HEIDENHAIN

Technical Manual

iTNC 530

NC Software	340 420-07
	340 421-07
	340 422-01
	340 423-01

October 2002

363 812-11 · 5 · 10/2002 · Printed in Germany · Subject to change without

Foreword

This Technical Manual has been written for all machine tool manufacturers and distributors. It contains all of the information necessary for the mounting, electrical connection, commissioning and PLC programming of HEIDENHAIN contouring controls.

Every time the hardware or software of HEIDENHAIN's contouring control is updated, you will receive a set of supplementary pages free-of-charge. Always sort these pages into your Technical Manual immediately. In this way, your manual will always be up-to-date.

You can use extracts from this manual to supplement your machine documentation. If you increase the size of the manual format (17 cm x 24 cm) by the factor 1.225, you will have DIN A4 format.

No documentation can be perfect. To stay up to date, documentation must change constantly. It also thrives on your comments and suggestions for improvement. Please help us by sending us your ideas.

DR. JOHANNES HEIDENHAIN GmbH
E/P Department
Dr.-Johannes-Heidenhain-Str. 5
83301 Traunreut

1 Update Information

1.1 General Information

You will receive Update Information supplements for the iTNC 530 at irregular intervals.

Please file this information here.

1 Update Information No. 2

1.1 Releases

The following versions of the NC software were released:

- NC software 340 420-06 and 340 421-06 July 2002
- NC software 340 420-07 and 340 421-07 August 2002
- NC software 340 422-01 and 340 423-01 August 2002

1.2 NC Software 340 420-xx

NC software	Export version	Release
340 420-06	340 421-06	July 2002

Machine parameters

- MP110.x and MP111.x were expanded:
  An error message appears when values are entered for position encoder inputs that do not exist.
- The maximum input range of MP960.x was increased from +/– 99 999.9999 to +/– 1.79769313486E+308 (= $1.79769313486 \cdot 10^{308}$).
- MP2160 has been expanded into MP2160.x:
  Input value 2 selects operation with HEIDENHAIN EcoDyn synchronous motors. For this to function, you must select from the motor table the motors with the designation **QSY1xxx EcoDyn** or **QSY1xxx EcoDyn EnDat** for MP2200.x.
- MP2195 is new:
  Suppress error messages from the HEIDENHAIN supply units
  Input: %xxxxxxxx
  0: Error message is not suppressed
  1: Error message is suppressed
  Bit 0: Reserved
  Bit 1: ERR.UZ.GR signal
  Bit 2: ERR.TMP signal
  Bit 3: Reserved
  Bit 4: ERR.IZ.GR signal
  Bit 5: RDY.PS signal
  Bit 6: ERR.ILEAK signal
  Bit 7: Reserved
- MP2220 has been expanded:
  Bit 1 – Monitoring the rotational direction
  This monitoring function **cannot** be switched off for synchronous motors (entry **SM** in the column **TYPE** of the motor table).
  Bit 3 – Activated motor brakes cause a change of the controlled system. This change can lead to vibrations when the controller is switched off while the brakes are on.
  Bit 3 = 0: Suppress vibrations
  Bit 3 = 1: Vibrations are allowed
- MP2304.x is new:
  Reference value for I^2t monitoring of the power module
  Input: 0 to 1000.000 [· rated current of power module]
  0: I^2t monitoring of the power module switched off
  1: Rated current of power module as reference value

- MP2308.x is new:
  Time between the output of the braking signal $\overline{\text{BRK}}$ (X51 to X62) and the switch-off of the controller (overlap time).
  Input: 0.001 to 0.500 [s]
  0 = 0.200 s
- MP7263 has been expanded:
  With bit 1 you can configure the output of the column in the pocket table during backup and during conversion from binary format to ASCII.
  Bit 1 = 0: Output only the displayed columns
  Bit 1 = 1: Output all columns
- MP7357.x has been expanded:
  Color settings of the "machine" soft-key display
  MP7357.2 = $0000000 Inactive soft-key row
  MP7357.3 = $00000FF Active soft-key row
- MP7358.x has been expanded:
  Color settings of the "programming" soft-key display
  MP7357.2 = $0000000 Inactive soft-key row
  MP7357.3 = $00000FF Active soft-key row
- MP7370.x has been expanded:
  Color settings of the small PLC window
  MP7370.15 = $0FF2020 Color 15
- MP7481.x is new:
  For each tool change operation between magazines you can specify the sequence with which the new tool and the tool be returned are to be output. M4540 remains in effect.
  Input: %xxxx
  0: First, output the pocket of the tool to be returned
  1: First, output the pocket of the new tool
  Bit 0: New tool from magazine 1
  Bit 1: New tool from magazine 2
  Bit 2: New tool from magazine 3
  Bit 2: New tool from magazine 4
  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
- MP7482 is new:
  You can specify independently for each magazine whether it should operate with variable or fixed pocket coding. MP7480.x must be set to 3 or 4.
  Input: %xxxx
  0: Magazine does not use fixed pocket coding
  1: Magazine uses fixed pocket coding
  Bit 0: Magazine 1
  Bit 1: Magazine 2
  Bit 2: Magazine 3
  Bit 3: Magazine 4
- MP7684 has been expanded:
  Bit 7 – Reserved

Configuring the axes and spindle

- After switching on the controller (Module 9161) there is a delay of 50 ms in the switch-on of the current controller. This also delays the acknowledgment over Module 9162 by 50 ms. This may have to be taken into account in the PLC program.

Machine integration

- In the log, the end of an NC macro is indicated with the entry **MACEND.**
- If a *.PET table contains more than 999 PLC error messages, the excessive messages are ignored and the error message **PET table: Too many lines** appears.
- In the manual operating modes, the screen switchover key is active after the M or S function has been started. This makes it possible to display a PLC window when an M function starts.
- M4185 can determine whether an internal stop was performed. The marker must be reset by the PLC.
- New possible entries in the soft-key resource file *.SPJ:
  - The entry **VROOT** in the header of the main menu defines a menu for the vertical soft-key row, while **HROOT** defines it for the horizontal row.
  - The entry **EMODE** in the heading of the main menu defines a menu for the programming modes, and the entry **MMODE** defines a menu for the machining modes.
  - The entry **ENABLE: <marker>** locks (marker = 1) or enables (marker = 0) a soft key.
  - The entry **STATUS: <marker or word>** assigns an operand to a soft key (in addition to W302/W304). When the soft key is pressed, the marker is set or the soft-key number is entered in the word.
  - The entry **POPUPMENU: <menu name>** displays a soft-key menu in the respective other soft-key row. The entry **CLOSEPOPUPMENU** closes this soft-key menu again.
  - The entry **LARGEWINDOW: <mask file for PLC window>** or **SMALLWINDOW: <mask file for PLC window>** opens a large PLC window with the specified mask file. **LARGEWINDOW** opens a large PLC window over the entire screen, **SMALLWINDOW** opens a large PLC window instead of the graphic/status window. The entry **CLOSEPLCWINDOW** closes the PLC window again.
  - The entry **FirstInGroup** indicates the first RADIO soft key in a group of RADIO soft keys, if more than one group with RADIO soft keys exists in a soft-key menu. If in addition to **FirstInGroup,** the entry **STATUS: <word>** also exists, the number of the pressed soft key is saved in the word (beginning with 0). The entries **FirstInGroup** and **STATUS: <word>** can also be used for groups of CHECK soft keys.

PLC programming

- The WATCH LIST soft key in the PLC main menu is for PLC diagnosis. With the WATCH LIST function you can create a table with dynamic display of the states of the selected operands
- **FN18: SYSREAD ID52 NR2 IDX<tool number>** finds the corresponding tool magazine.

PLC modules

Module 9136 Switching the touch probe on/off

Module 9136 switches a touch probe on 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.

Call:

PS	B/W/D/K	<Touch probe state> 0: Switch off touch probe 1: Switch on touch probe
CM	9136	

Error detection:

Marker	Value	Meaning
M4203	0	Touch probe on or off
	1	Error code in W1022
W1022	1	Invalid touch probe state

Module 9157 Drive controller status

The module was expanded by the status information 4 (spindle in operating mode 0 (bit 15 = 0) or operating mode 1 (bit 15 = 1)).

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.

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 detection:

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

Module 9321 Find the current block number
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 detection:

Marker	Value	Meaning
M4203	0	Block number has been found
	1	Error code in W1022
W1022	2	Invalid string number

Module 9305 Tool exchange in the pocket table
The module was expanded by an error code.

Error detection:

Marker	Value	Meaning
M4203	0	Tools exchanged
	1	Error code in W1022
W1022	2	Invalid pocket number
	6	Magazine management using magazine rules is active
	20	Module was not called in a submit job or spawn job
	21	Module was called during NC program run
	30	No valid tool in the original pocket

Module 9306 Exchange tools between tool magazines
The module was expanded by an error code.

Error detection:

Marker	Value	Meaning
M4203	0	Tools exchanged
	1	Error code in W1022
W1022	1	Invalid pocket number
	2	Invalid magazine number
	6	Magazine management using magazine rules is active
	20	Module was not called in a submit job or spawn job
	21	Module was called during NC program run
	30	No valid tool in the original pocket
	36	File error

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 detection:

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 submit job or spawn job
	30	Tool not found
	36	File error in pocket table

Miscellaneous

- If a **tolerance for rotary axes** is programmed with inactive HSC filter (MP1094 = 0), the error message **rotary axis tolerance not allowed** appears.
- In the **Editing machine parameters** operating mode, ASCII files with the extension .A (e.g. READ_MP.A) can be displayed.
- With **M140 MB<retraction length> F<feed rate,>** a feed rate can be optionally programmed in order to move away from the contour in the current tool axis direction.
- A network ping can be run in the network settings through the code number **NET123.**
- The software version of the control (standard or export version) is saved in the SIK. If the NC software is exchanged for another software version, after the control starts up a message appears that the control can be operated only as a programming station. This message must be acknowledged.
  If you exchange software from a standard version to the export version, the control asks whether the compressed files of the standard versions should 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.
- If the control hardware components have to be loaded with a new controller during control startup, a display of progress appears.
- With the NC block **CYCLE CALL POS** you can move to the specified position and call a cycle.

NC software	Export version	Release
340 420-07	340 421-07	August 2002

1.3 NC Software 340 422-xx

HEIDENHAIN released the new NC software 340 422-01 for the iTNC 530 in August 2002. This NC software will only be delivered if you specifically order it. There are new User's Manuals for this NC software:

- HEIDENHAIN conversational programming: Id. Nr. 375 738-xx
- Touch probe cycles: Id. Nr. 375 319-xx

NC software	Export version	Release
340 422-01	340 423-01	August 2002

Expansions since NC software 340 420-07:

Machine parameters

- MP7294 is new:
  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

Machine integration

- After COPY SAMPLE FILE is executed, two new prototypes for pallet tables (PROTOPR.P and PROTO_TOPR.P), containing the column **PRESET,** are saved in the directory PLC:\PROTO.

Miscellaneous

- Multiple presets can be managed with the preset table TNC:\PRESET.PR. The presets are recalculated using the defined tilting-axis geometry.
- The cycle structure of the machining cycles was revised.
- If MP7475 = 1 (Use machine datum as datum for datum tables) and Cycle 7 (DATUM) is programmed, the error message **Use preset table!** appears.
- In Cycle 205 (UNIVERSAL PECKING) a lowered start point can be entered in Q379.
- In Cycle 220 (POLAR PATTERN) the type of traverse (straight line or circular arc) can be entered in Q365.
- Cycle 247 (SET DATUM) activates a preset from the preset table.
- In Cycles 400 (BASIC ROTATION), 401 (ROT OF 2 HOLES) and 402 (ROT OF 2 STUDS) you can enter in Q305 the line in the preset table in which the basic rotation is to be entered.
- In Cycles 410 to 418 you can decide in Q303 if the measured values are to be written in a datum table or preset table.
- In Cycles 414 (DATUM OUTSIDE CORNER) and 415 (DATUM INSIDE CORNER) the line in the preset table for the datum and basic rotation can be entered in Q305.
- Cycle 419 (DATUM IN ONE AXIS) can be used to set a datum in any axis.
- Cycles 420 to 430 take an active rotation into account.

2 Introduction

2 Introduction

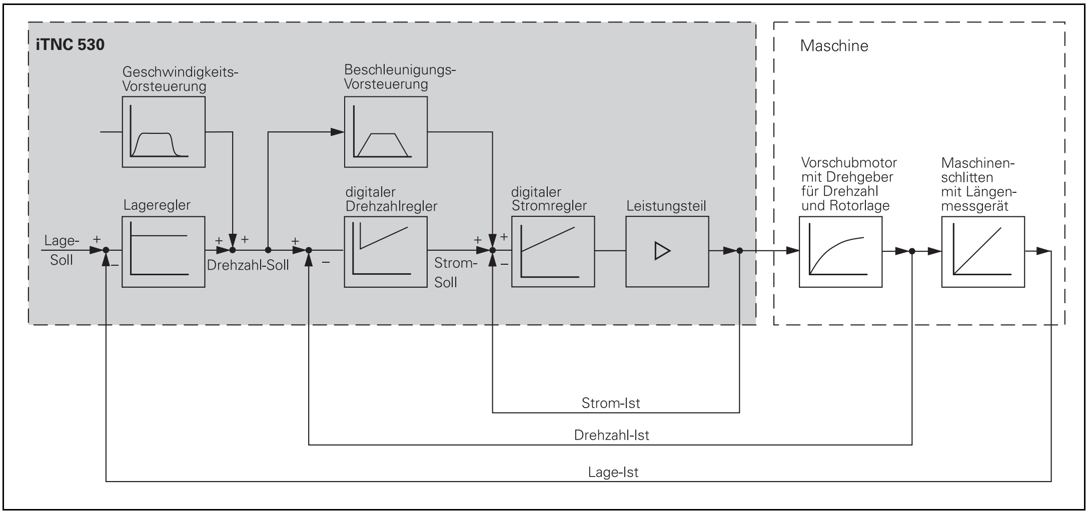
2.1 General Information

HEIDENHAIN contouring controls are designed for use with milling, drilling and boring machines as well as 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 or 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 five or eleven axes and for spindle speeds up to 40 000 rpm.

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

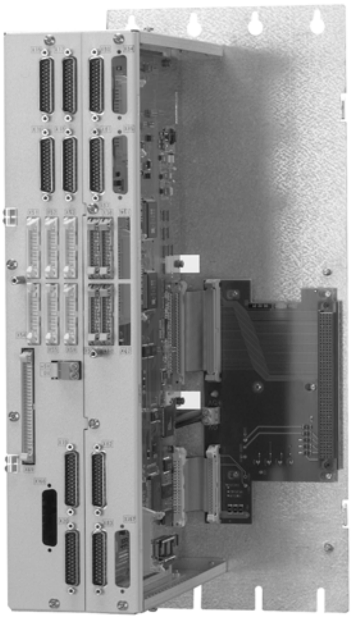
2.2 Overview of Components

2.2.1 Main Computer, Controller Unit, Power Supply Unit

The iTNC 530 comprises 2 components:

- MC 422 Main Computer (MC = Main Computer)
- CC 422 Controller Unit (CC = Controller Computer)

HEIDENHAIN drives are available together with a basic version of the MC 422 with 5 position encoder inputs and a reduced number of functions. However, these functions can be activated with two code numbers. See page 2 – 29 for the performance range.



CC 422/10 control loops

MC 422/10 position encoder inputs

Main computer (standard version)	Signal inputs	Id. Nr. of MC for BF 120 display unit	Id. Nr. of MC for BF 150 display unit
MC 422			
5 position encoder inputs	Position: 1 V_{PP}/EnDat	359 629-0x	359 630-0x
10 position encoder inputs		359 632-0x	359 633-0x
MC 422 (basic version)			
5 position encoder inputs	Position: 1 V_{PP}/EnDat	367 224-0x	367 225-0x
MC 422 (with Windows 2000)			
5 position encoder inputs	Position: 1 V_{PP}/EnDat	–	372 037-0x
10 position encoder inputs		–	369 717-0x

Main computer (export version)	Signal inputs	Id. Nr. of MC for BF 120 display unit	Id. Nr. of MC for BF 150 display unit
MC 422 E			
5 position encoder inputs	Position: 1 V_{PP}/EnDat	359 629-5x	359 630-5x
10 position encoder inputs		359 632-5x	359 633-5x
MC 422 E (basic version)			
5 position encoder inputs	Position: 1 V_{PP}/EnDat	367 224-5x	367 225-5x
MC 422 E (with Windows 2000)			
5 position encoder inputs	Position: 1 V_{PP}/EnDat	–	372 037-5x
10 position encoder inputs		–	369 717-5x

Option	Id. Nr.
Software option 1	367 591-01
Software option 2	367 590-01

SIK	Id. Nr.
SIK for standard software (basic version)	372 171-01
SIK for export software (basic version)	372 171-51

Controller unit	Signal inputs	Enabled control loops	Possible analog control loops	Id. Nr. of CC 422
CC 422				
Max. 6 digital speed control loops	1 V_{PP}/ EnDat	4	6 additional	359 651-xx
Max. 10 digital speed control loops		7	5 additional	359 652-xx
Max. 12 digital speed control loops		7	3 additional	359 653-xx

In addition to the digital control loops of the CC 422, analog control loops can also be used (as of NC software 340 420-04). These must also be enabled.

Option	Id. Nr.
1st additional control loop	354 540-01
2nd additional control loop	353 904-01
3rd additional control loop	353 905-01
4th additional control loop	367 867-01
5th additional control loop	367 868-01
6th additional control loop	370 291-01
7th additional control loop	370 292-01
8th additional control loop	370 293-01

Designation of MC 422 and CC 422

ID numbers of MC 422 and CC 422

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

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)

UV 105 Power Supply Unit

The UV 105 serves to supply the power to the CC 422 if a non-HEIDENHAIN inverter is used, or, if required, to supply additional power if a HEIDENHAIN inverter is used. See “Power Supply for the iTNC 530” on page 3 – 12.

When using a CC 422 in connection with the MC 422 (with Windows 2000), the UV 105 is absolutely mandatory.

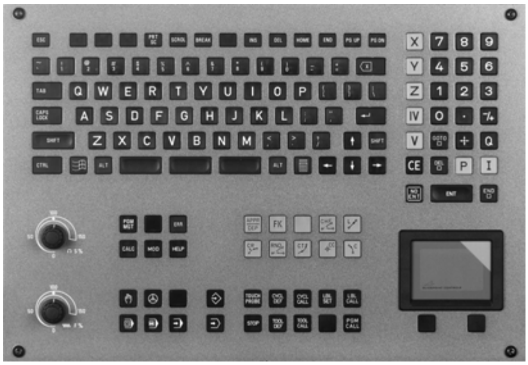
If a non-HEIDENHAIN inverter system is used, the adapter connector is connected to X69 of the UV 105.

The cover for the UV 105 and the adapter connector for X69 are included in the items supplied.

Id. Nr. 344 980-xx	UV 105
Id. Nr. 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

2.2.2 Monitors and Keyboard Units

TE 420 Operating Panel The IV and V keys are snap-ons, and can be switched. See “Key Symbols” on page 2 – 13. Id. Nr. 313 038-11 Horizontal rows to match the design of the flat-panel display Id. Nr. 316 343-01	
TE 530 Operating Panel with Touchpad The IV and V keys are snap-ons, and can be switched. See “Key Symbols” on page 2 – 13. Id. Nr. 359 906-01 Horizontal rows to match the design of the flat-panel display Id. Nr. 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. Nr. 353 522-02

Horizontal rows depending on design
Id. Nr. 339 516-02 (at bottom)
Id. Nr. 339 516-04 (at top)

BF 120 Visual Display Unit

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. Nr. 313 506-02

BTS 1x0 Screen-Keyboard Switching Unit

With the BTS 1x0, it is possible to connect two monitors and two operating panels to an MC 422.

Id. Nr. 353 544-01	BTS 150 (2 x BF 150)
Id. Nr. 329 965-02	BTS 120 (2 x BF 120)

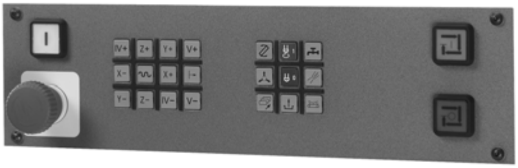
MB 420 Machine Operating Panel

Machine operating panel with snap-on (switchable) keys. See “Key Symbols” on page 2 – 13.

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. Nr. 283 757-33	Standard assignment
Id. Nr. 293 757-45	Machine key assignment for HEIDENHAIN basic PLC program



2.2.3 Handwheels

HR 410 Handwheel

Portable handwheel with snap-on (switchable) keys. See “Key Symbols” on page 2 – 13.

Key assignment:

- Keys for selection of 5 axes
- Keys for traverse direction
- Keys for preset feeds
- Key for actual value position capture
- Three keys for machine functions (definable via PLC)
  - Spindle right, Spindle left, Spindle stop
  - NC start, NC stop, Spindle start (corresponds to the HEIDENHAIN basic PLC program)
- Two permissive buttons
- Emergency stop
- Magnetic holding pads

Id. Nr. 296 469-44	HR 410 handwheel (Spindle right, Spindle left, Spindle stop)
Id. Nr. 296 469-45	HR 410 handwheel (NC start, NC stop, Spindle start)
Id. Nr. 312 879-01	Connecting cable for cable adapter (spiral cable 3 m)
Id. Nr. 296 467-xx	Connecting cable for cable adapter (normal cable)
Id. Nr. 296 687-xx	Connecting cable for cable adapter (metal armor)
Id. Nr. 296 466-xx	Adapter cable to MC 422
Id. Nr. 281 429-xx	Extension to adapter cable
Id. Nr. 271 958-03	Dummy plug for emergency-stop circuit

HR 130 Handwheel

Panel-mounted handwheel

Id. Nr. 254 040-05	With ergonomic knob, radial cable outlet

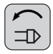
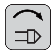
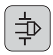
HRA 110 Handwheel Adapter

For connecting up to three **HR 150** handwheels with the TNC.
The axes and the subdivision factor are selected via rotary switch.

Id. Nr. 261 097-03	HRA 110
Id. Nr. 257 061-09	HR 150 with radial cable outlet
Id. Nr. 270 908-01	Handwheel and subdivision selection switch

2.2.4 Key Symbols

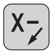
Key symbols for the spindle

Key	Designation Print/Background Id. Nr.	Key	Designation Print/Background Id. Nr.
	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 axis Black/Gray 330 816-48		

Key symbols with axis designations

Key	Designation Print/Background Id. Nr.	Key	Designation Print/Background Id. Nr.
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

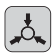
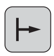
Axis direction keys for the principle axes

Key	Designation Print/Background Id. Nr.	Key	Designation Print/Background Id. Nr.
	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

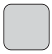
Key symbols for axis-direction keys for rotary and secondary linear axes

Key	Designation Print/Background Id. Nr.	Key	Designation Print/Background Id. Nr.
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	Designation Print/Background Id. Nr.		Key	Designation Print/Background Id. Nr.
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	Designation Print/Background Id. Nr.		Key	Designation Print/Background Id. Nr.
	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 key White/Green 330 816-22
	Permissive key 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			

2.2.5 Touch Probe Systems

TT 130 Tool Touch Probe

Touch-trigger probe for measuring tools.

Id. Nr. 296 537-xx	TT 130
Id. Nr. 335 332-xx	Adapter cable for connection to the MC 422

TS 220 Touch Probe

Touch-trigger probe with cable connection for workpiece setup and measurement during machining.

Id. Nr. 293 488-xx	TS 220
Id. Nr. 274 543-xx	Adapter cable for connection to the MC 422

TS 632 Touch Probe

Touch-trigger probe with infrared transmission, for workpiece setup and measurement during machining.

Id. Nr. 331 397-xx	TS 632
Id. Nr. 346 322-xx	EA 632 receiver unit
Id. Nr. 346 323-xx	EA 652 receiver unit
Id. Nr. 354 656-xx	APE 652 interface electronics for connecting two EA 652 to the MC 422
Id. Nr. 310 197-xx	Adapter cable for connecting the EA 632 or the APE 511 to the MC 422

2.2.6 Other Accessories

PL 410 B PLC Input/Output Unit
For the expansion of PLC inputs and outputs

Id. Nr. 263 371-12	64 inputs 31 outputs
Id. Nr. 263 371-02	64 inputs 31 outputs 4 analog inputs ± 10 V 4 inputs for Pt 100 thermistors

PL 405 B PLC Input/Output Unit

Id. Nr. 263 371-22	32 inputs 15 outputs

Further components	Id. Nr.
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 422	317 505-05
11 μA_{PP} /1 V_{PP}	313 119-01

2.2.7 Documentation

Items supplied with the control include:

- 1 User’s Manual for conversational programming
- 1 User’s Manual for ISO programming
- 1 Pilot (brief user's programming guide)

The HEIDENHAIN inverters and motors for the iTNC 530 are described in the Technical Manual “Inverter Systems and Motors.”

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

Technical Data	iTNC 530
MC 422	
	■ Processor: AMD K6/2 with 266 MHz ■ 64 MB SDRAM ■ 5 or 10 position encoder inputs 1 V_{PP} or EnDat ■ Unambiguous identification of MC 422 through SIK (**S**ystem **I**dentification **K**ey)
MC 422 (with Windows 2000)	
	■ Two processors: AMD K6/2 with 266 MHz ■ 64 MB SDRAM for the control ■ 128 MB SDRAM for Windows 2000 ■ 2 x USB ■ 2 x PS/2 ■ 5 or 10 position encoder inputs 1 V_{PP} or EnDat ■ Unambiguous identification of MC 422 through SIK (**S**ystem **I**dentification **K**ey)
CC 422	
	All speed encoder inputs 1 V_{PP} or EnDat
	■ 6 speed encoder inputs with 1 V_{PP} or EnDat for axes and spindles Basic version: 4 speed control loops ■ 10 speed encoder inputs with 1 V_{PP} or EnDat for axes and spindles Basic version: 7 speed control loops ■ 12 speed encoder inputs with 1 V_{PP} or EnDat for axes and spindles Basic version: 7 speed control loops
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 gigabytes
Input resolution and display step	
	Up to 0.1 µm for linear axes Up to 0.0001° for angular axes

Technical Data		iTNC 530
Interpolation		
	Straight lines	5 of 9 axes
	Straight lines (with MC 422 E[a])	4 of 9 axes
	Circle	■ 2 of 9 axes ■ 3 of 9 axes with tilted working plane
	Helix	Superimposition of circular and linear paths
	Spline (software option 2)	Cubic splines can be executed
Block processing time		
		0.5 ms Basic version: 3.6 ms (0.5 ms with software option 2)

a. Export version

Machine Integration	iTNC 530
Feedback control	
Position loop resolution	$\frac{\text{Signal period}}{1024}$
Cycle time, position controller	1.8 ms
Cycle time, speed controller	600 µs
Cycle time, 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
Feed rate	Maximum feed rate: $\frac{60000}{\text{No. of pole pairs}} \bullet \text{Ballscrew pitch}$ ■ Up to approx. 40 m/min (33 kHz) or approx. 420 m/min (350 kHz) for encoders with 20 µm grating period ■ Up to approx. 200 m/min (33 kHz) or approx. 2100 m/min (350 kHz) for encoders with 100 µm grating period Minimum feed rate: $\frac{(600\ \mu s \bullet 10^{-6})^{-1} \bullet 60}{\text{Line count} \bullet 1024} \bullet \text{Ballscrew pitch}$
Rotational speed	Maximum revolutions per minute: CC 422/6: $\frac{60\,000\ \text{rpm}}{\text{No. of pole pairs}}$ +5 control loops CC 422/6: $\frac{80\,000\ \text{rpm}}{\text{No. of pole pairs}}$ +3 control loops CC 422/10: $\frac{80\,000\ \text{rpm}}{\text{No. of pole pairs}}$ +9 control loops CC 422/12: $\frac{60\,000\ \text{rpm}}{\text{No. of pole pairs}}$ +11 control loops CC 422/12: $\frac{80\,000\ \text{rpm}}{\text{No. of pole pairs}}$ +9 control loops Maximum speed depending on the PWM frequency: $\frac{80\,000\ \text{rpm}}{\text{No. of pole pairs}} \cdot \frac{\text{PWM frequency}}{5000\ \text{Hz}}$ or $\frac{60\,000\ \text{rpm}}{\text{No. of pole pairs}} \cdot \frac{\text{PWM frequency}}{5000\ \text{Hz}}$

Machine Integration	iTNC 530
Error compensation	
	■ Linear and nonlinear axis error ■ Backlash ■ Reversal spikes during circular movements ■ Reversal error ■ Thermal expansion ■ Stiction ■ 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 ■ Cycle time of the 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
Main memory (RAM)	512 KB
PLC cycle time	10.8 ms (can be set)
PLC inputs 24 Vdc	56 (additional inputs as option)
PLC outputs 24 Vdc	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 Integration	iTNC 530
Commissioning aids	
	■ Oscilloscope ■ Trace function ■ Table function ■ Logic diagram ■ Log ■ PC software TNCopt
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 100 BaseT
Permissible temperature range	Operation: 0 to +40 °C (+32 °F to +113°F) Storage: –35 °C to +65 °C (–31 °F to +149 °F)

User functions

User functions	iTNC 530
Program entry	HEIDENHAIN conversational and ISO
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 face milling plane and oblique surfaces ■ Cycles for milling linear and circular slots ■ Hole patterns on circle and line ■ Contour pockets — also contour parallel ■ Contour train ■ OEM cycles (special cycles developed by the machine tool builder) can also be integrated
Touch probe cycles	■ Touch probe calibration ■ Compensating workpiece tilt manually and automatically ■ Setting the datum manually and automatically ■ Automatic workpiece measurement ■ Cycles for automatic tool measurement
Contour elements	■ Straight lines ■ Chamfer ■ Circular path ■ Circle center ■ Circle radius ■ Tangentially connected arc ■ Corner rounding
Contour approach and departure	■ Via straight line: tangential or perpendicular ■ Via circle
FK free contour programming	FK free contour programming in HEIDENHAIN conversational format with graphic support for workpiece drawings not dimensioned for NC
Background programming	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 ■ 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. ■ Tool perpendicular to 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 (=, =/, <, >) ■ Parentheses ■ tan α, arc sin, arc cos, arc tan, a^n, e^n, ln, log, absolute value of a number, constant π, negation, truncation before or after decimal point ■ Functions for calculating a circle
Programming aids	■ Pocket calculator ■ Context-sensitive help function for error messages ■ Graphic support for the programming of cycles ■ Comment blocks in the NC program
Position data	■ Nominal positions for straight lines and circles in Cartesian or polar coordinates ■ Absolute or incremental dimensional data ■ Display and input in mm or inches ■ Display of handwheel path during machining with handwheel superpositioning
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 editing tool data at a later date without a renewed program computation
Tool tables	Multiple tool tables with any number of tools
Cutting-data tables	For automatic calculation of spindle speed and feed rate from tool-specific data (cutting speed, feed rate per tooth)
Constant contour speed	■ With respect to the path of the tool center ■ With respect to the tool cutting edge (M109, M110, M111)
Program jumps	■ Subroutines ■ Program-section repeat ■ Any desired 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 part program
Test graphics	Graphical simulation before a program run, also while another program is being run ■ Plan view, view in three planes, 3-D view ■ Magnification of details
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

User functions	iTNC 530
Machining 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** mode of operation ■ Display of the current machining time in the Program Run modes of operation
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 panel-mounted HR 130 handwheel, or ■ Up to three panel-mounted HR 150 handwheels via the HRA 110 handwheel adapter
Touch probe systems	■ TS 220 triggering 3-D touch probe with cable connection, or ■ TS 632 triggering 3-D touch probe with infrared transmission, or ■ TT 130 triggering 3-D touch probe for tool measurement
Data transfer software	TNCremoNT, TNCremo
PLC development software	PLCdesign (comprises PLCdesign, TNCremo, TNCremoNT, CycleDesign and TNCopt)
Software for generating cycle structure	CycleDesign
Software for remote diagnosis	TeleService
PLC input/output unit	Up to four PL 410 B or one PL 405 B
	PL 410 B version 1: Additional 64 PLC inputs and 31 PLC outputs per PL PL 410 B 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 405 B: Additional 32 PLC inputs and 15 PLC outputs per PL

Software options

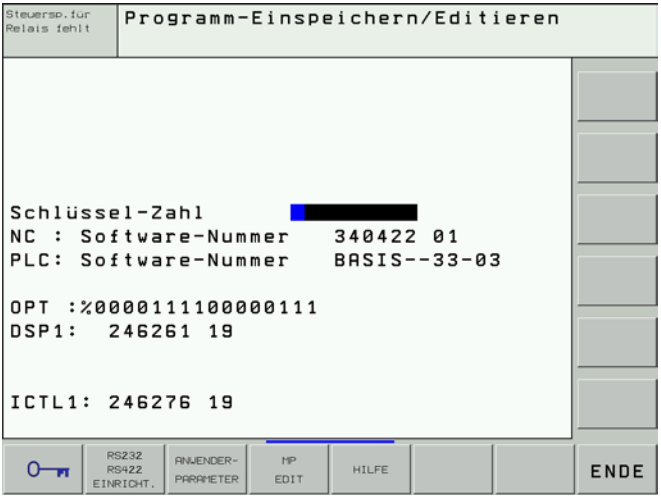
Software options	iTNC 530
Software option 1	■ Cylinder surface interpolation ■ Feed rate in mm/min ■ Tilting the working plane ■ Circular interpolation in 3 axes with tilted working plane
Software option 2	■ Particularly jerk-free path control ■ 3-D tool compensation through surface-normal vectors ■ TCPM = Tool Center Point Management ■ Tool perpendicular to contour ■ Tool radius compensation vertical to the tool direction ■ Straight-line interpolation in 5 axes (permit required for export) ■ Spline interpolation: Execution of splines (third-degree polynomials) ■ 0.5-ms block processing time

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).



Type

The iTNC 530 is shipped with the following NC software versions:

Standard	Export	Note
340 420-xx	340 421-xx	iTNC 530
340 422-xx	340 423-xx	iTNC 530
340 480-xx	340 481-xx	iTNC 530 with Windows 2000

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

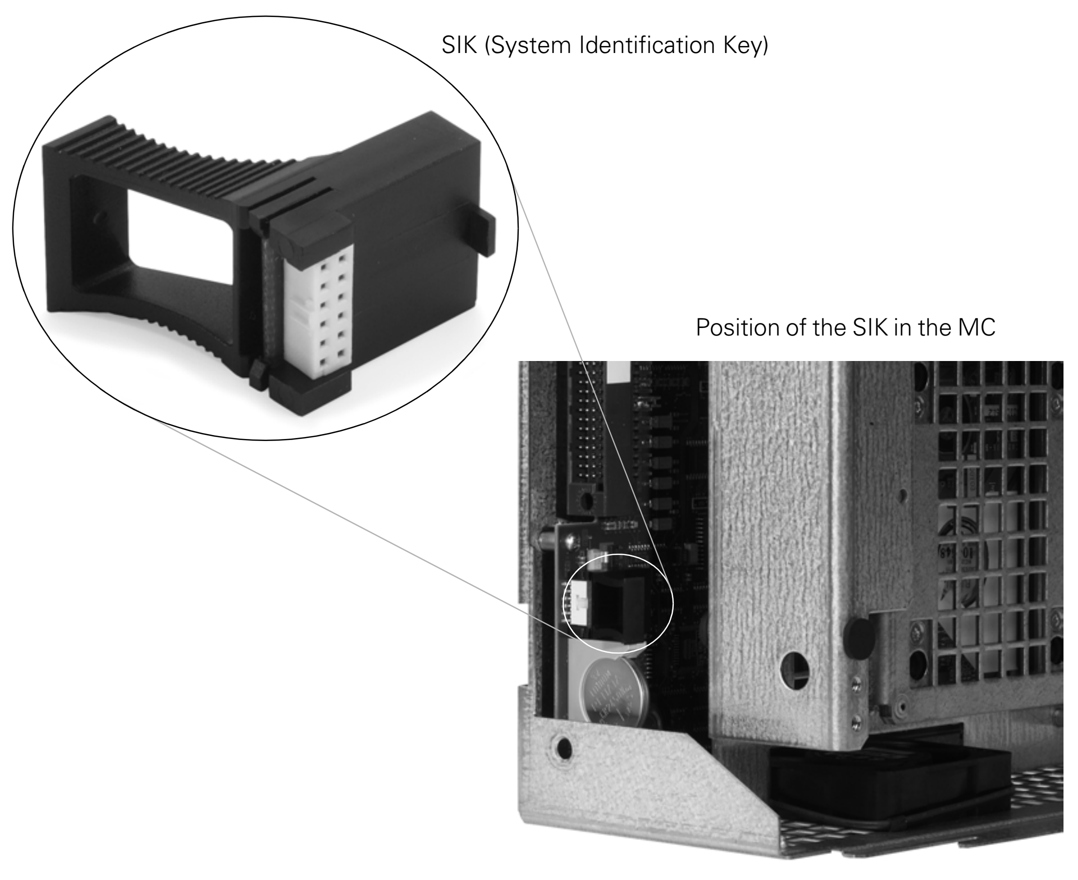
The PLC software is stored on the hard disk of the iTNC. You can order a PLC basic program directly from HEIDENHAIN. With the PLC development software **PLCdesignNT,** the PLC program can very easily be adapted to the requirements of the machine.

2.4.3 Enabling Additional Control Loops or Software Options

For each MC 422, 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).

Up to two software options can be enabled on the basic version of the MC 422. This is also done by entering a code number.

Each MC 422 can clearly be identified by the SIK (System Identification Key). You will find the SIK number on the outside of the MC 422 housing (below the ID label) and on the SIK board.



If you wish to enable additional control loops or software options, please contact HEIDENHAIN for the code number. After you have informed us of the SIK number, we can give you the required code number.

Note

If you replace the MC 422, you must also replace the SIK in order to ensure that the enabled control loops will also be enabled on the new MC 422.

To enable additional control loops, 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:

Display	Meaning
SIK ID:	SIK number
Control Type:	Control model (=iTNC 530)
General Key:	Enter the master code number 65535 to enable all options for the duration of two weeks. **NONE:** Master code number has not been entered yet. **dd.mm.yyyy:** Date up to which all options will be enabled. It is not possible to enable the control loops again by entering the master code number. **EXPIRED:** The two weeks since the master code number was entered have expired.
Option column	Description of the individual options
Active column	YES: Option is enabled NO: Option is not enabled
Keycode column	Enter the code number for enabling an option.

- Use the arrow keys to select the row containing the option to be enabled. Options that have not been enabled yet are identified by the entry **NO** in the **Active** column.
- Enter the code number for enabling the option in the **Keycode** column and confirm your entry by pressing the ENT key. HEIDENHAIN can give you the code number after having been informed of the SIK number.
- Press the END soft key. The iTNC performs a reset.
- If the code number is correct, the enabled option is identified by the entry **YES** in the **Active** column.
- After enabling all required options, reset the control.

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.

Displaying the status of an option

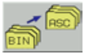
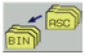
The status of an option (enabled or not) can be displayed either as a menu by entering the code number SIK (see previous page), or bit-encoded after pressing the MOD key in the OPT line.

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

2.4.4 NC Software Exchange

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 SAMPLE FILE	Copy cutting-data tables, tables for tilting-axis geometry, and tables of M-function macros from the SYS partition into the corresponding directories of the TNC or PLC partition.
	Activate or delete existing NC software.
	Exchange the NC software.

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.

Information about the cycles

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

Before exchanging the NC software, ensure that the free space on the hard disk of the MC 422 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 2 – 39.)

- 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 MOUNT DEVICE soft key.
  - To exit the list of network drives, press the END soft key.
  - To exit the program management, 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.

Equivalent file name extensions in binary and ASCII 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 carries out a reset. With the END key, the NO soft key or the END soft key, you 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 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 tables for tilting-axis geometry, and the table of M-function macros 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 saved 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 Id. Nr., see page 2 – 3); for the location of the SIK in the MC 422, See "Enabling Additional Control Loops or Software Options" on page 2 – 31.
- 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 informing HEIDENHAIN of the SIK number, HEIDENHAIN can give you the code number for enabling the functions.

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 MC 422 (E).

Warning

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 2 – 39.

Activating existing NC software

Before activating existing NC software, ensure that the free space on the hard disk of the MC 422 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 binary and ASCII format					
.H	.H%	.I	.I%	.T	.T%
.TCH	.TC%	.D	.D%	.P	.P%
.PNT	.PN%	.COM	.CO%	.CMA	.CM%

- Press the NCVer 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. With the END key, the NO soft key or the END soft key, you 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 reconvert the files on the hard disk from ASCII format into 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 tables for tilting-axis geometry, and the table of M-function macros 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 NCVer 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. With the NO soft key or the END soft key, you 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" on page 2 – 35).
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 log 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 number
- 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 space 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

2.4.5 Data Backup

HEIDENHAIN provides a data backup program called TNCBACK.EXE free of charge.

HEIDENHAIN recommends that the machine manufacturer 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.

2.5 Software Releases

2.5.1 NC Software 320 420-xx

NC software 320 420-01 (export 340 421-01)

Release: 10/2001

Initial version

NC software 320 420-02 (export 340 421-02)

Release: 12/2001

- MP1086.x: Maximum permissible jerk during single-axis movements at rapid traverse
- MP7365.x: Color settings for the oscilloscope
- The path acceleration is calculated from the axis proportions.
- New signals in the oscilloscope (I^2t monitoring of the motor and power module, utilization of the motor, position difference for gantry axes)
- 6 channels in the oscilloscope
- Reference the signals of the oscilloscope to the datum line and adjust
- Starting and ending times for the log
- FAILTEST code number for testing an internal EMERGENCY STOP
- **Power Fail Interrupt** is entered if POWERFAIL is used to switch off
- Error message **No measured value saved <axis>** if no value is saved during probing
- Expanded ranges for PLC operands (M0 to M9999, B0 to B9999, T0 to T999, C0 to C143)
- Write data to the PLC partition with **FN15: PRINT** and **FN16: F-PRINT**
- Connect Windows computer as network drives
- Entry of an ASCII file (with IP addresses and computer names) in the column **DOMAIN** of the network settings
- Negative spindle angle in Q336 for Cycles 202, 204 and 209
- New, expanded search function
- At startup, inspect the file system in case the control was not shut down properly

NC software 320 420-03 (export 340 421-03)

Release: 01/2002

- MP4040, MP4041, MP4042: Set PLC output after shutting down the control
- The error message **<Number> cannot be displayed** is displayed if more than 4 channels from the current/speed controller are to be displayed
- Display of progress during field orientation
- Display of progress while the log file is created with the code number **LOGBOOK**
- The results of evaluation of the soft-key resource file are saved in the ASCII file called **<Name of the soft-key resource file>.SYS.LOG**
- When the pocket table **TOOL_P.TCH** is reset, the data is taken over from the file **PLC:\PROTO\PROTOTYP.TCH** if it exists.
- Module 9279: Mode 2: Control shut down depending on MP4040, MP4041 and MP4042
- A, B and C axes as helical axes

NC software 320 420-04 (export 340 421-04)

Release: 03/2002

NC software 320 420-05 (export 340 421-05)

Release: 05/2002

- MP1011: Limit of rapid traverse on the path
- MP1061: Limitation of the path acceleration
- MP1146: Permissible difference between the position at shutdown and the position read in via the EnDat interface
- MP1355: Double reference run
- MP1356.x: Distance between speed and position encoder for double reference run
- MP2202.x: Overwrite **Line count of the encoder** in the motor table
- MP2204.x: Overwrite **Counting direction** in the motor table
- MP2206.x: Overwrite **Type of encoder** in the motor table
- MP7370.x: Color settings of the small PLC window
- MP7691: Size of a log file with messages from the operating system
- New code number KINEMATIC, for choosing a kinematic from a selection list. The name from the new column DOC in the assignment table is used.
- Message window before simulating an internal emergency stop via the code number FAILTEST
- Soft key **RESTORE SCREEN** in the oscilloscope, in order to read in *.DTA files
- **FN18: SYSREAD ID56** to receive file information
- Module 9035: New status information 28 (tool or pocket table in editing mode)
- Module 9163: New error codes W1022 = 1 and W1022 = 2
- Support graphics for cycles for the BF 150 revised
- Maximum input value is 3.2767 for the column **LBREAK** in the tool table
- After pressing MOD, the Id. Nr. of the setup is no longer displayed
- Cycle 403: Q337 (Set to zero after alignment)
- Status window with program section repeats and subprograms
- Zoom function for 3-D graphics in the Program Test mode
- Status of the soft key **Machining time ON/OFF** remains in effect after a power interruption
- Message window if the NC software in the SIK does not match the NC software being used
- Two-line display of NC error messages with more then 32 characters

NC software 320 420-06 (export 340 421-06)

Release: 07/2002

- MP110.x, MP111.x: Error message for position encoder inputs that do not exist
- MP960.x: Input range expanded to +/– 1.79769313486E+308
- MP2160.x: Axis-specific and 2 as new input value for HEIDENHAIN EcoDyn synchronous motors
- MP2195: Suppress error messages of the HEIDENHAIN supply units
- MP2220: Monitoring of the direction of rotation for synchronous motors cannot be switched off. Bit 3: Suppress vibrations when switching off the drive with applied motor brakes.
- MP2304.x: Reference value for I^2t monitoring of the power module
- MP2308.x: Time between the braking signal and switch-off of the controller
- MP7263 bit 1: Output of the columns in the pocket table for file functions
- MP7357.x: Colors for inactive and active soft-key row in the Machine operating mode
- MP7358.x: Colors for inactive and active soft-key row in the Programming operating mode
- MP7370.15: Color 15 of the small PLC window
- MP7481.x: Sequence for new and returned tool when changing tools
- MP7482: Magazine with variable or fixed pocket coding
- MP7684 bit 7: Reserved
- Switch-on of the current controller delayed by 50 ms after switching on the controller (Module 9161)
- Entry **MACEND** (end of an NC macro) in the log file
- Error message and limitation to 999 error messages in the *.PET table
- Screen switchover key active after M or S function has been started
- M4185: Internal stop performed
- New entries for the soft-key resource file *.SPJ: **VROOT, HROOT, EMODE, MMOD, ENABLE, STATUS, POPUPMENU, CLOSEPOPUPMENU, LARGEWINDOW, SMALLWINDOW, CLOSEPLCWINDOW, FirstInGroup**
- WATCH LIST in the PLC main menu
- **FN18: SYSREAD ID52 NR2 IDX<tool number>** finds the corresponding tool magazine.
- Module 9136: Switching the touch probe on/off
- Module 9148: Use nominal value as actual value
- Module 9157: Status information 4 (spindle in operating mode 0 or 1)
- Module 9321: Ascertain current block number
- Module 9305, 9306: Error code 6
- Module 9342: Find magazine and pocket number
- Error message when a **Tolerance for rotary axes** is programmed when the HSC filter is not active
- Display *.A ASCII files in the **Save machine parameters** mode of operation
- **M140 MB<Retraction value> F<Feed rate>** with optimized feed rate
- Network ping in the network settings

- Message window if the NC software in the SIK does not match the NC software being used. In this case it can only be operated as a programming station. After changing the NC software from the standard to the export version, compressed files can automatically be deleted
- Display of progress when loading new controllers for hardware components
- New NC block **`CYCLE CALL POS`**

NC software 320 420-07 (export 340 421-07)

Release: 08/2002

2.5.2 NC Software 340 422-xx

NC software 320 422-01 (export 340 423-01)

Release: 08/2002

Initial version

Expansions since NC software 340 420-07:

- Presets are managed with the preset table TNC:\PRESET.PR. Presets are recalculated using the defined tilting-axis geometry.
- MP7294: Disable axis-specific datum setting in the preset table
- New prototypes for pallet tables (PROTOPR.P and PROTO_TOPR.P) with the column **PRESET**
- Cycle structure of the machining structures revised.
- Error message **Use preset table!** appears when MP7475 = 1 and Cycle 7 is programmed.
- Cycle 205: Parameter Q379 (**START POINT**)
- Cycle 220: Parameter Q365 (**TYPE OF TRAVERSE**)
- Cycle 247: Entry of the preset number from the preset table
- Cycles 400, 401, 402: Parameter Q305 (**NO. IN TABLE**)
- Cycles 410 to 418: Parameter Q303 (**MEAS. VALUE TRANSFER**)
- Cycles 414, 415: Parameter Q305 (**NO. IN TABLE**) for datum and basic rotation
- Cycle 419: **DATUM IN ONE AXIS**
- Cycles 420 to 430: Take active rotation into account

3 Mounting and Electrical Installation

3 Mounting and Electrical Installation

3.1 General Information

Warning

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 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

3.1.2 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

Noise 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 422, CC 422 and its leads to interfering equipment.
- Keep a minimum distance of 10 cm from the MC 422, CC 422 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 EN 50 178
- Use potential compensating lines with 6 mm^2 cross-sections
- Use only genuine HEIDENHAIN cables, connectors and couplings

3.2 Shipping Brace of the Hard Disk

The hard disks of the MC 422 are fitted with a shipping brace. The shipping brace consists of two screws, which are used to secure the hard disk to the housing of the MC 422. There are also two additional free threaded holes in the housing, into which the screws are inserted after the hard disk has been released.

When is it necessary to use the shipping brace?

- Before beginning operation, the shipping brace of the hard disk must be removed and the screws inserted in the storage holes.
- The shipping brace for the hard disk is not required when the machine is being transported.
- Should servicing become necessary (i.e. the MC 422 is being shipped on its own), the hard disk must be secured with the shipping brace.

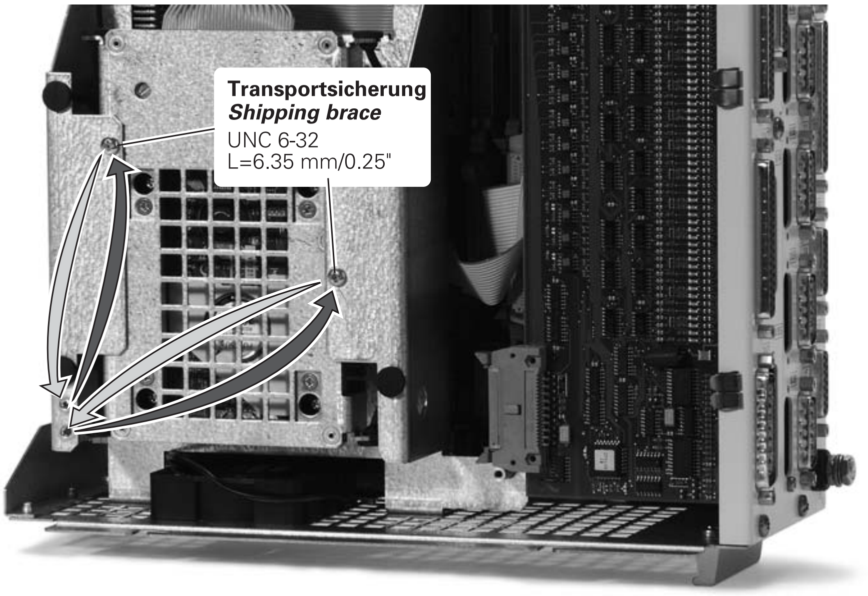
Achtung!
Note!

Vor Inbetriebnahme Schrauben für die **Transportsicherung** der Festplatte entfernen und in die freien Gewindebohrungen einschrauben.

*Before beginning operation, release the **shipping brace** of the hard disk by removing the screws from the locking holes and screwing them into the storage holes.*

Im Servicefall Festplatte vor dem Transport mit den Schrauben sichern.

***Before shipping the unit**, secure the hard disk by reinserting the screws.*



3.3 Environmental Conditions

3.3.1 Heat Generation and Cooling

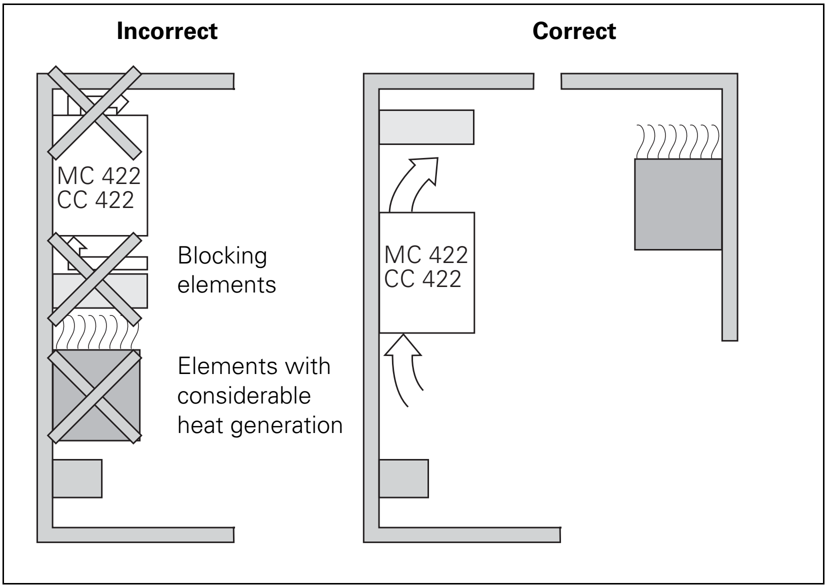
The permissible ambient temperature in operation is between 0 °C and 40 °C. Any deviation from this will impair the operating safety of the machine. The permissible storage temperature is between –35 °C and +65 °C.

The following measures can ensure adequate heat removal:

- Provide sufficient space for air circulation.
- Build in a fan to intensify the natural convection. The fan should extract the warm air from the MC 422 and CC 422. There must be no pre-warmed air blown into the unit. The warmed air should flow over surfaces that have good thermal conductivity to the external surroundings.
- For a closed steel housing without assisted cooling,the proposed value for heat conduction is 3 watt/m^2 of surface per °C air temperature difference between inside and outside.

- Use of a heat exchanger with separate internal and external circulation.

HEIDENHAIN advises against blowing external air through the control cabinet to replace the internal air. Electronic assemblies may be adversely affected by fine dust or vapors. If no other method of cooling is possible, then ensure that the fan extracts the warmed air from the electrical cabinet and that the air drawn in is adequately filtered. Regular servicing of the filter is essential.



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 Mechanical Vibration

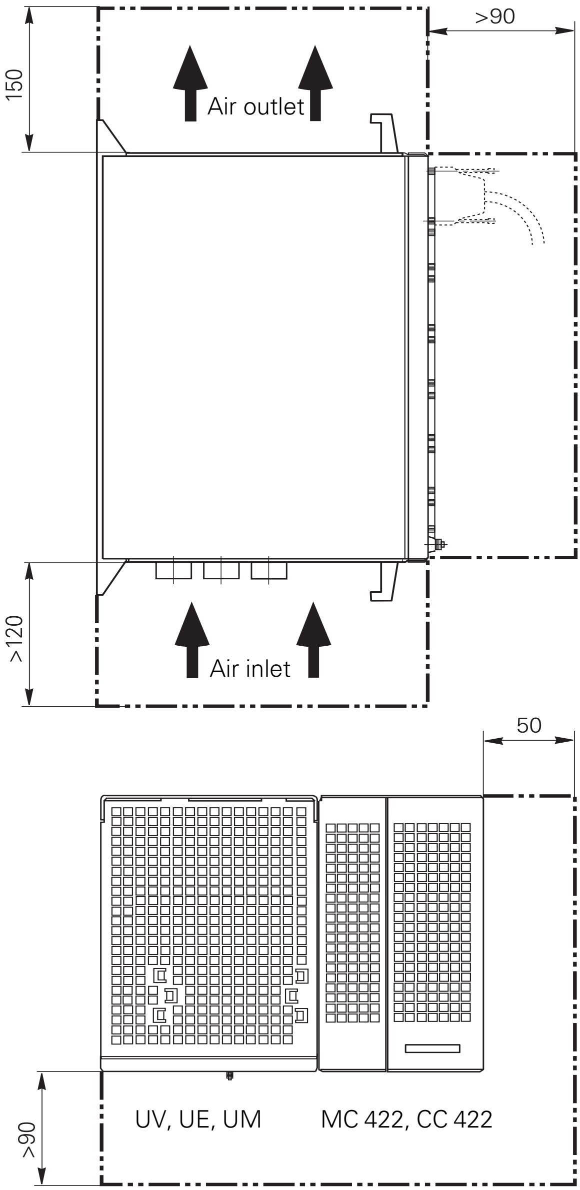
Permissible vibration: ± 0.075 mm, 10 to 41 Hz
5 m/s^2, 41 Hz to 500 Hz

Permissible shock: 50 m/s^2, 11 ms

Permissible shock with shipping brace for hard disk: 300 m/s^2, 11 ms

3.3.4 Mounting Attitude of MC 422, CC 422, UV xxx, UM xxx, UE 2xx B

> **Warning**
>
> 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 21x with integral braking resistor; Do not mount any temperature-sensitive

Leave space for servicing!

Leave space for air circulation and servicing!

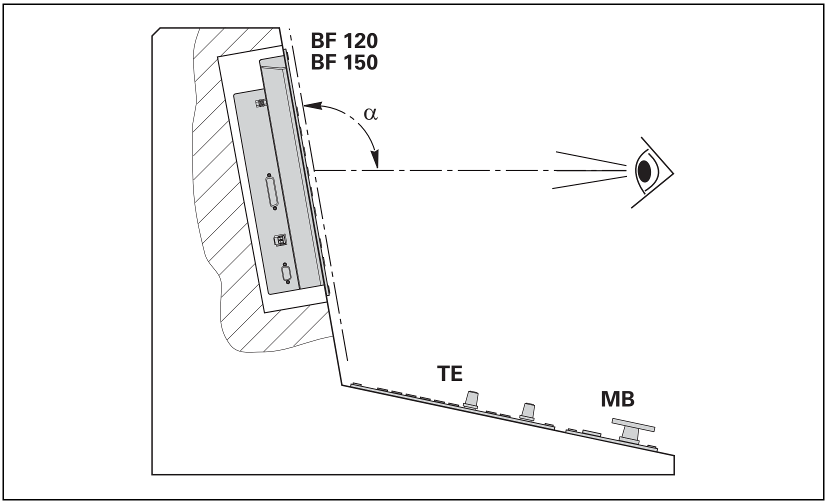
Leave space for servicing and connecting cable!

3.3.5 Mounting Position of Screen

BF 120, BF 150

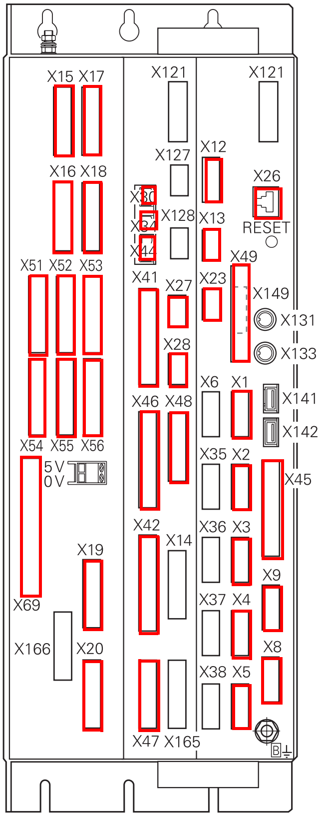
The BF 120 and BF 150 flat-panel displays must be viewed with a slight backward slant.

▶ During installation, ensure a viewing angle of 150° > α > 90°.



3.4 Connection Overview for iTNC 530

MC 422 M/5 position encoder inputs and CC 422 with 6 speed control loops



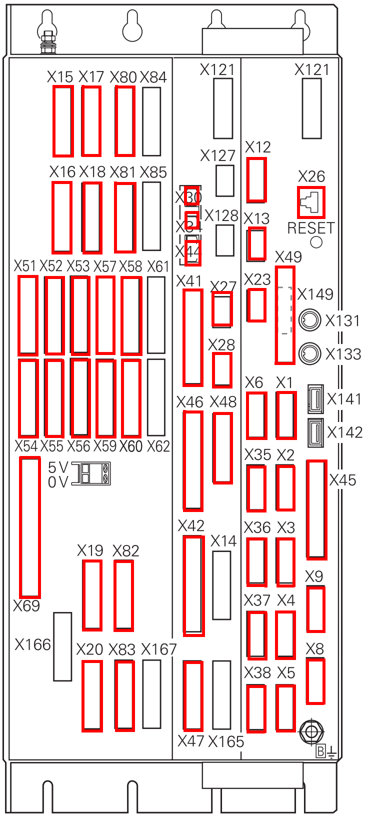
X150 at bottom of housing

X1 to X5	Position encoder 1 V_{PP}
X35 to X38	Vacant
X15 to X20	Speed encoder 1 V_{PP}
X51 to X60	PWM output
X8, X9	Nominal value output, analog
X12	TS touch trigger probe
X13	TT 130 touch trigger probe
X23	Handwheel
X26	Ethernet data interface
X27	RS-232-C/V.24 data interface
X28	RS-422/V.11 data interface
X127	RS-232-C/V.24 (only for Windows 2000)
X128	RS-422/V.11 (only for Windows 2000)
X141, X142	USB interface
X30	24 V reference signal for spindle
X34	24 V for control-is-ready signal output
X41	PLC output
X42	PLC input
X44	24 V PLC supply voltage
X45	Keyboard unit (TE 4xx)
X46	Machine operating panel
X47	PLC expansion (PL 4xxB)
X48	PLC analog input
X149 (X49)	BF 150 (BF 120) visual display unit
X131, X133	PS/2 interface
X69	Power supply
X121, X125	Reserved
X165, X166	Reserved
5V/0V	Power supply for processor
X150	Axis-specific drive release 1 to 6
B	Signal ground
⏚	Equipment ground (YL/GN)

Warning

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

MC 422 M/10 position encoder inputs and CC 422 with 10 or 12 speed control loops



X150, X151 at bottom of housing

X1 to X6	Position encoder 1 V_{PP}
X35 to X38	Position encoder 1 V_{PP}
X15 to X20	Speed encoder 1 V_{PP}
X80 to X83	Speed encoder 1 V_{PP}
X84, X85	Speed encoder 1 V_{PP} (12 control loops)
X51 to X60	PWM output
X61, X62	PWM output (12 control loops)
X8, X9	Nominal value output, analog
X12	TS touch trigger probe
X13	TT 130 touch trigger probe
X23	Handwheel
X26	Ethernet data interface
X27	RS-232-C/V.24 data interface
X28	RS-422/V.11 data interface
X127	RS-232-C/V.24 (only for Windows 2000)
X128	RS-422/V.11 (only for Windows 2000)
X141, X142	USB interface
X30	24 V reference signal for spindle
X34	24 V for control-is-ready signal output
X41	PLC output
X42	PLC input
X44	24 V PLC supply voltage
X45	Keyboard unit (TE 4xx)
X46	Machine operating panel
X47	PLC expansion (PL 4xxB)
X48	PLC analog input
X149 (X49)	BF 150 (BF 120) visual display unit
X131, X133	PS/2 interface
X69	Power supply
X121, X125	Reserved
X165, X166, X167	Reserved
5V/0V	Power supply for processor
X150	Axis-specific drive release 1 to 6
X151	Axis-specific drive release 7 to 10
B	Signal ground
⏚	Equipment ground (YL/GN)

Warning

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

3.5 Power Supply for the iTNC 530

The UV 1x0 or UV 105 power supply unit supplies the iTNC 530 with power. Power is supplied through X69 and, with the UV 105, in addition through a 5-V terminal on the CC 422.

For information on the UV 1x0 supply units, refer to the Technical Manual "Inverter Systems and Motors."

Whether the UV 1x0 supply unit is sufficient or whether a UV 105 is needed as additional equipment depends on the current consumption of the used units.

Device	Load capacity
UV 1x0, UE 2xx B	8.50 A
UV 105	20.00 A

Device	Current consumption of the 5-V supply
MC 422	4.80 A
MC 422 (with 2 processors)	7.80 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
LS, LB	0.15 A
ERN, ROD, RON	0.20 A
ECN, EQN, ROC, ROQ, RCN	0.25 A (+0.085 A with line drop compensator)[b]
LC	0.30 A (+0.085 A with line drop compensator)[b]

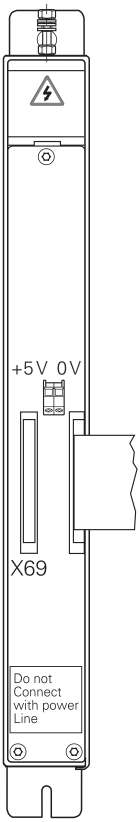
a. If USB components require more than 0.5 A, a separate power supply becomes necessary for these components. One possibility is the USB hub from HEIDENHAIN (368 735-01).
b. For cable lengths > 10 m between the logic unit and the encoders with EnDat interfaces, a line drop compensator is required (efficiency = 75 %).

Example:

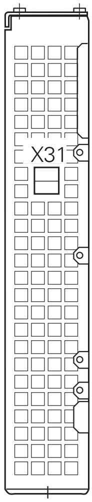
Device	Current consumption
MC 422	4.80 A
CC 422/6 control loops	1.50 A
3 x LS for X, Y, Z	0.45 A
3 x ERN for X, Y, Z	0.6 A
1 x ERN for spindle	0.2 A
2 x ROD for B, C	0.4 A
2 x ERN for B, C	0.4 A
Total	8.35 A < 8.50 A

A UV 105 unit is not needed for this application.

Connection overview UV 105



Conductor bar	Power supply over U_Z
Free cable	5 V power supply for CC 422 (to 5 V terminal on CC 422)
Free ribbon cable	Power supply for CC 422 (to X69 on CC 422)
X69	Status signals from UV 1x0



X31	Power supply

Warning

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

X69: NC supply voltage and control signals

Note

For the NC to be able to evaluate the status signals of the UV 1x0 power supply units, connector X69 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. Nr. 349 211-01) must be connected to X69 on the UV 105. This connector is delivered with the UV 105.

Pin layout:

Ribbon cnnctr., 50-pin	Assignment	Ribbon cnnctr., 50-pin	Assignment
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	ERR.ILEAK
9a	+15 V	18b	GND
9b	-15 V	19a	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	RES.PS	21b	GND
12b	0 V	22a	Do not assign
13a	PF.PS.ZK	22b	GND
13b	GND	23a	Reserved (SDA)
14a	ERR.UZ.GR	23b	GND
14b	GND	24a	Reserved (SLC)
15a	ERR.IZ.GR	24b	GND
15b	GND	25a	RES.LE
16a	ERR.TMP	25b	GND

5-V connection of the UV 105

Pin layout:

Wire color of 5-V connection	5-V terminal on the CC 422
Black	0 V
Red	+5 V

X31: Supply voltage of the UV 105

Supply voltage: 400 V ± 10 %

Pin layout:

Connecting terminal	Assignment
U	U[a]
V	V
⏚	Equipment ground (YL/GY)

a. Connecting cable: 1.5 mm^2, shielded

Note

The supply voltage at terminals U and V must:

- be supplied via an isolating transformer (300 VA, basic isolation in accordance with EN 50 178 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.

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 6 – 174.

3.6 Power Supply for PLC Outputs

The PLC of the iTNC 530 as well as the PL 410 B/PL 405 B are powered by the 24 Vdc 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.

EN 61 131-2:1994 permits:

- 5% alternating voltage component is permissible
- Minimum absolute value: 19.2 Vdc
- Maximum absolute value: 30 Vdc

Warning

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 422: 48 W
PL 410 B: Approx. 460 W
PL 405 B: Approx. 235 W

Nominal operating current per output

MC 422: 0.125 A (with a simultaneity factor of 0.5)
PL 410 B: 2 A (with max. current consumption of 20 A)
PL 405 B: 2 A (with max. current consumption of 20 A)

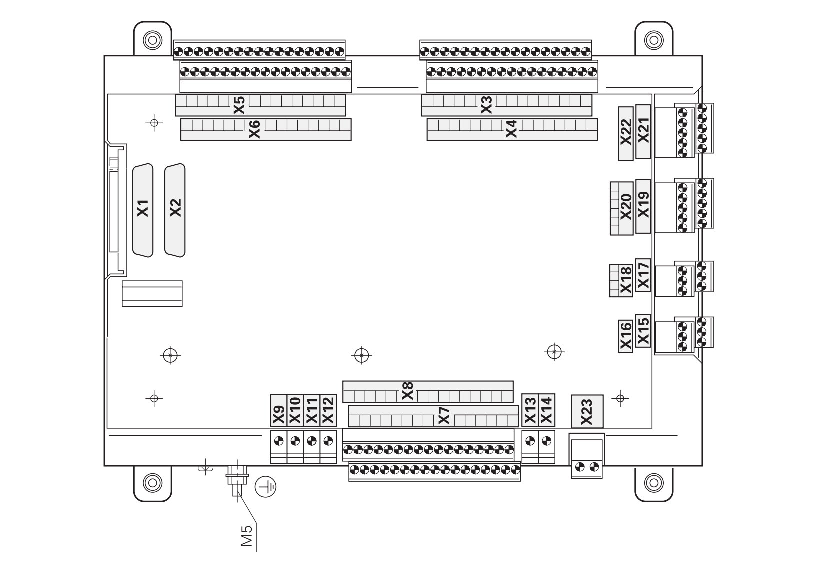
X44: PLC supply voltage

Pin layout on the MC 422:

Connection terminal	Assignment	PLC outputs
1	+24 V can be switched off via EMERGENCY STOP	O24 to O30 control-is-ready signal
2	+24 V can be switched off via EMERGENCY STOP	O16 to O23
3		O0 to O15
4	0 V	

3.6.1 Supply Voltage for PLC Outputs on the PL 4xx B

Connection overview



X9 to X14: Supply voltage

Pin layout on the PL 410 B:

Terminal	Assignment	PL 1	PL 2	PL 3	PL 4
X9	0 V				
X10	+24 Vdc logic power supply and for control-is-ready signal				
X11	+24 Vdc power supply for outputs	O32 – O39	O64 – O71	O128 – O135	O160 – O167
X12	+24 Vdc power supply for outputs	O40 – O47	O72 – O79	O136 – O143	O168 – O175
X13	+24 Vdc power supply for outputs	O48 – O55	O80 – O87	O144 – O151	O176 – O183
X14	+24 Vdc power supply for outputs	O56 – O62	O88 – O94	O152 – O158	O184 – O190

Pin layout on the PL 405 B:

Terminal	Assignment	PL 1	PL 2	PL 3	PL 4
X9	0 V				
X10	+24 Vdc logic power supply and for control-is-ready signal				
X13	+24 Vdc power supply for outputs	O48 – O55	O80 – O87	O144 – O151	O176 – O183
X14	+24 Vdc power supply for outputs	O56 – O62	O88 – O94	O152 – O158	O184 – O190

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

Terminal	Assignment
1	+24 Vdc as per EN 50 178, 5.88
2	+0 V

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 Vdc 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

Power supply with basic insulation in accordance with EN 50 178:

Connecting terminal X1	Assignment
1	+24 V
2	0 V

Power consumption: BF 120: 15 W
BF 150: 45 W

3.9 Buffer Battery

Danger

When exchanging the buffer battery, remember:

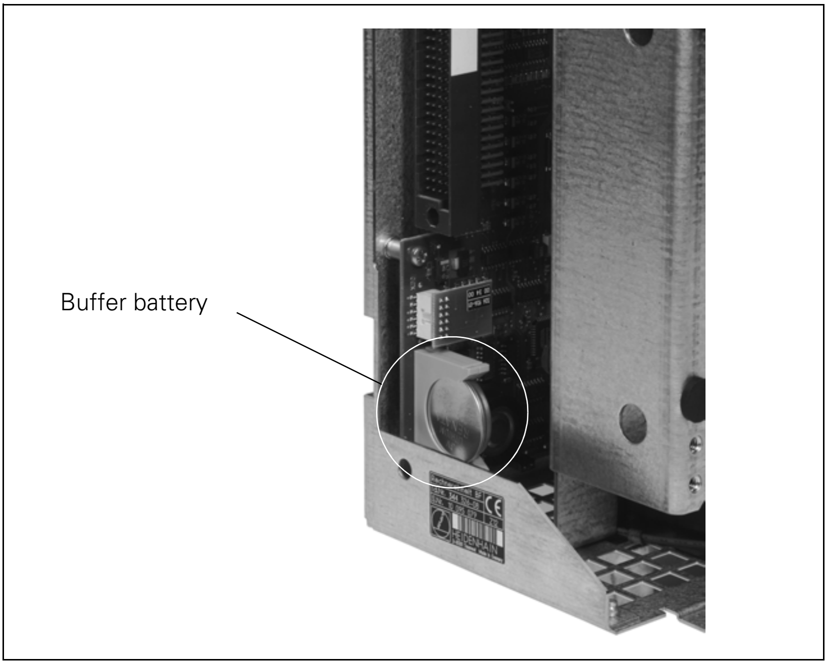
- Switch off the machine and the iTNC 530.
- The buffer battery may be exchanged only by trained personnel.

Battery type: 1 lithium battery, type CR 2450N (Renata), Id. Nr. 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 exceed 2.6 V, the error message is reactivated after 30 minutes.

To exchange the battery:

- The buffer battery is located on the rear side of the MC 422.
- Exchange the battery; the new battery can be inserted in only one position.



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 422.

- 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	Reserved, do not assign	Reserved, do not assign
5	Reserved, do not assign	Reserved, do not assign
6	Reserved, do not assign	Reserved, do not assign
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 enable

Pin layout:

D-sub connctn. (male) 37-pin	Assignment
..	..
33	+24 V (drive controller enable)
..	..

3.11 Encoder Connections

3.11.1 General Information

HEIDENHAIN contouring controls are designed for use with incremental linear and angular encoders as measuring systems. The encoder signals are interpolated 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}: 50 kHz/350 kHz (can be set with MP)	350 kHz

11 μA_{PP} encoders can be connected through the adapter plug with the Id. Nr. 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 rpm

$$x = \frac{350 \cdot 60 \cdot 1000}{10000} \approx 2048$$

3.11.2 Input of Position Encoder

X1 to X6, X35 to X38: Position encoder 1 V_{PP}

Pin layout:

MC 422		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 (U_P)	9	Blue	2	2	Blue
10	R+	10	Red	3	3	Red
11	0 V (U_P)	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.	External shield	Hsg.	External shield	Hsg.	Hsg.	External shield

Note

The interface complies with the requirements of EN 50 178 for “low voltage electrical separation.”

X1 to X6, X35 to X38: Position encoder with EnDat interface

Pin layout:

MC 422		Adapter cable 332 115-xx			Connecting cable 323 897-xx				Adapter cable 313 791-xx		
Male	Assign-ment	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 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	5a
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 line)	9	Blue	1	1	Blue	1		1	Blue	6a
10	Free	10		3	3	Red	3		3		
11	0 V (sensor line)	11	White	4	4	White	4		4	White	6b
12	Free	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.	Ext. shield	Hsg.		Ext. shield			Hsg.	Ext. shield	

Note

The interface complies with the requirements of EN 50 178 for “low voltage electrical separation.”

3.11.3 Input of Speed Encoder

Warning

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 current controllers for such applications. More information is in the Cable Overviews.

X15 to X20, X80 to X85: Speed encoder 1 V_{PP}

Pin layout:

CC 422		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	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 (U_P)	14	Blue	16	16	Blue	16
15	Do not assign						
16	0 V (U_N)	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.	External shield	Hsg.	Hsg.	External shield	Hsg.

Note

The interface complies with the requirements of EN 50 178 for “low voltage electrical separation.”

X15 to X20, X80 to X85: Speed encoder with EnDat interface

Pin layout:

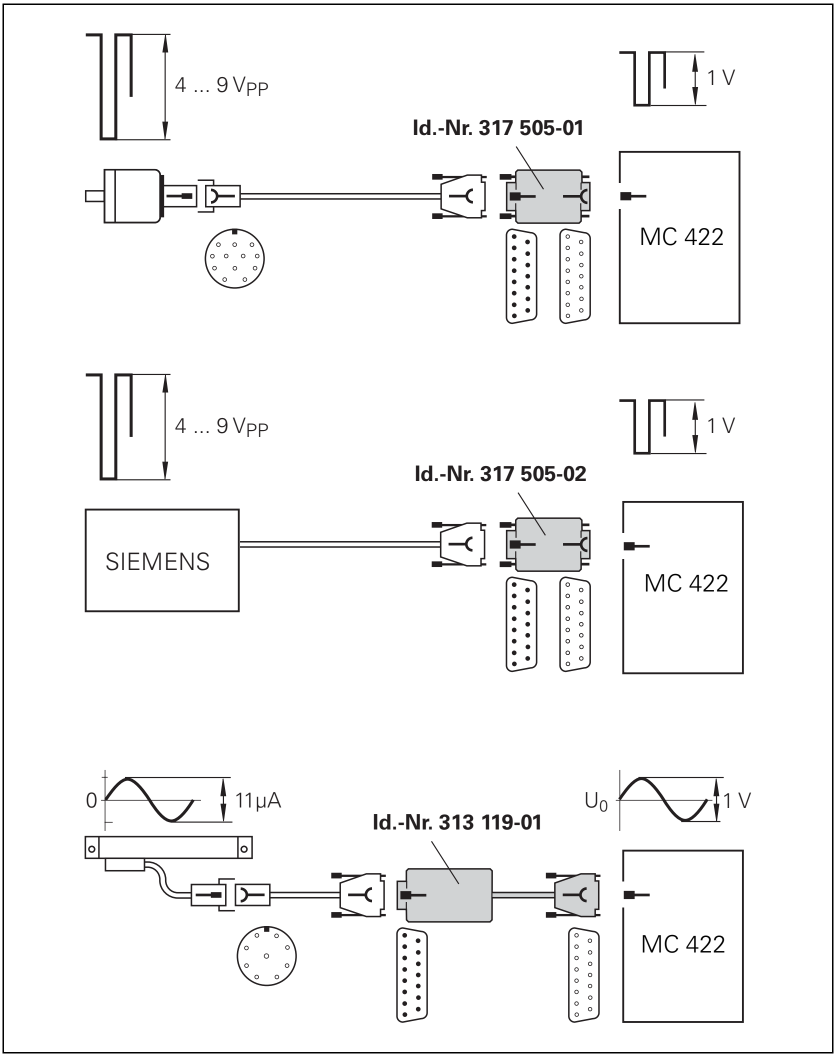
CC 422		Adapter cable 336 376-xx				Connecting cable 340 302-xx		
Male	**Assignment**	**Female**	**Color**	**Female**		**Male**	**Color**	**Female**
1	+5 V (U_P)	1	Brown/Green	10	Line drop compensator 336 697-01, if required	10	Brown/Green	10
2	0 V (U_N)	2	White/Green	7		7	White/Green	7
3	A+	3	Green/Black	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 line)	14	Blue	16		16	Blue	16
15	Data	15	Red	3		3	Red	3
16	0 V (sensor line)	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 50 178 for "low voltage electrical separation."

3.12 Adapters for Encoder Signals

Encoder signals with 11 μA_{PP} or TTL levels 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.

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	Assignment	D-sub connection (male) 15-pin	Assignment
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	Assignment	D-sub connection (male) 15-pin	Assignment
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	Assignment	D-sub connection (male) 15-pin	Assignment
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 are assigned to different controller groups, see "Maximum spindle speed" on page 6 – 14 and "PWM Frequency" on page 6 – 233.

The following applies to the output signals to the power stage:

Logic level:	5 V
Analog signals I_{ACTL}:	±7.5 V
PWM frequency:	MP2180 can be used to set it at 3333 Hz, 4166 Hz, 5000 Hz, 6666 Hz, 8166 Hz and 10000 Hz

X51 to X62: PWM output

Pin layout:

Ribbon cable 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 50 178 for "low voltage electrical separation."

3.14 Analog Input

The MC 422 and the PLC input/output board PL 410 B have analog inputs and inputs for Pt 100 thermistors.

The PL 410 B is available with and without analog inputs.

	Analog inputs (±10 V)	**Inputs for Pt 100 thermistors**
MC 422, X48	3	3
PL 405 B	–	–
PL 410 B (263 371-02)	4	4

Analog inputs

Voltage range:	–10 V to +10 V
Input resistance:	> 250 kΩ
Resolution (W480, W482, W484):	100 mV
Resolution (Module 9003):	10 mV (MC 422) 100 mV (PL 410 B)
Internal value range:	–100 to +100, at a resolution of 100 mV –1000 to +1000, at a resolution of 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):	0.1 °C (MC 422) 0.5 °C (PL 410 B)
Internal value range:	0 to 200, at a resolution of 0.5 °C 0 to 1000, at a resolution of 0.1 °C

X48: Analog input (PLC) on the MC 422

Pin layout:

Warning

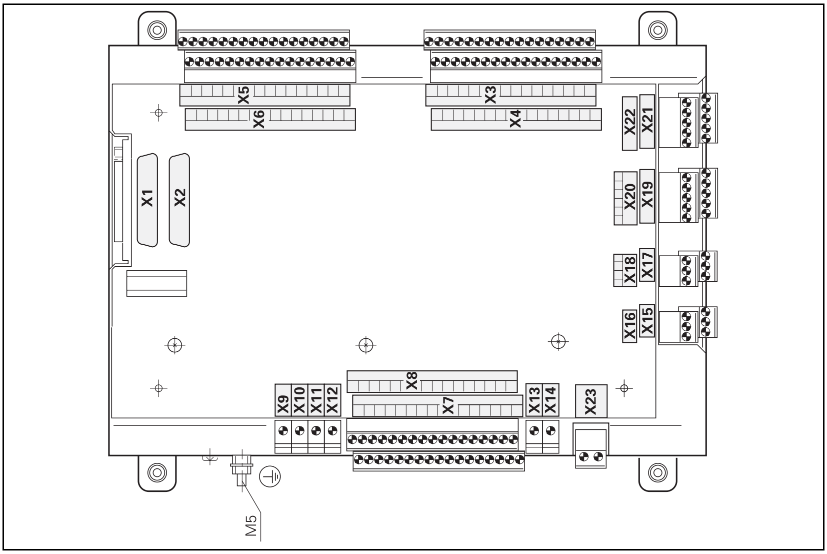
Remember to connect the analog inputs with the correct polarity!

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	External shield

Note

The interface complies with the requirements of EN 50 178 for “low voltage electrical separation.”

Connection overview for the PL 410 B



X15 to X18: Analog input on the PL 410 B

Pin layout

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

Note

The interfaces comply with the requirements of EN 50 178 for "low voltage electrical separation."

X19 to X22: Connection for Pt 100 on the PL 410 B

Pin layout:

Connecting terminals	Assignment
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 50 178 for "low voltage electrical separation."

Connection of analog voltage

Characteristics of the connecting cable:

- Shielding
- 2 conductors with 0.14 mm²
- Maximum length 50 meters

Connection of the Pt 100 thermistor inputs

- 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 capacity:	2 nF

13 analog outputs are available:

- Connection X8: Analog outputs 1 to 6
- Connection X9: Analog outputs 7 to 12

PLC analog output

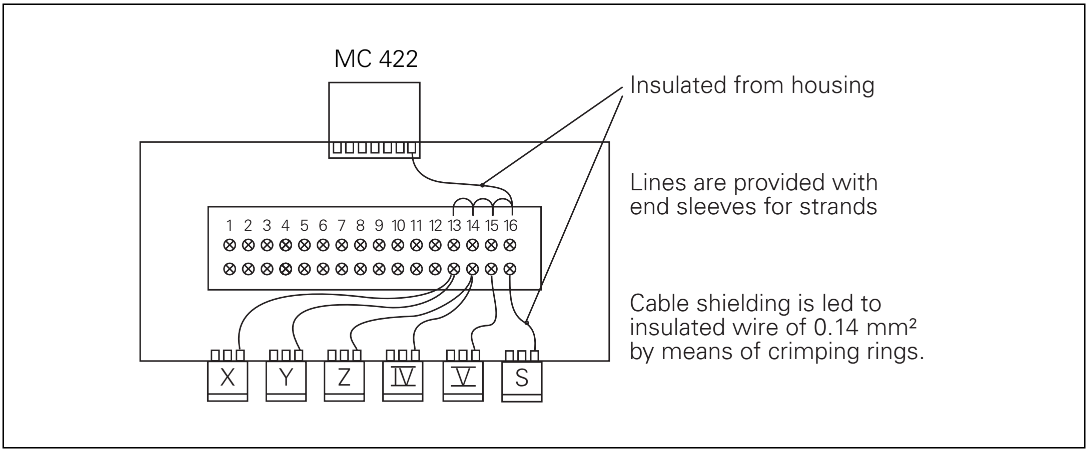
The PLC analog outputs can be controlled through Module 9130.

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. Nr. 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 422 and connecting cables:

MC 422		**Connecting cable**	
D-sub connctn. (female) 15-pin	**Assignment**	**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	External shield	Housing	External shield

X9: Analog outputs 7 to 13

For connecting cables, see “Cable Overview” at the end of this chapter.

Pin layout on the MC 422 and connecting cables:

MC 422		Connecting cable	
D-sub connctn. (female) 15-pin	**Assignment**	**D-sub connctr. (male) 15-pin**	**Color**
1	Analog output 7: ±10 V	1	Brown
2	Do not assign	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	Do not assign	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	External shield	Housing	External shield

3.16 Touch Probe Systems

The following touch probes can be connected to the iTNC 530:

- TS 220, a touch-trigger probe with cable connection for workpiece setup and measurement during machining
- TS 632, a touch-trigger probe 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 50 178 for "low voltage electrical separation."

Pin layout for TS 220:

MC 422		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	Ready	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 632 with EA 632:

MC 422		Adapter cable 310 197-xx			EA 632 346 322-xx		TS 632
Female	**Assignment**	**Male**	**Color**	**Female**	**Male**	**Color**	
1	0 V (internal shield)	1	White/ Brown	7	7	White/ Brown	
2	Do not assign						
3	Ready	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.	External shield	Hsg.	External shield	Hsg.	Hsg.		

a. Stylus at rest means logic level HIGH.

Two EA 652 can be connected to the MC 422 via the APE 652. This is necessary for example on large machines or on machines with swivel heads.

Pin layout for TS 632 with two EA 652 via the APE 652:

MC 422	Adapter cable 310 197-xx	APE 652 354 656-xx		Connecting cable 336 157-xx			EA 652 346 323-xx		TS 632
		Male	**Female**	**Male**	**Color**	**Female**	**Male**	**Color**	
See TS 632 with EA 550 for the layout		7	7	7	White/ Brown	7	7	White/ Brown	
		5	5	5	Gray	5	5	Gray	
		3	3	3	Yellow	3	3		
		2	2	2	Brown	2	2	Brown	
		6	6	6	Blue	6	6	Blue	
		1	1	1	White	1	1	White	
		4	4	4	Green	4	4	Green	
		Hsg.	Hsg.	Hsg.	External shield	Hsg.	Hsg.		

3.16.2 Triggering Touch Probe for Tool Measurement

X13: Connection of the touch probe

Pin layout on the MC 422:

> **Note**
>
> The interface complies with the requirements of EN 50 178 for "low voltage electrical separation."

Pin layout on adapter cable and touch probe:

MC 422		Adapter cable 335 332-xx			TT 130 296 537-xx	
Female	**Assignment**	**Male**	**Color**	**Female**	**Male**	**Color**
1	Ready	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.	External shield	Hsg.	External shield	Hsg.	Hsg.	

a. Stylus at rest means logic level HIGH.

3.17 Data Interfaces

Please note:

- Max. cable length with Ethernet is 400 m (shielded), 100 m (unshielded).
- Maximum cable length with RS-232-C/V.24 is 20 meters.
- Maximum cable length with RS-422-C/V.11 is 1000 meters.

For connecting cables, see “Cable Overview” at the end of this chapter.

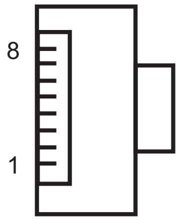
X26: Ethernet interface RJ45 port

Maximum data transfer rate:
Approx. 2 to 5 Mbps (depending on file type and network utilization)

Maximum cable length, shielded: 100 m

RJ-45 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	External shield

Face of the connector:



Note

The interface complies with the requirements of EN 50 178 for “low voltage electrical separation.”

Meaning of the LEDs on the Ethernet interface X26:

LED	Status	Meaning
Green	Blinks	Interface is active
	Off	Interface is inactive
Yellow	On	100-Mb network
	Off	10-Mb network

X27, X127: RS-232-C/V.24 data interface

Pin layout:

> **Note**
>
> The interface complies with the requirements of EN 50 178 for "low voltage electrical separation."

Adapter block 25-pin:

MC 422		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	8
5	Signal GND	5	Red	7	7	7	7	Red	7
6	DSR	6	Blue	6	6	6	6		6
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					8	Violet	20
Hsg.	External shield	Hsg.	Ext. shield	Hsg.	Hsg.	Hsg.	Hsg.	Ext. shield	Hsg.

Adapter block 9-pin:

MC 422		Connecting cable 355 484-xx			Adapter block 363 987-02		Connecting cable 366 964-xx		
Male	Assignment	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.	External shield	Hsg.	External shield	Hsg.	Hsg.	Hsg.	Hsg.	External shield	Hsg.

X28, X128: RS-422/V.11 data interface

Pin layout:

MC 422		Connecting cable 355 484-xx			Adapter block 363 987-01	
Female	**Assignment**	**Male**	**Color**	**Female**	**Male**	**Female**
1	$\overline{\text{RTS}}$	1	Red	1	1	1
2	$\overline{\text{DTR}}$	2	Yellow	2	2	2
3	$\overline{\text{RXD}}$	3	White	3	3	3
4	$\overline{\text{TXD}}$	4	Brown	4	4	4
5	Signal GND	5	Black	5	5	5
6	$\overline{\text{CTS}}$	6	Violet	6	6	6
7	$\overline{\text{DSR}}$	7	Gray	7	7	7
8	RXD	8	White/ Green	8	8	8
9	TXD	9	Green	9	9	9
Hsg.	External shield	Hsg.	External shield	Hsg.	Hsg.	Hsg.

Note

The interface complies with the requirements of EN 50 178 for "low voltage electrical separation."

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 the HRA 110 handwheel adapter
- HR 410 Portable Handwheel

X23: Handwheel input

Pin layout:

D-sub connection (female) 9-pin	Assignment
1	CTS
2	0 V
3	RTS
4	+12 V
5	Do not assign
6	DTR
7	TxD
8	RxD
9	DSR
Housing	External shield

Note

The interface complies with the requirements of EN 50 178 for "low voltage electrical separation."

3.18.1 HR 410 Portable Handwheel

The HR 410 is a portable electronic handwheel with the following functions:

- Keys for the selection of 5 axes
- Traverse direction keys
- Keys for 3 preset feed rates for latched traverse
- Actual-position-capture key
- 3 machine-function keys to be defined by the machine tool builder
- 2 permissive buttons
- EMERGENCY STOP button
- Holding magnets

Pin layout

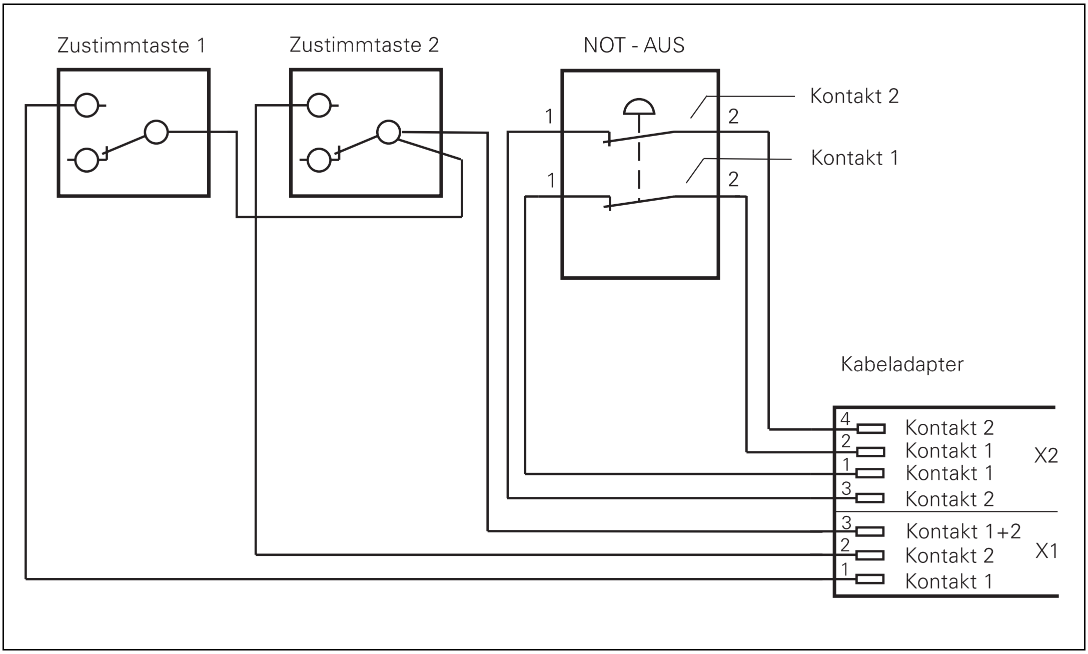
Pin layout for the various extension cables, adapter cables, connecting cables, and the handwheel:

Extension cable Id. Nr. 281 429-xx			**Adapter cable Id. Nr. 296 466-xx**			**Connecting cable see above**			**HR 410 Id. Nr. 296 469-01**	
D-sub connector (male) 9-pin		**D-sub connctr. (female) 9-pin**	**D-sub connector (male) 9-pin**		**Cplg. on mntg. base (female) (5+7)-pin**	**Cnnctr. (male) (5+7)-pin**		**Cnnctr. (female) (5+7)-pin**	**Connector (male) (5+7)-pin**	
Hsg.	Shield	Housing	Hsg.	Shield	Housing	Housing	Shield	Housing	Housing	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	RD/BL	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	Contact 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				

The adapter includes plug-in terminal strips for the contacts of the EMERGENCY STOP button and permissive button (max. load 24 Vdc, 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:



Additional components		Id. Nr.
Dummy plug for EMERGENCY STOP circuit		271 958-03
Connecting cable		
	Spiral cable	312 879-01
	Normal cable	296 467-xx
	Metal armor	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.

