 stop bit
MP5030.0 = 0	Standard data transfer

- Call the MOD functions.
- Choose the EXT1 operating mode.
- Set the baud rate for EXT1 to 9600 bps.

10.8 Data Transmission Protocols

10.8.1 Selection of transmission protocols

The operating modes are assigned the following transmission protocols:

Operating modes	Communications protocol
FE1	Select a protocol with BCC and with fixed control characters 1 start bit, 7 data bits, 1 stop bit
EXT1, EXT2, EXT3, EXT4	Select data format and transmission protocol using machine parameters
LSV2	Start this protocol from a PC or from the iTNC. The protocol runs in the background of the iTNC.

The following applies to all data transmission protocols except LSV2:

If an incoming file is already stored in the iTNC, the TNC will ask you whether you really wish to overwrite this file:

- Press a soft key to continue the transmission.

If you attempt to overwrite a write-protected file, the iTNC displays the **`Protected file!`** error message:

- Press the MORE FUNCTIONS soft key and then the UNPROTECT soft key to cancel write-protection and continue transmission.

If a file has been read out and the data transfer menu has been terminated with the END key, the iTNC outputs the characters <EXT> and <EOT>.

If a transmission is terminated with the END key, the error message **`Program incomplete`** is issued.

10.8.2 Standard communications protocol

General information

To set the standard data communication protocol in the operating modes EXT1/EXT2/EXT3/EXT4:

- Select MP5030.x = 0.

When outputting a file, the <NUL> character is sent exactly 50 times at the start of file. When reading in, however, the control unit ignores this character, regardless of how often the peripheral sends the <NUL> character before the file.

The program blocks are not checked for correctness but are transmitted one after the other.

If you wish to signal an error to the iTNC in the standard data transmission protocol, you must send the following sequence of instructions:
<ESC><1><Error number>

If the receiver's data buffer is full, the transmission can be stopped and resumed in one of two ways:

- Software handshaking
  - Stop transfer by sending the character <DC3> (XOFF)
  - Continue by transmitting the character <DC1> (XON)
- Hardware handshaking
  - By suitable levels on the control and message lines RTS and CTS of interfaces RS-232-C/V.24 or RS-422/V.11

Twelve characters before the receiving buffer is full, the iTNC transmits the character <DC3> to the transmitter in order to terminate transmission.

Example: Protocol for conversational NC program

<NUL><NUL><NUL><NUL> 50 times

0 BEGIN PGM 1 MM<CR><LF> 1st program block

1 TOOL DEF 1 L+0 R+3<CR><LF> 2nd program block

26 END PGM 1 MM<CR><LF> Program end

... ...

<ETX><EOT> Close the data transmission menu

Example of software handshake

iTNC to peripheral	Peripheral to iTNC
12 Z + 2 FMAX<CR><LF>	Receiving buffer full: <DC3> Receiving buffer ready again: <DC1>
13 Z -10 FMAX<CR><LF>	...

Output selected file

The EXT1 operating mode is set with software handshake.

The iTNC outputs all of the program lines in order.

The peripheral unit can:

- Stop transmission with <DC3>
- Resume transmission with <DC1>

iTNC to peripheral	Peripheral to iTNC
<NUL> <NUL> <NUL>... 1. line of file <CR> <LF>... 5. line of file <CR> <LF>	 Transmission stop: <DC3> Resume transmission: <DC1>
6. line of file <CR> <LF>... Last line of file <CR> <LF>	...

Load selected file

The EXT1 operating mode is set with software handshake.

To transfer a file from a peripheral unit:

- Enter the file name in the iTNC.

The iTNC can:

- Stop transmission with <DC3>
- Resume transmission with <DC1>

iTNC to peripheral	Peripheral to iTNC
100.H “START” <DC1>	 <NUL><NUL> 1. line PGM100<CR><LF> ... Last line PGM 100 <CR><LF><ETX>
<EOT>	

If the file name in the first line and the name indicated in the iTNC are not identical, the iTNC reads each block in and searches for the correct file name. If the END PGM block has been read in, and the selected name is not known, the iTNC remains static without an error message:

- In this case, terminate transfer with the END key.

10.8.3 Communication protocol with block check character

This protocol is specific to HEIDENHAIN and operates with its own control characters and an additional data check feature when transmitting.

The protocol is set with the following operating modes:

- FE1 mode
- EXT1/EXT2/EXT3/EXT4 mode if MP5030.x = 1

The data transfer protocol is identical for all these modes except for the FE1 mode. In the FE1 mode, a command sequence is output at the beginning to request the contents directory from the peripheral unit. For the FE1 mode, bit 1 of MP5020.x must be set to 1.

Header

When a file is transferred, the first block — called the header — consists of the following characters:
<SOH><K><Name><M><ETB><BCC><DC1>

Character	Meaning
<SOH>	Identifies the beginning of the header
<K>	File code
<Name>	File name
<M>	Data transfer mode (E = input, A = output)
<ETB>	Identifies the end of the header
<BCC>	Block Check Character
<DC1>	XON

Block Check Character (BCC)

In addition to checking the parity of the individual characters, the parity of the complete transferred block is also checked. The BCC always rounds the individual bits of the transferred characters in a data transfer block to even parity.

Example of BCC generation:

In this example, program 15, which has been written in HEIDENHAIN plain-language text (H), is input through the data interface (E).

Character	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SOH	0	0	0	0	0	0	1
H	1	0	0	1	0	0	0
1	0	1	1	0	0	0	1
5	0	1	1	0	1	0	1
E	1	0	0	0	1	0	1
ETB	0	0	1	0	1	1	1
BCC	**0**	**0**	**1**	**1**	**1**	**1**	**1**

A parity bit is also generated for the BCC. With even parity, the parity bit in this example is assigned the value 1.

At the end of every block, the receiver checks whether it has been transferred correctly.

To do this, the receiver computes a BCC from the received block and compares it with the received BCC. If the received BCC and the computed BCC are identical, the receiver transmits the character <ACK> for positive acknowledgment. If the two BCCs are not identical, the data block was not transmitted correctly. The receiver transmits the character <NAK> for negative acknowledgment. The block must be re-transmitted. This process is repeated up to 15 times, then the error message **`Transferred data incorrect E`** is output. The transmission is aborted.

If the header is acknowledged with <ACK>, the first file block can be transmitted:

The beginning of a file block is identified by the control character <STX>. The remaining control characters in this block are identical with the control characters in the header. If this block is acknowledged by <ACK>, then the next program block is transmitted. If <NAK> is transmitted, the same block has to be re-transmitted, etc. Once the last program block has been acknowledged by <ACK>, the transmission is terminated by the characters <ETX> (end of text) and <EOT> (end of transmission).

Handshaking

The character <DC1> (XON) follows the BCC. This character is required by many devices to explicitly request the transmission once again from the transmitter.

The <DC1> character is not required for reading in a file in the BCC format.

The transmitter waits and only resumes data transmission when the receiver has transmitted a positive (ACK) or negative (NAK) acknowledgment to indicate that the receiving buffer is ready.

To disable transmission of the <DC1> character in the EXT1, EXT2, EXT3 and EXT4 modes:

- Set MP5020.x bit 3 = 0.

Example:

To read out a pallet file with the name PPP to a peripheral device (e.g. FE 401).

iTNC to peripheral	Peripheral to iTNC
<SOH><L>PPP<A><ETB>BCC	<ACK>
<STX> “1st line”<ETB>BCC	<ACK>
...	...
<STX> “10th line”<ETB>BCC	<NAK>
<STX> “10th line”<ETB>BCC	<ACK>
<STX> “11th line”<ETB>BCC	<ACK>
...	...
<STX>“last line”<ETB>BCC	<ACK>
<ETX><EOT>	

Report error to the iTNC

FE1 mode is set.

If an error occurs at a peripheral device, the following block must be sent to the iTNC:

<SOH><Error text><ETB>BCC

Peripheral to iTNC	iTNC to peripheral
<SOH> “Error”<ETB>BCC	<ACK><EOT>

The received error message is displayed on the iTNC. To continue

- Press the CE key.

Request external directory

FE1 mode is set.

This protocol is not available in the EXT mode. In FE1 mode the following ‘Escape’ sequence is sent to request the external directory:

<DC3><ESC><DC1><0><SP><D><CR><LF>

The iTNC expects the following input to this request:

xxxxxx<code letter><sectors><name><P[1)]><CR><LF>
[1)] P = Protected (optional)

The first four lines, each ending in <CR><LF>, are ignored. In subsequent lines ending with <CR><LF>, the program name and, after any number of blank characters, the number of sectors are stored.

If the character combination <FREE:> is detected, only a number — the number of free sectors — will be read in.

The iTNC requests the complete directory. The directory is saved and the files of the selected type are displayed.

The peripheral device ends transmission with <EXT>. The iTNC sends an <EOT>.

Output selected file

iTNC to peripheral	Peripheral to iTNC
<SOH><K>Name<A><ETB>BCC	<SOH><K>Name<A><ETB>BCC
<STX> “1st line”<ETB>BCC	<ACK>
...	...
<STX> “last line”<ETB>BCC <DC1>	<ACK>
<ETX><EOT>	

Output marked files

Marked files are output in the same protocol as for outputting the selected files. After each file, the control characters <EXT><EOT> are sent to the peripheral device.

Load selected file

To read in a file from an external memory, the iTNC sends a header with the file name.

iTNC to peripheral	Peripheral to iTNC
<SOH><K>Name<E><ETB>BCC <DC1>	<ACK> <STX>"1st line"<ETB>BCC<DC1>
<ACK>	...
...	<STX> "last line"<ETB>BCC <DC1>
<ACK>	<ETX>
<EOT>	

10.8.4 LSV2 transmission protocol

The LSV2 protocol is a data transfer protocol for the two-way transfer of commands and data.