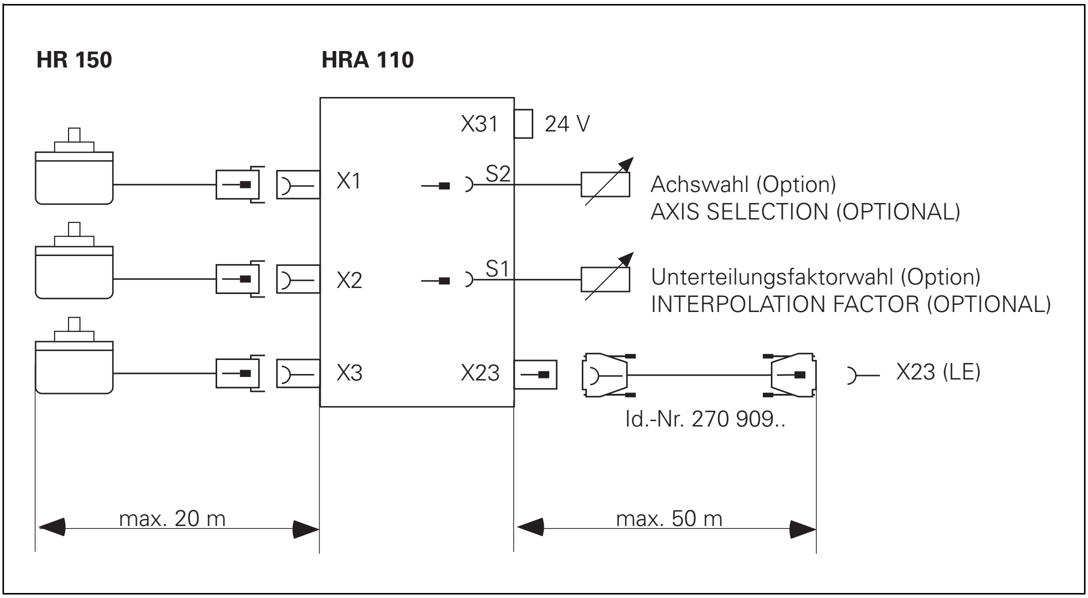
Pin layout for extension cable and handwheel:

Extension cable Id. Nr. 281 429-xx			HR 130 Id. Nr. 254 040-xx	
D-sub connctr. (male) 9-pin		**D-sub connctr. (female) 9-pin**	**D-sub connctr. (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 interpolation factor for the handwheel. In the PLC you must evaluate the current position of the handwheel selection switch and activate the corresponding interpolation 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	External shield

X23: Connection to MC 422

Pin layout on the HRA 110:

HRA 110	
D-sub connection (female) 9-pin	**Assignment**
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	External shield

X31: HRA 110 supply voltage

Pin layout on the HRA 110:

Warning

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	**Assignment**
1	+ 24 Vdc as per IEC 742 (VDE 551)
2	0 V

Maximum current consumption 200 mA.

3.19 Input: 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	Assignment
1	+24 V
2	0 V

3.20 Switching Inputs 24 Vdc (PLC)

3.20.1 Input Signals and Addresses

Input signals of the switching inputs on the MC 422 and the PL 4xx B:

Voltage range	MC 422	PL 4xx B
"1" signal: U_i	13 V to 30.2 V	
"0" signal: U_i	–20 V to 3.2 V	

Current ranges	MC 422	PL 4xx B
"1" signal: I_i	3.8 mA to 8.9 mA	2.5 mA to 6 mA
"0" signal: I_i when U_i = 3.2 V	1.0 mA	0.65 mA

Addresses of the switching inputs at:

Address	Number	Device
I0 to I31	31 + Control-is-ready signal	MC 422, X42 (PLC input)
I128 to I152	25	MC 422, X46 (machine operating panel)
I64 to I127 I64 to I95	64 32	First PLC I/O unit PL 410B First PLC I/O unit PL 405B
I192 to I255 I192 to I223	64 32	Second PLC I/O unit PL 410B Second PLC I/0 unit PL 405B
I256 to I319 I256 to I287	64 32	Third PLC I/O unit PL 410B Third PLC I/O unit PL 405B
I320 to I383 I320 to I351	64 32	Fourth PLC I/O unit PL 410B Fourth PLC I/O unit PL 405B

X42: PLC inputs on the MC 422

Pin layout:

MC 422		Connecting cable Id. Nr. 244 005-xx, Id. Nr. 263 954-xx	
D-sub connctn. (female) 37-pin	**Assignment**	**D-sub connctn. (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

MC 422		**Connecting cable Id. Nr. 244 005-xx, Id. Nr. 263 954-xx**	
D-sub connctn. (female) 37-pin	**Assignment**	**D-sub connctn. (male) 37-pin**	
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	External shield	Housing	External shield

3.20.2 PLC Inputs on the PL 410B

X3 to X6: PLC inputs

Pin layout on the PL:

X3				
Terminal	**Assignment**			
	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	**Assignment**			
	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	Assignment			
	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	Assignment			
	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	I379
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	**Assignment**			
	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	**Assignment**			
	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

3.21 Switching Outputs 24 Vdc (PLC)

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
- Short circuiting of **one** output is **permissible. No more than one** output may be short-circuited at one time.
- No more than half the PLC outputs may be driven at the same time (simultaneity factor 0.5)

Output signals:

	MC 422	**PL 4xxB**
Min. output voltage for “1” signal	3 V below supply voltage	
Nominal operating current per output	0.125 A (simultaneity factor 0.5)	2.0 A (at max. PL current consumption of 20 A)

Note

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

Addresses:

Address	**Number**	**Device**
O0 to O30	31	MC 422, X41 (PLC output)
O0 to O7	8	MC 422, X46 (machine operating panel)
O32 to O62	31	First PLC I/O unit
O64 to O94	31	Second PLC I/O unit
O128 to O158	31	Third PLC I/O unit
O160 to O190	31	Fourth PLC I/O unit

X41: PLC outputs on the MC 422

Pin layout:

MC 422		Connecting cable Id. Nr. 244 005-xx Id. Nr. 263 954-xx	
D-sub connctn. (female) 37-pin	**Assignment**	**D-sub connctr. (male) 37-pin**	
Supply via X44, pin 3; can be switched off with EMERGENCY STOP			
1	O0	1	Gray/Red
2	O1	2	Brown/Black
3	O2	3	White/Black
4	O3	4	Green/Black
5	O4	5	Brown/Red
6	O5	6	White/Red
7	O6	7	White/Green
8	O7	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/Red
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 422		Connecting cable Id. Nr. 244 005-xx Id. Nr. 263 954-xx	
D-sub connctn. (female) 37-pin	**Assignment**	**D-sub connctr. (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	34	Brown
35, 36, 37	Do not assign	35	Yellow/Pink, Violet, White
Housing	External shield	Housing	External shield

X7, X8: PLC outputs on the PL 410 B

Pin layout on the PL:

X7				
Terminal	**Assignment**			
	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	**Assignment**			
	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			

X8: PLC outputs on the PL 405 B

Pin layout on the PL:

X8				
Terminal	**Assignment**			
	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			

3.22 PL 4xx B PLC Input/Output Units

Up to four PL 4xx B can be connected to the TNC.

The PL 410 B is available with and without analog inputs.

Device	Id. Nr.	Switching inputs 24 Vdc	Switching outpt. 24 Vdc	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

Pin layout:

MC 422		Conn. cable Id. Nr. 289 111-xx / Id. Nr. 317 788-xx			1st PL 410 B/PL405 B	
D-sub connctn. (male) 25-pin	**Assignment**	**D-sub connctr. (female) 25-pin**		**D-sub connctr. (male) 25-pin**	**X1 D-sub connctn. (female) 25-pin**	**Assignment**
1	0 V	1	Brown, Yellow, Pink, Red, Violet	1	1	0 V
2	0 V	2	Red/Blue, Brown/Green, Yellow/Brown, Gray/Brown, Pink/Brown	2	2	0 V
3	0 V	3	Brown/Blue, Brown/Red, Brown/Black, Yellow/Gray, Yellow/Pink	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

MC 422		Conn. cable Id. Nr. 289 111-xx / Id. Nr. 317 788-xx			1st PL 410 B/PL405 B	
D-sub connctn. (male) 25-pin	**Assignment**	**D-sub connctr. (female) 25-pin**		**D-sub connctr. (male) 25-pin**	**X1 D-sub connctn. (female) 25-pin**	**Assignment**
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	External shield	Housing	External shield	Housing	Housing	External shield

X2: PLC expansion PL 4xx B on the PL 410 B

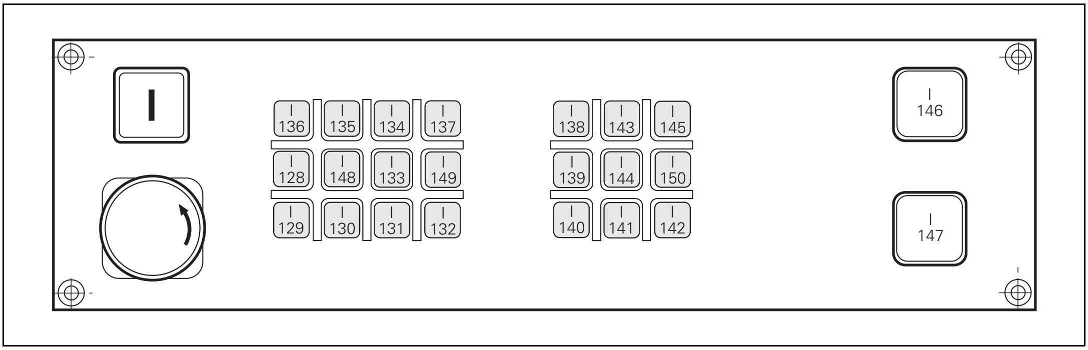
Pin layout:

PL 410 B		Conn. cable Id. Nr. 289 111-xx / Id. Nr. 317 788-xx			PL 410 B PL 405 B on the PL 410 B	
D-sub connctn. (male) 25-pin	**Assignment**	**D-sub connctr. (female) 25-pin**		**D-sub connctr. (male) 25-pin**	**X1 D-sub connctn. (female) 25-pin**	**Assignment**
1	0 V	1	Brown, Yellow, Pink, Red, Violet	1	1	0 V
2	0 V	2	Red/Blue, Brown/Green, Yellow/Brown, Gray/Brown, Pink/Brown	2	2	0 V
3	0 V	3	Brown/Blue, Brown/Red, Brown/Black, Yellow/Gray, Yellow/Pink	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	External shield	Housing	Housing	Ext. shield

3.23 Machine Operating Panel

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	Assignment
1	I151
2	I152
3	+24 V

X4: PLC outputs

Pin layout:

Terminal	Assignment
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 422, connecting cables and machine operating panel:

Warning

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 422		**Connecting cable Id. Nr. 263 954-xx**			**MB 420**	
D-sub connctn. (female) 37-pin	**Assignment**	**D-sub connctr. (male) 37-pin**		**D-sub connctr. (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 422		Connecting cable Id. Nr. 263 954-xx			MB 420	
D-sub connctn. (female) 37-pin	Assignment	D-sub connctr. (male) 37-pin		D-sub connctr. (female), 37-pin	D-sub connctn. (male) 37-pin	Key
26	O0*	26	Gray/Brown	26	26	Via X4
27	O1*	26	Yellow/Black	27	27	Via X4
28	O2*	28	White/Yellow	28	28	Via X4
29	O3*	29	Gray/Blue	29	29	Via X4
30	O4*	30	Pink/Blue	30	30	Via X4
31	O5*	31	Pink/Red	31	31	Via X4
32	O6*	32	Brown/Blue	32	32	Via X4
33	O7*	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	External shield	Housing	Housing	

3.24 iTNC Keyboard Unit

X1: Connection of soft keys on the visual display unit with the iTNC keyboard

Pin layout:

Connector (male) 9-pin	Assignment
1	SL0
2	SL1
3	SL2
4	SL3
5	Do not assign
6	RL15
7	RL14
8	RL13
9	RL12

X45: iTNC keyboard unit

Pin layout:

MC 422		Connecting cable Id. Nr. 263 954-xx			TE
D-sub connctn. (female) 37-pin	Assignment	D-sub cnnctr. (male) 37-pin		D-sub cnnctr. (female) 37-pin	X2: 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
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

MC 422		Connecting cable Id. Nr. 263 954-xx			TE
D-sub connctn. (female) 37-pin	**Assignment**	**D-sub cnnctr. (male) 37-pin**		**D-sub cnnctr. (female) 37-pin**	**X2: D-sub connctn. (male) 37-pin**
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	External shield	Housing	External shield	Housing	Housing

3.25 Flat-Panel Display

X3: Connection of screen soft keys

X3 Soft keys (see "iTNC Keyboard Unit" on page 3 – 72).

X49: BF 120 flat-panel display

Pin layout:

MC 422, X49		Connecting cable Id. Nr. 340 300-xx			BF 120, X2
D-sub connctn. (female) 62-pin	**Assign-ment**	**D-sub connctr. (male) 62-pin**		**D-sub connector (female) 62-pin**	**D-sub connctn. (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	Free	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 422, X49		**Connecting cable Id. Nr. 340 300-xx**			**BF 120, X2**
D-sub connctn. (female) 62-pin	**Assign-ment**	**D-sub connctr. (male) 62-pin**		**D-sub connector (female) 62-pin**	**D-sub connctn. (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	Free	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	Free	53 to 56	53 to 56
57 to 62	0 V	57 to 62	Free	57 to 62	57 to 62
Housing		Housing		Housing	Housing

X149: BF 150 flat-panel display

Pin layout:

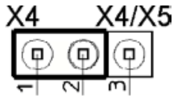
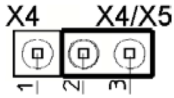
MC 422, X149		Connecting cable Id. Nr. 353 545-xx			BF 150, X2
D-sub connctn. (female) 44-pin	**Assignment**	**D-sub connctr. (male), 44-pin**		**D-sub connctr. (female), 44-pin**	**D-sub connctn. (male) 44-pin**
1	A7M	1		1	1
2	A6M	2	White/Brown	2	2
3	A5M	3	White/Green	3	3
4	A4M	4	Red/Gray	4	4
5	A3M			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
25	Not assigned			25	25
26	Not assigned			26	26
27	Not assigned			27	27
28	Not assigned			28	28
29	Not assigned			29	29
30	Not assigned			30	30
31	LVDSGND			31	31
32	LVDSGND			32	32
33	LVDSGND			33	33

MC 422, X149		Connecting cable Id. Nr. 353 545-xx			BF 150, X2
D-sub connctn. (female) 44-pin	**Assignment**	**D-sub connctr. (male), 44-pin**		**D-sub connctr. (female), 44-pin**	**D-sub connctn. (male) 44-pin**
34	LVDSGND			34	34
35	LVDSGND			35	35
36	LVDSGND			36	36
37	LVDSGND			37	37
38	LVDSGND			38	38
39	LVDSGND			39	39
40	Not assigned			40	40
41	Not assigned			41	41
42	Not assigned			42	42
43	Not assigned			43	43
44	Not assigned			44	44
Housing		Housing		Housing	Housing

3.26 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 422 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 422 (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 interfaces comply with the requirements of EN 50 178 for "low voltage electrical separation."

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: Supply voltage for BTS 1x0

Pin layout:

Connecting terminal	Assignment
1	+24 V
2	0 V

3.27 USB Interface

X141, X142

Pin layout:

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

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 the MC 422)
X141	USB output 1
X142	USB output 2
X143	USB output 3
X144	USB output 4

3.28 PS/2 interface

X131: PS/2 keyboard

Pin layout:

PS/2 connection (female) 6-pin	Assignment
1	KBDATA
2	$\overline{\text{EN.EKB}}$ (enable external keyboard)
3	GND
4	+5 V
5	BKCLK
6	Do not assign

X133: PS/2 mouse

Pin layout:

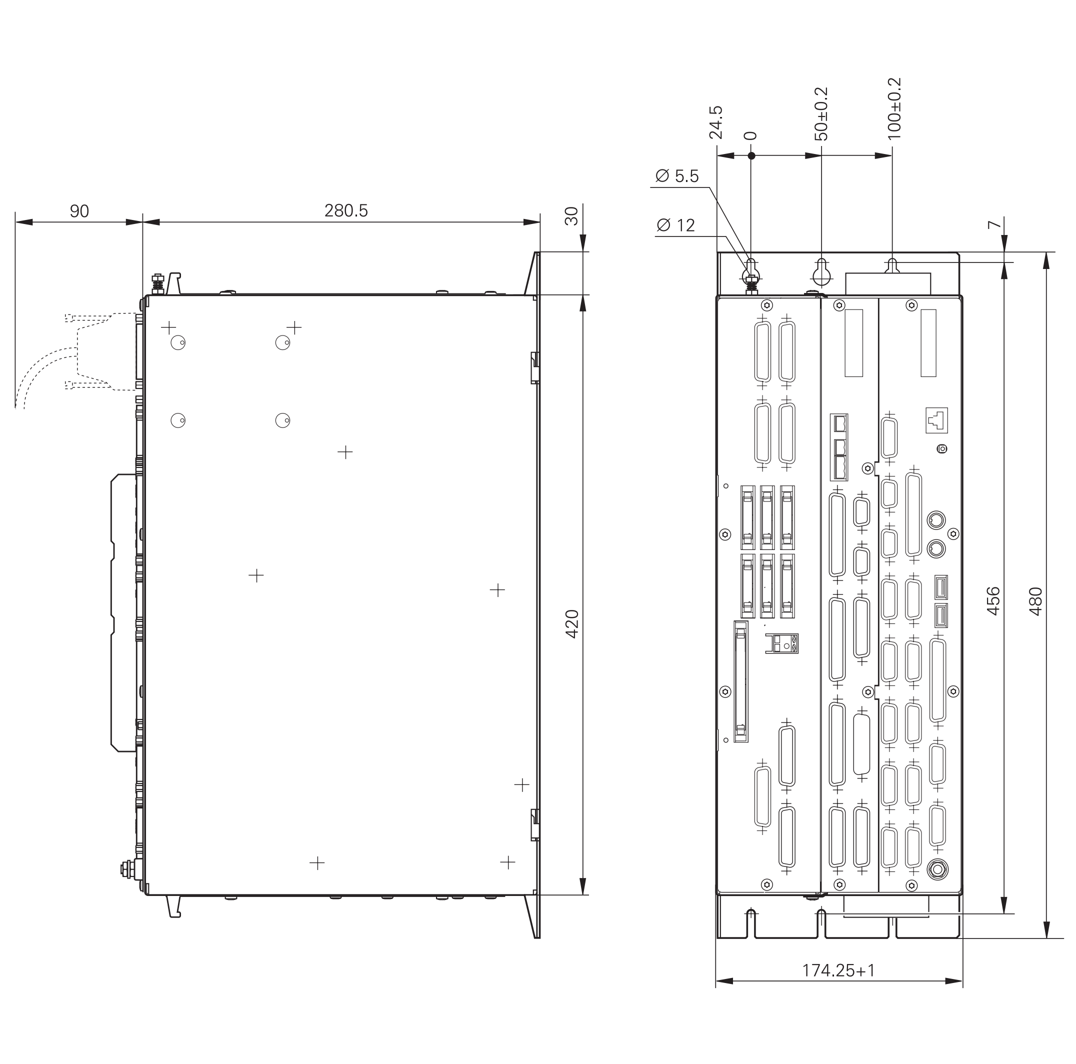
PS/2 connection (female) 6-pin	Assignment
1	MSDATA
2	Do not assign
3	GND
4	+5 V
5	MSCLK
6	Do not assign

3.29 Dimensions

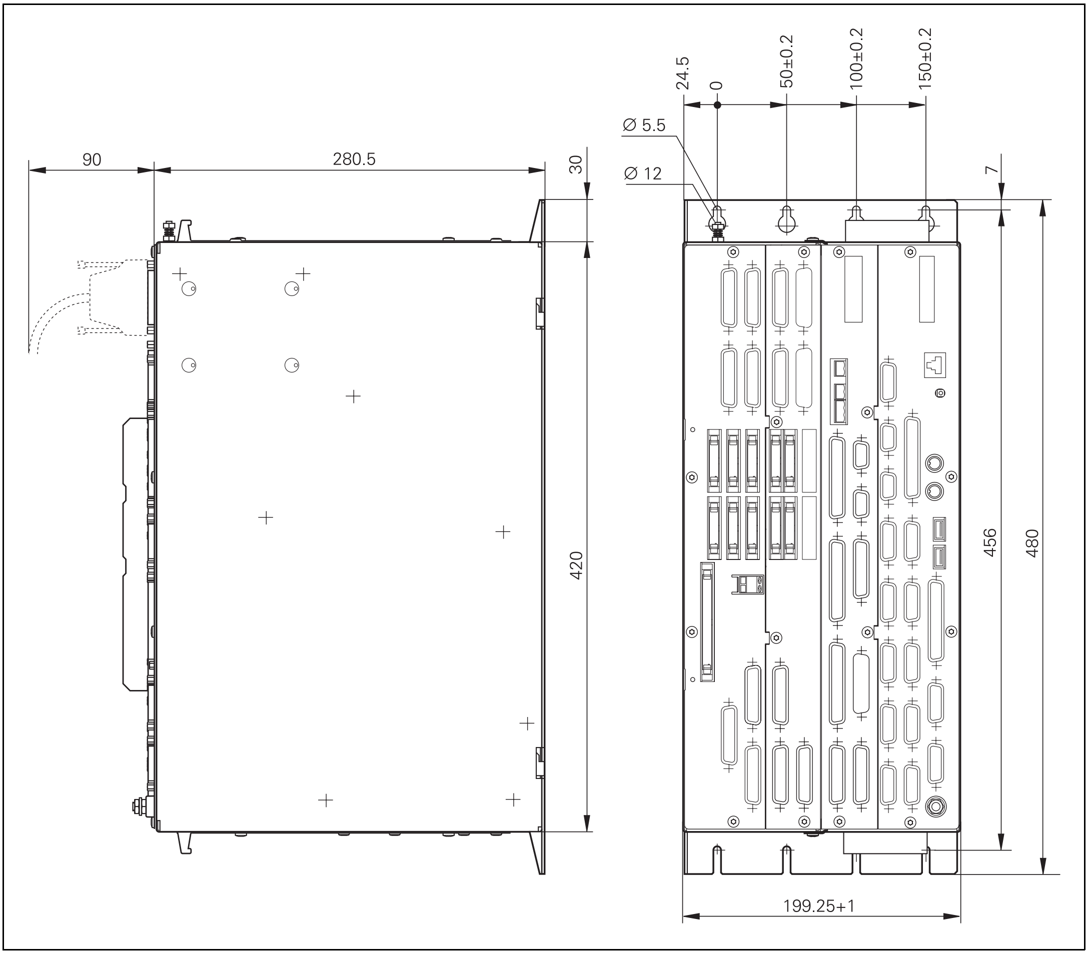
Note

All dimensions in [mm].

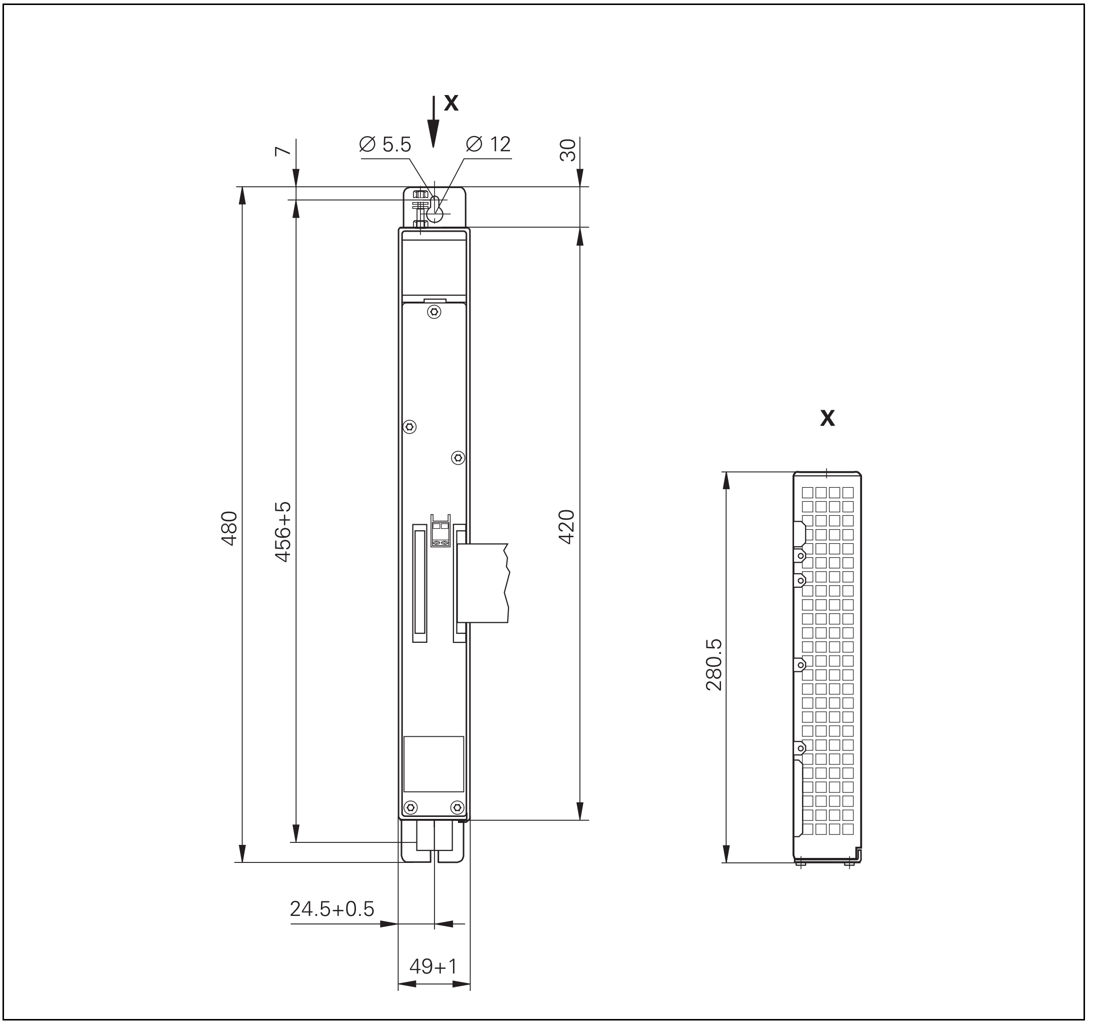
3.29.1 MC 422/5 Position Encoder Inputs and CC 422 with 6 Control Loops



3.29.2 MC 422/10 Position Encoder Inputs and CC 422 with 10 or 12 Control Loops

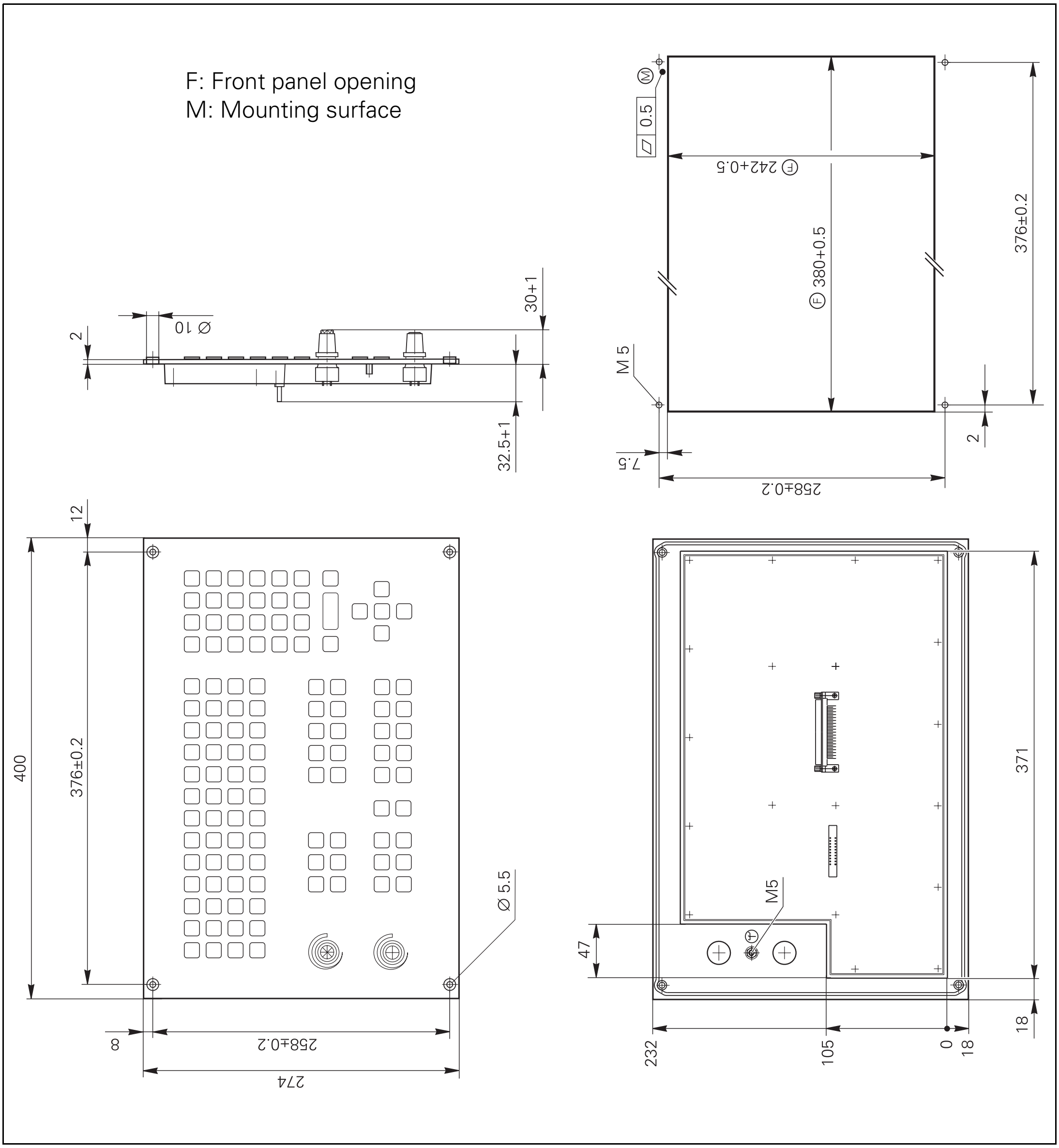


3.29.3 UV 105



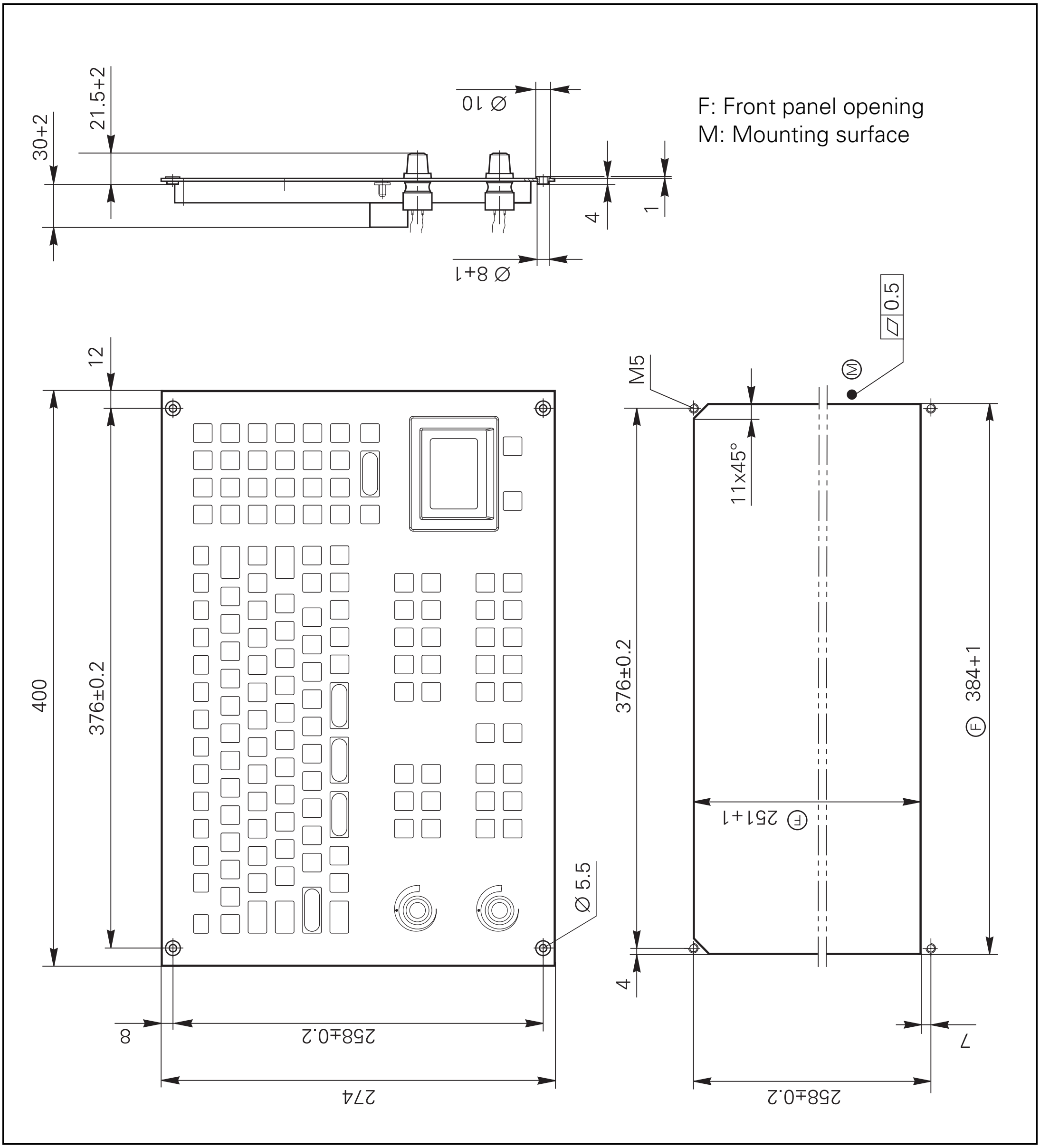
3.29.4 TE 420

Weight: 2.4 kg



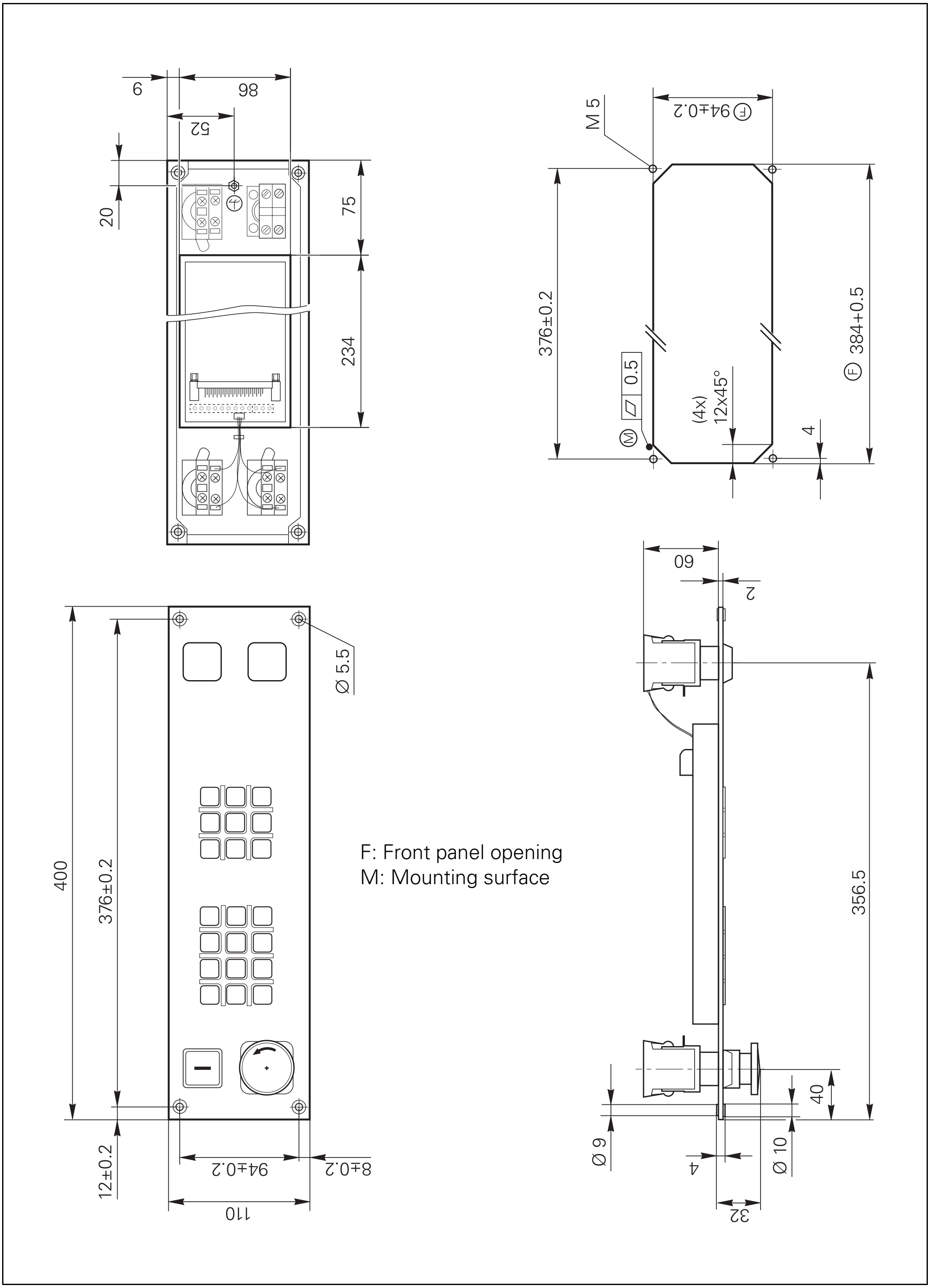
3.29.5 TE 530

Weight: 2.4 kg



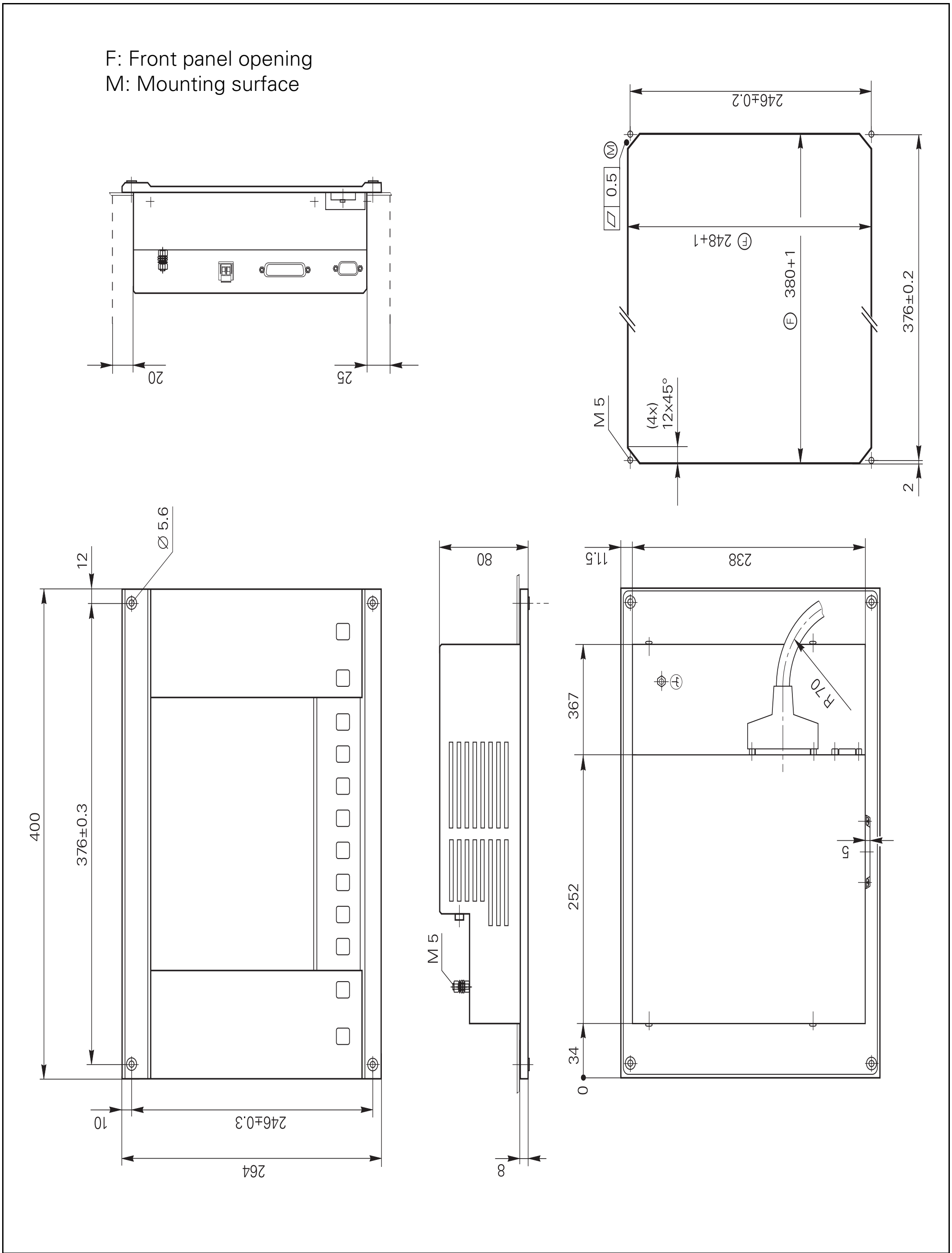
3.29.6 MB 420

Weight: 0.9 kg



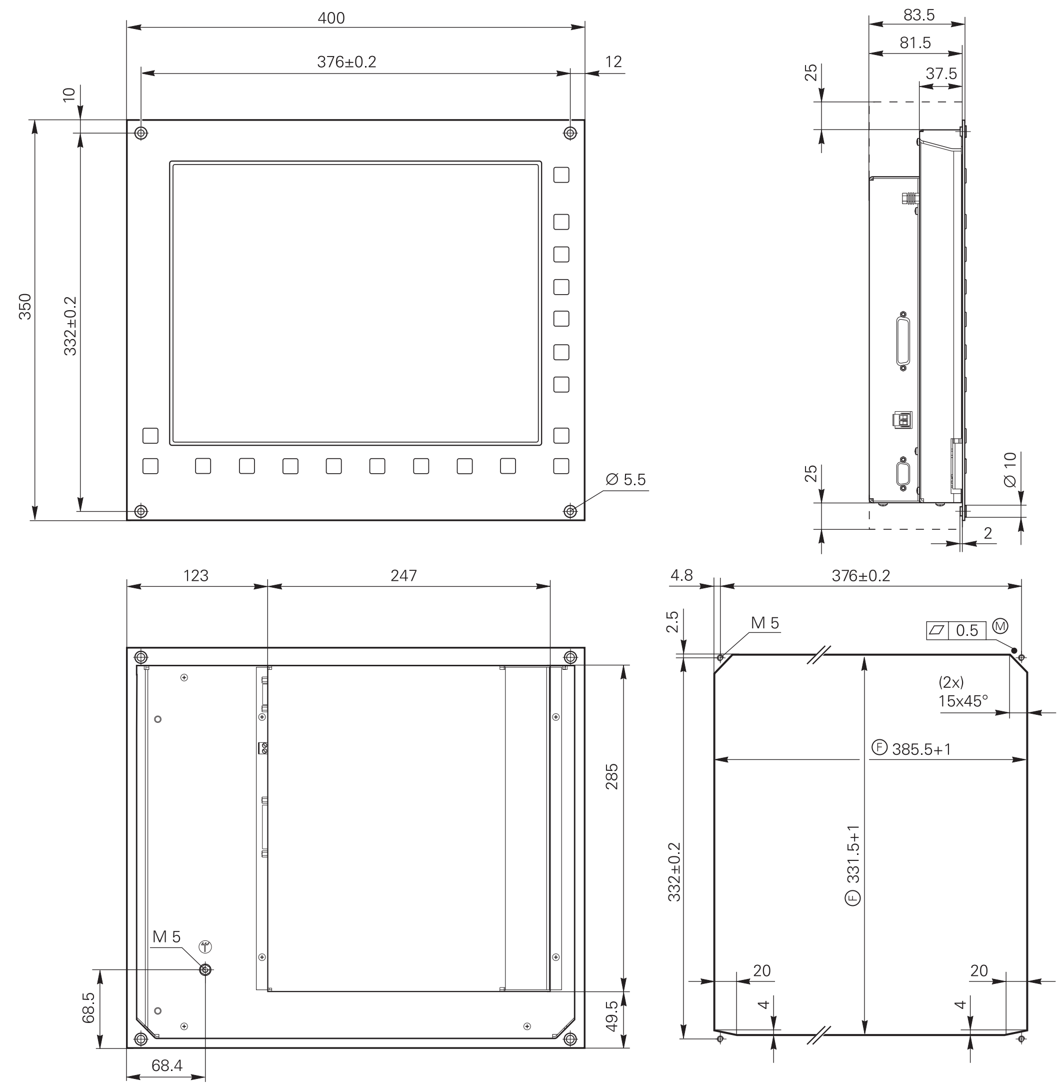
3.29.7 BF 120

Weight: 3 kg



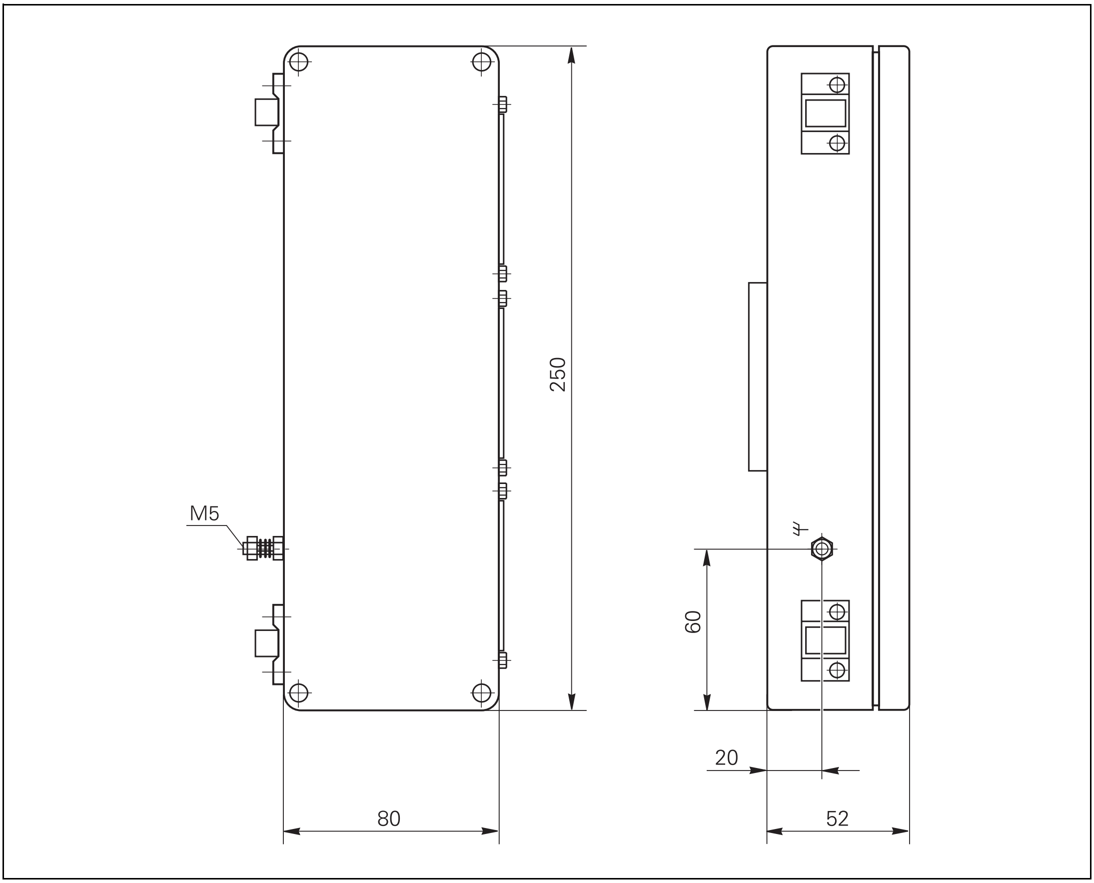
3.29.8 BF 150

Weight: 3 kg



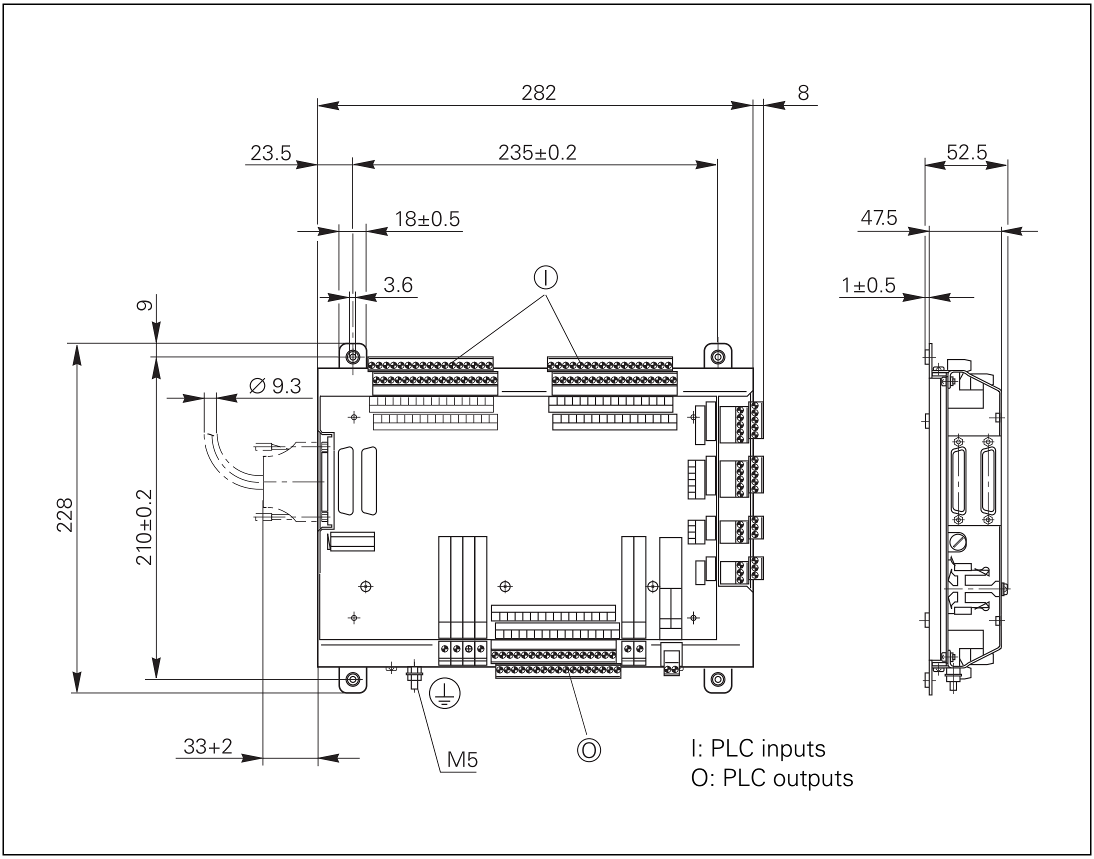
F: Front panel opening
M: Mounting surface

3.29.9 BTS 120/BTS 150



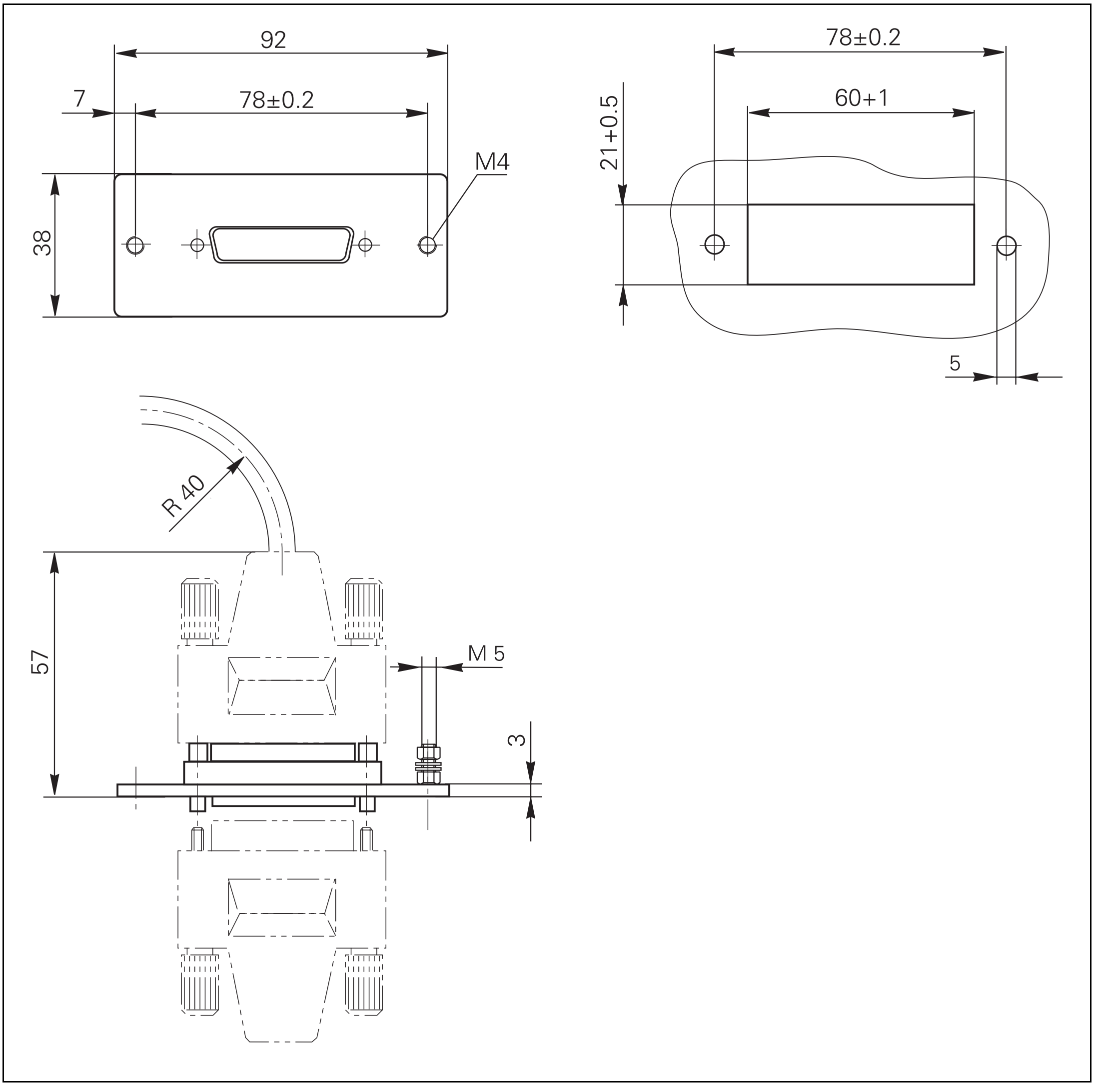
3.29.10 PL 4xx B

Weight: 1.5 kg

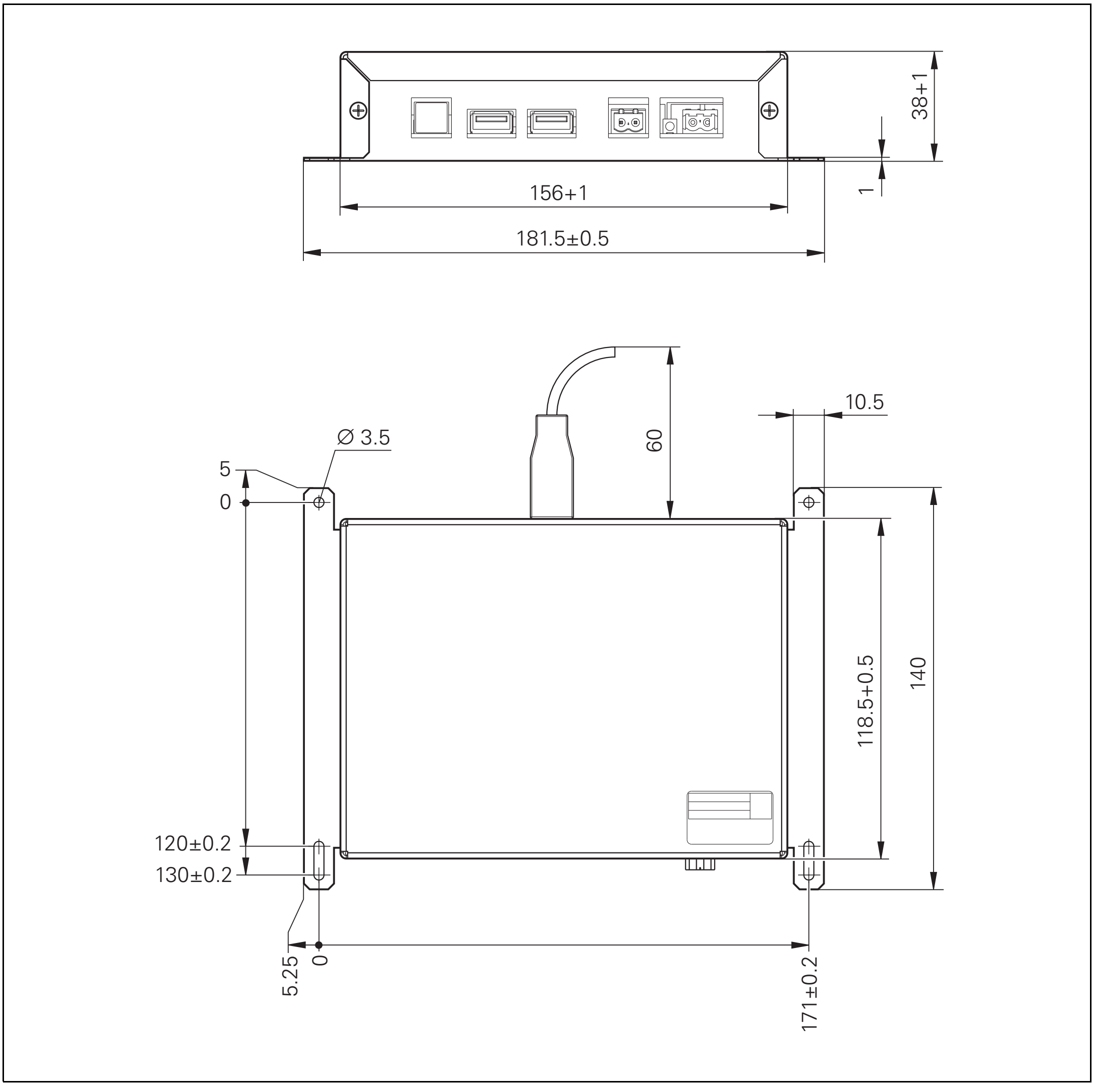


3.29.11 Adapter Block for the Data Interface

RS-232-C/V.24 adapter block and
RS-422/V.11 adapter block

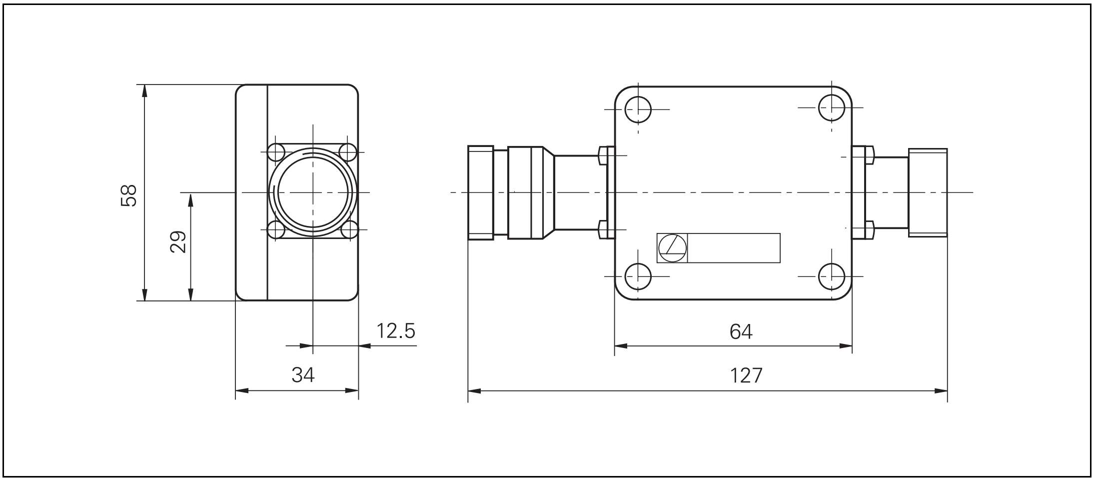


3.29.12 USB Hub



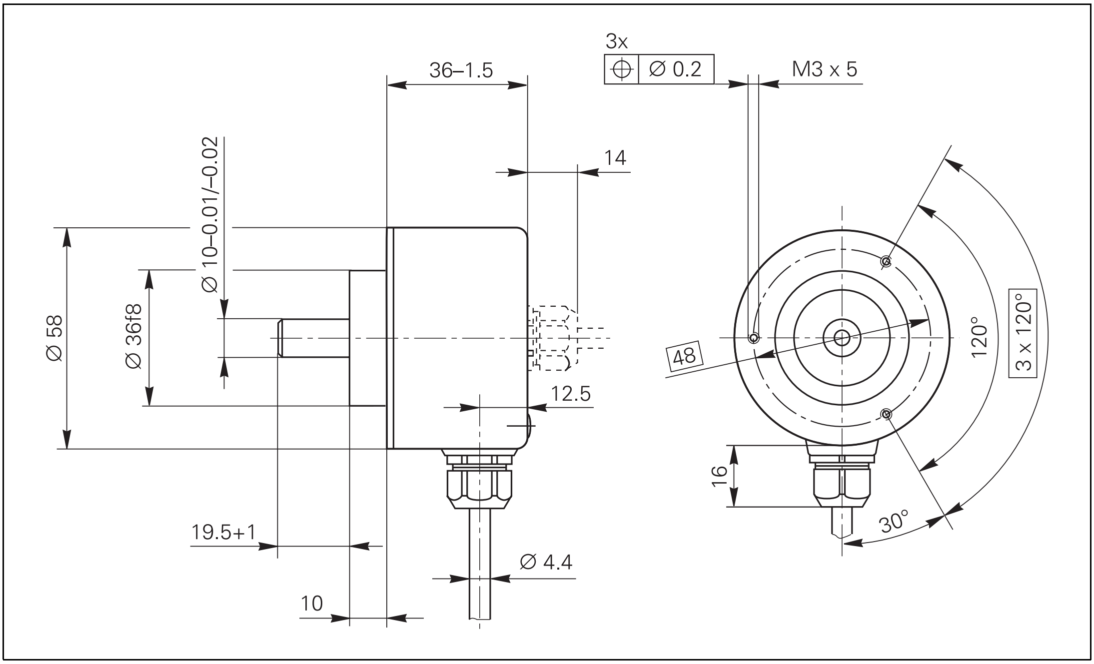
3.29.13 Voltage Controller

Voltage controller for encoders with EnDat interface

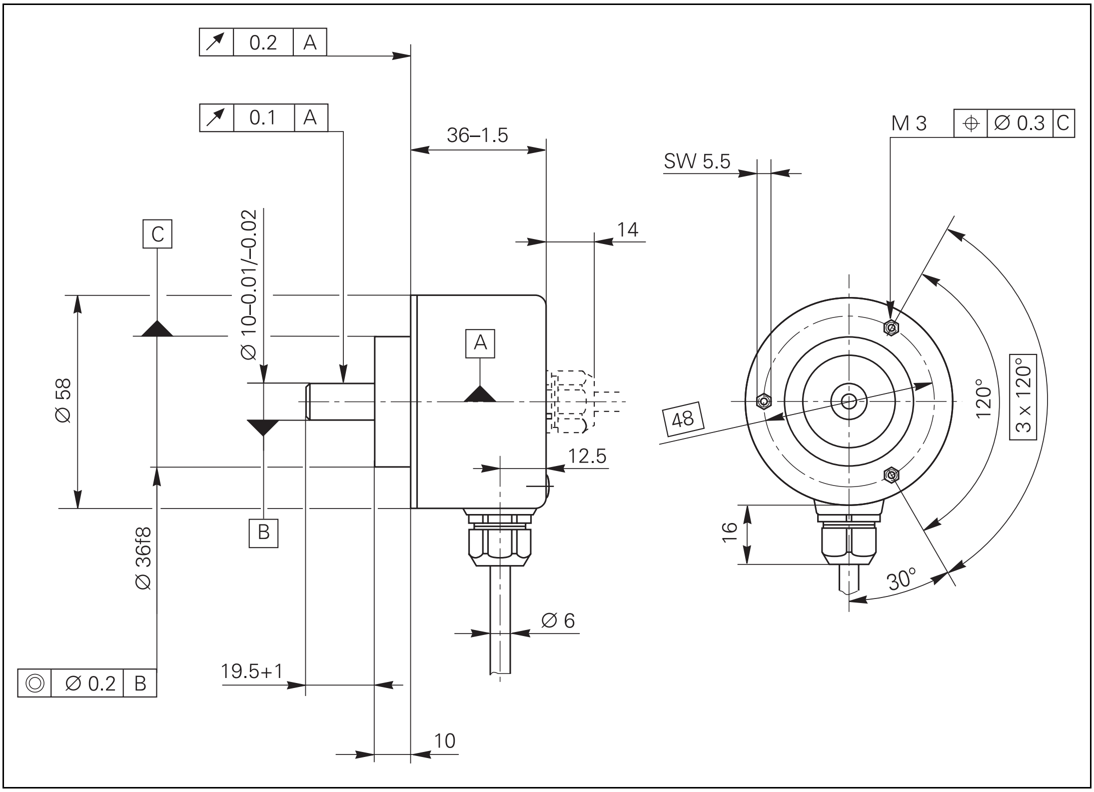


3.29.14 Handwheels

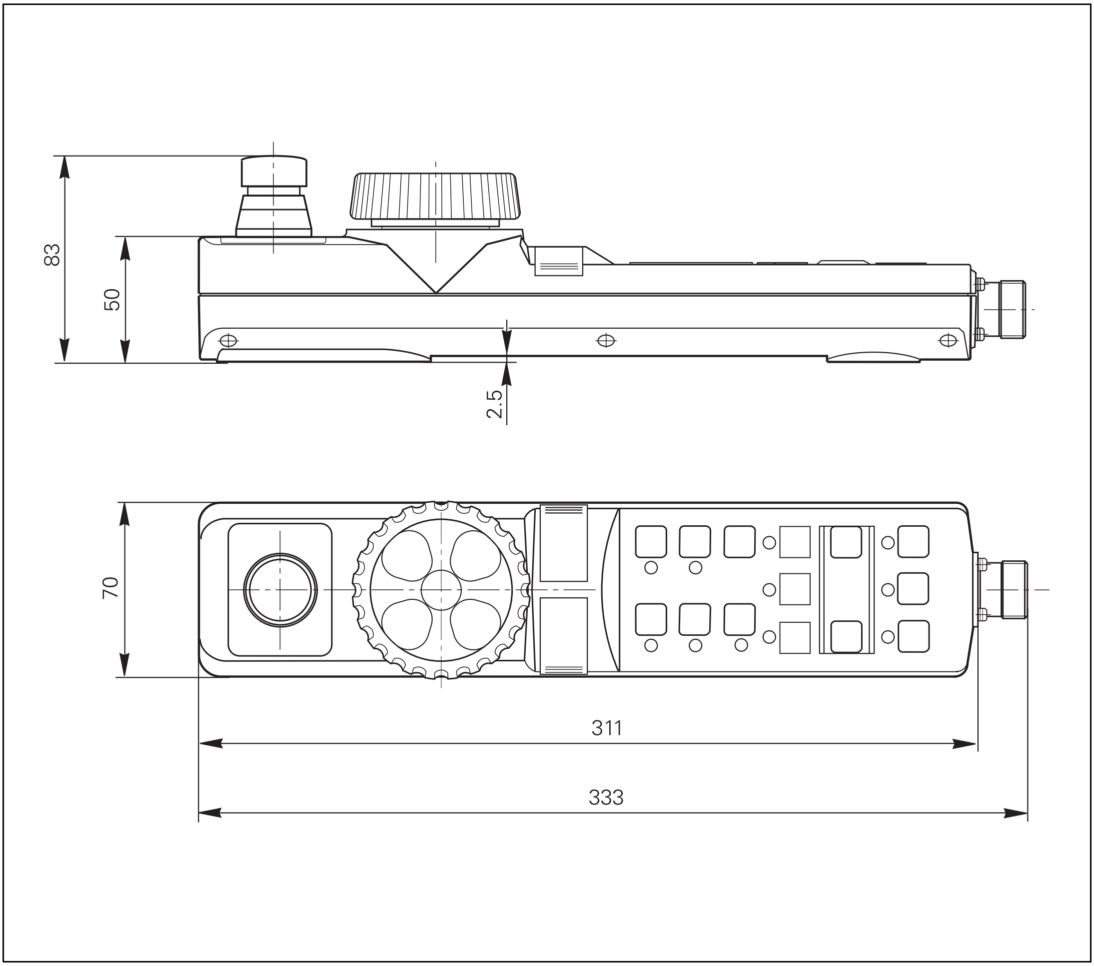
HR 130 Weight: approx. 0.7 kg



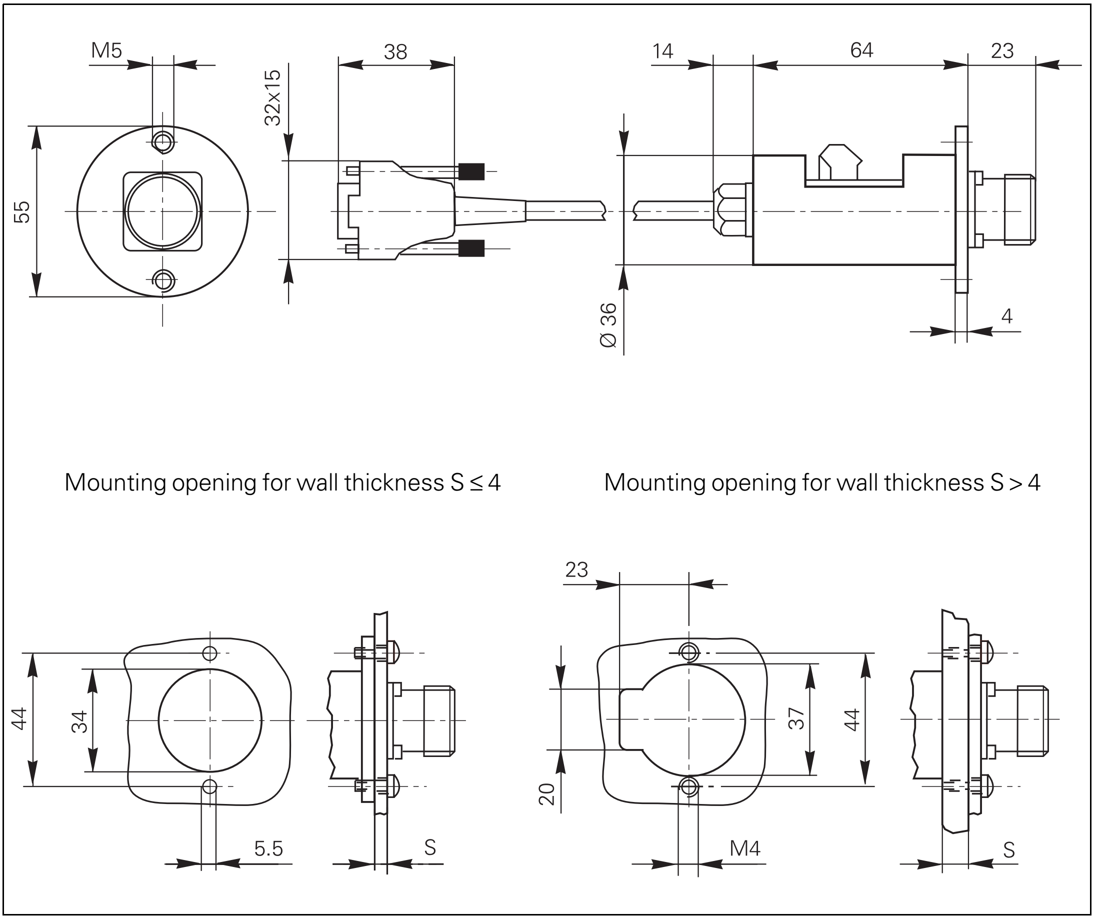
HR 150 Weight: approx. 0.7 kg



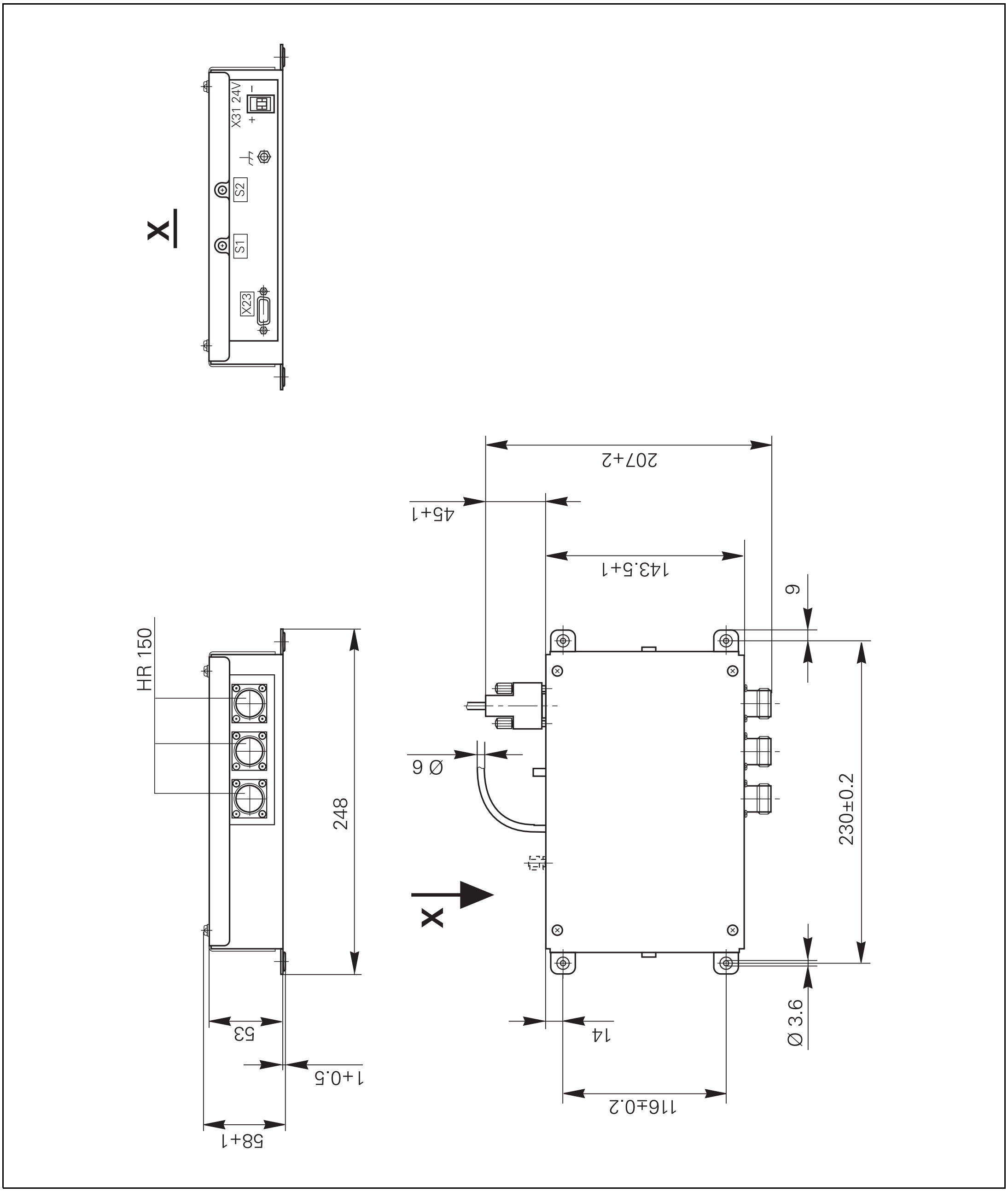
HR 410



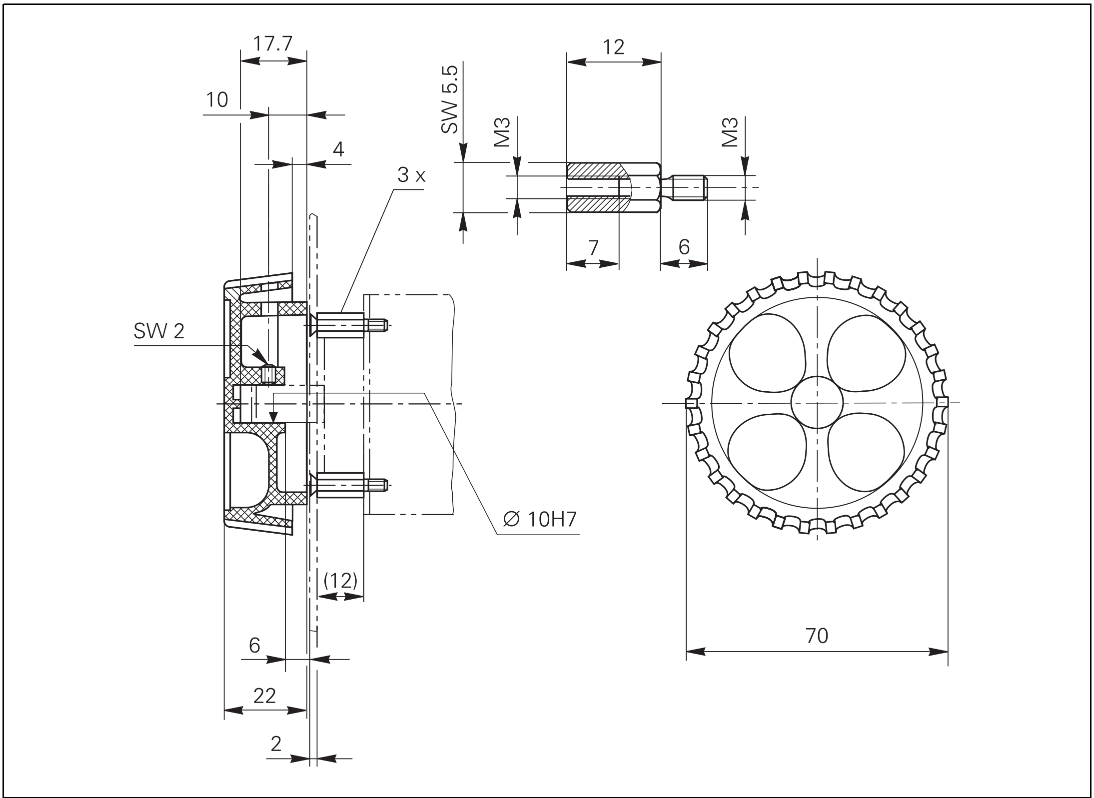
Adapter cables



HRA 110 Weight: approx. 1.5 kg

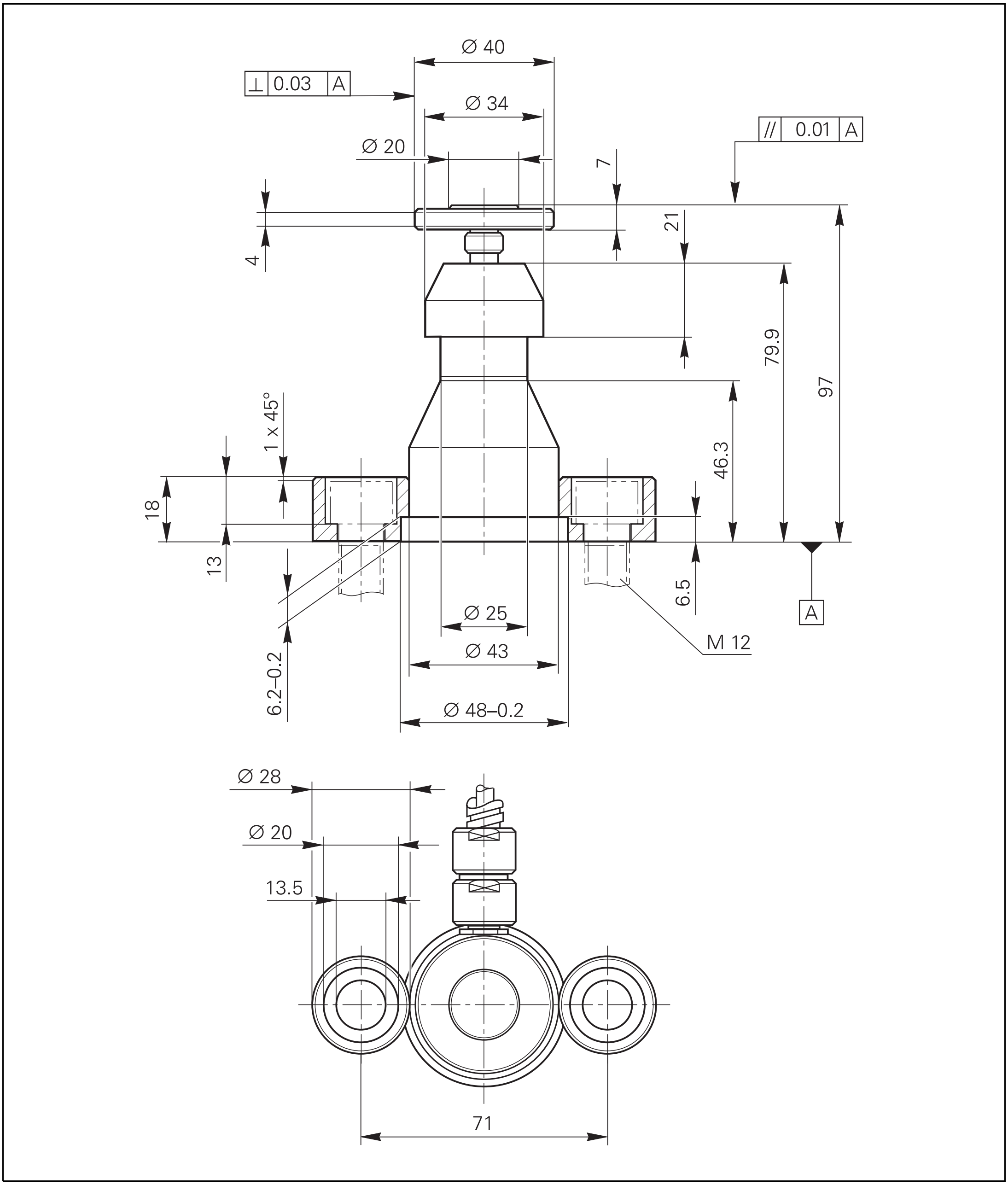


Control knob for HR 130 and HR 150

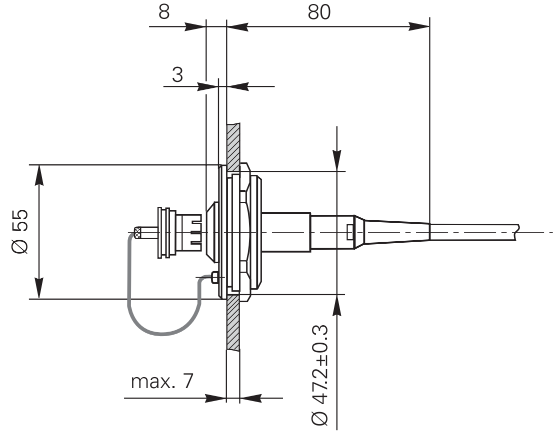


3.29.15 Touch Probe Systems

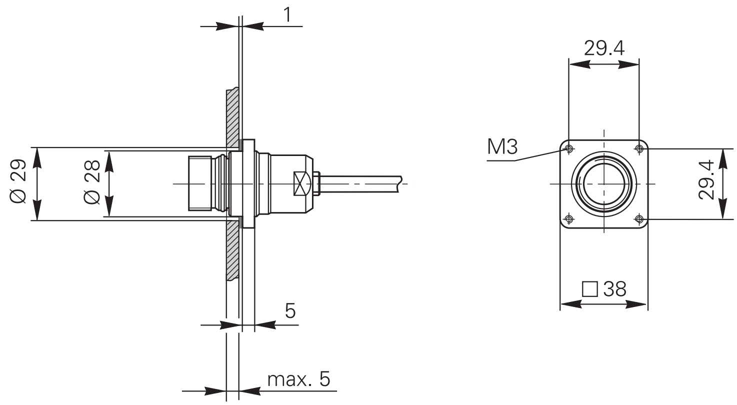
TT 130



Adapter cable for TT and TS

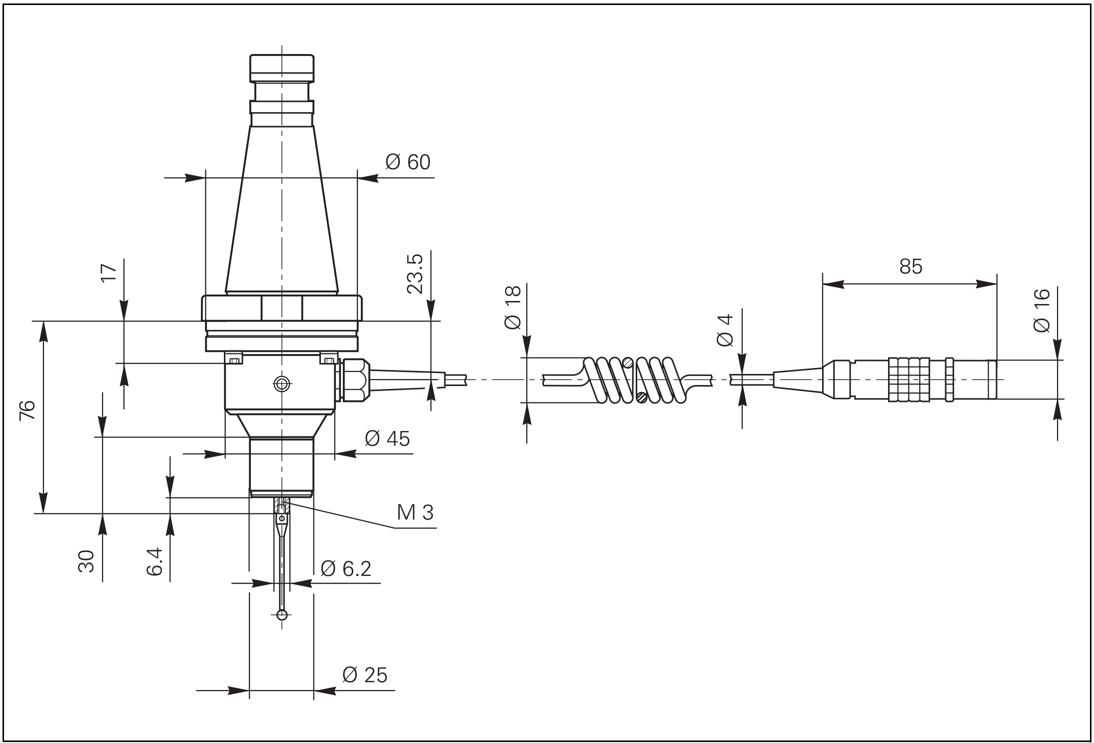


Mounting coupling for quick connection

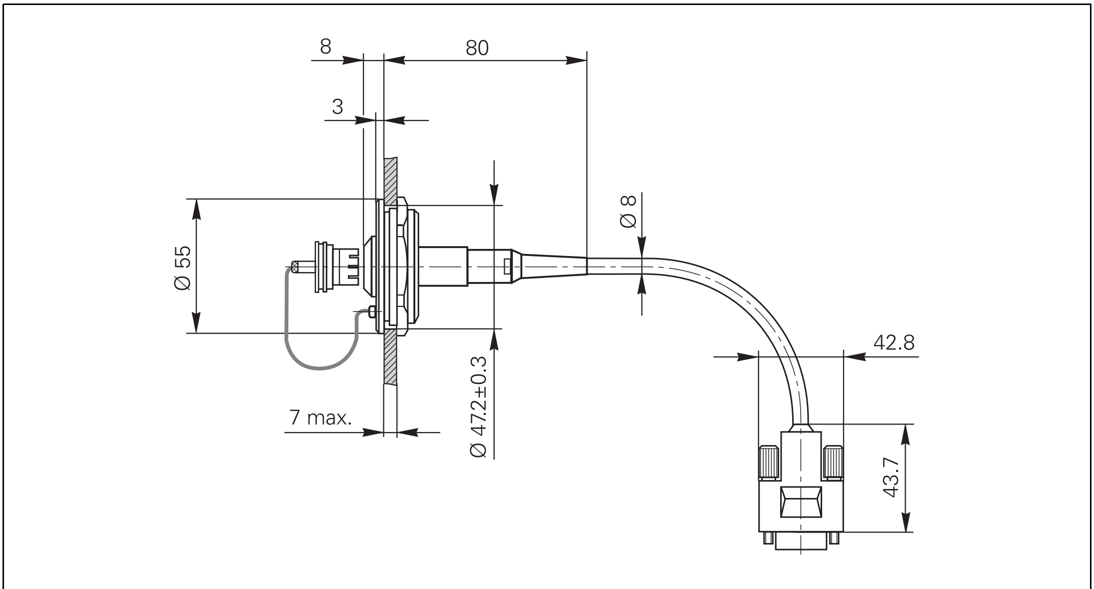


Mounting coupling for HEIDENHAIN standard connector

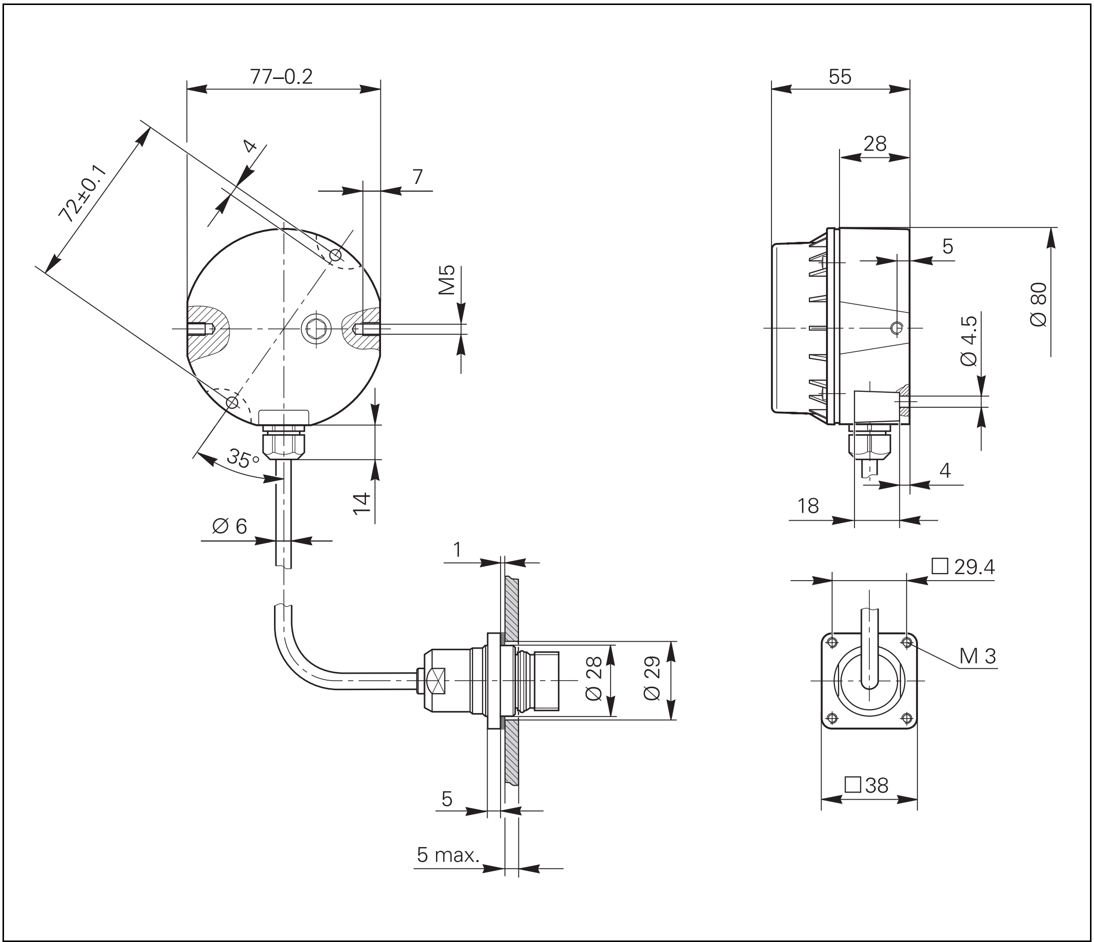
TS 220



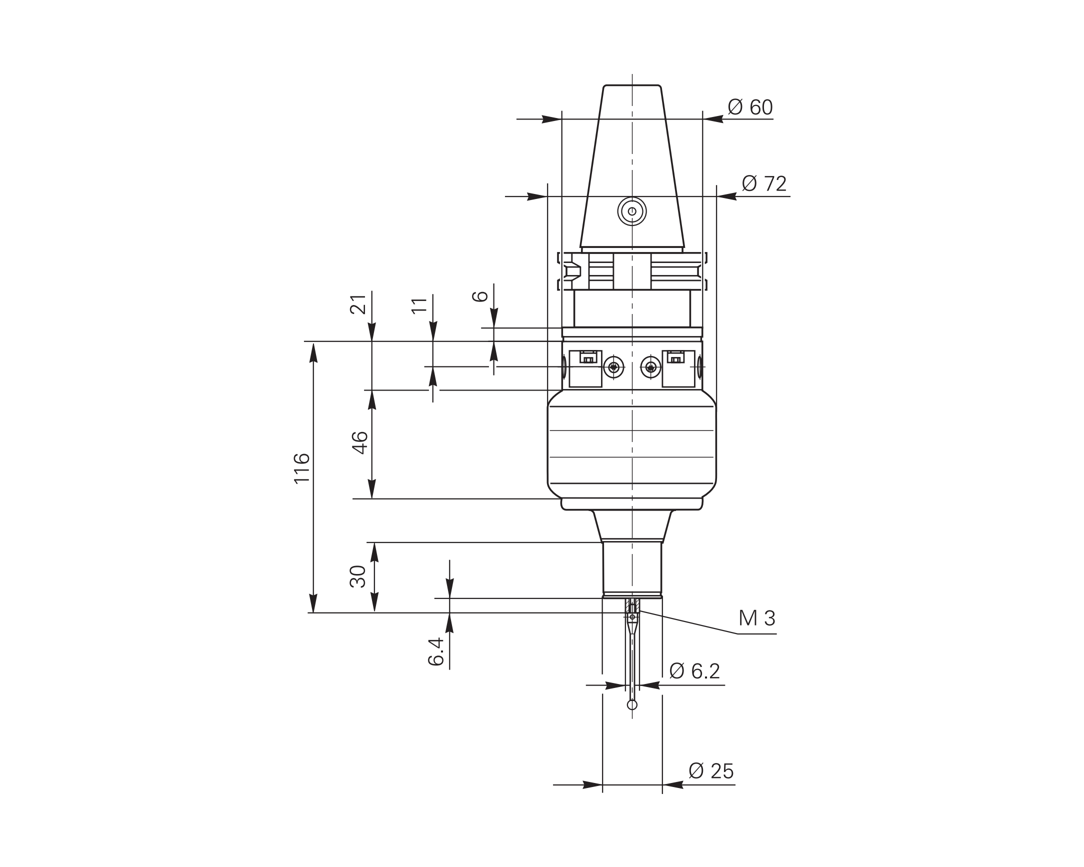
Adapter cable for
TS 120/TS 220



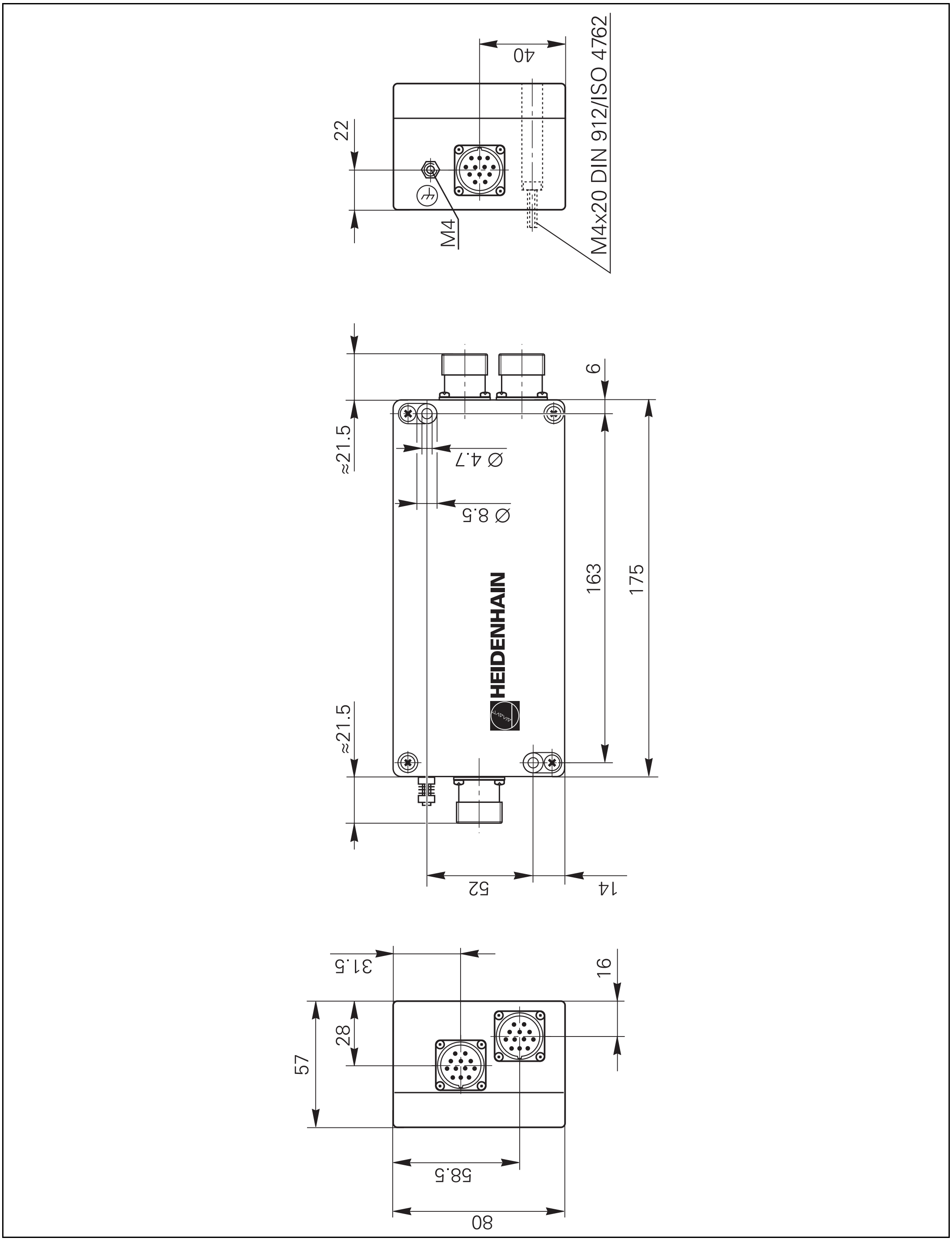
EA 6x2 receiver unit



TS 632



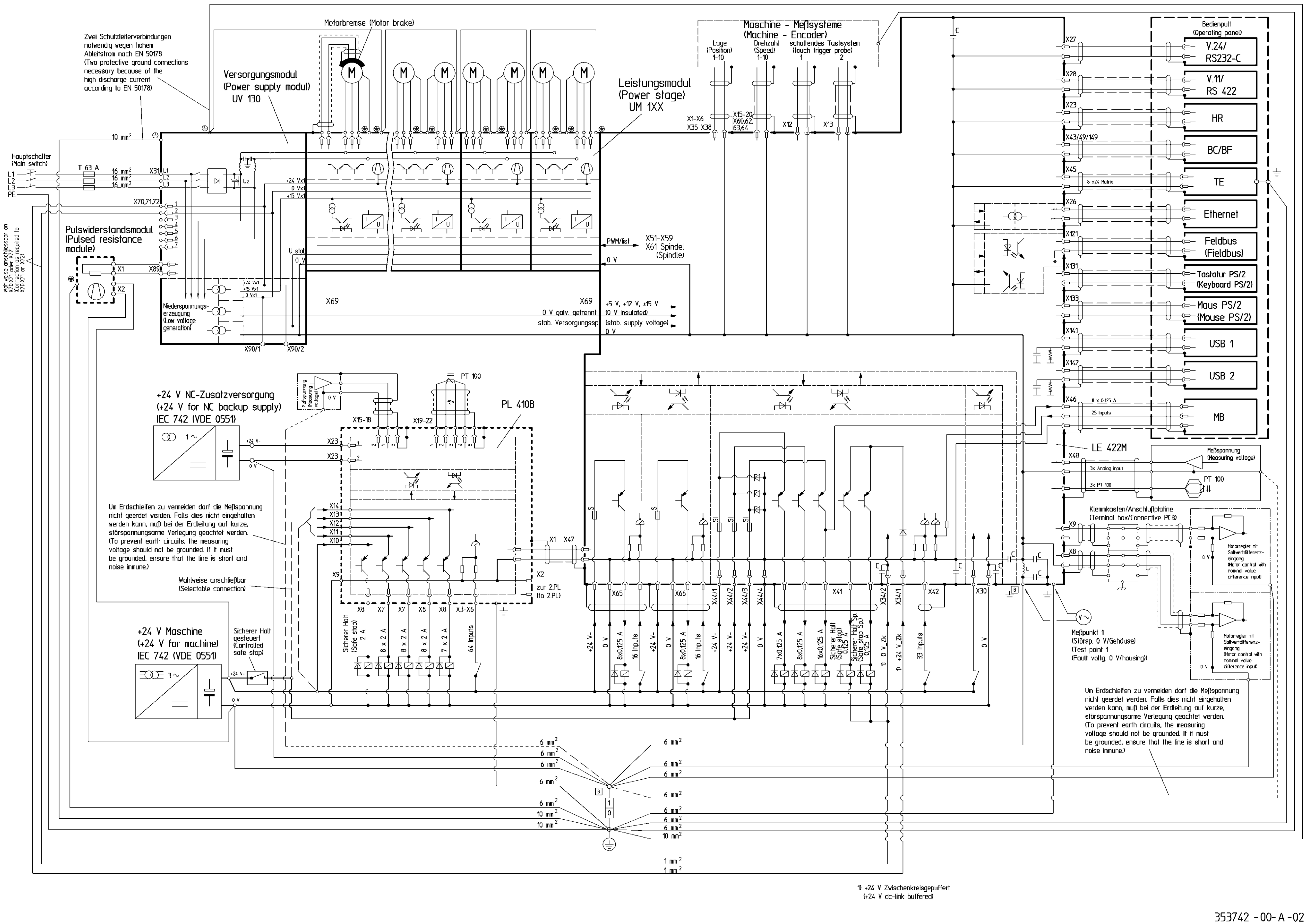
APE 652



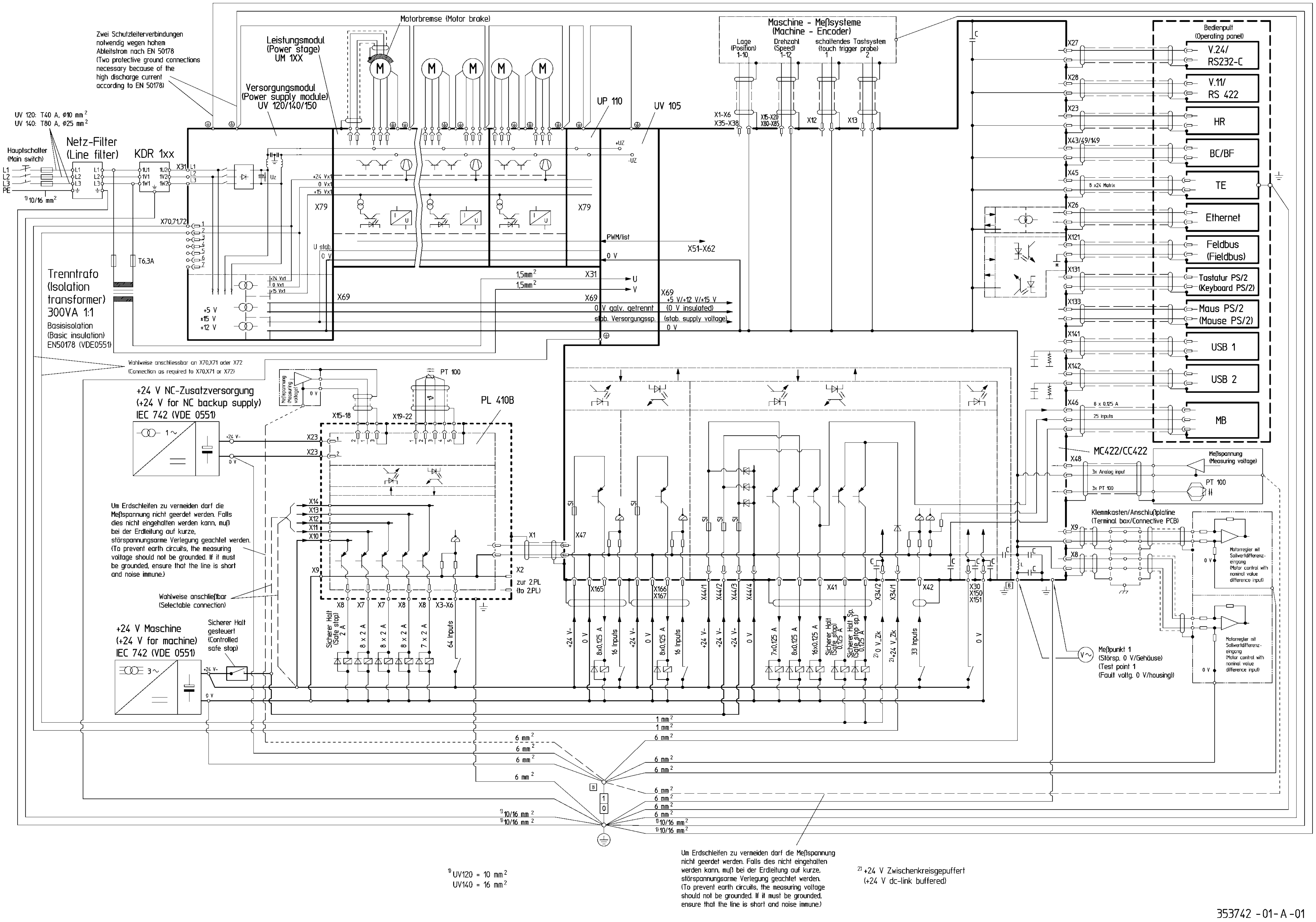
40
22
M4
M4x20 DIN 912/ISO 4762
6
≈21.5
Ø 4.7
Ø 8.5
HEIDENHAIN
163
175
≈21.5
52
14
31.5
16
28
57
58.5
80

3.30 Grounding Diagrams

3.30.1 Grounding Diagram for iTNC 530 with Modular Non-Regenerative HEIDENHAIN Inverter System



3.30.2 Grounding Diagram for iTNC 530 with Modular Regenerative HEIDENHAIN Inverter System



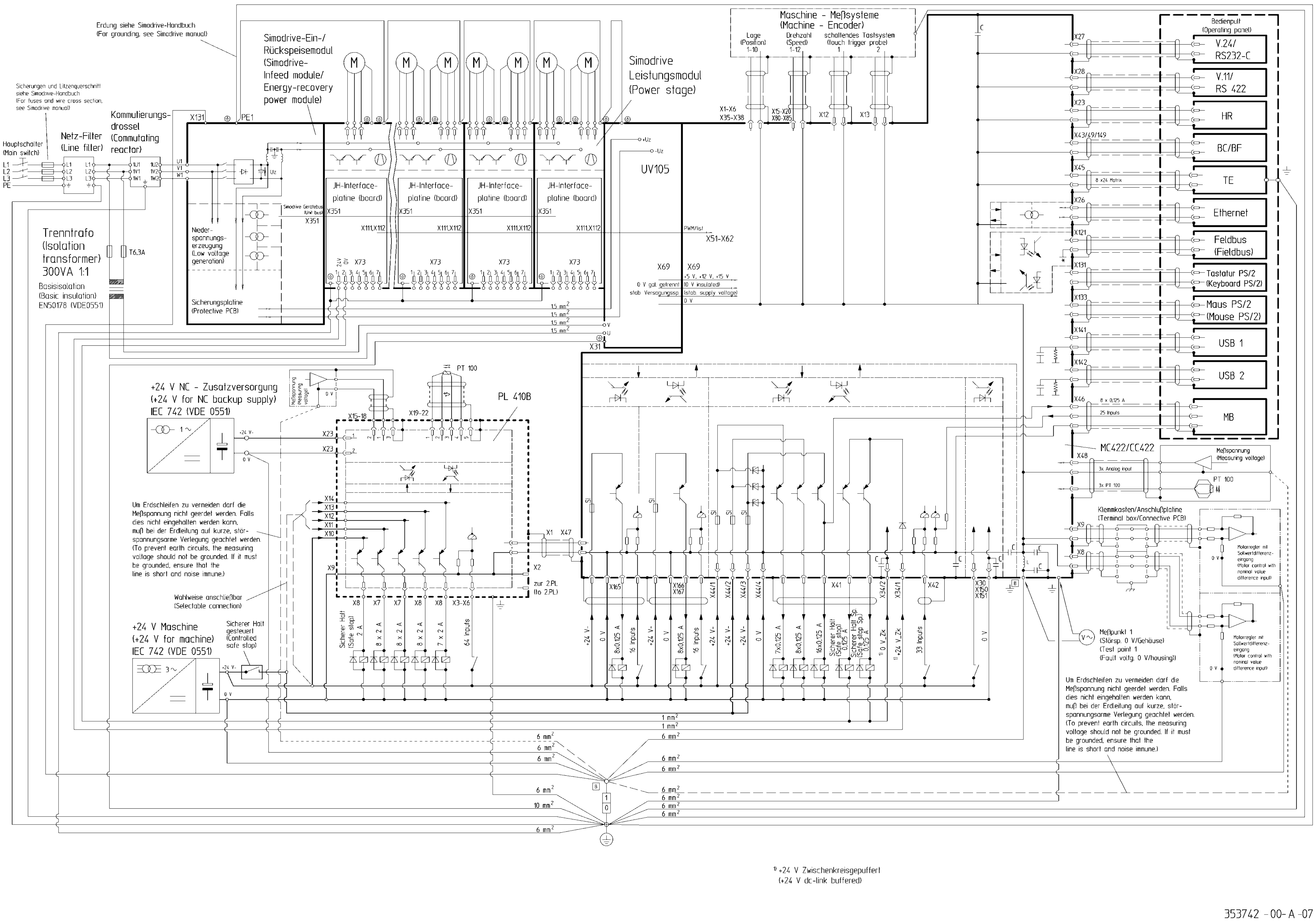
In preparation

In preparation

In preparation

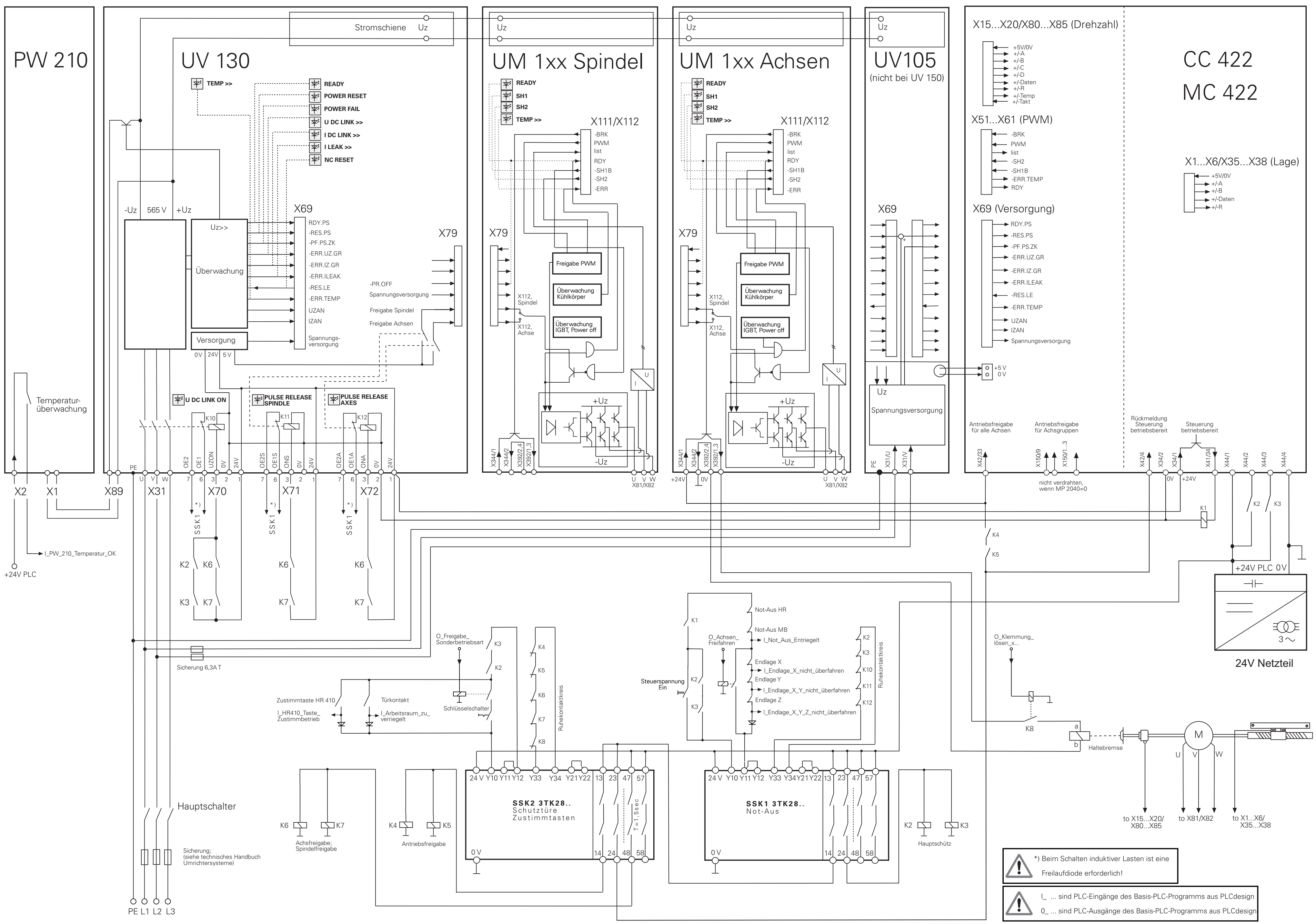
In preparation

3.30.7 Grounding Diagram for iTNC 530 with SIMODRIVE Inverter System

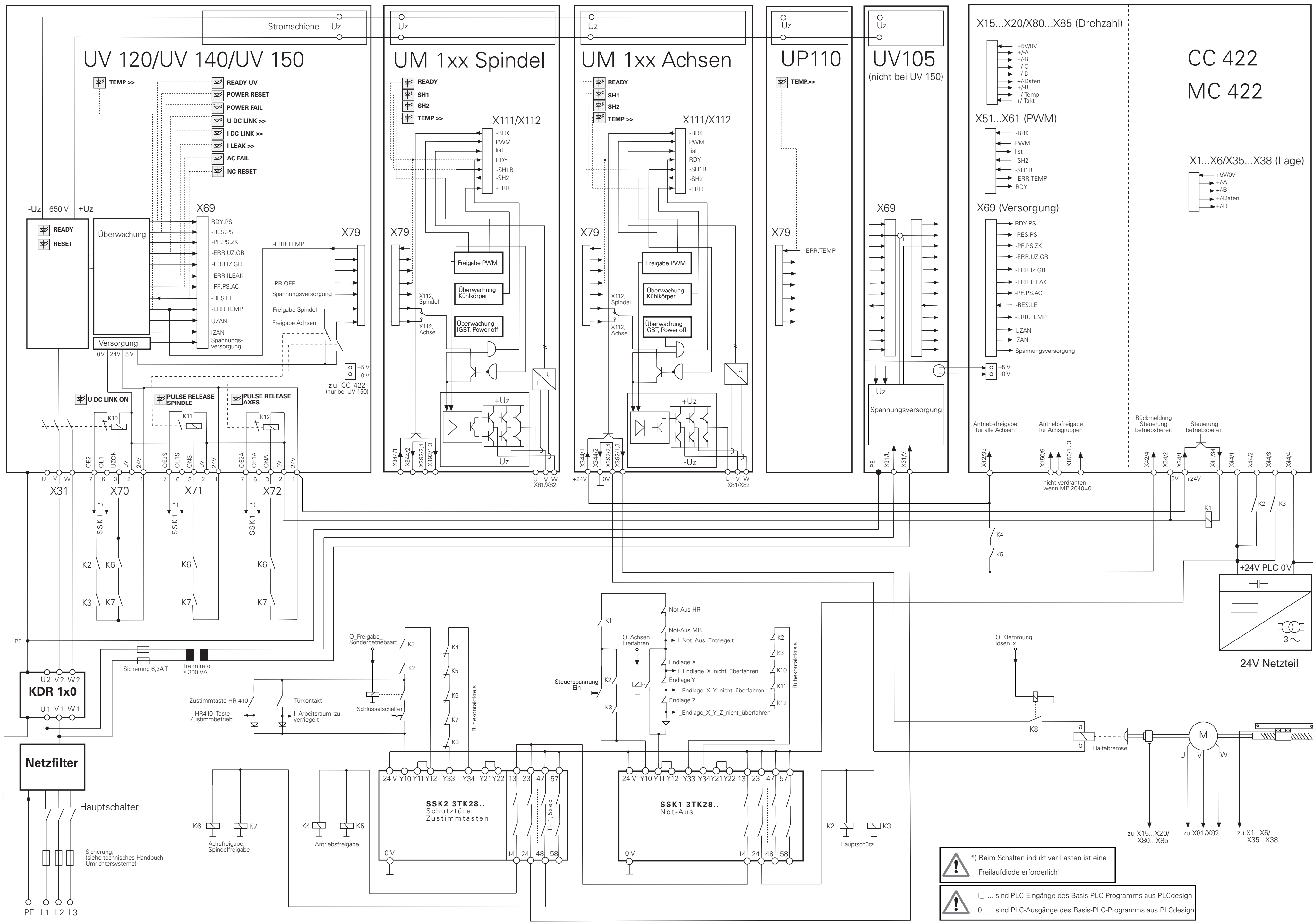


3.31 Basic Circuit Diagrams

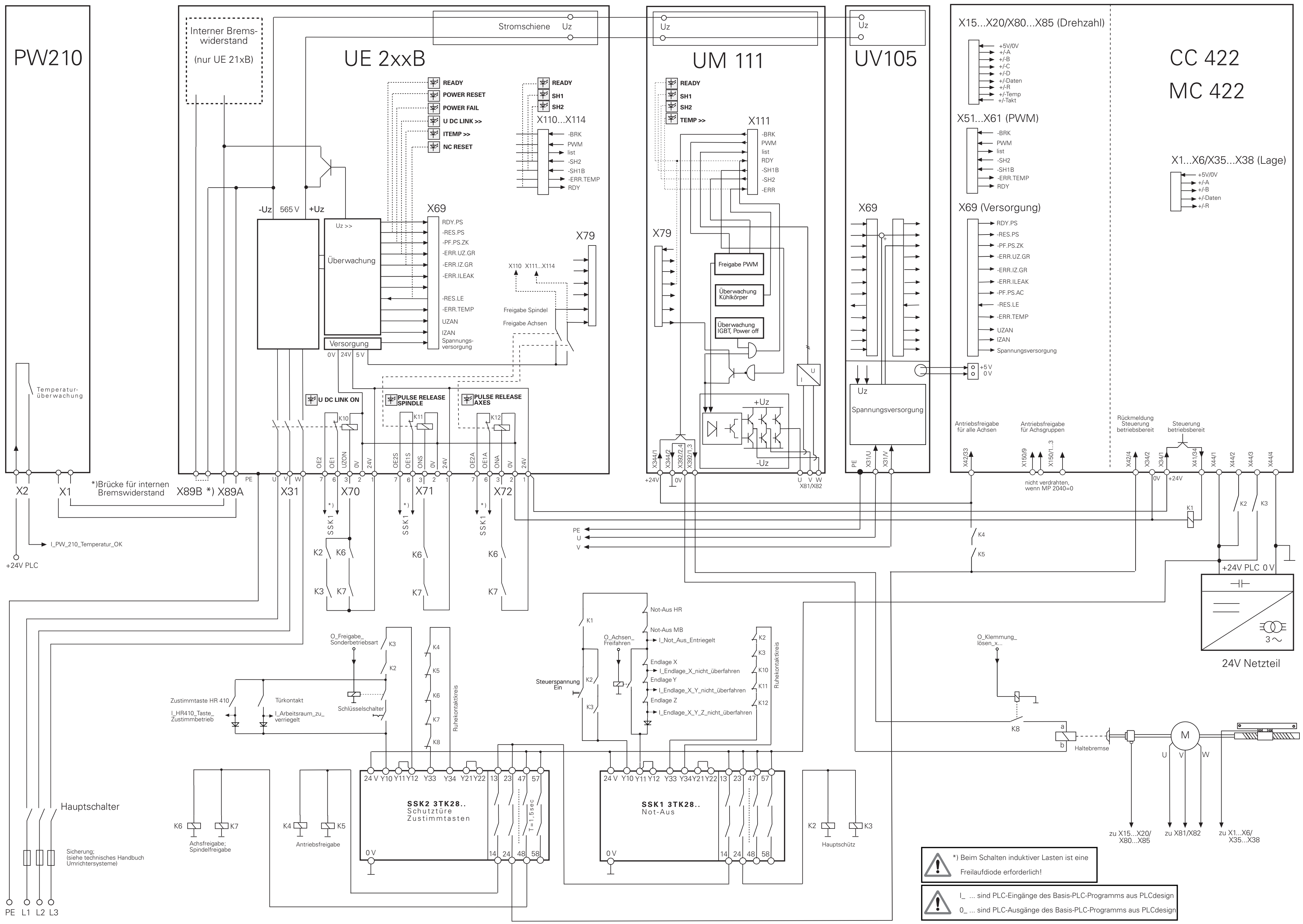
3.31.1 Basic Circuit Diagram for iTNC 530 with Modular Non-Regenerative HEIDENHAIN Inverter System



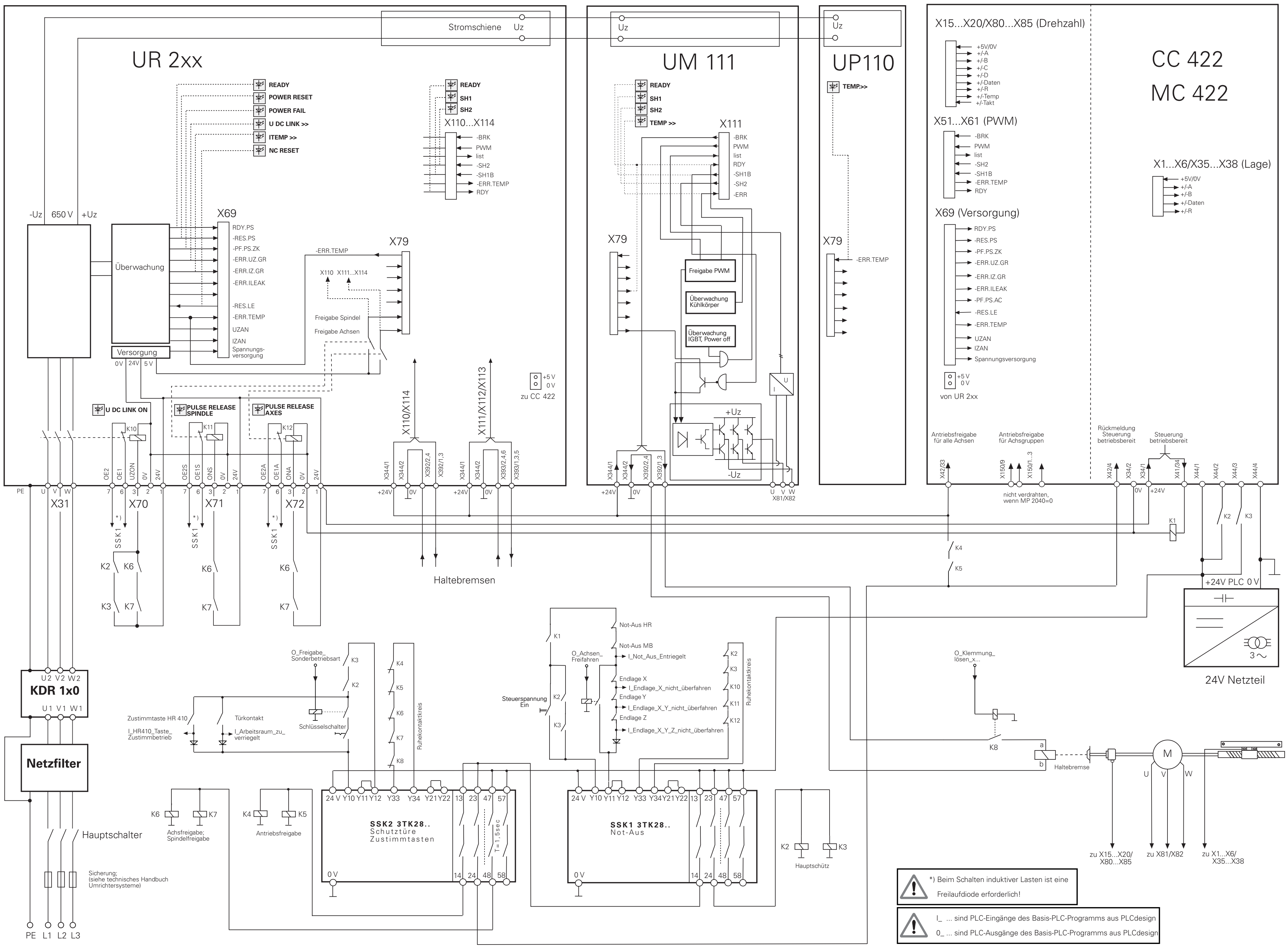
3.31.2 Basic Circuit Diagram for iTNC 530 with Modular Regenerative HEIDENHAIN Inverter System



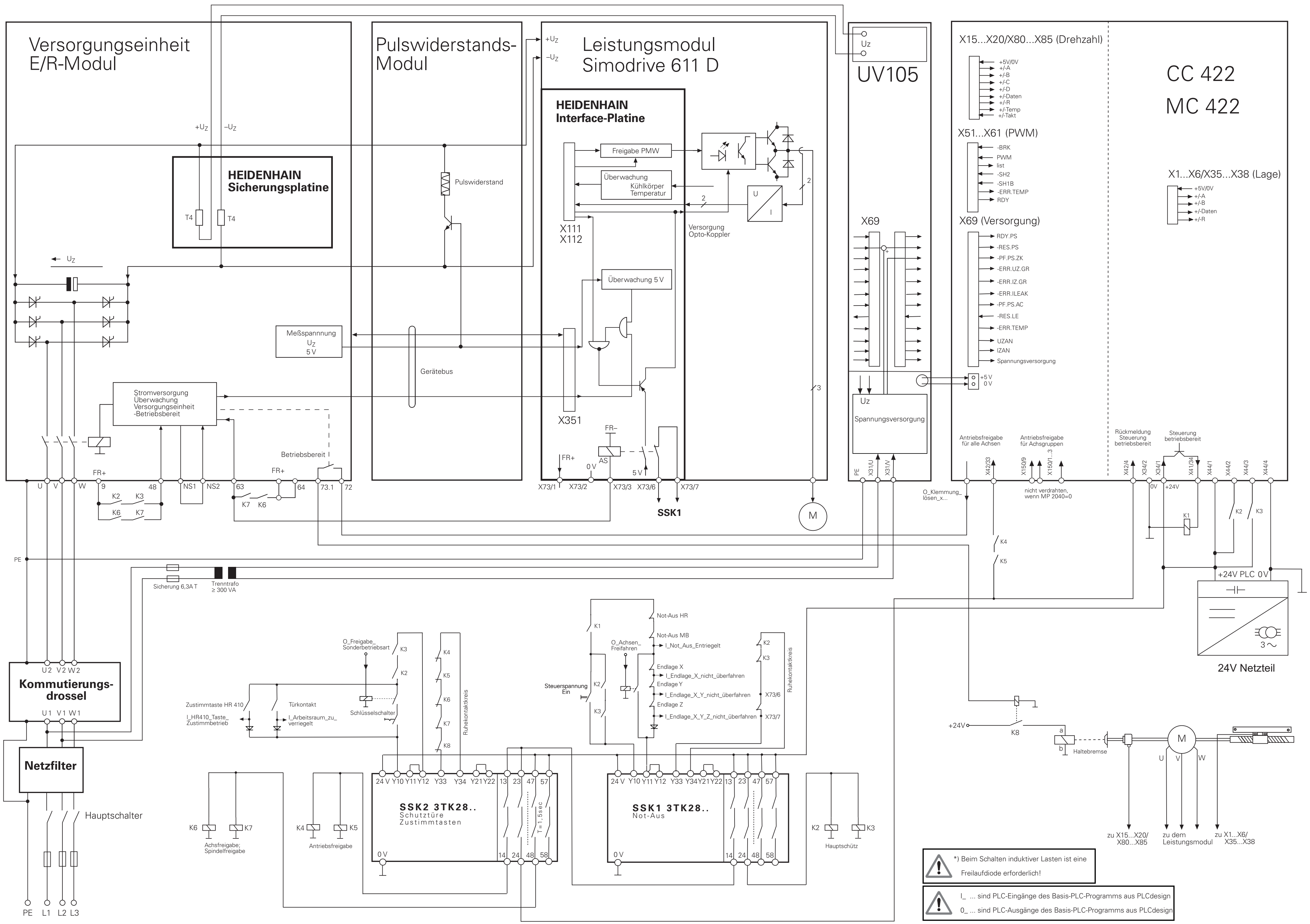
3.31.3 Basic Circuit Diagram for iTNC 530 with UE 2xxB Non-Regenerative HEIDENHAIN Compact Inverter



3.31.4 Basic Circuit Diagram for iTNC 530 with UR 2xxB Regenerative HEIDENHAIN Compact Inverter

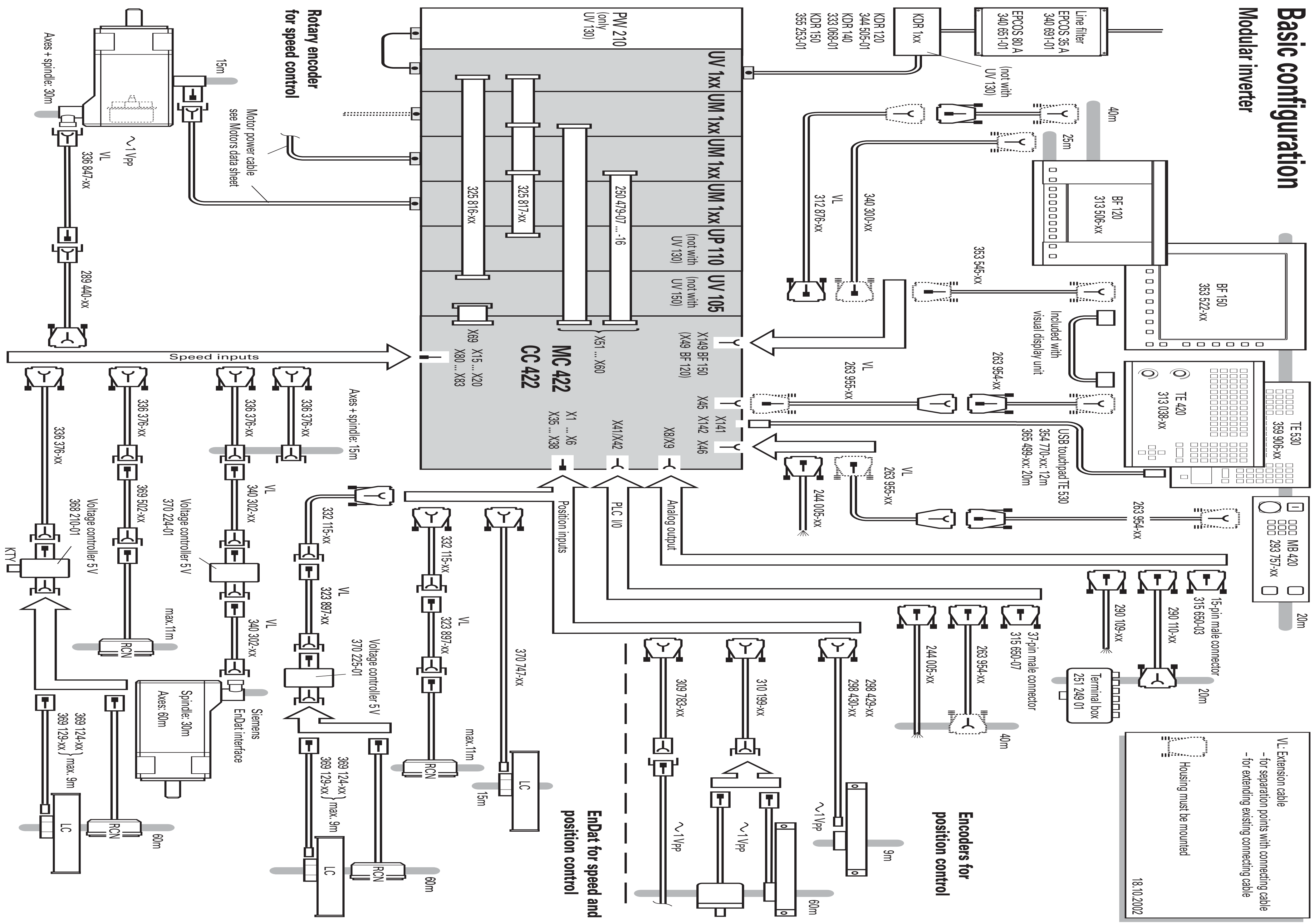


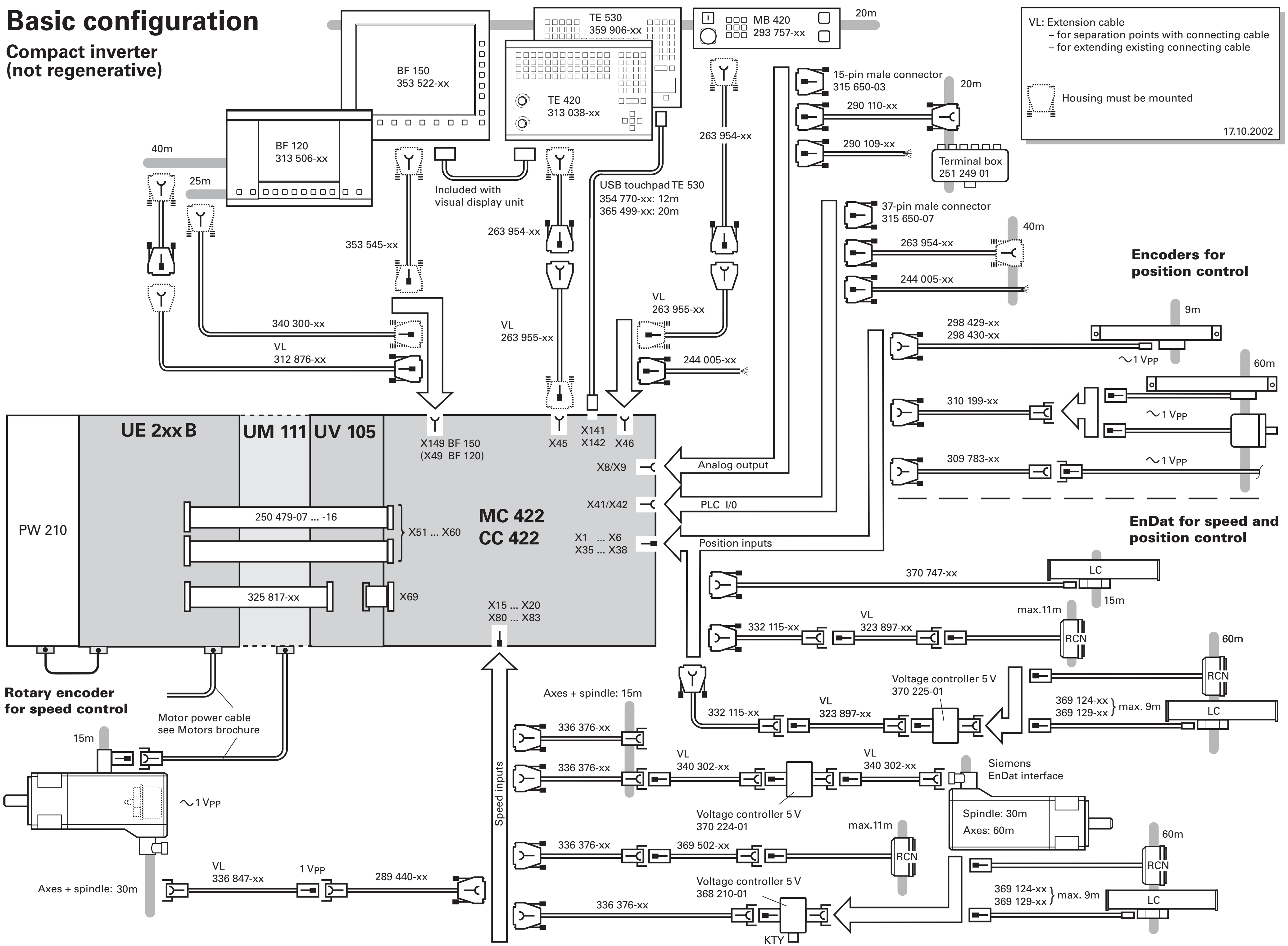
3.31.5 Basic Circuit Diagram for iTNC 530 with SIMODRIVE Inverter System

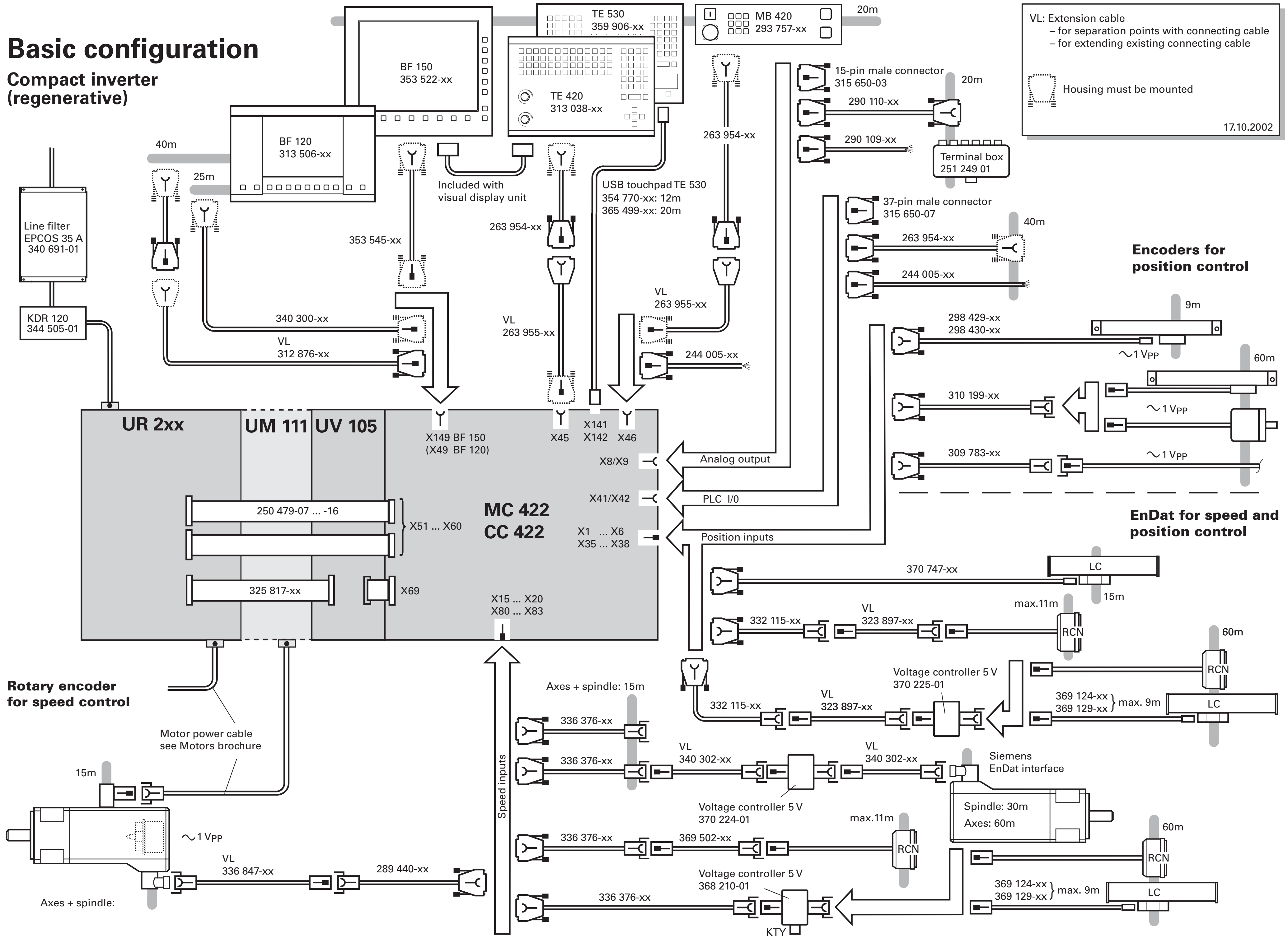


3.32 Cable Overviews

3.32.1 Cable Overview for iTNC 530 with Modular HEIDENHAIN Inverter System – Basic Configuration

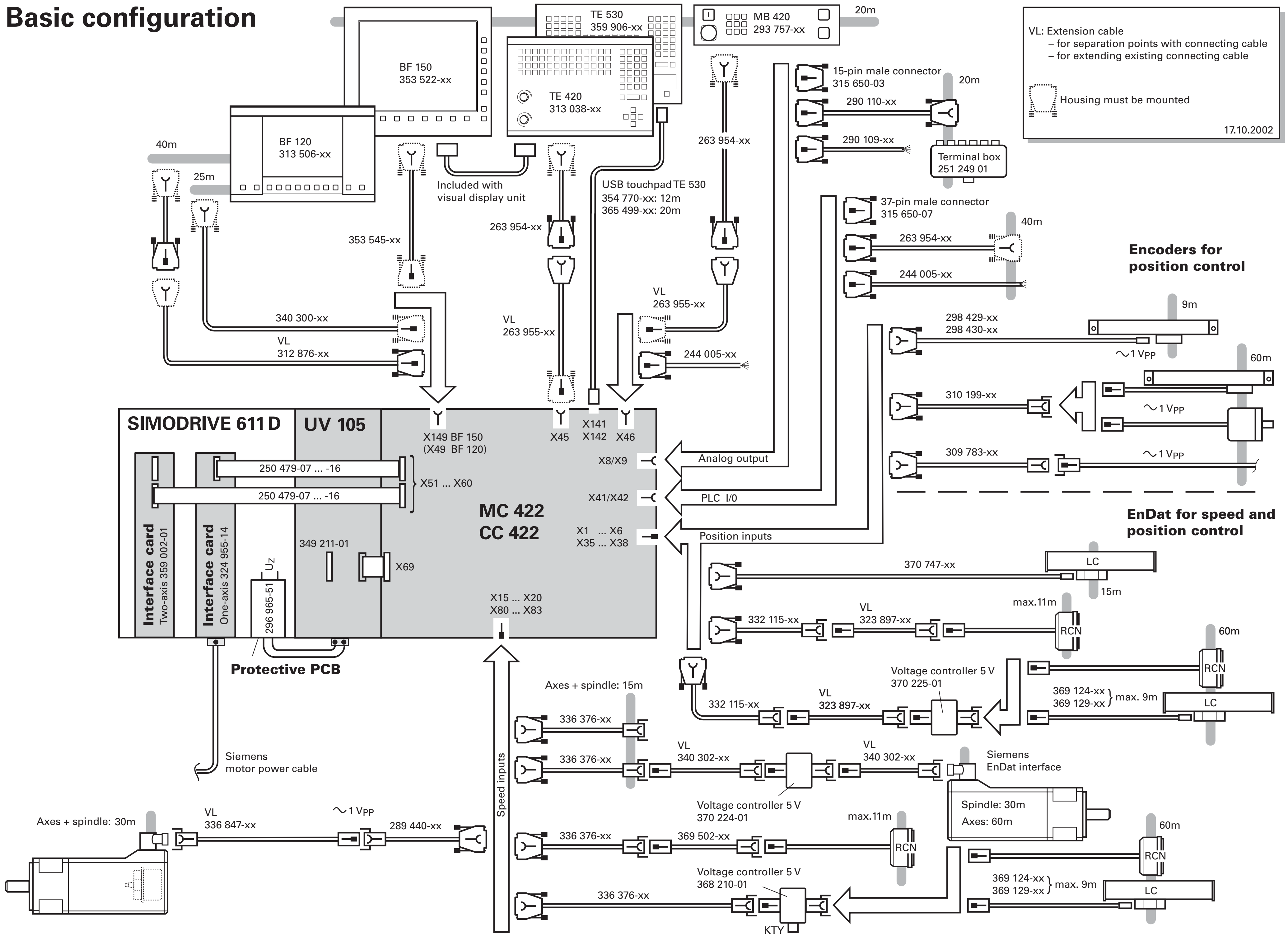


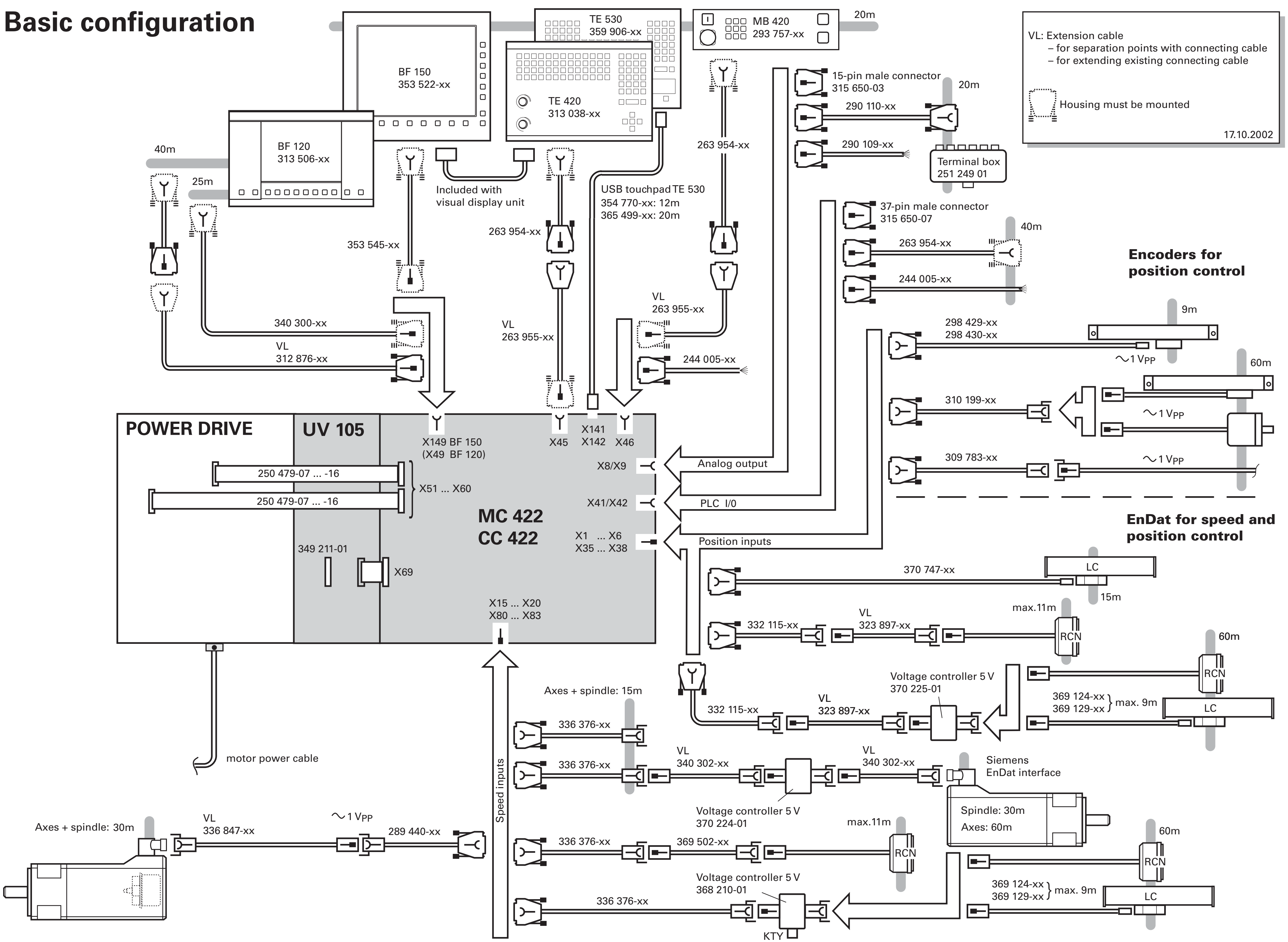

Basic configuration
Compact inverter
(not regenerative)
BF 150
353 522-xx
TE 530
359 906-xx
MB 420
293 757-xx
20m
TE 420
313 038-xx
BF 120
313 506-xx
40m
25m
VL: Extension cable
– for separation points with connecting cable
– for extending existing connecting cable
Housing must be mounted
17.10.2002
15-pin male connector
315 650-03
20m
290 110-xx
290 109-xx
Terminal box
251 249 01
263 954-xx
Included with
visual display unit
353 545-xx
263 954-xx
USB touchpad TE 530
354 770-xx: 12m
365 499-xx: 20m
37-pin male connector
315 650-07
40m
263 954-xx
244 005-xx
Encoders for
position control
VL
263 955-xx
340 300-xx
VL
312 876-xx
VL
263 955-xx
244 005-xx
298 429-xx
298 430-xx
9m
1 VPP
60m
310 199-xx
1 VPP
309 783-xx
1 VPP
UE 2xx B
UM 111
UV 105
X149 BF 150
(X49 BF 120)
X45
X141
X142
X46
X8/X9
Analog output
X41/X42
PLC I/0
X1 ... X6
X35 ... X38
Position inputs
MC 422
CC 422
PW 210
250 479-07 ... -16
X51 ... X60
325 817-xx
X69
X15 ... X20
X80 ... X83
EnDat for speed and
position control
LC
370 747-xx
15m
max.11m
332 115-xx
VL
323 897-xx
RCN
60m
RCN
Voltage controller 5 V
370 225-01
332 115-xx
VL
323 897-xx
369 124-xx
369 129-xx
max. 9m
LC
Rotary encoder
for speed control
Motor power cable
see Motors brochure
15m
1 VPP
Axes + spindle: 15m
336 376-xx
336 376-xx
VL
340 302-xx
VL
340 302-xx
Siemens
EnDat interface
Spindle: 30m
Axes: 60m
Voltage controller 5 V
370 224-01
Speed inputs
max.11m
336 376-xx
369 502-xx
RCN
60m
RCN
Axes + spindle: 30m
VL
336 847-xx
1 VPP
289 440-xx
Voltage controller 5 V
368 210-01
336 376-xx
KTY
369 124-xx
369 129-xx
max. 9m
LC



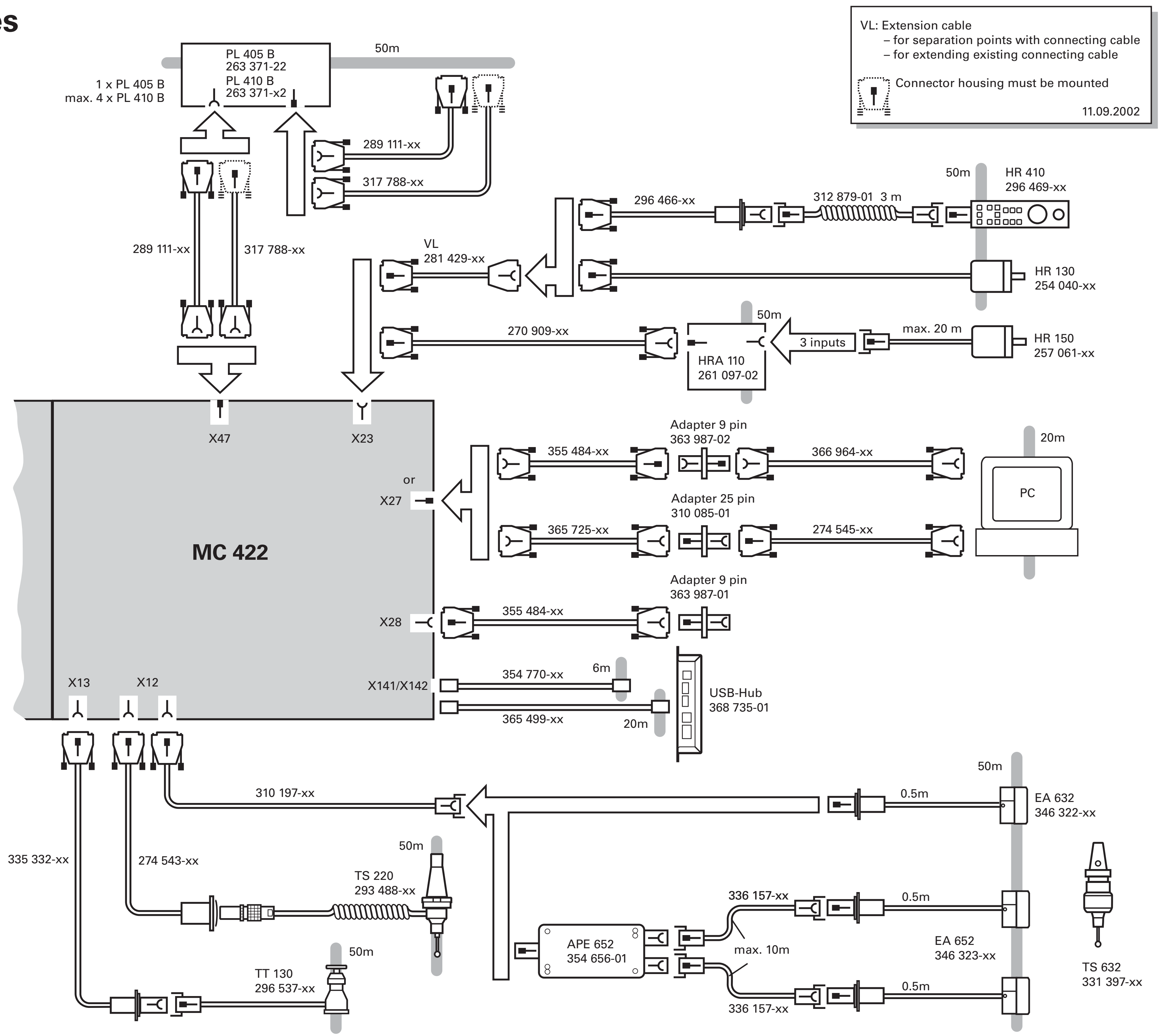
Basic configuration
Compact inverter (regenerative)
BF 150 353 522-xx
TE 530 359 906-xx
MB 420 293 757-xx
20m
TE 420 313 038-xx
BF 120 313 506-xx
40m
25m
VL: Extension cable
– for separation points with connecting cable
– for extending existing connecting cable
Housing must be mounted
17.10.2002
Line filter EPCOS 35 A 340 691-01
KDR 120 344 505-01
353 545-xx
Included with visual display unit
263 954-xx
USB touchpad TE 530
354 770-xx: 12m
365 499-xx: 20m
340 300-xx
VL 312 876-xx
VL 263 955-xx
244 005-xx
15-pin male connector 315 650-03
290 110-xx
290 109-xx
Terminal box 251 249 01
37-pin male connector 315 650-07
40m
244 005-xx
Encoders for position control
9m
298 429-xx
298 430-xx
1 VPP
60m
310 199-xx
309 783-xx
UR 2xx
UM 111
UV 105
X149 BF 150
(X49 BF 120)
X45
X141
X142
X46
X8/X9
Analog output
X41/X42
PLC I/0
X1 ... X6
X35 ... X38
Position inputs
MC 422
CC 422
250 479-07 ... -16
X51 ... X60
325 817-xx
X69
X15 ... X20
X80 ... X83
EnDat for speed and position control
370 747-xx
LC
15m
332 115-xx
VL 323 897-xx
max.11m
RCN
Voltage controller 5 V 370 225-01
369 124-xx
369 129-xx
max. 9m
Rotary encoder for speed control
Motor power cable see Motors brochure
15m
Axes + spindle: 15m
336 376-xx
VL 340 302-xx
Siemens EnDat interface
Spindle: 30m
Axes: 60m
Voltage controller 5 V 370 224-01
369 502-xx
Speed inputs
VL 336 847-xx
289 440-xx
Axes + spindle:
Voltage controller 5 V 368 210-01
KTY

Basic configuration




Basic configuration
BF 150
353 522-xx
TE 530
359 906-xx
MB 420
293 757-xx
20m
TE 420
313 038-xx
BF 120
313 506-xx
40m
25m
Included with
visual display unit
USB touchpad TE 530
354 770-xx: 12m
365 499-xx: 20m
263 954-xx
353 545-xx
263 954-xx
VL
263 955-xx
VL
263 955-xx
340 300-xx
VL
312 876-xx
244 005-xx
15-pin male connector
315 650-03
290 110-xx
290 109-xx
20m
Terminal box
251 249 01
37-pin male connector
315 650-07
263 954-xx
244 005-xx
40m
VL: Extension cable
– for separation points with connecting cable
– for extending existing connecting cable
Housing must be mounted
17.10.2002
Encoders for
position control
298 429-xx
298 430-xx
9m
1 VPP
60m
310 199-xx
1 VPP
309 783-xx
1 VPP
POWER DRIVE
UV 105
X149 BF 150
(X49 BF 120)
X45
X141
X142
X46
X8/X9
Analog output
250 479-07 ... -16
250 479-07 ... -16
X51 ... X60
MC 422
CC 422
X41/X42
PLC I/0
X1 ... X6
X35 ... X38
Position inputs
349 211-01
X69
X15 ... X20
X80 ... X83
EnDat for speed and
position control
370 747-xx
LC
15m
max.11m
332 115-xx
VL
323 897-xx
RCN
60m
RCN
Voltage controller 5 V
370 225-01
369 124-xx
369 129-xx
max. 9m
LC
332 115-xx
VL
323 897-xx
Axes + spindle: 15m
336 376-xx
336 376-xx
VL
340 302-xx
VL
340 302-xx
Siemens
EnDat interface
motor power cable
Speed inputs
Spindle: 30m
Axes: 60m
Voltage controller 5 V
370 224-01
Axes + spindle: 30m
VL
336 847-xx
1 VPP
289 440-xx
336 376-xx
369 502-xx
max.11m
RCN
60m
RCN
Voltage controller 5 V
368 210-01
336 376-xx
369 124-xx
369 129-xx
max. 9m
LC
KTY

Accessories



4 Machine Parameters

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	Programming and Display
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 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.

4.2.1 Input Format

You can enter the input values either 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	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

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

Only for MP1054.x (linear distance of one motor revolution) and for MP7530.x (type of dimension for transformation).

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

Variable:

REF	Current position of the axis relative to the machine datum (resolution 0.0001 mm or °)

4.2.2 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.

4.2.3 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:

Symbol	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 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.
- 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.

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

The machine parameters that you can change with Module 9031 or Module 9034 are indicated with PLC in the overview.

Module 9031 Overwrite machine parameters

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 detection:

Marker	Value	Meaning
M4203	0	MP was overwritten
	1	MP could not be overwritten

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 or spawn job 7: MP is of the “string” type 8: No system memory

Error detection:

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.

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
                  6: Call was not in a submit or spawn job
                  7: MP is of the "string" type
                  8: No system memory
```

Error detection:

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 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.

> **Warning**
>
> 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!

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
```

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!

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 3
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

4.3 List of Machine Parameters

4.3.1 Encoders and Machines

MP	Function and input	Software version and behavior	Page
MP10	Active axes Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Axis not active 1: Axis active		6 – 3
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	PLC RUN	6 – 10
MP20.0	Absolute position of the 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	PLC RUN	6 – 10
MP21.0	No function		
MP21.1	Amplitude of encoder signals		
MP21.2	Edge separation of encoder signals		
MP100	Designation of axes Format: -wvucbazyxWVUCBAZYX Input: Characters 1 to 9 from the right represent axes 1 to 9	PLC RUN	6 – 3, 6 – 26
MP100.0	Traverse range 1		
MP100.1	Traverse range 2		
MP100.2	Traverse range 3		
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	RESET	6 – 13

MP	Function and input	Software version and behavior	Page
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	REF	6 – 15, 6 – 188
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	6 – 13
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	6 – 15, 6 – 191
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: Nonfunctional 0: 1 V_{PP} 1: 11 μA_{PP}	RESET	6 – 8
MP115.1	Reserved Format: %xxxxxxxxxxx Input: Enter %00000000000		
MP115.2	Input frequency of the 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: Nonfunctional With 1 V_{PP}: 0: 33 kHz 1: 350 kHz With 11 μA_{PP}: 0: 33 kHz 1: 150 kHz		

MP	Function and input	Software version and 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	PLC RUN	6 – 13
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	PLC RUN	6 – 15
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	PLC RUN	
MP130.x	Y index of the machine parameters MP2xxx.y for the axes Input: 0 to 12	PLC RUN	6 – 13
MP131.x	Y index of the machine parameters MP2xxx.y for the spindle(s) in operating mode 0 Input: 0 to 12	PLC RUN	6 – 15
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	6 – 15
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	RESET	6 – 9

MP	Function and input	Software version and behavior	Page
MP331.x	Distance for the number of signal periods in MP332 Input: 0.0001 to 99.999 9999 [mm] or [°]	PLC RUN REF	6 – 7
MP332.x	Number of signal periods for the distance in MP331 Input: 1 to 16 777 215	PLC RUN REF	6 – 7
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	6 – 7
MP340.x	Interpolation factor for external interpolation Input: 0 to 99 0 = 1: No external interpolation	RESET	6 – 7
MP410	Assignment of axis keys IV and V Input: Axis designation XYZABCUVWxyzabcuvw–	PLC RUN	6 – 3
MP410.3	Axis key IV		
MP410.4	Axis key V		
MP420.x	Hirth coupling Input: 0: No Hirth coupling 1: Hirth coupling	PLC RUN	7 – 164
MP430.x	Prescribed increment for Hirth coupling Input: 0.0000 to 30.0000 [°]	PLC RUN	7 – 164
MP710.x	Backlash compensation Input: -1.0000 to +1.0000 [mm] or [°]	PLC RUN	6 – 34
MP711.x	Height of peaks during circular movement (only analog) Input: -1.0000 000 to +1.0000 999 [mm] (digital: 0)	PLC RUN	6 – 45
MP712.x	Compensation value per control loop cycle time Input: 0.000 000 to 99.999 999 [mm] (digital: 0)	PLC RUN	6 – 45
MP715.x	Height of peaks during circular movement (only analog) with M105 Input: -1.0000 to +1.0000 [mm] (digital: 0)	PLC RUN	6 – 45
MP716.x	Compensation value per control loop cycle time with M105 Input: 0.000 000 to 99.999 999 [mm] (digital: 0)	PLC RUN	6 – 45
MP720.x	Linear axis error compensation Input: -1.000 to +1.000 [mm/m]	PLC RUN	6 – 36
MP730	Selection of linear/nonlinear axis error compensation Format: %xxxxxxxxxxxxxx Input: Bits 0 to 3 correspond to axes 1 to 14: 0: Linear axis error compensation 1: Nonlinear axis error compensation	PLC RUN	6 – 36, 6 – 41

MP	Function and input	Software version and behavior	Page
MP750.x	Reversal error Input: -1.0000 to +1.0000 [mm] or [°]	PLC RUN	6 – 35
MP752.x	Compensation time for reversal error Input: 0 to 1000 [ms]	PLC RUN	6 – 35
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	7 – 5
MP812	Activate software limit switches for tilting axes with modulo display, M94 and encoders with EnDat interface Format: %xxxxxxxxxxxxxx Input: Bits 0 to 3 correspond to axes 1 to 14: 0: Software limit switch not active 1: Software limit switch active	RESET	7 – 5
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	6 – 88
MP855.x	Synchronization monitoring Input: 0 to 100.0000 [mm] 0: Monitoring not active	PLC RUN	6 – 90
MP860.x	Datum for synchronous control Input: 0: Datum at position after switch-on 1: Datum at reference marks 2: Axis is torque slave axis	PLC RUN	6 – 90, 6 – 95
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	6 – 23
MP911.x	Positive software limit switches, traverse range 2 Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	6 – 23
MP912.x	Positive software limit switches, traverse range 3 Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	6 – 23
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	6 – 23

MP	Function and input	Software version and behavior	Page
MP921.x	Negative software limit switches, traverse range 2 Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	6 – 23
MP922.x	Negative software limit switches, traverse range 3 Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	6 – 23
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	7 – 32
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	7 – 40
MP960.x	Machine Datum Input: –1.79769313486E+308 to +1.79769313486E+308 [mm] or [°] Values with respect to the scale reference point	PLC RUN REF	6 – 103, 7 – 32

4.3.2 Positioning

MP	Function and input	Software version and behavior	Page
MP1010.x	Rapid traverse Input: 10 to 300 000 [mm/min or °/min]	PLC RUN	6 – 132
MP1011	Limit of rapid traverse on the path Input: 10 to 300 000 [mm/min or °/min]	340 420-05 PLC RUN	6 – 132
MP1020.x	Manual feed Input: 10 to 300 000 [mm/min]	PLC RUN	6 – 132
MP1030.x	Positioning window Input: 0.0001 to 2.0000 [mm]	PLC RUN	6 – 172
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		6 – 9
MP1050.x	Analog axes: Analog voltage at rapid traverse Input: 1.000 to 9.000 [V] Digital axes: without function Input: 1	PLC RUN	6 – 132
MP1054.x	Linear distance of one motor revolution Input: Analog axes: without function Digital axes: 0 to 100.000 [mm] or [°]		6 – 171
MP1060.x	Acceleration Input: 0.001 to 100.000 [m/s or 1000°/s^2]	PLC RUN	6 – 120
MP1061	Limitation of the path acceleration Input: 0.001 to 100.000 [m/s or 1000°/s^2]	340 420-05 PLC RUN	6 – 120
MP1070	Radial acceleration Input: 0.001 to 100.000 [m/s or 1000°/s^2]	PLC RUN	6 – 163
MP1080.x	Analog axes: Integral factor for offset adjustment Input: Enter 0 to 65 535 Digital axes: Nonfunctional Input: 0	PLC RUN	6 – 162
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 0.1 to 1000.0 [m/s or 1000°/s]	340 420-02 PLC RUN	6 – 120

MP	Function and input	Software version and behavior	Page
MP1087.x	Maximum permissible axis-specific jerk for Manual mode Input: 0.1 to 1000.0 [m/s or 1000°/s]	PLC RUN	6 – 120
MP1089.x	Maximum permissible axis-specific jerk for Pass Over Reference Point mode Input: 0.1 to 1000.0 [m/s or 1000°/s]	PLC RUN	6 – 120
MP1090	Maximum permissible jerk on the tool path Input: 0.1 to 1000.0 [m/s or 1000°/s]	PLC RUN	6 – 120
MP1090.0	With machining feed rate		
MP1090.1	Beginning with feed rate from MP1092		
MP1092	Feed rate threshold from which MP1090.1 becomes effective Input: 10 to 300 000 [mm/min]	PLC RUN	6 – 120
MP1094	HSC filter Input: 0: HSC filter inactive 0.1 to 166.0: Cutoff frequency for HSC filter		6 – 120
MP1095	Nominal position value filter Input: 0: Single filter 1: Double filter	PLC RUN	6 – 120
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 Input: 0: No nominal position value filter 0.001 to 3.000 [mm]	PLC RUN	6 – 121, 6 – 164
MP1097.x	Maximum permissible axis-specific jerk (single/HSC filter) Input: 0.1 to 1000.0 [m/s or 1000°/s]	PLC RUN	6 – 121
MP1098.x	Maximum permissible axis-specific jerk (double/HSC filter) Input: 0.1 to 1000.0 [m/s or 1000°/s]	PLC RUN	6 – 121
MP1099	Minimum filter order Input: 0 to 20	PLC RUN	6 – 121
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	6 – 172
MP1140.x	Threshold at which the movement monitoring goes into effect 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	6 – 171

MP	Function and input	Software version and behavior	Page
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	6 – 171
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 [°]	340 420-05 PLC RUN	6 – 169
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	6 – 134, 6 – 167; 6 – 169
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]		
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	6 – 103
MP1330.x	Velocity for traversing the reference marks Input: 80 to 300 000 [mm/min]	PLC RUN	6 – 103
MP1331.x	Velocity for leaving the reference mark end position for axes 1 to 9 (only for rotary encoders MP1350 = 2) Input: 10 to 300 000 [mm/min]	PLC RUN	6 – 103
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	6 – 103
MP1350.x	Type of reference-mark traverse 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 over fast PLC input	PLC RUN REF	6 – 103

MP	Function and input	Software version and behavior	Page
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	6 – 103
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	6 – 103
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	6 – 104
MP1391	Velocity feedforward control in the MANUAL and HANDWHEEL operating modes 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	6 – 47, 6 – 124
MP1392	Velocity feedforward in the POSITIONING WITH MANUAL DATA INPUT, PROGRAM RUN SINGLE BLOCK and PROGRAM RUN FULL SEQUENCE operating modes 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	6 – 124
MP1396.x	Feedback control with velocity semifeedforward Input: 0.001 to 0.999 1: Velocity feedforward control	PLC RUN	6 – 130

4.3.3 Operation with Velocity Feedforward Control

MP	Function and input	Software version and 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	6 – 169
MP1420.x	Position monitoring for operation with velocity feedforward control (EMERGENCY STOP) Input: 0.0010 to 30.0000 [mm] Recommended: 2 mm	PLC RUN	6 – 169
MP1510.x	k_V factor for velocity feedforward control Input: 0.100 to 1000.000 [(m/min)/mm]	PLC RUN	6 – 128
MP1511.x	Factor for static friction compensation Input: 0 to 16 777 215 [s]	PLC RUN	6 – 47
MP1512.x	Limitation of the amount of the static friction compensation Input: 0 to 16 777 215 [counting steps]	PLC RUN	6 – 47
MP1513.x	Feed-rate limitation for static friction compensation Input: 0 to 300 000 [mm/min]	PLC RUN	6 – 47
MP1515.x	k_V factor for velocity feedforward control effective after M105 Input: 0.100 to 20.000 [(m/min)/mm]	PLC RUN	6 – 128
MP1516.x	k_V factor for velocity semifeedforward control Input: 0.100 to 20.000 [(m/min)/mm]	PLC RUN	6 – 130
MP1521	Transient response during acceleration and deceleration Input: 1 to 255 [ms] 0: Function inactive	PLC RUN	6 – 121

4.3.4 Operation with Following Error (Servo Lag)

MP	Function and input	Software version and 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	6 – 169
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	6 – 169
MP1810.x	k_V factor for control with following error Input: 0.100 to 20.000 [(m/min)/mm]	PLC RUN	6 – 126
MP1815.x	k_V factor for control with following error effective after M105 Input: 0.100 to 20.000 [(m/min)/mm]	PLC RUN	6 – 126
MP1820.x	Multiplier for the k_V factor Input: 0.001 to 1.00000	PLC RUN	6 – 133
MP1830.x	Characteristic curve kink point Input: 0.000 to 100.000 [%]	PLC RUN	6 – 133

4.3.5 Integrated Speed and Current Control

MP	Function and input	Software version and behavior	Page
MP2040	Groups for drive enabling through X150/X151 Format: %xxxxxxxxxxxxxx Input: 0: Axis not assigned 1: Axis assigned	PLC RUN	6 – 147
MP2040.0-2	Axis group 1 to 3		
MP2040.3-7	Reserved, enter %00000000000000		
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		6 – 147
MP2100.x	Power module model Input: Name of the selected power module (entered by the iTNC)	RESET	6 – 230
MP2150	Signal for powerfail Input: 0: AC fail 1: Powerfail and AC fail 2: Neither powerfail nor AC fail 3: Powerfail		6 – 174
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		6 – 160
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]		6 – 147
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	RESET	6 – 233
MP2190	dc link voltage U_Z Input: 0 to 10 000 [V] HEIDENHAIN inverters: UE 2xx, UE 2xxB, UV 130: 565 V UR 2xx, UV 120, UV 140, UV 150: 650 V		6 – 239

MP	Function and input	Software version and behavior	Page
MP2195	Suppress error messages of the HEIDENHAIN supply units Format: %xxxxxxxx 0: Error message is not suppressed 1: Error message is suppressed Input: Bit 0: Reserved Bit 1: ERR.UZ.GR signal Bit 2: ERR.TMP signal Bit 3: Reserved Bit 4: ERR.IZ.GR signal Bit 5: RDY.PS signal Bit 6: ERR.ILEAK signal Bit 7: Reserved	340 420-06	6 – 181
MP2200.x	Motor model Input: Name of the selected motor (entered by the iTNC)	RESET	6 – 230
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 RESET	6 – 230
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	6 – 230
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: Incremental linear encoder 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 (not aligned) 8: Incremental linear encoder with distance-coded reference marks (not aligned)	340 420-05 RESET	6 – 230

MP	Function and input	Software version and behavior	Page
MP2220.x	Monitoring functions Format: %xxxx 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 – Monitoring the ERR-IZ signal 0: Monitoring inactive (Non-HEIDENHAIN inverters, UE 2xx) 1: Monitoring active (All HEIDENHAIN inverters except UE 2xx) Bit 3 – Switching off the controller when the motor brakes are activated 0: Suppress vibrations 1: Vibrations are allowed	PLC RUN	6 – 158; 6 – 182, 6 – 191
MP2302.x	Reference value for I^2t monitoring of motor Input: 0 to 1 000.000 [· rated current of motor] 0: I^2t monitoring of motor switched off 1: Rated current of motor as reference value		6 – 179
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	6 – 179
MP2308.x	Time between output of the braking signal $\overline{\text{BRK}}$ and switching off of the controller (overlap time) Input: 0.001 to 0.500 [s] 0: 0.200 s	340 420-06	6 – 182
MP2312.x	Reference value for utilization of feed motors for axes 1 to 9 Input: 0 to 1000.000 [· rated current of motor] 0 or 1: Reference value is rated current of motor		6 – 180
MP2390.x	Maximum braking power Input: 0.1 to 3000.000 [kW] 0: Braking power is not limited		6 – 155
MP2392.x	Power limit Input: 0: No power limit 0.1 to 3000.000 [kW]		6 – 158
MP2394.x	Maximum braking power during a power fail Input: 0.1 to 3000.000 [kW] 0: Braking power is not limited		6 – 155
MP2396.x	Maximum torque Input: 0.1 to 30 000.0 [Nm] 0: Torque is not limited	PLC	6 – 158

MP	Function and input	Software version and behavior	Page
MP2420.x	Proportional factor of the current controller Input: 0.00 to 9999.99 [VA]		6 – 152
MP2430.x	Integral factor of the current controller Input: 0.00 to 9999.99 [V/As]		6 – 152
MP2500.x	Proportional factor of the shaft speed controller Input: 0 to 1 000 000.000 [As]	PLC RUN	6 – 139
MP2510.x	Integral factor of the shaft speed controller Input: 0 to 100 000 000 [A]	PLC RUN	6 – 139
MP2512.x	Limiting the integral factor of the speed controller Input: 0.000 to 30.000 [s] (realistic values: 0.1 to 2.0)	PLC RUN	6 – 47, 6 – 142
MP2520.x	Differential factor of the shaft speed controller Input: 0 to 1.0000 [As]	PLC RUN	6 – 140
MP2530.x	PT_2 element of the shaft speed controller (2nd-order delay) Input: 0 to 1.0000 [s]	PLC RUN	6 – 141
MP2540.x	Band-rejection filter damping Input: 0.0 to 18.0 [dB]	PLC RUN	6 – 141
MP2550.x	Band-rejection filter for center frequency Input: 0.0 to 999.9 [Hz]	PLC RUN	6 – 141
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	6 – 140
MP2590.x	Braking ramp in an emergency stop Input: 0.1 to 999.9 [rpm/ms] 0: Function inactive	PLC RUN	6 – 154
MP2600.x	Acceleration feedforward Input: 0 to 100.0000 [A/(rev/s)]	PLC	6 – 142
MP2602.x	IPC time constant T_1 Input: 0.0001 to 1.0000 [s] 0: IPC inactive	PLC RUN	6 – 144
MP2604.x	IPC time constant T_2 Input: 0.0001 to 1.0000 [s] 0: IPC inactive	PLC RUN	6 – 144
MP2606.x	Following error in the jerk phase Input: 0.000 to 10.000	PLC RUN	6 – 144
MP2610.x	Friction compensation at low speeds (effective only with velocity feedforward control) Input: 0 to 30.0000 [A] 0: No friction compensation (or axis is analog)	PLC RUN	6 – 48

MP	Function and input	Software version and 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	6 – 48
MP2620.x	Friction compensation Input: 0 to 30.0000 [A] 0: No friction compensation (or axis is analog)	PLC RUN	6 – 48
MP2630.x	Holding current Input: -30.000 to +30.000 [A]	PLC RUN	6 – 145
MP2700	Reserved, enter 0		
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	6 – 98
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	6 – 98
MP2920.x	Factor for variable torque distribution of the master-slave torque control (entry for the slave axis) Input: 0.000 to 100.000 1: Master and slave axes have identical motors	PLC	6 – 98
MP2930.x	Speed compensation ratio for master-slave torque control (entry for the slave axis) Input: -100.00 to +100.00 [%]	PLC	6 – 98

4.3.6 Spindle

MP	Function and input	Software version and 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	6 – 187
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)		7 – 191
MP3012	Feed rate from output of an analog voltage of 10 V, MP3011 = 1 Input: 0 to 300 000 [mm/min]		7 – 191
MP3013.x	Characteristic curve kink points (velocity) for output of the analog voltage with M202 Input: 10 to 300 000 [mm/min]	PLC RUN	7 – 192
MP3014.x	Characteristic curve kink points (voltage) for output of the analog voltage with M202 Input: 0.000 to 9.999 [V]	PLC RUN	7 – 192
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	6 – 200

MP	Function and input	Software version and 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	6 – 196, 7 – 194
MP3120	Zero speed permitted Input: 0: S = 0 allowed 1: S = 0 not allowed	PLC RUN	6 – 195
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	6 – 194
MP3140	Counting direction of spindle position encoder output signals Input: 0: Positive counting direction with M03 1: Negative counting direction with M03	PLC RUN	6 – 194
MP3142	Line count of the spindle position encoder Input: 100 to 9999 [lines]	PLC RUN	6 – 188
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.	PLC RUN	6 – 188
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 rpm]	PLC RUN	6 – 195

MP	Function and input	Software version and behavior	Page
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 rpm]	PLC RUN	6 – 195, 6 – 196
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 rpm]		
MP3310 MP3310.0 MP3310.1	Limitation for spindle speed override Input: 0 to 150 [%] Upper limit Lower limit	PLC RUN	6 – 198
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 rpm/ms]	PLC RUN	6 – 193
MP3412 MP3412.0 MP3412.1 MP3412.2 MP3412.3	Multiplication factor for MP3411.x Input: 0.000 to 1.999 With M05 With oriented spindle stop With tapping with floating tap holder With rigid tapping	PLC RUN	6 – 193, 6 – 204, 6 – 209, 6 – 213
MP3415 MP3415.0 MP3415.1 MP3415.2 MP3415.3	Overshoot behavior of the spindle with M03, M04 and M05 Input: 0 to 1000 [ms] With M03, M04 and M05 For oriented spindle stop With tapping With rigid tapping	PLC RUN	6 – 193, 6 – 204, 6 – 209, 6 – 213
MP3420	Spindle positioning window Input: 0 to 360.0000 [°]	PLC RUN	6 – 204
MP3430	Deviation of the reference mark from the desired position (spindle preset) Input: 0 to 360 [°]	PLC RUN	6 – 204
MP3440.0-7	k_V factor for spindle orientation for gear ranges 1 to 8 Input: 0.1 to 10 [(1000°/ min) /°]	PLC RUN	6 – 204

MP	Function and input	Software version and behavior	Page
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	6 – 188
MP3451.0-7	Number of spindle revolutions for gear ranges 1 to 8 Input: 0 to 65 535 0: No transmission	PLC RUN	6 – 188
MP3510.0-7	Rated speed for the gear ranges 1 to 8 Input: 0 to 99 999.999 [rpm]	PLC RUN	6 – 195
MP3515.0-7	Maximum spindle speed for gear ranges 1 to 8 Input: 0 to 99 999.999 [rpm]	PLC RUN	6 – 198
MP3520.0 MP3520.1	Speed activation through marker M4011 Input: 0 to 99 999.999 [rpm] Spindle speed for oriented stop Input: 0 to 99 999.999 [rpm]	PLC RUN	6 – 204, 6 – 207

4.3.7 Integral PLC

MP	Function and input	Software version and behavior	Page
MP4000.0-15	Options for the conditional compilation of the PLC program		8 – 17
MP4020	PLC Functions Format: %xxxxxxxxxxxxx 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	RESET	6 – 134, 6 – 215, 7 – 113, 7 – 160
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	7 – 154
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 and behavior	Page
MP4040	Set PLC output after shutdown	340 420-03 PLC RUN	7 – 55
MP4041	Time after shutdown until setting of the PLC output from MP4042 Input: 0 to 1000 [s]	340 420-03 PLC RUN	7 – 55
MP4042	PLC output to be set after shutdown Input: 0 to 31	340 420-03 PLC RUN	7 – 55
MP4050.0-8	Traverse distance for lubrication of axes 1 to 9 Input: 0 to 99 999.999 [m/s]	PLC RUN	6 – 24
MP4070	Compensation amount per PLC cycle for lagged-tracking axis error compensation Input: 0.0001 to 0.5000 [mm]	PLC RUN	6 – 42
MP4110.0-47	Run time PLC timer T0 to T47 Input: 0 to 1 000 000.000 [s]	PLC RUN	8 – 48
MP4111.96-x	Run time PLC timer T96 to x (defined in OEM.SYS) Input: 0 to 1 000 000.000 [s]	PLC RUN	8 – 48
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	8 – 51
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 255 [no. of the PLC input]		6 – 167, 8 – 52
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		6 – 167, 8 – 52
MP4210.0-47	Setting a number in the PLC (D768 to D956) Input: -99 999.9999 to +99 999.9999		6 – 207, 8 – 43
MP4220.0-4	Setting a number in the PLC (W960 to W968) Input: 10 to 30 000		8 – 43
MP4230.0-31	Setting a number in the PLC (Module 9032) Input: -99 999.9999 to +99 999.9999		8 – 43

MP	Function and input	Software version and behavior	Page
MP4231.0-31	Setting a number in the PLC (Module 9032) Input: -99 999.9999 to +99 999.9999		8 – 43
MP4310.0-6	Setting a number in the PLC (W976 to W988, M4300 to M4411) Input: 10 to 30 000		8 – 43

4.3.8 Configuration of the Data Interface

MP	Function and input	Software version and 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	PLC RUN	9 – 14
MP5020	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 stop bit 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	PLC RUN CN123	9 – 17
MP5020.0	Operating mode EXT1		
MP5020.1	Operating mode EXT2		
MP5020.2	Operating mode EXT3 (PLC)		
MP5030	Data transfer protocol Input: 0 = Standard data transfer protocol 1 = Blockwise transfer 2 = Without protocol (only for MP5030.2)	PLC RUN CN123	9 – 17
MP5030.0	Operating mode EXT1		
MP5030.1	Operating mode EXT2		
MP5030.2	Operating mode EXT3 (PLC)		

MP	Function and input	Software version and behavior	Page
MP5040	Data transfer rate in operating mode EXT3 (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: 115 200 bps	PLC RUN	9 – 29

4.3.9 3-D Touch Probe

MP	Function and input	Software version and behavior	Page
MP6010	Selection of the touch probe Input: 0: Touch probe with cable transmission 1: Touch probe with infrared transmission	PLC CN123	7 – 166
MP6120	Probing feed rate Input: 1 to 3000 [mm/min]	PLC RUN CN123	7 – 170
MP6130	Maximum measuring range Input: 0.001 to 99 999.9999 [mm]	PLC RUN CN123	7 – 170
MP6140	Setup clearance over measuring point Input: 0.001 to 99 999.9999 [mm]	PLC RUN CN123	7 – 170
MP6150	Rapid traverse in probing cycle Input: 10 to 20 000 [mm/min]	PLC RUN CN123	7 – 170
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	7 – 173
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	7 – 171
MP6162	Orientation angle Input: 0 to 359.9999 [°]	PLC RUN CN123	7 – 171
MP6163	Minimum difference between the current spindle angle and MP6162 before executing an oriented spindle stop Input: 0 to 3.0000 [°]	PLC RUN CN123	7 – 171
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	7 – 171

MP	Function and input	Software version and behavior	Page
MP6170	Number of measurements in a programmed measurement (touch probe block) Input: 1 to 3	PLC RUN CN123	7 – 174
MP6171	Confidence range for programmed measurement (MP6170 > 1) Input: 0.002 to 0.999 [mm]	PLC RUN CN123	7 – 174
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]	PLC CN123	7 – 173
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]	PLC CN123	7 – 173
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]	PLC CN123	7 – 174
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	7 – 174

4.3.10 Tool Measurement with TT

MP	Function and input	Software version and 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	7 – 181, 7 – 182, 7 – 182, 7 – 184, 7 – 186, 7 – 188, 7 – 188

MP	Function and input	Software version and 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 routine 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	7 – 183
MP6505.0	Traverse range 1		
MP6505.1	Traverse range 2		
MP6505.2	Traverse range 3		

MP	Function and input	Software version and 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	7 – 186
MP6510 MP6510.0 MP6510.1	Permissible measuring error for tool measurement with rotating tool Input: 0.002 to 0.999 [mm] First measurement error Second measurement error	PLC RUN CN123	7 – 187
MP6520	Probing feed rate for tool measurement with non-rotating tool Input: 1 to 3000 [mm/min]	PLC RUN CN123	7 – 186
MP6530 MP6530.0 MP6530.1 MP6530.2	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] Traverse range 1 Traverse range 2 Traverse range 3	PLC RUN CN123	7 – 183
MP6531 MP6531.0 MP6531.1 MP6531.2	Diameter or edge length of the TT 130 probe contact for 3 traverse ranges Input: 0.001 to 99.9999 [mm] Traverse range 1 Traverse range 2 Traverse range 3	PLC RUN	7 – 184
MP6540 MP6540.0 MP6540.1	Safety zone around the probe contact of the TT 130 for pre-positioning Input: 0.001 to 99 999.9999 [mm] Safety clearance in tool axis direction Safety clearance in the plane perpendicular to the tool axis	PLC RUN CN123	7 – 183
MP6550	Rapid traverse in probing cycle for TT 130 Input: 10 to 20 000 [mm/min]	PLC RUN CN123	7 – 183
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	PLC RUN CN123	7 – 182

MP	Function and input	Software version and behavior	Page
MP6570	Max. permissible surface cutting speed at the tooth edge Input: 1.0000 to 129.0000 [m/min]	PLC RUN CN123	7 – 186
MP6572	Maximum permissible speed during tool measurement Input: 1 to 1000 [rpm] 0: 1000 [rpm]	PLC RUN CN123	7 – 186
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	7 – 184
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	7 – 184
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	7 – 184
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	7 – 187
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	7 – 187
MP6586.0	A axis		
MP6586.1	B axis		
MP6586.2	C axis		
MP6586.3	U axis		
MP6586.4	V axis		
MP6586.5	W axis		

4.3.11 Tapping

MP	Function and input	Software version and behavior	Page
MP7110.0	Minimum for feed-rate override during tapping Input: 0 to 150 [%]	PLC RUN	6 – 209
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	6 – 209, 6 – 210
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	6 – 213
MP7150	Positioning window of the tool axis during rigid tapping Input: 0.0001 to 2 [mm]	PLC RUN	6 – 213
MP7160	Spindle response during Cycle 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 control 0: Active 1: Not active	PLC RUN CN123	6 – 213