The data is transferred in blocks — so-called telegrams — into which the data is split up.

The following functions are possible:

- Data transfer
- File management, such as deleting, copying and renaming files
- Changing, creating and deleting paths
- Remote operation of the control functions. The TNC screen appears on the computer monitor. All functions can be executed from the computer.
- Real DNC operation. Starting and stopping the machine from the PC.
- Diagnosis of iTNC error messages and keystrokes for service purposes. The last 1000 events are stored in the iTNC.
- An LSV2 command can be used to disable/re-enable write-access on tables exclusively for HEIDENHAIN-DNC.
- The selected NC programs can be stopped and unselected in the **Program Run Single Block** and **Full Sequence** operating modes.

HEIDENHAIN offers two LSV2 software packages. Please contact HEIDENHAIN for further information.

Timeouts

You can define your own times for timeouts in the system file OEM.SYS:

- LSV2TIME0 = Timeout for receiving block STX to ETX (default 3 s)
- LSV2TIME1 = Timeout for acknowledging ENQ or check sum (default 3 s)
- LSV2TIME2 = Timeout when sending DLE 0, DLE 1 or NAK until a valid character is received (default 1 s)

Input range: 0.001 to 3.6 s

Note

If the code words are not defined or if the input range is exceeded, the default values are used.

10.9 Saving and Loading Files

The table lists all the files that can be saved to external memory devices and loaded from them.

File	File extension	Identifier
NC program in HEIDENHAIN plain-language	.H	H
NC program in DIN/ISO language	.I	D
Tool table	.T	T
Pallet table	.P	L
Datum table	.D	N
Machine parameters	.MP	M
Compensation table	.COM	V
Compensation assignment	.CMA	S
PLC program	.PLC	P
Text file	.A	A
Pocket table	.TCH	R
Help files	.HLP	J
Point table	.PNT	U
PLC error table	.PET	F
System file	.SYS	O
Cutting-data table	.CDT	–
Freely definable tables	.TAB	–
Motor table (asynchronous motors)	.ASN	–
Motor table (synchronous motors)	.SN	–
Motor table (servo amplifiers)	.AMP	–
Error file	.JOU	–
OEM cycles	.CYC .DES .PIC .ELE	–
Oscilloscope recordings	.DTA	–
Network settings	.N00 .M00 .P00	–

To write to or read from machine parameter files, compensation tables or PLC files, you must enter the correct code numbers with the MOD function:

You can output the current values of Q parameters, PLC error messages and dialogs over the RS-422/V.11 and RS-232-C/V.24 interfaces. (NC program: **FN 15: PRINT**).

During transmission with a Block Check Character (BCC), each device outputs and receives the correct file code.

Example:

If a pallet table is stored, it is given the file extension *.LNC.

For data transmission with the **TNCremoNT** PC software from HEIDENHAIN the identifier has no significance. The files are saved on the PC with the same extension as on the iTNC.

Note

Files that have no code (-) can only be transmitted with the LSV2 protocol of TNCremoNT.

10.10 Data Transfer by PLC

10.10.1 Settings

PLC modules make it possible for the PLC to transfer data via the RS-232-C/V.24 or RS-422/V.11 data interfaces. These modules, for example, permit communication between two MC 42x(B) at PLC level via the interface.

During data transfer, use of the interface is inhibited for the input/output program of the user interface.

- Select a standard operating mode, FE1 or
- Configure the data interface with MP5020.x to MP5040.x in EXT3/EXT4 mode.

MP5040	**Data transfer rate in operating mode EXT3 or EXT4 (data transfer through PLC)**
Input:	0: 110 bps 1: 150 bps 2: 300 bps 3: 600 bps 4: 1200 bps 5: 2400 bps 6: 4800 bps 7: 9600 bps 8: 19200 bps 9: 38400 bps 10: 57600 bps 11: 115200 bps
MP5040.0	Operating mode EXT3 (PLC)
MP5040.1	Operating mode EXT4 (PLC)

10.10.2 PLC modules

With the following PLC modules you can control the data interfaces from the PLC:

- Modules 9100 and 9101: Assign/release the data interfaces
- Module 9102: Interrogate the status of the interface
- Modules 9103 and 9104: Transmit and receive a string from the string memory. The transmit and receive buffers for the PLC are 128 characters long. Since every STRING ends with an END character, a STRING can only be up to 127 characters long.
- Modules 9105 and 9106: Transfer a block of binary values (bytes) from the word memory
- Module 9107: Read bytes from the receiving buffer without erasing the buffer

STRINGS and binary data are transferred using ASCII characters.

Example: Transferring a binary block

Address	Value	ASCII character
.	.	.
B126	11111010	$FA
.	10000001	$81
.	.	
.	.	
.	.	

When transferring binary data starting from the address B126, the ASCII characters <F> <A> <8> <1> etc. are transmitted in sequence from the word memory through the interface. Each byte contains two ASCII characters. The transmitting and receiving buffers each hold 63 bytes.

Module 9100 Assign data interface

With Module 9100 you assign an interface to the PLC and specify the transfer parameters. They initialize the interface, thereby erasing any errors that may have occurred. The interface is then ready to receive.

Once assigned to the PLC, the interface is disabled for use by the input/output program of the user interface. The assignment is canceled when the PLC program is recompiled.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	B/W/D/K	<Transfer parameters> 0: From MP50x0.2 1: From MOD function 2: From MP50x0.3
CM	9100	

Error recognition:

Marker	Value	Meaning
M4203	0	Interface was assigned
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect transfer parameter
	13	No connection
	14	Interface busy or input/output not ready
	17	Incorrect data transfer rate
	20	Module was not called in a spawn job or submit job

Module 9101 Release data interface

Module 9101 cancels the assignment of an interface to the PLC. The receive mode of the interface is canceled.

Can only be called in a submit job or spawn job!

Call:

```
PS      B/D/W/K   <Interface>
                  0: RS232
                  1: RS422
CM      9101
```

Error recognition:

Marker	Value	Meaning
M4203	0	Interface enabled
	1	Error code in W1022
W1022	1	Incorrect interface
	14	Interface not assigned
	20	Module was not called in a spawn or submit job

Module 9102 Status of data interface

Module 9102 reads the status information about an interface in bit-coded form.

The information “interface ready” is updated when the interface is assigned to the PLC or NC. If the interface is not assigned, the module reads the last valid status.

Call:

```
PS      B/W/D/K   <Interface>
                  0: RS232
                  1: RS422
CM      9102
PL      B/W/D     <Interface status>
                  –1: Error code in W1022
                  Bit 0: Interface is assigned
                  Bit 1: Interface is assigned to the PLC
                  Bit 2: Interface is ready
                  Bit 3: Transmit buffer is empty
                  Bit 4: Error during transmission
                  Bit 5: Receive buffer is full
                  Bit 6: Error in reception
                  Bit 7: ETX was received (not ready to receive)
                  Bit 8: Internal buffer from Module 9113 contains characters
```

Error recognition:

Marker	Value	Meaning
M4203	0	Status read
	1	Error code in W1022
W1022	1	Incorrect interface

Module 9103 Transmit string through data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9103 transmits a string from a string memory through one of the two interfaces. Links to the PLC error file and PLC dialog file are deleted.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	K/B/W/D	<Number of source string in the string buffer>
CM	9103	

Error recognition:

Marker	Value	Meaning
M4203	0	String sent
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect string number
	12	No string end found
	13	Interface not ready
	14	Interface not assigned
	15	Transmit buffer not empty
	20	Module was not called in a spawn job or submit job

Module 9104 Receive string through data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9104 reads a string from the receive buffer of a serial interface in a string memory and resets the receive buffer.

Can only be called in a submit job or spawn job!

Call:
```
PS      B/W/D/K    <Interface>
                   0: RS232
                   1: RS422
PS      K/B/W/D    <Number of the string in the string buffer>
CM      9104
```

Error recognition:

Marker	Value	Meaning
M4203	0	String was received
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect string number
	12	String too long
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9105 Transmit binary data through data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9105 transmits a block of binary values from the word memory of the PLC to one of the two interfaces. The transfer is in the form of ASCII-coded hexadecimal values. Every byte in the source block makes two ASCII characters at the interface.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	K/B/W/D	<Number of the first byte in the binary block>
PS	K/B/W/D	<Length of the binary block (0 to 63)>
CM	9105	

Error recognition:

Marker	Value	Meaning
M4203	0	Data was transmitted
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect byte number or block too long
	4	Block outside value range
	13	Interface not ready or no connection
	14	Interface not assigned
	15	Transmit buffer not empty
	20	Module was not called in a spawn or submit job

Module 9106 Receive binary data through data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9106 reads a block of binary values from one of the two interfaces into the word memory of the PLC. The transfer is in the form of ASCII-coded hexadecimal values. Every two ASCII characters from the serial interface make one byte in the binary block.

The length of the read binary block is returned as the initial variable.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	K/B/W/D	<Number of the first byte in the binary block>
CM	9106	
PL	B/W/D	<Length of binary block in bytes> –1: Incorrect module call

Error recognition:

Marker	Value	Meaning
M4203	0	Data was received
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect byte number or block too long
	4	Block outside value range
	11	Odd number of characters or illegal character
	12	String too long
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9107 Read from receiving buffer

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9107 reads two ASCII characters from the receiving buffer to one of the two interfaces and codes them to a binary value.