4.3.12 Display and Operation

MP	Function and input	Software version and behavior	Page
MP7210	Programming station Input: 0: Controlling and programming 1: Programming station with PLC active 2: Programming station with PLC inactive	CN123	7 – 59
MP7212	Power interrupted message Input: 0: Acknowledge the **Power interrupted** message with CE key 1: **Power Interrupted** message does not appear	PLC RUN CN123	7 – 57
MP7220	Block number increment for ISO programs Input: 0 to 250	PLC RUN CN123	7 – 33
MP7224.0	Disabling soft keys for file types Format: %xxxxxxxx Input: 0: Do not disable 1: Disable 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 – HELP files *.HLP Bit 7 – Point tables *.PNT	PLC RUN CN123	7 – 121
MP7224.1	Protecting file types Format: %xxxxxxxx Input: 0: Do not protect 1: Protect 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 – HELP files *.HLP Bit 7 – Point tables *.PNT		
MP7226.0	Reserved	PLC	
MP7226.1	Size of the datum table Input: 0 to 255 [lines]	RUN CN123	7 – 123

MP	Function and input	Software version and behavior	Page
MP7229 MP7229.0 MP7229.1	Depiction of the NC program Line number for program testing Input: 100 to 9999 Program length to which FK blocks are allowed Input: 100 to 9999	PLC RUN CN123	7 – 34
MP7230 MP7230.0 MP7230.1 MP7230.2 MP7230.3	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 NC conversational language, soft keys for OEM cycles PLC conversational language (user parameters) Write PLC error messages Help files	PLC RUN CN123	7 – 73
MP7235	Time difference to Universal Time (Greenwich Mean Time) Input: -23 to +23 [hours]	PLC RUN CN123	7 – 21

MP	Function and input	Software version and behavior	Page
MP7237 MP7237.0 MP7237.1 MP7237.2	Display and reset the operating times Display PLC operating times Input: Bits 0 to 7 represent PLC operating times 1 to 8 0: Do not display 1: Display Reset PLC operating times with the code number 857282 Input: Bits 0 to 7 represent PLC operating times 1 to 8 0: Do not reset 1: Reset Reset NC operating times with the 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	7 – 18
MP7238.0-7	Dialog messages for PLC operating times 1 to 8 Input: 0 to 4095 Dialog no. from the file (OEM.SYS)	PLC RUN	7 – 18
MP7245	Disabling auxiliary cycles Input: 0: Auxiliary cycles disabled 1: Auxiliary cycles permitted	PLC RUN	7 – 53
MP7246	Disabling paraxial positioning blocks Input: 0: Paraxial positioning block enabled 1: Paraxial positioning block disabled	PLC RUN	7 – 57
MP7251	Number of global Q parameters that are transferred from the OEM cycle to the calling program Input: 0 to 100	PLC RUN	–
MP7260	Number of tools in the tool table Input: 0 to 30 000	CN123	7 – 197
MP7261.0-3	Number of pockets in the tool magazine 1 to 4 Input: 0 to 254	CN123	7 – 197
MP7262	Maximum tool index number for indexed tools Input: 0 to 9	CN123	7 – 218
MP7263	Pocket table Format: %xx 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	CN123	7 – 197

MP	Function and input	Software version and behavior	Page
MP7266	Elements of the tool table Input: 0: No display 1 to 99: Position in the tool table	CN123	7 – 197
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 for 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 tables (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	

MP	Function and input	Software version and behavior	Page
MP7267	Elements of the pocket table Input: 0: No display 1 to 99: Position in the pocket table	CN123	7 – 198
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)		
MP7270	Feed rate display in the operating modes MANUAL OPERATION and ELECTRONIC HANDWHEEL Input: 0: Display of feed rate by 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	7 – 9
MP7280	Decimal character Input: 0: Decimal comma 1: Decimal period	PLC RUN CN123	7 – 74
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	7 – 33
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	7 – 5

MP	Function and input	Software version and behavior	Page
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	7 – 4
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	7 – 4
MP7291 MP7291.0 MP7291.1 MP7291.2	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 9. Display in traverse range 1 Display in traverse range 2 Display in traverse range 3	PLC RUN	6 – 3
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 PLC RUN CN123	7 – 32
MP7295	Disabling "datum setting" Format: %xxxxxxxxxxxxxx Input: Bits 0 to 13 represent axes 1 to 14 0: Not disabled 1: Disabled	PLC RUN CN123	7 – 32
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	7 – 32

MP	Function and input	Software version and behavior	Page
MP7300	Erasing the status display and Q parameters Input: 0: Erase the status display, Q parameters and tool data when a program is selected. 1: Erase the status display, Q parameters and tool data if a program is selected and M02, M30, and END PGM occur. 2: Erase the status display and tool data when a program is selected. 3: Erase the status display and tool data when a program is selected and in the event of M02, M30, END PGM. 4: Erase the status display and Q parameters when a program is selected. 5: Erase the status display and Q parameters when a program is selected and in the event of M02, M30, END PGM. 6: Erase the status display when a program is selected and in the event of M02, M30, END PGM. 7: Erase the status display when a program is selected and in the event of M02, M30, END PGM.	PLC RUN CN123	7 – 13
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: No display 1: Display Bit 4 – Reserved Bit 5 – Graphics during program test 0: 2.5-D 1: 3D Bit 6 – Graphics during program run 0: 2.5-D 1: 3D Bit 7 – Reserved	PLC RUN CN123	7 – 66
MP7315	Tool radius for graphic simulation without TOOL CALL Input: 0.0000 to 99 999.9999 [mm]	PLC RUN CN123	7 – 193

MP	Function and input	Software version and behavior	Page
MP7316	Penetration depth of the tool Input: 0.0000 to 99 999.9999 [mm]	PLC RUN CN123	7 – 193
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	7 – 193
MP7330.0-15	Specifying the user parameters 1 to 16 Input: 0 to 9999.00 (no. of the user parameter)	PLC RUN	7 – 58
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	7 – 58

4.3.13 Colors

MP	Function and input	Software version and behavior	Page
MP7350	Window frames	PLC RUN	7 – 60
MP7351	Error messages	PLC RUN	7 – 60
MP7352 MP7352.0 MP7352.1 MP7352.2	“Machine” operating mode display Background Text for operating mode Dialog	PLC RUN	7 – 60
MP7353 MP7353.0 MP7353.1 MP7353.2	“Programming” operating mode display Background Text for operating mode Dialog	PLC RUN	7 – 60
MP7354 MP7354.0 MP7354.1 MP7354.2 MP7354.3	“Machine” program text display Background General program text Active block Background of inactive window	PLC RUN	7 – 61
MP7355 MP7355.0 MP7355.1 MP7355.2 MP7355.3	“Programming” program text display Background General program text Active block Background of inactive window	PLC RUN	7 – 61
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	7 – 61
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	7 – 61
MP7358 MP7358.0 MP7358.1 MP7358.2 MP7358.3	“Programming” soft-key display Background Text color Inactive soft-key row Active soft-key row	PLC RUN	7 – 61

MP	Function and input	Software version and behavior	Page
MP7360	Graphics: 3-D view and plan view	PLC	7 – 61
MP7360.0	Background	RUN	
MP7360.1	Top 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	7 – 61
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	7 – 61
MP7362.0	Background of graphic window	RUN	
MP7362.1	Background of status display		
MP7362.2	Status symbols		
MP7362.3	Status values		
MP7363	Programming graphics	PLC	7 – 61
MP7363.0	Background	RUN	
MP7363.1	Resolved contour		
MP7363.2	Subprograms and frame for zooming		
MP7363.3	Alternative solutions		
MP7363.4	Unresolved contour		
MP7364	Color of the help illustrations for cycles	PLC	7 – 62
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		
MP7365	Oscilloscope	340 420-02	7 – 62
MP7365.0	Background	PLC	
MP7365.1	Grid	RUN	
MP7365.2	Cursor and text		
MP7365.3	Selected channel		
MP7365.4-9	Channel 1 to 6		

MP	Function and input	Software version and behavior	Page
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	7 – 62
MP7367 MP7367.0 MP7367.1-14	Large PLC window Background Colors 1 to 14	PLC RUN	7 – 62
MP7368 MP7368.0 MP7368.1 MP7368.2 MP7368.3	Pocket calculator Background Background of displays and keys Key texts ("os" in "cos") Key symbols	PLC RUN	7 – 62
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	7 – 62
MP7370 MP7370.0 MP7370.1-15	Small PLC window Background Colors 1 to 15	340 420-05 PLC RUN	7 – 62
MP7392	Screen saver Input: 1 to 99 [min] 0: No screen saver	PLC RUN CN123	7 – 62

4.3.14 Machining and Program Run

MP	Function and input	Software version and behavior	Page
MP7410	Scaling cycle in two or three axes Input: 0: Scaling cycle is effective in all three principle axes 1: Scaling cycle is effective only in the working plane	PLC RUN CN123	7 – 38
MP7411	Tool data in the touch probe block Format: %xx 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: Manage several blocks of touch probe calibration data in the tool tablea	PLC RUN CN123	7 – 171
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: 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	7 – 38
MP7430	Overlap factor for pocket milling Input: 0.001 to 1.414	PLC RUN CN123	7 – 36

MP	Function and input	Software version and behavior	Page
MP7431	Arc end-point tolerance Input: 0.0001 to 0.016 [mm]	PLC RUN CN123	7 – 57
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	6 – 45, 6 – 84, 6 – 126, 6 – 128, 7 – 52, 7 – 194
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	7 – 52
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	6 – 203

MP	Function and input	Software version and behavior	Page
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	7 – 40
MP7451.0-8	Feed rate for returning to the contour for axes 1 to 9 Input: 10 to 300 000 [mm/min]	PLC RUN	7 – 40
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 principle axes during compensating movements through M128 Input: 0 to 300 000 [mm/min]	PLC RUN CN123	6 – 84
MP7475	Reference for datum table Input: 0: Reference is workpiece datum 1: Reference is machine datum (MP960.x)	PLC RUN CN123	7 – 123
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: Tool number output for every TOOL CALL block 3: Output of the pocket number and tool number only when tool number changes 4: Output of the 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 the pocket number and tool number for every TOOL CALL block. Pocket table is not changed.	PLC RUN	7 – 226
MP7480.1	With 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: Output of the pocket number and tool number only when tool number changes 4: Output of the pocket number and tool number for every TOOL DEF block		

MP	Function and input	Software version and behavior	Page
MP7481.x	Sequence for new and returned tool when changing tools Format: %xxxx 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 2: New tool from magazine 4	340 420-06 PLC RUN	7 – 229
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		
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	7 – 226
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	6 – 23, 7 – 171, 7 – 181

MP	Function and input	Software version and behavior	Page
MP7500	Tilting working plane 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 – Nonfunctional Bit 4 – Nonfunctional 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 – Nonfunctional Bit 8 – Nonfunctional	PLC RUN	6 – 81
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	6 – 85

MP	Function and input	Software version and behavior	Page
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	PLC RUN	6 – 81
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 0: Incremental dimension for swivel head 1: Absolute with respect to the machine datum for tilting table	PLC RUN	6 – 81
MP7520.0-14	Transformation 1 to transformation 15		
MP7530	Type of dimension for transformation Input: -99 999.9999 to +99 999.9999 0: Free tilting axis	PLC RUN	6 – 82
MP7530.0-14	Transformation 1 to transformation 15		
MP7550	Home position of the tilting element Input: -99 999.9999 to +99 999.9999	PLC RUN	6 – 82
MP7550.0	A Axis		
MP7550.1	B Axis		
MP7550.2	C Axis		

4.3.15 Hardware

MP	Function and input	Software version and behavior	Page
MP7600.0	Position controller cycle time = MP7600.0 · 0.6 ms Input: 1 to 20 Proposed input value: 3 (= 1.8 ms)	RESET	6 – 124
MP7600.1	PLC cycle time = position controller cycle time · MP7600.1 Input: 1 to 20 Proposed input value: 7 (= 12.5 ms)		6 – 124, 8 – 4
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 – Nonfunctional 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 – Nonfunctional Bit 5 – Reserved Bit 6 – Feed-rate smoothing 0: Not active 1: Active	PLC RUN	6 – 122, 6 – 198, 7 – 9
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	PLC RUN	7 – 147
MP7641	Entry of the interpolation factor Input: 0: Through iTNC keyboard 1: Through PLC Module 9036	PLC RUN	7 – 147

MP	Function and input	Software version and behavior	Page
MP7645 MP7645.0 MP7645.0 MP7645.1 MP7645.2 MP7645.3-7	Initializing parameter for handwheel Layout of the handwheel keypad for HR 410 Input: 0: Evaluation of the keys by NC, including LEDs 1: Evaluation of the keys by PLC 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 3rd handwheel axis IV Switch position 5 3rd handwheel axis V 2: Switch position 3 3rd handwheel axis Z Switch position 4 3rd handwheel axis IV Switch position 5 3rd handwheel axis V Fixed assignment of third handwheel if MP7645.2 = 1 Input: 1: Axis X 2: Axis Y 4: Axis Z 8: Axis IV (MP410.3) 16: Axis V (MP410.4) 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 No function	PLC RUN	7 – 150, 7 – 151
MP7650	Handwheel counting direction for each axis Format: %xxxxxxxxx Input: 0: Negative counting direction 1: Positive counting direction	PLC RUN	7 – 147

MP	Function and input	Software version and behavior	Page
MP7660	Threshold sensitivity for electronic handwheel Input: 0 to 65 535 [increments]	PLC RUN	7 – 147
MP7670 MP7670.0 MP7670.1 MP7670.2	Interpolation factor for handwheel Input: 0 to 10 Interpolation factor for low speed Interpolation factor for medium speed (only HR 410) Interpolation factor for high speed (only HR 410)	PLC RUN	7 – 147, 7 – 150
MP7671 MP7671.0 MP7671.1 MP7671.2	Handwheel feed rate in the Handwheel operating mode with HR 410 Input: 0 to 1000 [% of MP1020] Low speed Medium speed (only HR 410) High speed (only HR 410)	PLC RUN	7 – 150
MP7680	Machine parameter with multiple function Format: %xxxxxxxxxxxxxx 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	6 – 164, 6 – 165, 7 – 38, 7 – 40, 7 – 119, 7 – 217

MP	Function and input	Software version and behavior	Page
MP7680	Machine parameter with multiple function 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 activate **APPROACH POSITION**	PLC RUN	
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.	PLC RUN	7 – 42

MP	Function and input	Software version and behavior	Page
MP7682	Machine parameter with multiple function Format: %xxxxx 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 of rotary axes with M128 0: With consideration of head dimensions 1: Without consideration of head dimensions	PLC RUN	6 – 82, 6 – 122, 7 – 5, 7 – 5
MP7683	Executing pallet tables and NC programs Format: %xxxxx Input: Bit 0 – Nonfunctional 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	PLC RUN	7 – 141, 7 – 35

MP	Function and input	Software version and behavior	Page
MP7684	Nominal position value filter and path control with M128 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 curvature changes 1: Do not include curvature changes Bit 5 – Feed-rate reduction at corners with M128 0: Include only maximum compensatory path 1: Do not include compensatory paths Bit 6 – Feed-rate reduction at corners with M128 0: Include compensatory paths depending on the entry in bit 5 1: Include all compensatory paths Bit 7 – Reserved	PLC RUN	6 – 84, 6 – 121
MP7690	Reserved, enter 0		–
MP7691	Size of a log file with operating system messages. Can only be evaluated by HEIDENHAIN. Set MP7691 = 0. Input: 0: Function inactive 1 to 10 [MB]	340 420-05	–

4.3.16 Second Spindle

MP	Function and input	Software version and behavior	Page
MP13010 to MP13520	Machine parameter block for the second spindle Input: Function and input range are identical with MP3010 to MP3520.		6 – 215

5 Modules, Markers and Words

5 Modules, Markers and Words

5.1 Overview of Modules

Module	Function	SW Vers.	Page
9000/ 9001	Copy in the marker or word range		8 – 150
9002	Reading all inputs of a PLC input/output unit		7 – 156
9003	Reading in analog inputs		7 – 159
9004	Edges of PLC inputs		7 – 158
9005	Update all outputs of a PLC input/output unit		7 – 157
9006	Set and start PLC timer		8 – 49
9007	Diagnostic information of the PL		7 – 155
9008	Reading specific inputs of a PLC input/ output unit		7 – 157
9009	Update certain outputs of a PLC input/ output unit		7 – 158
9010/ 9011/ 9012	Read in the word range		8 – 151
9019	Size of the processing stack		8 – 54
9020/ 9021/ 9022	Write in the word range		8 – 152
9031	Overwrite machine parameters		4 – 7
9032	Read machine parameters		4 – 8, 8 – 43
9033	Select machine parameter file		4 – 10
9034	Load a machine parameter subfile		4 – 11
9035	Reading status information		6 – 19, 6 – 41, 7 – 14, 7 – 163
9036	Writing status information		7 – 148, 7 – 162
9038	Reading general axis information		6 – 17
9040	Reading the axis coordinates (format 0.001 mm)		7 – 6
9041	Reading the axis coordinates (format 0.0001 mm)		7 – 6
9042	Reading the spindle coordinates (format 0.001°)		6 – 189

Module	Function	SW Vers.	Page
9044	Reading the spindle coordinates (format 0.0001°)		6 – 189
9050	Conversion from binary numbers → ASCII		8 – 153
9051	Conversion from binary numbers → ASCII		8 – 154
9052	Conversion from ASCII → Binary		8 – 155
9053	Conversion from binary → ASCII/ hexadecimal		8 – 155
9054	Conversion from ASCII/hexadecimal → binary		8 – 156
9055	Local time		7 – 22
9060	Status of M functions		7 – 50
9061	Status of non-modal M functions		7 – 50
9066	Status of HEIDENHAIN inverter		6 – 181
9070	Copy a number from a string		8 – 130
9071	Find the string length		8 – 131
9080	Clearing the small PLC window		7 – 83
9081	Interrogating the status of the small PLC window		7 – 83
9082	Showing a string in the small PLC window		7 – 84
9083	Showing a moving-bar diagram in the small PLC window		7 – 85
9085	Display PLC error messages		7 – 24
9086	Erase PLC error messages		7 – 25
9087	Status of PLC error message		7 – 25
9088	Displaying the M functions		7 – 11
9089	Control in operation		7 – 12
9090	Selection of a line in the pallet table		7 – 142
9091	Finding the line number of a tool in the tool table		7 – 218
9092	Searching for an entry in the tables selected for execution (.T/.D/.TCH)		7 – 199
9093	Read data from tables selected for program (.T/.D/.TCH)		7 – 201
9094	Writing data into a tool and datum table		7 – 202
9095	Select active line in configuration file		6 – 41
9096	Deletion of a line in the tool table		7 – 203
9097	Selecting the geometry description		6 – 66
9098	Finding the active geometry description		6 – 65

Module	Function	SW Vers.	Page
9100	Assign data interface		9 – 30
9101	Release data interface		9 – 31
9102	Status of data interface		9 – 31
9103	Transmit string through data interface		9 – 32
9104	Receive string through data interface		9 – 33
9105	Transmit binary data through data interface		9 – 34
9106	Receive binary data through data interface		9 – 35
9107	Read from receiving buffer		9 – 36
9110	Transmit a message via LSV2		9 – 37
9111	Receive a message via LSV2		9 – 38
9112	Transmit ASCII characters via data interface		9 – 39
9113	Receive ASCII characters via data interface		9 – 40
9120	Starting a PLC axis		6 – 27
9121	Stopping a PLC axis		6 – 27
9122	Status of PLC axis		6 – 28
9123	Traversing the reference marks of PLC axes		6 – 29
9124	Feed rate override for PLC axis		6 – 29
9125	Stop PLC axis at next Hirth grid position		6 – 30
9130	Output of an analog voltage		7 – 160
9133	Temperature of the MC 422		6 – 176
9135	Switch on 3-D touch probe		7 – 166
9136	Switching the touch probe on/off	340 420-06	7 – 167
9145	Actual-to-nominal value transfer		6 – 136
9146	Saving and reestablishing actual position values		6 – 218
9147	Assigning the reference value to an axis		6 – 101
9148	Use nominal value as actual value	340 420-06	–
9151	Select traverse range and axis designation		6 – 21
9152	Selecting traverse range, axis display and axis designation		6 – 22
9153	Switching the touch probe axis		7 – 171
9155	Axis switchover from closed loop to open loop		6 – 219
9156	Axis switchover from open loop to closed loop		6 – 219
9157	Drive controller status		6 – 148
9158	Maximum torque		6 – 159
9159	Drives that are switched off in 200 ms		6 – 148

Module	Function	SW Vers.	Page
9160	Status request for temperature monitoring and I^2t monitoring		6 – 179
9161	Enabling the drive controller		6 – 148
9162	Status request of the drive controller		6 – 149
9163	Switching the operating modes		6 – 214
9164	Reading the actual speed value of the motor		6 – 139
9165	Sampling the current motor temperature		6 – 176
9166	Momentary utilization of the drive motor		6 – 180
9167	Supply voltage monitoring		6 – 175
9168	Interrogating the commissioning status		6 – 244
9169	Axes for which I32 does not switch off the drives		6 – 149
9171	Oriented spindle stop		6 – 205
9175	Spindle switchover		6 – 216
9180	Simulation of NC keys		7 – 114
9181	Disabling individual NC keys		7 – 114
9182	Re-enabling individual NC keys		7 – 115
9183	Disabling groups of NC keys		7 – 115
9184	Re-enabling groups of NC keys		7 – 116
9186	Call a soft-key function		7 – 116
9187	Status of a soft-key function call		7 – 117
9189	Shutting down the control		7 – 56
9190	Starting the operating times		7 – 18
9191	Stopping the operating times		7 – 18
9192	Reading the operating times		7 – 19
9193	Setting the operating times		7 – 19
9194	Alarm when operating time exceeded		7 – 20
9195	System time		7 – 21
9196	Finding the PLC cycle time		8 – 4
9197	Start cycle timer		8 – 49
9200	Display/delete PLC soft-key row		7 – 111
9201	Display/delete PLC soft key		7 – 112
9202	Select/deselect PLC soft keys and PLC windows		7 – 112
9203	Activate PLC soft-key menu		7 – 104
9204	Update the PLC soft keys		7 – 105
9205	Setting the word for acknowledgment of PLC soft keys		7 – 106
9206	Change setting of the PLC soft keys		7 – 107
9207	Replace PLC soft keys		7 – 108
9208	Status information of the PLC soft keys		7 – 109
9210	Opening or erasing screen mask for the PLC window		7 – 97

Module	Function	SW Vers.	Page
9211	Status of the large PLC window		7 – 98
9215	Activating a PLC pop-up window		7 – 30
9216	Pop-up window with tool selection list	340 420-03	7 – 206
9220	Renewed traversing of the reference marks		6 – 105
9221	Starting a PLC positioning movement		6 – 32
9222	Status request of PLC positioning movement		6 – 32
9223	Free rotation		7 – 7
9225	Compensation value for the reference mark		6 – 100
9230	Datum shift		7 – 165
9231	Compensation of thermal expansion		6 – 43
9240	Opening a file		7 – 133
9241	Closing a file		7 – 134
9242	Positioning in a file		7 – 135
9243	Reading from a file line by line		7 – 136
9244	Writing to a file line by line		7 – 137
9245	Reading a field out of a table		7 – 125
9246	Writing to a field in a table		7 – 127
9247	Searching for a condition in a table		7 – 129
9250	Starting the PLC editor for tables		7 – 131
9251	Ending the PLC editor for tables		7 – 132
9252	Positioning the cursor in the PLC editor		7 – 132
9255	Reading a field from a table as an integer value		7 – 126
9256	Writing to a field in a table		7 – 128
9260	Receiving events and waiting for events		8 – 139
9261	Sending events		8 – 140
9262	Context change between spawn processes		8 – 141
9263	Interrupting a spawn process for a defined time		8 – 141
9270	Reading a code word		8 – 23
9271	Writing a code word		8 – 23
9275	Writing ASCII data into the log		7 – 79
9276	Writing operand contents into the log		7 – 80
9279	Control reset		7 – 56
9280	Start the NC macro (Run pallet entry)		7 – 144
9281	Selection of a line in the pallet table		7 – 142
9290	Selecting a file		7 – 122
9291	Calling an NC macro		8 – 25
9300	Locking/releasing the pocket table		7 – 211

Module	Function	SW Vers.	Page
9301	Find the number of an entry in the pocket table		7 – 213
9302	Search for a free pocket in the tool magazine		7 – 213
9304	Copying columns P1 to P5 to the pocket table	340 420-03	7 – 207
9305	Tool exchange in the pocket table		7 – 212
9306	Exchange tools between tool magazines		7 – 214
9310	Read the machine parameter from the run-time memory		4 – 9
9320	Status of the NC program end		7 – 34
9321	Find the current block number	340 420-06	7 – 33
9340	Searching for a pocket depending on magazine rules	340 420-03	7 – 208
9341	Editing a pocket table depending on magazine rules	340 420-03	7 – 209
9342	Find magazine and pocket number	340 420-06	7 – 210

5.2 Overview of Markers and Words

	Marker	Description	Set	Reset	SW Vers.	Page
M	1900 - 1999	Decoded M function if M4571 is set	NC	NC		7 – 49
M	4000	Spindle in position	NC	NC		6 – 205
M	4001	Nominal speed command signal of the spindle not in the ramp	NC	NC		6 – 194
M	4002	Nominal speed value = 0	NC	NC		6 – 194
M	4003	Nominal speed value output analog or digital (MP3010 = 3 to 8)	NC	NC		6 – 192
M	4004	Impermissible speed was programmed	NC	NC		6 – 195
M	4005	Status display and nominal speed value output for M03	PLC	PLC		6 – 194, 7 – 10
M	4006	Status display and nominal speed value output for M04	PLC	PLC		6 – 194, 7 – 10
M	4007	Status display M05 and spindle stop	PLC	PLC		6 – 194, 7 – 10
M	4008	Disable speed output for spindle	PLC	PLC		6 – 195, 7 – 10
M	4009	Counterclockwise spindle rotation (for gear change)	PLC	PLC		6 – 197
M	4010	Clockwise spindle rotation (for gear change)	PLC	PLC		6 – 197
M	4011	Activate rotational speed MP3520.0 and direction of rotation from M4013	PLC	PLC		6 – 207
M	4012	Opening the spindle control loop	PLC	PLC		6 – 205, 7 – 173
M	4013	Direction for spindle orientation from a standstill (M03 = 0; M04 = 1)	PLC	PLC		6 – 207
M	4014	Reverse the direction of spindle rotation	PLC	PLC		6 – 194
M	4015	Renewed evaluation of the spindle reference mark	PLC	NC		6 – 205
M	4016	Cycle 13 is executed	NC	PLC		6 – 207
M	4017	Spindle moving in feedback control	NC	NC		6 – 205
M	4018	Reference mark for spindle not yet traversed	NC	NC		6 – 205
M	4019	Reversing the counting direction of the position encoder on the spindle	PLC	PLC		6 – 194
M	4030	Cycle 2 or Cycle 17 active	NC	NC		6 – 209, 6 – 213
M	4031	Cycle 17 or Cycle 18 active	NC	NC		6 – 213

	Marker	Description	Set	Reset	SW Vers.	Page
M	4040	Status display M07, M08, and M09 highlighted	PLC	PLC		7 – 10
M	4041	Status display M07, M08, M09, MK	PLC	PLC		7 – 10
M	4042	Status display M07, M08, M09, MK	PLC	PLC		7 – 10
M	4050	Touch probe not ready, ready signal is missing	NC	NC		7 – 170
M	4051	Stylus deflected before start of probing cycle	NC	NC		7 – 170
M	4052	Stylus is deflected, probing process is completed	NC	PLC		7 – 170
M	4053	Probing process has been ended or canceled	NC	NC		7 – 170
M	4054	Battery voltage too low (battery warning at touch probe connection); evaluated only during the probing process	NC	NC		7 – 170
M	4055	Enable the probing process	NC	PLC		7 – 170
M	4056	NC stop in all operating modes if stylus is deflected	PLC	PLC		7 – 170
M	4060	Cycle for tool measurement started	NC	NC		7 – 189
M	4061	0: Measure the tool 1: Check the tool	NC	NC		7 – 189
M	4062	0: Wear tolerance not exceeded 1: Wear tolerance exceeded	NC	NC/ PLC		7 – 189
M	4063	0: Breakage tolerance not exceeded 1: Breakage tolerance exceeded	NC	NC/ PLC		7 – 189
M	4065	Workpiece dimensions are OK	NC	PLC		7 – 174
M	4066	Workpiece must be reworked	NC	PLC		7 – 174
M	4067	Workpiece is scrap	NC	PLC		7 – 174
M	4070	Strobe signal for gear code	NC	NC		6 – 197
M	4071	Strobe signal for S code	NC	NC		6 – 200
M	4072	Strobe signal for M functions	NC	NC		7 – 49
M	4073	Strobe signal T code (P code) with TOOL CALL	NC	NC		7 – 226, 7 – 244
M	4074	Strobe signal T code (P code) with TOOL DEF	NC	NC		7 – 226, 7 – 244
M	4075	Transfer active with FN19	NC	NC		8 – 27
M	4090	Acknowledgment of “gear change completed”	PLC	PLC		6 – 197
M	4091	Acknowledgment of S code	PLC	PLC		6 – 200
M	4092	Acknowledgment of M functions	PLC	PLC		7 – 49
M	4093	Acknowledgment of T code (P code) with TOOL CALL	PLC	PLC		7 – 226, 7 – 244
M	4094	Acknowledgment of T code (P code) with TOOL DEF	PLC	PLC		7 – 226, 7 – 244
M	4095	Acknowledgment of transfer with FN19	PLC	PLC		8 – 27

	Marker	Description	Set	Reset	SW Vers.	Page
M	4120 - 4128	PLC positioning axis 1 to 9 active	NC/ PLC	NC/ PLC		6 – 33
M	4130	Activation of spindle orientation, or spindle orientation has been started with Module 9171	NC/ PLC	NC		6 – 207
M	4131	Activation of Q-parameter transfer to the NC; data from D258, Q number from W516	PLC	NC		8 – 28
M	4132	Activate datum shift from D528 to D544, or call Module 9230	PLC	NC		7 – 165
M	4133	Starting and stopping the free rotation function	PLC	NC		7 – 8
M	4134	Activation of a gear range and speed through the PLC	PLC	NC		6 – 197
M	4135	Strobe marker for selecting the traverse range	PLC	NC		6 – 20
M	4150	Operating mode: Manual operation	NC	NC		–
M	4151	Operating mode: Electronic handwheel	NC	NC		–
M	4152	Operating mode: Positioning with manual data input	NC	NC		–
M	4153	Operating mode: Program run, single block	NC	NC		–
M	4154	Operating mode: Program run, full sequence	NC	NC		–
M	4155	Operating mode: Traversing the reference marks	NC	NC		–
M	4156	MANUAL TRAVERSE soft key pressed	NC	NC		7 – 41
M	4157	Returning to the contour (MOVE TO POSITION) is active	NC	NC		7 – 41
M	4158	Block scan active	NC	NC		7 – 41
M	4159	PLC editor: END key or soft key pressed	NC	NC/ PLC		7 – 131
M	4160	Pallet table selected	NC	NC		
M	4161	M/S/T/Q transfer after block scan	NC	NC		7 – 42
M	4170	END PGM, M02 or M30 was executed	NC	NC		7 – 39
M	4172	1st PLC scan after power on	NC	NC		–
M	4173	1st PLC scan after interruption of the PLC program	NC	NC		–
M	4174	1st 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		7 – 12
M	4176	Control is in operation, control-in-operation symbol is on or is blinking	NC	NC		7 – 12
M	4177	Erasable error message is displayed	NC	NC		6 – 183
M	4178	Error message EMERGENCY STOP is displayed	NC	NC		6 – 183
M	4179	Control is shut down	NC	NC		7 – 55
M	4180	Rapid traverse programmed (FMAX)	NC	NC		7 – 10

	Marker	Description	Set	Reset	SW Vers.	Page
M	4181	NC program selected	NC	PLC		7 – 33
M	4182	AUTOSTART active	NC	NC		7 – 35
M	4183	Time from AUTOSTART expired	NC	NC		7 – 35
M	4185	Internal stop performed	NC	PLC	340 420-06	7 – 34
M	4200	Overflow during multiplication	NC	PLC		8 – 89, 8 – 103, 8 – 132
M	4201	Division by 0	NC	PLC		8 – 90, 8 – 103, 8 – 132
M	4202	Incorrectly executed modulo	NC	PLC		8 – 91, 8 – 103, 8 – 132
M	4203	Error status for PLC module	NC	NC/ PLC		8 – 103, 8 – 132
M	4204	Reserved for errors that the PLC programmer would like to catch	NC	NC		8 – 132
M	4220	Error from PET table with F stop active	NC	NC		7 – 28
M	4221	Error from PET table with NC stop active	NC	NC		7 – 28
M	4222	Error from PET table with EM. STOP active	NC	NC		7 – 28
M	4230	NC start via LSV2	NC	NC		7 – 119
M	4231	NC stop via LSV2	NC	NC		7 – 119
M	4300 - 4315	Value from MP4310.0	NC	NC		8 – 42
M	4316 - 4331	Value from MP4310.1	NC	NC		8 – 42
M	4332 - 4347	Value from MP4310.2	NC	NC		8 – 42
M	4348 - 4363	Value from MP4310.3	NC	NC		8 – 42
M	4364 - 4379	Value from MP4310.4	NC	NC		8 – 42
M	4380 - 4395	Value from MP4310.5	NC	NC		8 – 42
M	4396 - M4411	Value from MP4310.6	NC	NC		8 – 42
M	4520	Additional T code (P code) follows with TOOL CALL	NC	NC		7 – 228, 7 – 244
M	4521	Tool number zero programmed	NC	NC		7 – 226
M	4522	Tool with pocket number programmed is in effect with MP7480.0 = 3 or 4 and TOOL CALL	NC	NC		7 – 228
M	4523	Tool without pocket number programmed is in effect with MP7480.0 = 3 or 4 and TOOL CALL	NC	NC		7 – 228

	Marker	Description	Set	Reset	SW Vers.	Page
M	4524	Special tool called, TOOL CALL	NC	NC		7 – 228, 7 – 244
M	4525	TOOL CALL after expiration of tool life	NC	NC		7 – 228
M	4526 - 4534	Axis 1 to Axis 9 is the tool axis	NC	NC		6 – 18
M	4538	Geometry of the tool from W264	PLC	NC		7 – 41, 7 – 226
M	4540	Sequence of tool number or pocket number transfer (M4520 = 1)	PLC	PLC		7 – 228, 7 – 244
M	4541	Special tool in original pocket in spite of variable pocket coding	PLC	PLC		7 – 215, 7 – 228, 7 – 244
M	4542	Do not update pocket number in the pocket table	PLC	PLC		7 – 41, 7 – 228
M	4543	Tool life 1 expired (TIME1 in the tool table)	NC	NC/ PLC		7 – 217
M	4546	Tool life 2 expired (TIME2 in the tool table)	NC	NC/ PLC		7 – 217
M	4547	T and G strobes with TOOL CALL	NC	NC		6 – 197, 7 – 226
M	4560	NC stop (0: Stop)	PLC	PLC		7 – 119
M	4561	Rapid traverse	PLC	PLC		7 – 119
M	4562	Memory function for axis direction keys (MP7680 Bit 0 = 1)	PLC	PLC		7 – 119
M	4563	Feed-rate enable for all axes	PLC	PLC		6 – 135
M	4564	NC start	PLC	PLC		7 – 119
M	4570	Unit of measure for transfer with FN19	NC	NC		8 – 27
M	4571	Activation of decoded M-code transfer in M1900 to M1999	PLC	PLC		7 – 49
M	4572	Enabling the incremental jog positioning	PLC	PLC		7 – 161
M	4574	Select the traverse range (with M4575)	PLC	PLC		6 – 20, 7 – 170
M	4575	Select the traverse range (with M4574)	PLC	PLC		6 – 20, 7 – 170
M	4576	Locking the handwheel	PLC	PLC		7 – 147
M	4577	Disabled key was pressed	NC	PLC		7 – 113
M	4579	INCREMENT OFF/ON soft key	NC	NC		7 – 161
M	4580	Suppress EMERGENCY STOP, open all position control loops, NC stop	PLC	PLC		6 – 134, 6 – 183
M	4581	Open all position control loops, NC stop, activate “Approach position”	PLC	PLC		6 – 134
M	4586	Enable AUTOSTART	PLC	NC/ PLC		7 – 35
M	4587	Feed rate limit exceeded F MAX	PLC	PLC		6 – 132

	Marker	Description	Set	Reset	SW Vers.	Page
M	4590	Status fast PLC input from MP4130.2	NC	PLC		8 – 52
M	4591	Status fast PLC input from MP4130.3	NC	PLC		8 – 52
M	4592	Status fast PLC input from MP4130.4	NC	PLC		8 – 52
M	4593	Status fast PLC input from MP4130.5	NC	PLC		8 – 52

	Marker	Description	Set	Reset	SW Vers.	Page
W	256	Gear code	NC/ PLC	NC/ PLC		6 – 197
W	258	S code	NC	NC		6 – 200
W	260	Code for M functions	NC	NC		7 – 49
W	262	Tool pocket number	NC	NC		7 – 226, 7 – 244
W	264	Tool number	NC	NC		7 – 226, 7 – 244
W	266	Index number of a programmed indexed tool	NC	NC		7 – 218
W	268	Tool magazine number	NC	NC		7 – 212
W	270	Line number in help file	NC	NC		7 – 27
W	272	Mode of operation	NC	NC		6 – 104
W	274	Code of the depressed key	NC	NC		7 – 113
D	276	Code of the code number last entered via MOD	NC	NC		7 – 59
D	280	First numerical value from FN19	NC	NC		8 – 27
D	284	Second numerical value from FN19	NC	NC		8 – 27
W	302	Number of the horizontal PLC soft key that was pressed	NC	NC		7 – 110
W	304	Number of the vertical PLC soft key that was pressed	NC	NC		7 – 104
W	320	Nominal speed value [rpm]	NC	NC		6 – 192
W	322	Actual speed value [rpm]	NC	NC		6 – 192
D	356	Programmed speed [0.001 rpm]	NC	NC		6 – 192, 6 – 197
D	360	Programmed feed rate	NC	NC		6 – 132
D	364	Nominal speed value [rpm]	NC	NC		6 – 192
D	368	Actual speed value [rpm]	NC	NC		6 – 192
D	388	Current tool feed rate [mm/min]	NC	NC		6 – 132
W	480-484	Analog input at X48 [0.1 V]	NC	NC		7 – 159
W	486 - 490	Temperature input at X48 [0.5 °C]	NC	NC		6 – 42, 7 – 159
W	492	Percentage for spindle override (NC to PLC)	NC	NC		6 – 198
W	494	Percentage for feed-rate override (NC to PLC)	NC	NC		7 – 10
W	516	Q No. 0-7 for numerical data transfer PLC to NC	PLC	PLC		8 – 28
B	518	Defining the free rotation function	PLC	PLC		7 – 8
B	519	Traverse direction for free rotation	PLC	PLC		7 – 8
W	522	Enabling the high-speed PLC inputs	PLC	PLC		6 – 167, 8 – 52
W	524	Drive Enabling for Axis Groups	PLC	PLC		6 – 147

	Marker	Description	Set	Reset	SW Vers.	Page
D	528	Double word with multiple function, here data for transfer from PLC to NC	PLC	PLC		8 – 28
D	528 - 544	Target position for PLC positioning	PLC	PLC		6 – 33
D	528 - 544	Datum shift for axis 1 to 5	PLC	PLC		7 – 165
W	560 - 568	Feed rate for PLC positioning	PLC	PLC		6 – 33, 7 – 8
W	576 - 584	Lag-tracking axis error compensation	PLC	PLC		6 – 42
D	592	Nominal position for spindle orientation	PLC	PLC		6 – 207
D	596	Max. feed rate from PLC [mm/min]	NC/ PLC	PLC		6 – 132
D	604	Maximum possible spindle speed	PLC	NC/ PLC		6 – 192
W	754	% function for feed-rate override for free rotation	PLC	PLC		7 – 8
D	756	Programmed rotational speed or rotational speed of the PLC [0.001 rpm]	NC/ PLC	NC/ PLC		6 – 197
D	760	Offset in tilting axes touch probe center offset [1/10 000°]	PLC	PLC		7 – 173
W	764	Percentage for spindle override (PLC to NC)	NC/ PLC	NC/ PLC		6 – 198
W	766	Percentage for feed-rate override (PLC to NC)	NC/ PLC	NC/ PLC		7 – 10
D	768 - 956	Values from MP4210.0 to MP4210.47	NC	NC		8 – 41
W	960 - 968	Value from MP4220.0 to MP4220.4	NC	NC		8 – 42
W	976 - 988	Value from MP4310.3 to MP4310.6	NC	NC		8 – 42
W	1008	S code for minimum speed	NC	NC		6 – 200
W	1018	Number of files opened by the PLC	NC	NC		7 – 122
W	1020	Number of open files	NC	NC		7 – 122
W	1022	Error status of the module last called	NC	NC		–
W	1024	Axis release	NC	NC		6 – 134
W	1026	Axes in position	NC	NC		6 – 172
W	1028	Axes in motion	NC	NC		6 – 173
W	1030	Current direction of traverse	NC	NC		6 – 9
W	1032	Reference marks not yet traversed	NC	NC		6 – 104
W	1034	Positive software limit switch was traversed	NC	NC		6 – 23
W	1036	Negative software limit switch was traversed	NC	NC		6 – 23
W	1038	Preparing opening of the position control loop	PLC	PLC		6 – 135
W	1040	Axis-specific opening of the position control loop	PLC	PLC		6 – 135
W	1042	Deactivation of monitoring functions	PLC	PLC		6 – 166

	Marker	Description	Set	Reset	SW Vers.	Page
W	1044	Actual-to-nominal value transfer	PLC	PLC		6 – 136
W	1046	Manual traverse in positive direction	PLC	PLC		7 – 119
W	1048	Manual traverse in negative direction	PLC	PLC		7 – 119
W	1050	Incremental jog positioning in positive direction	PLC	PLC		7 – 161
W	1052	Incremental jog positioning in negative direction	PLC	PLC		7 – 161
W	1054	Reference end position	PLC	PLC		6 – 104
W	1056	Lubrication pulse: Value in MP4060 exceeded	NC	NC		6 – 24
W	1058	Resetting the accumulated distance	PLC	PLC		6 – 24
W	1060	Axis-specific feed-rate enable	PLC	PLC		6 – 135
W	1062	Lock the handwheel for specific axes	PLC	PLC		7 – 147

6 Configuring the Axes and Spindle

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 correspond to 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 9 from the right represent axes 1 to 9
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 9 from the right represent lines 1 to 9 Character 10 is the spindle "S", which is always output in line 9.
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 and V

On the keyboard unit and the HR 410 handwheel, you can assign the axis keys IV and V as desired.

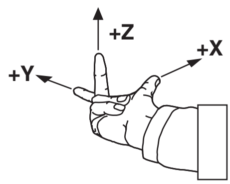
MP410	**Assignment of axis keys IV and V**
Input:	Axis designation XYZABCUVWxyzabcuvw-
MP410.3	Axis key IV
MP410.4	Axis key V

6.1.2 Axis Designation

Principal axes X, Y, Z

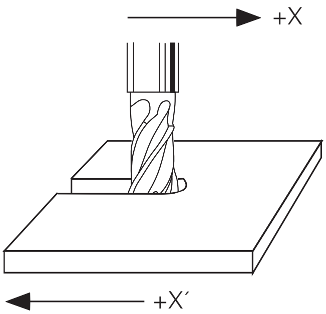
The coordinate axes and their directions of motion are defined in the international standard ISO 841.

An easy way to remember this system is to use the "right-hand rule":



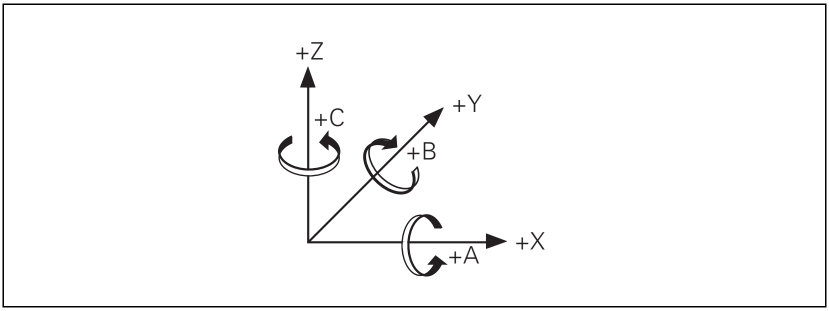
Algebraic signs of the axes

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´:



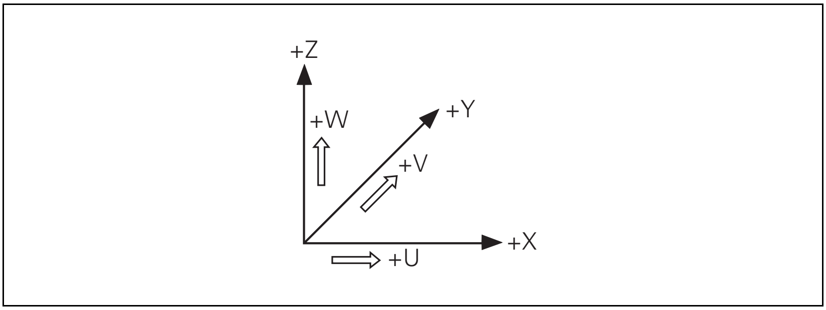
Rotary axes A, B, C

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.

From this data 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

$$\text{MP334.x} = \frac{20\text{ mm}}{0.02\text{ mm}} = 1000 \text{ (or 0)}$$

MP331.x **Distance for the number of signal periods in MP332**
Input: 0.0001 to 99 999.9999 [mm] or [°]

MP332.x **Number of signal periods for the distance in MP331**
Input: 1 to 16 777 215

MP334.x **Nominal increment between two fixed reference marks on encoders with distance-coded reference marks**
Input: 1 to 65 535
0: 1000

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 422.

- 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: Nonfunctional
0: 1 V_{PP}
1: 11 μA_{PP}

MP115.1 Reserved

Format: %xxxxxxxxxxx

Input: Enter %00000000000

MP115.2 Input frequency of the 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: Nonfunctional
For 1 V_{PP}: 0: 33 kHz
1: 350 kHz
For 11 μA_{PP}: 0: 33 kHz
1: 150 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 correspond to 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 Nonfunctional
Input: 0
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 "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 small
0x00000004	Incorrect position value
0x00000008	Overvoltage
0x00000010	Undervoltage
0x00000020	Overcurrent
0x00000040	Replace battery
0x00000080	Reserved
0x00000100	Reserved
0x00000200	Reserved
0x00000400	Reserved
0x00000800	Reserved
0x00001000	Reserved
0x00002000	Reserved
0x00004000	Reserved
0x00008000	Reserved

Error codes if the access to the encoder via the EnDat interface is faulty:

Error code	Meaning
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 series 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.

Warning

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 6 – 14.

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

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

MP120.x **Nominal speed command outputs of the axes**
Input: 0: No servo-controlled 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 62: Digital nominal speed value output X51 to X62

MP130.x **Y index of the machine parameters MP2xxx.y for the axes**
Input: 0 to 12

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	**Rotational speed**	**Nominal value**	**Position**	**Rotational speed**	**Nominal value**
X1 to X6, X35 to X38	X15 to X20, X80 to X83	Digital: X51 to X56, X57 to X60	X1 to X6, X35 to X38	X15 to X20, X80 to X83	Digital: X51 to X56, X57 to X60
X1 to X6, X35 to X38	X15 to X20, X80 to X83	Digital: X51 to X56, X57 to X60	X1 to X6, X35 to X38	–	Analog: 1 to 12
X1 to X6, X35 to X38	–	Analog: 1 to 12	X1 to X6, X35 to X38	–	Analog: 1 to 12

Maximum spindle speed

The individual PWM outputs are assigned to different controller groups:

- Controller group 1: X51, X53, X54
- Controller group 2: X52, X55, X56
- Controller group 3: X57, X59, X60
- Controller group 4: X58

If all PWM outputs of a controller group are used, the maximum spindle speed is:

$$\frac{60\,000 \text{ rpm}}{\text{No. of pole pairs}}$$

If only the first PWM output of a controller group is used, the maximum spindle speed is:

$$\frac{80\,000 \text{ rpm}}{\text{No. of pole pairs}}$$

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 PWM frequency can be set separately for each of the controller groups, see "PWM Frequency" on page 6 – 233.

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
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 85: Speed encoder inputs X80 to X85
MP113.0	Speed encoder input for the first spindle
MP113.1	Speed encoder input 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 62: Digital nominal speed value output X51 to X62
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/spindles in operating mode 0**
Input:	0 to 12
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/spindles in operating mode 1**
Input:	0 to 12
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 ask for the status of a specific axis or of all axes at once. Bits 0 to 8 represent axes 1 to 9. Bit 15 corresponds to 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 servo-controlled axis (spindle), only display or not active 1: Servo-controlled axis (spindle)
3	Maximum temperature of the motor [°C]
4	0: No 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)

Call:

```
PS    B/W/D/K  <Axis>
               Axis-specific: 0 to 8 represent axes 1 to 9,
               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 detection:

Marker	Value	Meaning
M4203	0	Information was read
	1	Error code in W1022
W1022	1	Status information not available on this iTNC
	2	Axis does not exist

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 marker:

		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

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 Reading 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 detection:

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

		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

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.

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 detection:

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 detection:

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 traversed** Bits 0 to 8 represent axes 1 to 9	NC	NC
W1036	**Negative software limit switch was traversed** Bits 0 to 8 represent axes 1 to 9	NC	NC

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.

MP4050.0-8 Traverse distance for lubrication of axes 1 to 9
Input: 0 to 99 999.999 [m] or [1000°]

		Set	Reset
W1056	**Lubrication pulse: Value in MP4060 exceeded** Bits 0 to 8 represent axes 1 to 9	NC	NC
W1058	**Resetting 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.

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 "-".

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 scan.
- 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 scan.

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 – Close-loop or open-loop axis 0: Close-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 Traversing the reference marks of PLC axes

Traverse the reference marks as for NC axes.

- 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

Enters the feed rate override for one PLC axis.

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)

6.3 PLC Positioning

You can position the axes directly through the PLC. For PLC positioning of the main spindle, See page 6 – 187.

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, see "PLC Axes" on page 6 – 26.

Programming

You start a PLC positioning movement with Module 9221, and you can interrogate the status with Module 9222. After Module 9221 has been called, markers M4120 to M4128 are set (depending on MP4020 bit 2). If you reset these markers, positioning is canceled. 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.

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”
- An EMERGENCY STOP
- An error message that results in a STOP
- A reset of the Markers M4120 to M4128 (depending on MP4020 bit 2)

Module 9221 Starting a PLC positioning movement

Starts a PLC positioning movement in one axis.

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 – Definition of the target position
		0: Absolute, i.e. relative to the machine datum
		1: Incremental
		Bit 1 – Software limit switch
		0: Inactive
		1: Active
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

Module 9222 Status request of PLC positioning movement

With this module you can interrogate the status of a PLC positioning movement.

Call:

PS	B/W/DK	<Axis>
		0 to 8 represent axes 1 to 9
CM	9222	
PL	B/W/D	<Status>
		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”)

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.

Warning

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 spikes during circular movements
- Stiction

Per axis you can activate either the linear or the nonlinear axis error compensation. All other types of compensation are nonexclusive.

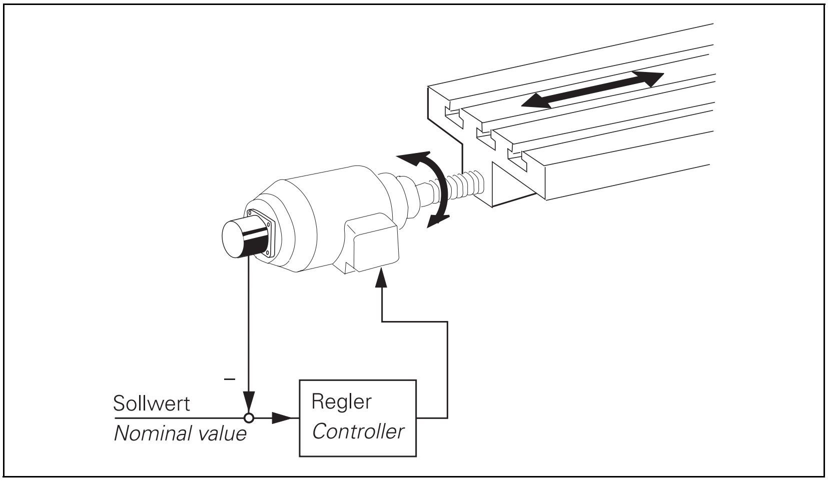
6.4.1 Backlash Compensation

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.



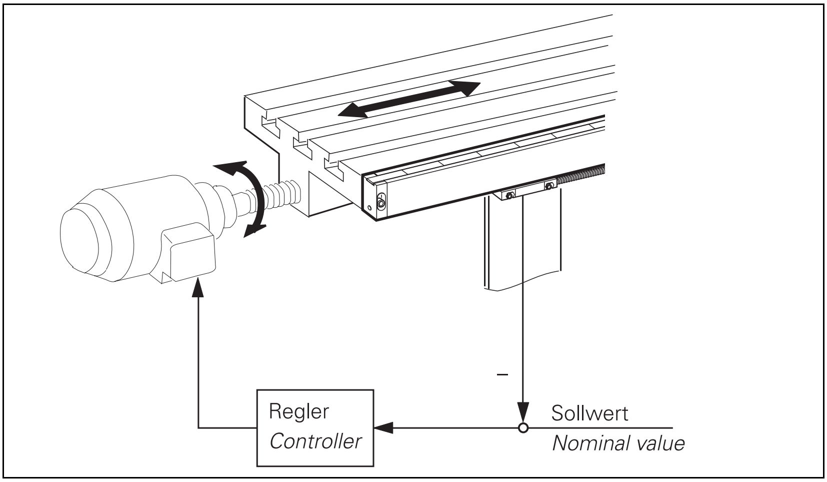
Compensation:

▶ In MP710, enter the value that the iTNC should add to or subtract from the encoder signal after a reversal in direction.

MP710.x **Backlash compensation**
Input: -1.0000 to +1.0000 [mm] or [°]

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: -1.0000 to +1.0000 [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:

$$\frac{0.03\text{ mm}}{15\text{ ms}} = 120\text{ mm/min}$$

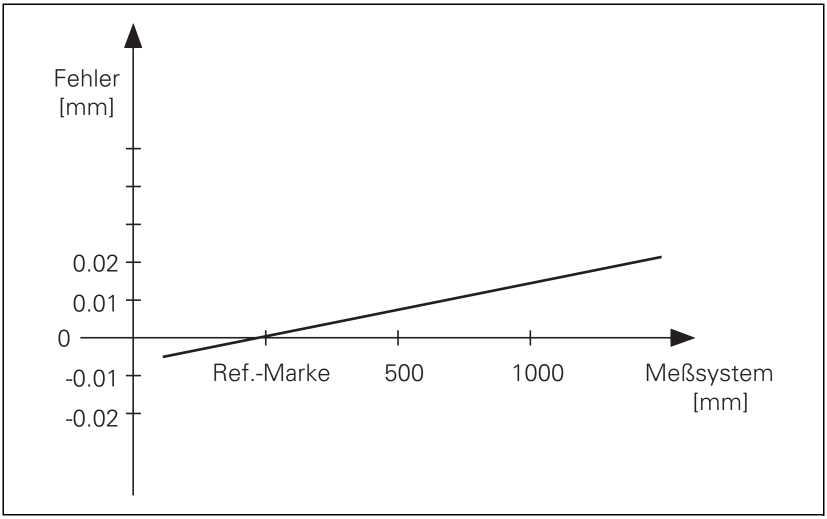
6.4.2 Linear Axis Error Compensation

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:

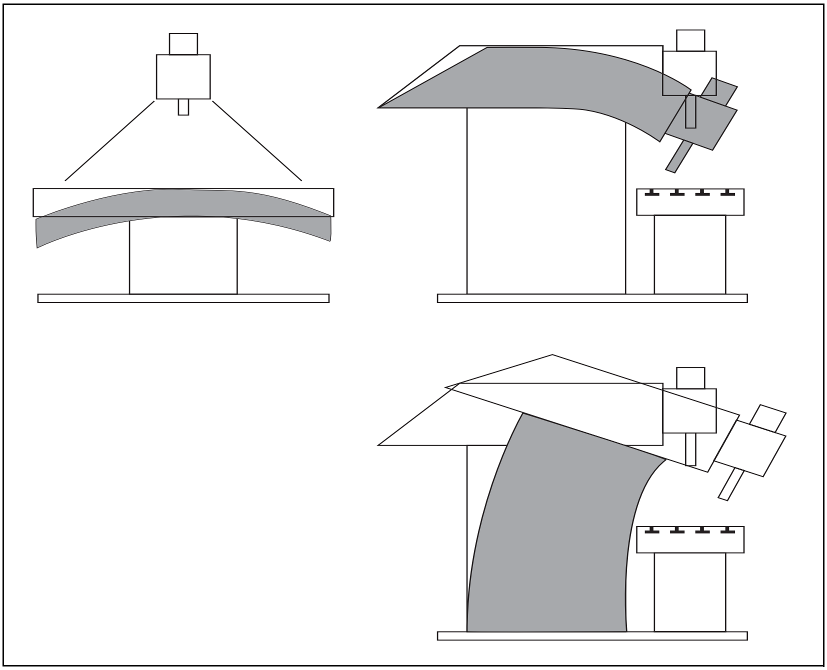
- In MP720, enter the axis error in [mm/m].
- With MP730, activate the linear axis error compensation.

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 correspond to 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:

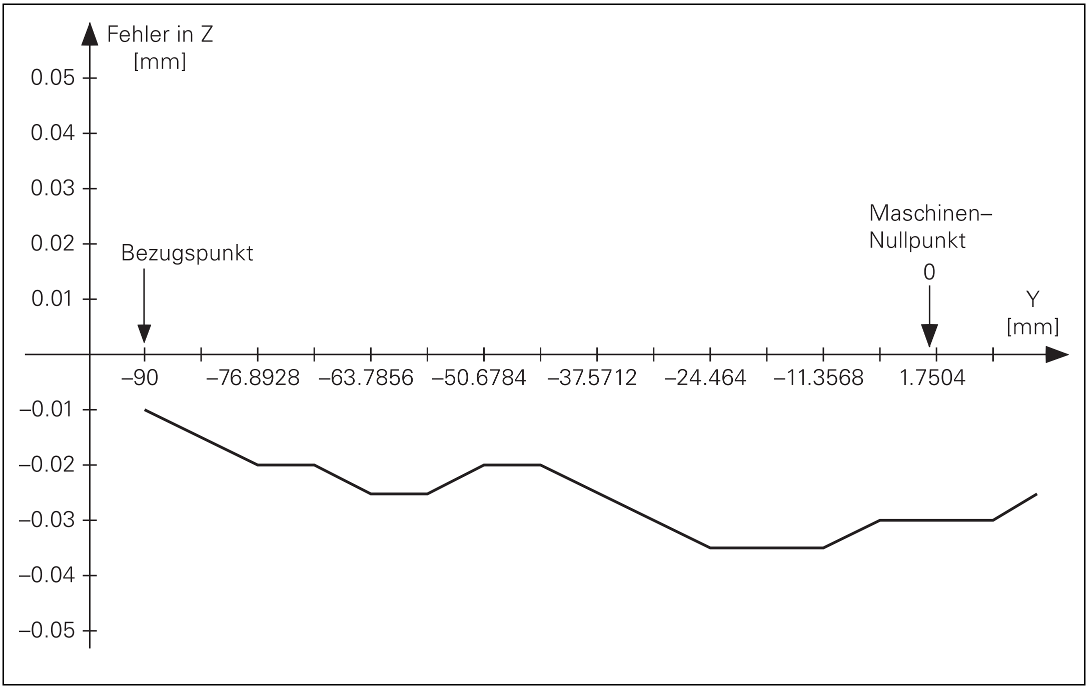


The best way to measure nonlinear axis error is with a comparator measuring system such as the HEIDENHAIN VM 101.

Note

The iTNC can compensate screw-pitch error and axis sag simultaneously.

The following graphic shows the trace of an axis sag error as a function of Y (Z = f(Y)):



Inputting the error trace

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 maximum input value is 23.
  Example: The input value 16 represents 2^{16} = 65536 = 6.5536 mm
- Exit the header by pressing END.
- With the soft key APPEND N LINES, enter the number of compensation points:
  - Maximum of 256 compensation points per column
  - Maximum of 10 columns in all active compensation value tables
  - Total maximum of 1280 compensation points
- To enter compensation values: Enter only the break points of the error trace. The iTNC interpolates linearly between the break points.

Example

The following dependencies apply for axes 2 = Y and 3 = Z:

- Ballscrew 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

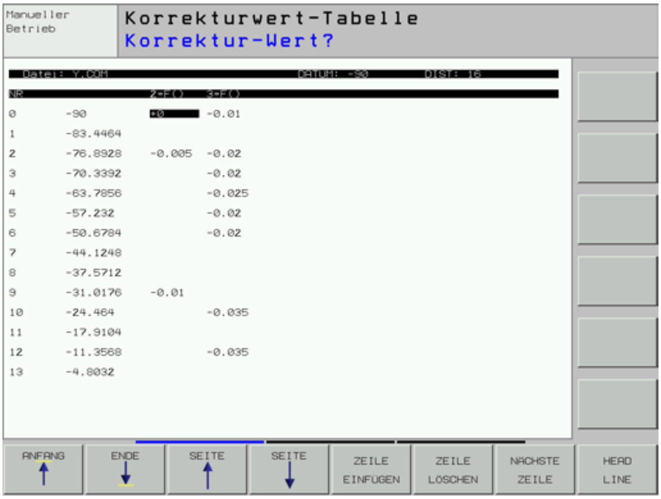
Calculations:

Input values for the spacing of the compensation points:
possible powers of 2^{16} = 6.5536 mm

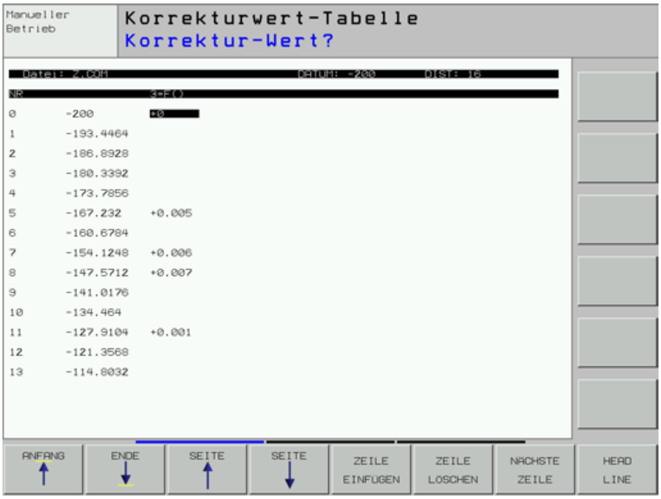
Number of compensation points:

$$\frac{500 \text{ mm}}{6.5536 \text{ mm}} = 77 \text{ compensation points in Y}$$

$$\frac{800 \text{ mm}}{6.5536 \text{ mm}} = 123 \text{ compensation points in Z}$$



Y axis:
Ballscrew-pitch error in column 2 = F(), sag error in column 3 = F()



Z axis:
Ballscrew-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 SET ACTIV LINE or with Module 9095. With Module 9035 you can interrogate the active line.
- ▶ Enter the complete name of the configuration file with the extension CMA in the system file OEM.SYS with the command TABCMA=.

Example

Entry in the configuration file for axes 2 = Y and 3 = Z:

Compensation-value table valid for 20° = AXIS-Y.COM and AXIS -Z.COM
Compensation-value table valid for 35° = AXIS -YT.COM and AXIS -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 Reading status information

Call:

PS	B/W/D/K	<19> Active line in the configuration file (*.CMA) –1: No CMA file active
CM	9035	
PL	B/W/D	<Active line number> 0: Line number –1: No CMA file active

Error detection:

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

A rotary axis is a special case

For a rotary axis, only the compensation values for the entries of 0° to +60° are effective, relative to the machine datum. Therefore, the datum for the nonlinear compensation must lie within the 0° to +360° range. To compensate a full circle, set the compensation value datum to the machine datum.

Example: Rotary axis from –180° to +180°

Rotary axis:	0	...	+180	...	–179	...	–1	...	0
Corresponding angle for compensation values:	0	...	+180	...	+181	...	+359	...	0

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 the kinematics table. See "Temperature Compensation with Tilting Axes" on page 6 – 73.

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:

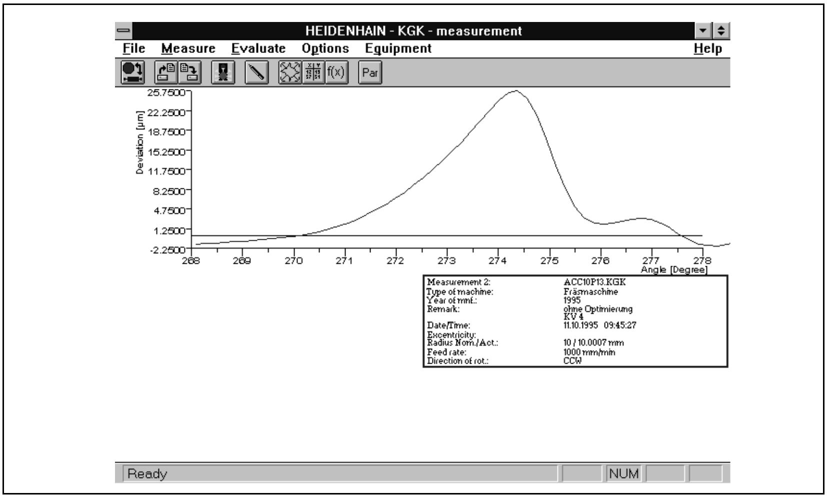
PS	B/W/D/K	<Axis> Axes 0 to 8
PS	B/W/D/K	<Compensation value> Range: –30 000 to +30 000 [1/10 µm]
CM	9231	

Error detection:

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 static 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:

$$\text{Comp. [mm =]} \frac{\text{Reversal peaks [µm]} \cdot \text{control loop cycle time} \cdot [s] \cdot 10^{-3}}{0.5 \cdot t_{SpD}[s]}$$

The compensation value is entered in MP712.x.

Compensation

Digital axes:

You must compensate friction in the range of the speed controller (MP2610 to MP2620). Do not compensate with MP711 to MP716, see "Compensation of Sliding Friction (Only for Digital Axes)" on page 6 – 48.

Analog axes:

If you have compensated the **backlash** with MP750, there should be no more reversal spikes. If there are, compensate them with MP711 to MP716.

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 (only analog)**
Input: -1.0000 to +1.0000 [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 (only analog) with M105**
Input: -1.0000 to +1.0000 [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 Static Friction

On guideways with high static friction (stick-slip 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 static 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 static friction, an additive nominal velocity is output whose value F_{zus} is calculated from the factor for static 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 static 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 static 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 static friction compensation (approximate value: 5000 to 10 000).
- In MP1512.x, enter a limit for the amount of the static friction compensation (approx. value: < 50).
- In MP1513.x, limit the maximum feed rate up to which the static friction compensation remains in effect.

MP1511.x **Factor for static friction compensation**
Input: 0 to 16 777 215 [µs]

MP1512.x **Limitation of the amount of the static friction compensation**
Input: 0 to 16 777 215 [counting steps]

MP1513.x **Feed-rate limitation for static friction compensation**
Input: 0 to 300 000 [mm/min]

MP1391 **Velocity feedforward control in the MANUAL and HANDWHEEL operating modes**
Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Operation with following error (lag)
1: Operation with velocity feedforward control

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.

MP2512.x **Limiting the 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 speed nominal 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 30.0000 [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 30.0000 [A] 0: No friction compensation (or axis is analog)

6.5 Tilting Axes

Swivel heads and tilting tables are often used on milling machines 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** 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

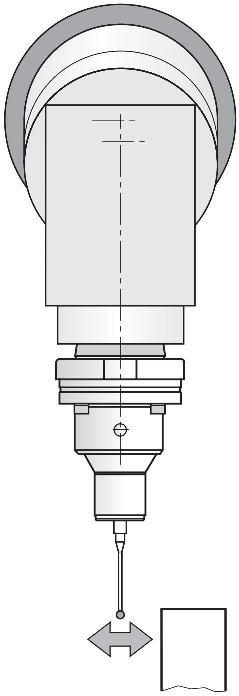
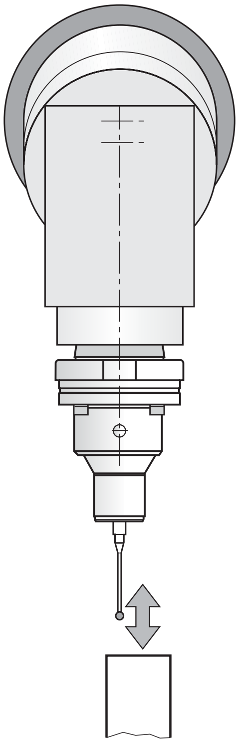
The 3-D ROT function must be inactive during the entire measuring process.

Double swivel head 45°

In this example, the mechanical offset of a double swivel head is determined by using a 3-D touch probe.

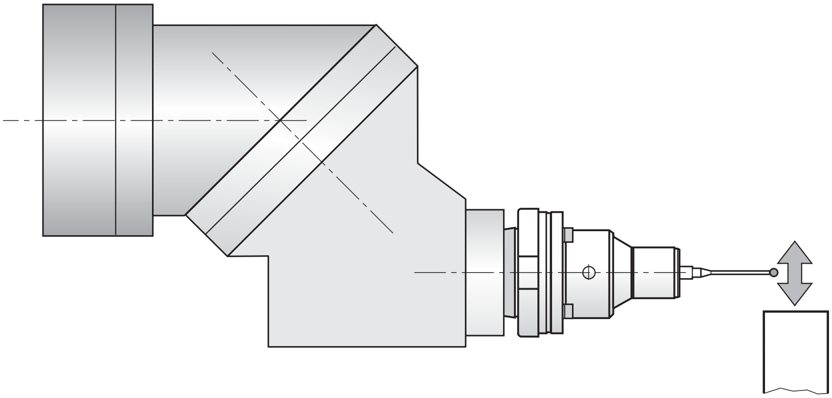
Input values for the machine parameters:

- 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

Step 1a ■ Move tilting axes B and C to 0° position ■ Probe surface X1 ■ Set X = 0	
Step 1b ■ Probe surface Z ■ Set Z = 0	

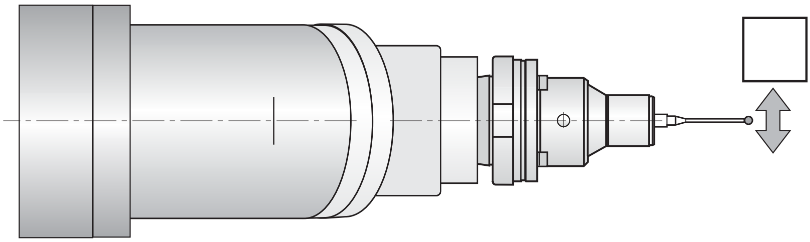
Step 2

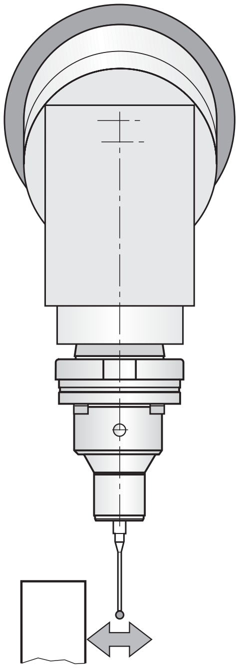
- Position C = 180
- Probe surface Z
- MP7530.0 = determined value – probe length + ball radius



Step 3

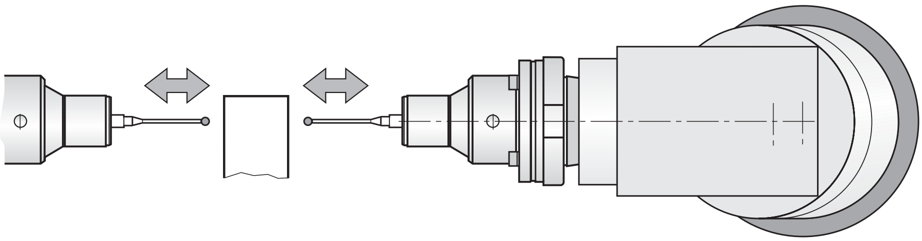
- Probe surface X1
- MP7530.1 = -0.5 * determined value



Step 4a	
■ Position C = 0 ■ Probe surface X2 ■ L = determined value	

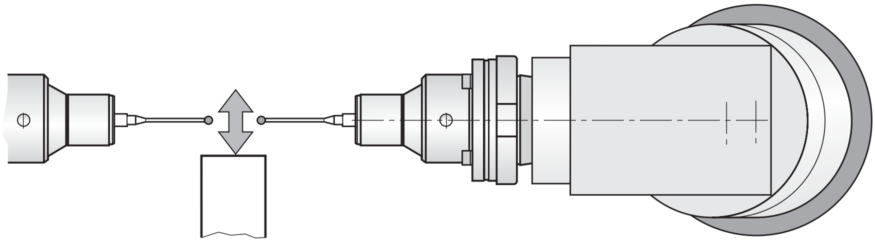
Step 4b

- Position B = -90
- Probe surface X1
- ΔX1 = determined value
- Position B = +90
- Probe surface X2
- ΔX2 = determined value
- MP7530.3 = {[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
- MP7530.6 = –0.5 * determined value – MP7530.1

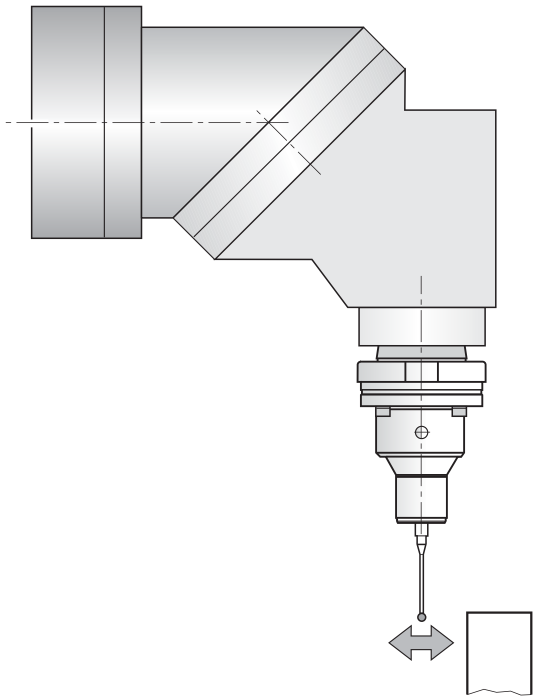


Step 5

- 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 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.

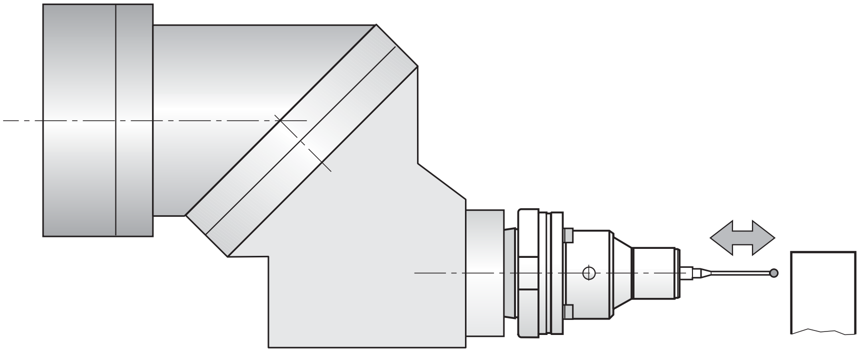
Step 6

- Probe surface Y2
- Set Y = 0



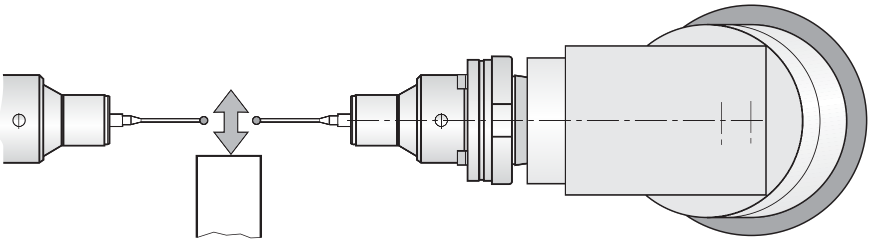
Step 7

- 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)

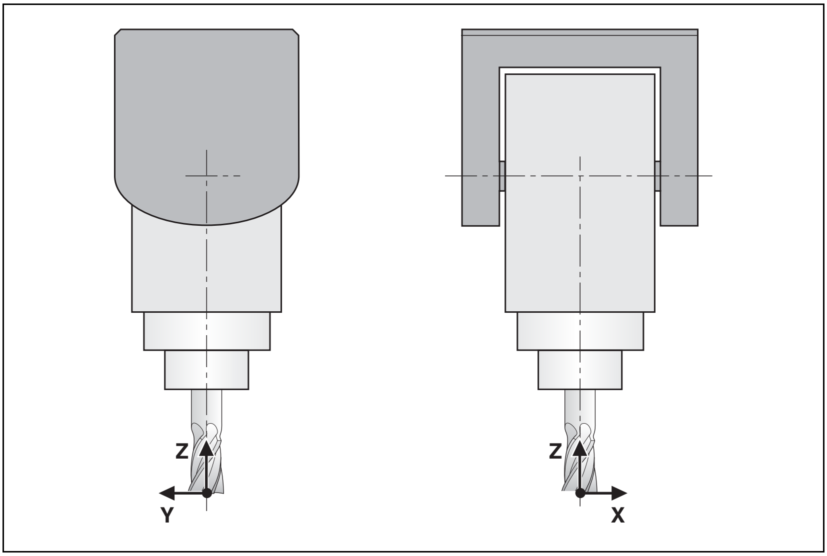


Step 8

- 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)



Forked swivel head In this example, the mechanical offset of a forked swivel head is determined with a dial indicator and a cylinder with a known diameter.



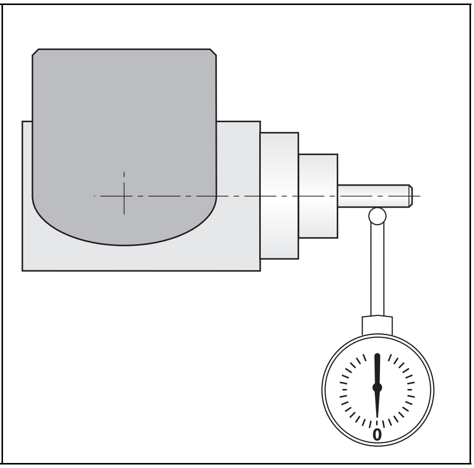
Temporary input values for the machine parameters:

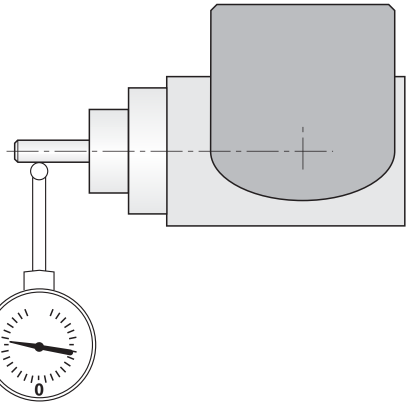
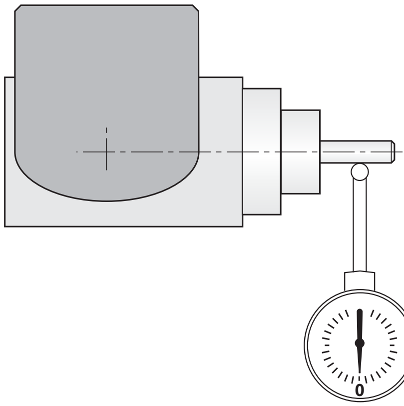
- MP7500 = %xxxx101
- MP7510.0 = %000100 ; Shift in Z axis
- MP7510.1 = %000010 ; Shift in Y axis
- MP7510.2 = %001000 ; Free tilting axis A
- MP7510.3 = %000001 ; Shift in X axis
- MP7510.4 = %000010 ; Shift in Y axis
- MP7510.5 = %100000 ; Free tilting axis C
- MP7510.6 = %000000 ; End transformation

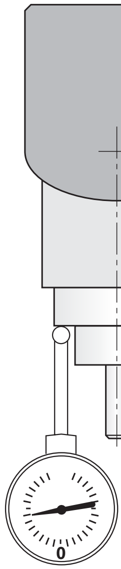
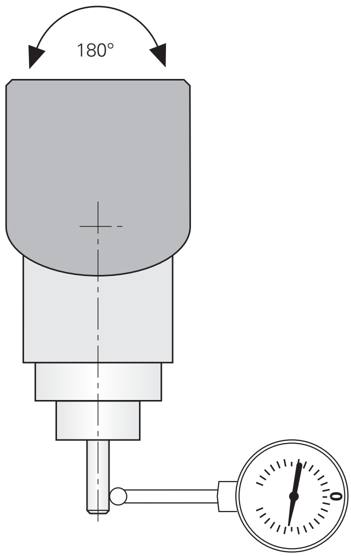
Step 1a

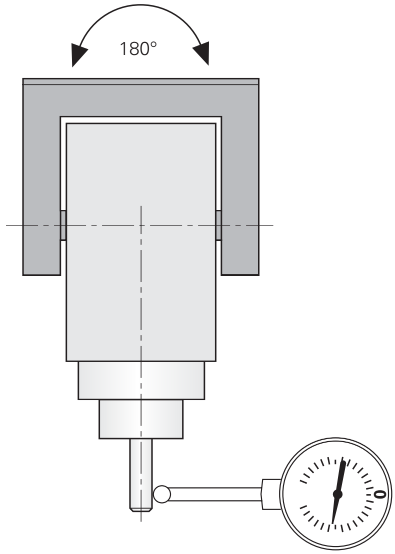
Determining the Y offset:

- Position A = -90
- Set dial indicator to 0



Step 1b Determining the Y offset: ■ Position A = +90 ■ Offset = 0.5 * determined value ■ If the determined value > 0, then MP7530.1 = **–** offset ■ If the determined value < 0, then MP7530.1 = **+** offset	
Step 1c Checking the settings: ■ Activate 3-D ROT ■ Position A = +90 ■ Set dial indicator to 0 ■ Set datum ■ Position A = -90 ■ Probe same position again ■ Display and dial indicator must read 0	
Step 2a Determining the Z offset: ■ Position A = -90 ■ Set dial indicator to 0	

Step 2b Determining the Z offset: ■ Position A = 0 ■ Place dial indicator against the spindle nose ■ **MP7530.0** = determined value – cylinder radius	
Step 3 Determining the Y offset: ■ Position A = 0 ■ Position C = 0 and set C = 0 ■ Position C = 180 ■ Read Y offset from the dial indicator ■ **MP7530.4** = (0.5 * determined value) – MP7530.1	

Step 4 Determining the X offset: ■ Position C = –90 and set C = 0 ■ Position C = 180 ■ Read X offset from the dial indicator ■ MP7530.3 = 0.5 * determined value	

6.5.2 Describing the Mechanical Offset

> **Note**
>
> The description of the mechanical offset is only taken into account for tilting functions, such as M128 or the "tilted 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.
- In MP7550.x, enter the home position of the tilting device 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 M128, etc. See page 6 – 74. A formula for a permanently effective temperature compensation may be entered in the TEMPCOMP column. See page 6 – 74.

Working with the description of the mechanical offset in tables:

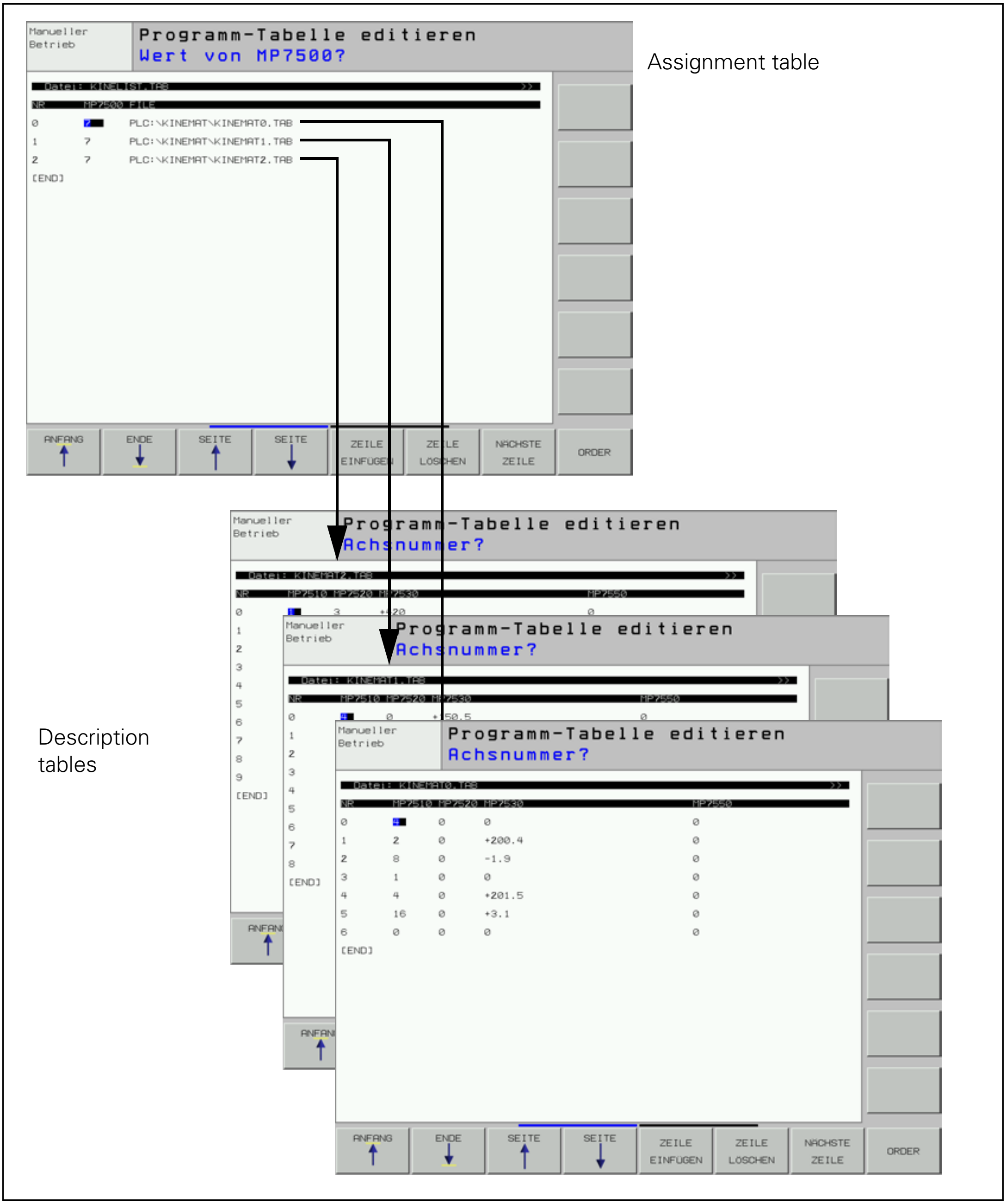
- Switch to the **`Programming and Editing`** operating mode, press the MOD key and enter the code number 807667.
- Choose the file PLC:\OEM.SYS from within program management.
- Enter the code word **`KINEMATIC=`** , followed by the file name with its complete path from the assignment table.
  (e.g., **`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.

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** when a new datum is set.



Assignment table
Description tables
Programm-Tabelle editieren
Wert von MP7500?
Achsnummer?

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]					

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 EDIT FORMAT 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.

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.

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 detection:

Marker	Value	Meaning
M4203	0	Name and/or line number was found
	1	Error code in W1022
W1022	2	Incorrect parameter for string number
	20	Module was not called in a submit job or spawn 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.

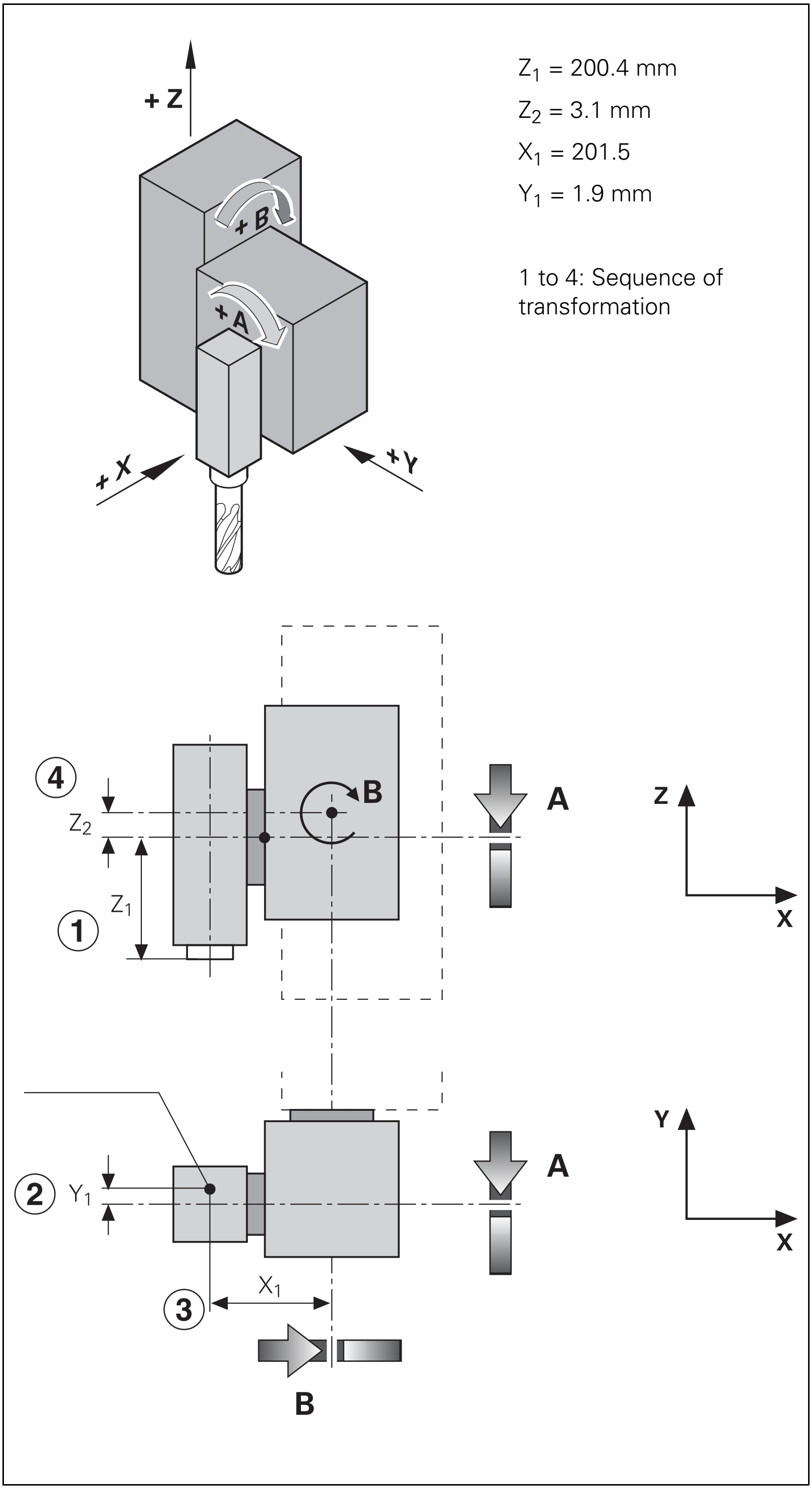
Call:

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

Error detection:

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

Example 1: Rectangular double swivel head

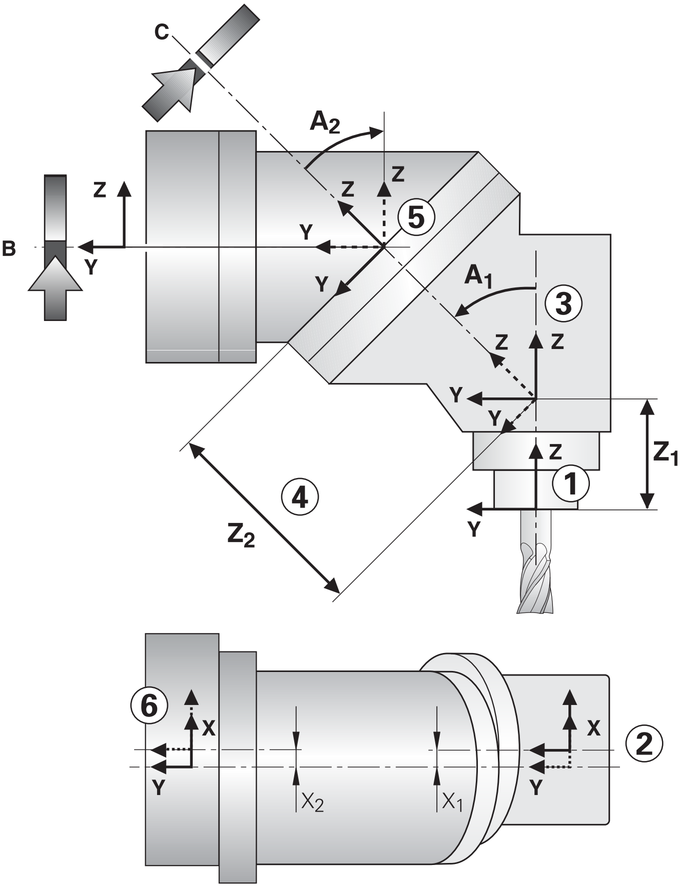


```
MP 7510.0    : %000100    ;Shift in Z axis (Z1)
MP 7510.1    : %000010    ;Shift in Y axis (Y1)
MP 7510.2    : %001000    ;Free tilting axis A
MP 7510.3    : %000001    ;Shift in X axis (X1)
MP 7510.4    : %000100    ;Shift in Z axis (Z2)
MP 7510.5    : %010000    ;Free tilting axis B
MP 7510.6    : %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)
```

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

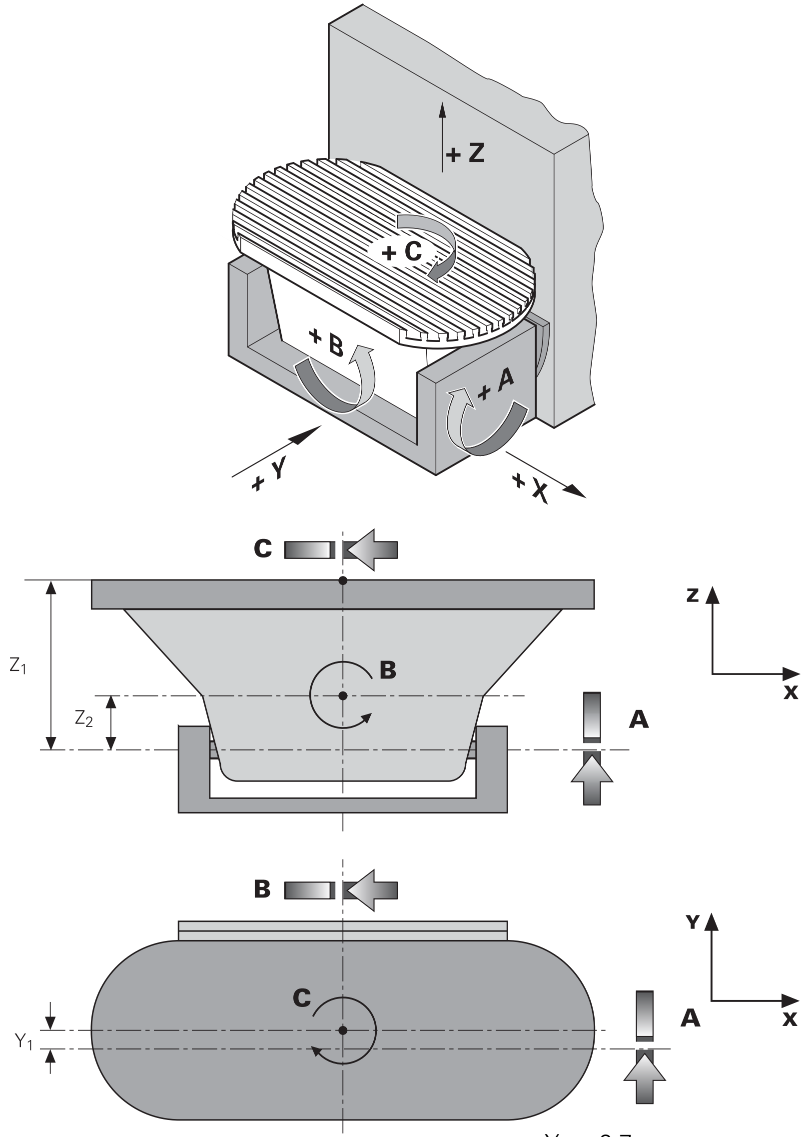
1 to 6: Sequence of transformation

```
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 (A1)
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)
```

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:

X_R = 420.0 mm
Y_R = 151.2 mm
Z_R = –395.4 mm

```
MP 7510.0   : %000001   ;X coordinate of the center of rotation of
                         axis C
MP 7510.1   : %000010   ;Y coordinate of the center of rotation of
                         axis C
MP 7510.2   : %000100   ;Z coordinate of the center of rotation of
                         axis C
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)
```

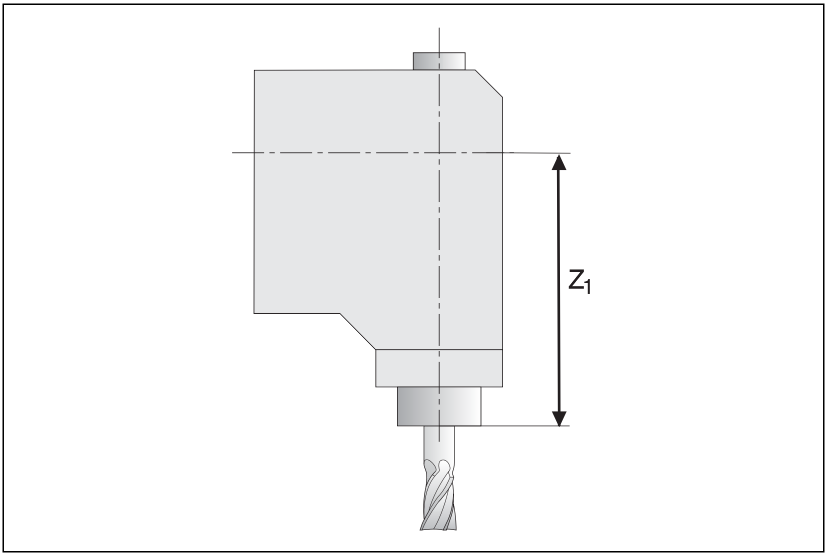
6.5.3 Temperature Compensation with Tilting Axes

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 with a "tilted working plane"
  - by entering a formula in MP7530.x
  - 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: $\Delta l = l \cdot \Delta T \cdot \alpha$
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 or TEMPCOMP = 300 + 300 * 11.5e–6 * (W486 – 20)
better: MP7530.x or TEMPCOMP = 300 + 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
- Maximum of 16 variables per formula
- 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

An erroneous syntax of the formula is not recognized until the NC program is started. The error message `MP75xx not defined` appears.

Temperature compensation

If the "tilted working plane" function is active, the position of the tilting element is calculated for each positioning movement.
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.

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.
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:

NR	MP7510	MP7520	MP7530	TEMPCOMP	MP7550
0	4	0	+150.5	+150.5+1.73e–3*(W486-20)	0
1	1	0	–1.2		0
2	8	0	–45		0
3	4	0	+251.5		0
4	32	0	0		0
5	8	0	+45		0
6	1	0	+0.8		0
7	16	0	0		0
8	0	0	0		0
[END]					

6.5.4 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. 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.
- 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

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.

- 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 correctly entered again.
- Switch on drive for the new axis/axes.
- After the drive has been switched on, the affected motors should make at least one revolution.

6.5.5 "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.

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

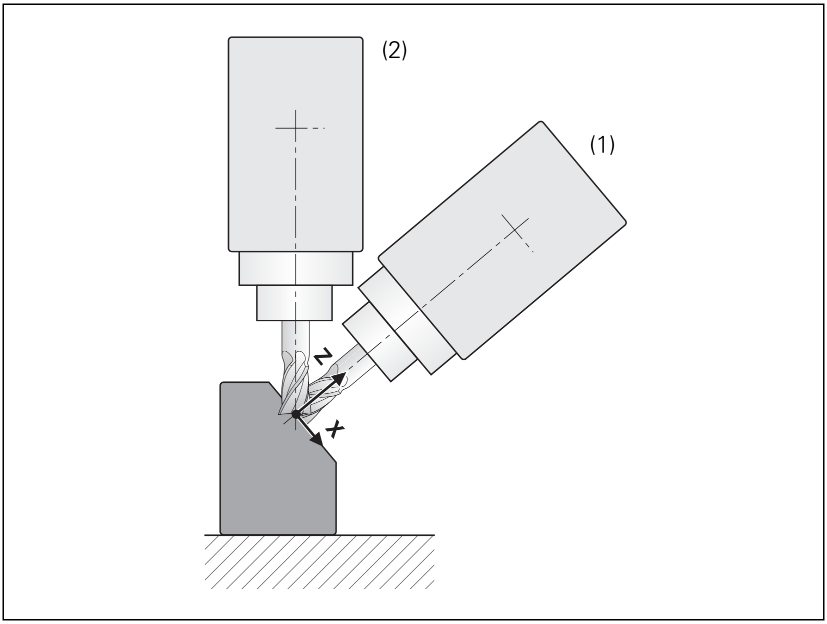
After the coordinate transformation, the Z axis remains parallel to the tool axis, perpendicular to the X/Y plane. With MP7500 bit 2 you define whether the "tilted working plane" function automatically positions the tilting axes (bit 2 = 1). In this case the user can enter the feed rate and setup clearance 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.

Datums

Servo-controlled 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.**



Behavior during datum setting can also be influenced via MP7500 bit 5:

MP7500 bit 5 = 0
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
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).

No servo-controlled 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.

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.

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.

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
- Swivel head 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

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

With tool axis X:

- Universal swivel head: Axis sequence B fixed; A variable; B fixed; C variable

MP7500 **“Tilted working plane”**
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 – Nonfunctional
Bit 4 – Nonfunctional
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 – Nonfunctional
Bit 8 – Nonfunctional

MP7510 **Transformed axis**
Format: %xxxxxx
Input: 0: End of the transformation sequence
Bit 0 represents axis X
Bit 1 represents axis Y
Bit 2 represents axis Z
Bit 3 represents axis A
Bit 4 represents axis B
Bit 5 represents 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
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:	-99 999.9999 to +99 999.9999
	0: Free tilting axis
MP7530.0–14	Transformation 1 to transformation 15

Note

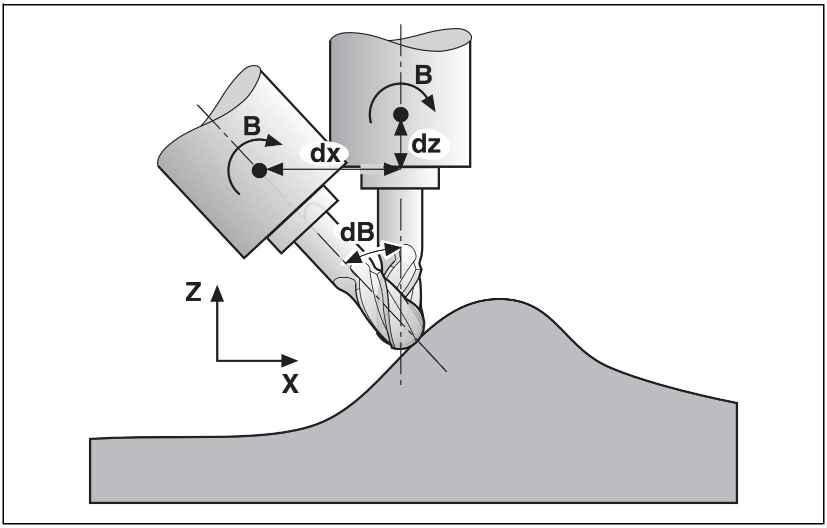
MP7530 cannot be overwritten with Module 9031 (overwrite machine parameters), since the MP contains a string, but the module transfers an integer value.

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**
Format:	%xxx
Input:	Bit 1 – Reference value for calculating the preset during "datum setting"
	0: Actual value is calculated
	1: Nominal value is calculated

6.5.6 Automatic Compensation of Offset for Tilting Axes

Unlike the “tilted working plane,” here the coordinate system is not tilted. With M114 or M128, 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.

Miscellaneous function M128

Automatic compensation with M128:

- 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 remains 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 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.

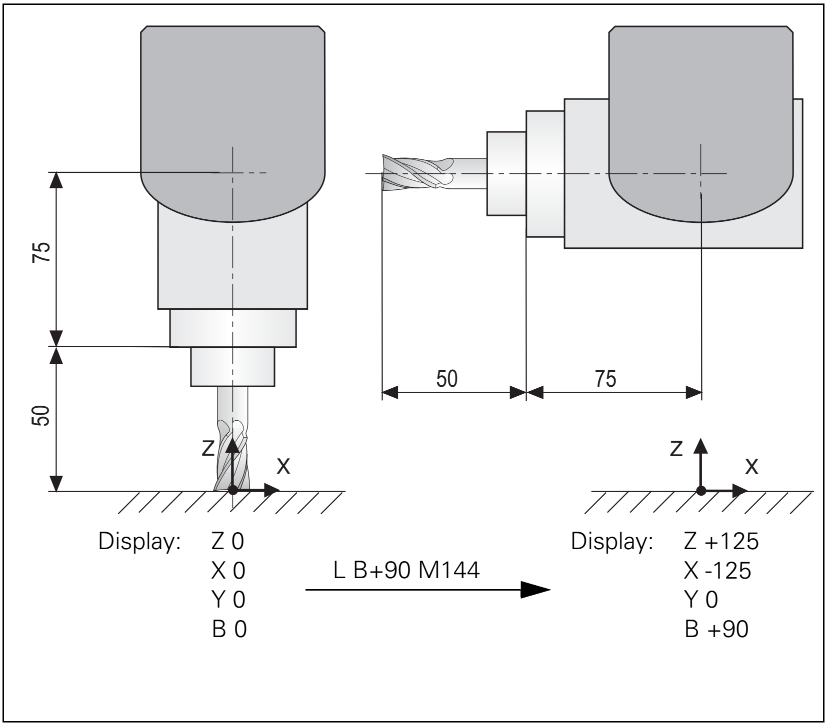
When M128 is used, the principal axes make compensating movements:

- In MP7471, define the maximum velocity of the principal axes during compensating movements.
- In MP7684 bits 5 and 6, set how the compensatory movements of the reference axes are to affect the feed-rate reduction at corners with M128.

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**
Input:	0 to 300 000 [mm/min]
MP7684	**Nominal position value filter and path control with M128**
Format:	%xxxxxxxx
Input:	Bit 5 - Feed-rate reduction at corners with M128 0: Include only maximum compensatory movement 1: Do not include compensatory movements Bit 6 - Feed-rate reduction at corners with M128 0: Include compensatory movements depending on the entry in bit 5 1: Include all compensatory movements

Miscellaneous function M144/ M145

With M144 the movement of a tilted axis is recorded in the display. There is no need for the axes to traverse a compensatory path. 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

6.5.7 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:

- In MP7510 to MP7530, the center of rotation of a rotary axis must be defined (see example 3). MP7500 is not needed if only one rotary axis is present.
- If a 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.

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 only move simultaneously. The main axis is referred to as the master, and the tracking axis as the slave. From a maximum of nine controlled axes, four times two axes can be controlled synchronously.

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 (MP860.x = 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.

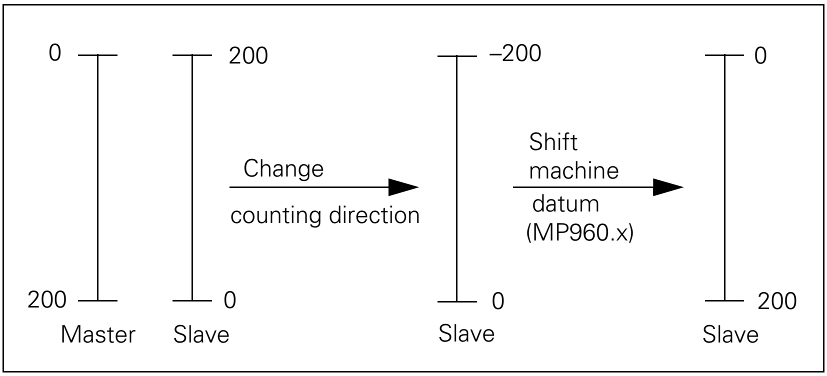
- For gantry axes, one position encoder is sufficient.
- The nonlinear axis-error compensation can be used separately for master and slave axes.
- For the nonlinear axis-error compensation, master and slave axes may be dependent on each other.

MP855.x	**Synchronization monitoring**
Input:	0 to 100.0000 [mm] 0: Monitoring not active

MP860.x	**Datum for synchronous control**
Input:	0: Datum at position after switch-on 1: Datum at reference marks

Example

- Gantry axes with two position encoders
- Position encoder of the slave axis is mounted mirror-inverted.



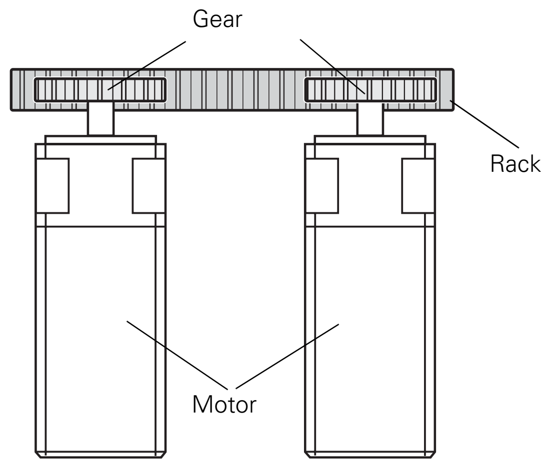
6.6.2 Master-Slave Torque Control

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.

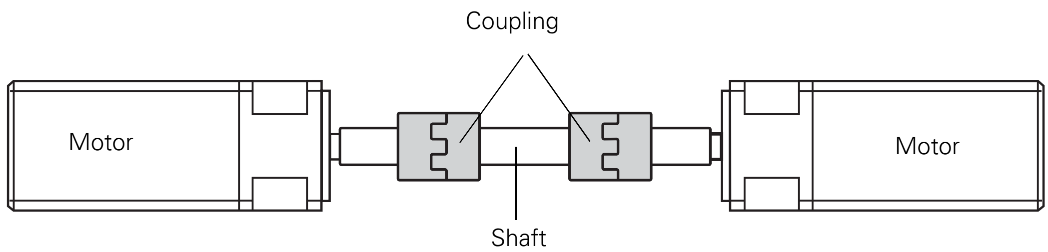
From a maximum of nine controlled axes, four times two axes can be controlled in the torque-master-slave-control, whereby you must keep in mind that the master and slave axis are on the same speed controller PCB.
First speed controller PCB: X15 to X20
Second speed controller PCB: X80 to X83

In principle there are two applications:

- Minimization of mechanical play through mutual tensioning
- Distribution of torque with a rigid coupling

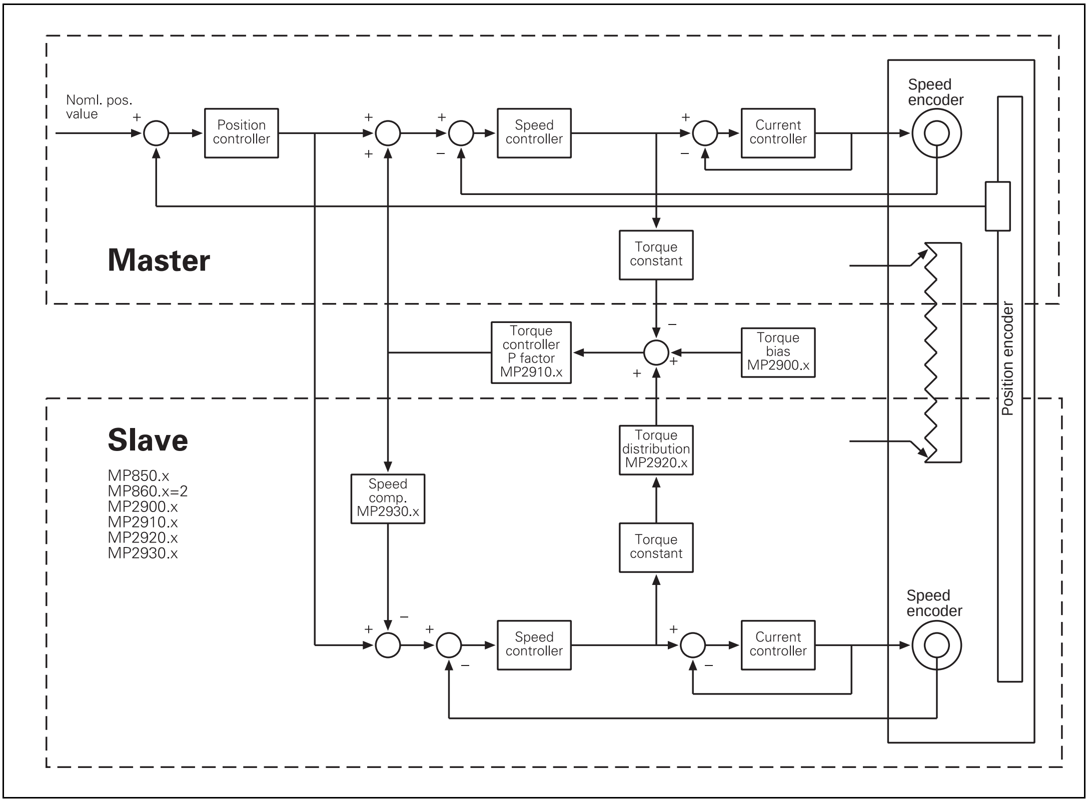


Minimization of mechanical play through mutual tensioning



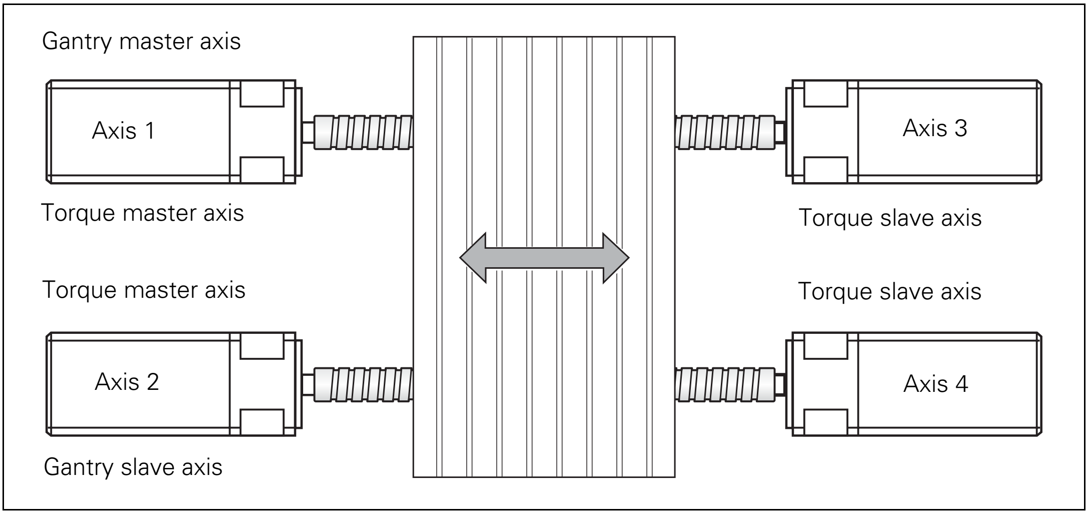
Distribution of torque with a rigid coupling

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 = 2 for the slave axis.

MP860.x	**Datum for synchronous control**
Input:	2: Axis is torque slave 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 6 – 228.
- 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, in MP2930.x enter 100 for the slave axis; if you do not use a position encoder, enter the value 0.
- 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 "Commissioning" on page 6 – 228. 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

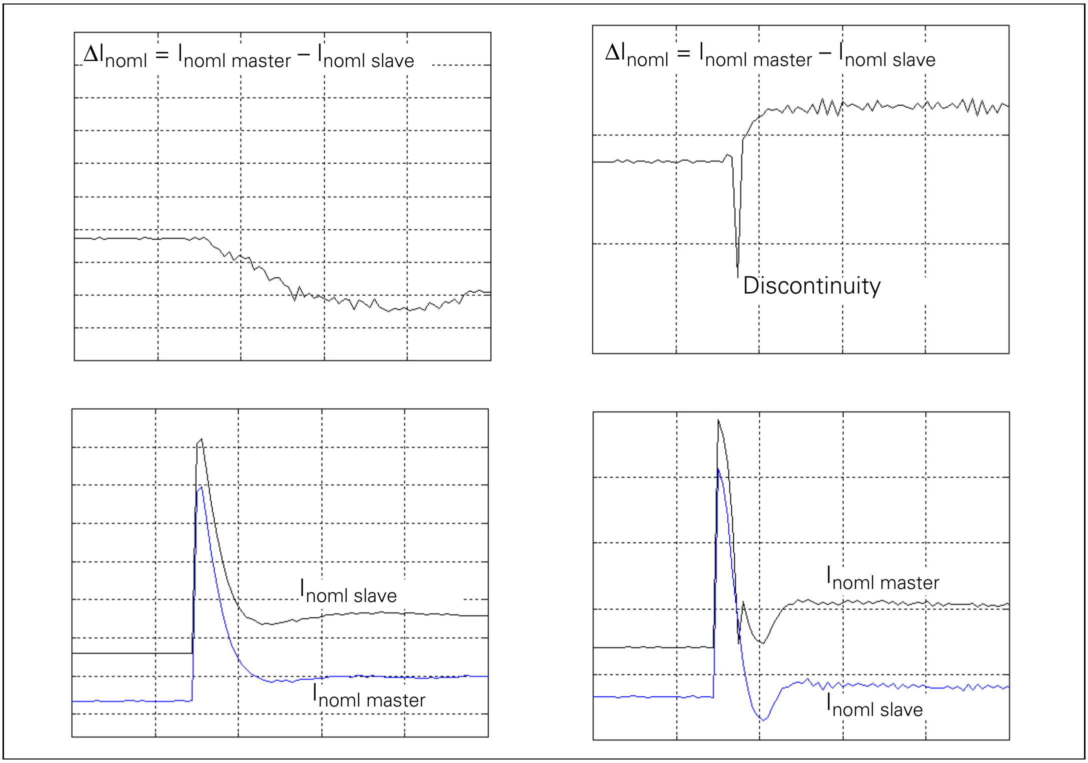
For low-frequency oscillations (< approx. 200 Hz) you should not use a filter, because it may have a negative influence on the dynamics of the control.
For the mid-range frequency (approx. 200 Hz to approx. 400 Hz) ensure that you do not excite any low-frequency oscillation.
The higher the frequency 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 "Commissioning" on page 6 – 228.
- 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.
- 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 "Commissioning" on page 6 – 228.
- 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

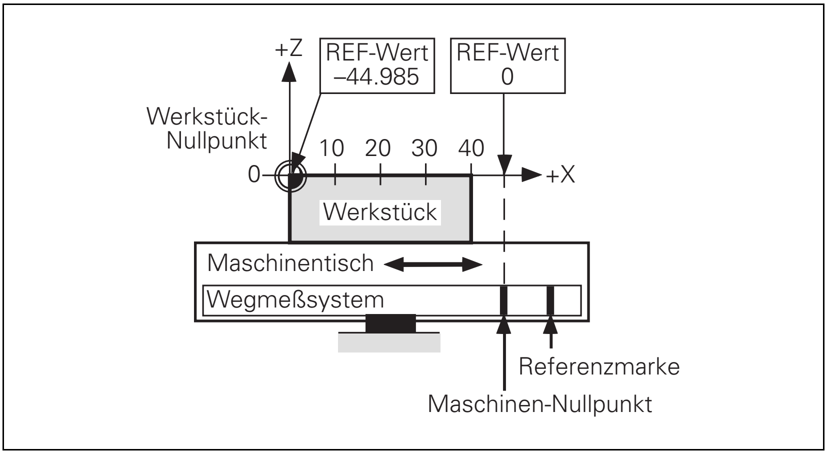
The position value (the coordinates) of an axis position is defined with respect to a freely selectable datum. When the axes are moved, the ACTUAL position is calculated incrementally. If there is an interruption in power, the reference between the axis position and the position value is lost.

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 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 detection:

Marker	Value	Meaning
M4203	0	No error
	1	Axis does not exist

Assigning a reference value

In some cases a new reference mark 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 detection:

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 job 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 it is not possible to use the reference mark of the encoder, for example due to an unsuitable transmission ratio between the motor and rotary axis, then you can use an external reference pulse:

- 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

"Pass Over Reference Point" mode of operation

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) are reset.

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.
To synchronize the current machine status and the look-ahead calculation with an NC macro call,

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."

Encoders with EnDat interface

Encoders with EnDat interface can be connected to the position and speed inputs of the MC 422 and CC 422. 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.

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.

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 300 000 [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 300 000 [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 **Type of reference-mark traverse**
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 over fast PLC input

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 [°]

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
W272	**Mode of operation** 1: MANUAL OPERATION 2: ELECTRONIC HANDWHEEL 3: POSITIONING WITH MANUAL DATA INPUT 4: PROGRAM RUN, SINGLE BLOCK 5: PROGRAM RUN, FULL SEQUENCE 7: REFERENCE MARK TRAVERSE	NC	NC
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 position is 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 for traversing the reference mark the first time.
- An axis cannot be started for reference mark traverse until all axes are in position.

Servo-controlled spindles:

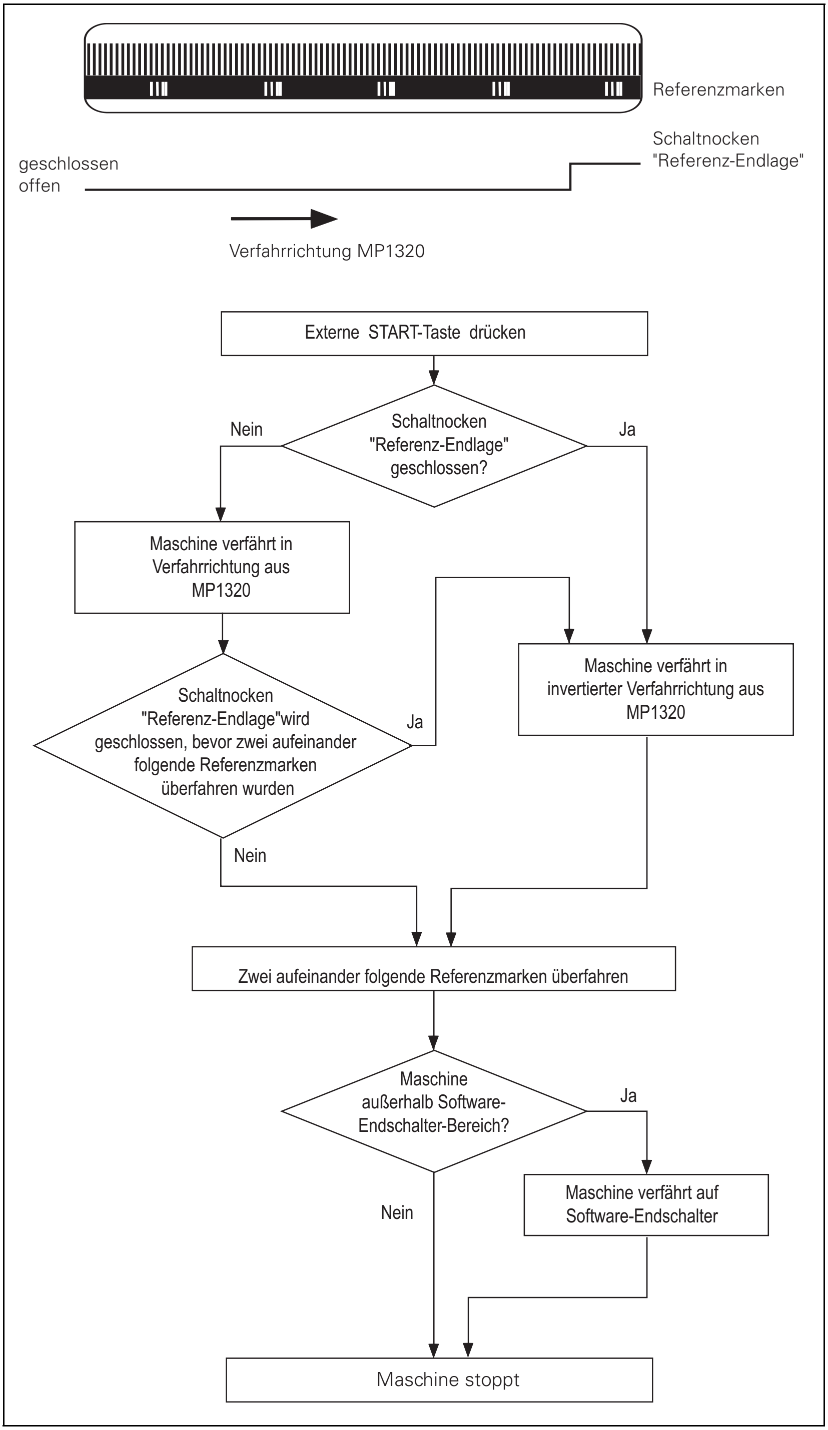
- The speed for traversing the reference mark is defined in the module.
- The spindle must be started from a standstill to traverse the reference mark.
- If the spindle is started for reference mark traverse, marker M4018 is set.

Call:

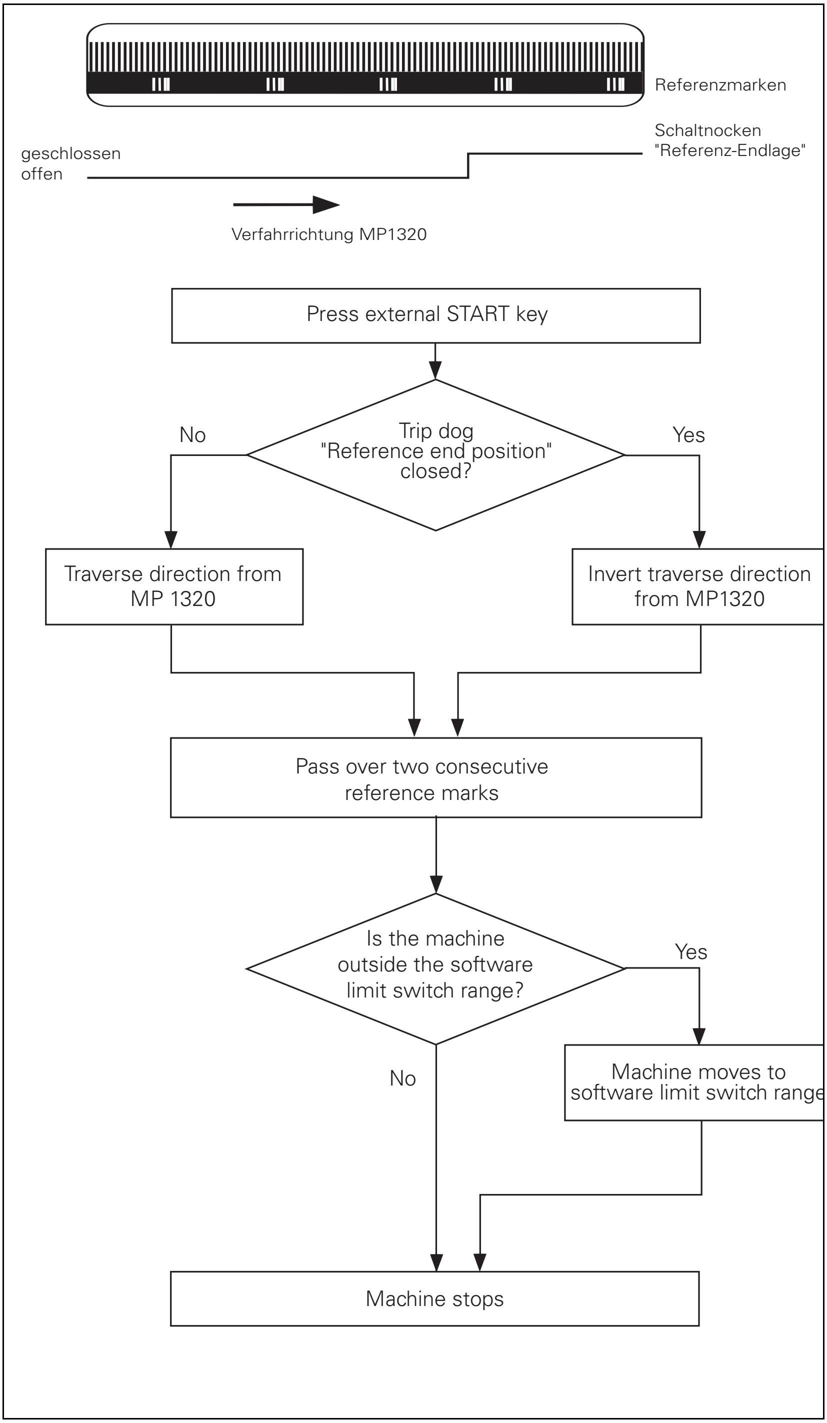
PS	B/W/D/K	<Axis/spindle> 0 to 8: Axes 1 to 9 15: Spindle
PS	B/W/D/K	<Feed rate/shaft speed> 0: Feed rate MP1330.x >0: Feed rate in mm/min or shaft speed in 1/1000 rpm
PS	B/W/D/K	<Direction of traverse> –1: Negative direction 0: Direction from MP1320.x 1: Positive direction
CM	9220	
PL	B/W/D	<Error code> 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



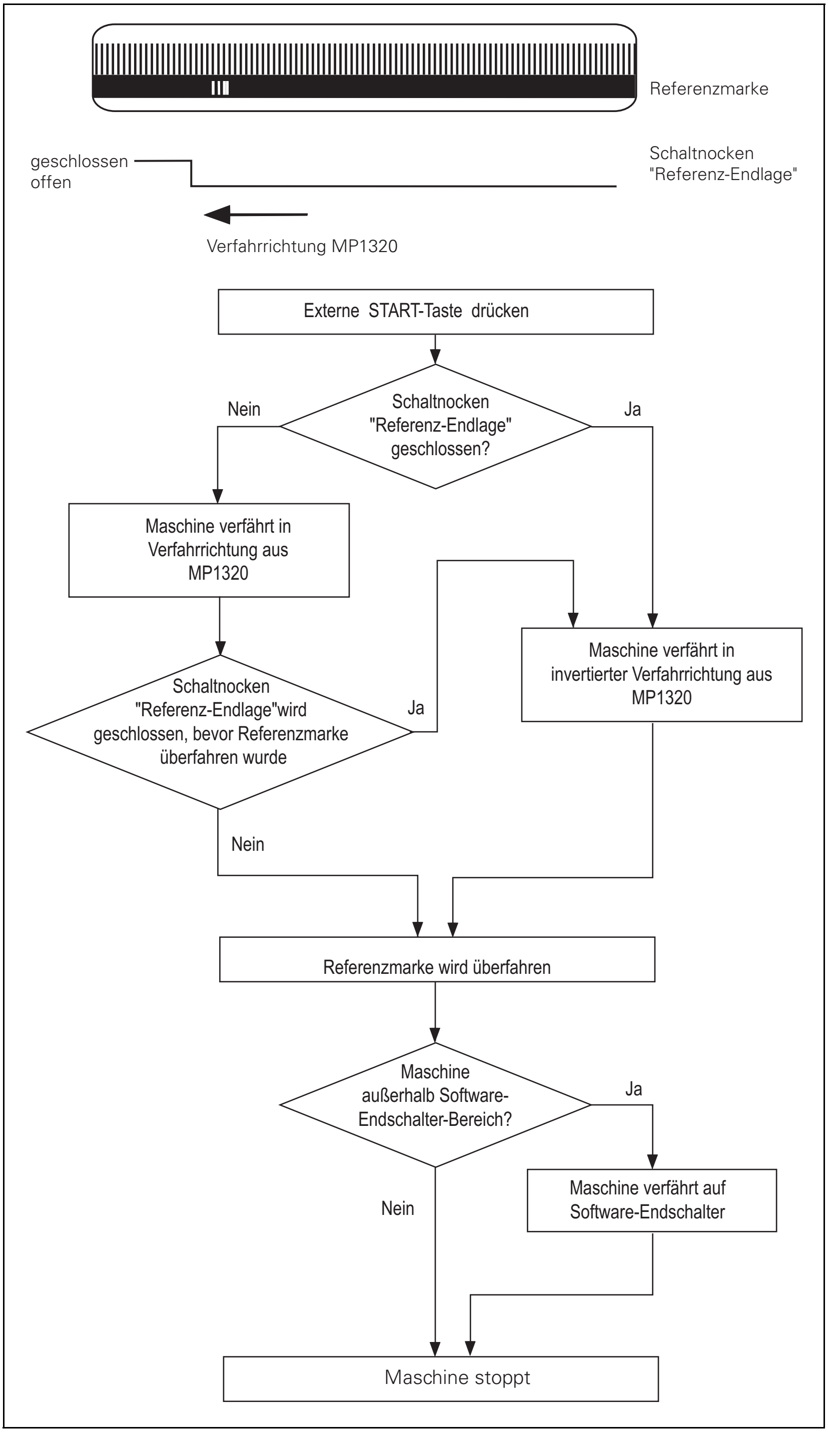
Function when MP1350.x = 0. This setting is used only to ensure compatibility. Do not use for new installations.



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 be at least two reference marks in the range of the reference end position.

Position encoder with one reference mark

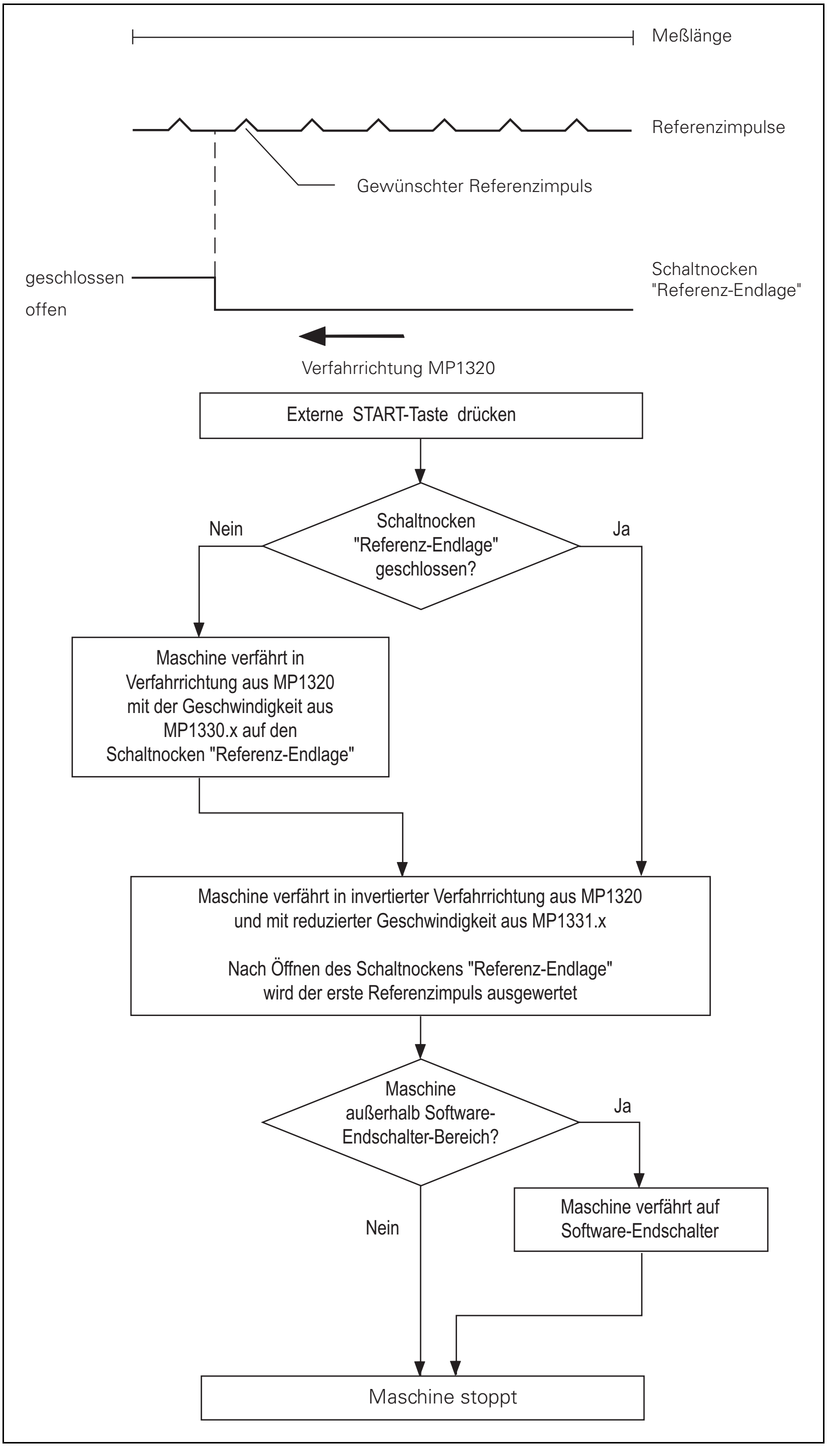
Function when MP1350.x = 1



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

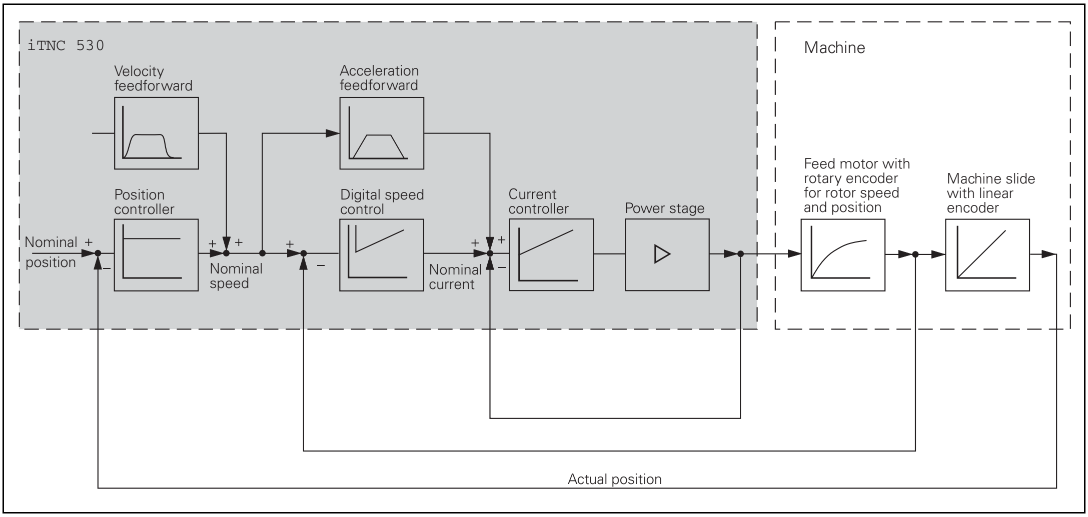
Machine tools normally function on the principle of cascade control. Here the position control loop is prior to the speed and current control loops.

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 422 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 rpm for motors with two pole pairs are possible.



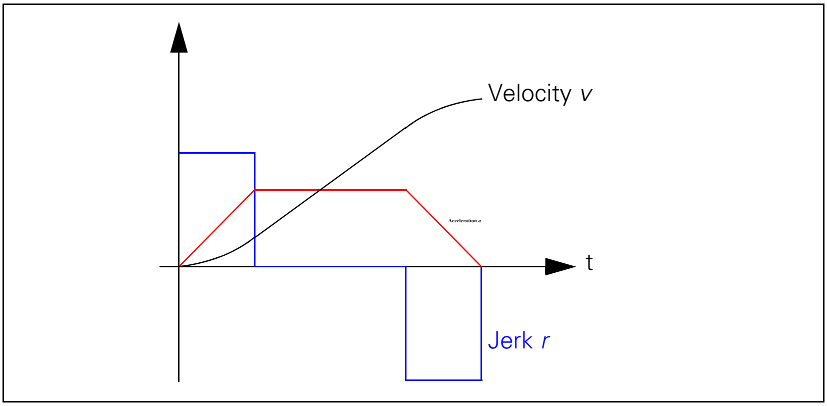
6.8.1 Relation Between Jerk, Acceleration, Velocity and Distance

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}.

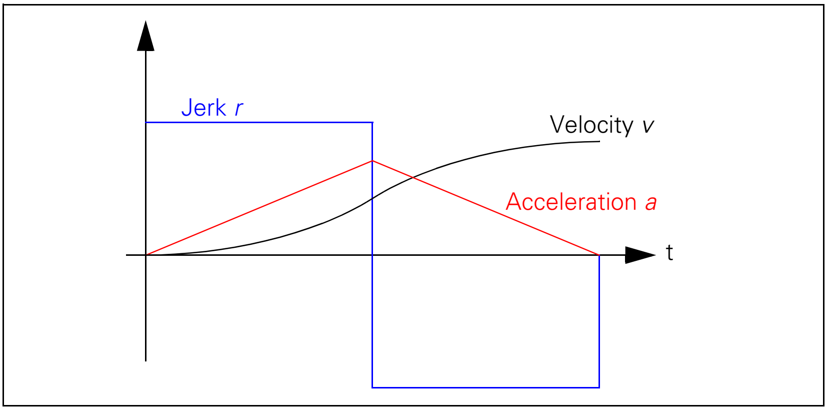
Maximum acceleration

Taking into account the motor and the power module, the machine should be specified in such a way that acceleration during the acceleration phase is as constant as possible. This ensures maximum utilization of the drive current.



The machine, on the other hand, should be designed to fulfill the following 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.

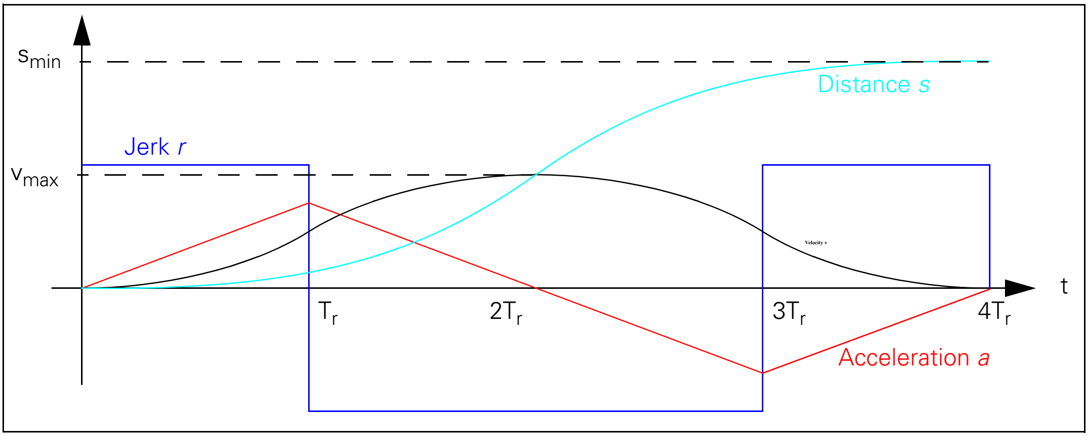
$$a_{max} = \sqrt{v_{max} \cdot r_{max}}$$



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:

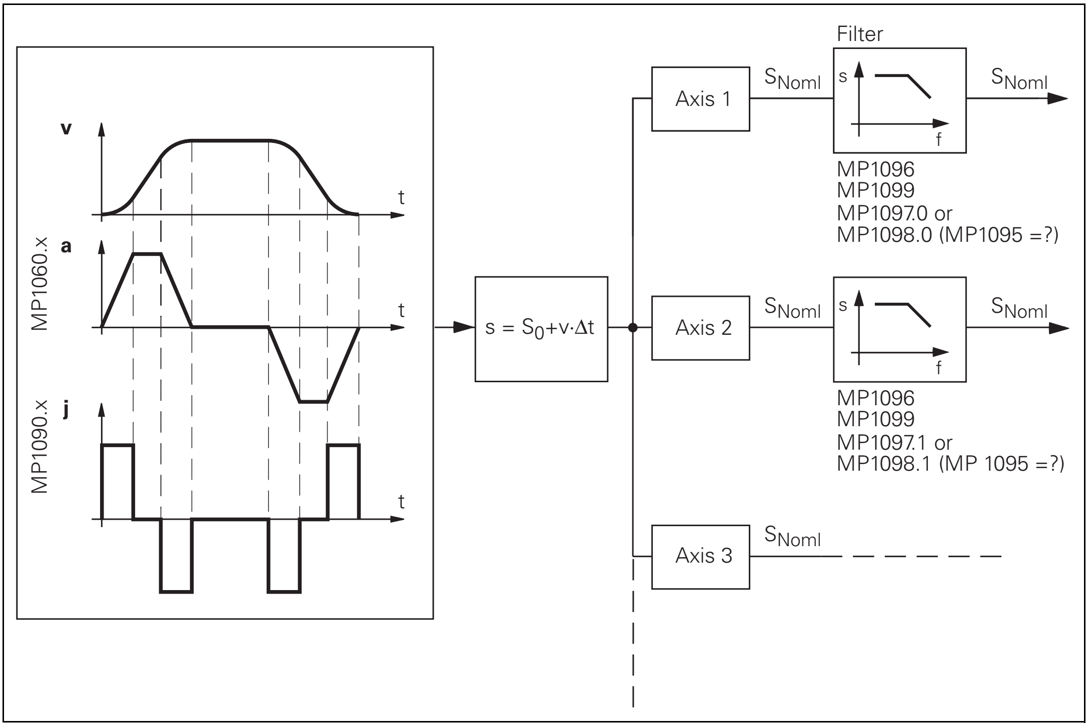
$$s_{min} = 2 \cdot v_{max} \cdot \sqrt{\frac{v_{max}}{r_{max}}} = 2 \cdot 0.5\frac{m}{s} \cdot \sqrt{\frac{0.5\frac{m}{s}}{70\frac{m}{s^3}}} = 0.085\ m = 85\ mm$$

Note

The rectangular jerk curve is rounded through the use of a nominal position value filter (MP1096 ≠ 0). As a result, acceleration is reduced and the minimum distance required for attaining the maximum velocity is increased.

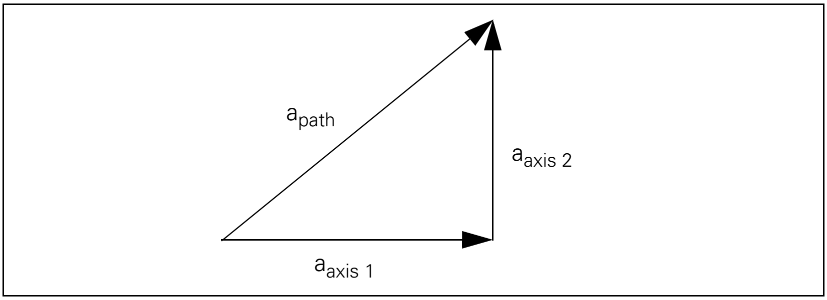
6.8.2 Interpolator

Schematic of the Interpolator:



The interpolator calculates a velocity every 1.8 ms from the programmed feed rate. 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" on page 6 – 131).



If the inverter is not designed for such accelerations, you can limit the path acceleration:

▶ In MP1061, enter 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. The jerk is the rate of change in acceleration. The greater the entered value, the more the system will tend to oscillate.
- In MP1086.x, you limit the jerk for single-axis motions at rapid traverse in the **Program run full sequence, Program run single block** and **Positioning with manual data input** modes of operation.
- Use MP1087.x to limit the axis-specific jerk in **Manual mode.**
- 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:

$$\text{Jerk} \geq \frac{a^2}{v}$$

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 is the jerk on the tool path. The input value is determined by the weakest axis.
- In MP1092, define a machining feed rate beginning at which MP1090.1 becomes effective.

A nominal position value is acquired every 1.8 ms from the calculated velocity. For linear interpolation:

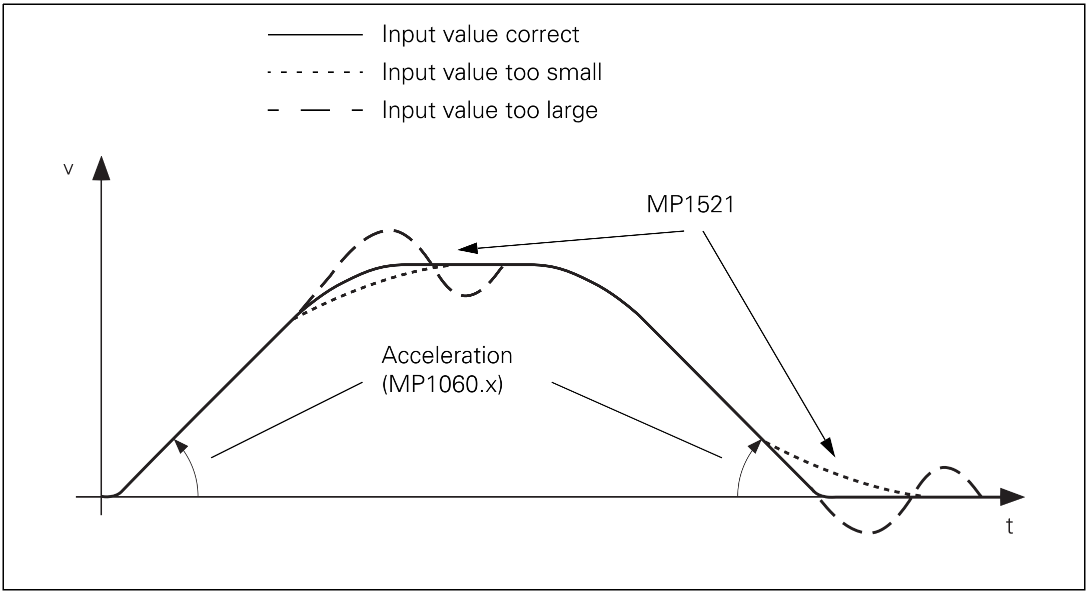
$$s = s_O + v \cdot \Delta t$$

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.



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. Here the iTNC always complies with the tolerance (MP1096, 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, define a tolerance for contour transitions. This tolerance can 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, 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	35	23	17	14	11	8	6	5	5	4	3	3
6	49	32	24	19	16	12	9	7	6	5	4	4
9	58	38	29	23	19	14	11	9	8	7	5	4
12	66	44	33	26	22	16	13	10	9	8	6	5
15	72	48	36	29	24	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	29	19	14	11	9	7	5	4	4	3	3	2
9	35	23	17	14	11	8	7	5	5	4	3	3
12	39	26	19	15	13	10	8	6	5	5	4	3
15	43	29	22	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	24	16	12	9	8	6	5	4	3	3	2	2
6	35	23	17	13	11	8	7	5	5	4	3	3
9	42	28	21	16	14	10	8	7	6	5	4	3
12	48	32	24	19	16	12	9	8	7	6	4	4
15	54	36	26	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	25	16	12	10	8	6	5	4	3	3	2	2
12	29	19	14	11	9	7	5	4	4	3	3	2
15	32	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 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 deceleration path (MP1095.1 = 0).

▶ Test the three filter settings using a test part made of short line segments.

- Single filter
- Double filter
- HSC filter

Note

If you have selected the best nominal position value filter for your application, please note that your input value can be overwritten by the machine user through Cycle 32. If you have switched off the nominal position value filter (MP1096 = 0), the machine user can also switch it on using Cycle 32.

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 filter
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	MP1096 = 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

MP1060.x **Acceleration**
Input: 0.001 to 100.000 [m/s^2 or $1000°/s^2$]

MP1061 **Limitation of the path acceleration**
Input: 0.001 to 100.000 [m/s^2 or $1000°/s^2$]

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
0.1 to 1000.0 [m/s^3 or $1000°/s^3$]

MP1087.x **Maximum permissible axis-specific jerk for Manual mode**
Input: 0.1 to 1000.0 [m/s^3 or $1000°/s^3$]

MP1089.x **Maximum permissible axis-specific jerk for Pass Over Reference Point mode**
Input: 0.1 to 1000.0 [m/s^3 or $1000°/s^3$]

MP1090 **Maximum permissible jerk on the tool path**
Input: 0.1 to 1000.0 [m/s^3 or $1000°/s^3$]
MP1090.0 With machining feed rate
MP1090.1 Beginning with feed rate from MP1092

MP1092 **Feed rate threshold from which MP1090.1 becomes effective**
Input: 10 to 300 000 [mm/min]

MP1094 **HSC filter**
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**
Input:	0: No nominal position value filter 0.001 to 3.000 [mm]: Permissible tolerance at contour transitions
MP1097.x	**Max. permissible axis-specific jerk (single/HSC filter)**
Input:	0.1 to 1000.0 [m/s^3 or $1000°/s^3$]
MP1098.x	**Max. 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)
MP1521	**Transient response during acceleration and deceleration**
Input:	1 to 255 [ms] 0: Function inactive
MP7684	**Nominal position value filter and path control with M128**
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 2 - Nominal position value filter 0: Include the radial acceleration 1: Do not include the radial acceleration Bit 4 - Nominal position value filter 0: Include curvature changes 1: Do not include curvature changes

Feed-rate smoothing

Fluctuations in feed rate sometimes 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

During program run with M128 the head dimensions are also included in the tolerance consideration (MP1096, Cycle 32). This means that the control tries to observe the tolerance, taking the head dimensions into account. As a result, the tolerance is reduced, which leads to a reduction of the feed rate and might cause the rotary axis to jerk.

To switch off the consideration of the head dimensions for rotary axes with M128:

- Enter bit 4 = 1 in MP7682.

MP7682 **Machine parameter with multiple function**
Format: %xxxxx
Input: Bit 4 – Tolerance of rotary axes with M128
0: With consideration of head dimensions
1: Without consideration of head dimensions

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	**Position controller cycle time = MP7600.0 · 0.6 ms**
Input:	1 to 20 Proposed input value: 3 (= 1.8 ms)

MP7600.1	**PLC cycle time = position controller cycle time · MP7600.1**
Input:	1 to 20 Proposed input value: 6 (= 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 POSITIONING WITH MANUAL DATA INPUT, PROGRAM RUN SINGLE BLOCK and PROGRAM RUN FULL SEQUENCE operating modes**
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

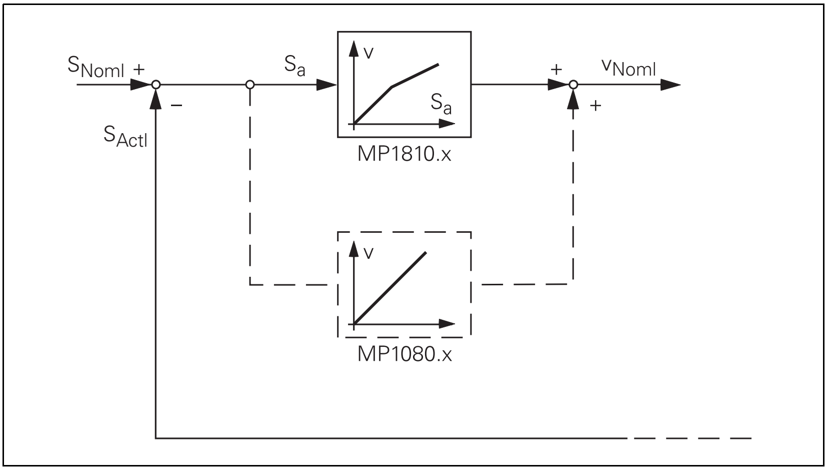
M90 (lag mode: Constant contouring speed at corners) is effective only if operation with following error is selected for all axes (MP1392 = %00000000000000).

MP1391	**Velocity feedforward control in the MANUAL and HANDWHEEL operating modes**
Format:	%xxxxxxxxxxxxxx
Input:	Bits 0 to 13 represent axes 1 to 14 0: Operation with following error (lag) 1: Operation with velocity feedforward control

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, the so-called 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 select too high a k_v factor, the following error is very small; however, this can result in 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.

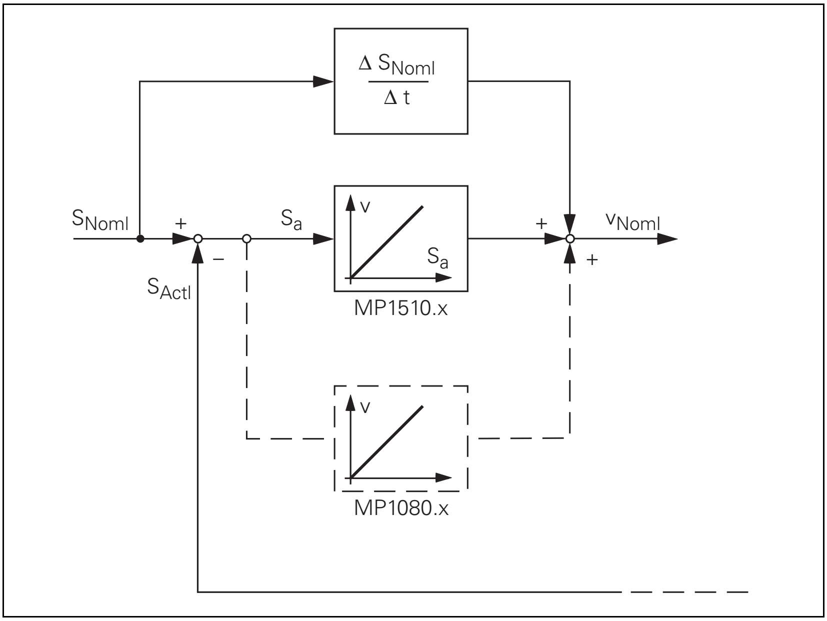
For the **`Positioning with manual data input`**, **`Program run, single block`** and **`Program run, full sequence`** operating modes:

▶ Switch on the velocity feedforward control with MP1392.

For the MANUAL and HANDWHEEL operating modes:

▶ Switch on the velocity feedforward control with MP1391.

Block diagram:

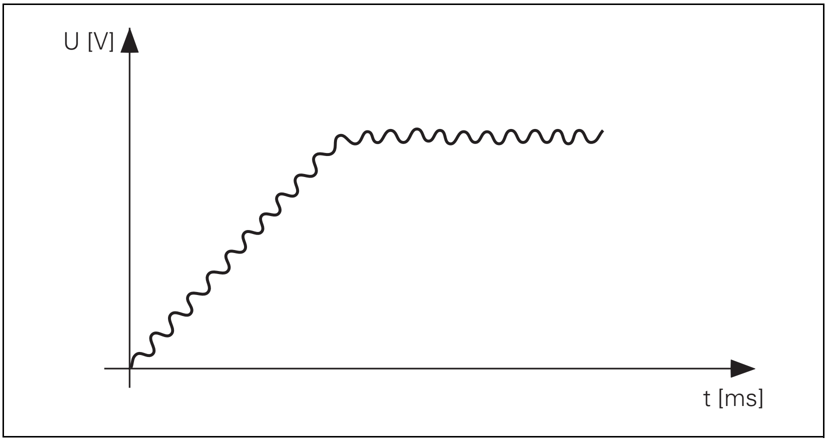


Analog axes:
For stationary axes, the integral factor has an additional effect (MP1080.x). It produces an offset adjustment.

Digital axes:
There is no offset. MP1080.x has no function.

You can influence the control of the forward-fed velocity with the k_v factor:

▶ In MP1510.x, enter a k_v factor.



Warning

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 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 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

To use feedback control with velocity semifeedforward:

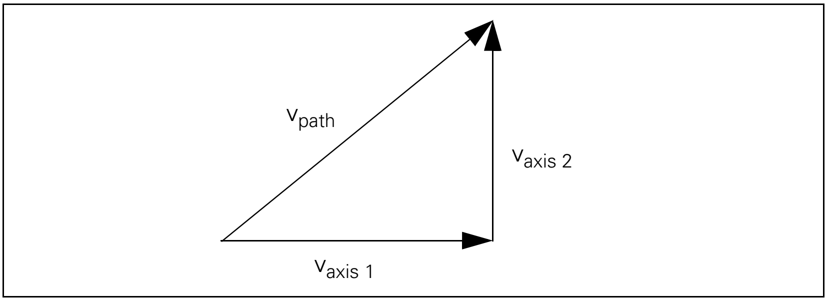
- Activate the velocity feedforward control with MP1391 and/or MP1392.
- Determine the k_v factor for velocity feedforward control (MP1510.x).
- Activate the velocity semifeedforward control by entering the desired factor in MP1396.x.
- Determine the k_v factor for velocity semifeedforward control (MP1516.x).
- 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]

Rapid traverse

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 6 – 114).



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.

You can reduce the rapid traverse on the path through the PLC:

- Enter the reduced value in D596.
  If the value in D596 is larger 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.

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 6 – 8 and see "Encoder Connections" on page 3 – 22.

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 ballscrew.

$$v_{max}\ [\text{mm/min}] = \frac{30\,000}{\text{No. of pole pairs}}\ [\text{1/min}] \cdot \text{ballscrew pitch [mm]}$$

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.

MP1010.x **Rapid traverse**
Input: 10 to 300 000 [mm/min or °/min]

MP1011 **Limit of rapid traverse on the path**
Input: 10 to 300 000 [mm/min or °/min]

MP1020.x **Manual feed**
Input: 10 to 300 000 [mm/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	**Feed rate limit exceeded F MAX**	PLC	PLC
D596	**Max. feed rate from PLC [mm/min]**	NC/PLC	PLC
D360	**Programmed feed rate**	NC	NC
D388	**Current tool 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:

$$\Delta U = \frac{\text{MP1050.x [mV]}}{S_a\text{[µm]}}$$

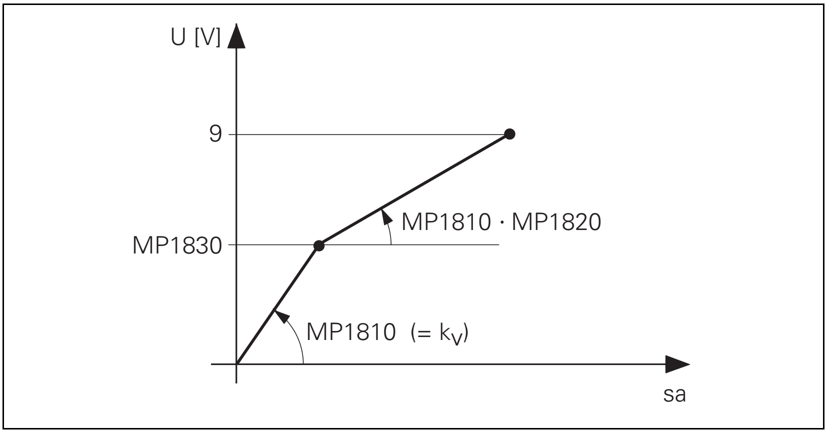
If ΔU is divided by the smallest possible voltage step (0.15 mV), the result is the number *n* of the possible voltage steps per position error.

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):

$$s_a = \left(\frac{\text{MP1830.x [\%]}}{100\text{ [\%]}} + \frac{100\text{ [\%]} - \text{MP1830.x [\%]}}{\text{MP1820.x} \cdot 100\text{ [\%]}}\right) \cdot \frac{v_e}{k_v}$$

MP1820.x **Multiplier for the k_v factor**
Input: 0.001 to 1.000 00

MP1830.x **Characteristic curve kink point**
Input: 0.000 to 100.000 [%]

Opening the 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 release** Bits 0 to 8 represent axes 1 to 9 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:

- 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.

		Set	Reset
W1038	**Preparing opening of the position control loop** Bits 0 to 8 represent axes 1 to 9 0: Not active 1: Active	PLC	PLC
W1040	**Axis-specific opening of the position control loop** Bits 0 to 8 represent axes 1 to 9 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 8 represent axes 1 to 9 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.

Warning

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 8 represent axes 1 to 9 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

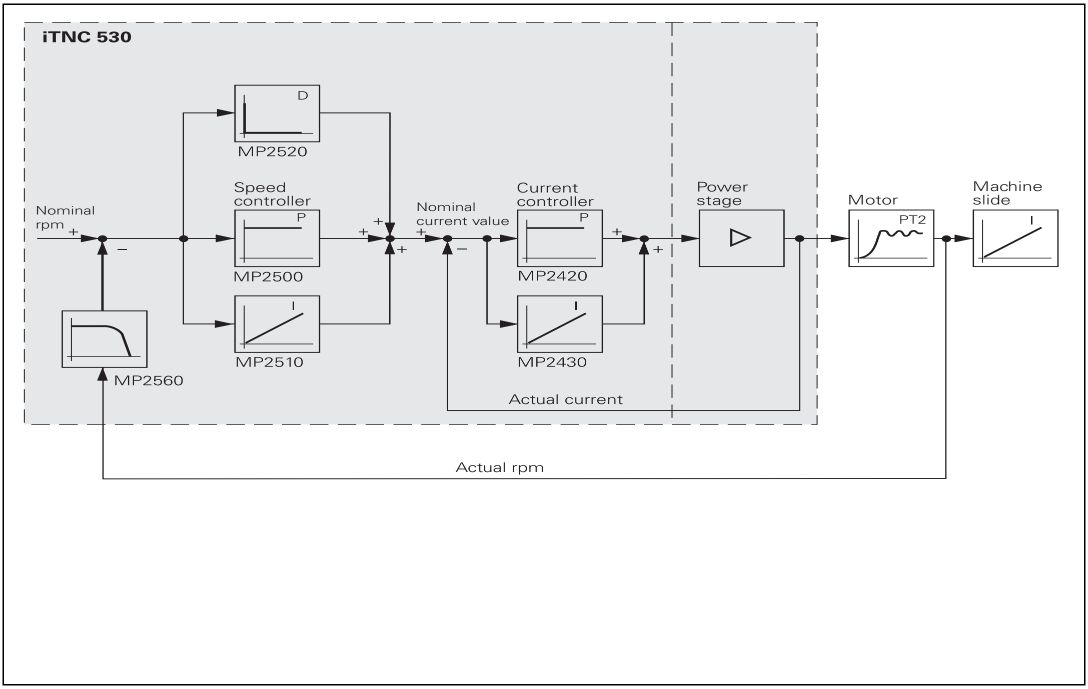
Error detection:

Marker	Value	Meaning
M4203	0	Actual value was assumed as nominal value
	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 job 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.



See "Commissioning" on page 6 – 228.

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 shaft speed controller
Input: 0 to 1 000 000.000 [As]

MP2510.x Integral factor of the shaft speed controller
Input: 0 to 100 000 000 [A]

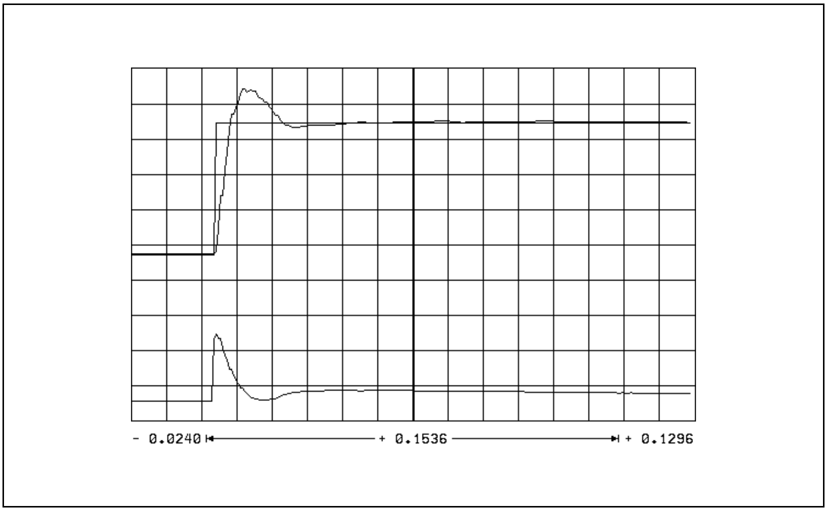
Module 9164 Reading the actual speed value of the motor
The resolution of the actual speed value depends on the encoder being used:

$$\text{Resolution} = \frac{1}{\text{Line count} \cdot 1024} \cdot 100\,000 \text{ [rpm]}$$

Call:
PS B/W/D/K <Axis>
0 to 8: Axes 1 to 9
15: Spindle
CM 9164
PL B/W/D <Actual speed value in the format 0.001 [rpm]>

Error detection:

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:

$$\mathrm{MP2520.x} \approx \frac{T \cdot \mathrm{MP2500.x}}{8}$$

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 shaft speed controller
Input: 0 to 1.0000 [As^2]

Low-pass filter

With the low-pass filter you can damp high-frequency oscillations (> approx. 600 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)
600 Hz to 700 Hz (approx.)	First order (MP2560.x = 1)
> 700 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]

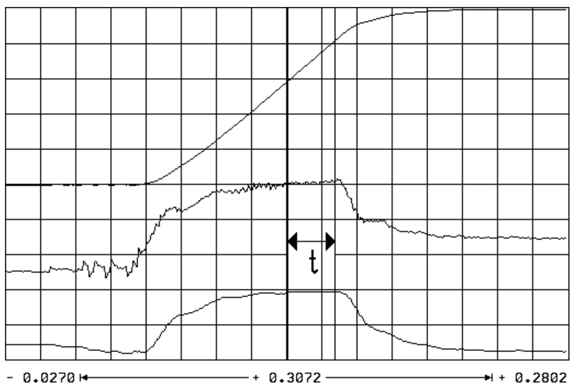
MP2550.x **Band-rejection filter center frequency of the speed controller**
Input: 0.0 to 999.9 [Hz]

Acceleration feedforward

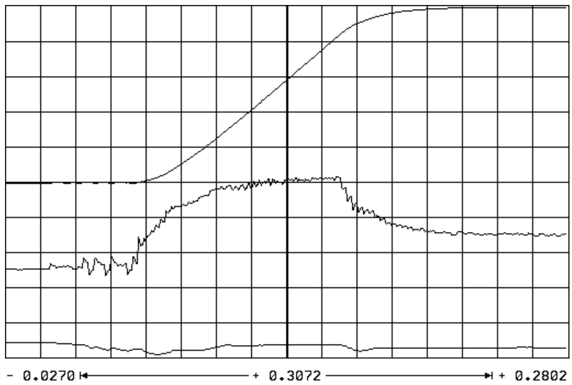
Acceleration feedforward functions only in velocity feedforward control in parallel with the speed controller.

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.x at optimum setting

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.

$$\text{MP2600.x} = \frac{\text{I (INT RPM) [A]} \cdot \text{t [s]} \cdot \text{60 [s/min]} \cdot \text{MP2020.x [mm]}}{\Delta\text{V (ACT RPM) [mm/min]}}$$

I (INT RPM) = integral-action component of the nominal current value

t = acceleration time in which I (INT RPM) remains constant

ΔV (ACT RPM) = actual speed value during change

MP1054.x = traverse distance per motor revolution

MP2600.x **Acceleration feedforward**
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 **Limiting the 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

Minimizing the following error during the jerk phase

An increased following error during the jerk phase can be minimized with MP2606.x. The preceding adjustment of the IPC must have been carried out for this to function.
The typical entry value for MP2606.x is between 0.5 and 2.

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.
- Change MP2606.x until a very small following error occurs in the jerk phase (positive following error: MP2606.x > 1, negative following error: MP2606 < 1)

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 rpm).
- Calculate the holding current from the mean of the measured current values and enter the result in MP2630.x.

Mean:

$$\text{MP2630.x} = \frac{\text{I NOML}_1 + \text{I NOML}_2}{2}$$

Note

If the ready signal (RDY) is missing from the speed encoder inputs of vertical axes, the DSP error message **`8B40 No drive release <axis>`** appears.
A vertical axis is defined with an entry in MP2630.x.

MP2630.x **Holding current**
Input: -30.000 to +30.000 [A]

6.8.5 Enabling the Drive Controller

At X51 to X62 the ready signal is available at pin 10b. As soon as the readiness signal is reset, the drive 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.

The drive controller cannot be switched on if the ready signal of the inverter, the global drive enabling through I32 (X4/33) or the axis group enabling through X150/151 is missing. To switch on the drive controller:

- Determine the functionality of the global drive enabling through I32 (X42/33) with MP2050.
- Assign 24 Vdc to pin 33 (I32) of connection X42 to enable the control.
- 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 = %00000000000111 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 3 – 21). Use Module 9157 to interrogate the status of X150/X151.
- Use W524 to enable the drive for the axis groups.
  The monitoring functions for axis groups 2 and 3 are deactivated for the duration entered in MP1150.1. This prevents the monitoring functions from becoming effective after the drive release has been deactivated.

Note

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 %00000000000000 and in W524 to zero.

- Activate the drive controllers with module 9161. You can use, for example, the axis release W1024 as a criterion for drive enabling.

Note

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.

If you disconnect the voltage for

- X42/33, all drive controllers are switched off.
- X150/151, the drive controllers of the corresponding axis group(s) are switched off.

Note

If drive enabling through X150/151 or X42/33 is missing, the error message **8B40 No drive release <axis>** appears.

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.

You can determine by PLC which axes are switched off in 200 ms:

- Call Module 9159. The drives that are switched off are returned in bit code.

You can request the status of the drive controller with Module 9162, and you can determine if the drive controller is ready to be switched on with Module 9157.

The iTNC monitors the time between the switch-on of the drive hardware and the READY signal (from the PWM cable). If the READY signal is missing after the waiting time has passed, the error message **`8B40 No drive release <axis>`** appears.

- Enter the permissible time in MP2170.

MP2040	**Axis groups for drive enabling through X150/X151**
Format:	%xxxxxxxxxxxxxx
Input:	0: Axis not assigned 1: Axis assigned
MP2040.0-2	Axis group 1 to 3
MP2040.3-7	Reserved, enter %00000000000000

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

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]

		Set	**Reset**
W524	**Drive Enabling for Axis Groups** 0: Inactive 1: Active Bit 0: Axis group 1 Bit 1: Axis group 2 (monitoring functions off) Bit 2: Axis group 3 (monitoring functions off) Bit 3 to bit 15: Nonfunctional	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 <Status information>
0: Drive controller readiness
1: Drive controller status (as in Module 9162)
2: Via X150/X151, axis enabled (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 detection:

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 Drives that are switched off in 200 ms

Call:
CM 9159
PL W/D <Drives, in bit code, that are switched off in 200 ms>

Module 9161 Enabling 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.

Call:
PS W/D/K <Released axes>
Bit: 15 876543210
Axis: S xxxx 987654321
0: No drive controller enabling
1: Drive controller enabling
CM 9161

Error detection:

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 15 0
		Axis Sxxxxxx987654321
		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	

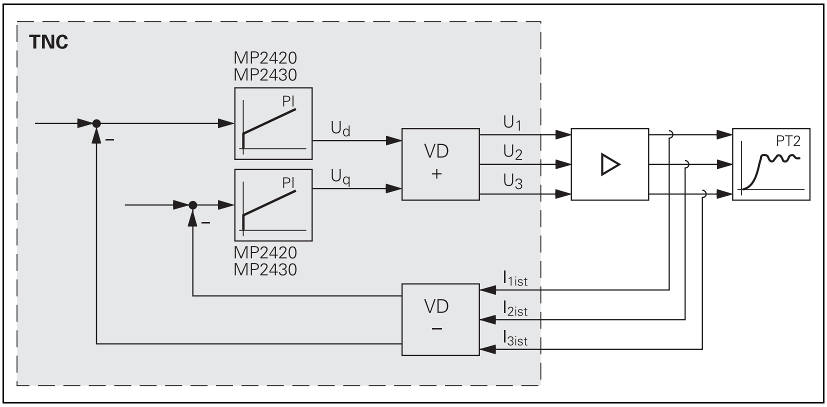
6.8.6 Current Controller

6, 10 or 12 digital current controllers for the axes and spindle(s) are integrated in the iTNC 530.

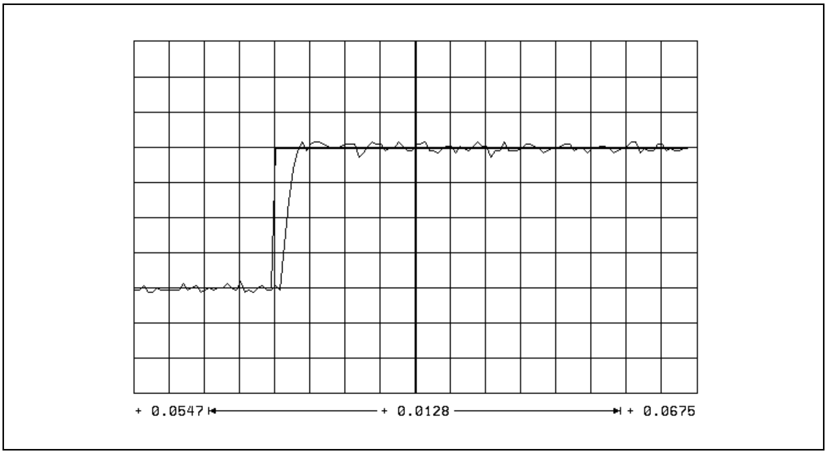
The nominal values for magnetizing current I_{dnom} and torque-producing current I_{qnom} are divided into the PWM signals U_1, U_2 and U_3 through a PI controller and vector rotator VD+, and are transferred to the power module through X51 to X60.

The actual current values I_{1act}, I_{2act} and I_{3act} 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-producing 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.
- In MP2430.x, define the I factor of the current controller.

MP2420.x **P factor of the current controller**
Input: 0 to 9999.99 [V/A]

MP2430.x **I factor of the current controller**
Input: 0 to 9 999 999 [Vs/A]

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.
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.

Preferably the axes should 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. However, an additional braking ramp should be used for axes, and brake power limiting for 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.

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.
Problems with inverters with braking resistors can arise if the drive is switched off too early. The strain on the mechanics is reduced, but can also be influenced with braking strategies.

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.

Note

The value entered in MP2590.x refers to the motor speed, meaning the ballscrew pitch is not considered.

MP2590.x **Braking ramp in an emergency stop**
Input: 0.1 to 999.9 [rpm/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]
(PW 110, PW 210 = 18 Ω, PW 120 = 10 Ω, UP 110 = 9 Ω)
U_Z = dc-link voltage [V]
(UV 130, UE 2xx, UE 2xxB = 565 V; UV 120, UV 140 = 650 V)

Braking the spindle/spindles during a power fail

During a power fail, the "SH1B" signal on X51 to X60 is maintained for 3 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 braking power during a power fail**
Input:	0: No limiting of the braking power in a power failure 0.001 to 3000.000 [kW]

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.

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.

If a non-HEIDENHAIN inverter or a UE 2xx HEIDENHAIN compact inverter is used, the maximum torque current , and as a result the maximum torque, is limited to 70% of the maximum current, since these inverters do not provide any signal for an excessive dc-link current.
The modular HEIDENHAIN inverter system or the UE 2xxB HEIDENHAIN compact inverters do provide such a signal (ERR-IZ). As a result, the maximum torque current is not limited.
This monitoring function must be set for the respective inverter in MP2220.x bit 2.

The torque can be calculated for any speed:

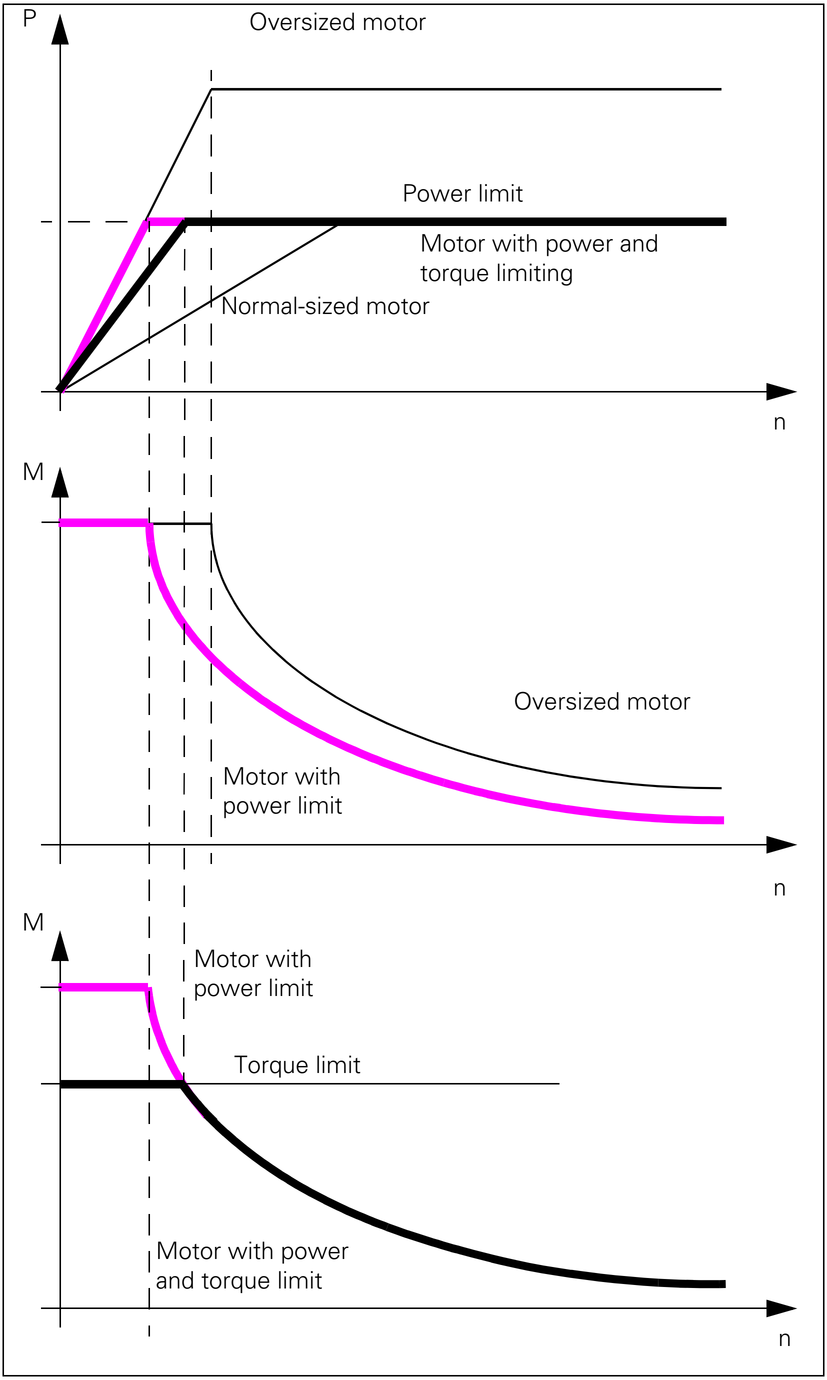
$$M = \frac{P \cdot 60}{n \cdot 2 \cdot \pi}$$

M: Torque [Nm]
P: Power [W]
n: Speed [rpm]

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.
- For the **modular HEIDENHAIN inverter system** or the **UE 2xxB** compact inverter, activate the monitoring of the ERR-IZ signal with MP2220.x bit 2 = 1.
  For **non-HEIDENHAIN inverters** or the **UE 2xx HEIDENHAIN compact inverter**, deactivate the monitoring of the ERR-IZ signal with MP2220.x bit 2 = 0.


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**
Format:	%xxx
Input:	Bit 2 – Monitoring of the ERR-IZ signals (only with HEIDENHAIN inverters except for UE 2xx) 0: Inactive 1: Active
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]

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 the torque limit 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
$I_q = \frac{M \cdot \sqrt{2}}{k_M}$ I_q: Torque-producing current M: Desired torque k_M: Torque constant (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)

Call:
PS B/W/D/K/S<Axis or spindle>
 0 to 8: Axes 1 to 9
 15: Spindle
PS B/W/D/K/S<Torque-producing current in mA>
 –1 = Torque given in motor specifications
CM 9158

Error detection:

Marker	Value	Meaning
M4203	0	Torque preset active
	1	Error code in W1022
W1022	1	0 Nm torque transferred
	2	Invalid axis number
	24	Module was called in a spawn job or submit job

6.8.9 Synchronous Motors in Field Weakening Range

Synchronous motors can also be operated with a weakened field. This is necessary, for example, for high-speed synchronous spindle motors, since the inverter voltage is not sufficient at high speeds.

If the power supply fails, the dc-link voltage increases sharply. As a result, the inverters and possibly the motor might be damaged. A safety feature would be the use of a voltage protection module (e.g. SM 110, see the "Inverter Systems and Motors" Technical Manual).

Warning

A braking resistor, such as PW xxx or UP 110, does not offer sufficient protection.

For synchronous motors, the operation with a weakened field is automatically activated if MP2160.x = 1 or 2 has been entered. You do not need to make any further settings.

- Enter MP2160.x = 0 if you want to operate synchronous motors **without** a weakened field.
- Enter MP2160.x = 1 if you want to operate synchronous motors with a weakened field **and** use voltage protection modules.

The EcoDyn synchronous motors from HEIDENHAIN are operated with a 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 voltage protection module 1: Voltage protection module present 2: Limited field weakening without voltage protection module for EcoDyn motors

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.

With the integral factor you can adjust an offset automatically:

- Enter an integral factor in MP1080.x. The speed with which the offset is eliminated depends on the size of the factor.
- Play in the drives can result in instability in the control loop.
  In this case, enter the factor zero.

MP1080 is effective only at a standstill.

MP1080.x	**Analog axes: Integral factor for offset adjustment**
Input:	Input 0 to 65 535
	Digital axes: nonfunctional
	Input: 0

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 · 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 100.000 [m/s^2 or $1000°/s^2$]

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":

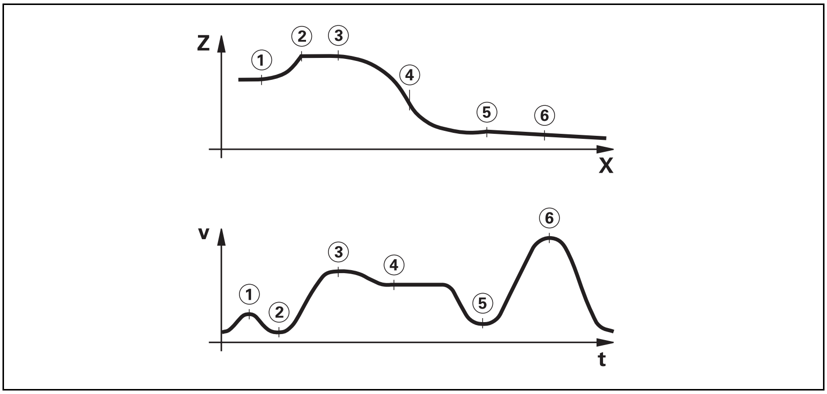
- Define the permissible tolerance for contour transitions in MP1096. The larger the tolerance, the greater the tool velocity.

The user can overwrite this tolerance with Cycle 32 "Tolerance."

Jerk limitation (see "Interpolator" on page 6 – 114) 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. A spline also has the advantage of reducing the jerk:

- Enter MP7680 bit 10 = 1.



MP1096	**Tolerance for contour transitions**
Input:	0: No nominal position value filter 0.001 to 3.000 [mm]: Permissible tolerance at contour transitions

MP7680	**Machine parameter with multiple function**
Format:	%xxxxxxxxxxxxxx
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 User's Manual).

If you program M112 or M124, defined arcs will be inserted at the corners regardless of the feed rate (see 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. But it takes more computing time than an inserted arc.

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**
Format:	%xxxxxxxxxxxxx
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

Warning

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.

Note

The inputs of the PL 4xx B 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 255 [no. of the PLC input]
The inputs of the PL 4xx B 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.

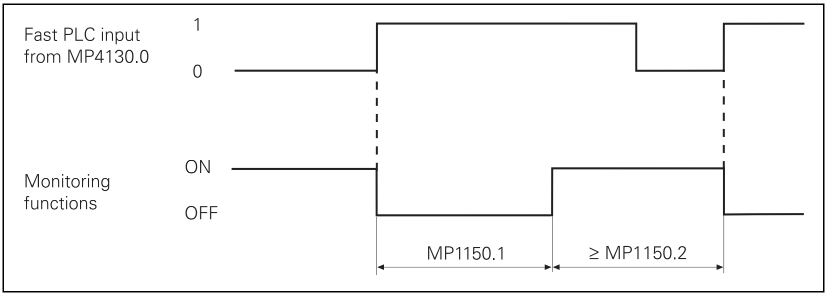
If the first limit is exceeded, the error message **EXCESSIVE SERVO LAG IN <AXIS>** appears. The machine stops.
You can clear this message with the CE key. An actual-to-nominal value transfer is then executed for the respective axes.

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.

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 [°]

6.11.2 Nominal Speed Value Monitoring

For the axes, the nominal speed value monitoring is effective only in operation with velocity feedforward.

For the spindle, it is effective in operation with following error as long as the position control loop is closed (orientation).

If the nominal speed value calculated by the position controller is greater than the maximum possible nominal value, the blinking error message **NOMINAL SPEED VALUE TOO HIGH <AXIS>** appears and the control-is-ready output is reset.

Analog axes: Maximum nominal value = 10 V

Analog spindle: Maximum nominal value = 20 V

Digital axes and spindle: Maximum nominal value = maximum motor speed from motor table

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 (See page 6 – 6).
- 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.

Warning

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 at which the movement monitoring goes into effect**	
Input:	Analog axes:	0.030 to 10.000 [V]
	Digital axes:	0.030 to 10.000 [1000 rpm] Recommended: 0.030 [1000 rpm]

MP1054.x **Linear distance of one motor revolution**
Input: Analog axes: Nonfunctional
Digital axes: 0 to 100.000 [mm] or [°]

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.

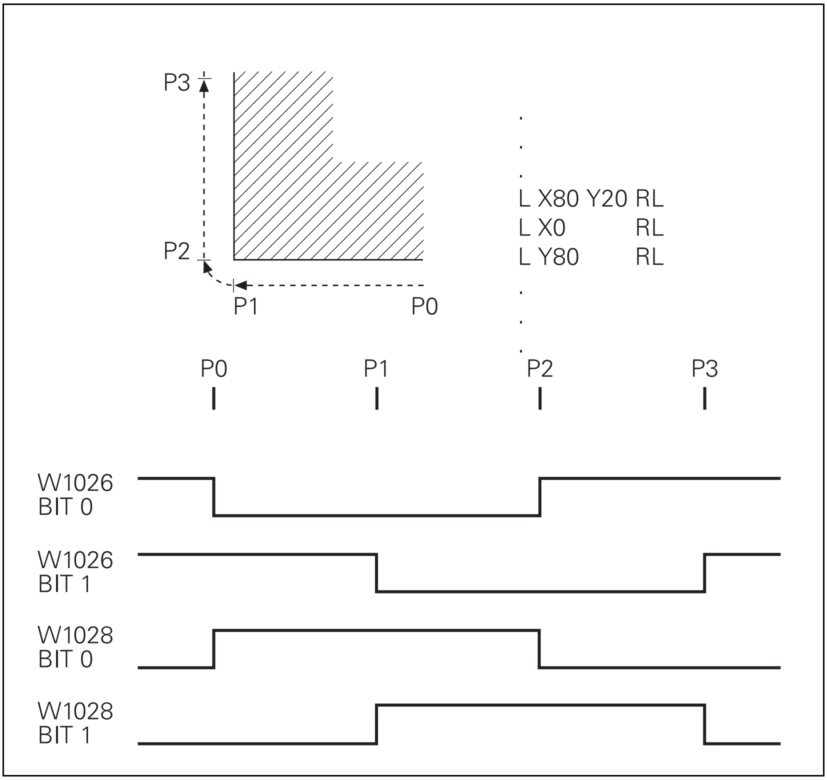
On contours that can be machined with constant surface speed, W1026 is not set.

		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 in motion** 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 Vdc (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 Vdc (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 Vdc (UV 120, UV 140, UV 150, UR 2xx, UV 105: approx. 200 V), the control is reset (signal RES.PS).
- Below approx. 135 Vdc (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 the following HEIDENHAIN power supply units provide the AC-fail signal:

- UV 120
- UV 140
- UV 150
- UR 2xx

For all other HEIDENHAIN components, the AC-fail signal must not be selected 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 displays the error message **`POWERFAIL.`** 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: Neither powerfail nor AC fail
3: Powerfail

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 detection:

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 422

The internal temperature of the MC 422 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 machine is switched on again and the temperature does not go below 60 °C, the error message is reactivated after about 10 to 20 seconds.

The temperature of the MC 422 can be found with Module 9133.

Module 9133 Temperature of the MC422

Call:

PS	B/W/D/K	<Code> 0: Internal temperature of the MC 422 1: Temperature of the CPU
CM	9133	
PL	B/W/D	<Temperature in °C>

Error detection:

Marker	Value	Meaning
M4203	0	Internal temperature was read
	1	Error code in W1022
W1022	2	Invalid code

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 Sampling 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 8 and 15 = Axes 1 to 9 and the spindle
CM	9165	
PL	B/W/D	<Temperature> Range: 0 to 255 °C

Error detection:

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 13.

If the permissible temperature of the heat sink on the power module is exceeded, this signal is reset.

Warning

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.

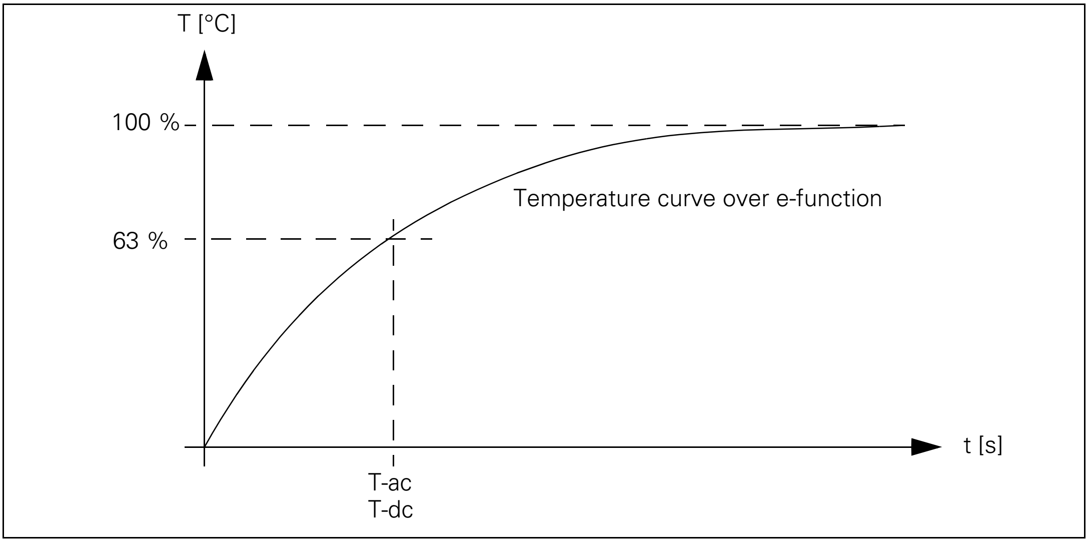
The temperature warning signal is not evaluated in the NC:

- Use Module 9160 or 9066 to interrogate the temperature warning, and take appropriate measures.

6.11.8 I^2t Monitoring

The instantaneous motor current is limited to either the maximum current of the power module, or the maximum motor current, whichever is lower. The values result from the type of power supply unit and type of motor, and are saved in the motor or power-supply-unit table.

In addition, a separate I^2t monitoring is performed for the power module and the motor:
The temperature increase of the motor and power stage is proportional to the square of the current output. Since the heat dissipation can be uneven if the motor is stationary or moving slowly, the monitor distinguishes between two ranges. This is the purpose of the F-ac entry (**`Cutoff frequency for T-ac [Hz]`**) in the motor-and-power-module table. Above this frequency, the T-ac entry (**`Therm. time constant for ac [s]`**) applies, and below it the T-dc entry (**`Therm. time constant for dc [s]`**) applies. The T-ac and T-dc entries identify the point in the temperature curve at which 63% of the maximum temperature is reached. This defines a temperature model of the motor or power stage.



With the aid of this temperature model, a mean current value is permanently calculated. If this calculated mean current value exceeds the rated current (for motors, plus MP2302.x and additionally for power modules MP2304.x), the I^2t monitoring (Module 9160) responds. In this case, you should reduce the machining feed rate in the PLC program. If the calculated mean current value is more than 1.1-fold of the rated current, (for motors, plus MP2302.x) an error message appears; the drives are not switched off.

- In MP2302.x, enter a reference value for I^2t monitoring of the motor. The input value is a factor of the rated current of the motor (1 = rated current of the motor). If you enter zero, the I^2t monitoring for the motor (not for the power supply unit) is switched off.
- In MP2304.x, enter a reference value for I^2t monitoring of the power module. The input value is a factor of the rated current of the power module (1 = rated current of the power module). If you enter zero, the I^2t monitoring for the power module (not for the motor) is switched off.

Note

In the oscilloscope you can display the current value of the I^2t monitoring of the motor and power stage, as well as the current load of the drive.

If the value 0 is entered in the motor or power module table in the F-DC, T-DC, F-AC and T-AC columns, the following default values apply:

- Axis drives:
  - F-DC = 0
  - T-DC = 10
  - F-AC = 0
  - T-AC = 10
- Spindle drives:
  - F-DC = 0
  - T-DC = 150
  - F-AC = 0
  - T-AC = 150

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

MP2304.x Reference value 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 as reference value

Module 9160 Status request for temperature monitoring and I^2t monitoring

Call:

```
CM   9160
PL   D      <Temperature monitoring>
            Bit      15     876543210
            Axis:    Sxxxxx987654321
PL   D      <I2t monitoring>
            Bit      15     876543210
            Axis     Sxxxxx987654321
```

Error detection:

Marker	Value	Meaning
M4203	0	No error
	1	Control has no current controller

6.11.9 Read Actual Utilization of Drive Motors

Module 9166 provides the momentary utilization of the given drive motor as a percentage value.

Utilization means:

Speed range	$\mathbf{n}_{act} < \mathbf{n}_N$	$\mathbf{n}_{act} \geq \mathbf{n}_N$
Asynchronous motor	$\frac{\lvert M \rvert}{\lvert M_{rated} \rvert}$	$\frac{\lvert P \rvert}{\lvert P_{rated} \rvert}$
Synchronous motor	$\frac{\lvert M \rvert}{\lvert M_{rated} \rvert}$	–

Instead of the drive torque, one uses the effective component I_q of the current, which is proportional to the torque.

I_{qMean} is formed as mean value of the individual current values I_{qx} of the last 20 ms:

$$I_{qMean} = \frac{\sum(I_{q1}..I_{qn})}{n}$$

$$\text{Utilization} = 100\ \% \cdot \frac{I_{qMean}}{I_{qRated}}$$

For asynchronous motors:

$$I_{qRated} = \sqrt{I_N^2 - I_{mag}^2}$$

I_N: Motor rated current
I_{mag}: Magnetizing current

For synchronous motors:

I_{qRated} = <Motor rated current>

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 8 and 15 = Axes 1 to 9 and the spindle
CM	9166	
PL	B/W/D	<Utilization of the drive in %>

Error detection:

Marker	**Value**	**Meaning**
M4203	0	No error
	1	Control has no current controller

MP2312.x **Reference value for utilization of motors**
Input: 0 to 1000.000 [· rated current of motor]
0 or 1: Reference value is rated current of motor

6.11.10 Status of HEIDENHAIN Inverters

Status information of the HEIDENHAIN inverters can be read with Module 9066.

Module 9066 Status of HEIDENHAIN inverter

Call:

PS	B/W/D/K	<Code> 0: HEIDENHAIN inverter
CM	9066	
PL	B/W/D	<Status information> Bit 0: Nonfunctional Bit 1: DC-link voltage too high ($\overline{\text{ERR.UZ.GR}}$) Bit 2: Heat sink temperature too high ($\overline{\text{ERR.TEMP}}$) Bit 3: Short-circuit of a motor phase with U_Z ($\overline{\text{AXISFAULT}}$) Bit 4: DC-link current too high ($\overline{\text{ERR.IZ.GR}}$) Bit 5: Power supply unit not ready (RDY.PS) Bit 6: Leakage current too high ($\overline{\text{ERR.ILEAK}}$)

Error detection:

Marker	Value	Meaning
M4203	0	Status has been read
	1	Error code in W1022
W1022	2	Invalid code
	24	Module was called in a spawn job 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.

MP2195	**Suppress error messages of the HEIDENHAIN supply units**
Format:	%xxxxxxxx 0: Error message is not suppressed 1: Error message is suppressed
Input:	Bit 0: Reserved Bit 1: $\overline{\text{ERR.UZ.GR}}$ signal Bit 2: $\overline{\text{ERR.TMP}}$ signal Bit 3: Reserved Bit 4: $\overline{\text{ERR.IZ.GR}}$ signal Bit 5: $\overline{\text{RDY.PS}}$ signal Bit 6: $\overline{\text{ERR.ILEAK}}$ signal Bit 7: Reserved

6.11.11 Controlling the Motor Brakes

The motor brakes are controlled with the $\overline{\text{BRK}}$ braking signal, which reaches the HEIDENHAIN inverters via the PWM outputs (X51 to X62). The corresponding outputs are activated there. See the basic circuit diagrams. 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).

Activated brakes cause a change in the controlled system. The motor with the changed controlled system is controlled during the overlap time. When the controller is switched off, the NC software can cause vibrations. These vibrations are suppressed. MP2220 bit 3 can be used to not suppress the vibrations. HEIDENHAIN does not recommend switching off the suppression of the vibrations.

MP2220	**Monitoring functions**
Format:	%xxxx
Input:	Bit 3 – Switching off the controller when the motor brakes are activated 0: Suppress vibrations 1: Vibrations are allowed
MP2308.x	**Time between output of the braking signal $\overline{\text{BRK}}$ and switching off of the controller (overlap time)**
Input:	0.001 to 0.500 [s] 0: 0.200 s

6.11.12 EMERGENCY STOP monitoring

On the control there is a PLC input (X42/4) and a PLC output (X41/34) with the designation control-is-ready for the EMERGENCY STOP routine.

If a functional error is detected, the iTNC switches the control-is-ready output off. A blinking error messages appears and the PLC program is stopped. You **cannot** clear this error message with CE:

- Correct the error and restart the switch-on routine.

If the "control-is-ready signal acknowledgment" input is switched off by a process external to the control, the error message `EXTERNAL EMERGENCY STOP` appears. The NC sets M4177 and M4178. The nominal speed value 0 is output and the drives are switched off. You can clear this error message with CE after switching the machine control voltage back on.

The "control-is-ready signal acknowledgment" input is passed directly onto the NC; it can **not** be manipulated by the PLC.

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	**Erasable error message is 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

Testing an 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

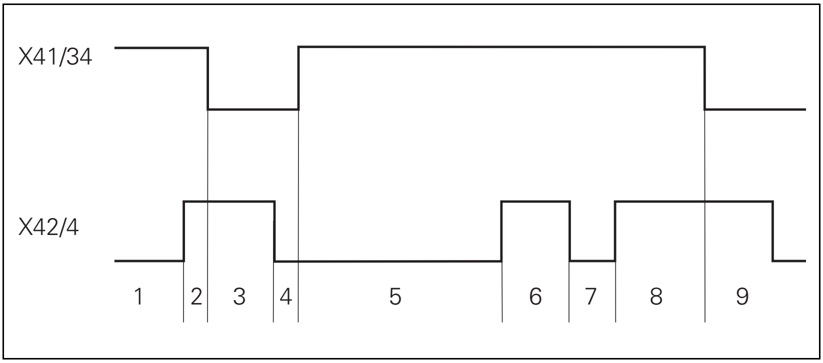
In the event of an error, the control-is-ready output must trigger an emergency stop. The control therefore checks this output every time that line power is switched on.

Note

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

Ensure that the control-is-ready acknowledgment occurs within 1 second.

Flowcharts



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-is-ready output and acknowledgment are high.	
7	Control voltage is switched off externally.	**EMERGENCY STOP**
8	After switching on again, the machine control voltage can be switched off, and then the control operates normally.	
9	After detecting a fault, the control switches off the control-is-ready output (X41/34).	Blinking error message

6.12 Spindle

Two spindles can be controlled alternately. See page 6 – 215.
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: Spindle jog
Third: M3/M4
Lowest: 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 servo-controlled spindle for oriented spindle stop
7: Same as 4, but with servo-controlled spindle for oriented spindle stop
8: Same as 5, but with servo-controlled spindle for oriented spindle stop

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 rotary encoder to be used.
$1\text{-}V_{PP}$ 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.

▶ Evaluate the reference mark with Module 9220, see "Renewed traversing of the reference marks" on page 6 – 105.

If MP3143 = 2, then the reference pulse release for the spindle position encoder is set with X30, pin 1. Ensure that the same reference signal is always evaluated.

If MP3143 = 1 or 3, then X30 pin 1 is evaluated as the reference signal. The reference mark of the position encoder is not evaluated. In this case the reference signal **must** be evaluated with Module 9220 (See page 6 – 105).

Warning

Due to its low accuracy, this solution is not recommended.

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 9999 [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.
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 following error and distance to go are considered to be zero.

Call:
PS B/W/D/K <Target address Cxxxx>
CM 9042

Error detection:

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 speeds 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.

Warning

For axes, monitoring of the rotational direction (MP2220 Bit1) 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 input for the first spindle
MP113.1	Speed encoder input for the second spindle

MP2220.x	**Monitoring functions**
Input:	%xxx 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 rpm.

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 rpm cannot be represented in words W320 and W322.
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.

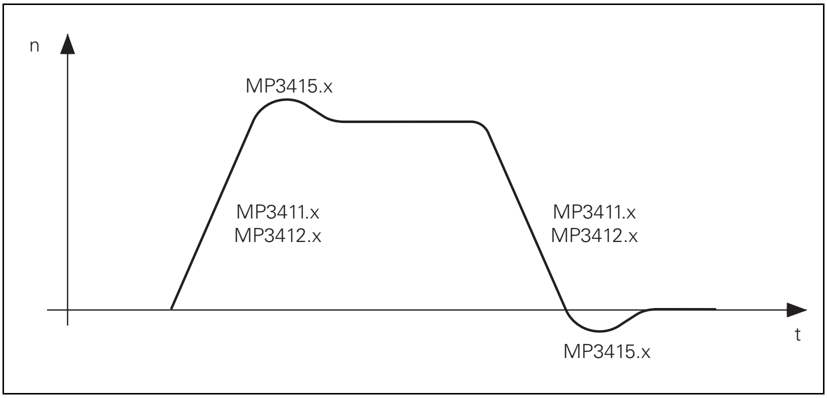
Digital spindles:
The nominal speed value is transferred to the internal speed controller.

		Set	Reset
M4003	**Nominal speed value output analog or digital (MP3010 = 3 to 8)**	NC	NC
D356	**Programmed speed [0.001 rpm]**	NC	NC
D364	**Nominal speed value [rpm]**	NC	NC
W320	**Nominal speed value [rpm]**	NC	NC
D368	**Actual speed value [rpm]**	NC	NC
W322	**Actual speed value [rpm]**	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 rpm/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

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 [rpm]

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 rpm]

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 rpm]

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.

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. This may be necessary, for example, for horizontal/vertical spindles. The programmed speed is saved by the NC in D356 and D756:

- 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.

A changing nominal speed value 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 rpm]

		Set	Reset
W256	**Gear code**	NC/PLC	NC/PLC
D356	**Programmed speed [0.001 rpm]**	NC	NC
D756	**Programmed speed or speed from PLC [0.001 rpm]**	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.
  As soon as a new value is entered here, it is assumed by the NC.

The spindle override functions either in 1% steps or according to a nonlinear characteristic curve:

- With MP7620, bit 3, 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%

MP3310.0-1 Limit for spindle override
Input: 0 to 150 [%]
MP3310.0 Upper limit
MP3310.1 Lower limit

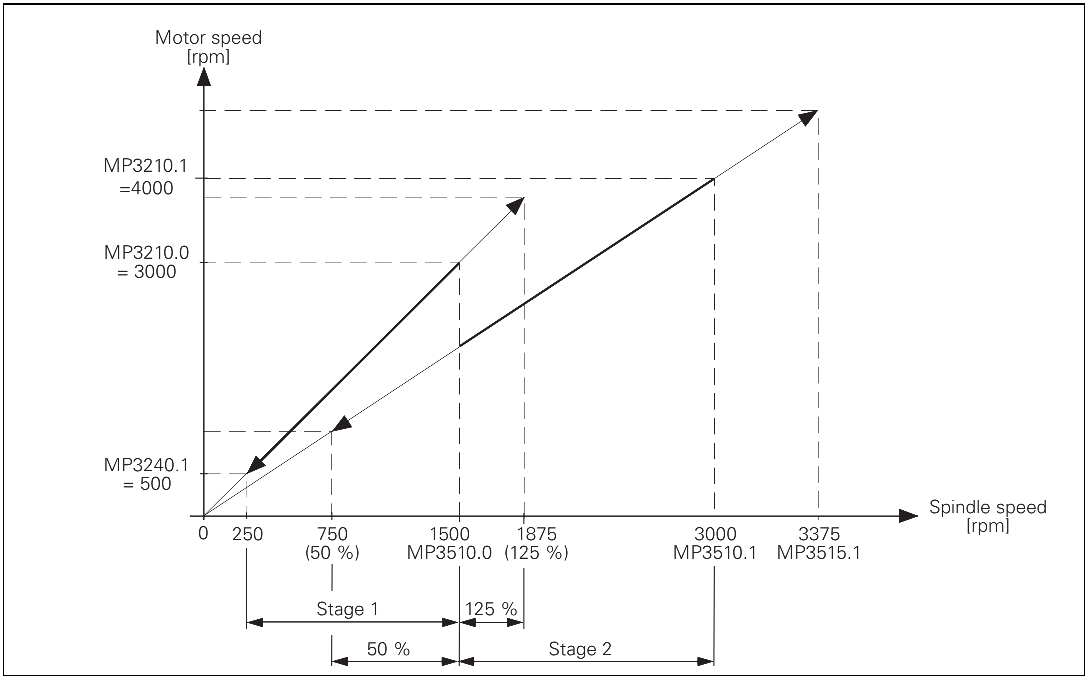
MP3515.0-7 Maximum spindle speed for gear ranges 1 to 8
Input: 0 to 99 999.999 [rpm]

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

		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 rpm with motor 3000 rpm (MP3210.0 = 3000; MP3510.0 = 1500)
- Gear range II: Spindle 3000 rpm with motor 4000 rpm (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 rpm (MP3515.1 = 3375)
- Minimum motor speed: 500 rpm (MP3240.1 = 500)



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 rpm are possible:

▶ Specify in MP3020 the speed range and the speed increment. The S code for the minimum speed is saved in W1008.

Example:
Minimum speed = 1 rpm (S code 20)
Maximum speed = 1000 rpm (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

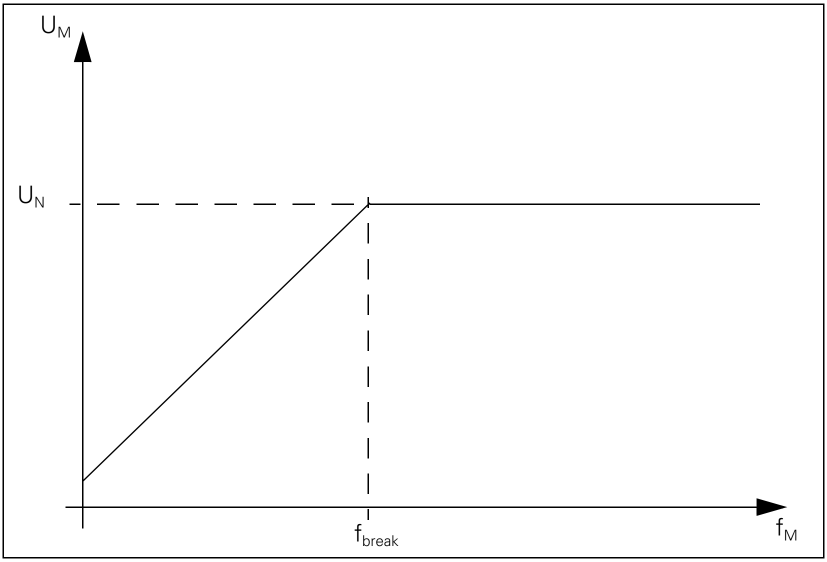
S code	rpm
S 00	0
S 01	0.112
S 02	0.125
S 03	0.14
S 04	0.16
S 05	0.18
S 06	0.2
S 07	0.224
S 08	0.25
S 09	0.28
S 10	0.315
S 11	0.355
S 12	0.4
S 13	0.45
S 14	0.5
S 15	0.56
S 16	0.63
S 17	0.71
S 18	0.8
S 19	0.9
S 20	1
S 21	1.12
S 22	1.25
S 23	1.4
S 24	1.6
S 25	1.8
S 26	2
S 27	2.24
S 28	2.5
S 29	2.8
S 30	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	rpm
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	rpm
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 with a U/f component:

- 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 machining 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. After the marker is set, the spindle is not feedback controlled any longer.

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 when the reference mark is 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 [rpm]

		Set	Reset
M4000	**Spindle in position**	NC	NC
M4012	**Opening the spindle control loop**	PLC	PLC
M4015	**Renewed evaluation of the spindle reference mark**	PLC	NC
M4017	**Spindle moving in feedback control**	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.

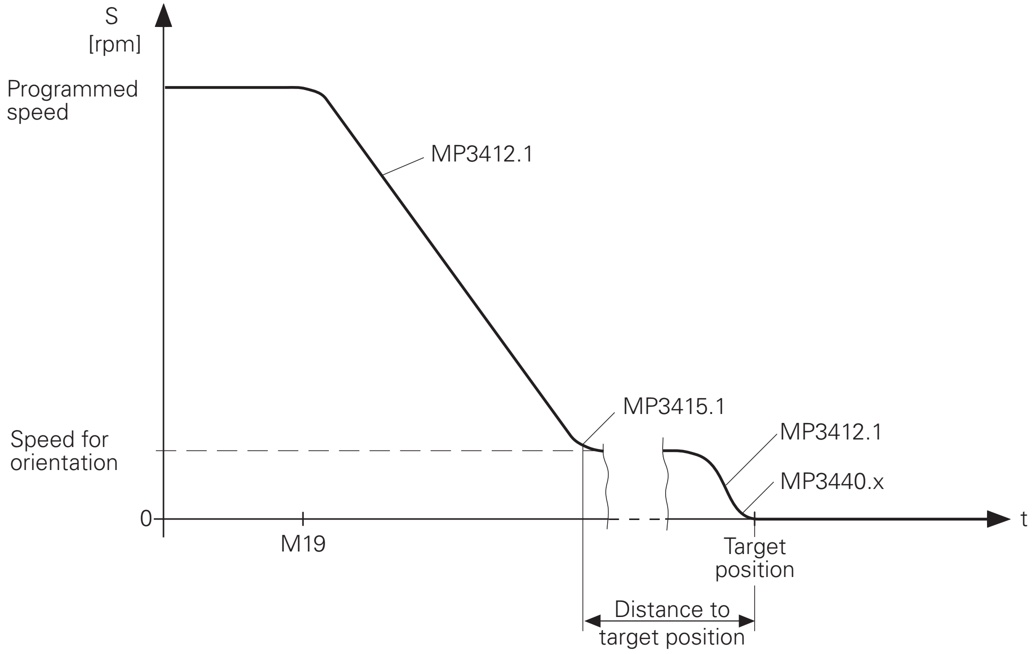
Call:

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
CM	9171	

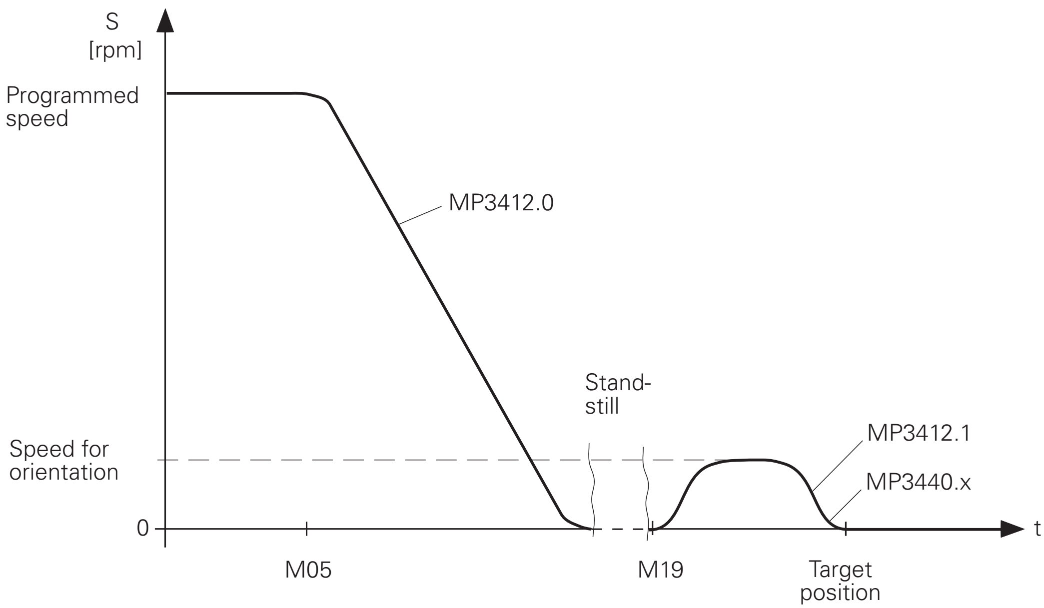
Error detection:

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

Orienting a moving spindle



Orienting 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 [rpm]

		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 rpm
MP3411.x = 0.025 [1000 rpm/ms]

$$\frac{1000\ \text{rpm}}{0.025 \cdot [\text{rpm/ms}]} = 40\ \text{ms}$$

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:

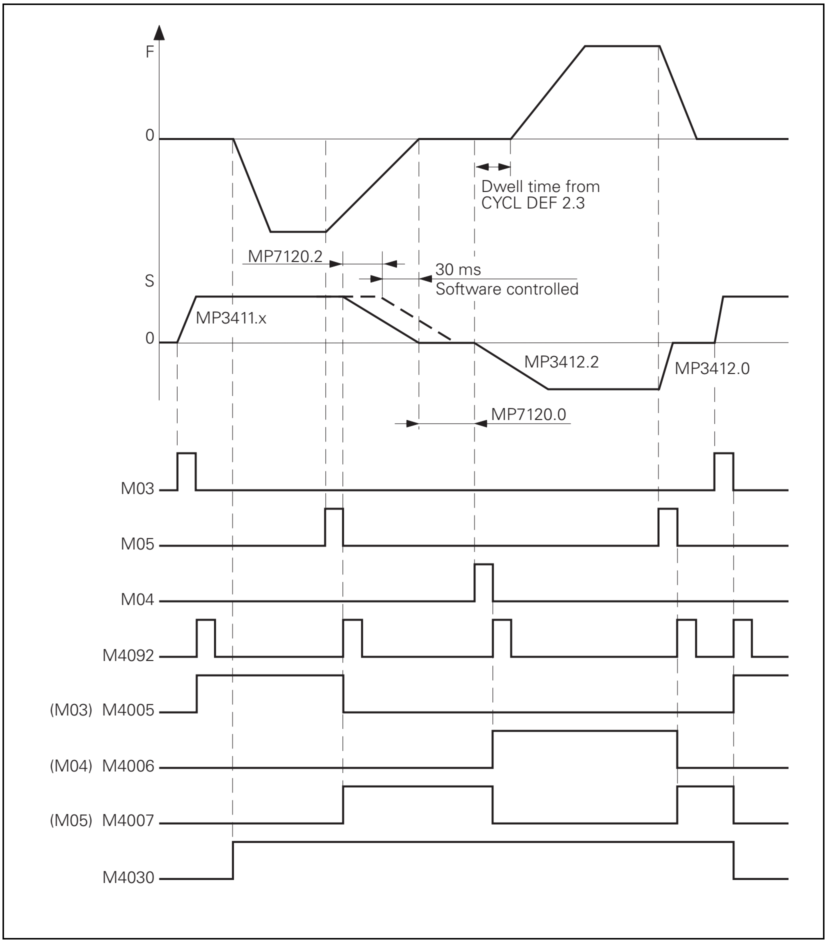
- Change the programmed dwell time in the cycle.

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]

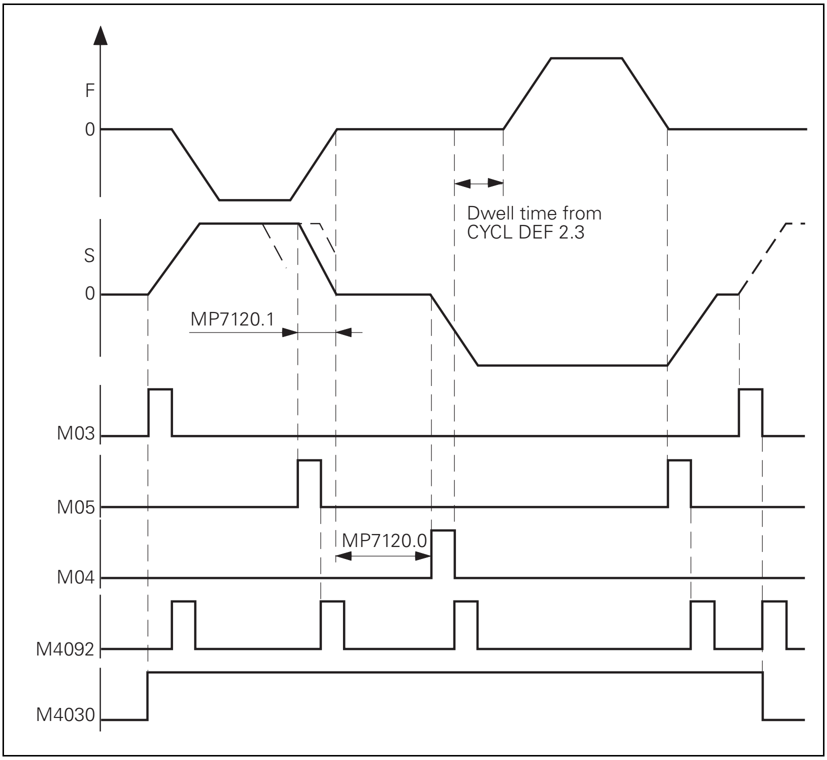
6.12.8 Tapping with Floating Tap Holder and Coded Spindle-Speed Output

If the spindle speed is output in code, the spindle and feed-rate ramps cannot be synchronized:

- Enter the advanced switching time of the spindle in MP7120.1.

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 switches automatically 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 6 – 111 and „Spindle" on page 6 – 187.

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.

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:

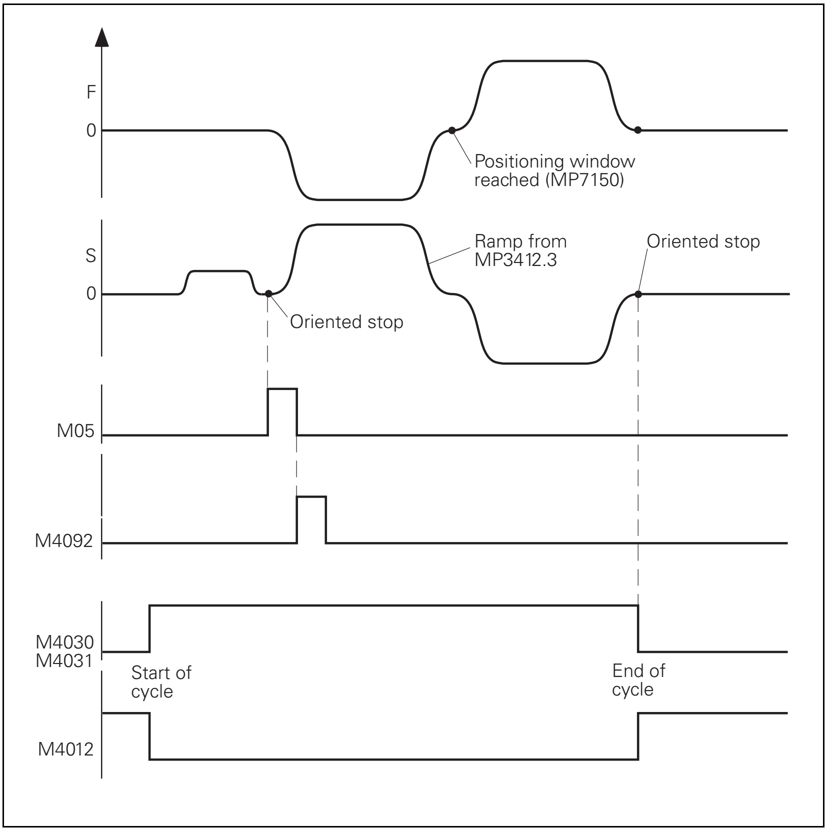
- In MP7160, set bit 3 = 1 to switch off acceleration feedforward control for Cycle 17.

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). See also „Oriented Spindle Stop" on page 6 – 203.

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.

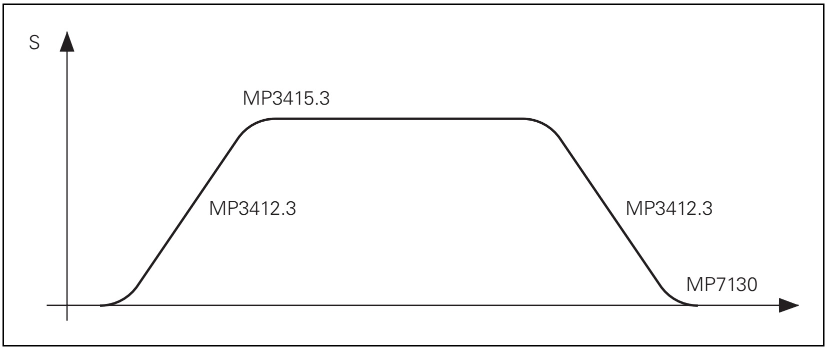


While Cycle 17 is running, the positioning window from MP7150 applies to the tool axis:

- Enter a value in MP7150 smaller than or equal 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 Cycle 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

		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 blocks with the same name can be saved in the motor table. This may be necessary if

- Another parameter block 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.

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 block 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 detection:

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

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 6 – 13.

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.

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, see "Spindle" on page 6 – 187.
- 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 Spindle switchover

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 is not set).

Call:

PS	B/W/D/K	<Spindle number> 0: First spindle 1: Second spindle
CM	9175	

Error detection:

Marker	Value	Meaning
M4203	0	Specified spindle active
	1	Error code in W1022
W1022	2	Invalid spindle number
	6	M4157 = 1 (RESTORE POSITION active)
	20	Module was called in a spawn job or submit job
	21	Missing strobe in M4176 = 1

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.

Assignment of encoder inputs and speed command outputs to the axis and spindle:

- In MP110.x, enter the position encoder input of the axis (if present).
- In MP111.x, enter 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).
- In MP113.x, enter 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.
- Begin 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 detection:

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 spindle or the axis.

Call:
PS B/W/D/K <Axes bit-encoded>
CM 9155

Error detection:

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 job 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 spindle or the axis.

Call:
PS B/W/D/K <Axes bit-encoded>
CM 9156

Error detection:

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 job 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 [µm]
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 isn't included.
Feed rate	Contouring feed rate [mm/min]
Position: I1	Signal 1 of the position encoder
Position: I2	Signal 2 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]
I nominal	Nominal current value [A] that determines torque
PLC	The PLC operands (B, W, D, I, O, T, C) are recorded. Enter the operands in the input field next to the PLC.
a nominal	Nominal value of the acceleration [m/s^2]
r nominal	Nominal value of the jerk [m/s^3]
Pos. Diff.	Difference between position and speed encoder [mm]
a actual	Current acceleration value [m/s^2]. Calculated from position encoder.
r actual	Current jerk value [m/s^3]. Calculated from position encoder.
I2-t (mot.)	Current value of the I^2t monitoring of the motor [%]
I2-t (p.m.)	Current value of the I^2t monitoring of the power module [%]
Utilization	Utilization of drive motors [%]
Block no.	Block number of the NC program
Gantry diff.	Difference between synchronous axes [mm]
U nominal	Nominal voltage [V]

Signal	Meaning
P mech.	Mechanical power [W]
P elec.	Electrical power [W]
M actual	Actual torque value [Nm]
S noml (f.)	Nominal position of nominal position value filter [mm]

The oscilloscope provides additional functions for commissioning the current controller, see "Commissioning" on page 6 – 228.

The recorded data remain stored until you start recording again or activate another graphic function.

Colors

- In MP7365.x, define the colors for the oscilloscope.

Setup

▶ Activate the oscilloscope with the code number 688379.

After you enter the code number, the setup menu appears:

▶ Choose the parameters to be entered with the cursor keys.

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 opened.

Feed rate:

▶ Enter the height of the step for the nominal velocity value (in mm/min). This entry has no effect for ramp output.

Sample time:

▶ Set the time interval for recording the signals.
Entry: 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

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. If you press the STOP soft key, the last 4096 events are stored.
  - **`Single shot:`** If you press the START soft key, the next 4096 events are stored.
  - **`Channel 1`** to **`6:`** Recording begins when the triggering threshold of the selected channel is exceeded.

Trigger threshold:

- Enter the trigger threshold (you will find the appropriate units in the signals table on Page 6–221):

Slope:

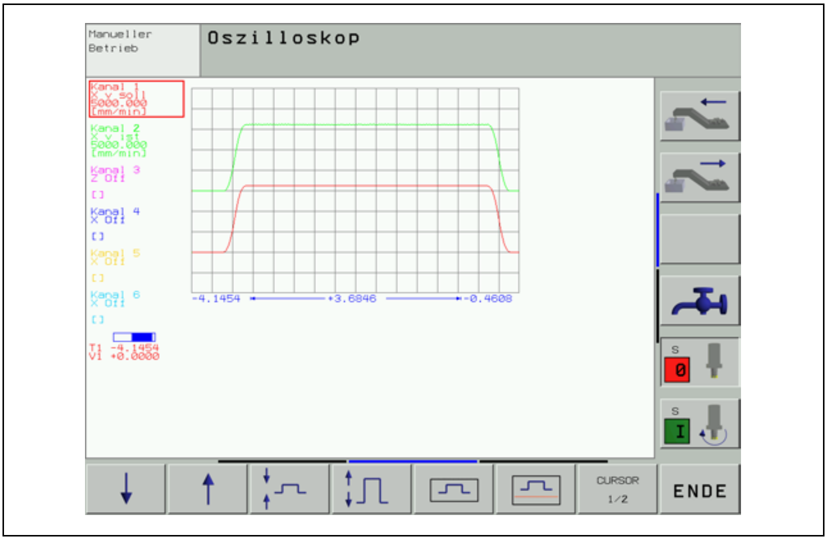
- Select whether the rising edge (positive slope) or falling edge (negative slope) of the signal acts as trigger.

Pre-Trigger:
Recording begins at a time preceding the trigger time point by the value entered here

- Enter a value.

Oscilloscope display:

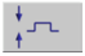
- Press the OSCI Soft key



During recording, the selected signals are continuously displayed. After recording ends, the memory contents are displayed. For every channel, the type of signal and the resolution are also shown. 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 status field shows the amplitude of the selected channel and the time with respect to the beginning of recording.
- 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. With this function you can measure the acceleration time of an axis, for example.

Meaning of the soft keys:

Meaning of the soft keys:	
SET UP	Back to setup menu.
START	Start recording. The recording is ended either with a 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 picture.
	Optimum vertical resolution. The signal is referenced to the datum line.
CURSOR 1/2	Switch to second cursor.
	Move the signal to the left.
	Move the signal to the right.
	Decrease the horizontal resolution.
	Increase the horizontal resolution.
INVERT	Invert the signal.
ENDE	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 called by the PLCdesign PLC development software again, or
- 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.

6.14 Commissioning

6.14.1 Table for Power Module, Supply Module and Motor

In the machine parameter editor you select the installed power modules and the motors:

- Call a list of power modules or motors 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.	
	ANFANG ↑	Jump to the beginning of the list of power modules.
	ENDE ↓	Jump to the end of the list of power modules.
	SEITE ↓	Scroll one page forward in the list of power modules.
	SEITE ↑	Scroll one page backward in the list of power modules.
	ACHSEN WAHLEN	Select a power module with the arrow keys and transfer it with the corresponding STORE MP2100.X soft key.
	AKTUELLEN WERT	Open the table of power modules and jump to the selected power module.
	APPEND AMPLIFIER	Add the new power module to the table of power modules.
	ENDE	Return to the machine parameter editor.

Meaning of the soft keys:		
	Call a list of motors.	
		Jump to the beginning of the list of motors.
		Jump to the end of the list of motors.
		Scroll one page forward in the list of motors.
		Scroll one page backward in the list of motors.
		Select a motor with the arrow keys and transfer it with the corresponding "STORE MP2200.X" soft key.
		Open the table of motors and jump to the selected motor.
		Add the new motor to the table of motors.
		Return to the machine parameter editor.

After you have selected the motor and the power module, the models are automatically entered in MP2100.x and MP2200.x.

If you use motors or power 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 motor models or power modules, the changed tables are filed in the PLC partition:

- PLC:\MP\MOTOR.MOT (motor table)
- PLC:\MP\MOTOR.AMP (power-module table)

These tables are then taken into account by the iTNC. If at any time you want to use the HEIDENHAIN standard tables again, you must erase the above-mentioned tables in the PLC 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 operating memory.

Note

The original entry from the motor table is used only when MP2202.x = *, MP2204.x = * and MP2206.x = *.

MP2100.x **Power module model**
Input: Name of the selected power module (entered by the iTNC)

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: Incremental linear encoder
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 (not aligned)
8: Incremental linear encoder with distance-coded reference marks (not aligned)

Entries in the power module table

- 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-10 000) in A

Entries in the power supply module table

- Designation of power supply module (NAME)
- Type of power supply module (E/R)
  - 0 = Nonregenerative
  - 1 = Regenerative
- Rated power output (P-N) in W
- Peak power for the duty cycle S6-40% (P-S6-40) in W
- Peak power for 0.2 s (P-MAX02) in W
- DC-link voltage (UZ) in V
- Analog value of the dc-link voltage with HEIDENHAIN power supply modules (UZ-AN) in V/V
- Analog value of the dc-link current with HEIDENHAIN power supply modules (IZ-AN) in V/A
- Status signals of the HEIDENHAIN power supply modules
  - Bit 0: Signal $\overline{\text{PF.PS.AC}}$ (AC fail)
  - Bit 1: Signal $\overline{\text{PF.PS.ZK}}$ (power fail)
  - Bit 2: Signals $\overline{\text{ERR.TEMP}}$ (temperature)
  - Bit 3: Signal RDY.PS (ready)
  - Bits 4 to 7: reserved

Entries in the motor table

- 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 rpm
- 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 rpm
- Threshold speed for field weakening (N-FS) in rpm
- Maximum speed (N-MAX) in rpm
- 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
  - 7 = Incremental rotary encoder with distance-coded reference marks (not aligned[1])
  - 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
- 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

The inductance of the series reactor is calculated as follows:

$$L = \frac{700\ \mu H \cdot 5000\ Hz}{f_{PWM}} - \frac{(X_1 + X_2) \cdot 1000}{2 \cdot \pi \cdot f_N}$$

- L: Inductance of the series reactor in µH
- f_{PWM}: PWM frequency [Hz]
- X_1: Stator leakage reactance [mΩ]
- X_2: Rotor leakage reactance [mΩ]
- f_N: Rated frequency [Hz]

A negative result means that there is no series reactor.

Note

If a series reactor is installed later, the current controller must be readjusted.

1. See “Field Orientation” on page 6 – 237

6.14.2 PWM Frequency

With MP2180.x, certain controller groups can be assigned different PWM frequencies.

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.

Warning

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. Nr. 359 651-02
- CC 422/10 control loops with Id. Nr. 359 652-02
- CC 422/12 control loops with Id. Nr. 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 speed, but it means that the axis and the spindle(s) must be commissioned again.

For the commissioning of new machines, HEIDENHAIN recommends adjusting the PWM frequency to fit axis modules (normally 4 kHz, see 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 list of power modules.

  $$I_{Nnew} = I_N \cdot (100\ \% - X_R[\%])$$

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

If a synchronous spindle is used along with an encoder without Z1 track or a nonaligned encoder with EnDat interface, there is no assignment between the encoder and rotor magnets.
With the FIELD ORIENTATION function, which must be run once during commissioning, the iTNC 530 automatically determines the assignment between the encoder and the rotor magnets (field angle) and saves this information 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>**

Encoder with EnDat interface	Encoder without Z1 track
As soon as the absolute position of the encoder has been read, the assignment between absolute position and field angle is determined from the file.	After the drive has been switched on, the spindle is oriented automatically. Following that, the drive is ready for operation. As soon as the reference mark is traversed during the first movement of the spindle, the assignment of the field angle is determined from the file.

- 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.** 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 spindle rotates at the rated speed for approx. 2 s. During this period the field angle at the reference mark or datum is determined and automatically saved in a file on the hard disk.

Note

Standstill monitoring is active while the field orientation function is being carried out. If it responds during field orientation, increase the threshold in MP1110.x. After field orientation has finished, reset MP1110.x to the original value.

- Press the END soft key.

The control carries out a reset. Then the assignment of the field angle is available.

If an encoder with EnDat interface is used, the field angle is assigned to the zero position of the encoder.
If an encoder without Z1 track is used, the spindle is first roughly oriented after it has been started. Then the field angle can be assigned to the reference mark and the spindle starts, taking the field angle into account.

Note

If the speed encoder is exchanged, the field orientation function must be rerun.

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 6 – 183.
- 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:

- In MP2190, enter the dc-link voltage at the power module.

MP2190 **DC-link voltage U_Z**
Input: 0 to 10 000 [V]
HEIDENHAIN inverters:
UE 2xx, UE 2xxB, UV 130: 565 V
UR 2xx, UV 120, UV 140, UV 150: 650 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	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
MP1140.x	0.03	Movement monitoring (for digital axes the minimum value is entered)
MP1144.x	0	Motion monitor for position and speed

MP	Temporary input value	Meaning
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
MP2400.x	0.1	Gain for current controllers
MP2500.x	0.5	Proportional factor of the shaft 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	Limiting the integral factor of the speed controller
MP2520.x	0	Differential factor of the shaft speed controller
MP2530.x	0	PT_2 element of the speed controller
MP2540.x	0	Band-rejection filter damping
MP2550.x	0	Band-rejection filter for center frequency
MP2600.x	0	Acceleration feedforward
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 motor speed
MP2612.x	0	Delay of the 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 6 – 244 and see "Commissioning the Digital Spindle" on page 6 – 273).

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 to switch to the **Oscilloscope** 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.
  - Clamp the axes.
- In the **Programming and Editing** mode of operation, use the MOD key to enter the code number 688379 to switch to the **Oscilloscope** mode of operation.
- 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.
- 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 the value 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.
- 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.
- 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`**. 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 axis to be adjusted.
- With the I factor / P factor 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:

$$\text{Starting value} = \frac{100\,000 \cdot L}{T_a}$$

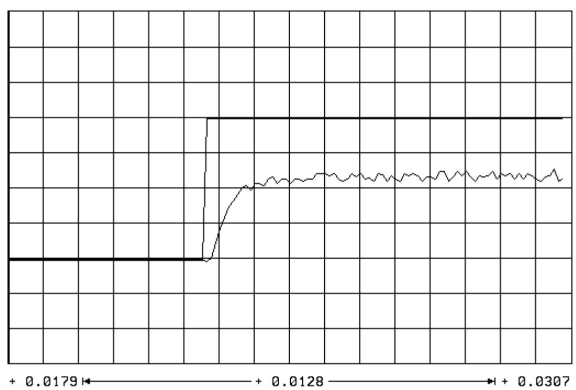
T_a	**f_{PWM} (MP2180.x)**
150	3 333 Hz
120	4 166 Hz
100	5 000 Hz
75	6 666 Hz
60	8 333 Hz
50	10 000 Hz

Synchronous motor: $L = \dfrac{XH}{2 \cdot \pi \cdot (F\text{-}N) \cdot 1000}$

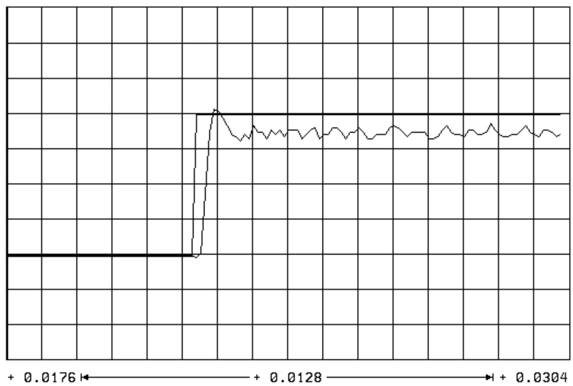
Asynchronous motor: $L = \dfrac{XStr1 + XStr2}{2 \cdot \pi \cdot (F\text{-}N) \cdot 1000}$

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.

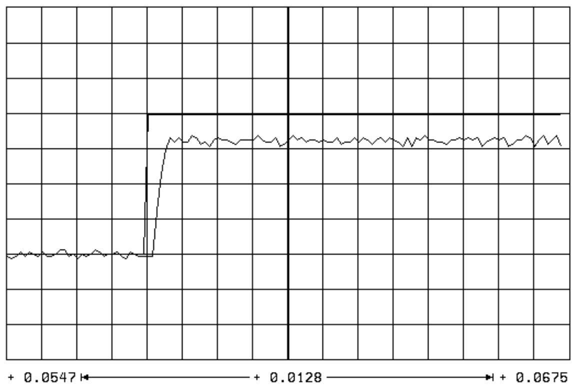
- Set this P factor (MP2420.x) with the ↑ soft key.
- Press the START STEP soft key.
  This sends a step function to the current controller and measures the step response. The height and length of the step function are automatically calculated by the iTNC.
- With the ↑ soft key, increase the P factor (MP2420.x) step by step until just barely no undershoot is visible.



MP2420.x too small



MP2420.x too large

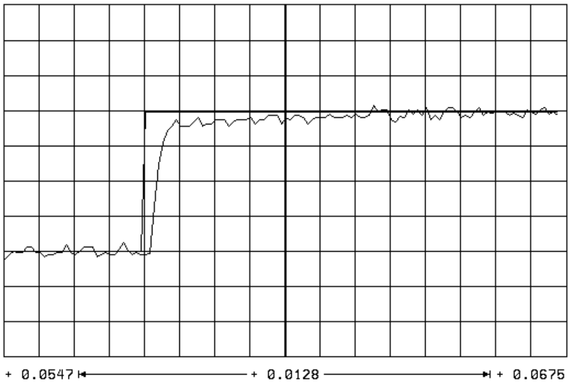


MP2420.x optimum

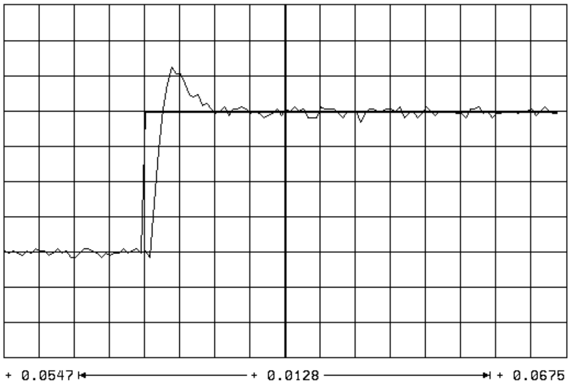
- Save this value with the STORE MP2420.x soft key.
- With the I factor / P factor soft key, select the I factor.

▶ With the ↑ soft key, increase the I factor (MP2430.x) step by step until

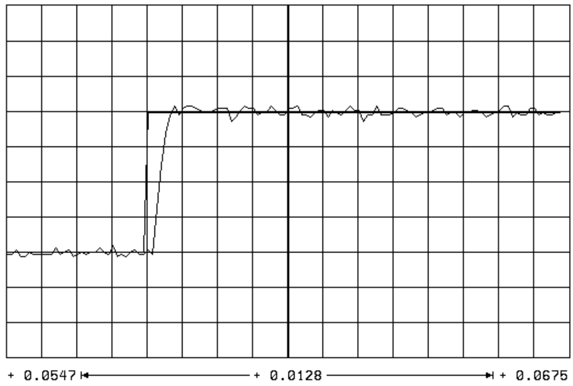
- Spindle: Just barely no overshoot is visible, and so that the nominal value is reached as quickly as possible (short rise time).
- Axes: You see an overshoot but no undershoot.



MP2430.x too small



Axes: MP2430.x optimum



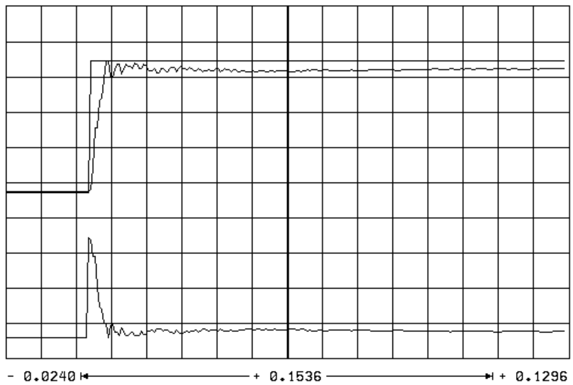
Spindle: MP2430.x optimum

- Save this value with the STORE MP2430.x soft key.
- Press the END key to exit the **Commission Current Controller** mode.

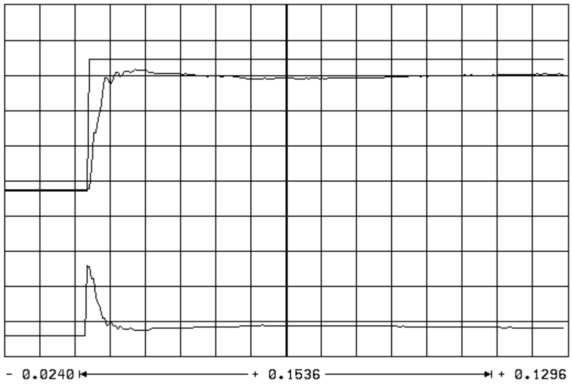
Speed Controller

Adjusting the speed controller:

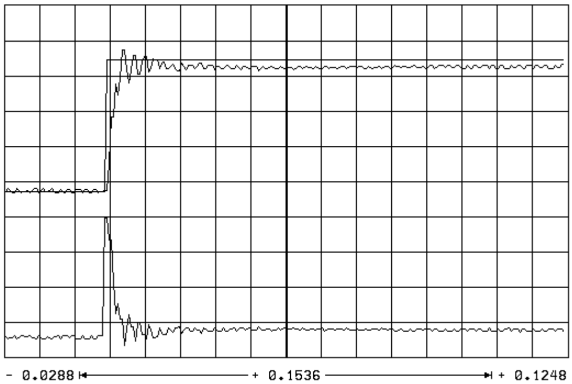
- Deselect “Pass over reference point” by setting MP1340 = 0.
- Ensure that the loaded PLC program fulfills the following conditions:
  - Position control loop is opened (W1038/W1040), because the NC opens the position control loop only during the step function. If the position controller is not optimized, 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) up to the oscillation limit.



MP2500.x at the oscillation limit



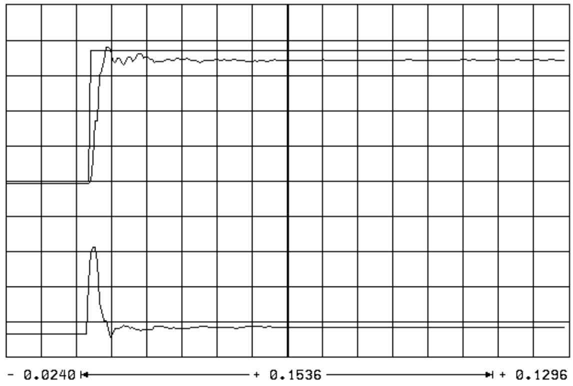
MP2500.x too small



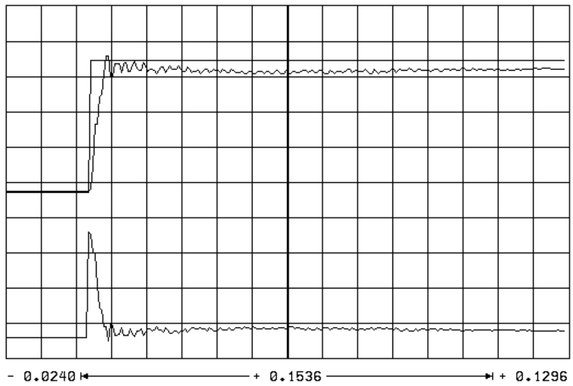
MP2500.x too large

▶ Input value for MP2500.x = <determined value> · 0.6

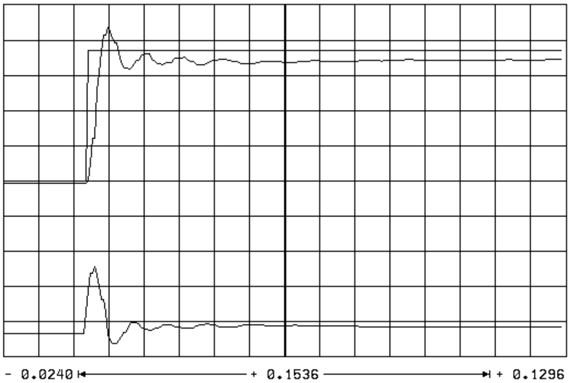
▶ Compensate high-frequency interference oscillations (> 400 Hz) with MP2530.x or MP2560.x.



MP2530.x at optimum setting



MP2530.x too small



MP2530.x too large

Warning

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:

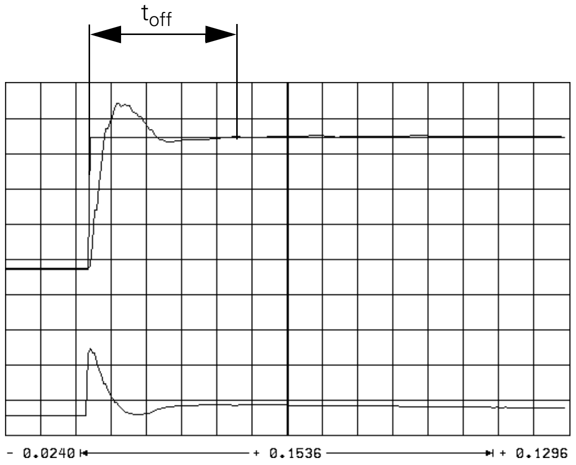
- Calculate the frequency of the oscillation and enter it in MP2550.x.
- Increase the band-rejection filter damping in MP2540.x until the interfering oscillation is minimized. Realistic input values: 3 to 9 [dB]

Note

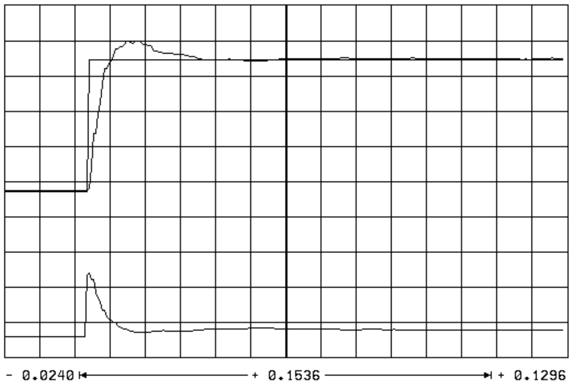
The compensation dampens the control loop. Try first to remove the mechanical causes of the disturbance oscillations.

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).

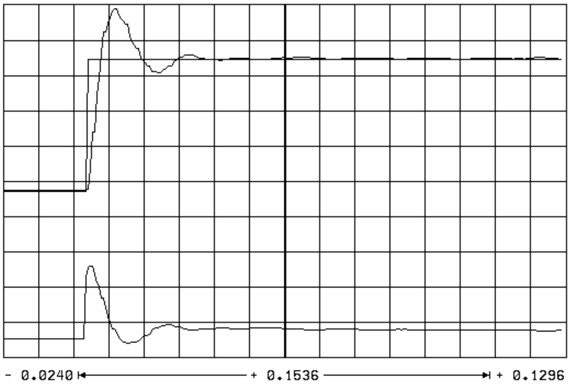
- Increase MP2510.x (I factor) until you see one overshoot followed by a slight undershoot and the settling time t_{off} is as small as possible (realistic value: 3 ms to 15 ms)



MP2510.x at optimum setting



MP2510.x too small



MP2510.x too large

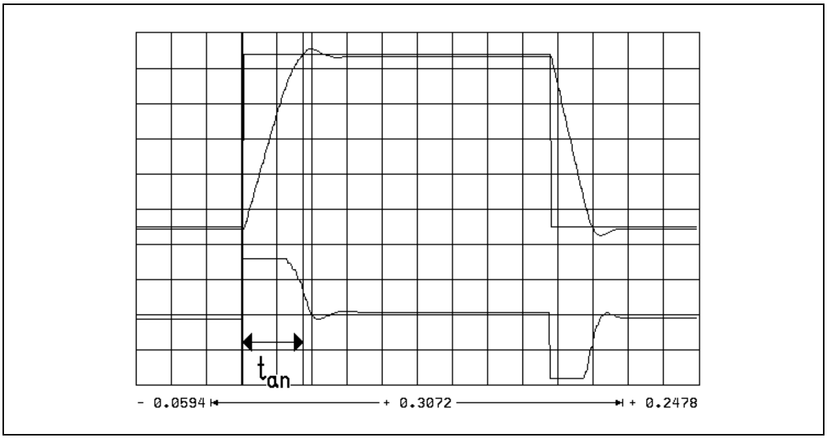
Determining the acceleration

- Clamp an object of maximum permissible weight on the machine table.
- Enter the rapid traverse 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.
- To 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

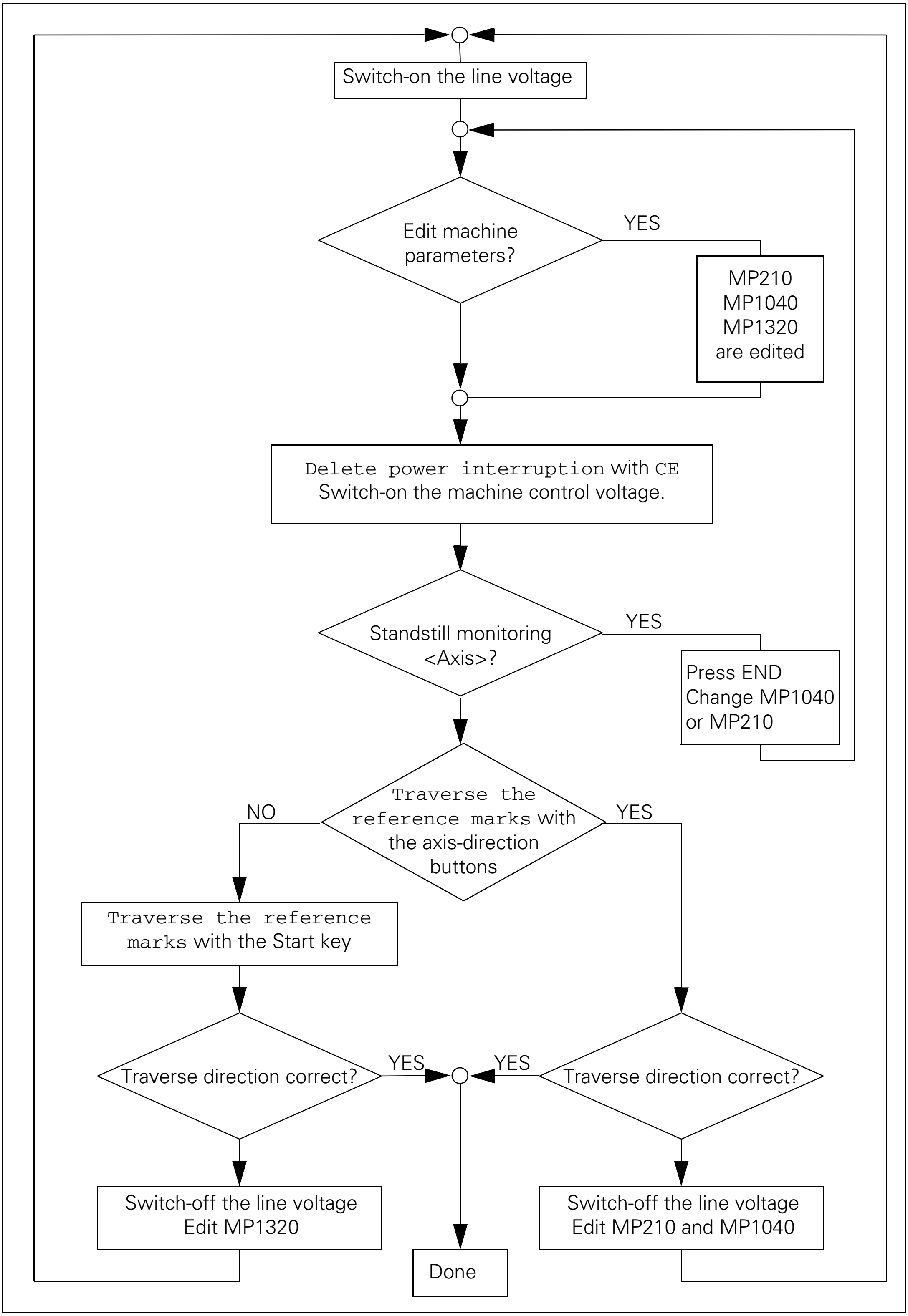
Adjusting the position control loop:

- 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
MP210
MP1040
MP1320
are edited
Delete power interruption with CE
Switch-on the machine control voltage.
Standstill monitoring <Axis>?
YES
Press END
Change 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
Done
Switch-off the line voltage
Edit MP210 and MP1040

Step 2: Set the traverse range

You can enter up to three traverse ranges.

See “Traverse Ranges” on page 6 – 19. 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:

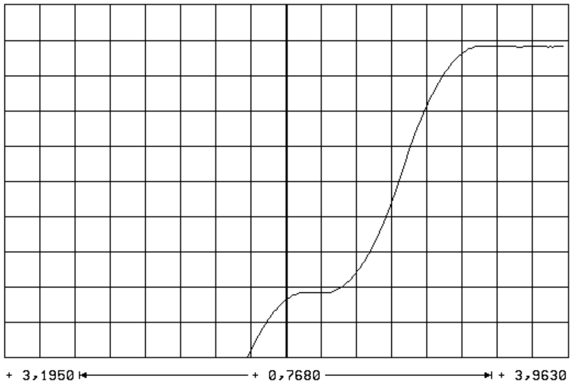
- Enter the temporary input values.

Machine Parameters	Temporary input value
MP1391 or MP1392	Set to velocity feedforward control
MP1090.0	Enter a very small jerk, e.g. 1 (dependent on the machine)
MP1090.1	
MP1092	<greater than rapid traverse>
MP1095	0
MP1096	0
MP1099.0	5
MP1099.1	3
MP1396.x	1
MP1521.x	0

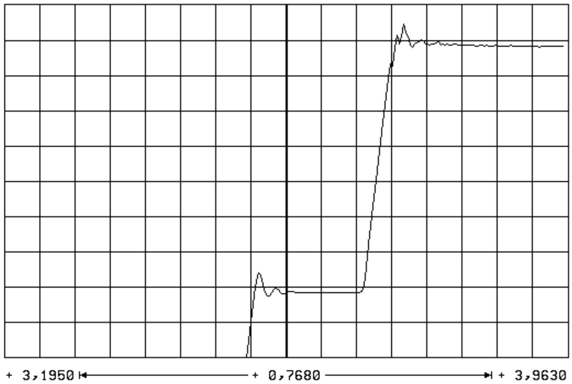
- Enter the following test program:

```
LBL 1
L X <maximum traverse>
R0 FMAX
LX0 FMAX
CALL LBL1 REP 100/100
```

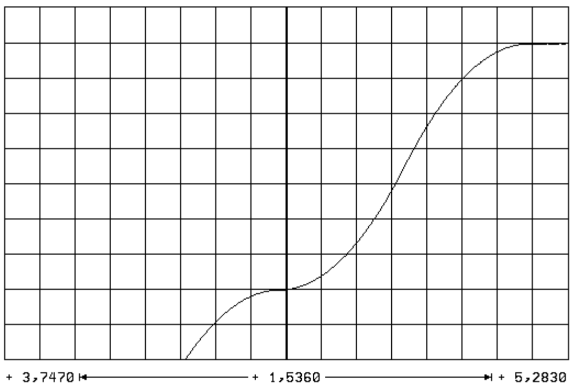
- Display the actual feed rate (v actual) with the integrated oscilloscope and, if necessary, also show the following error (lag).
- Start the test program with feed-rate override = 100%.
- In MP1090.0 increase the jerk until the overshoot just disappears.



MP1090.x at optimum setting

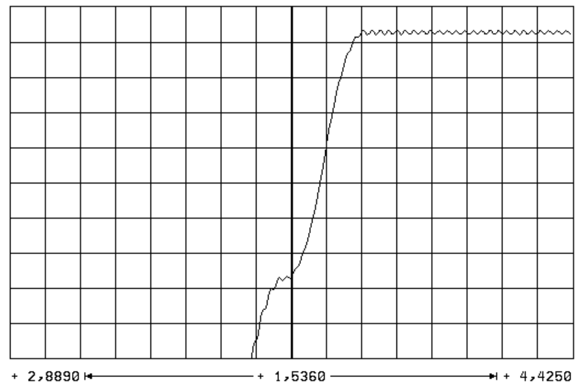


MP1090.x too large

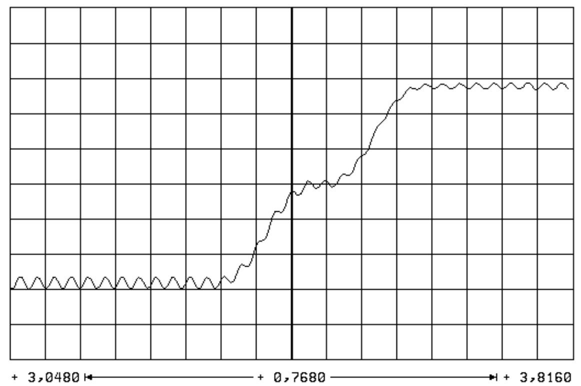


MP1090.x too small

- Enter the jerk determined from MP1090.0 in the axis-specific parameters 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



k_v 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 6 – 111. MP1090.x applies to all axes. The worst axis determines the input value.

Procedure:

- Assume the axis specific values in MP1097.x and MP1098.x.
- 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.
- From the tables on Page 6–117, select the input values for MP1099.x or MP1094. Consider the lowest determined frequency and the desired damping at this frequency.
- With MP1095 you select the single or double filter. With MP1094 the HSC filter is switched on, and the single and double filters are switched off.
- Test the three filter settings using a test part made of short line segments.
  - Single filter
  - Double filter
  - HSC filter

Note

If you have selected the best nominal position value filter for your application, please note that your input value can be overwritten by the machine user through Cycle 32.
If you have switched off the nominal position value filter (MP1096 = 0), the machine user can also switch it on using Cycle 32.

Control with following error (servo lag):

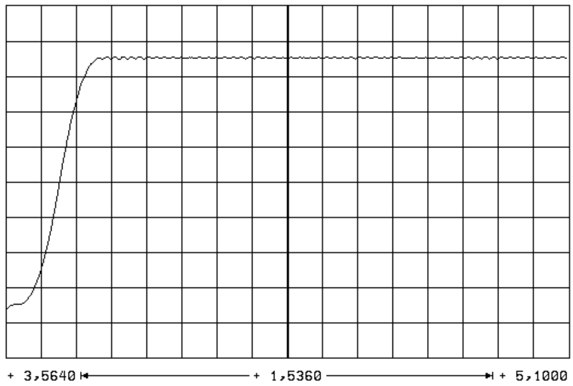
The adjusted maximum jerk works during operation with following error. MP1090 is not changed.

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 6 – 111.

Procedure for defining a characteristic curve kink point:

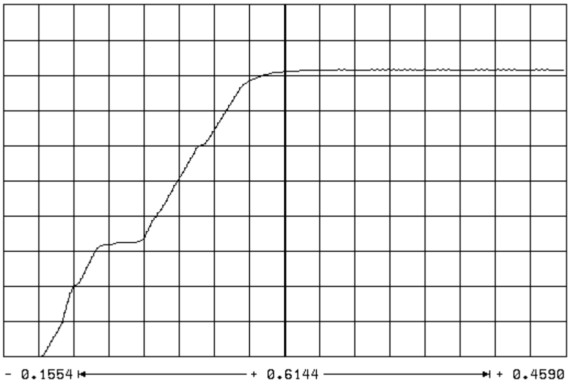
- k_v factor for rapid traverse (characteristic curve kink point):

$$\text{MP1830.x} = \frac{\text{Max. contouring feed rate} \cdot 100\ \%}{\text{Rapid traverse}}$$

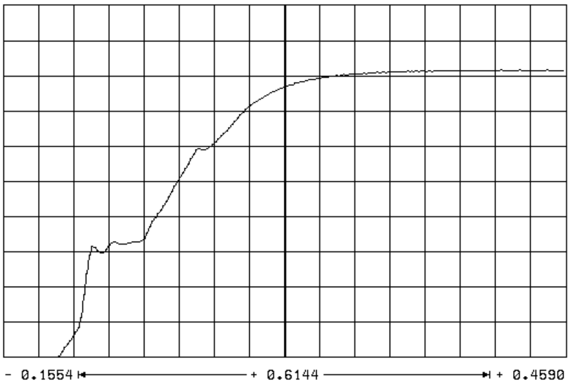
MP1820.x = 1

Set to operation with following error.

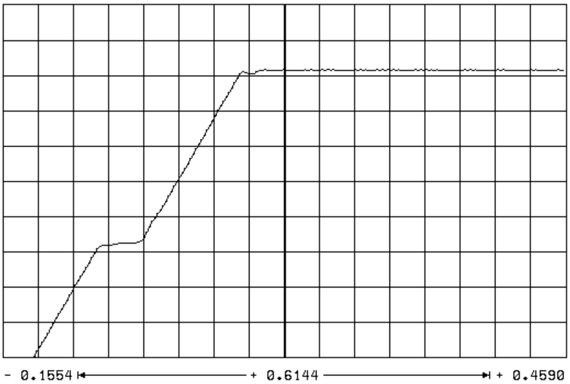
- Enter the following test program:
  LBL2
  L X <maximum traverse> R0 FMAX
  L X0 R0 FMAX
  CALL LBL1 REP 100/100
- Start the test program.
- Display the actual feed rate (v actual) with the internal oscilloscope:
  - If no oscillations are recognizable, no kink point is required.
  - If oscillations 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, enter a defined tolerance (e.g. 0.02 mm).

Step 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 rpm]	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* · ΔV ACTUAL (keep in mind the units for *t* and ΔV ACTUAL)
  - MP752 = Approx. 20 ms (determined in test)

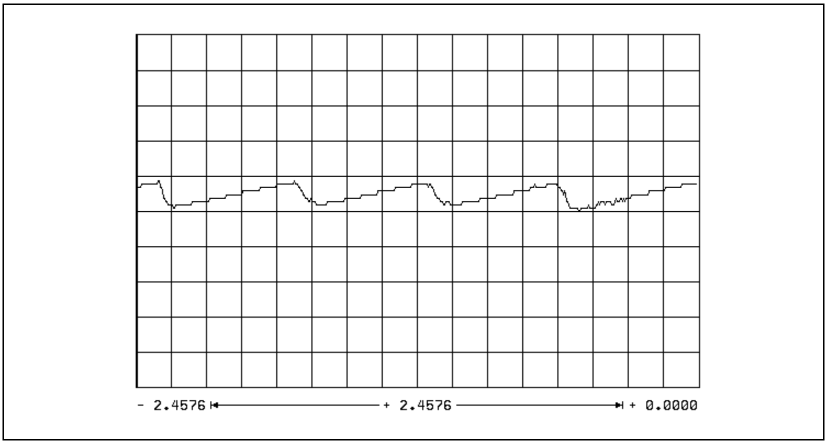
$$\Delta V\ ACTUAL = |V\ ACTUAL - V\ (ACT\ RPM)|$$

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):

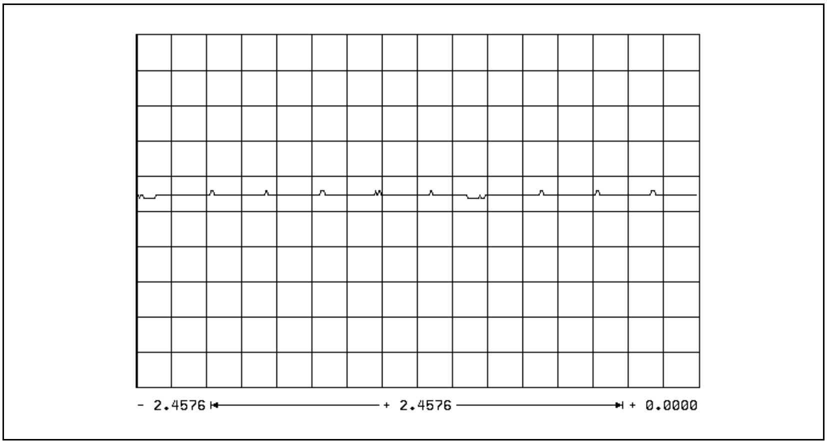
```
LBL 1
L X+400 IY+0.5 R0 F200
L X0 IY+0.5 R0
CALL LBL1 REP 100/20
```

- 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.

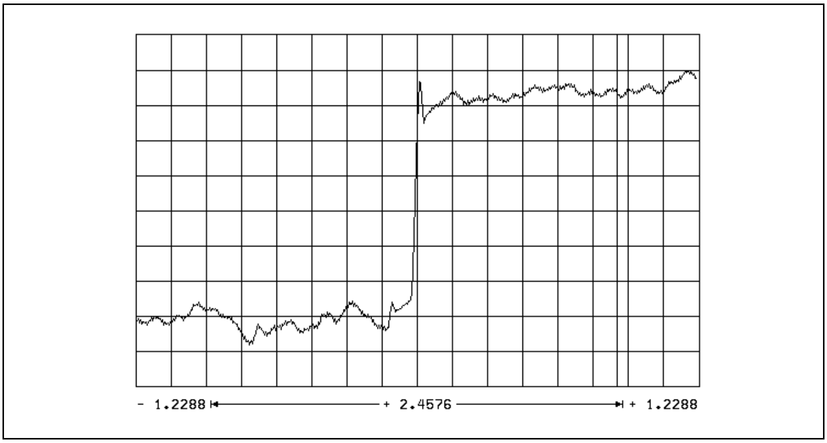
- Increase MP2512.x until the axis remains stationary.

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 rpm (MP1054.x).
- Determine the current (I NOMINAL) in both directions of rotation.



- Calculate MP2630.x:

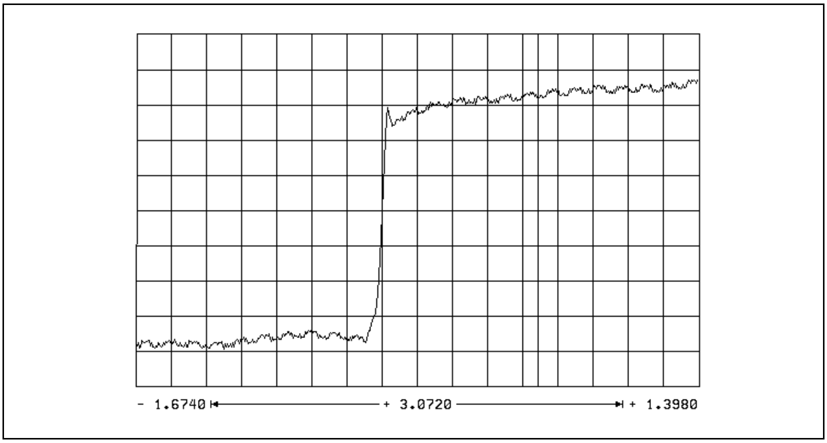
$$MP2630.x = \frac{I\ NOML_1 + I\ NOML_2}{2}$$

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
```

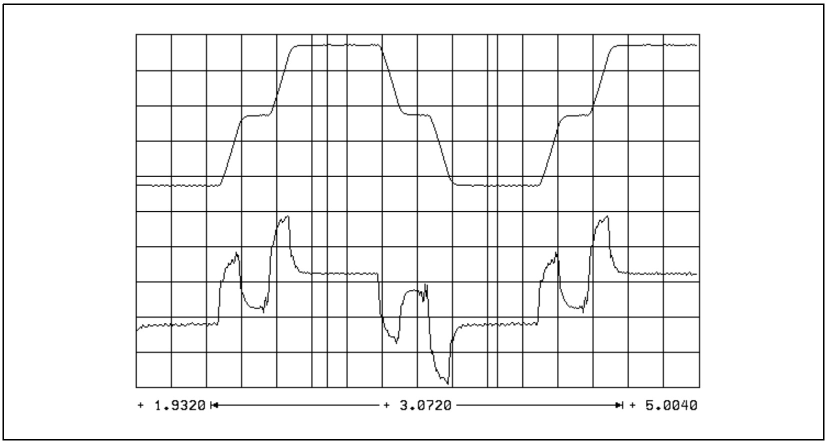
- Use the integrated oscilloscope to record the actual shaft speed (ACT RPM) and the nominal current value (I NOMINAL).
- Start the test program.
- With the feed rate override knob, adjust the motor speed to 10 rpm (MP1054.x).
- Determine the current (I NOMINAL) in both directions of rotation.



- Calculate MP2610.x:

$$MP2610.x = \frac{I\ NOML_1 + I\ NOML_2}{2}$$

- 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:

$$MP2620.x = \frac{(I_{max} - MP2610.x) \cdot \text{<rated speed>}}{n_{max}} + MP2610.x$$

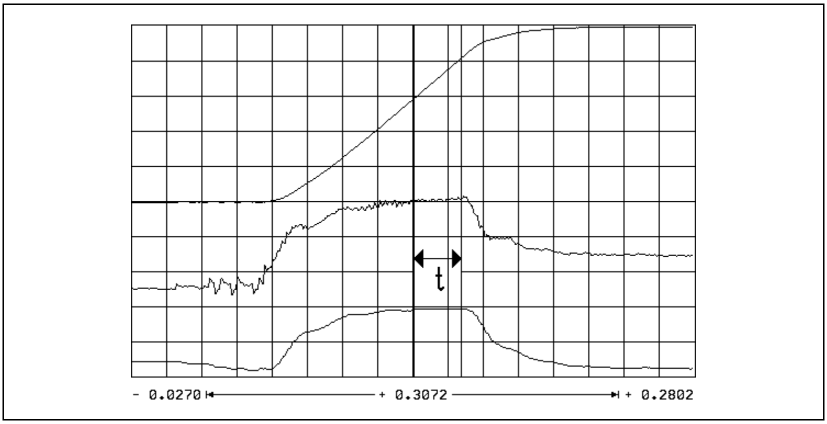
I_{nmax}: Current at rapid traverse

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:

$$\text{MP2600.x} = \frac{\text{I (INT RPM) [A]} \cdot \text{t [s]} \cdot \text{60 [s/min]} \cdot \text{MP1054.x [mm]}}{\Delta\text{V (ACT RPM) [mm/min]}}$$

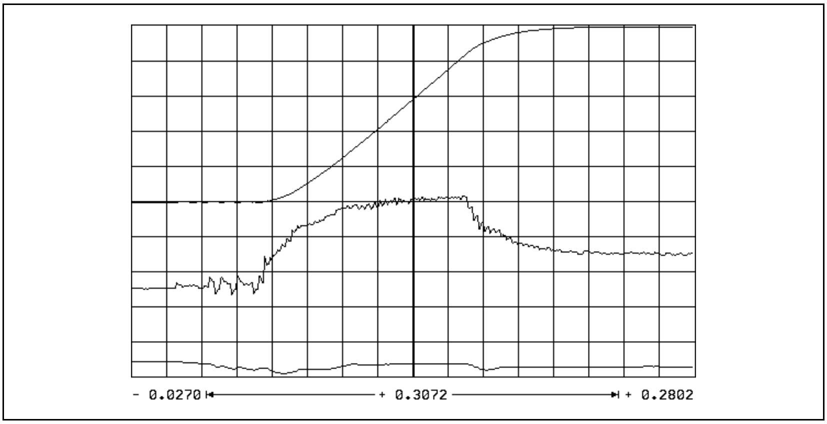
I (INT RPM) = Integral-action component of the nominal current value

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 rpm 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

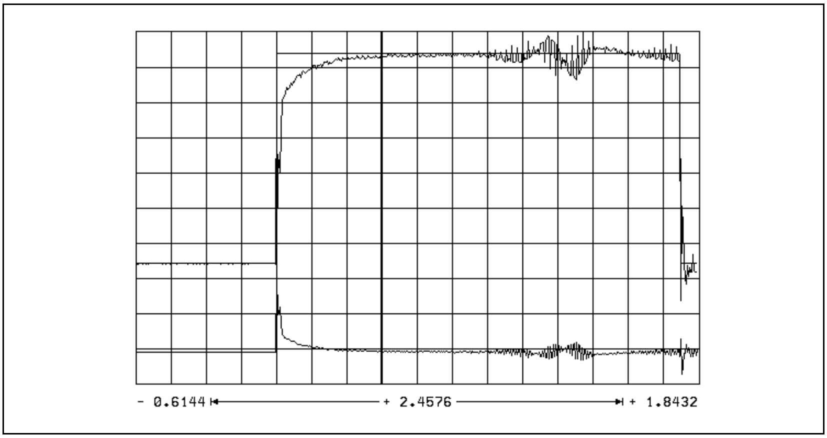
Current controller

Same procedure as for digital axes.

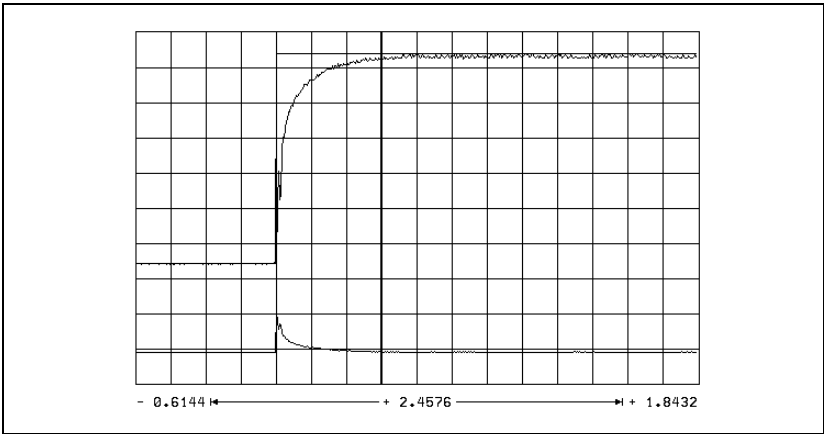
Speed controller

Define the step function:

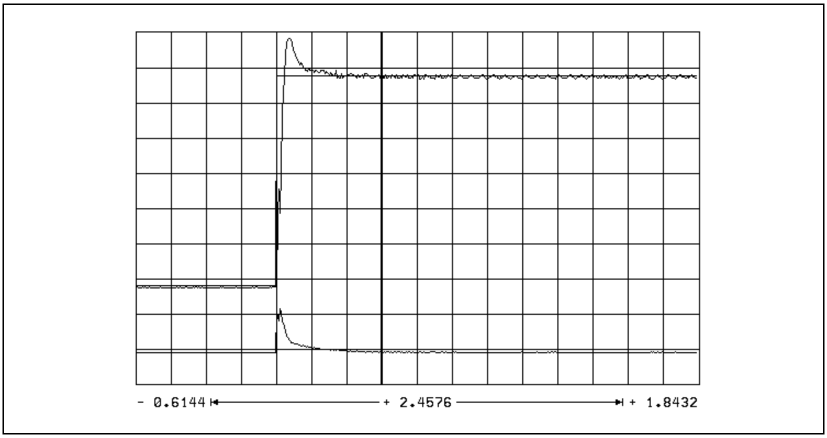
- In MP3411.x, enter the maximum acceleration and start the step by switching the spindle on.
- Activate a spindle speed from the highest gear range.
- With the integrated oscilloscope, record the nominal velocity value V (NOM RPM), the actual speed value V (ACT RPM), and the nominal current value (I 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

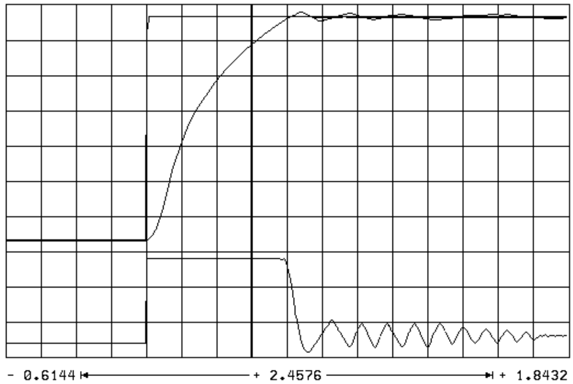


- Increase the I factor (MP2510.x) until you see one overshoot followed by a slight undershoot.

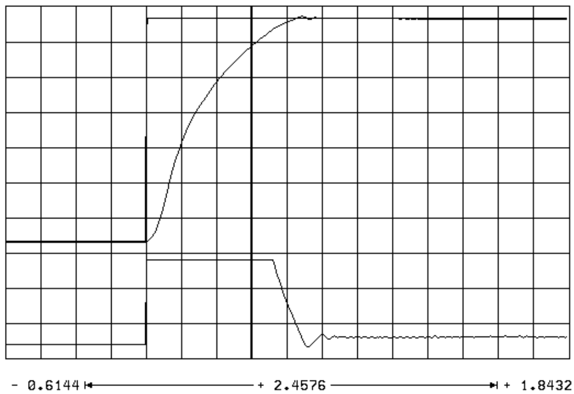


▶ Output the step with maximum shaft speed. I NOMINAL is within the limitation during acceleration. I NOMINAL must not oscillate after reaching the maximum speed. If I NOMINAL oscillates:

- Reduce MP2500.x and MP2510.x evenly until the overshoots are minimized.