You can specify an offset that corresponds to the position of the byte to be read in a binary block read by Module 9106. The contents of the receiving buffer are retained and can be read by Modules 9104 and 9106.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	B/W/D/K	<Offset of byte to be read in binary block>
CM	9107	
PL	B/W/D	<Binary value read>

Error recognition:

Marker	Value	Meaning
M4203	0	Receiving buffer was read
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect byte number
	11	Illegal character
	12	String too long or offset too large
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9110 Transmit a message via LSV2

Module 9110 transmits a message (binary data or string) to a host computer connected by LSV2 protocol.

The message is transmitted to the host by the LSV2 command "M_PC<msg.l>".

Call:

PS	B/W/D/K	<Data type> 0: Binary data double word 1: String
PS	B/W/D/K	<Source address> With binary: Number of the double word With string: Number of the string
CM	9110	
PL	B/W/D	<Error code> 0: Message is being transmitted 1: No connection to host 2: Transmit buffer full 3: Incorrect data type (not 0 or 1) 4: Incorrect source address

Error recognition:

Marker	Value	Meaning
M4203	0	Message was transmitted
	1	Error code in W1022
W1022	2	Incorrect data type
	4	No double word address, or incorrect string number
	11	String too long
	13	No connection
	15	Transmit buffer not empty
	16	Receiving buffer empty

Module 9111 Receive a message via LSV2

Module 9111 reads a message (double word or string) that has been received from a host computer connected by LSV2 protocol.

The message must be transmitted from the host by the LSV2 command "M_PC<msg.l>".

Call:

PS	B/W/D/K	<Data type> 0: Binary data double word 1: String
PS	B/W/D/K	<Target address> With binary: Number of the double word With string: Number of the string
CM	9111	
PL	B/W/D	<Error code> 0: Message was read 1: No connection to host 2: No message of this type in receiving buffer 3: Incorrect data type (not 0 or 1) 4: Incorrect target address

Error recognition:

Marker	Value	Meaning
M4203	0	Message received
	1	Error code in W1022
W1022	2	Incorrect data type
	4	No double word address, or incorrect string number
	11	String too long
	13	No connection
	15	Transmit buffer not empty
	16	Receiving buffer empty

Module 9112 Transmit ASCII characters via data interface

You must first assign the interface to the PLC and initialize it with Module 9100. Module 9112 transmits a single ASCII character.

Note

Set MP5030.x = 2 so that the transmitted characters do not disturb the set protocol procedure.

Define the characters in at least one word so that the values to 255 can be recognized.

Can only be called in a submit job or spawn job!

Call:

```
PS    B/W/D/K   <Interface>
                0: RS232
                1: RS422
PS    W/D/K     <ASCII code [0 to 255]>
CM    9112
```

Error recognition:

Marker	**Value**	**Meaning**
M4203	0	Character was transmitted
	1	Error code in W1022
W1022	1	Incorrect interface
	13	Interface not ready or no connection
	14	Interface not assigned
	15	Transmit buffer not empty
	20	Module was not called in a spawn job or submit job

Module 9113 Receive ASCII characters via data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9113 reads a single ASCII character from the receiving buffer of a serial interface and resets the receiving buffer.

If there is more than one character in the receiving buffer, the first is returned and the others are stored in a special buffer.

You can interrogate the current state with Module 9102, bit 8.

As long as data remains in the buffer, no further characters are collected from the interface.

If MP5030.x < 2, the characters cannot be read from the interface until the line with the character requested in the protocol has been executed.

Note

Store the result in at least one word so that the values to 255 will be recognized.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
CM	9113	
PL	W/D	<ASCII character read> [0 to 255] = ASCII characters; [–1] = error

Error recognition:

Marker	Value	Meaning
M4203	0	Character was received
	1	Error code in W1022
W1022	1	Incorrect interface
	12	String too long
	13	Interface not ready or no connection
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job
	37	Receiving queue is full

10.11 External Programming

Please remember the following when programming externally for subsequent transmission:

- At the program beginning and after every program block, <CR><LF> or <LF> must be programmed.
- After the End of Program block, <CR><LF> and also <EXT> must be programmed.
- For NC programs, the spaces can be omitted between the individual words.
- When reading-in DIN blocks, the asterisk character (*) is not required at the end of the block.
- Comments are separated from the NC block with a semicolon (;).
- Comments located before the program are not saved.
- With conversational programming, the block numbers are generated by the iTNC. They need not be programmed.

11 iTNC 530 with Windows 2000/Windows XP

11.1 General Information and Important Notes

> Note
>
> When the control is put into service, the end user accepts the Microsoft license conditions. which are printed in the User's Manual.

When operating an iTNC 530 with Windows 2000 or Windows XP, please note the following:

- HEIDENHAIN offers no support for the installation of non-HEIDENHAIN software and cannot guarantee the function of Windows applications.
- HEIDENHAIN is not liable for faulty hard disk contents caused by installing updates to non-HEIDENHAIN software or additional application software.
- If such changes in programs or data make service visits from HEIDENHAIN necessary, the service costs will be invoiced.
- The machine manufacturer or end customer are responsible for any changes to the preinstalled Windows system on a dual-processor control (this especially regards changes to system settings and the installation of updates or additional software). HEIDENHAIN does not guarantee that this will have no negative effects on the operation of the control software, and therefore on the quality of the parts produced.

> Note
>
> HEIDENHAIN recommends:
>
> Under no circumstances should you perform comprehensive operations on the Windows system while manufacturing parts! This could adversely affect the operation as a numerical control, and therefore the quality of the parts produced.
> Comprehensive operations are operations that require a considerable portion of operating-system resources (computing time, RAM, accesses to the hard disk, network traffic, etc.).

Only if the control is at standstill (emergency stop) is it safe to perform such operations on the Windows computer. If any changes are to be made to the Windows computer, especially the installation of updates or additional software, then a comprehensive inspection and/or intensive testing is necessary (see page 1865). Even the change or exchange of shared components (DLL, Registry settings, etc.), which frequently occurs with Windows, can lead to undesired impairments at completely unexpected locations.

The following guidelines, for example, can be derived from the recommendation above:

- No automatic updates (neither from Windows nor from other software)
  During the update itself as well as in subsequent operation, the changes performed can lead to impairments of the control software.
- No starting of additional software while booting
  This applies in particular to services such as the real-time scan components of virus scanners.
- No automatic connection of network drives
  Under Windows, connections to non-existent network drives can lead to an increased system load. Therefore network drives should only be connected when needed, instead of automatically.

Microsoft service packs and patches

> Attention
>
> Service packs and patches from Microsoft may only be installed if they have been **approved** by HEIDENHAIN!

HEIDENHAIN does not assume any liability for the compatibility of these service packs and patches with other installed Windows applications.

As of software version 340 49x-04 with Windows 2000

Hard disks supplied by HEIDENHAIN with Microsoft Windows 2000 and NC software 340 49x-04 contain the following patches from Microsoft: KB937143, KB938127, KB927779, KB917008, KB917344, KB918118, KB920213, KB920670, KB920683, KB920685, KB921398, KB921503, KB923191, KB923414, KB923980, KB924270, KB924667, KB925902, KB926122, KB926436, KB928843, KB930178, KB931784, KB935839, KB935840, KB935843, KB936021, KB938827, KB938829, KB925398

As of software version 340 49x-05 with Windows XP

Hard disks supplied by HEIDENHAIN with Microsoft Windows XP and NC software 340 49x-05 contain the following patches from Microsoft: KB951376, KB952954, KB943460, KB951698, KB946026, KB941569, KB948590, KB950749, KB938464, KB956390, KB943055, KB944338, KB946648, KB956803, KB933729, KB957095, KB936782, KB950974, KB954211, KB956841, KB950762, KB951066, KB943485, KB945553, KB951748, KB944653, KB937894, KB950760, KB958644

You will also find a current list of the Microsoft patches approved by HEIDENHAIN on the HEIDENHAIN homepage.

The following changes were made to the Microsoft Windows 2000 and Windows XP installations:

- Automatic Windows updates were deactivated.
- The energy saver for the screen was deactivated.
- The Microsoft Java Virtual Machine (MSJVM) was removed.

Note

For existing installations, HEIDENHAIN recommends deactivating the energy saver for the screen, and keeping the above-mentioned changes to the Microsoft Windows installation.

If you already have a hard disk with older NC software versions, you can install these expansions at any time.

Drives in PGM MGT

In PGM MGT of the iTNC 530 software, the Windows partition is available with the standard name **C:** as a connected network drive.
Network drives connected with the Windows computer are shown with the drive letters defined under Windows. Access to drives with so-called UNC network names, such as \\PC0815\directory, is not possible.
The **NETWORK** soft key no longer exists. Network drives must be connected and disconnected over the Windows Explorer. In such cases you must press the **UPDATE TREE** soft key in PGM MGT in order to refresh the view of the drives.
The control partitions (TNC:, PLC: and SYS:) are **not** displayed with their Windows drive letters (D:, E: and F:).

Attention

The subdirectories **\RECYCLER** and **\System Volume Information** on the TNC and PLC partition are created by Windows and must not be deleted or changed.

All drives connected with the Windows computer (including C:\) are treated by PGM MGT as connected network drives, i.e. during copying between these drives and the control partitions the data are converted from ASCII to binary format and vice versa.

Attention

During copying between network drives and the control partitions under Windows (D:, E: and F:) there is **no** ASCII to binary conversion. The copied files can become unusable and cause undefined behavior by the control.

- In MP7225 you define the Windows drives that are not supposed to appear in the TNC file management (PGM MGT).

MP7225	**Disable Windows drives in the TNC file manager**
Format:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Input:	If there is more than one drive, they are entered without spaces, e.g. MP7225 = CDE

Serial interfaces

The iTNC 530 (without Windows 2000) features two serial interfaces (X27: RS-232, X28: RS-422).

The following applies to the iTNC 530 with Windows 2000 or Windows XP:

- X27 and X28 can be operated via Windows 2000 and XP, and are only visible in the Windows operating system.
- X127 and X128 can be operated via the HeROS real-time operating system, and are therefore available in the Program Manager of the TNC (PGM MGT key). X127 and X128 are not visible in the Windows operating system (X127: RS-232, X128: RS-422).

Hard disk

The hard disk of the iTNC 530 with Windows 2000 has the following partitions.

The following sizes are valid for the partitions on hard disks for the iTNC 530 up to NC software 340 422-08:

Partition	Windows designation	Contents	Size
C	C	Windows partition	8.29 GB
TNC	D	User files	8.29 GB
PLC	E	OEM files	998 MB
SYS	F	System files	998 MB

The remaining memory is free.

The following sizes are valid for the partitions on hard disks for the iTNC 530 with Windows 200 that were delivered with NC software 340 422-09 or higher.

Partition	Windows designation	Contents	Size
C	C	Windows partition	13 GB
TNC	D	User files	13 GB
PLC	E	OEM files	1 GB
SYS	F	System files	1 GB

The following guidelines, for example, can be derived from the recommendation above:

The MC 422C with Windows XP supports the following partition sizes (approx.):

Partition	MC 422C	MC 422C (with Windows XP)
C:	–	17 GB
TNC:	25 GB	17 GB
PLC:	1 GB	1.95 GB
SYS:	2 GB	1.3 GB

The partitions of the hard disk can be displayed through **Start/Settings/ Control Panel/Administrative Tools/Computer Management/Storage/ Disk Management.**

System time

The HeROS CPU of the iTNC 530 with Windows 2000 or Windows XP internally uses UNIX system time. The time difference between the local time and the system time is taken from the Windows CPU. Therefore, setting the local time via the SET DATE/TIME soft key has no effect on the iTNC 530 with Windows.

Processor temperature

Module 9133 is used to determine the temperature of the second processor (See "Temperature of the MC 42x(B)" on page 908).

Module 9133 for determining the temperature of the second processor is not available on controls with Windows XP.

Set PLC output after shutdown

The automatic setting of a PLC output after a control has been shutdown (MP4040 to MP4042) is **not** possible on the iTNC 530 with Windows 2000 or Windows XP.

Operation

Both the Windows key and the ALT+TAB key combination remove the focus from the current Windows application (which could be the iTNC 530 software). If the focus is removed from the iTNC 530 software, the last keys pressed (such as soft keys) remain active. If the focus returns to the iTNC 530 software, the activation can be undone.

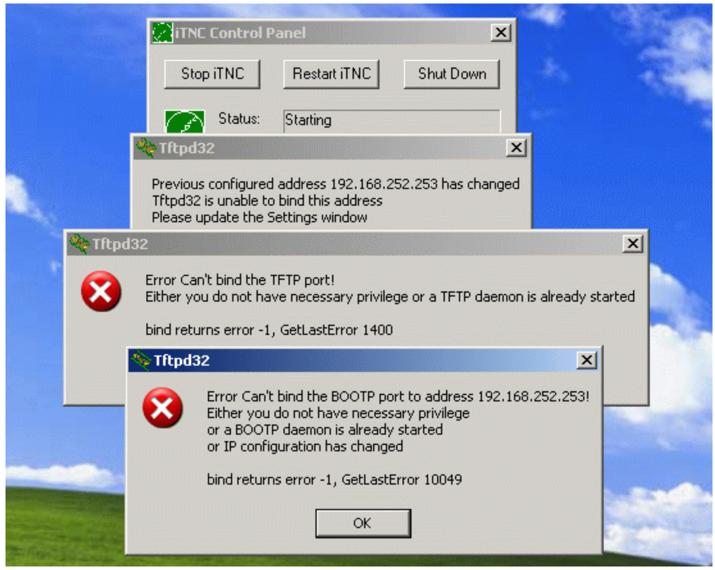
In general, the usual Windows operating properties are valid for the iTNC 530 software!

Setting the Windows language

Hard disks that are shipped as of NC software 340 480-06 support the Windows MultiLanguage version. In this version you can select the language of the operating system under **Start/Settings/Control Panel/Regional Settings.** This new function cannot be retrofitted.

11.2 Starting and Shutting Down the iTNC

On the MC 422C DP with Windows XP, X129 and X126 must be connected with each other via an RJ-45 cable (Ethernet cable). This is the connection between the HeROS computer and the Windows computer. If this connection is missing, the following Windows error messages of the Tftpd32 are displayed when you start the control application ("iTNC – Control Panel").



If you receive this error message, the main computer must be shut down and switched off with the main switch to ensure that it is no longer under power. The HeROS computer cannot be rebooted until the connection between X129 and X126 has been restored.

After the user logs on, the control software starts automatically under Windows 2000 or Windows XP. In the iTNC Control Panel the message **Starting** appears. Do not operate any Windows programs while this message is displayed. After the control software starts up, the iTNC Control Panel minimizes to a symbol in the status line.

The control software is shut down with the "Shutdown" soft key; in special cases also through the "iTNC Control Panel".

11.2.1 Logging a user off

A Windows user can log off at any time without impairing the iTNC 530 software. In this case the control screen is no longer visible and the user cannot make any entries.

Attention

Keys that are evaluated by the PLC, such as axis direction keys, remain active.

After the user logs on again, the control screen reappears and entries can be made again.

11.2.2 Exiting Windows

If you attempt to shut down or restart Windows after the control software is started, a message appears.

No aborts the process, and neither Windows nor the control software are shut down.
If you want to shut down the control software and Windows, you must first select **EMERGENCY OFF** and then **Yes.**
It is not possible to restart Windows with this method.

11.2.3 Setting up the manual start of the control software

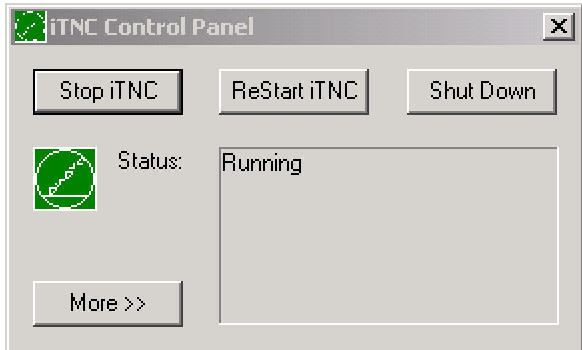
Starting of the iTNC Control Panel can be set from "automatic" to "manual". To do so, under **Start/Settings/Control Panel/Administrative Tools/ Services/iTNC530 Control Panel/Properties/Startup Type,** switch from **automatic** to **manual.**

Then the control software can only be started through **Start/Programs/ iTNC530/iTNC530 Start.**

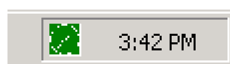
Note

A user with limited rights (e.g. TNC) might not be authorized to start the control software.

11.3 The iTNC Control Panel



The iTNC Control Panel appears after the user has logged on until the control software has been started. It can be called later at any time by double-clicking the symbol in the status display.

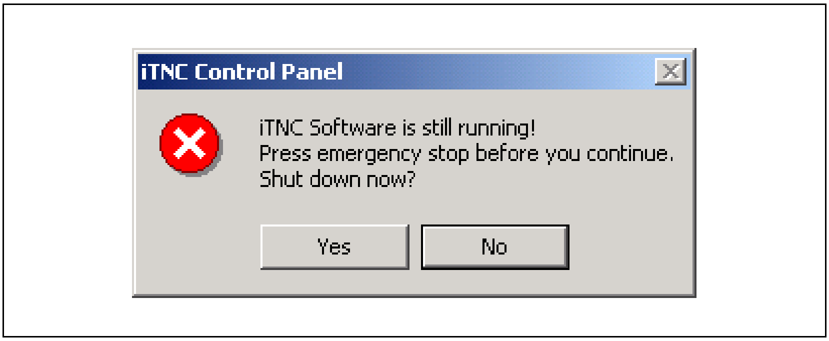


The iTNC Control Panel should not normally be used. The control software starts automatically when the user logs on to Windows. The control software and Windows are shut down with the "Shutdown" soft key.
The use of the iTNC Control Panel should be limited to servicing for initial operation, troubleshooting, software exchange, and installation of additional Windows software.

11.3.1 Functions

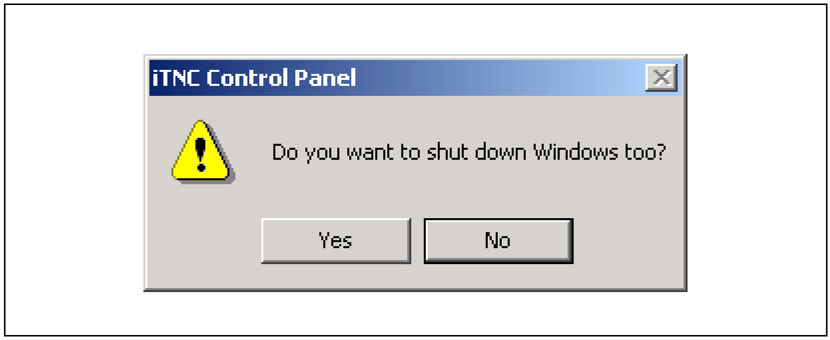
The iTNC Control Panel offers the following functions:

- **Status** text field: This text field displays the condition of the control software:
  - **Stopped:** The control software is not running.
  - **Starting:** The control software is being started.
  - **Running:** The control software is running.
  - **Aborting:** The control software is being shut down.
  - **Shut down iTNC:** The control software is being shut down and the iTNC Control Panel closed.
  - **Shut down iTNC and Windows:** The control software and Windows are being shut down.
  - **Restarting:** The control software is being started again.
  - **Error: <text>**: A control software error has occurred. Inform your service agency.
- **Stop iTNC** button: This button shuts down the control software. Open files are saved, the drives are switched off, and the PLC outputs are reset. **The EMERGENCY STOP button must be pressed before clicking this button.**
- **ReStart iTNC** button: If the control software is running, this button runs the **Stop iTNC** function and then restarts the software. **The EMERGENCY STOP button must be pressed before clicking this button.**
- **Shut Down** button: This button calls a message window with the question of whether the control software should be shut down:



- **No:** The control software will not be shut down.