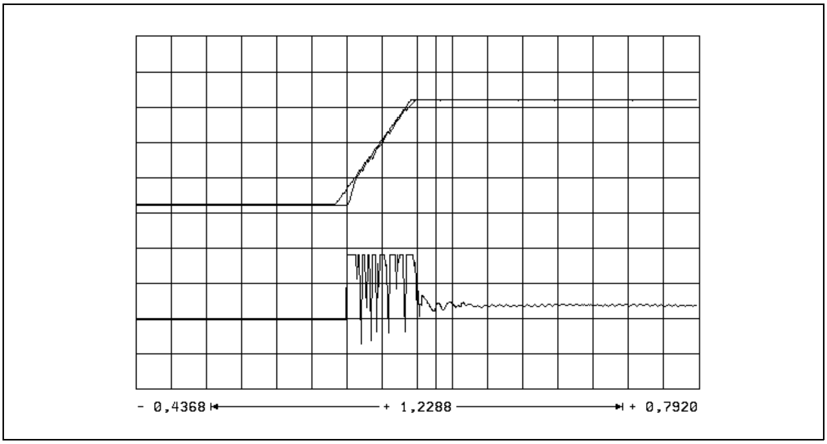
I NOMINAL oscillating



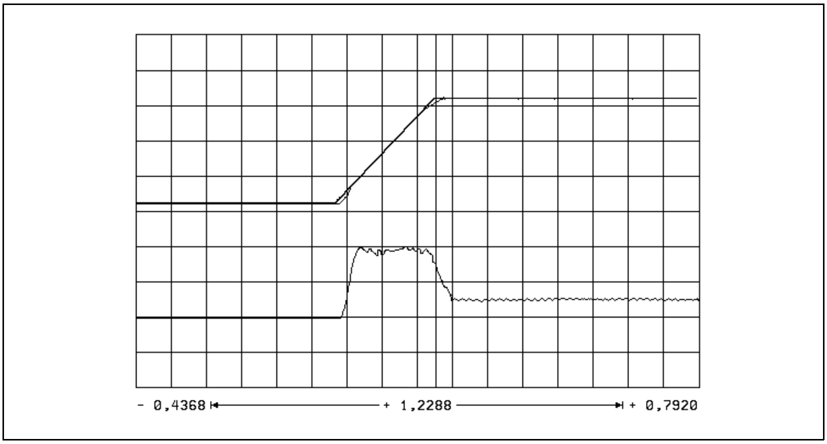
Only one overshoot

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.

Checking 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, see "Oriented Spindle Stop" on page 6 – 203.
  - 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.

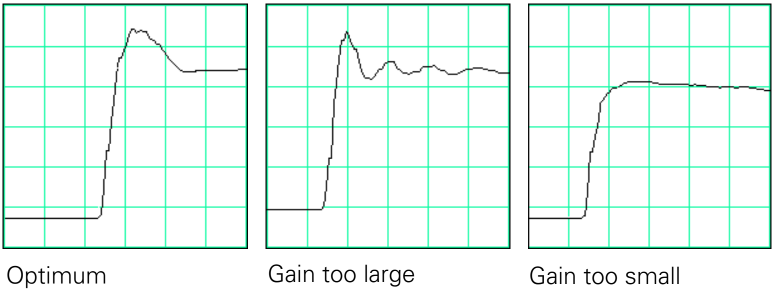
Procedure:

- Disconnect the nominal-value connection between the servo amplifier and the MC 422.
- 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 supply voltage 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 422.

▶ 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 VOLT.ANALOG 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.
- Test 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

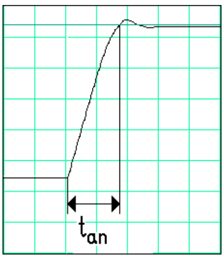
After adjusting the servo amplifier, you can determine from the step response the maximum possible acceleration:

$$a = \frac{F_{max}}{t_{an} \cdot 60\,000}$$

a: Acceleration [m/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.

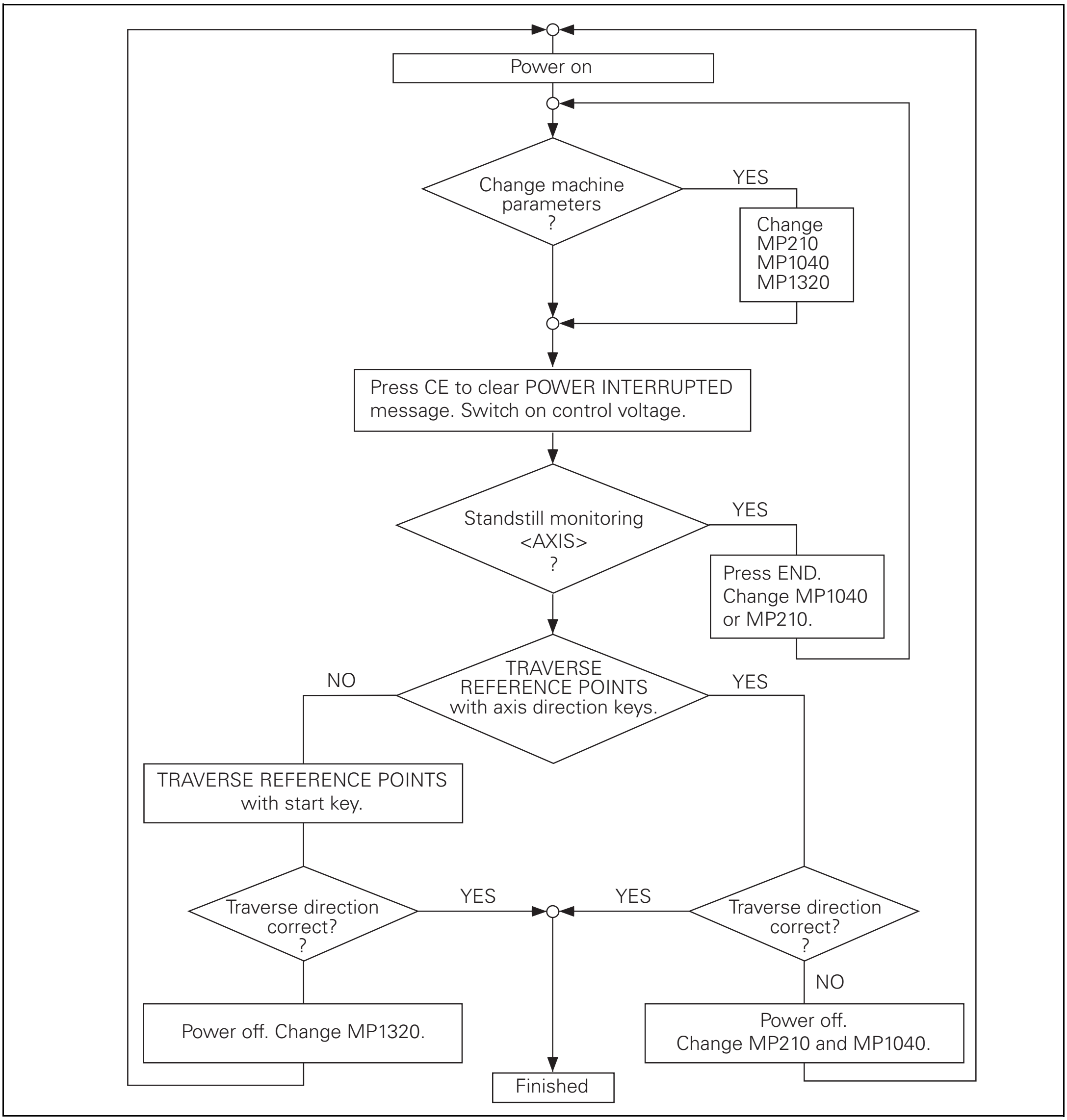
Position controller

> **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)



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
On the iTNC: see "The Control Loop" on page 6 – 111.

Step 5: Activate monitoring functions

- Enter the following temporary input values when you begin: see "Commissioning of Digital Axes" on page 6 – 244

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

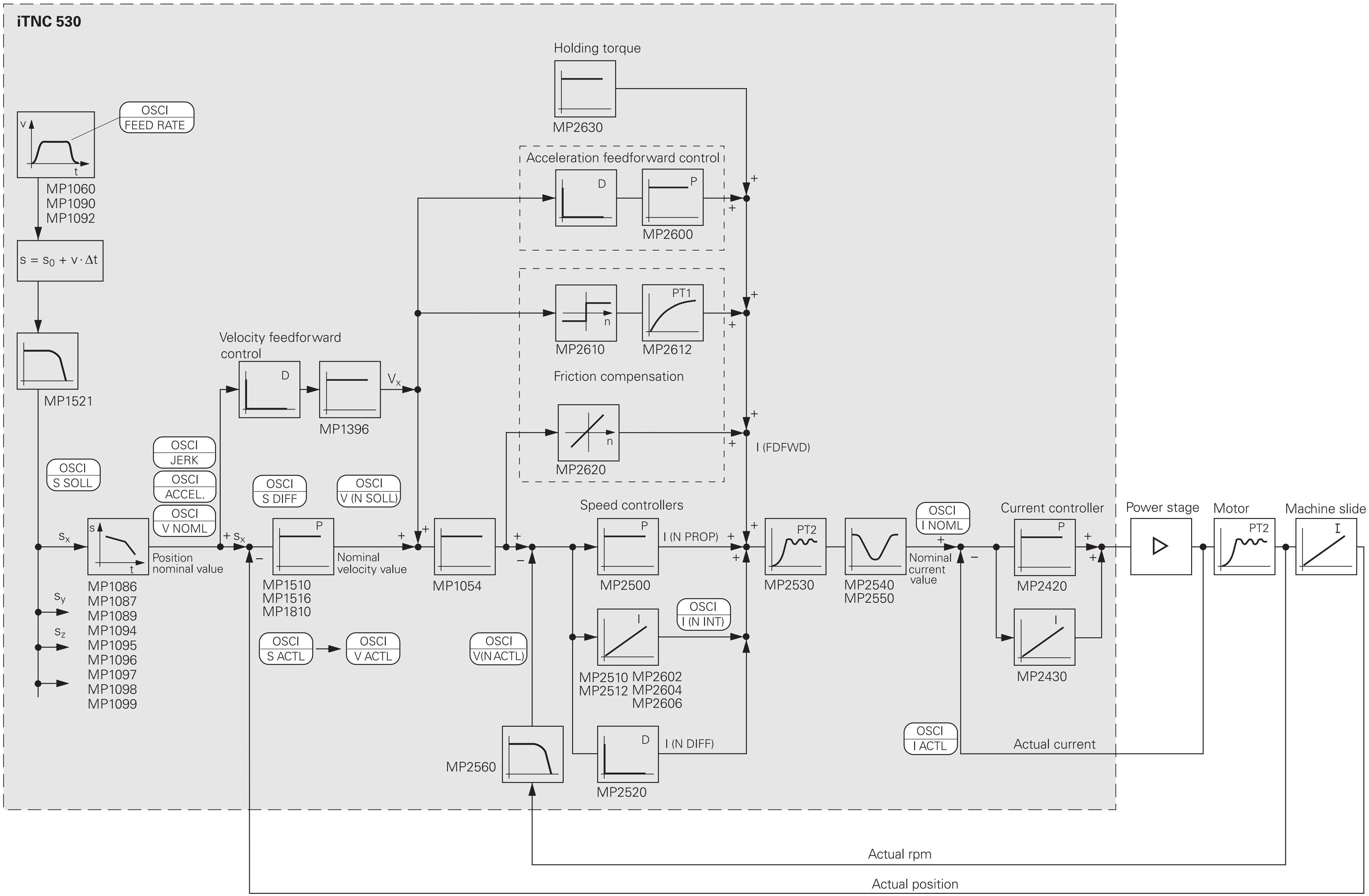
Adjusting the servo amplifier
Same procedure as for analog axes.

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 spindle.

6.15 Block Diagram iTNC 530



7 Machine Integration

7 Machine Integration

7.1 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 through soft keys. Refer to the User's Manual.

7.1.1 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.

This information includes:

- Axis positions
- Tools
- Feed rate
- M functions

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: %xxxxx
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 define the modulo value for the counting mode (i.e., the value after which the axis display returns to zero). Also, you can activate or deactivate the software limit switches of the traverse ranges:

▶ 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.

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

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: %xxxxx
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.

The values for all axes are read in, regardless of whether individual axes are excluded through MP10. Values for excluded axes are undefined.

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)
Call:
SEE MODULE 9041.

Module 9041 Reading of axis coordinates (format 0.0001 mm)
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: Servo lag
               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 titling-axis geometry
CM    9040 OR
CM    9041
```

Error detection:

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.

The maximum feed rate is 300 000 °/min. 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 taken into account.

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.

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 to 3), 4 and 5 for (X, Y, Z), axes IV and V
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

		Set	Reset
M4133	**Starting and stopping the free rotation function**	PLC	NC
B518	**Defining 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%.

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. See also page 6 – 50.

Output of 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.

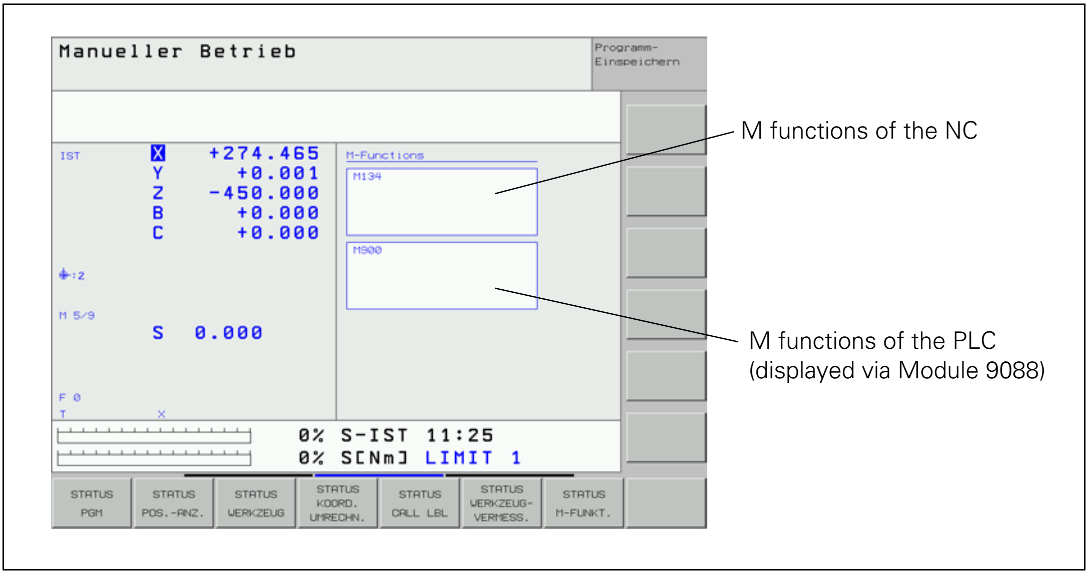
Other M functions of the NC are shown in the status window.

		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 detection:

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 erased because it is already being displayed by the NC

Error detection:

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 display

To erase the status display, tool data and contents of the Q parameters:

▶ Select the conditions with MP7300. All programmed values in the status display, such as scaling factor, datum shift, and feed rate are reset. The Q parameters and tool data are set to zero.

MP7300 **Erasing the status display and Q parameters**

Input: 0: Erase the status display, Q parameters and tool data if a program is selected.
1: Erase the status display, Q parameters and tool data if a program is selected and in the event of M02, M30, and END PGM.
2: Erase the status display and tool data if a program is selected.
3: Erase the status display and tool data if a program is selected and in the event of M02, M30, END PGM.
4: Erase the status display and Q parameters if a program is selected.
5: Erase the status display and Q parameters if a program is selected and in the event of M02, M30, END PGM.
6: Erase the status display if a program is selected.
7: Erase the status display if a program is selected and in the event of M02, M30, END PGM.

Input	Erase if PGM MGT	Erase if M02, M30, END PGM	Status display	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.

Transferred number		Return code
0	Editor mode in foreground	0: **Programming and Editing** 1: **Test Run**
1	Machine mode in foreground	0: **Cross over 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: Directory/external screen active 3: MP editor active 4: PLC editor active
3	Machine mode in background	0: None (main operating mode active) 1: MOD active 2: Directory/external screen 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=1: PLC status window active Bit 5=1: Status/Graphics window active Bits 6/7: Reserved Bits 8 to 15: Machine screen Bit 8=1: Machining screen is displayed Bit 9=1: Window for operating mode active Bit 10=1: Block display/program select/setup window active Bit 11=1: Position display active Bit 12=1: PLC status window active Bit 13=1: Status/Graphics window active Bits 14/15: Reserved

Transferred number		Return code
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)
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 interpolation 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 (ISO NC PGM)
17	Display format	0: mm 1: inches
18	Tilting working plane	Bit 0=1: Tilting is active Bit 1=1: Tilting is selected for manual operation Bit 2=1: Tilting is selected for 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) 20: ATEK M

Transferred number		Return code
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	Machine operating mode in background, table in editing mode	0: No table in editing mode 1: Tool table in editing mode 2: Pocket table in editing mode
	Handwheel interpolation factor	
31 32 33 34 35 36 37 38 39	Axis 1 Axis 2 Axis 3 Axis 4 Axis 5 Axis 6 Axis 7 Axis 8 Axis 9	0 to 10
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 processing	–1: Main program is not a pallet table 0: Processing 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 the 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 detection:

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

7.1.2 Operating Times

The iTNC can measure up to 11 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 PLCTIME7	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 8:

- 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 2 – 41.

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: %xxxxxxxx

MP7237.0 Displaying PLC operating times

Input: Bits 0 to 7 represent PLC operating times 1 to 8
0: Do not display
1: Display

MP7237.1 Resetting PLC operating times with the code number 857282

Input: Bits 0 to 7 represent PLC operating times 1 to 8
0: Do not reset
1: Reset

MP7237.2 Resetting NC operating times with the 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

MP7238.0-7 **Dialog messages for PLC operating times 1 to 8**

Input: 0 to 999
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 7 represent PLC operating times 1 to 8
CM      9190
```

Error detection:

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 7 represent PLC operating times 1 to 8
CM      9191
```

Error detection:

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 7: PLC operating times 1 to 8
CM	9192	
PL	B/W/D	<Current time [s]> -1: Error

Error detection:

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 7: PLC operating times 1 to 8
PS	B/W/D/K	<New time [s]>
CM	9193	

Error detection:

Marker	Value	Meaning
M4203	0	Operating time was overwritten
	1	Incorrect transfer value, or module was not called in a spawn job 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 7: PLC operating times 1 to 8
PS    B/W/D/K   <Alarm threshold [s]>
PS    B/W/D/K   <Number of the alarm markers>
CM    9194
```

Error detection:

Marker	Value	Meaning
M4203	0	Alarm function activated
	1	Incorrect transfer value, or module was not called in a spawn job or submit job

System 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.

In MP7235, enter the time difference between the local time and UNIX system time (Universal Time, Greenwich time) so that the time of the program management 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 MP7235 and always refers to Universal 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 summer 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 by the difference to local time as entered in MP7235.

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 detection:

Marker	Value	Meaning
M4203	0	String was generated
	1	Incorrect transfer value

7.1.3 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 9085 or by activating a marker (M4800 to M4899):

- 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 999 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.

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 page7 – 26).
  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 M4899). 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 blinks.

- **NC STOP**
  0: No NC stop upon activation of the error message
  1: NC stop upon activation of the error message
- **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
- **F STOP**
  0: Feed-rate enable is not influenced
  1: Feed rate-enable is reset upon activation of the error message
- **EMER.STOP**
  0: No EMERGENCY STOP upon activation of the error message
  1: EMERGENCY STOP upon activation of the error message
- **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 is the highest. The active PLC error messages are displayed in order of priority.
- **MType**
  This column is reserved for future applications. Enter "E".

Module 9085 Display PLC error messages

Up to 32 error message 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 999: Line number –1: Blinking error message **EMERGENCY STOP PLC**
CM 9085	

Error detection:

Marker	Value	Meaning
M4203	0	Error message displayed or in queue
	1	Error code in W1022
W1022	1	Line number not available
	8	Incorrect operating mode, error marker compatibility set
	23	Overflow of PLC error message queue, or too many error messages from string memory

Module 9086 Erase PLC error messages

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 999: Line number –1: Erase all PLC error messages
CM 9086	

Error detection:

Marker	Value	Meaning
M4203	0	Error message displayed or in queue
	1	Error code in W1022
W1022	1	Line number not available
	8	Incorrect operating mode, error marker compatibility set

Module 9087 Status of PLC error message

Call:

PS B/W/D/K	<Line number of the *.PET table, status code> 0 to 999: 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 –1 to 999: 0: No error message with the number, or message deleted –1: Line number does not exist Bit 0 – PLC error message is displayed Bit 1 – 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 detection:

Marker	Value	Meaning
M4203	0	Status information was read
	1	Error code in W1022
W1022	1	Invalid line number of status code

7.1.4 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 *.HLP file.

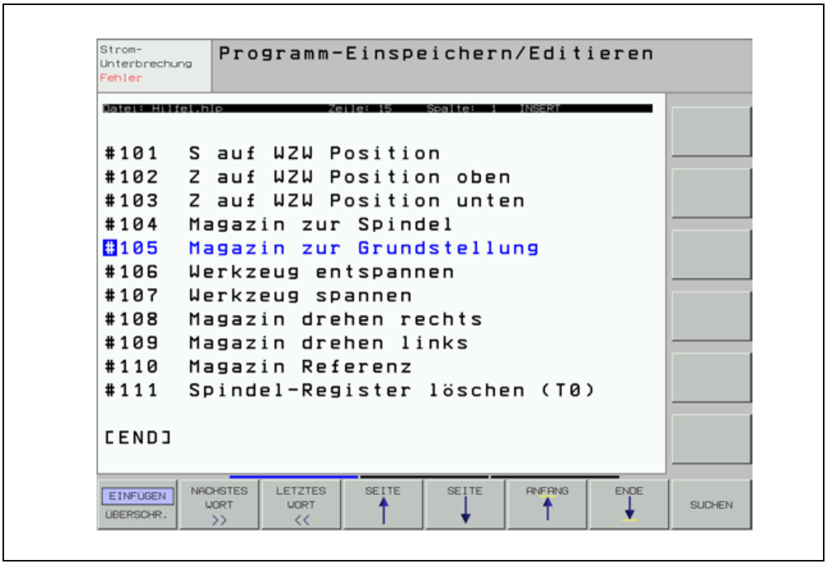
With machine commands:

- 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 the 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.

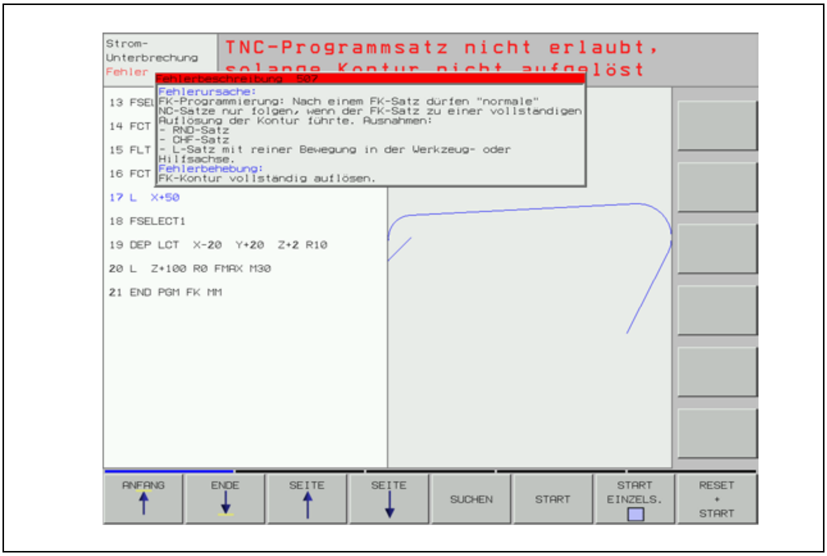
HELP files are saved externally with the identifier “J”.

		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 M4222 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

7.1.5 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 the pop-up window with Module 9215.

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):

▶ Transfer the selection list to the module in a file.

You separate the individual entries with <LF>.

If you transfer the 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 module does not return until the pop-up window is closed. It 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.

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.

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 detection:

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

7.1.6 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 datums:

- 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.

The datum can be set either only by the “Datum setting” soft key or only 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.

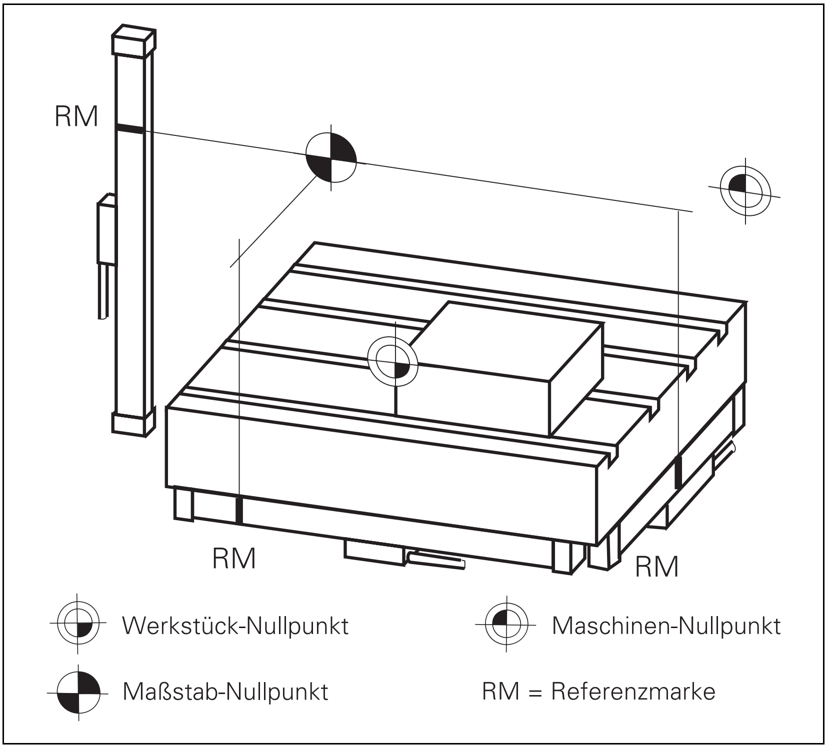
▶ Enter the code number 555 343.

▶ 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

As of NC software 340 420-01, 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.



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

MP7294 **Disable axis-specific “Datum setting” in the preset table**
As of software version:340 422-01
Format: %xxxxxxxxxxxxxx
Input: Bits 0 to 13 represent axes 1 to 14
0: Not disabled
1: Disabled

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

7.1.7 NC Program

Block number increment for ISO programs

▶ In MP7220 enter the block number increment for ISO programs.

MP7220 **Block number increment for ISO programs**
Input: 0 to 250

Finding the block number

Module 9321 Find the current block number
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 detection:

Marker	Value	Meaning
M4203	0	Block number has been found
	1	Error code in W1022
W1022	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

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 which 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	**Depiction 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 =**.

Automatic 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.
  If you want the program to be started by the PLC, use one of the two following markers:
  - 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.

7.1.8 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.

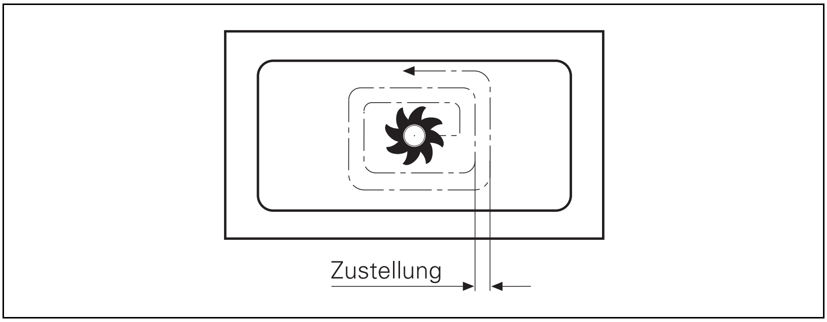
For more information on the tapping cycle and the oriented spindle stop cycle, see page 6 – 187.

See page 7 – 166 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.



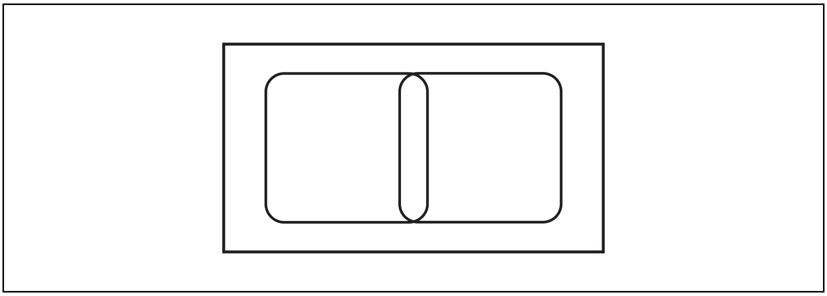
Infeed = (MP7430) · cutter radius

MP7430 **Overlap factor for pocket milling**
Input: 0.001 to 1.414

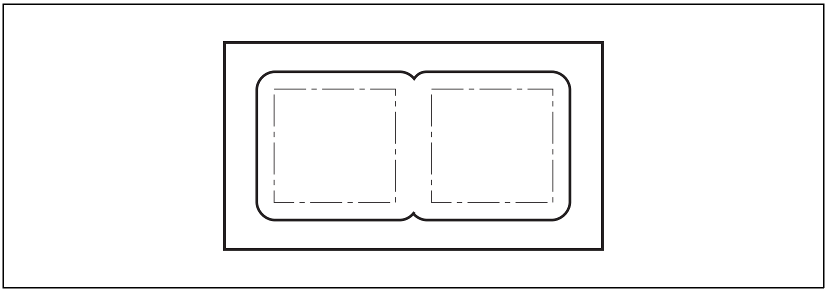
Cycles for milling pockets with combined contours

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

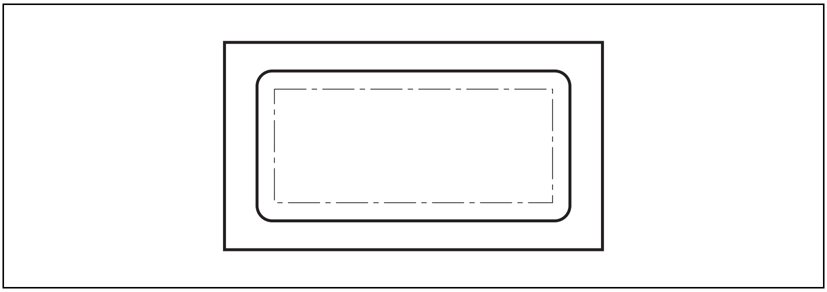


The programmed contours of two pockets intersect slightly.



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
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 moves only in the tool 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 and 28 you can machine a contour on a cylindrical surface (see the User's Manual).

- Define the rotation center of the rotary axis with MP75xx. See also page 6 – 50.
- Define the behavior of Cycle 28 with MP7680 bit 12.

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

7.1.9 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

7.1.10 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

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 7 – 196), 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 `Program Run, 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 8 – 20.

Note

The tool data cannot be correctly offset in the block scan if you change the data in the PLC or update it with M4538.

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 300 000 [mm/min]

MP7680 **Machine parameter with multiple function**
Format: %xxxxxxxxxxxxx
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 (MOVE TO 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 instruction **ORDER=PRIO**):

- 1st: M function that was defined with MFIRST
- 2nd: M/S/T/Q signals in the programmed sequence
- 3rd: 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, and so led to the **PLC function not permitted** error message, there are functions that can be executed from 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 codeword **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 **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

		Set	Reset
M4161	**M/S/T/Q transfer after block scan**	NC	NC

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.
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 7 – 196 and “Calling an NC macro with an M function” on page 7 – 53.
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`

7.1.11 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 functions	Meaning	Effectiveness
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 functions	Meaning	Effectiveness
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
M 97	Machine small contour steps	E
M 98	Machine open contours completely	E
M 99	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 defined datum	A
M105	Machine with second k_V factor	A
M106	Machine with first k_V factor	A
M107	Suppress error message for replacement tools	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 correction of machine geometry when working with tilting 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	LOOK AHEAD: Calculate the radius-compensated tool path ahead of time	A
M121 - M123		

M functions	Meaning	Effectiveness
M124	Ignore points for calculating the rounding radius with M112	A
M125		
M126	Permit zero crossover on 360° 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	Reduce jerk during axis-specific changes of acceleration	A
M133	Reset M132	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 kinematic 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 - M199		A
M200	Laser cutting: Direct output of the programmed voltage	A
M201	Laser cutting: Voltage output varies with the distance	A
M202	Laser cutting: Voltage output varies with the velocity	A
M203	Laser cutting: Voltage output varies with the time (ramp)	A
M204	Laser cutting: Voltage output varies with the 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 M functions 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 Status of M functions

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 detection:

Marker	Value	Meaning
M4203	0	Status was found
	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 detection:

Marker	Value	Meaning
M4203	0	Status was found
	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 7 – 191.

- 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, an exact stop is made 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

Help cycles

Cycles 18 (thread cutting) and 33 (thread on taper) are so-called 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 **Disabling 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 8 – 24.

For behavior during a block scan, See "Instructions in MGROUPS.SYS" on page 7 – 43.

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 acknowledgement 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.	

7.1.12 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.

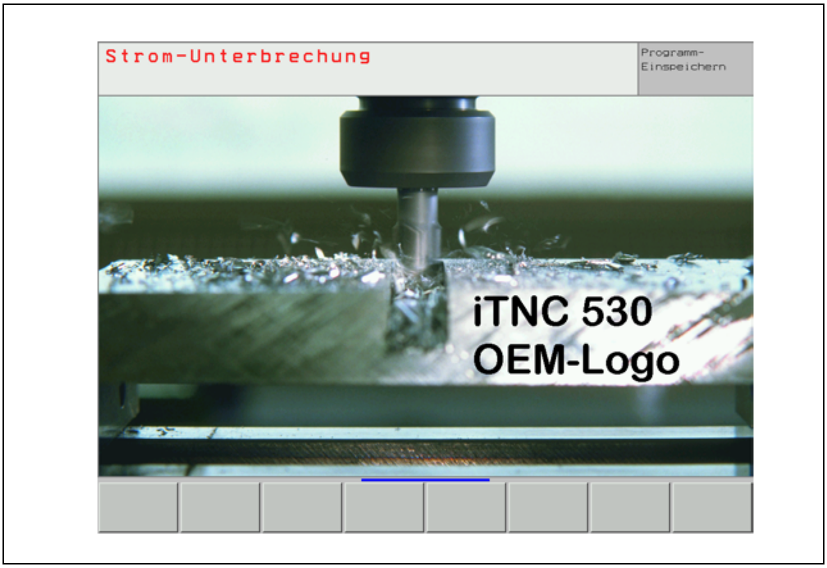
Requirements of the logo:

- 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



- Enter the keyword **LOGO =** in OEM.SYS, followed by the complete path of the logo, e.g. **LOGO = PLC:\LOGO\OEM-LOGO.BMP**
- On the TNC, create a new directory, for example **PLC:\LOGO.**
- Move the logo into this directory, for example **PLC:\LOGO\OEMLOGO.BMP.**

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.

		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 detection:

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 Shutting down the control
Module 9189 shuts down the control. The PLC is not executable after shut down. The information windows, which appear during shutdown via soft key, do not appear.

Call:
CM 9189

Error detection:

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

7.1.13 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]

7.1.14 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:

- Enter MP7246 = 1 to disable the input of paraxial positioning blocks.

MP7246	**Disabling paraxial positioning blocks**
Input:	0: Paraxial positioning block enabled 1: Paraxial positioning block disabled

7.1.15 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 Specifying the 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)

7.1.16 Code Numbers

You can enter certain code numbers in 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
NET123	Ethernet settings
LOGBOOK	Read out the log
FAILTEST	Simulate an internal EMERGENCY STOP
SIK	Option menu

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

7.1.17 Programming Station

With MP7210 you can set up the control for use as a programming station without a machine.

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.

MP7210 **Programming station**
Input: 0: Controlling and programming
1: Programming station with PLC active
2: Programming station with PLC inactive

7.1.18 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 setting 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 the time in minutes after which the screen saver should activate itself. Enter 0 to disable the screen saver.

MP7350 Window frames

MP7351 Error messages

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	Background of inactive window
MP7355	**“Programming” program text display**
MP7355.0	Background
MP7355.1	General program text
MP7355.2	Active block
MP7355.3	Background of inactive window
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”)
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

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 6

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
MP7367.1-14	Colors 1 to 14

MP7368	**Pocket 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	**Screen saver**
Input:	1 to 99 [min]
	0: No screen saver

The standard color setting is shown in the following list:

Machine parameters	Standard setting
MP7350	$0808080
MP7351	$0FF2020
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	$0C0C0C0
MP7355.0	$0FFFFFF
MP7355.1	$0000000
MP7355.2	$00000FF
MP7355.3	$0ECECEC
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
MP7361.0	$0AAAAAA
MP7361.1	$00000E8
MP7361.2	$00000E8
MP7361.3	$0FF0000
MP7361.4	$0FF00FF

Machine parameters	Standard setting
MP7362.0	$0ECECEC
MP7362.1	$0FFFFFF
MP7362.2	$00000FF
MP7362.3	$00000FF
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	$0FF0000
MP7367.2	$000FF00
MP7367.3	$00000FF
MP7367.4	$0C0C0C0
MP7367.5	$0FFFFFF
MP7367.6-14	$0000000

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	$0FFFFFF
MP7370.1	$0FFFFFF
MP7370.2	$0ECECEC
MP7370.3	$0ECECEC
MP7370.4	$0C0C0C0
MP7370.5	$0C0C0C0
MP7370.6	$0000000
MP7370.7	$00000FF
MP7370.8	$0808080
MP7370.9	$0000000
MP7370.10	$0000000
MP7370.11-14	$00000FF
MP7370.15	$0FF2020

7.1.19 Graphic Display

In the graphics window you can view the following graphics:

- Test graphics
- Parallel graphics
- Programming graphics
- Help illustration

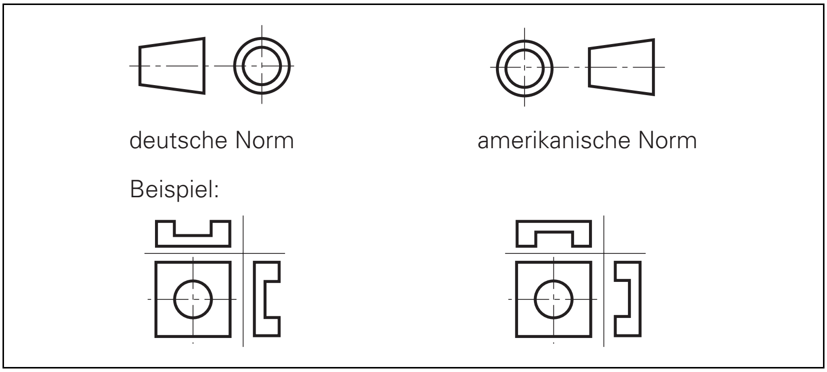
For the test graphics and parallel graphics you can choose one of three display modes:

- 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:

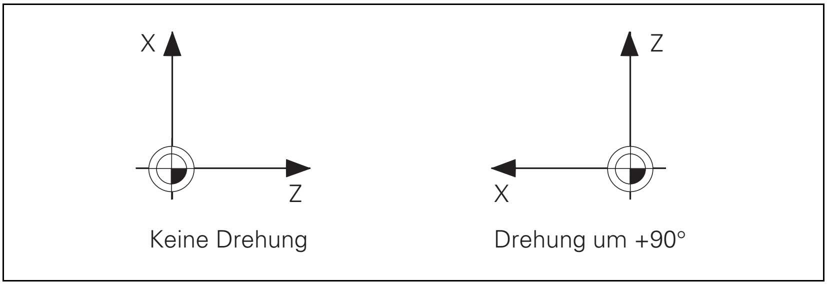
▶ Select the type of projection with MP7310, bit 0.



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:

- In MP7310 bit 2 define the BLK form shift.

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.

Graphics: 3-D view

The iTNC offers two display modes for 3-D graphics:

- 2.5-D graphics
- 3-D graphics

The 3-D graphics allows you to display 5-axis machining operations. Due to the increased computing effort, however, the display-building process takes longer.
If you have selected the 3-D graphics with MP7310, the 3-D graphics is only activated for 5-axis machining operations. Otherwise, the 2.5-D graphics remains effective.

- With MP7310 bit 5, you activate the 3-D graphics for the **Program Test** mode.
- With MP7310 bit 6, you activate the 3-D graphics for the **Program run, full sequence** and **Program run, single block** modes.

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 1: 2.5-D and 3-D Bit 6 – 3-D graphics during program run 0: 2.5-D 1: 2.5-D and 3-D Bit 7 – Reserved

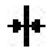
7.1.20 Special Characters

To enter special characters, use the following key combinations:

Key combination	Special characters
SHIFT + "	’
SHIFT + &	@
SHIFT + (	[
SHIFT +)	]
SHIFT + –	_
SHIFT + /	\
SHIFT + !	|
SHIFT + ^	~
SHIFT + #	ESC

7.1.21 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		

7.1.22 Conversational Language

The TNC is delivered with all 14 NC conversational 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\	CZECH\
	DANISH\
	DUTCH\
	ENGLISH\
	FINNISH\
	FRENCH\
	GERMAN\
	ITALIAN\
	POLISH\
	PORTUGUE\
	SPANISH\
	SWEDISH\
	HUNGARIA\
	RUSSIAN\

▶ 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:

▶ 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.

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
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

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 period

7.1.23 Log

The 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.

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)
KEY		Key strokes
STIB[a]	ON	Control-in-operation on
	OFF	Control-in-operation off
	BLINK	Control-in-operation symbol blinking
INFO	MAIN START	Control model and NC software
INFO	MAIN FILE DEL	Faulty files on the hard disk, to be erased when started up
INFO	MAIN HDD	Hard disk designation
INFO	MAIN CYCLES	Test results for fixed cycles and touch probe cycles

a. STIB = 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 started 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 (V.11/V.24) Byte 2/3 xx xx Internal error class Byte 4...7 xx xx xx xx Internal error code
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 kinematic table
INFO WARNING ERROR	PLC <log identifier>		Entries through PLC Modules 9275 and 9276

Entry		Description
INFO	REMO A_LG	Log in with LSV2 protocol
	REMO A_LO	Log out with LSV2 protocol
	REMO C_LK	LSV2 protocol: Locking and releasing the keyboard; the key codes between locking and releasing are sent via LSV2 protocol

You can read out the 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, or TNCremoNT offers you several functions for reading out the log.

Example of a log entry

The following example shows possible entries in the log:

```
INFO:     MAIN   START                      21.11.2001 07:30:51
          iTNC 530
INFO:     MAIN   START                      21.11.2001 07:30:51
          NC-SOFTWARE = 340420 01
INFO:     MAIN CYCLES                       21.11.2001 07:30:55
          CYCLE data are up to date
INFO:     MAIN CYCLES                       21.11.2001 07:30:55
          TCHPROBE data are up to date
ERR:      N-1 Power interruption            21.11.2001 07:31:02
Key:      0x01AE -> CE                      21.11.2001 07:31:15
Error:    P88 88 MPs are being read ...     21.11.2001 07:31:19
Key:      0x01F0 -> NC Start                21.11.2001 07:31:22
Key:      0x01F0 -> NC Start                21.11.2001 07:31:23
Key:      0x01F0 -> NC Start                21.11.2001 07:31:24
Key:      0x01F0 -> NC Start                21.11.2001 07:31:24
Key:      0x01F0 -> NC Start                21.11.2001 07:31:24
Key:      0x01F0 -> NC Start                21.11.2001 07:31:25
Key:      0x01C3 -> Auto                    21.11.2001 07:31:27
Key:      0x01F0 -> NC Start                21.11.2001 07:31:30
STIB:     ON                                21.11.2001 07:31:30
INFO:     MAIN PGM                          21.11.2001 07:31:30
          TNC:\STEFAN\GRAVUR.H
INFO:     MAIN LINE                         21.11.2001 07:31:30
          0
STIB:     OFF                               21.11.2001 07:31:31
INFO:     MAIN PGMEND                       21.11.2001 07:31:31
          01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
          00 05 00 08 00 00 00 38
          |                    |
          Byte 0               Byte 7

INFO:     MAIN PATH                         21.11.2001 07:31:32
          RUNBRKPGM = TNC:\STEFAN\GRAVUR.H
ERR:      N56 Limit switch X+               21.11.2001 07:31:32
Key:      0x01AE -> CE                      21.11.2001 07:31:43
```

The log can also be written to by the PLC for diagnostic purposes:

- 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.

Module 9275 Writing 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 detection:

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 job or submit job

Module 9276 Writing 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 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 detection:

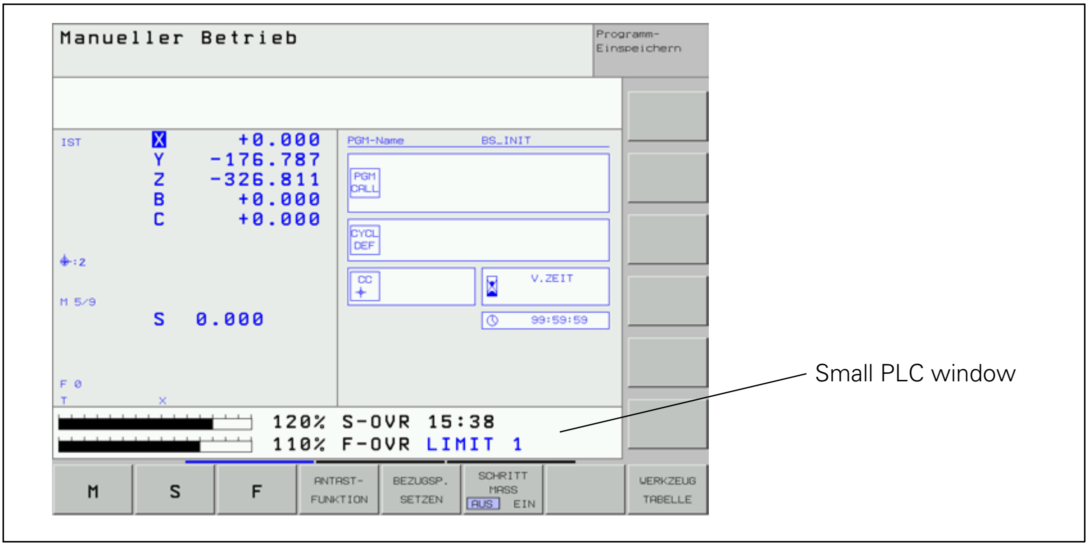
Marker	Value	Meaning
M4203	0	Entry was written
	1	Error code in W1022
W1022	1	Invalid priority
	2	Invalid identifier for operand name
	3	Invalid first operand address
	4	Sum of first operand address and number of operands invalid
	5	Address is not a word/double-word address
	12	No string end identifier
	20	Module was not called in a spawn job or submit job
	36	Entry in the log was shortened to 210 characters

7.2 PLC Window

7.2.1 Small PLC Window

The small PLC window is shown in the following operating modes:

- **`Manual Operation`**
- **`Electronic 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 7 – 60.

▶ 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 module through a CAN command!

Module 9080 Clearing the small PLC window
With this module you can clear the contents of the small PLC window.

Call:
CM 9080

Error detection:

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 Showing 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. In the event of error, no string is shown.

Column	**0**	**37**
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 37
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 detection:

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 column less than 0 or greater than 37 or incorrect string number or no end of the string or the last characters of the string cannot 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**		**0**																			**19**
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
```

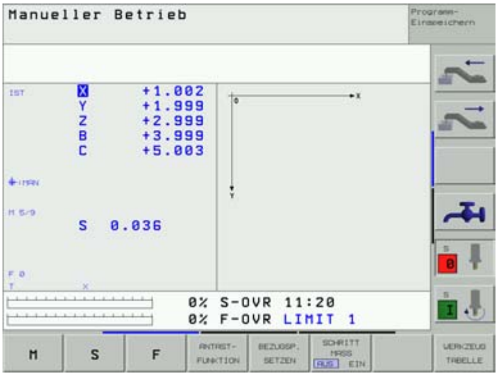
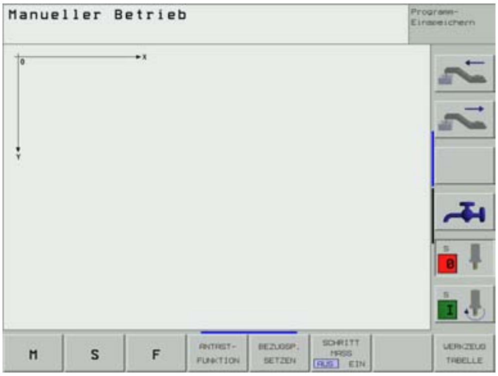
Error detection:

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

7.2.2 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 7 – 99.

▶ 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 7 – 91. The specified position refers to the lower left corner of the first character.

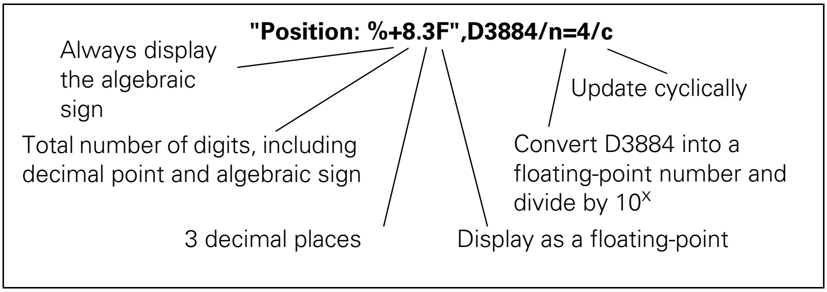
The content of the PLC window is defined in a screen mask – an ASCII file containing format instructions and special commands:

▶ 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 7 – 99.

Format instructions

Format instructions are enclosed in quotation marks (“ ”). Variables are transferred as parameters.

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
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 no. from 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 for 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 interp. 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 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 7 – 68.

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 number in inches.

`/e`
For B/W/D/M/S. Define the field length in the format string. The current contents of the variable 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.**

COLOUR=[f] or **COLOR=[f]**
Sets the foreground color. Value range for f: 1 to 14
Default setting: Color 11
The colors are defined in MP7367.
(See "Color Settings" on page7 – 60)

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.

Preset spacing:

Line	Column
SMALL	168 pixels
MEDIUM	2416 pixels
LARGE	4832 pixels

GRAPHICS= <fname> [/c]
Links a graphic into the window. With **POS, IPOS** or **ICPOS,** enter a position. The lower left corner is set to the current position. Graphics are created as *.DXF files in a CAD program, and are then converted with PLCdesign.
<fname> contains a file name with path, or the file name only. In this case the path in MP7230.3 is added (language for help files).
Switch **/c** : The graphic is cyclically refreshed.

TEXTFILE= <fname>
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 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 can therefore execute 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.
<fname>
Contains a file name with path, or the file name only. In this case the path in MP7230.3 is added (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.** Error number before the error text. Position in the error queue 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. Then the designated position is taken over as the actual 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 automatically closed.

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 for special functions a numerical value is expected, a mathematical expression can be written in integer arithmetic.

The operators and priority rules of the programming language C apply.

Available operations: +, –, *, /, %, &, |, ^.

The mathematical expressions may have the following variables:

Variable	Meaning
PAGE	Number of the current page, beginning with zero
XPOS	X position of the cursor pixel
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.

With the **/s = xxx** switch a field is produced in which no input is possible. 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. If necessary the current page is scrolled. Text between the input fields may no longer be displayable.

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	Deletes a displayed error message or the input field.
ENT	Takes the input value as the variable and sets the highlight on 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	Redisplays the original content of the field and sets the highlight on 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 put into the editing memory and the cursor is set on the end of the entry value.

Opening or erasing 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 7 – 99.

Module 9210 Opening or erasing screen mask for the PLC window
With this module you can activate or erase the display in the large PLC window.

Call:

PS	B/W/D/K/S<Erase no. of string memory/file name/PLC window>	
		–1: Delete PLC window
CM	9210	
PL	B/W/D	<Status / error>
		0: Mask opened / mask erased
		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 code word
`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 detection:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Invalid number of the status information

7.3 PLC Soft Keys

You can display your own soft keys through the PLC in all operating modes. You can create the soft keys with PLCdesign.

When a PLC soft key is pressed the NC enters the soft-key number in W302 (horizontal soft-key row) or W304 (vertical soft-key row). 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.

7.3.1 Soft-Key Project File

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 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.
PULSE, CHECK and RADIO soft keys are not reported via W304 to the PLC.
With Module 9205, you can also select a word address other than W304.
BLANK soft keys are not reported to the PLC.

Note

The states of the assigned operands of the CHECK and RADIO soft keys and the display are checked cyclically. If required, the display 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 formats *.BMX and *.BMP. 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
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 7 – 102.
ENDSKMENU	End of the definition of a soft-key menu.
NODE	Soft key jumps to a submenu. Is confirmed via 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 7 – 102.
BACK	Soft key jumps to a submenu. Is confirmed via 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 7 – 102.
BLANK	Vacant soft key. You can also specify a soft-key name.
ACTION	Function soft key. Is confirmed via W304 to the PLC. The soft-key name must be indicated. Also note the additional parameters for this keyword on page 7 – 102.
PULSE	The soft key is reported to the PLC via 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 7 – 102.

Entry in *.SPJ	Meaning
`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 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 7 – 102.
`RADIO`	From any group of these soft-key types, no more than one soft key can be pressed. The soft key is reported to the PLC via 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 7 – 102.
`END`	Closes an open pop-up menu. The soft key is reported to the PLC via W304 for the duration that the key is pressed. Also note the additional parameters for this keyword on page 7 – 102.
`#include`	An additional soft-key project file can be included. The name and path of this file must be entered.

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**	Depending on the marker status, the soft key is either locked (marker = 1) or enabled (marker = 0).
STATUS:<Marker or word>	**NODE, BACK, END, ACTION, PULSE, CHECK, RADIO**	An operand is assigned to the soft key (in addition to W302/W304). When the soft key is pressed, the **marker** is set or the soft-key number is entered in the **word.**
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.
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.
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**	Identifies the first soft key of each group. The parameter **STATUS:<Word>** can also be entered. Then the number of the soft key within the group is given in the **Word** (beginning with 0).

Example of a soft-key project file

Soft-key number (confirmed to the PLC):

```
     PLC SOFT-KEY 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

```
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
ACTION MAG_CW_SK
BACK BACK_SK
ENDSKMENU
```

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 large PLC window with mask file

		Set	Reset
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, otherwise transfer 0 or “ ”
PS	B/W/D/K	<Number of PLC soft-key root menu> –1: First menu found in the SYS file
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> –1: Error code in W1022

Error detection:

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 detection:

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 Setting the word for acknowledgment of PLC soft keys

Module 9205 can be used to define another word, in addition to 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 detection:

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 setting 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 detection:

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 detection:

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 detection:

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

7.3.2 Compatibility to 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 detection:

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 detection:

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 detection:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Transfer parameter out of value range

7.4 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 422 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 control panel are available on connector X46. You must evaluate the inputs and outputs in the PLC.

7.4.1 iTNC Keyboard Unit

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 page7 – 117)

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

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 detection:

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 Disabling 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 detection:

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 detection:

Marker	Value	Meaning
M4203	0	Disabling was canceled
	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: 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

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 detection:

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 value> maximum value

Error detection:

Marker	Value	Meaning
M4203	0	Disabling was canceled
	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 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
CM	9186	

Error detection:

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 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
```

Codes for keystroke simulation

Code	Key	Group
$00	No key	
$08	BACKSPACE	ASCII
$0A	RET	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

Code	Key	Group
$3A	:	ASCII
$3B	;	ASCII
$3C	<	ASCII
$3D	=	ASCII
$3E	>	ASCII
$3F	?	ASCII
$41	A	ASCII
$42	B	ASCII
$43	C	ASCII
$44	D	ASCII
$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

Code	Key	Group
$57	W	ASCII
$58	X	ASCII
$59	Y	ASCII
$5A	Z	ASCII
$5E	^	ASCII
$180	0 soft key	Soft key
$181	1 soft key	Soft key
$182	2 soft key	Soft key
$183	3 soft key	Soft key
$184	4 soft key	Soft key
$185	5 soft key	Soft key
$186	6 soft key	Soft key
$187	7 soft key	Soft key
$19C	FBACK	Soft key
$19D	FNEXT	Soft key
$19E	FNEXT-UP	Soft key
$1A0	C-UP	Cursor
$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

Code	Key	Group
$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
$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	

7.4.2 Machine Operating Panel

On socket 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.

MP7680 **Machine parameter with multiple function**
Format: %xxxxxxxxxxxxxx
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 8 correspond to axes 1 to 9: 0: Do not move axis 1: Move axis	PLC	PLC

		Set	Reset
W1048	**Manual traverse in negative direction** Bits 0 to 8 correspond to axes 1 to 9: 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

7.5 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:

- Choose with MP7224.1 the files that you want to protect.
  Protected files are displayed in the file overview with the color defined in MP7354.1 or MP7355.1.

MP7224.0 Disabling soft keys for file types

Format: %xxxxxxxx

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 – HELP files *.HLP
Bit 7 – Point tables *.PNT
0: Do not disable
1: Disable

MP7224.1 Protecting file types

Format: %xxxxxxxx

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 – HELP files *.HLP
Bit 7 – Point tables *.PNT
0: Do not protect
1: Protect

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 Selecting 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 detection:

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

		Set	Reset
W1018	**Number of files opened by the PLC**	NC	NC
W1020	**Number of open files**	NC	NC

7.5.1 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 7 – 196.

With **FN17: SYSWRITE** and **FN18: SYSREAD** you can read and overwrite values in the datum table (OEM cycles). See pages 8 – 33 and 8 – 41.

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)

7.5.2 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 page 8 – 4)
- In the **PLC:\PROTO** directory, create a table with the extension .TAB.

If you have not yet defined prototypes, a standard prototype will be supplied. 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 = alphanumerical input.
- **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.

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 Reading a field out of a table
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 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, result>
CM	9245	

Error detection:

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	Module could not read from the table
	20	Module was not called in a spawn job or submit job
	29	The opened file is not a table with the extension .TAB or .P
	30	Column name not found

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 detection:

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.

The module transfers 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, contents to be written>
CM	9246	

Error detection:

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.

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	<Numerical value to be written>

CM 9256

Error detection:

Marker	Value	Meaning
M4203	0	Field was written to
	1	Error code in W1022
W1022		See Module 9246

Module 9247 Searching 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
+,–,*,/	Arithmetical 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 detection:

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

Starting 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 with into the original table 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 Starting 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
CM	9250	

Error detection:

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
	7	The module could not read from the table or open the temporary file
	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

		Set	Reset
M4159	**PLC editor: END key or soft key pressed**	NC	NC/PLC

Module 9251 Ending 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 detection:

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 Positioning 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 detection:

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)

7.5.3 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 Opening 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 in 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)
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 detection:

Marker	Value	Meaning
M4203	0	File was opened
	1	Error code in W1022
W1022	1	Inadmissible mode
	3	Incorrect string number
	7	File could not be opened
	20	Module was not called in a submit job or spawn job

Module 9241 Closing 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 detection:

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 submit job or spawn 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 the 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 at 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 detection:

Marker	Value	Meaning
M4203	0	Cursor was positioned
	1	Error code in W1022
W1022	1	Inadmissible mode
	2	Incorrect file handle
	7	File system error
	20	Module was not called in a spawn job or submit job

Module 9243 Reading from a file line by line

To read from a table, use Module 9245.

Open the file with Module 9240.

With Module 9243 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); 126 characters at most.

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 detection:

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 Writing 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 amount 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 following 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 detection:

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

7.6 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 page 7 – 124)

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.
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, 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.

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 7 – 124.

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.

		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

Module 9090 Selection of 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 job or submit job 2: Call during running NC program 3: No pallet table selected in full sequence 4: Line does not exist

Module 9281 Selection of 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 job 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 execute 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.

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 8 – 24.

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 detection:

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 8 – 24.

7.7 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

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 interpolation factor specifies the traverse per handwheel revolution:

- Choose the **HANDWHEEL** operating mode and enter an interpolation 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.
- With MP7641, specify whether the interpolation factor is entered directly through the TNC keyboard or via PLC Module 9036.

Interpolation factor	Traverse distance per revolution [mm]	Effective beginning from rapid traverse: MP1010.x [mm/min]
0	20	12 000
1	10	6000
2	5	3000
3	2.5	1500
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

You can choose a larger input step for the traverse distance per rotation than that calculated by the NC:

- In MP7670.x, enter an interpolation factor.
- In MP7645.x, enter an initialization parameter for the handwheel. The parameters are evaluated by the HRA 110 and HR 410.

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

MP7641 **Entry of the interpolation factor**
Input: 0: Through iTNC keyboard
1: Through PLC Module 9036

MP7650 **Handwheel counting direction for each axis**
Format: %xxxxxxxxx
Input: Bits 0 to 8 represent axes 1 to 9
0: Negative counting direction
1: Positive counting direction

MP7660 **Threshold sensitivity for electronic handwheel**
Input: 0 to 65 535 [increments]

MP7670 **Interpolation factor for handwheel**
Input: 0 to 10
MP7670.0 Interpolation factor for low speed
MP7670.1 Interpolation factor for medium speed (only HR 410)
MP7670.2 Interpolation factor for high speed (only HR 410)

		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 interpolation 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 interpolation key X	0 to 10
1	Handwheel interpolation key Y	0 to 10
2	Handwheel interpolation key Z	0 to 10
3	Handwheel interpolation key IV (MP410.3)	0 to 10
4	Handwheel interpolation key V (MP410.4)	0 to 10
5	Handwheel interpolation of all axes	0 to 10
6	Select the handwheel axis (not for HRA 110)	0 to 8 axes 1 to 9
10	See "Incremental Jog Positioning"	
11	Handwheel interpolation of axis 1	0 to 10
12	Handwheel interpolation of axis 2	0 to 10
13	Handwheel interpolation of axis 3	0 to 10
14	Handwheel interpolation of axis 4	0 to 10
15	Handwheel interpolation of axis 5	0 to 10
16	Handwheel interpolation of axis 6	0 to 10
17	Handwheel interpolation of axis 7	0 to 10
18	Handwheel interpolation of axis 8	0 to 10
19	Handwheel interpolation 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 detection:

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

7.7.1 HR 130 Panel-Mounted Handwheel

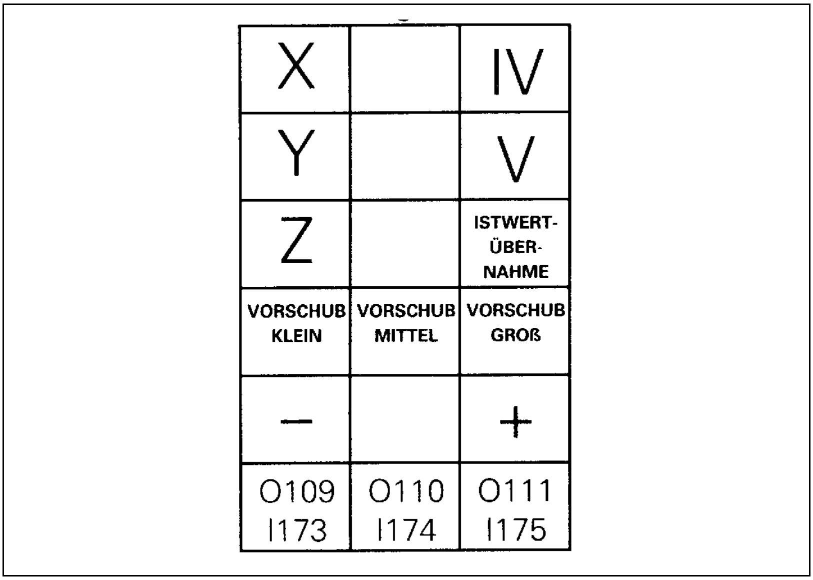
- Enter MP7640 = 2 (HR 130)

When the axis keys are pressed, the associated cursor and the handwheel symbol are displayed simultaneously.

7.7.2 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



With the exception of the function keys A, B and C, all keys are evaluated by the NC.

- With MP7670.x, select the interpolation factors for low, medium and high speed.
- With MP7671.x, define the values for low, medium and high speed. The speed is entered as a percentage of the manual feed rate (MP1020.x).

Evaluation of the keys by 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 interpolation.

With W766 you can influence the feed rate of the axis direction keys.

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 **Interpolation factor for handwheel**

Input: 0 to 10

MP7670.0 Interpolation factor for low speed

MP7670.1 Interpolation factor for medium speed (only HR 410)

MP7670.2 Interpolation factor for high speed (only HR 410)

MP7671 **Handwheel feed rate in the Handwheel operating mode with HR 410**

Input: 0 to 1000 [% of MP1020]

MP7671.0 Low speed

MP7671.1 Medium speed (only HR 410)

MP7671.2 High speed (only HR 410)

7.7.3 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 interpolation factor. See "HRA 110 Handwheel Adapter" on page 3 – 52. 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 2	3rd handwheel axis IV
		Position 3	3rd handwheel axis V
	1:	Switch position 1 (at the left stop)	3rd handwheel axis X
		Switch position 2	3rd handwheel axis Y
		Position 3	3rd handwheel axis Z
		Position 4	3rd handwheel axis IV
		Position 5	3rd handwheel axis V
	2:	Position 3	3rd handwheel axis Z
		Position 4	3rd handwheel axis IV
		Position 5	3rd handwheel axis V
MP7645.1	Fixed assignment of a third handwheel if MP7645.2 = 1		
Input:	1: Axis X 2: Axis Y 4: Axis Z 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 interpolation 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

7.8 PLC Inputs/Outputs

The MC 422 provides you with digital inputs/outputs and analog inputs/outputs for the PLC. If the available number of I/O is not enough, you can add up to four PL 4xx B input/output units.

	MC 422				**PLC input/output unit**	
	X9	**X41**	**X42**	**X48**	**PL 410 B**	**PL 405 B**
Switching inputs 24 Vdc	–	–	56	–	64	32
Switching outputs 24 Vdc	–	31	–	–	31	15
Analog inputs 10 Vdc	–	–	–	3	(4)	–
Inputs for Pt 100 thermistors	–	–	–	3	(4)	–
Analog outputs 10 Vdc	12[a])	–	–	–	–	–
Control-is-ready output	–	2	–	–	1	1
Control-is-ready input	–	–	2	–	–	–

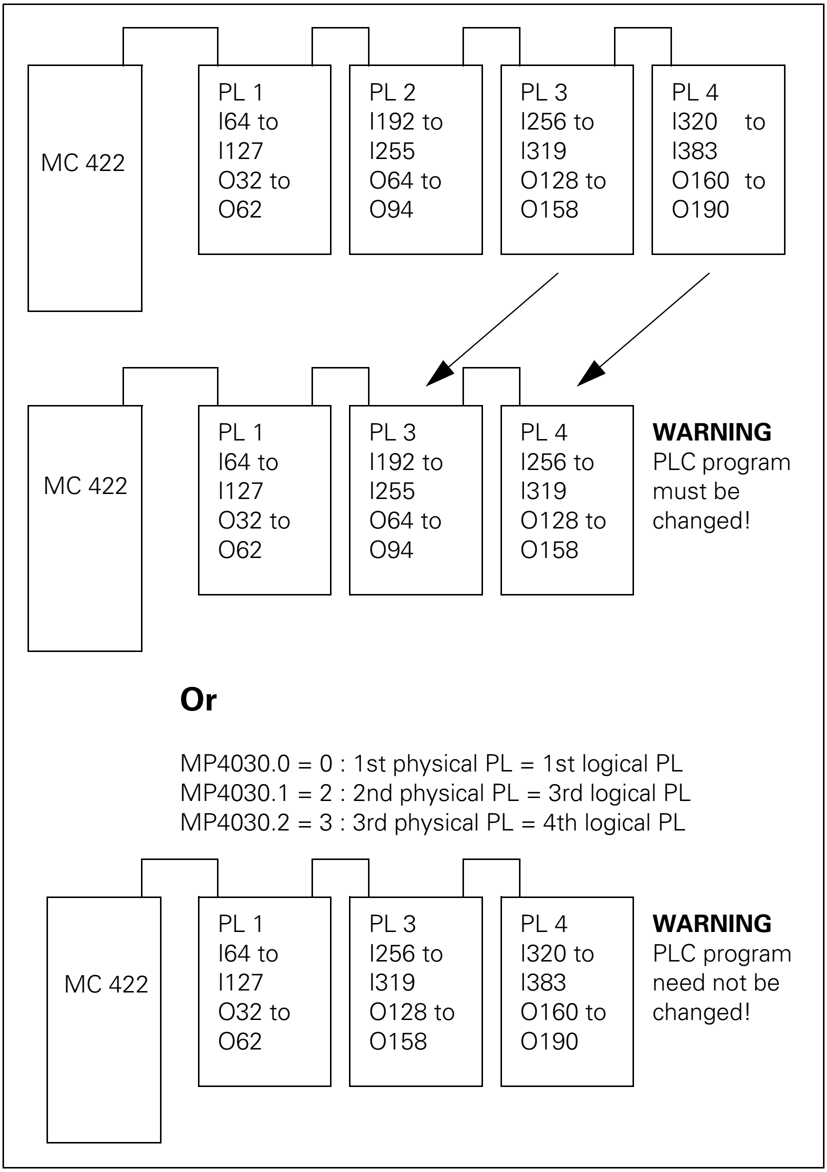
a. You need one analog output for each analog axis.

To interrogate and set the inputs and outputs of the PLC I/O unit you need PLC modules.

PL assignment

Up to four PL 4xx B PLC I/O units can be connected. The first PL is connected to the MC 422, 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 422).

Example:



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

Diagnosis of the PL

Module 9007 Diagnostic information of the PL

Module 9007 can ascertain diagnostic information of the PL 4xx B. To save computing time, refrain from continuously calling this module.

Call:

PS B/W/D/K <Number of the logical PL>
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 422
 4: Further PLs on this PL
 5: PL is a PL 410 B
CM 9007
PL B/W/D/K <Diagnostic information>
 0: Not available
 1: Available
 0 to 4: Number of PLs

Error detection:

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 job or submit job

7.8.1 24 Vdc Switching Input/Outputs

In PLC addresses you can find the current conditions 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.

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.

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 detection:

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 job or submit job

Module 9008 Reading specific 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 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 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	<Bits 0...31 = inputs 0..31>
PS	D/K	<Bits 0...31 = inputs 32...63>
CM	9008	

Error detection:

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 job or submit job

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.

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 detection:

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 job or submit job

Module 9009 Update certain outputs of a PLC input/output unit

Module 9009 overwrites certain 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 9005. 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 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 detection:

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 job or submit job

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 detection:

Marker	Value	Meaning
M4203	0	No error
	1	Invalid transfer parameter

7.8.2 Analog Inputs

Socket X48 of the MC 422 provides ±10-Vdc analog inputs and analog inputs for Pt 100 temperature resistors. See “Analog Input” on page 3 – 33.

The PLC input/output unit is available in a version with additional analog inputs. See “Overview of Components” on page 2 – 3.

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 Reading in analog inputs

Module 9003 reads the current value of the specified analog input, regardless of whether it is actually connected.

Value range ±10 Vdc input:	–10 to +10, at a resolution of 10 mV –100 to +100, at a resolution of 100 mV
Value range 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 only be called in the cyclic PLC program.

Call:

PS	B/W/D/K	<Number of analog input> 0 to 7: Analog inputs X15 to X22 on first unit 8 to 15: Analog inputs X15 to X22 on second unit 16 to 23: Analog inputs X15 to X22 on third unit 24 to 31: Analog inputs X15 to X22 on fourth unit 64 to 66: ±10 Vdc 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 Nr. 64 to 69: Natural number with the unit 0.01 V or 0.1 °C

Error detection:

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

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

7.8.3 Analog Outputs

You can drive analog outputs 1 to 12 at sockets 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 12 (X9)
PS   B/W/D/K   <Analog voltage in mV>
CM   9130
```

Error detection:

Marker	Value	Meaning
M4203	0	Analog voltage was output
	1	Error code in W1022
W1022	1	Invalid analog output
	2	Disabled analog output

7.9 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).

With marker M4579 you can interrogate the current state.

With Module 9036 you can limit the jog increment.

You can ascertain the current jog increment with Module 9035.

With Module 9186 you can switch the incremental jog function on and off through the PLC.

		Set	Reset
M4579	**INCREMENT OFF/ON soft key**	NC	NC

In the **Electronic Handwheel** mode you can enable the incremental jog function with M4572. The "interpolation factor" message is displayed in addition to the "jog increment" message.

- Activate incremental jog positioning with W1050/W1052 and set the corresponding bit. Activation of incremental jog positioning is linked with the axis direction keys.

		Set	Reset
M4572	**Enabling incremental jog positioning**	PLC	PLC
W1050	**Incremental jog positioning in positive direction** Bits 0 to 8 represent axes 1 to 9 0: Not active 1: Active	PLC	PLC
W1052	**Incremental jog positioning in negative direction** Bits 0 to 8 represent axes 1 to 9 0: Not active 1: Active	PLC	PLC

Module 9036 Writing status information

The information to be overwritten is designated with a transferred number.

Number of the status information	Function	Value
0 to 6	See "Handwheel"	
10	Jog increment limiting	0.0001 to 50 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 detection:

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 Reading status information

Call:

PS	B/W/D/K	<26>
CM	9035	
PL	B/W/D	<Jog increment>

Error detection:

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

7.10 Hirth Coupling

The 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.

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 the 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 [°]

7.11 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 6 – 50) 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 detection:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing strobe or control is active
	24	Module was called in a spawn job or submit job

7.12 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 632: 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
1: Touch probe with infrared transmission

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.

Module 9135 Switch on 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 detection:

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 on 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.

Call:

```
PS      B/W/D/K   <Touch probe state>
                  0: Switch off touch probe
                  1: Switch on touch probe
CM      9136
```

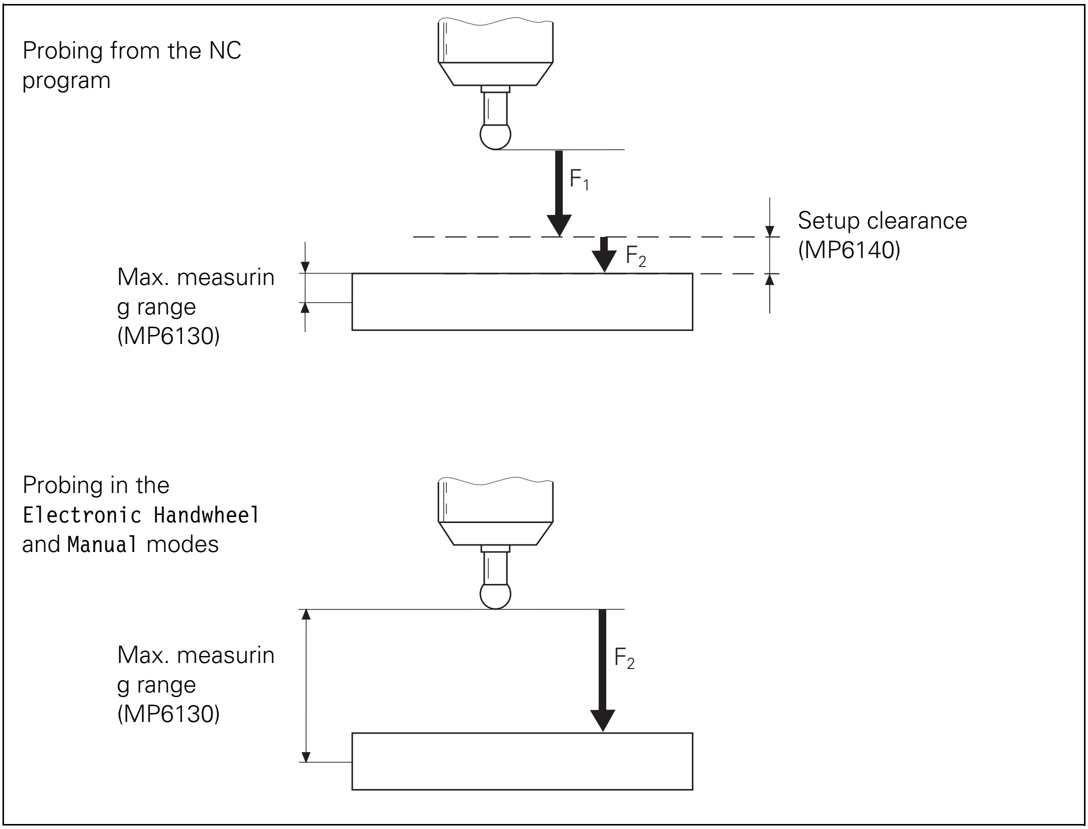
Error detection:

Marker	Value	Meaning
M4203	0	Touch probe on or off
	1	Error code in W1022
W1022	1	Invalid touch probe state

7.12.1 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.



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.

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

The NC takes over control of the probing process. Certain conditions are indicated in M4050 to M4054.

If you set M4056, the NC stops the machine in all operating modes as soon as the stylus is deflected. The maximum feed rate is limited to the value specified in MP6150. If M4056 is set and the infrared 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 the probing function has been started.

HEIDENHAIN recommends:

- Set M4056 as soon as the touch probe is in the spindle.
- 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 = 1, you can use soft keys to take the effective length and effective radius over into the tool table.
- In the **Manual** and **Electronic Handwheel** operating modes, enter the tool number in the menu for touch probe calibration.

The iTNC can save the calibration data for up to three touch probes at once:

- Set MP7490 bit 2.
- Use the traverse range switching function to activate the current data with M4574/M4575.

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.

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.

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 use the TS 632 infrared touch probe, you must orient the touch probe before the measuring process in order to align the touch probe and the receiver unit. Define an M function for automatically orienting the probe to a specific 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 if MP6163 < (current spindle angle – MP6162)

		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**	NC	NC
M4055	**Enable the probing process**	NC	PLC
M4056	**NC stop in all operating modes if stylus is deflected**	PLC	PLC
M4574	**Select the traverse range (with M4575)**	PLC	PLC
M4575	**Select the traverse range (with M4574)**	PLC	PLC

MP6120 **Probing feed rate**
Input: 1 to 3000 [mm/min]

MP6130 **Maximum measuring range**
Input: 0.001 to 99 999.9999 [mm]

MP6140 **Setup clearance over measuring point**
Input: 0.001 to 99 999.9999 [mm]

MP6150 **Rapid traverse in probing cycle**
Input: 10 to 20 000 [mm/min]

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 [°]

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

MP7411 **Tool data in the touch probe block**
Format: %xx
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

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

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

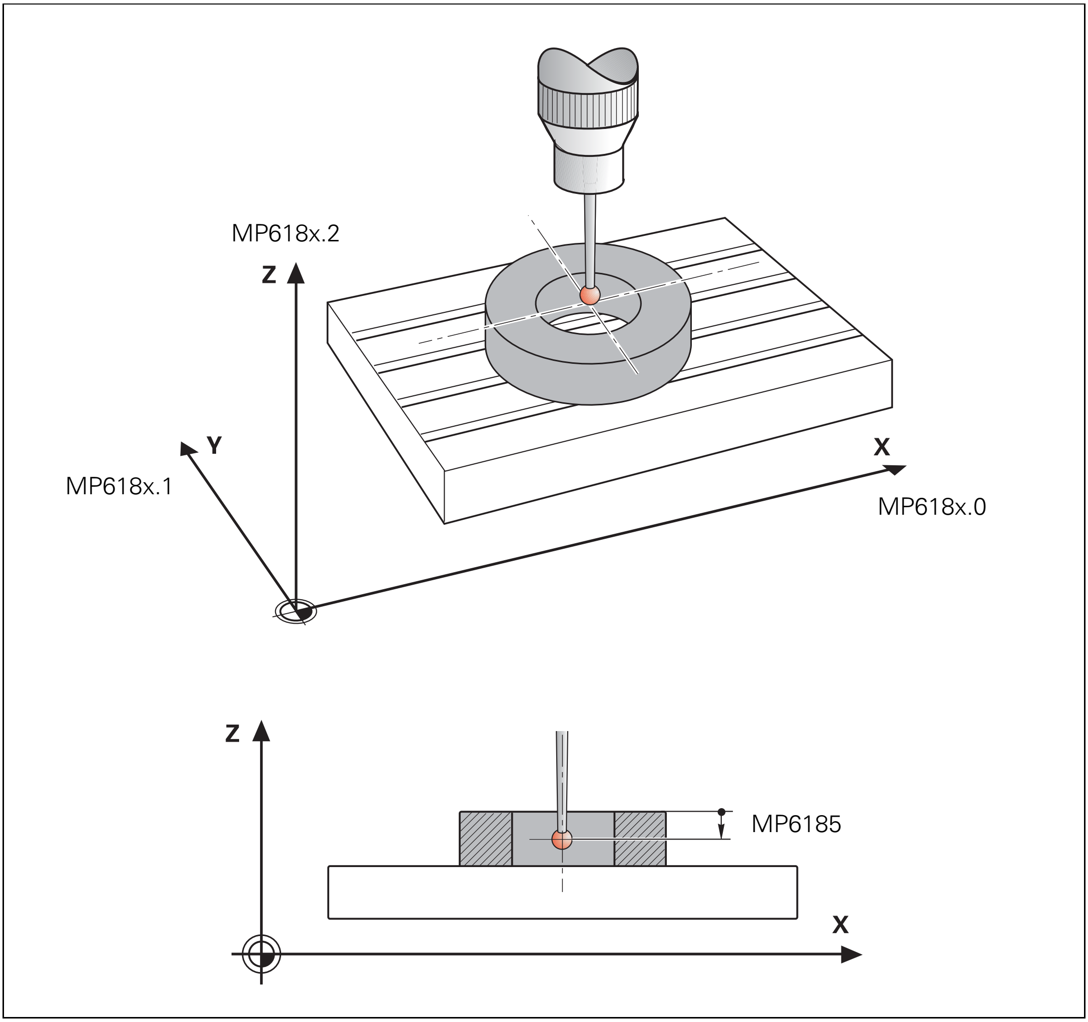
Error detection:

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 job or submit job

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` 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	**Opening 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.
- In MP6171 enter 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]

7.12.2 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 7 – 73.

File names of the individual print masks

- Calibration for length, touch-trigger probe: TSLCAL.A
- Calibration for radius, touch-trigger 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.
- The special commands **MM** and **INCH** switch the display to mm or inches. The commands affect only number types that allow an inch representation.

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 no. from real-time clock

Results or input from the manual measuring cycles in the control data		
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 probe 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	Datum
BEZA	string	String datum axis

Datum at corner, circle, 4 holes, 3 holes on a circle		
Name	**Format type**	**Description**
BZ_HA	double	Datum in reference axis
BZ_NA	double	Datum in minor axis
LKALBEZ	double	Datum 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;
```

7.12.3 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 ****************************";
```

7.12.4 Tool Measurement

With the HEIDENHAIN TT 130 touch probe you can measure and inspect tools. HEIDENHAIN provides standard cycles for automatic tool measurement and calibration of the TT 130 (see the Touch Probe Cycles User's Manual).

Technical prerequisites

You need:

- TT 130
- 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 130 must be mounted and interfaced:

- With MP6500 bit 0, enable the cycles for tool measurement.

MP6500	**Tool measurement with TT 130**
Format:	%xxxxxxxxxxxx
Input:	Bit 0 – 0: Cycles for tool measurement disabled 1: Cycles for tool measurement not disabled

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 130**
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.
- 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 130**
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 130.

▶ 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 130 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 130**
Input:	10 to 20 000 [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 saves 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 130**
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's 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 130 probe 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 130 probe 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 130 probe 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 130 probe 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 [rpm]
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 rpm:

- 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 so-called measuring tolerance.

$v = \text{measuring tolerance} \cdot n$

v: Probing feed rate [m/min]
Measuring tolerance: Measuring tolerance [mm] from MP6510.0 depending on MP6507
n: Speed [rpm]

- 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.

$$\text{Meßtoleranz} = \frac{r}{5\,[\text{mm}]} \cdot \text{MP6510.0}$$

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 130**
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 3000 [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 [rpm] 0: 1000 [rpm]

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 130**
Format:	%xxxxxxxxxxxxxx
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

Warning

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 130**
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 routine 0: After "tool checking" the tool table is changed 1: After "tool checking" the tool table is not changed

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

7.13 Special Functions for Laser Cutting Machines

You can activate special functions to interface the iTNC to laser cutting machines and water jet machines.

7.13.1 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 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.

7.13.2 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.

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

7.13.3 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

Warning

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

7.14 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 6 – 26. 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.

7.14.1 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 7 – 121.
- 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, 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 7 – 121.
- 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 MP7261.1-3 = 0. If you use multiple tool magazines, See "Managing multiple tool magazines" on page 7 – 212.
  - If MP72610.3 = 0, no pocket table is generated.

With Modules 9092, 9093, 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.

MP7260 **Number of tools in the tool table**
Input: 0 to 30 000

MP7261.0-3 **Number of pockets in the tool magazine 1 to 4**
Input: 0 to 254

MP7263 **Pocket table**
Format: %xx
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

MP7266 **Elements of the tool table**
Input: 0 = no display
1 to 99 = position in the tool table

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 (MILL=cutter/ DRILL=drill)	TYPE	5
MP7266.25	Tool material	TMAT	16
MP7266.26	Cutting data table	CDT	16
MP7266.27	PLC value	PLC-VAL	11
MP7266.28	Center misalignment in reference axis	CAL-OF1	11
MP7266.29	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

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)

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 Searching 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)

***.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 detection:

Marker	Value	Meaning
M4203	0	No error
	1	See above for errors

Module 9093 Read data from tables selected for program (.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 detection:

Marker	Value	Meaning
M4203	0	No error
	1	See above for errors

Module 9094 Writing 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 detection:

Marker	Value	Meaning
M4203	0	No error
	1	See above for errors

Module 9096 Deletion of a line in 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 detection:

Marker	Value	Meaning
M4203	0	The line was deleted
	1	Error code in W1022
W1022	2	Invalid pocket or tool number
	21	Module was not called in a submit job or spawn job
	24	File error

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.
- Create the file *.TCR with the following keywords.

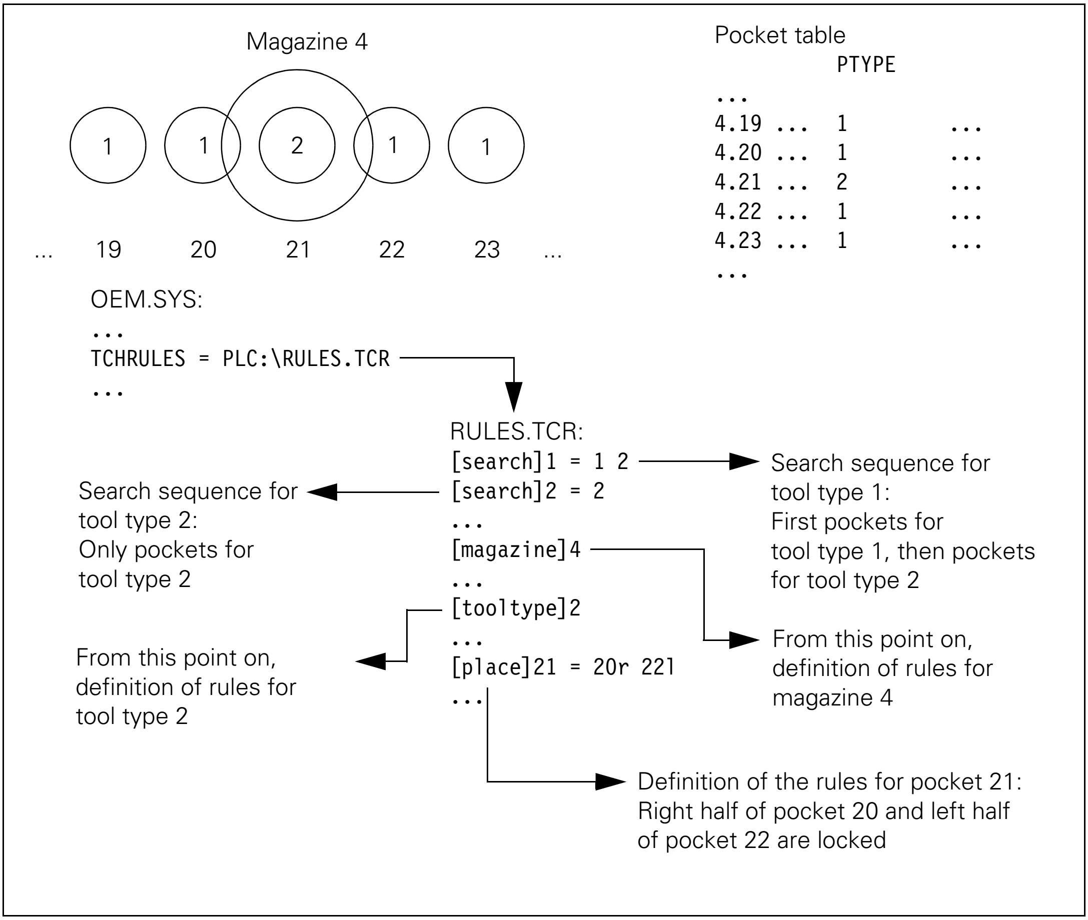
Keyword	Meaning
[search]a = a b ...	Definition of the search sequence for the tool type **a.** The tool type named first (here **a**) is searched for first, then the next tool type (separated by a space) (here **b**). You may only enter tool types in whose pockets tools of the type **a** may be placed! Example: **[search]1 = 1 2**
[magazine]a	All the following rules apply to the tool magazine **a.** Example: **[magazine]4**
[tooltype]a	All the following rules apply to the tool type **a.** 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

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

Module 9340 searches 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.

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
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: No selection list available –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 detection:

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 detection:

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 free 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 detection:

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

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 detection:

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

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 detection:

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 submit job or spawn job
	30	Tool not found
	36	File error in pocket table

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 or 9306, then releases it again. As long as the pocket table is locked, a tool change is not sent from the NC to the PLC. Instead, the error message **`tool preselection is running`** appears. As soon as the pocket table has been released, the tool change is output from the NC to the PLC.

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 submit job or spawn job 5: Module was called during the NC program run

Error detection:

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 job 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 change the tools in the pocket table. 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 detection:

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
	30	No valid tool in the original pocket

Managing multiple tool magazines

Up to four different tool magazines can be managed in the pocket table. In the pocket table the tool magazines are listed from 1 to 4, 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 4 in MP7261.0 to MP7261.3.

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 4: Number of the tool magazine**	NC	NC

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 detection:

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 detection:

Marker	Value	Meaning
M4203	0	Vacant pocket was found
	1	Error code in W1022
W1022	1	Invalid pocket number
	2	Invalid tool magazine number
	20	Module was not called in a spawn job 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.

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 detection:

Marker	Value	Meaning
M4203	0	Pocket has been exchanged
	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
	30	No valid tool in the original pocket
	36	Error in file handling

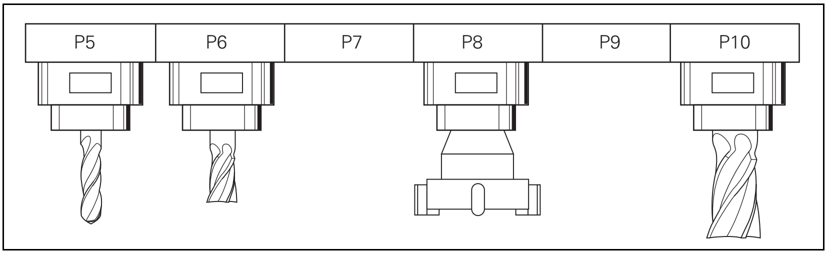
Special tools

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, activate the automatic insertion of the replacement tool after expiration of the tool age (**TIME1** or **TIME2**). With M102, deactivate the insertion. The tool is not changed immediately after expiration of the tool life, but rather it varies by a few NC blocks depending on the microprocessor load. In order to also be able to activate the automatic insertion of the replacement tool with M128, 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 8 – 24.

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 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** und **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, 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**
Format: %xxxxxxxxxxxxxx
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:

- With Module 9091 you can determine the line number of a tool in the tool table.

MP7262 **Maximum tool index number for indexed tools**
Input: 0 to 9

		Set	Reset
W266	**Index number of a programmed indexed tool**	NC	NC

Module 9091 Finding 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 number>
CM    9091
PL    B/W/D     <Line number>
```

Error detection:

Marker	Value	Meaning
M4203	0	Line number was found
	1	Error code in W1022
W1022	2	Invalid value for tool or tool index number
	20	Module was not called in a submit job or spawn job
	29	Tool table (TOOL.T) not found
	30	Tool number not found
	32	Tool index number not found

7.14.2 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 [rpm]

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 TYPE 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.

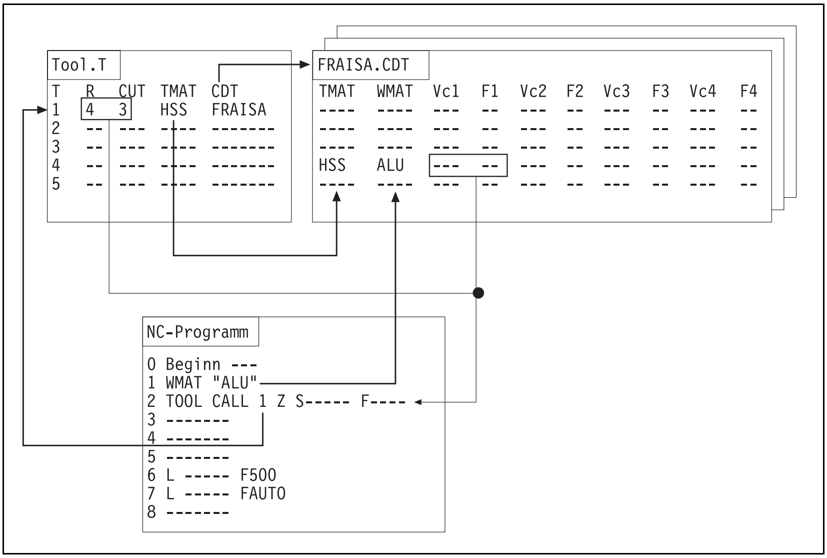
Calculation of cutting data

▶ 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



7.14.3 Automatic Tool Recognition

Automatic tool identification is possible with the Balluff tool identification system (BIS).

Please contact HEIDENHAIN for further information.

7.14.4 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 when the axis is stationary!

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.

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 8 – 24.
The tool geometry is not taken over then. You must program a TOOL CALL at another place to update the tool data.

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 8 – 33.

- 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 8 – 44.

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).

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 7 – 39). 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 cycle call, i.e. defined values remain only locally effective.