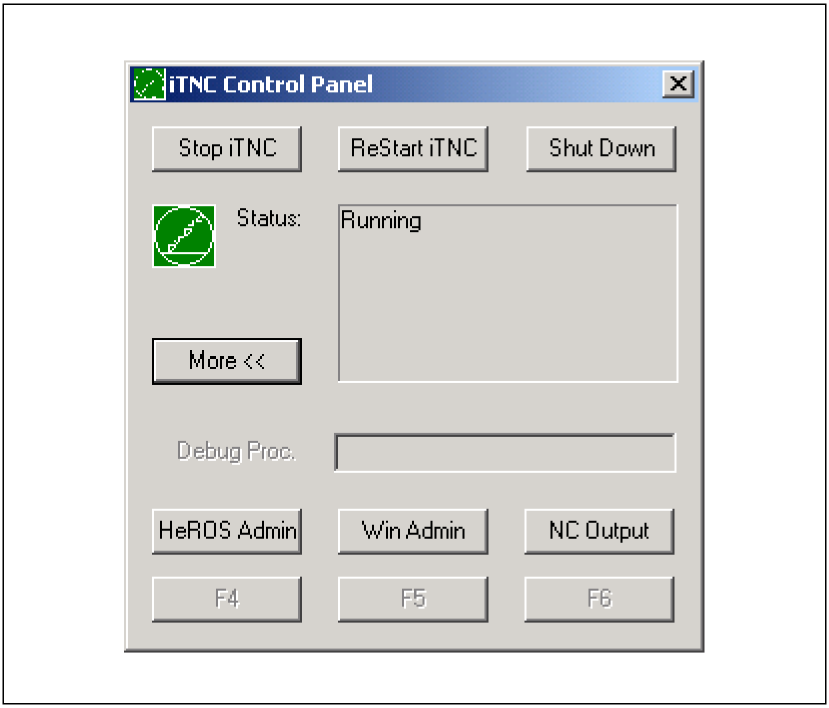
- If you want to shut down the control software, you must first select **EMERGENCY OFF** and then **Yes.** Another message window appears, asking whether Windows should also be shut down:



- **No:** Only the control software will be shut down. Windows will not be shut down
- **Yes:** First the control software is shut down, and then Windows.

11.3.2 Advanced functions

Clicking the **More >>** button opens additional functions of the iTNC Control Panel.



- **Debug Proc.:** This text box is reserved for HEIDENHAIN.
- **HeROS Admin** button: This button is reserved for HEIDENHAIN.
- **Win Admin** button: This button opens a text window for entering the current HEIDENHAIN system password. If the entry is correct, a text window opens with administrator rights. This enables HEIDENHAIN to grant administrator rights without disclosing the corresponding password.
- **NC Output** button: This button is reserved for HEIDENHAIN.
- **F4**, **F5**, **F6** buttons: These buttons have no function at present.

11.4 Network Settings

11.4.1 General information

The basis for the iTNC 530 with Windows 2000 is the MC 422(B) with two processors, and for Windows XP the MC 422C with two processors. One processor handles the real-time tasks with the HEIDENHAIN HeROS operating system. The second processor is dedicated to Windows.

The two operating systems (HeROS and Windows) communicate over TCP/IP. For this communication, HEIDENHAIN uses two fixed IP addresses (Subnet-Mask: 255.255.255.0):

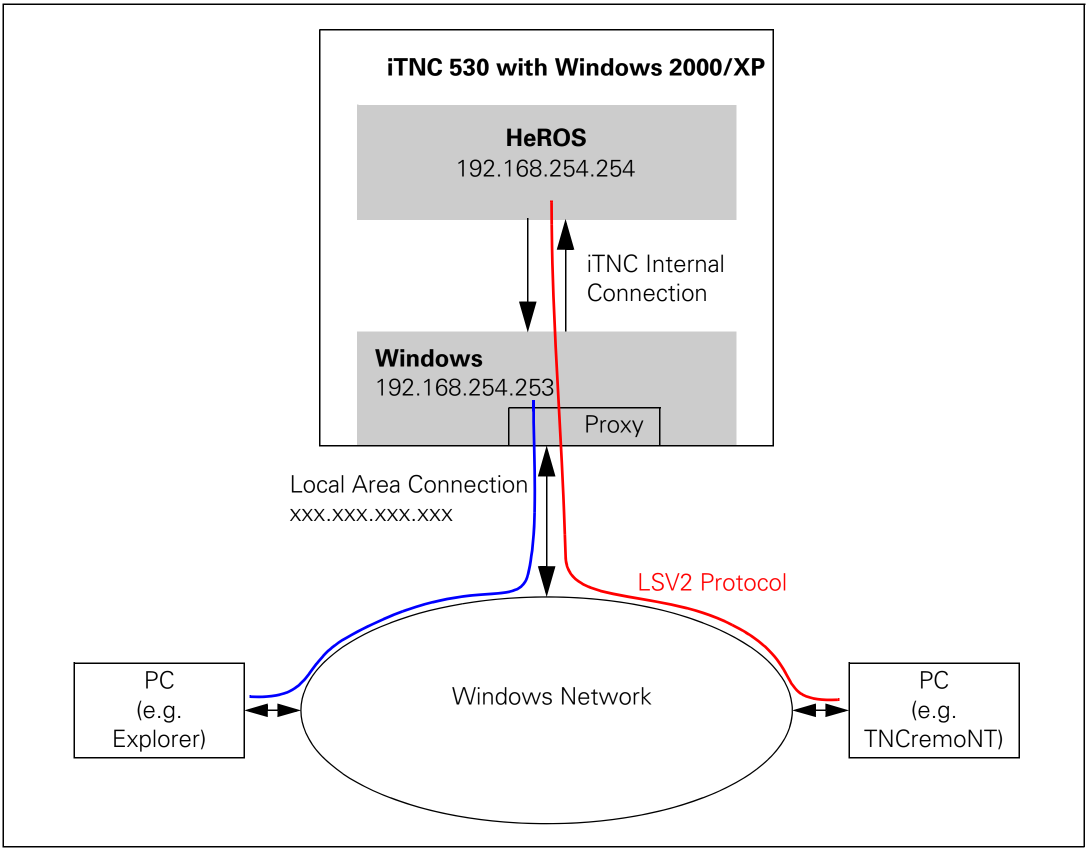
- 192.168.254.253: IP address of the Windows computer
- 192.168.254.254: IP address of the HeROS computer

The HeROS real-time operating system tests cyclically within 5 seconds whether a connection to the iTNC application exists under Windows, whether the file system can be accessed, and whether the X server for the display and keyboard is functioning. If one of these tests is impossible for more than 5 seconds, M4600 is set. M4600 is reset when all tests are possible again. If one test is impossible for more than 10 seconds, the control is shut down.

		Set	Reset
M4600	**Faulty internal communication between HeROS and Windows**	NC	NC

A proxy server is installed during installation of the iTNC 530 software under Windows. This proxy server ensures that incoming LSV2 commands are forwarded to the HeROS operating system.

The proxy server forwards a received LSV2 command to the IP address 192.168.254.254 of the HeROS computer and saves the IP address of the sender. After HeROS processes the command, the answer is sent to the proxy server over the IP address 192.168.254.253. The proxy server then forwards the answer to the saved IP address of the sender.

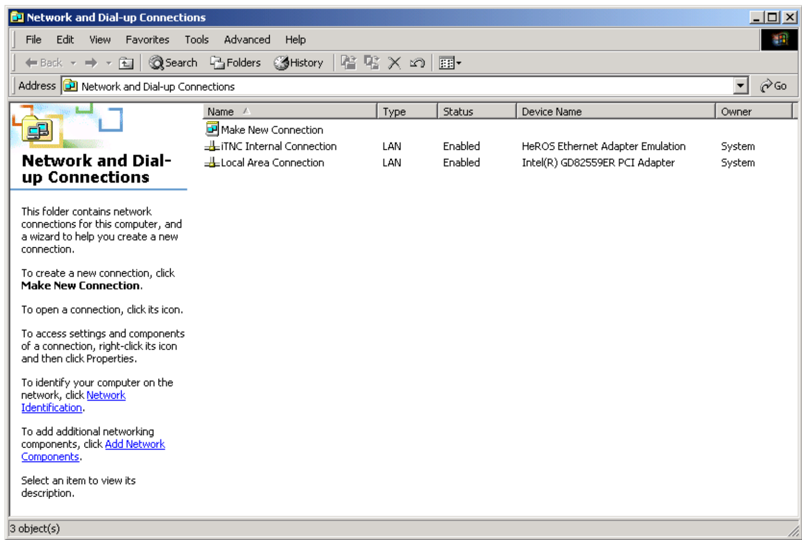


11.4.2 Windows settings

Two network adapters are integrated in Windows:

- Local Area Connection: Connection of the Windows computer to the Windows network (Intel(R) 8255xER PCI adapter)
- iTNC Internal Connection: Connection between the HeROS computer and the Windows computer (Intel(R) PRO/100 VE Network Connection)

Start/Settings/Network and Dial-up Connections displays an overview of the network connections:



Ask your network specialists for the network settings in your company network.

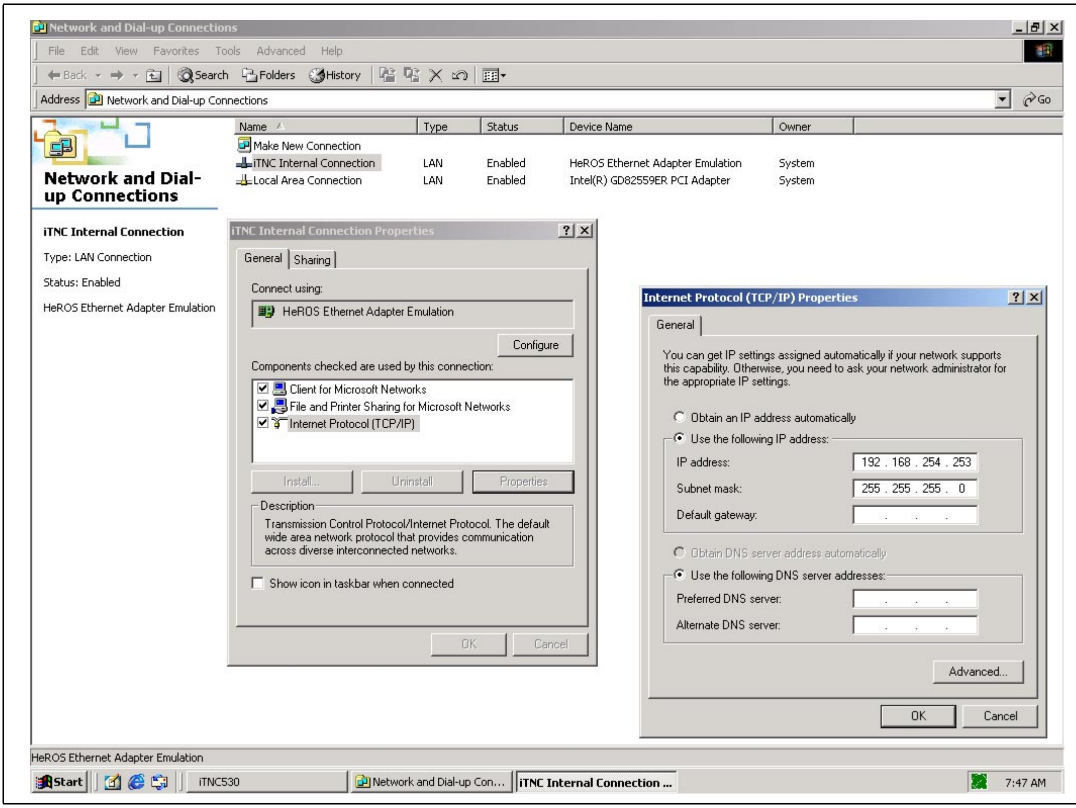
Note

Administrator rights are required for all network settings!

Attention

The network settings of the internal connection (iTNC Internal Connection) should not be changed!

The following figure shows the network settings for the connection between Windows and the HeROS computer.



Attention

These network settings should not be changed!

If the IP address 192.168.254.253 and the subnet mask 255.255.255.0 collide with the Windows network, in this exceptional case the IP address can be changed. The HeROS CPU is automatically given an IP address greater by 1 (e.g., if the Windows CPU is 192.168.254.253, then the HeROS CPU is 192.168.254.254).
The new IP address should also be a reserved IP address in the Internet. Reserved addresses on the Internet are:

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0. to 172.16.255.255
- 192.168.0.0 to 192.168.255.255

There must be no "routing" between the two networks ("iTNC Internal Connection" and "Local Area Connection").

Attention

Windows must be restarted for a changed IP address of the iTNC Internal Connection to go into effect.

11.5 Registered Users

Under Windows 2000, the following users are already registered by HEIDENHAIN:

User name	Password	User group	Editable	Description
Guest	(without password)	Guests	Yes, can also be deleted	User entered by Windows. Deactivated in default setting of iTNC 530.
TNC	(without password)	Users	Yes, can also be deleted	Example user name for the end user[a]
TNCP	SYS095148	Power Users	Yes, can also be deleted	Example user name for the end user[a]
OEM	SYS807667	Administrators	Yes, password can be changed	Administrator for the machine tool builder
Administrator	(not disclosed here)	Administrators	No	Only for HEIDENHAIN Service[b]
SYS_TNC		Administrators	No	User for access by the HeROS computer to the Windows computer[c]

a. The end user must not be assigned to the administrators' group because otherwise he would have access through Windows to the PLC and SYS partition.
b. The administrator must not be changed because otherwise HEIDENHAIN cannot offer any service.
c. The user SYS_TNC must not be changed because otherwise the control will not operate.

Under Windows XP, the following users are already registered by HEIDENHAIN:

User name	Password	User group	Editable	Description
TNC	(without password)	Users	Yes, can also be deleted	Example user name for the end user[a]
TNCP	SYS095148	Power Users	Yes, can also be deleted	Example user name for the end user[a]
OEM	SYS807667	Administrators	Yes, password can be changed	Administrator for the machine tool builder
Administrator	(not disclosed here)	Administrators	No	Only for HEIDENHAIN Service[b]
SYS_TNC		Administrators	No	User for access by the HeROS computer to the Windows computer[c]

a. The end user must not be assigned to the administrators' group because otherwise he would have access through Windows to the PLC and SYS partition.
b. The administrator must not be changed because otherwise HEIDENHAIN cannot offer any service.
c. The user SYS_TNC must not be changed because otherwise the control will not operate.

Note

Pay attention to capitalization when entering the password.

The Windows Help contains general information about user groups.

If the control is **not** in a domain, then under Windows 2000 you can enter users who will be logged on automatically. Please refer to the online help of Windows 2000 or Windows XP.

In the factory default setting, the MC 422C DP displays the two user options **TNC User** and **TNC Power User** on the screen when you log onto Windows XP. The user **TNC User** corresponds to the user **TNC**, and the user **TNC Power User** corresponds to the user **TNCP** of the table shown above.

Press the key combination **ALT CTRL+DEL** twice to switch from the displayed login screen to the same login window as in Windows 2000, which enables you to use all users listed in the table shown above after entering the user and the appropriate password from the table.

Note for administrators

All administrators have access from Windows to the TNC, PLC and SYS partitions (drives D:, E: and F:). Some of the data on these drives is in binary format.

Attention

During writing or copying processes from Windows to the control partitions there is no ASCII-to-binary conversion. The copied files can become unusable and cause undefined behavior by the control.

The "administrators" and "SYSTEM" user groups have access to the control partitions (D:, E: and F:). This is required for correct operation of the control. Therefore:

- Access for these user groups must not be limited.
- No other user groups can be added.
- Specific access types within the user groups cannot be prohibited.

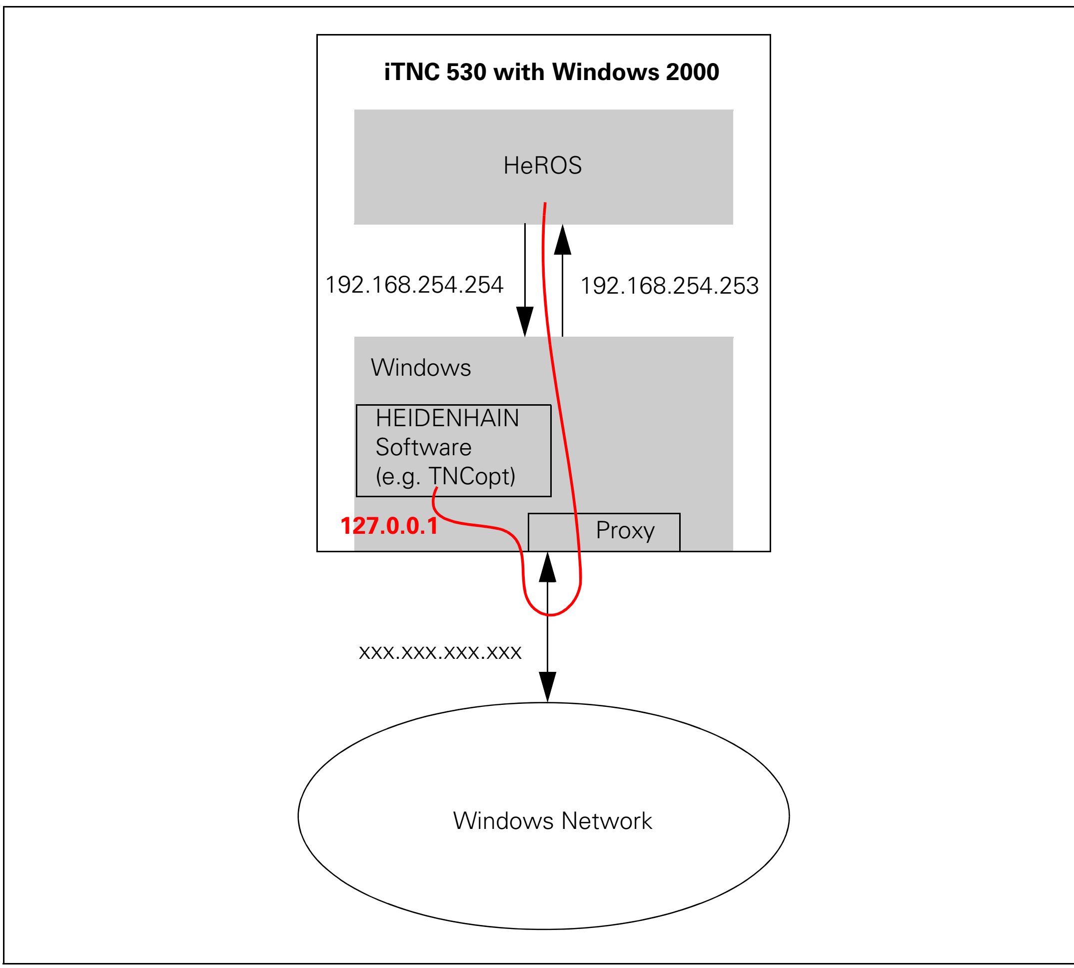
11.6 Software Installation on the Windows Computer

11.6.1 HEIDENHAIN software

If HEIDENHAIN software is installed on the Windows computer that communicates with the HeROS computer over LSV2 (e.g. TNCopt), the IP address of the HeROS computer is to be defined as **127.0.0.1** (loopback interface).

With this special IP address, the software sets up a connection with its own Windows computer. The proxy server forwards the LSV2 commands to the HeROS computer and sends back the answers from the HeROS computer.

An alternative to this IP address is the address of the HeROS computer. However, this can cause problems, for example due to address conflicts, if this IP address has to be changed (See "Windows settings" on page 1858).



11.6.2 Non-HEIDENHAIN software

To install non-HEIDENHAIN software on the Windows computer, you need administrator rights (e.g. user name: OEM).

Because the HeROS computer accesses the hard disk on the Windows computer and the iTNC user interface runs on the Windows computer, adequate system resources must be available at all times on the Windows computer.

- CPU power
- RAM
- Faster access to the hard disk

Brief interruptions in the availability of these resources are compensated by the HeROS computer through an "NC data buffer." Interruptions longer than these can influence the HeROS computer, which in the worst case can cause feed-rate interruptions during execution of an NC program.

The software to be installed:

- Must **not** place demands on the Windows computer approaching the limits (128 MB RAM, AMD K6/2 with 266 MHz)
- Must **not** be run in the following Windows priority stages:
  - **Above normal**
  - **High**
  - **Real time**

The Windows priority stage can be checked in the **Windows Task Manager** on the **Processes** tab in the **Base Priority** column.

Starting the **Windows Task Manager:**

- Click with the right mouse key on a free area of the taskbar.
- Click **Task Manager...**

Note

Special care must be taken when installing virus scanners, since they are very compute-intensive. A virus scanner should only be started if nothing is currently being machined!
In addition, the virus scanner is not permitted to install a monitor on the "iTNC Internal Connection" network connection. I.e., network data coming in over this connection is not allowed to be checked!

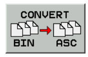
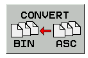
To check whether a non-HEIDENHAIN program could cause problems:

- Start an NC program with 3-D movements that fully exploit the power of the iTNC.
- Use the integrated oscilloscope to record the contouring feed rate.
- Start the program in question on the Windows computer.
- Run the program's most resource-intensive operations.

Note

If you have interruptions in the contouring feed rate, the non-HEIDENHAIN software should **not** be used on the Windows computer.

11.7 NC Software Exchange on the iTNC 530 with Windows

Soft key	Function
	Convert the files on the hard disk from binary format to ASCII format and save nonvolatile markers in the PLCMEM.A file.
	Convert the files on the hard disk from ASCII format to binary format and save nonvolatile markers in the PLCMEM.A file.
	Copy cutting-data tables, tables for tilting-axis geometry, and the table of M-function macros from the SYS partition into the corresponding directories of the TNC or PLC partition, and create prototypes of the tables.

Note

The NC software must be exchanged only by trained personnel.

To enable the user to exchange the NC software, HEIDENHAIN accompanies it with setup software that is installed under Windows.

Information about the cycles

Keep the following information in mind when you use OEM cycles in the HEIDENHAIN cycle tree instead of in an OEM cycle tree:

Change the OEM cycles into binary format before reconversion, otherwise the iTNC will not recognize these cycles, and will add ERROR blocks to the NC programs. These ERROR blocks must be deleted manually.

After an NC software exchange, to be able to use the latest HEIDENHAIN cycles together with your existing customized cycles, you will need the PC software **CycleDesign** to insert the new cycles in your *.CDF file. The new *.CDF file and the appropriate CONSTCYC.CDC for the HEIDENHAIN cycles are still in the folder PLC:\JH\ on the control after the NC software exchange. You can find more information in the User's Manual or in the Help for CycleDesign.

Procedure for exchanging the NC software up to and including 340 422-12

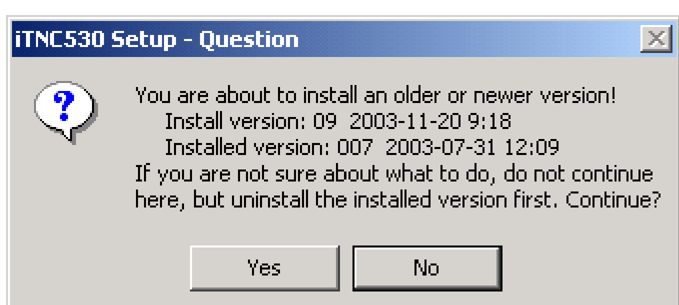
The iTNC 530 software can be installed only by a user with local administrator rights (e.g., user name: OEM).

> Attention
>
> All messages that differ from the description below indicate problems in the installation. In this case, inform your service agency.

- While in the `Programming and Editing` operating mode, press the MOD key.
- Enter the code number 95148 and confirm your entry with the ENT key.
- While in the `Machine-parameter programming` mode, press the MOD key.
- Press the UPDATE DATA soft key.
- The name and path of a log file can be entered after `Path =` in the header.
- Press the BIN → ASC soft key to convert the files on the hard disk from binary to ASCII format.

Equivalent file name extensions in ASCII and binary format					
.H	.H%	.I	.I%	.T	.T%
.TCH	.TC%	.D	.D%	.P	.P%
.PNT	.PN%	.COM	.CO%	.CMA	.CM%

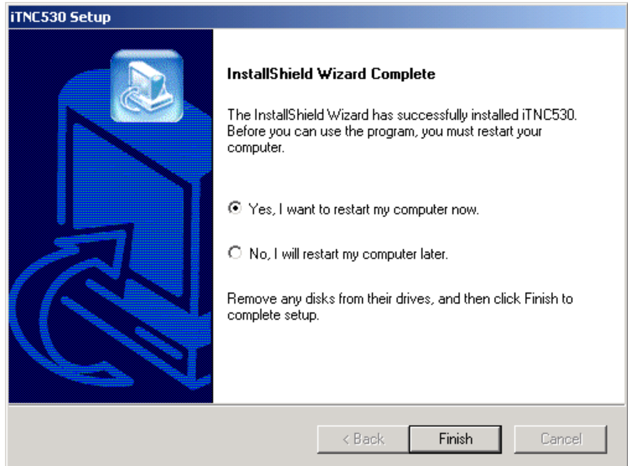
- Click the **Shut Down** button in the **iTNC Control Panel.** Click the **No** button on the question of whether Windows should also be shut down.
- Start the Setup.exe of the new iTNC 530 software, e.g. through **Start/Run/**.
- You may receive the following message after Setup.exe has started. Click the **Yes** button.



Then the installation wizard starts the installation.

After the installation, Windows must be restarted.

- Click the **Finish** button.



- Log in again with the user name under which the installation of the iTNC 530 software was started (e.g. OEM).

The control software starts automatically.

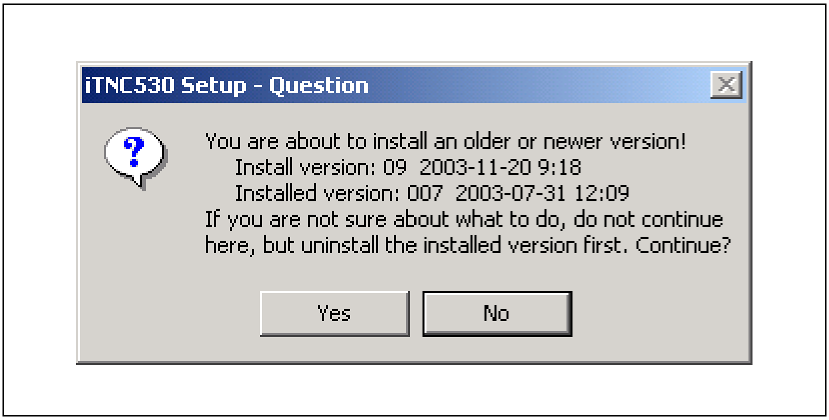
- If required, complete or delete the machine parameters.
- While in the **`Programming and Editing`** operating mode, press the MOD key.
- Enter the code number 95148 and confirm your entry with the ENT key.
- While in the **`Machine-parameter programming`** mode, press the MOD key.
- Press the UPDATE DATA soft key.
- Press the ASC → BIN soft key to reconvert the files on the hard disk from ASCII format into binary format.
- Read-in files which you had saved to a PC.
- The NC software exchange is completed.
- With the COPY SAMPLE FILES soft key, the HEIDENHAIN standard tables for cutting data, the tilting-axis geometry, and the M-function macros can be copied into the corresponding directories.

Procedure for exchanging the NC software if SW 340 492-01 or SW 340 493-01 is already installed

The iTNC 530 software can be installed only by a user with local administrator rights (e.g., user name: OEM).

> **Attention**
>
> All messages that differ from the description below indicate problems in the installation. In this case, inform your service agency.

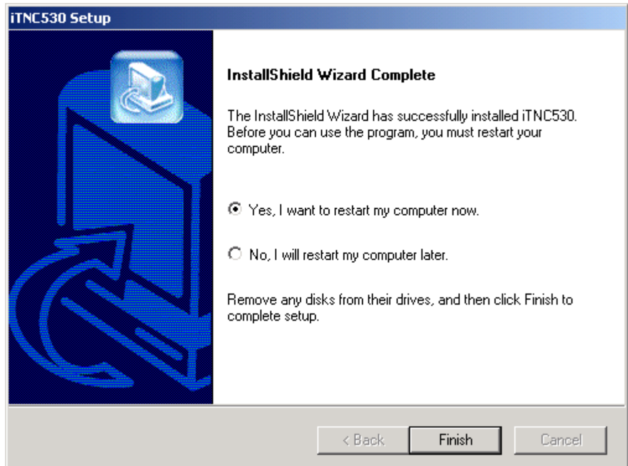
- Use the PC program TNCremoNT from HEIDENHAIN to make a backup of the control software.
- Start the Setup.exe of the new iTNC 530 software, e.g. through **Start/Run/**.
- You may receive a similar message after Setup.exe has started. Click the **Yes** button.



Then the installation wizard starts the installation.

After the installation, Windows must be restarted.

▶ Click the **Finish** button.



Note

- Service packs are also loaded in this manner.
- Stopping the NC software via the Control Panel is not necessary in the new update procedure.
- Binary-ASCII conversion is no longer necessary, as this is requested and performed internally by the update procedure.

▶ Log in again with the user name under which the installation of the iTNC 530 software was started (e.g. OEM).

The control software starts automatically.

▶ If required, complete or delete the machine parameters.

▶ Read-in files which you had saved to a PC.

▶ The NC software exchange is completed.

▶ With the COPY SAMPLE FILES soft key, the HEIDENHAIN standard tables for cutting data, the tilting-axis geometry, and the M-function macros can be copied into the corresponding directories.

Automated update

If one of the following directories exists when an iTNC with Windows is booted,
- **D:\install** (**D:** corresponds to the TNC partition) or, if a USB memory device is connected
- **G:\install** (**G:** corresponds to the drive letter of the USB memory device – network drives are not permitted!)
and if a **setup.ini** control file is saved in this directory, then an automated update is performed according to the instructions in this control file (See "Automated update (setup.ini)" on page 314). If this file does not exist, the update must be started manually via the **SETUP** keyword.

An automated update is usually a part of a manual update.

Entries in the log file

If errors occur during conversion, the TNC will display error messages and record them in the log file. During the NC software switch, the name and path of a log file can be entered in the header after **Path =**; the extension .A must be used. If no entry is made in this line, the file TNC:\CVREPORT.A is created.

Each error message contains

- Error message
- Error code
- Error cause
- File concerned

Example:

```
==================================================
ERROR              :REMANENT PLC DATA NOT RESTORED
ERRNO              :2
ERROR MESSAGE      :Program name not found
FILE               :PLCMEM.A
==================================================
```

Error message	Meaning
CANNOT OPEN DIRECTORY	File could not be opened
REMANENT PLC DATA NOT RESTORED	No access to the file PLCMEM.A
NOT ENOUGH SPACE	Too little free memory on the hard disk
CONVERSION BIN ASC FAILED	A binary file has an incorrect format (e.g., binary format from an old NC software)
CONVERSION ASC BIN FAILED	An ASCII file on the hard disk is incorrect

11.8 Special Functions of iTNC 530 with Windows

11.8.1 Operation and user interface

Controlling Windows windows

The iTNC offers the possibility in Windows of bringing the windows of active programs to the foreground or minimizing them. This can be necessary, for example, if the user is to pay attention to a specific program.

In order to influence the visibility of Windows windows, proceed as follows:

- Use PLC Module 9317 to determine the Windows title of the Windows program in the foreground at the time of the interrogation (name of the application in the title bar of the window).
- Use PLC Module 9316 to interrogate the relevant status before changing a Windows window.
- Use PLC Module 9315 to change the status of the desired Windows window (minimize or bring to the foreground).

In order to bring a Windows application to the foreground, the user-specific Windows system parameter "ForegroundLockTimeout" must be set to 0 (default: 200000). Otherwise the activation of the window is signaled only by a blinking of its icon in the task bar. The system parameter is automatically set by a program in the Startup folder for the iTNC 530 with Windows 2000. The system parameter is not automatically changed on the programming station. Here there are two shortcuts in the Start menu, with which the parameter can be set to 0 (ON) or the default value 200000 (OFF).

Module 9317 Determining a Windows window title
Module 9317 is used to determine the title of the Windows window in the foreground at the time of the request.

Call:
PS B/W/D/K <Mode>
0: Title of the Windows window in the foreground
PS B/W/D/K <PLC string address for window title>
CM 9317

Error recognition:

Marker	Value	Meaning
M4203	0	Windows window title determined
	1	Error code in W1022
W1022	1	Invalid mode programmed
	3	Invalid PLC string address programmed
	20	Call was not in a submit or spawn job
	52	Single-processor control, title interrogation not possible

Module 9316 Status interrogation of a Windows window
Module 9316 interrogates the current status of a Windows window.

Call:
PS B/W/D/K <Mode>
0: Window status
PS B/W/D/K/S<Window title>
CM 9316
PL B/W/D <Window status>
Bit 0: Window in foreground
Bit 1: Window minimized

Error recognition:

Marker	Value	Meaning
M4203	0	Windows window status determined
	1	Error code in W1022
W1022	1	Invalid mode programmed
	2	Window with this title does not exist
	3	Invalid PLC string address programmed
	11	Invalid string programmed for window title
	20	Call was not in a submit or spawn job
	52	Single-processor control, status interrogation not possible

Module 9315 Bring a Windows window to the foreground or minimize it

Module 9315 is used on the iTNC 530 with Windows 2000 and on the programming station to bring the window of a Windows application from the PLC to the foreground and make it active, or to minimize it. In the module call you enter the title of the window that is to be brought to the foreground. Uppercase and lowercase are ignored in the window title entered, as well as any leading or trailing blank spaces.

Call:

```
PS    B/W/D/K   <Mode>
                0: Bring window to the foreground
                1: Minimize window
PS    B/W/D/K/S<Window title>
CM    9315
```

Error recognition:

Marker	Value	Meaning
M4203	0	Windows window activated/minimized
	1	Error code in W1022
W1022	1	Invalid mode programmed
	2	Window with this title does not exist
	3	Invalid PLC string address programmed
	11	Invalid string programmed for window title
	20	Call was not in a submit or spawn job
	52	Single-processor control, switchover not possible

12 Error Messages

12.1 DSP/NC Error Messages

You can find a list of all NC and DSP error messages on the HEIDENHAIN homepage (www.heidenhain.de) under:

- Services and Documentation
- Documentation / Information
- User documentation
- TNCguide
- Select the desired language, such as English.
- TNC Controls
- NC Error Messages

Here you will find the current, control-specific help file (*.chm) with all error messages. These files are also part of the TNCguide context-sensitive help system for the respective control.

12.2 iTNC Error Messages during Data Transfer

The following error messages can occur during data transfer through an interface:

General error messages	
Interface already assigned	The interface is already transmitting, or the transmission was not concluded.
Program incomplete	A transmission was broken off or the file was not correctly concluded (no END character or END block).
Ext. in-/output not ready	The interface is not connected, or the peripheral unit is switched off or defective.
Data transfer erroneous x	x = error code (see table).

Error codes:

Error code	Meaning
E	During data transfer with BCC, the <NAK> signal was received 15 times in succession.
A to H except E	Error code of the receiver module with one of the following causes: ■ The transfer-rate setting of the iTNC and peripheral device do not match. ■ The parity bit is erroneous. ■ Erroneous data frame (e.g.: no stop bit). ■ The receiver module of the interface is defective.
K	During transmission of an error to the iTNC, the <1> character was not transmitted after the <ESC> character.
L	After the error sequence <ESC><1> an incorrect error number was received (error numbers 0 to 7 are permitted).
N	An expected acknowledgment <ACK> or <NAK> was not transmitted by a certain time.
M	During data transfer with BCC, the <NAK> signal was transmitted 15 times in succession.

Codes K and L are shown only during transmission with the standard data transmission protocol.

12.3 Error Messages of the File System

The following error messages can be displayed on the iTNC:

Error message	Corrective action
File system error 1	Inform your service agency.
File system error 2	
File system error 3	
File system error 4	
File system error 5	
File system error 6	
File system error 7	
File system error 8	
File system error 9	
File system error 10	
File system error A	
File system error B	
File system error C	
File system error D	
File system error E	
File system error F	
File system error G	
File system error H	
File system error I	
File system error J	
File system error K	
File system error L	

13 Subject Index

A

B

C

C

D

E

F

F

G

H

I

J

K

L

M

M

N

O

P

P

Q

R

S

S

S

T

T

U

V

W

X

HEIDENHAIN

DR. JOHANNES HEIDENHAIN GmbH
Dr.-Johannes-Heidenhain-Straße 5
83301 Traunreut, Germany
☏ +49 8669 31-0
FAX +49 8669 5061
E-mail: info@heidenhain.de

Technical support FAX +49 8669 32-1000
Measuring systems ☏ +49 8669 31-3104
E-mail: service.ms-support@heidenhain.de
TNC support ☏ +49 8669 31-3101
E-mail: service.nc-support@heidenhain.de
NC programming ☏ +49 8669 31-3103
E-mail: service.nc-pgm@heidenhain.de
PLC programming ☏ +49 8669 31-3102
E-mail: service.plc@heidenhain.de
Lathe controls ☏ +49 8669 31-3105
E-mail: service.lathe-support@heidenhain.de

www.heidenhain.de

581 295 · 06 · A · 02 · PDF · 2/2011