```
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 * - T-NEW POSITION AXIS 7 (FKA)
```

```
10 FN 18: SYSREAD Q6 = ID2000 NR70 IDX98; Read W98
11 * - POSITION AXIS 8 (FK)
12 FN 18: SYSREAD Q7 = ID2000 NR80 IDX98; Read D98
13 * - TC Z SAFETY CLEARANCE
14 L Z+0 R0 F MAX M91
15 * - ALIGN SPINDLE M21
16 M21
17 FN 18: SYSREAD Q8 = ID1000 NR4210 IDX11 ; Read MP4210.11
18 FN 18: SYSREAD Q9 = ID1000 NR4210 IDX12 ; Read MP4210.12
19 FN 18: SYSREAD Q10 = ID230 NR3 IDX2 ; Pos. software limit switch Y
20 * - OPEN TRAVERSE RANGE Y
21 Q9 = Q6 + 0.5
22 FN 17: SYSWRITE ID230 NR3 IDX2 = +Q9
23 * - Z APPROACH TC AREA
24 L Y+Q6 R0 F MAX M91
25 L Z+Q7 R0 F MAX M91
26 * - TOOL MACRO ACTIVE ?
27 FN 20: WAIT FOR SYNC M1999==1
28 * - WRITE TOOL CALL DATA ->PLC
29 FN 9: IF +Q2 EQU +0 GOTO LBL 11
30 FN 9: IF +Q2 EQU +1 GOTO LBL 12
31 FN 9: IF +Q2 EQU +2 GOTO LBL 13
32 FN 9: IF +Q2 EQU +3 GOTO LBL 14
33 FN 9: IF +Q2 EQU +4 GOTO LBL 15
34 FN 9: IF +Q2 EQU +5 GOTO LBL 16
35 TOOL CALL Q1 SQ3 DL+Q4 DR+Q5
36 FN 9: If +0 EQU +0 GOTO LBL 17
37 LBL 11
38 TOOL CALL Q1 X SQ3 DL+Q4 DR+Q5
39 FN 9: If +0 EQU +0 GOTO LBL 17
40 LBL 12
41 TOOL CALL Q1 Y SQ3 DL+Q4 DR+Q5
42 FN 9: If +0 EQU +0 GOTO LBL 17
43 LBL 13
44 TOOL CALL Q1 Z SQ3 DL+Q4 DR+Q5
45 FN 9: If +0 EQU +0 GOTO LBL 17
46 LBL 14
47 TOOL CALL Q1 U SQ3 DL+Q4 DR+Q5
48 FN 9: If +0 EQU +0 GOTO LBL 17
49 LBL 15
50 TOOL CALL Q1 V SQ3 DL+Q4 DR+Q5
51 FN 9: If +0 EQU +0 GOTO LBL 17
52 LBL 16
53 TOOL CALL Q1 W SQ3 DL+Q4 DR+Q5
54 LBL 17
55 * - Z LEAVE TC AREA
56 L Z+0 R0 F MAX M91
57 L Y+0 R0 F MAX M91
58 * - CLOSE TRAVERSE RANGE Y
59 FN 17: SYSWRITE ID230 NR3 IDX2 = +Q8
60 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.

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: Output of the pocket number and tool number only when tool number changes 4: Output of the 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 the pocket number and tool number 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: Output of the pocket number and tool number only when tool number changes 4: Output of the pocket number and tool number 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

		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: A normal tool follows a normal tool (N → N) Manual tool follows a manual tool (M → M) Special tool follows a special tool (S → S), when M4541 = 0 1: Special tool follows a manual tool (M → S), if M4541 = 1 Special tool follows a special tool (S → S), when 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:

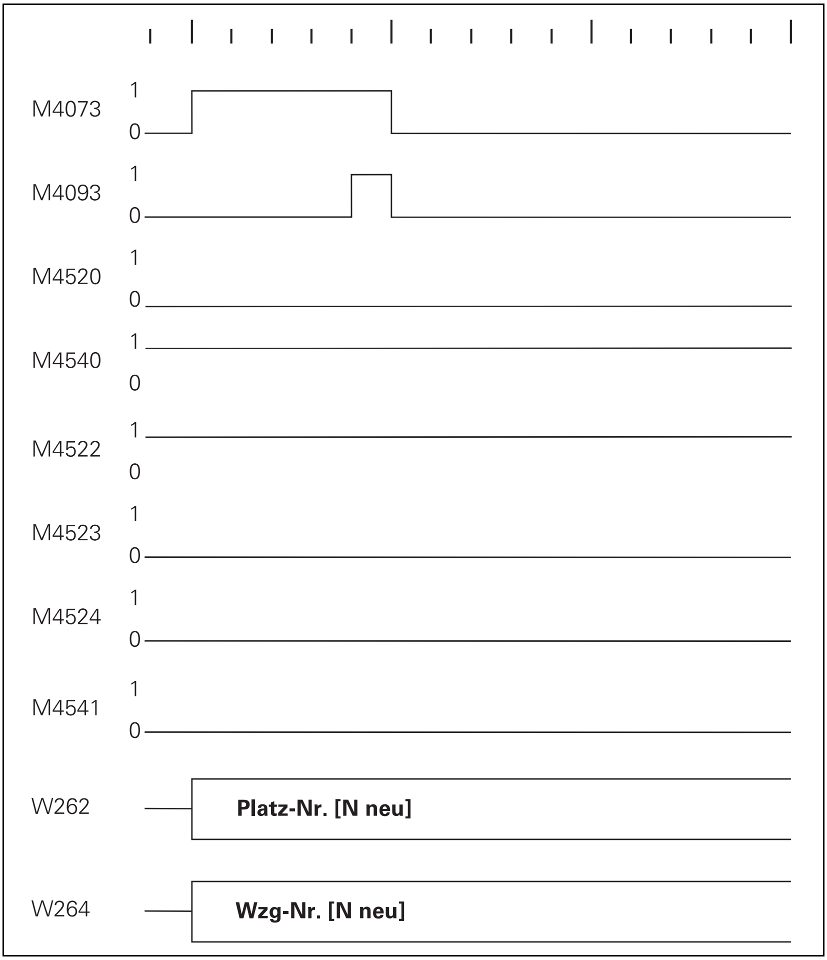
- Define the sequence in MP7481.x. In this case set M4540 = 0.
- Define the sequence in M4540. In this case set MP7481.x = %0000.

You must evaluate and acknowledge both pocket or tool numbers.

MP7481	**Sequence for new and returned tool when changing tools**
Format:	%xxxx
	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
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

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:

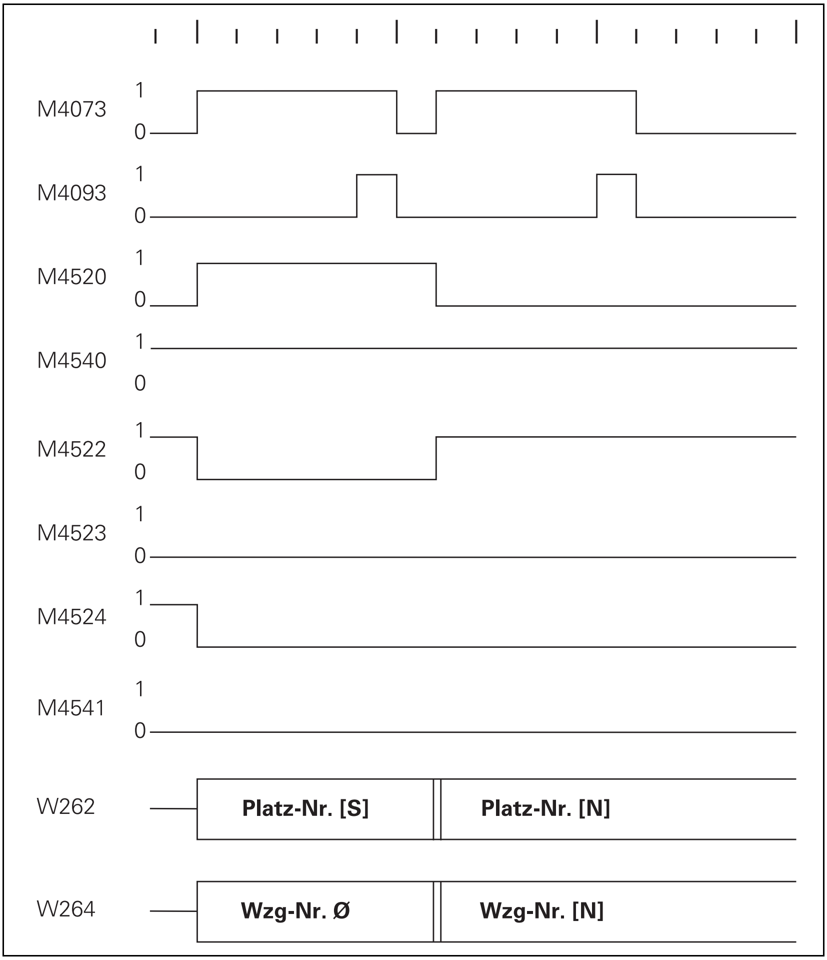
- With M4540 or MP7481.x specify the sequence in which the pocket numbers are transferred, depending on whether single or double changing arm.

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.

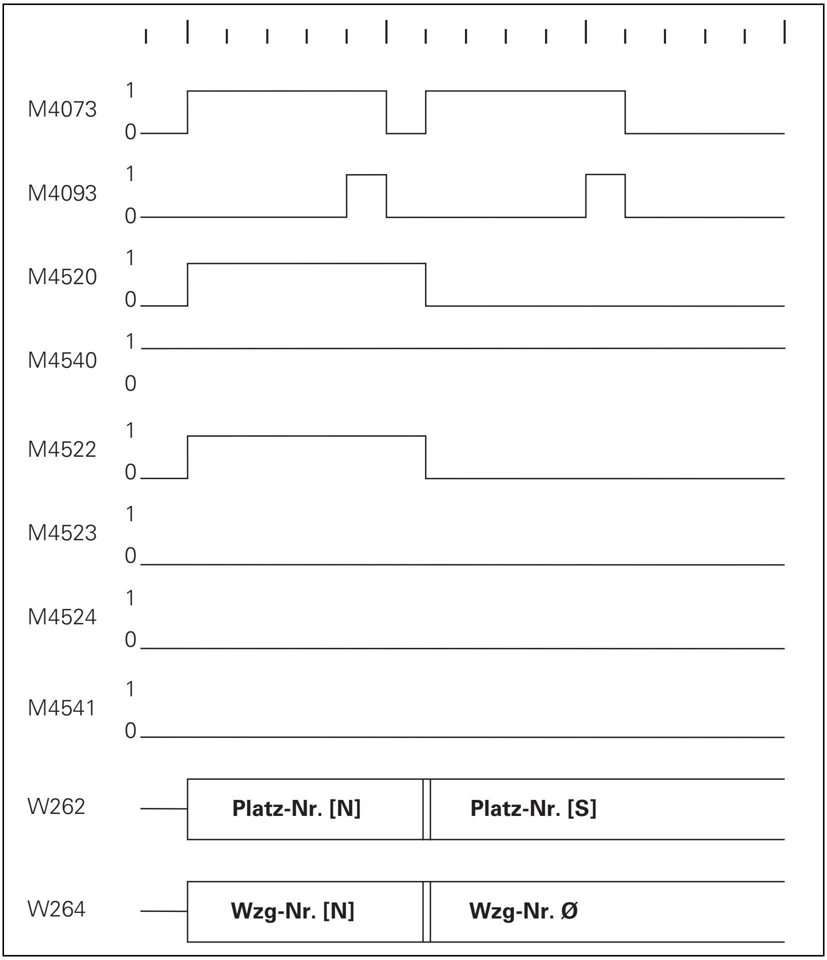


S → N, Double changing arm, M4540 = 1 or MP7481.x, bit x = 1

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!

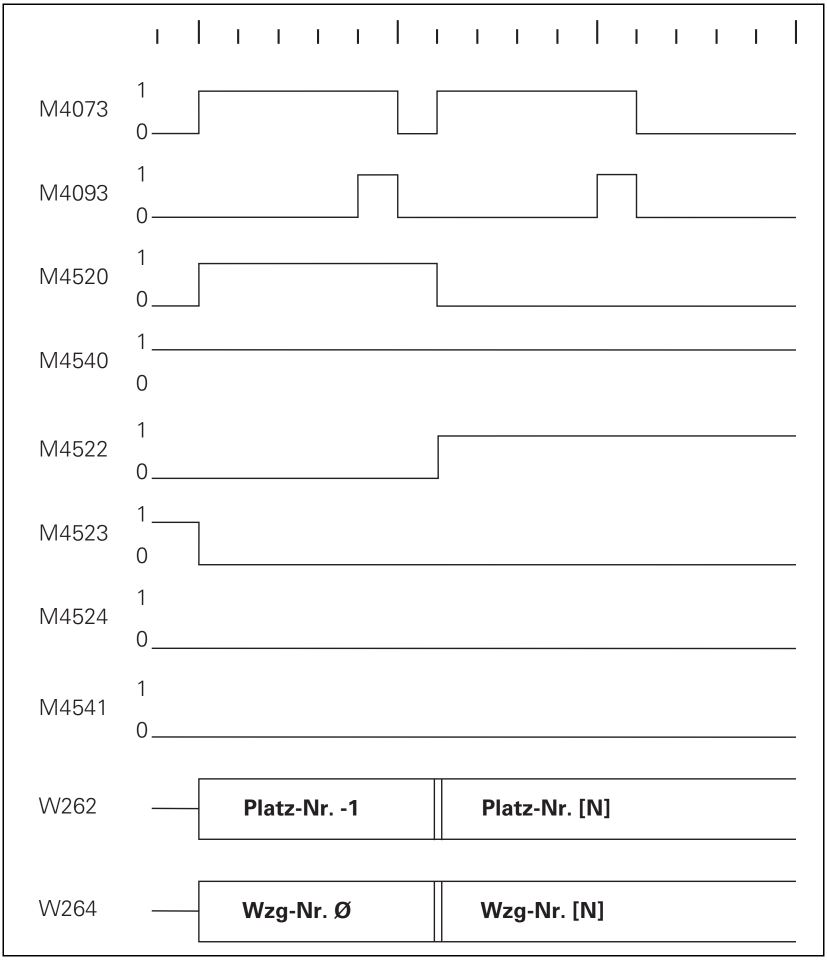


M → N: Normal tool follows a manual tool

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.

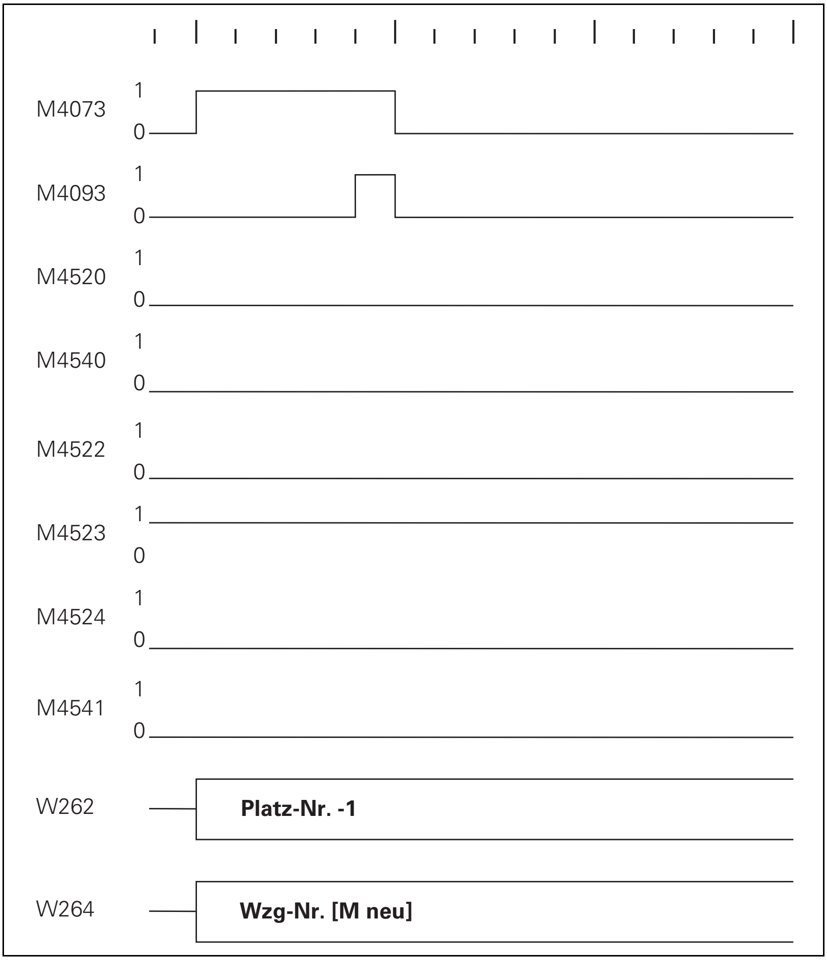
Zero means clear the spindle!
Pocket number –1 means: no pocket in the tool magazine!

▶ 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!



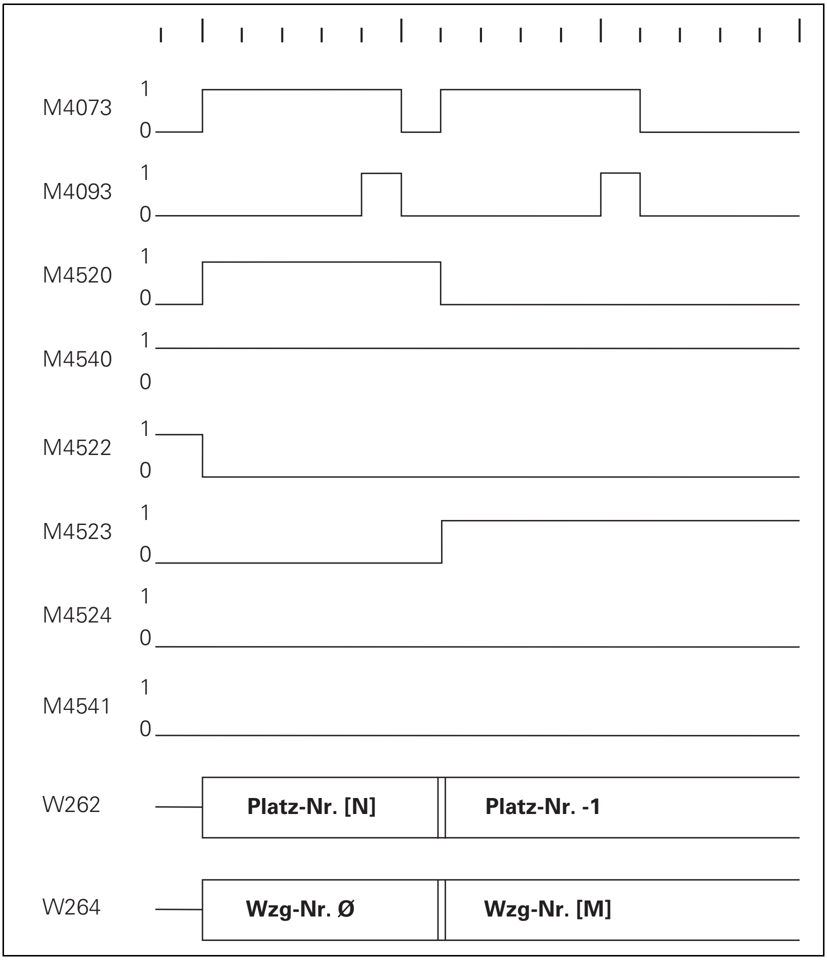
N → M: Manual tool follows a normal tool

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!



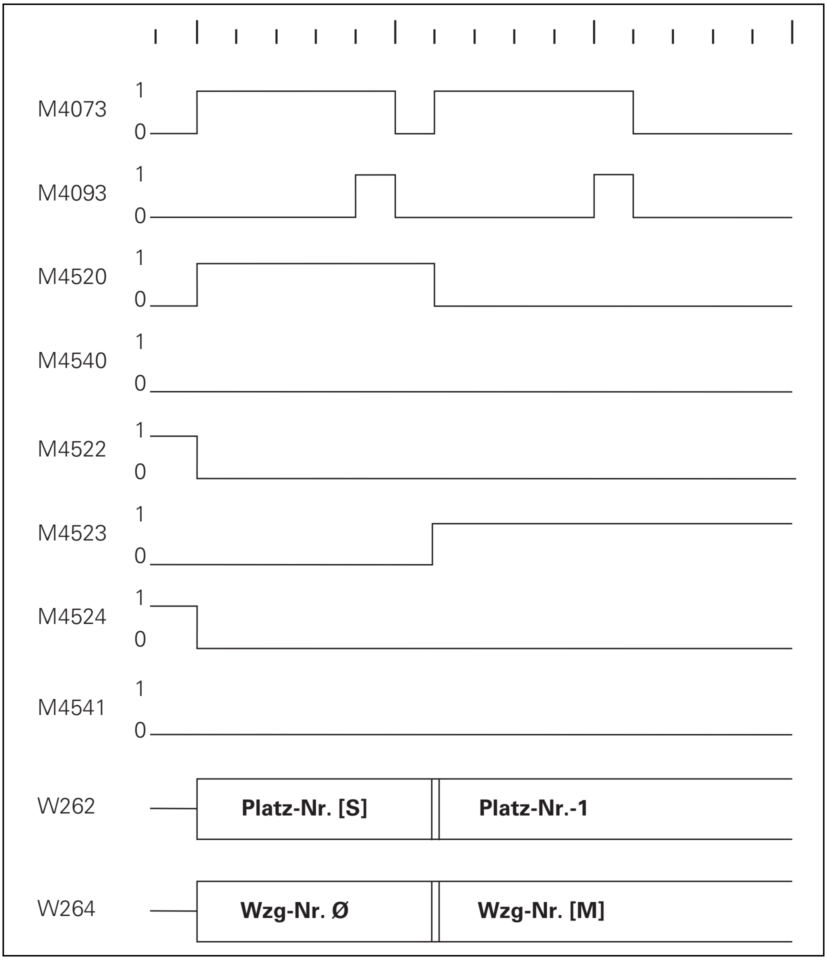
S → M: Manual tool follows a special tool

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.

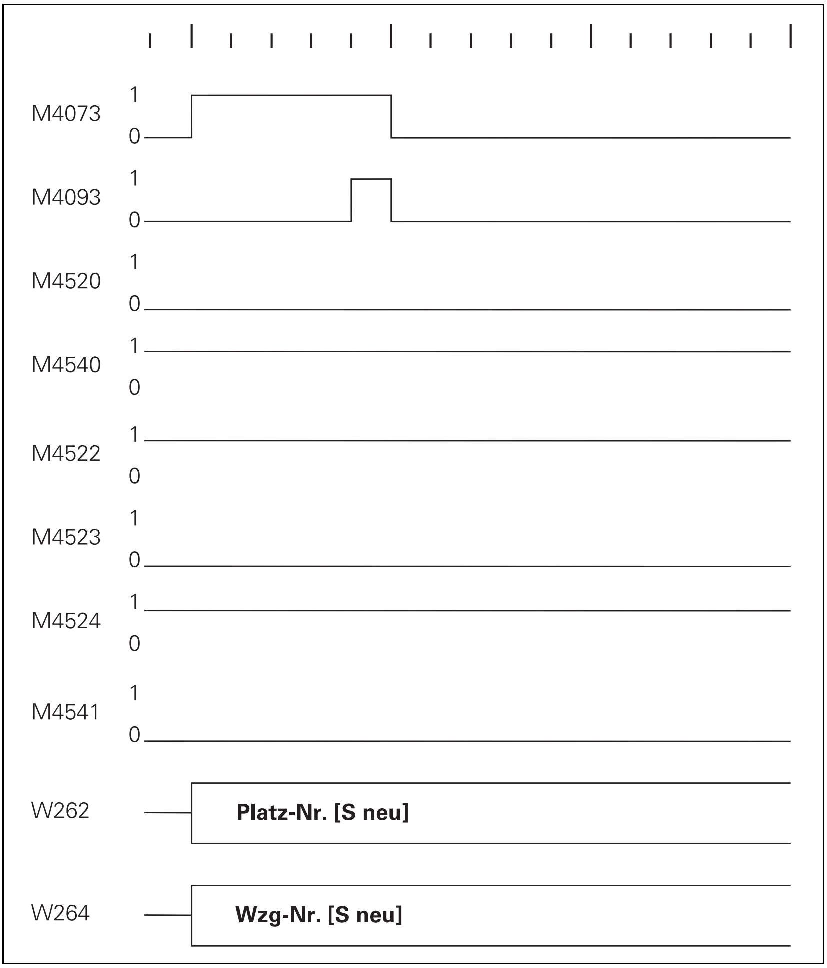
Pocket number –1 means: no pocket in the tool magazine!



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.

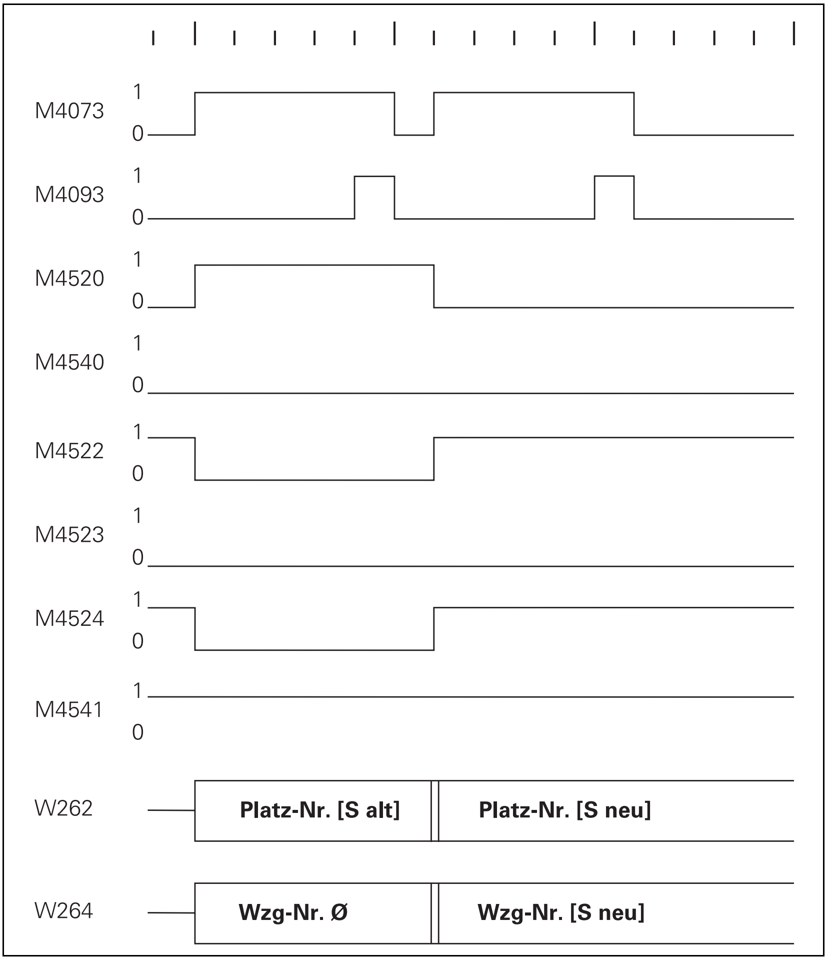


S → S, Single changing arm, M4540 = 0 or MP7481.x, bit x = 0

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.

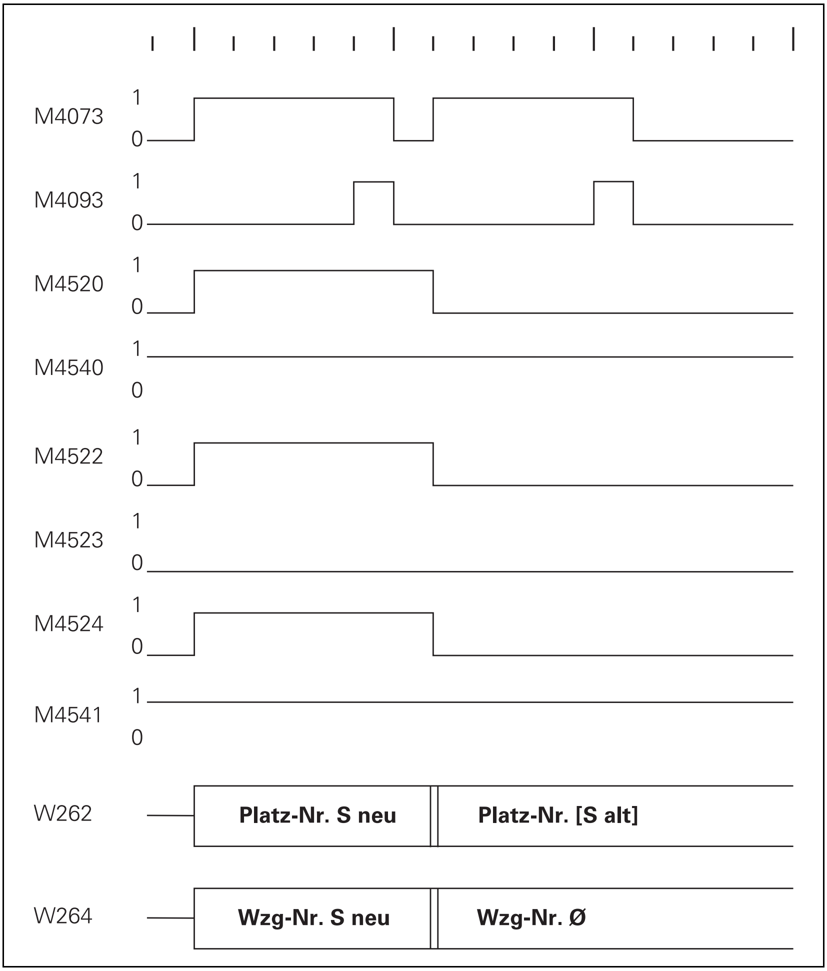


S → S, Double changing arm, M4540 = 1 or MP7481.x, bit x = 1

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

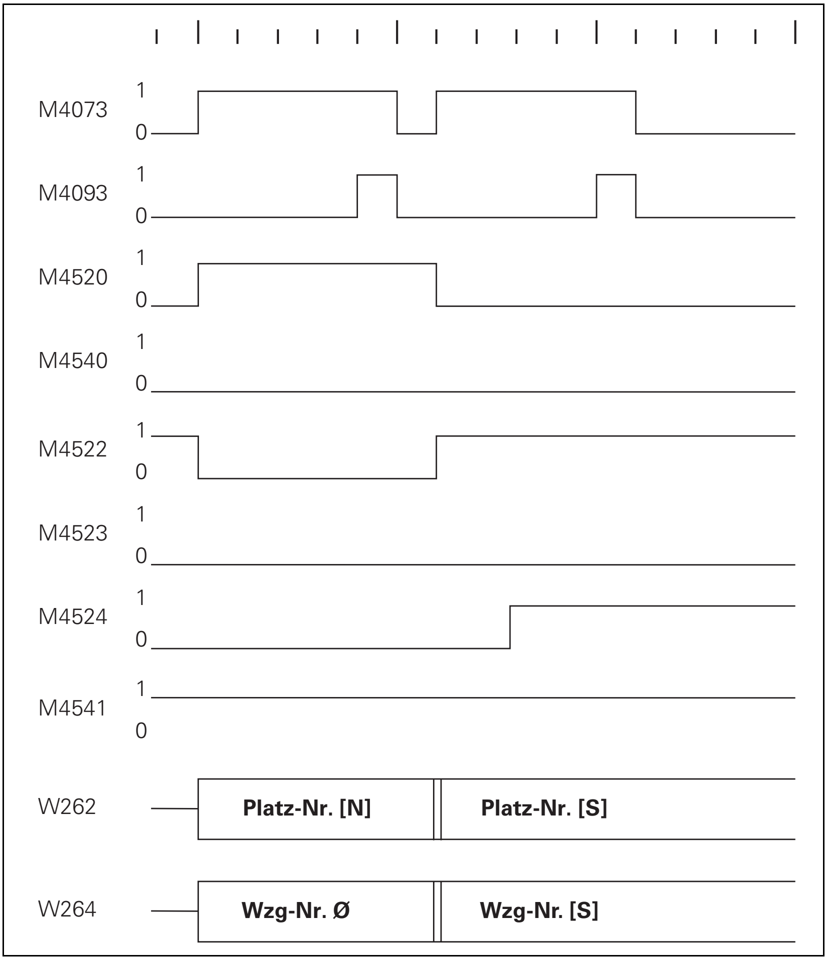
With this change sequence, two pocket numbers or two tool numbers must be transferred in succession. M4520 indicates that another **TOOL CALL** strobe (M4073) will follow. Regardless of M4541, there is a different sequence for the pocket number transfer for single and double-arm changers (M4540 or MP7481.x).

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.

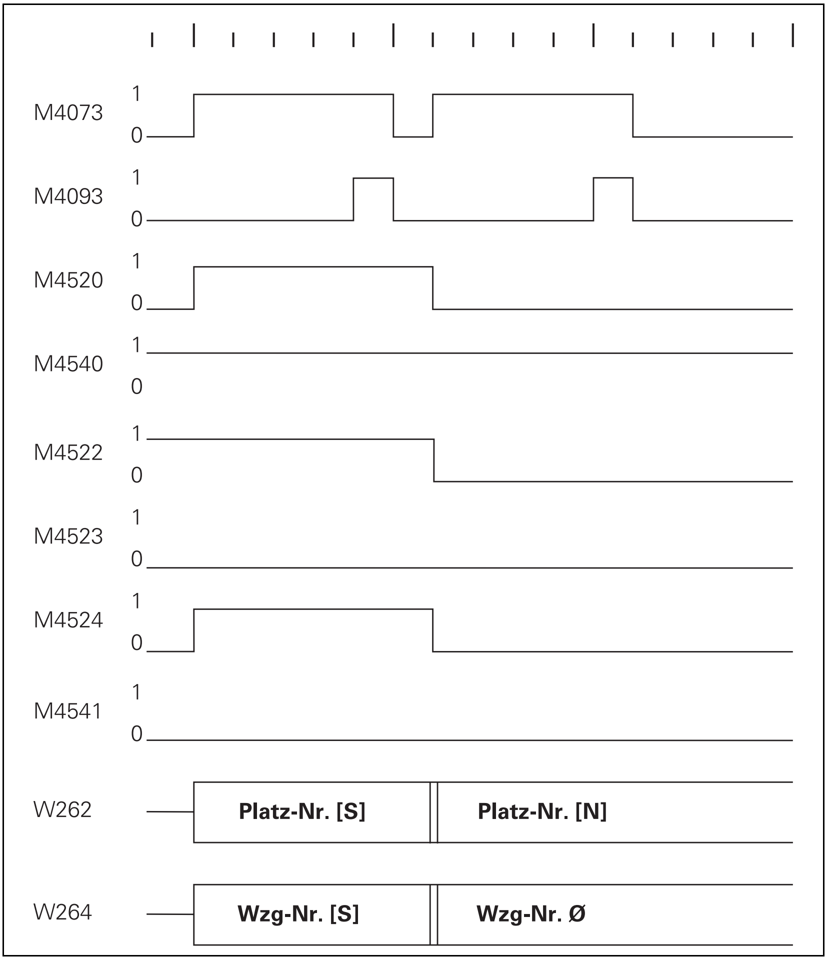


N → S, Double changing arm, M4540 = 1 or MP7481.x, bit x = 1

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!

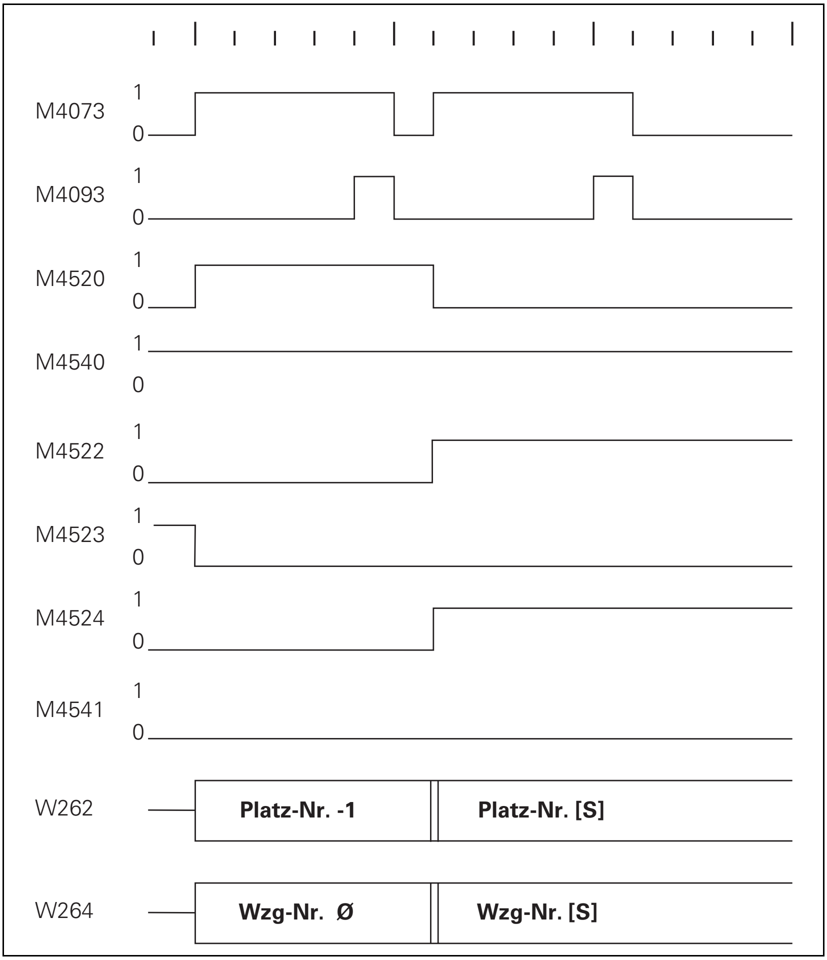


M → S: Special tool follows a manual tool

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.



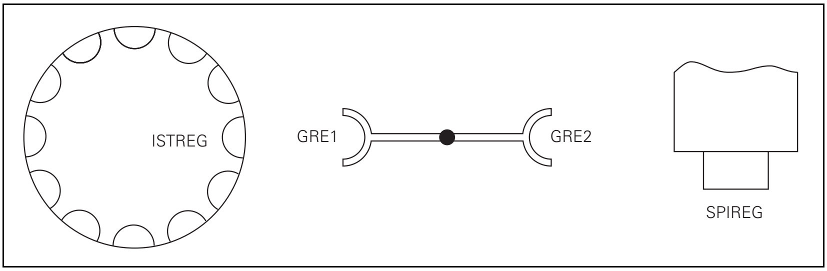
7.14.5 PLC Programming Example

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	**Further 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:

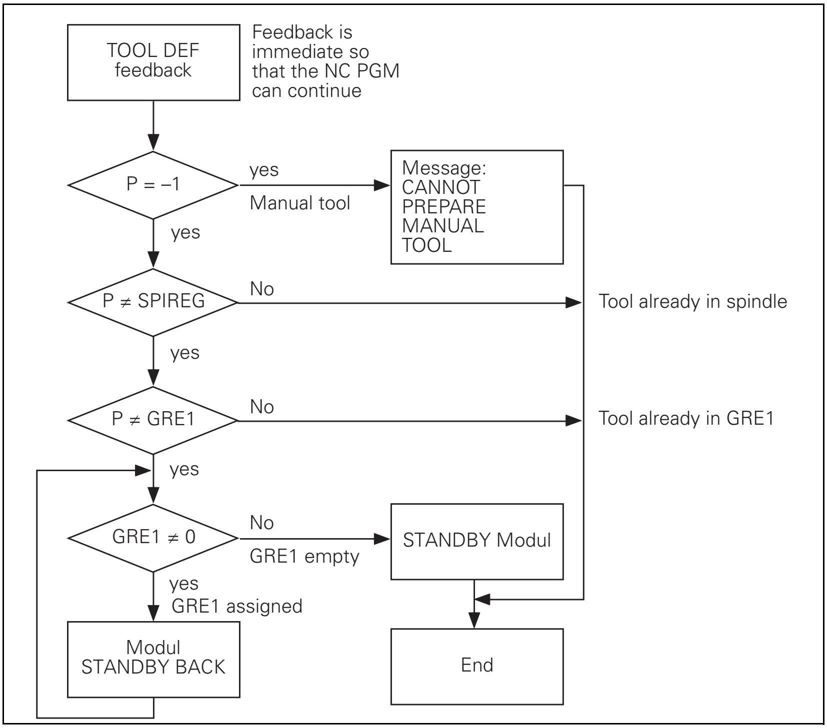
Machine parameters	Meaning
MP7260 = 90	Number of tools in the tool table
MP7261 = 12	Number of the pockets in the tool magazine
MP7480.0 = 4	Output of the pocket number and tool number for every TOOL CALL block
MP7480.1 = 4	Output of the pocket number and tool number for every TOOL DEF block

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	

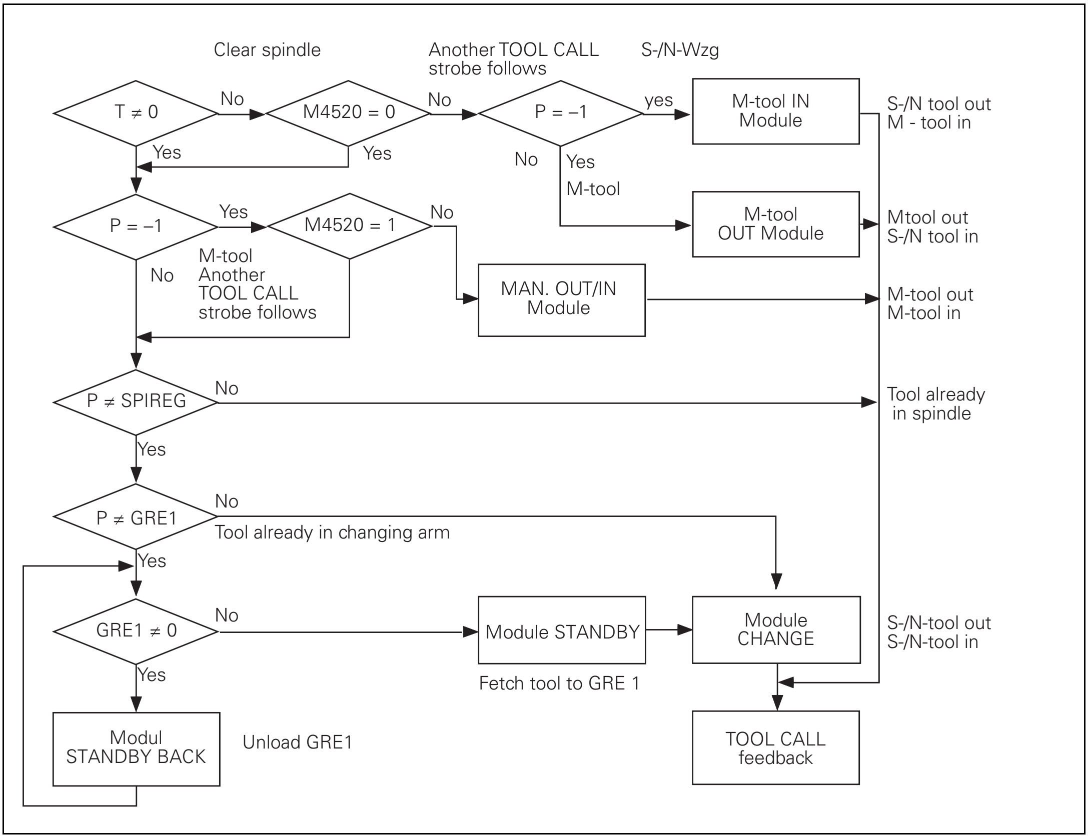
TOOL DEF program module

Search for tool and load in GRE1



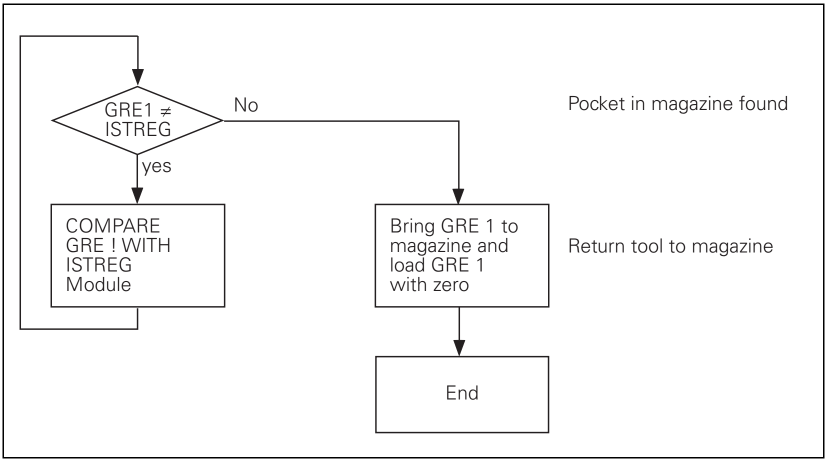
TOOL CALL program module

Automatic tool change, main program



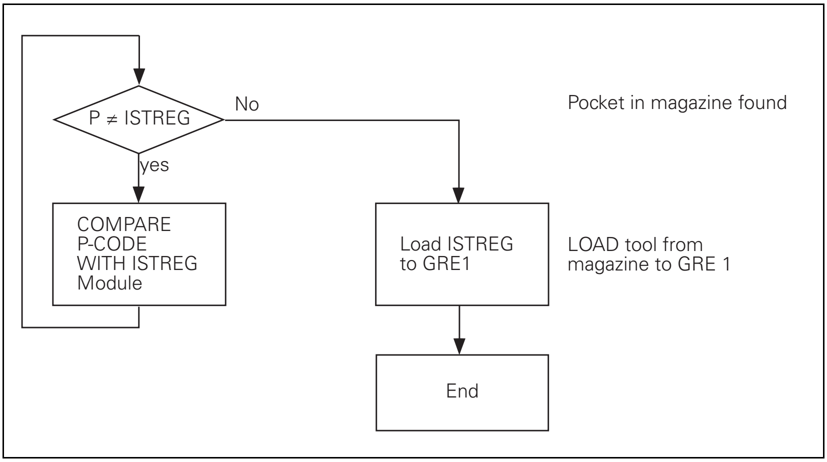
STANDBY program module

Search for tool and load in GRE1



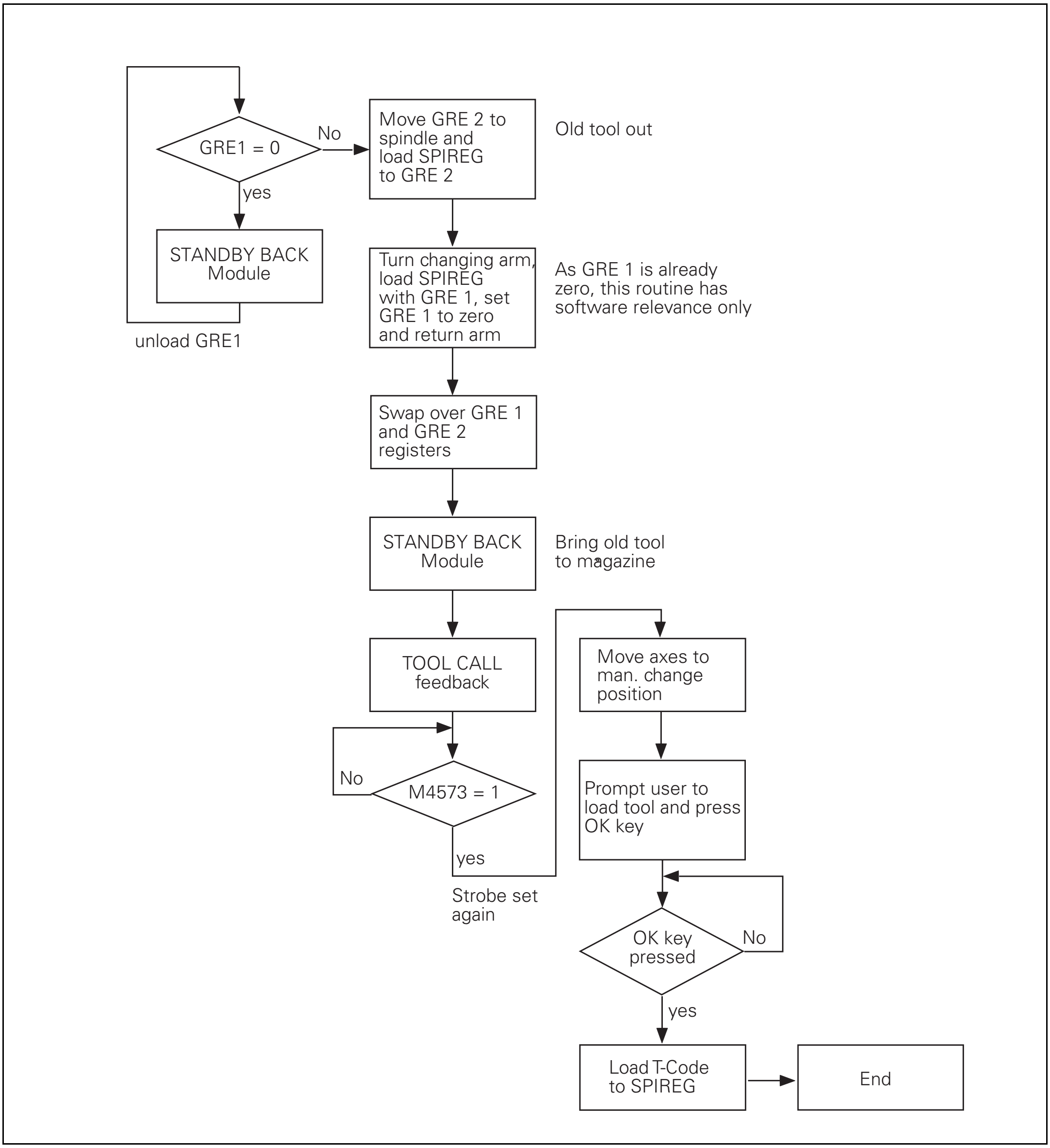
STANDBY BACK program module

Return tool from GRE1 to the tool magazine



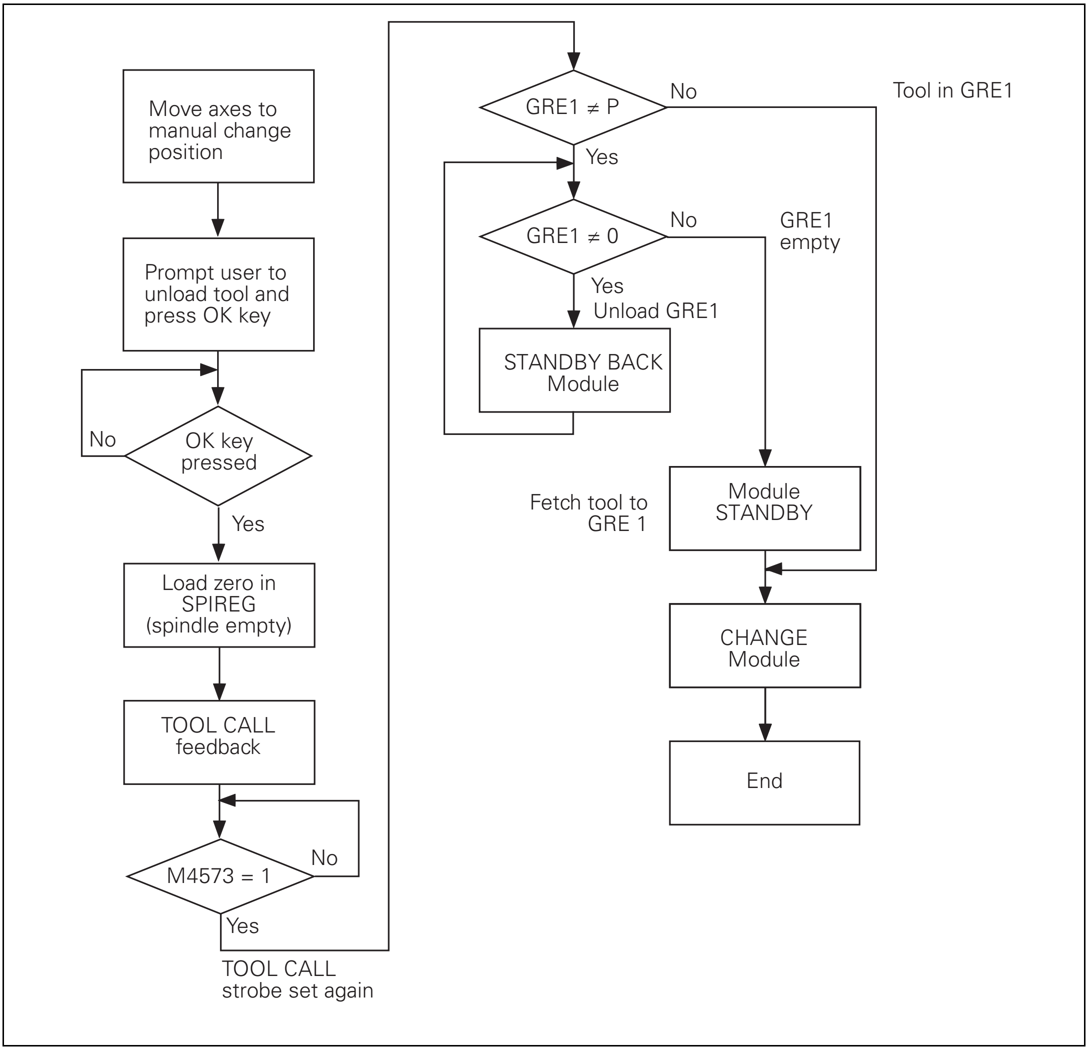
MANUAL TOOL IN program module

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).



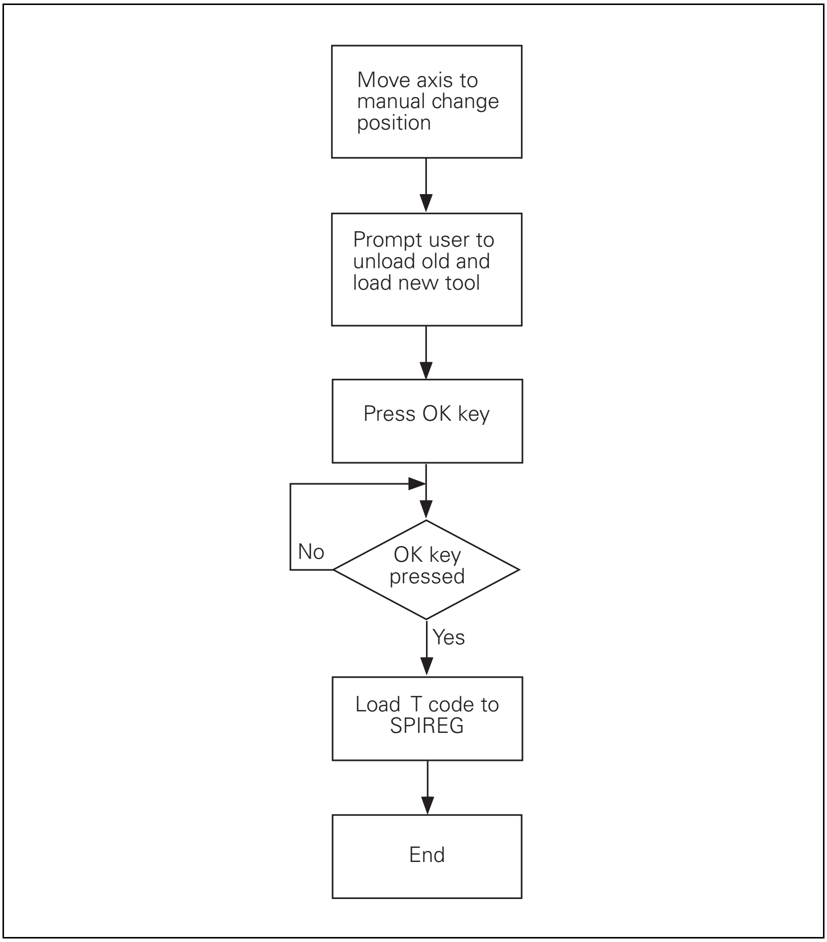
MANUAL TOOL OUT program module

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 inserted automatically.



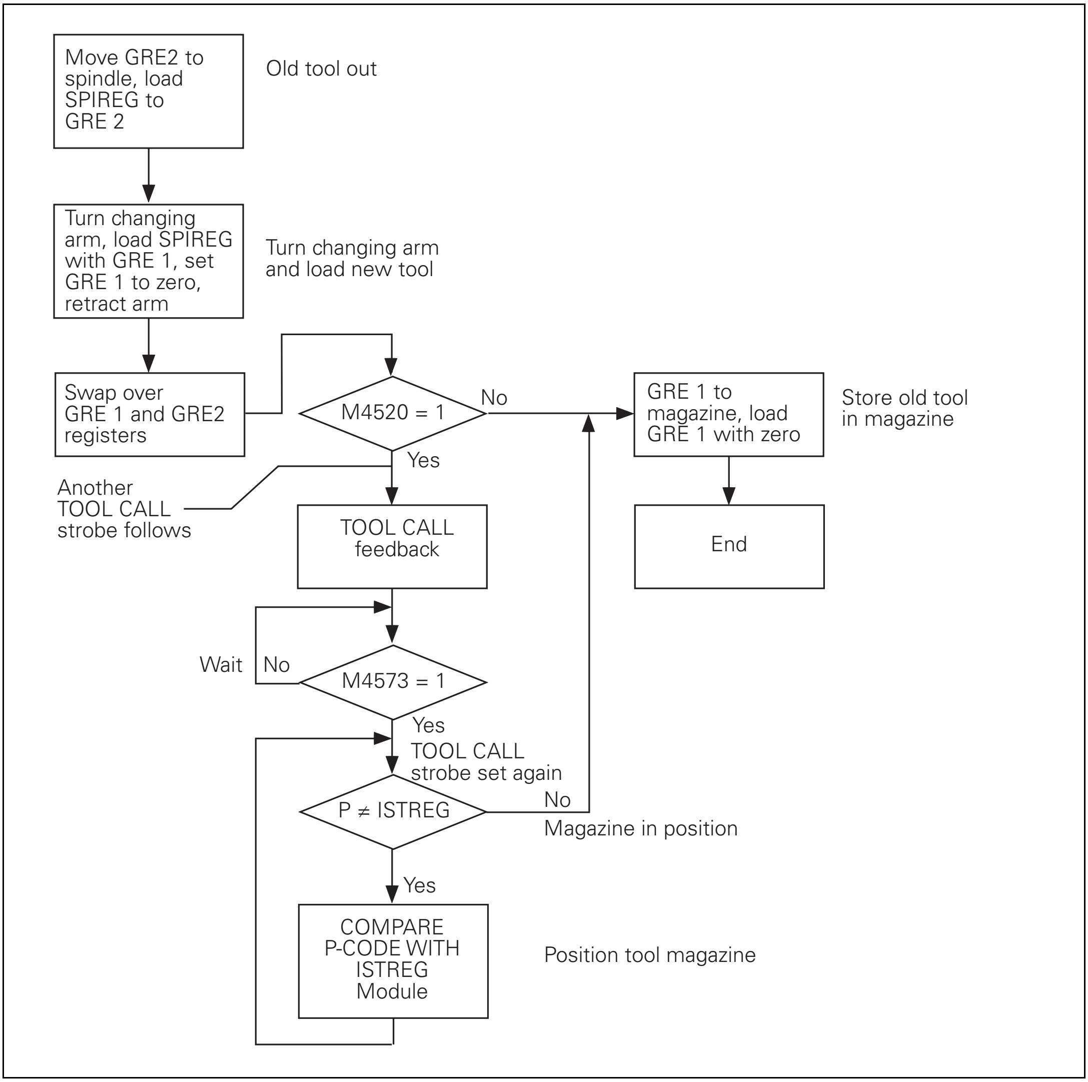
MANUAL TOOL OUT/IN program module

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.



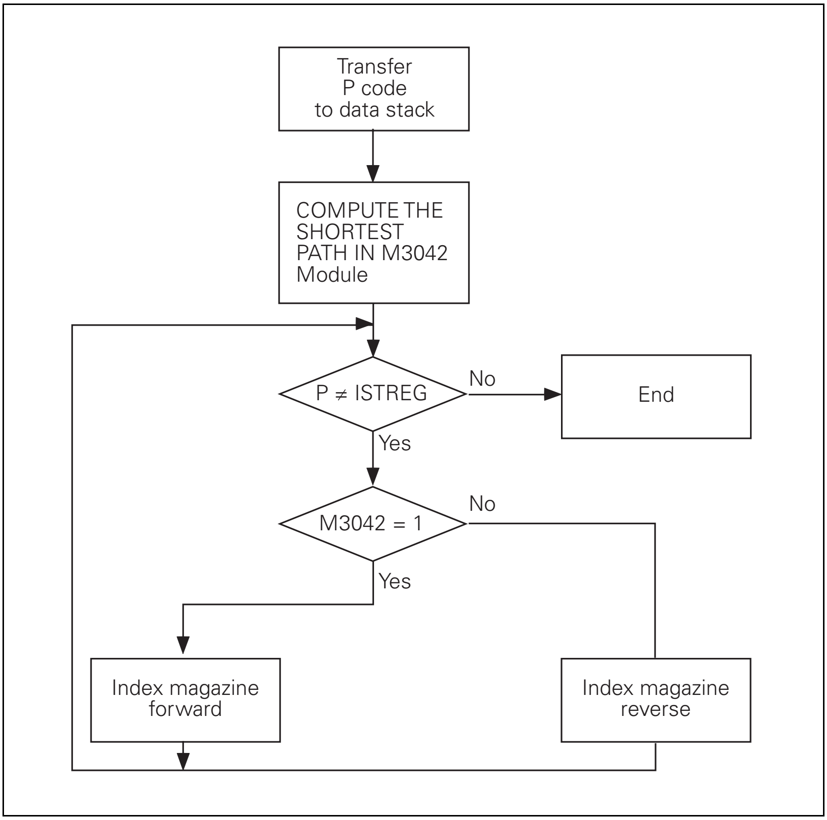
INSERT program module

The spindle is emptied and the new tool is automatically inserted. The PLC takes into account whether the tool should be returned to its original pocket (e.g., special tool).



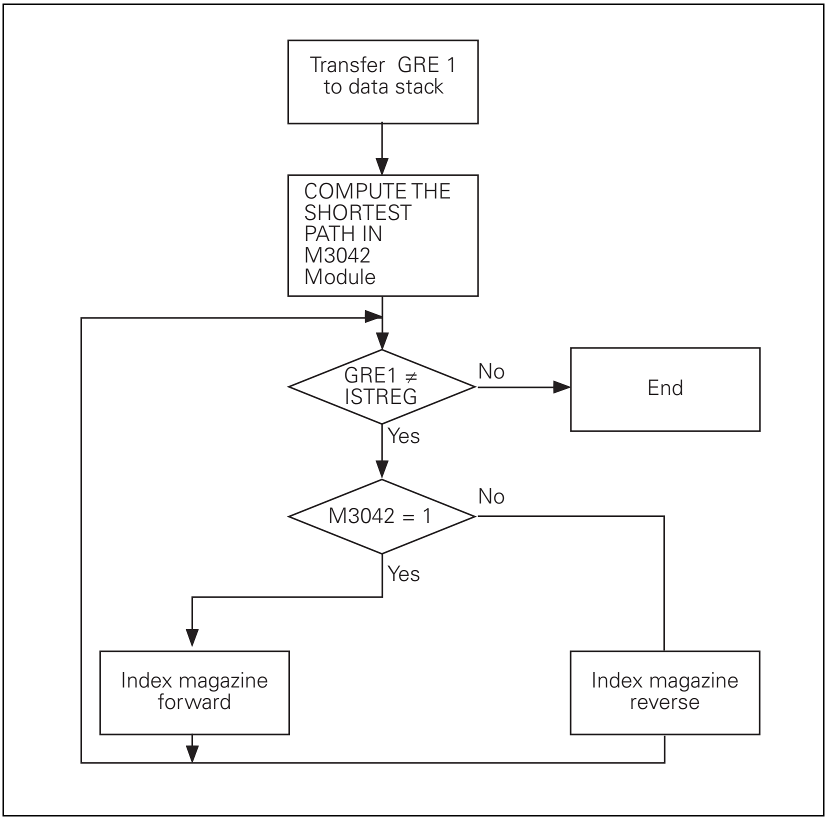
COMPARE P CODE WITH ISTREG program module

The tool magazine is positioned in the shortest direction to the desired pocket number.



COMPARE GRE1 WITH ISTREG program module

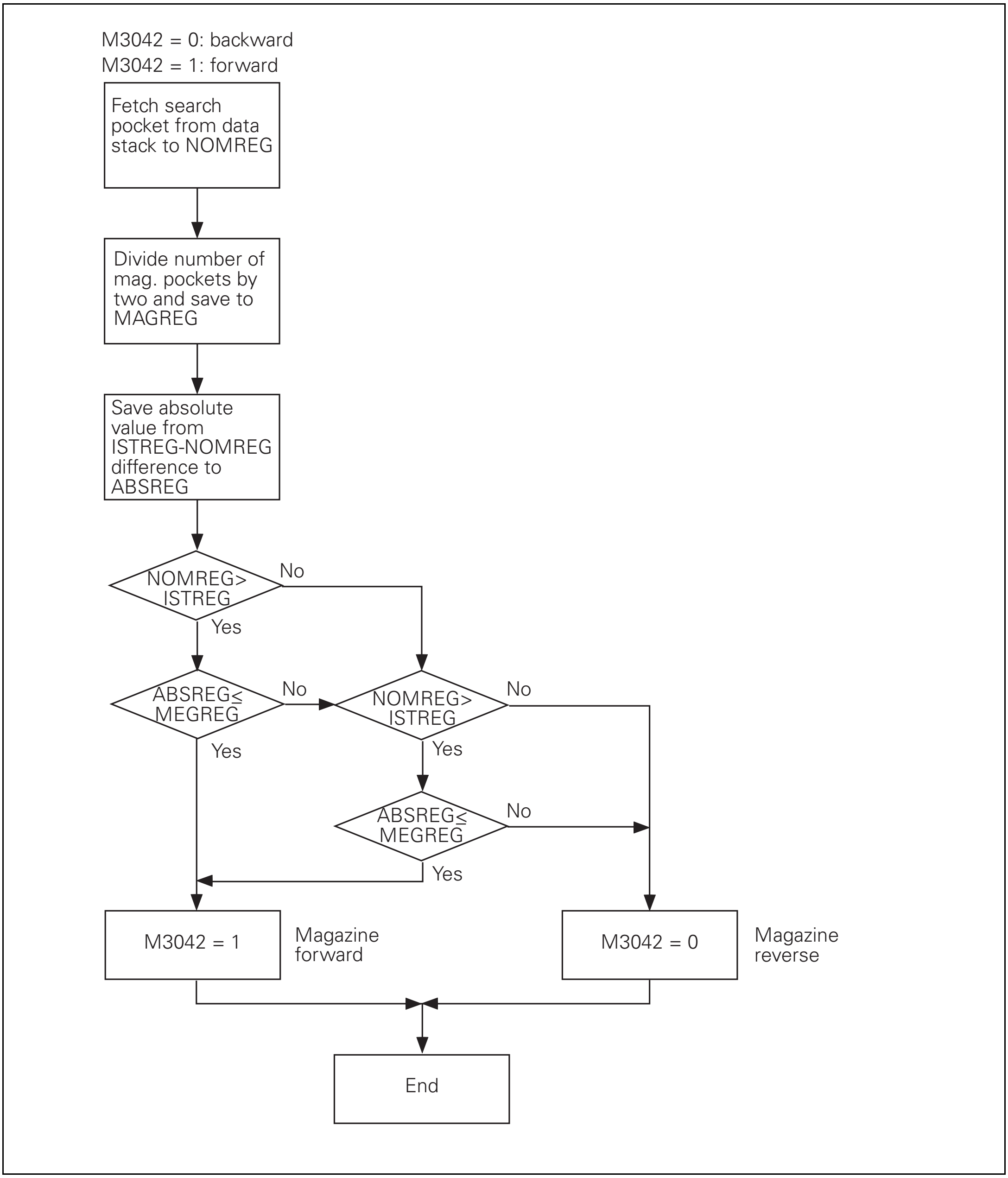
The tool magazine is positioned in the shortest direction to the pocket number that is located in GRE1.



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 PLC Programming

8 PLC Programming

8.1 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 through the TNC keyboard. It’s easier, however, to create your PLC programs on a PC with the PLC compiler software **PLCdesign.** For more information on **PLCdesign,** contact HEIDENHAIN.

The iTNC supports you with the COMPILE function, which checks the PLC program for logical errors, and the TRACE, TABLE and WATCH LIST functions, with which you can check the condition of the operands.

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 Finding the PLC cycle time
The PLC cycle time is determined in ms.

Call:
```
CM    9196
PL    D        <PLC cycle time in ms>
```

MP7600.1 PLC cycle time
= MP7600.1 * Position controller cycle time
= MP7600.0 * MP7600.0 * 0.6 ms
Input: 1 to 20 (recommended input value: 6)

8.1.1 Selecting the PLC Mode

Select PLC Mode:

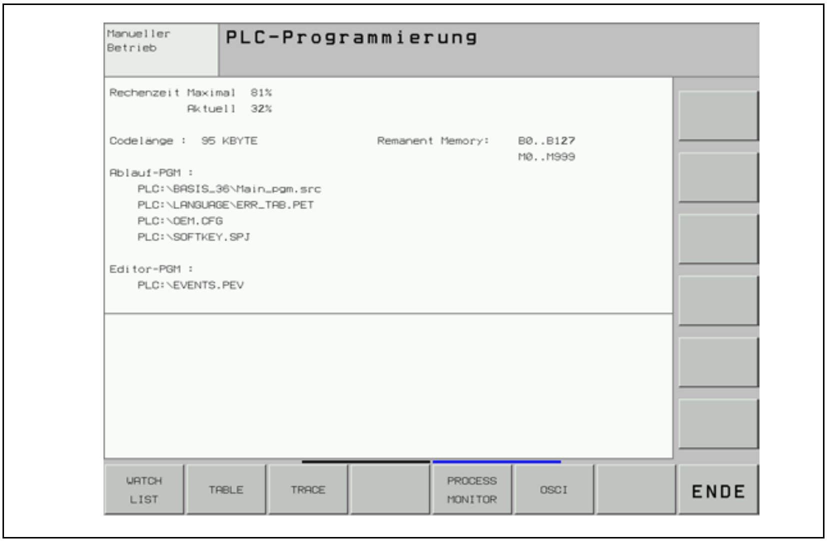
- Select the `Programming and Editing` operating mode.
- Press the MOD key.
- Enter the code number 807667 and confirm your entry with the ENT key, or if you already entered the code number, press the PLC EDIT soft key.

Exit PLC mode:

- Press the END hard key or soft key.

8.1.2 PLC Main Menu

After you have entered the code number (or pressed the PLC EDIT soft key) the iTNC displays the PLC main menu:



Processing time maximum:
Maximum run time of the PLC program
The PLC processing time (time for a PLC scan) 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 $t_{calc.}$ [%]:

$$t_{run} = \frac{t_{PLC} \cdot t_{calc.}}{21\ \text{ms}}$$

If the maximum run time of the sequential program is exceeded, the iTNC displays the blinking error message **PLC: time out.**

Processing time current:
The time taken for the latest PLC scan in %.

Code length:
Length of the compiled sequential program in KB. Maximum value: 512 KB.

PGM in exec.mem:
The PLC program, PLC error table, soft-key project file and possibly the configuration file for source-code programming are in the process memory. At start-up, the iTNC automatically compiles the files defined in OEM.SYS. The files only become active after they have been compiled.

PGM in edit mem:
Name of the file located in RAM memory.

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	Editing the file located in RAM memory.
COMPILE	Compile PLC program (See page 8 – 16).
MP EDIT	Display a list of machine parameters.
WATCH LIST	Display the statuses of selected operands in tabular format (See page 8 – 8).
TABLE	Display the logical states of the PLC operands (See page 8 – 11).
TRACE	Display the TRACE Function or logic diagram (See page 8 – 12 and Page 8–14).
PROCESS MONITOR	Process monitor (Page 8–138).
OSCI	Activate the integrated oscilloscope (Page 6–221).
ENDE	Exit the PLC mode.

8.1.3 File Management

File management in PLC mode is largely the same as in the **Programming and Editing** mode of operation (see User's Manual for iTNC 530). 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
ALLE ANZ.	Show all files
.PLC DATEIEN	Show only PLC programs (*.PLC)
ZEIGE .A	Show only ASCII files (*.A)
.HLP DATEIEN	Show only help files (*.HLP)
.SYS DATEIEN	Show only system files (*.SYS)
.COM DATEIEN	Show only compensation value tables (*.COM)
.CMA DATEIEN	Show only tables with compensation value assignments (*.CMA)
.PET DATEIEN	Show only PLC error tables (*.PET)
.SRC DATEIEN	Show only PLC source files (*.SRC)
.SPJ DATEIEN	Show only soft-key project files (*.SPJ)
ENDE	Return to previous menu

8.1.4 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 WACTH 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
ZEILE EINFÜGEN	Insert a new line above the current line
ZEILE LÖSCHEN	Delete active line
SYMBOL LIST	Display selection list with all symbolic operands used in the active PLC program
ANFANG ↑	Jump to the beginning of the selection list
ENDE ↓	Jump to the end of the selection list
SEITE ↑	Scroll back one page in the selection list
SEITE ↓	Scroll forward one page in the selection list
SUCHEN	Search the selection list for a specific text
AUSWÄHLEN	Load selected operands into the WATCH LIST
ENDE	Close the pop-up window
ANFANG ↑	Jump to the beginning of the WATCH LIST
ENDE ↓	Jump to the end of the WATCH LIST
SEITE ↑	Scroll back one page in the WATCH LIST
SEITE ↓	Scroll forward one page in the WATCH LIST
HEX ↕ DEZIMAL	Show contents of operands as decimals or hexadecimals
ENDE	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 selection window 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 selection window.

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 INSERT LINE soft key.
- In the **`ADDR`** column, enter the absolute address of the operand, i.e. W1022.
- Press the ENT key.

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.

8.1.5 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.

Soft keys within the TABLE function:

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
HEX ↕ DEZIMAL	Show contents of operands as decimals or hexadecimals
STRING	List of strings (only the first 70 characters). Overwriting is not possible.
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
ENDE	Return to previous menu

8.1.6 The TRACE Function

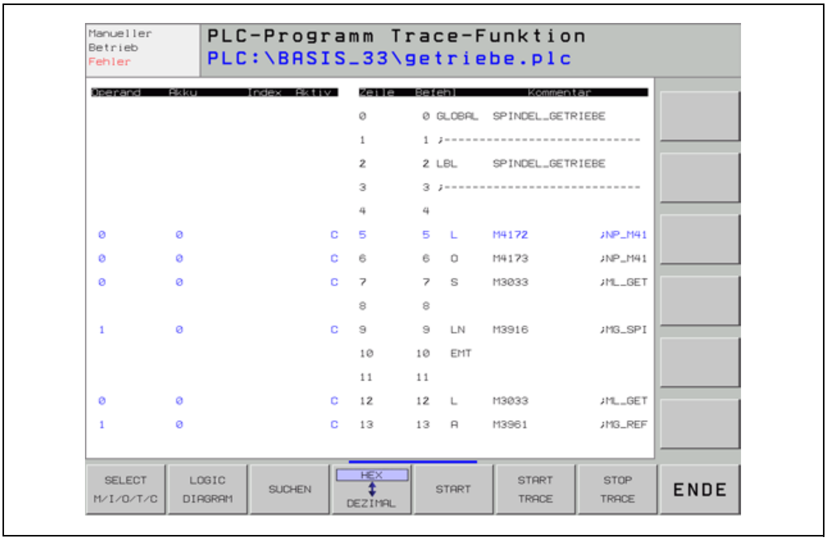
With the TRACE function you can:

- Control the logical states of markers, inputs, outputs, timers, and counters.
- Check the content of bytes, words and double words.

Select the TRACE function through the TRACE soft key in the PLC main menu. The iTNC displays:

- 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 selected 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
	Select the operand type for logic diagram
	Show the logic diagram
	Search for text in STL (TRACE IN CODE)
	Show operand or accumulator contents in hexadecimal or decimal notation
	Stop dynamic display of the operand content, the accumulator content, and the logic diagram with STOP; continuously update again with START.
	
	Start the trace
	End the trace
	Return to previous menu

8.1.7 The Logic Diagram

Soft keys within the LOGIC DIAGRAM function:

Soft key	Function
SELECT M/I/O/T/C	Select **M**arkers/**I**nputs/**O**utputs/**T**imers/**C**ounters for a logic diagram.
TRACE IN-CODE	Display the trace in code
SAVE TRACE BUFFER	Save current logic diagram in an ASCII file (*.A)
RESTORE TRACE BUFFER	Show the saved logic diagram
START	Stop dynamic display of the operand content, the accumulator content, and the logic diagram with STOP; continuously update again with START.
STOP	
START TRACE	Start the trace
STOP TRACE	End the trace
ENDE	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, during which the iTNC records up to 1024 PLC scans.

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 incorrect entries, simply press DEL.

For each operand you can enter one trigger condition. The iTNC records 512 states both before and after a trigger event. The following are possible trigger conditions:

- **1:** Record if operand is logically 1 (trigger on positive edge).
- **0:** Record if operand is logically 0 (trigger on negative edge).

If you do not need a trigger condition, answer the dialog prompt with NO ENT. If you enter no trigger condition for any of the operands, the iTNC records the states of the operands continuously. The 1024 most recent states remain saved.

To start recording:

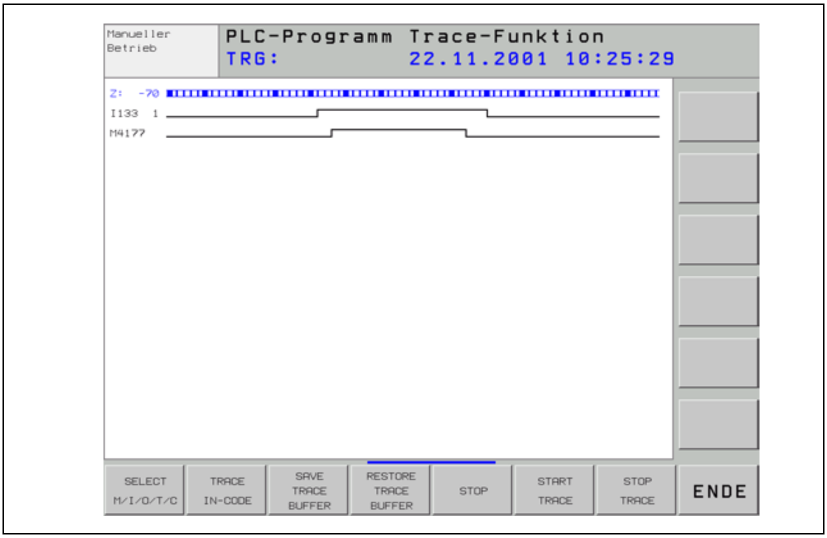
▶ Press the soft key START TRACE.

To stop recording:

▶ Press the STOP 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:



8.1.8 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.**

Soft keys within the COMPILE function:

Soft key	Function
OEM.SYS: ALL	Compile PLC program, PLC error table, and soft-key project file (entries **PLCMAIN=, PLCERRTAB=,** and **SOFTKEYPROJECT=** in OEM.SYS)
OEM.SYS: PLC-MAIN PROGRAM	Compile PLC program (entry **PLCMAIN=** in OEM.SYS)
OEM.SYS: PLC-ERRORTAB.	Compile PLC error table (entry **PLCERRTAB=** in OEM.SYS) (See page 7 – 23).
OEM.SYS: SOFTKEY-PROJECT	Compile soft-key project file (entry **SOFTKEYPROJECT=** in OEM.SYS) (See page 7 – 99).
AUSWAHLEN	Select file to be compiled
ENDE	Return to PLC main menu

To compile a PLC program:

- Press the COMPILE soft key: the iTNC displays the soft keys for the COMPILE function.
- Press the SELECT soft key: the iTNC displays an overview of the available programs.
- Use the arrow keys to select the PLC program to be compiled.
- Press ENT.

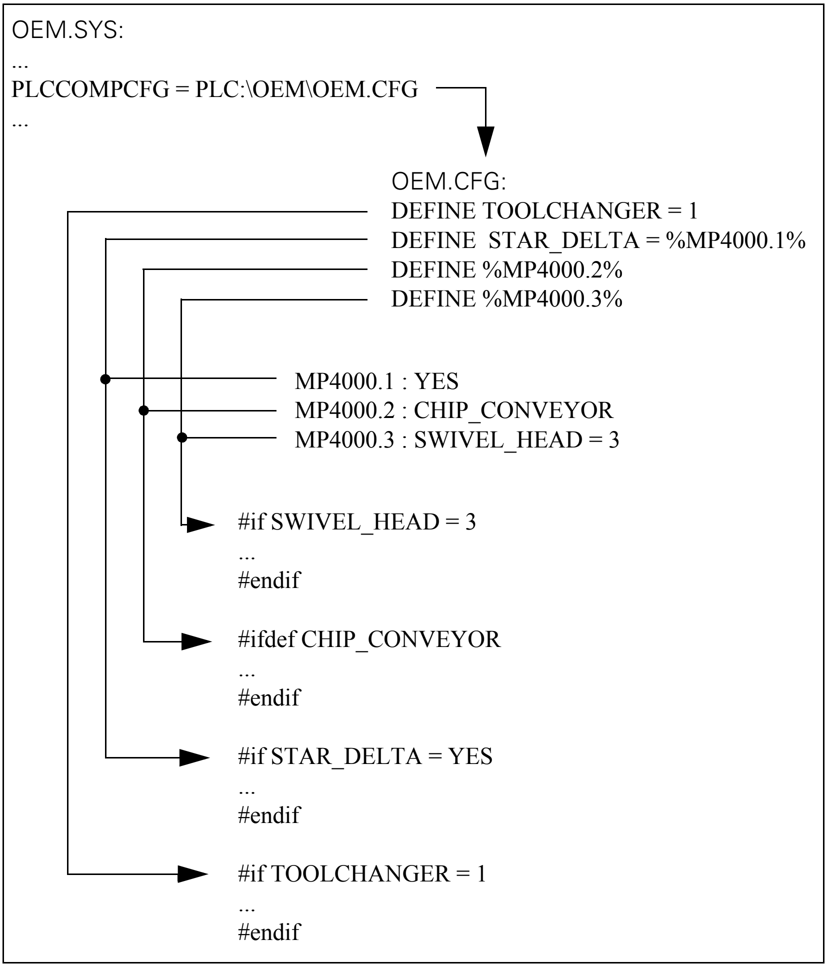
The name and path of the compiled PLC program are entered in OEM.SYS with the **PLCMAIN=** entry. Press the OEM.SYS: PLC-MAIN PROGRAM soft key if you only want to compile the PLC program from this entry.

8.2 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.

- 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:



MP4000.0-15 Options for the conditional compilation of the PLC program

8.3 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.

Warning

Alterations to the system partition can impair proper function of the iTNC!

Size of the partitions

Partition	Content	Size
SYS	System files	2 GB
PLC	OEM files	2 GB
TNC	User files	Remaining memory on hard disk (at least 2 GB)

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.

8.4 System Files

8.4.1 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.

The following code words are defined:

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`

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`

PLCPWM = Path for PLC program for commissioning of digital axes.

Input example: `PLCPWM = PLC:\IB_PGM\IB530.PLC`

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`

PLCERROR = Name for text file with PLC error messages; the path for the text file is permanently defined.

Input example: `PLCERROR = PLC_ERR.A`

PLCERRFIX = Path for "Corrective action" help text.

Input example: `PLCERRFIX = FIX.A`

PLCERRREASON = Path for "Cause of error" help text.

Input example: `PLCERRREASON = REASON.A`

PLCDIALOG = Name for text file with PLC dialogs; the path for the text file is permanently defined.

Input example: `PLCDIALOG = DIALOG.A`

PLCSOFTVERS = (mandatory entry): iTNC displays PLC software version when the MOD key is pressed.

Input example: **`PLCSOFTVERS = BASIS--33-03`**

TABCMA = Path for compensation value tables for axis error compensation. (See "Nonlinear axis error compensation" on page6 – 37)

Input example: **`TABCMA = PLC:\AXIS_COR\CORRECT.CMA`**

MODEHELP = Path for help texts and machine commands.

Input example: **`MODEHELP = PLC:\LANGUAGE\GERMAN\OPTIMIER.HLP`**

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!

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!

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`**

TTYP = Path and file name for list of the tool types.

PLCEVENTS = Path for event list (SPAWN command).

Input example: **`PLCEVENTS = PLC:\EVENTS.PEV`**

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.

KINEMATIC = Path for the assignment table of the tilting-axis geometry description.

Input example: **KINEMATIC = PLC:\KINELIST.TAB**

REMOTE. LOCKSOFTKEY VISIBLE = Display **External access ON/OFF** soft key.

Input example: **REMOTE.LOCKSOFTKEYVISIBLE = 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. PLCPASSWORD FORCED = Setup, machine backup and full backup only with the password from **PLCPASSWORD =**

Input example: **REMOTE.PLCPASSWORDFORCED = YES**

AXISNUMBER = Number of the indexes of the machine parameters (except MP2xxx.y) in the machine-parameter file.

Input example: **AXISNUMBER = 6**

PWM PARAMETER = Number of the indexes of machine parameters MP2xxx.y (for the current and speed controller) in the machine-parameter file.

Input example: **PWMPARAMETER = 6**

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)

LOGO = Path for customer-specific company logo during control power-up.

Input example: **LOGO = PLC:\LOGO\OEM-LOGO.BMP**

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**

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**

TCHRULES = Path for the *.TCR definition file containing magazine rules for tool magazines.

Input example: **TCHRULES = PLC:\RULES.TCR**

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 detection:

Marker	Value	Meaning
M4203	0	Interface was released
	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 detection:

Marker	Value	Meaning
M4203	0	Interface was released
	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

8.4.2 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 8 – 25).

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>
- STARTUPCANCEL = <Name of the macro called when mid-program startup is not completed with **Restore machine status**>

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.

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.**

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.

Call:
```
PS      B/W/D/K/S<Code word>
CM      9291
```

Error detection:

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 code word is not an NC program (*.H or *.I)
	36	The file given under the code word does not exist

8.4.3 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 7 – 39).

8.4.4 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 7 – 39).

8.4.5 PLCSOFTK.SYS

Path for the file names of the PLC soft-key pictures. (See “PLC Soft Keys” on page7 – 99)

8.4.6 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).

8.4.7 TNC.SYS

The end user can define certain paths and functions in this 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>

8.5 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, the PLC sets the acknowledgement marker M4092. The PLC must then reset M4092, otherwise no further strobes can be sent by the NC.

8.5.1 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

8.5.2 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

8.5.3 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 must enter the code number 555 343 before you can define function FN17 (soft keys: Q-parameter programming, special functions). After a control reset, the code number must be entered again if you wish to program **FN17.** The iTNC provides the following functions:

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
Spindle switchover				
	20	13	–	0 = Spindle 1 1 = Spindle 2
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 in tool radius DR2
		7	Tool no.	Tool locked TL 0 = not locked, 1 = locked
		8	Tool no.	Number of the 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 tool 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 = 99 999.9999
		20	Tool no.	TT: Offset in length L-OFFS
		21	Tool no.	TT: Break tolerance in length LBREAK
		22	Tool no.	TT: Break tolerance in radius RBREAK
		23	Tool no.	PLC value

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
Coordinate transformation				
	210	1	–	Basic rotation (manual)
		3	–	Active mirrored axes Bits 0 to 2 and 6 to 8: Axes X, Y, Z and U, V, W
		6	–	Tilt working plane during 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
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)
Tilting axes				
	290	1	–	Tilting axis geometry description
TS touch-trigger probe				
	350	10	–	Tool axis
		11	–	Effective radius
		12	–	Effective length
		13	–	Radius of calibration ring
		14	1	Center offset (reference axis)
			2	Center offset (minor axis)
		15	–	Center offset direction

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
TT touch probe for tool measurement				
	350	20	1	Center of axis 1
			2	Center of axis 2
			3	Center of 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 transformation				
	420	0	0	0 = Globally effective
Write values into active datum table				
	500	Line	Column	Depends on MP7475
	501	Line	Column	
Velocity semifeedforward control				
	600	1	Axis	Factor for velocity semifeedforward
		2	0 or NO ENT	Use factor from MP1396.x
Touch probe cycles				
	990	1	–	Approach behavior: 0 = Standard behavior 1 = Effective radius, safety clearance zero
		2	–	0 = Probe monitoring off 1 = Probe monitoring on
		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 data item
Coordinate transformation				
	990	4	1	Transformation of the manual mode coordinate system into the active coordinate system (e.g. rotated, shifted)
			2	Transformation of the active coordinate system (e.g. rotated, shifted) into the manual mode coordinate system
		5	5	Ask if due to a tilt motion, an axis is shown in an untilted coordinate system on top of another axis. The number of the first of two sequential Q parameters must be given. It contains the axis to be asked (0 = X, 1 = Y, 2 = Z). The second Q parameter should return the corresponding image (0 = X, 1 = Y, 2 = Z, –1 = Axis has no image).
		8	–	Spindle orientation including the angle
PLC data				
	2000	10	Marker no.	PLC markers

8.5.4 Data Transfer NC → NC program (FN18: SYSREAD)

You can use the **FN18: SYSREAD** function particularly for OEM cycles if you wish to access certain NC data, e.g., active tool compensation values, from the NC program. 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 data item
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 state				
	20	1	–	Tool number
		2	–	Prepared tool number
		3	–	Active tool number 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 the active tool
		15	Number of the 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)

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
Cycle parameters				
	30	1	–	Setup clearance
		2	–	Total hole depth/milling depth
		3	–	Plunging 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
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 in tool radius DR2
		7	Tool no.	Tool locked TL 0 = not locked, 1 = locked
		8	Tool no.	Number of the 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 tool 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 = 99 999.9999
		20	Tool no.	TT: Offset in length L-OFFS

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
		21	Tool no.	TT: Break tolerance in length LBREAK
		22	Tool no.	TT: Break tolerance in radius RBREAK
		23	Tool no.	PLC value
Data from the pocket table				
	51	1	Pocket number	Tool number
		2	Pocket number	0 = not a special tool 1 = special tool
		3	Pocket number	0 = not a fixed pocket 1 = fixed pocket
		4	Pocket number	0 = not a locked pocket 1 = locked pocket
		5	Pocket number	PLC status
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**)
Values programmed in TOOL CALL				
	60	1	–	Tool number T
		2	–	Active tool number 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 in tool radius DR2
		8	–	Tool index

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
Position programmed in TOOL CALL				
	70	1	–	1 = Valid position
		2	1	Position in X axis
			2	Position in Y axis
			3	Position in Z axis
		3	–	Feed rate (-1 = no feed rate programmed)
Tool compensation				
	200	1	–	Active radius (including oversizes) with algebraic signs
		2	–	Active length (including oversizes)
Coordinate transformation				
	210	1	–	Basic rotation (manual)
		2	–	Programmed rotation
		3	–	Active mirrored axes 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 the spindle and the tilted coordinate system
	214	8	–	Tolerance programmed in 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

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
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
Nominal position in the REF system				
	240	1	1 to 9 (X, Y, Z, A, B, C, U, V, W)	
Current position in the active coordinate system				
	270	1	1 to 9 (X, Y, Z, A, B, C, U, V, W)	
M128 active				
	280	1	–	–1 = M128 active, 0 = M128 not active
		2	–	Feed rate programmed with M128
Tilting axes				
	290	1	–	Current tilting axis geometry description
		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
TS touch-trigger probe				
	350	10	–	Tool axis
		11	–	Effective radius
		12	–	Effective length
		13	–	Radius of calibration ring
		14	1	Center offset (reference axis)
			2	Center offset (minor axis)
		15	–	Direction of the center offset with respect to spindle 0°

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
TT touch probe for tool measurement				
	350	20	1	Center of axis 1
			2	Center of axis 2
			3	Center of 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 probe length compensation, but with probe radius compensation (workpiece coordinate system)
		2	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 probe length or probe radius compensation (machine coordinate system)
		3	–	Measurement result of touch probe cycles 0 and 1 without probe radius or length compensation
Read values from active datum table				
	500	Line	Column	Read values
	501	Line	Column	Read REF values
	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 data item
Write values from active pallet table				
	510	1	–	Active lines
		2	–	Pallet number from column **Name**
		3	–	Active line 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 safety height in a pallet table for tool-oriented machining
Touch probe cycles				
	990	1	–	Approach behavior 0 = Standard behavior 1 = Effective radius, safety clearance 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
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 (not for machine parameters for which a formula must be entered)
	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 data item
PLC data				
	2000	10	Marker no.	PLC markers
		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

8.5.5 Data Transfer of Machine Parameters → PLC

In the PLC there are 122 machine parameters reserved for data transfer to the PLC. The iTNC saves the contents of machine parameters 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. You must evaluate the transmitted 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
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

		Set	Reset
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
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

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)
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

MP4310.0-6 Setting a number in the PLC (W976 to W988, M4300 to M4411)
Input: 10 to 30 000

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

8.5.6 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

Operator	Function
==	Equal
!= or <>	Not equal
<	Less than
>	Greater than
<=	Less than or equal
>=	Greater than or equal

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>255
Continue the NC program if the content of B3000 is greater than 255.

8.6 Operands

8.6.1 Operand Overview

Operand	Short designation	Address range
Marker	M	**M0 to M9999** **M0 to M999** are free. They are deleted only after entering the code number 531210, not during a reset (nonvolatile area). The range can be reduce in the *.CFG file of the PLC compiler. **M1000 to M3999** free, are deleted upon reset **M4000 to M5999** reserved for NC/PLC interface **M6000 to M9999** are free. They are deleted during reset.
Input	I	**I0 to I31** (MC 422) **I128 to I152** (machine operating panel) **I64 to I127** (first PL input/output board) **I192 to I255** (second PL) **I256 to I319** (third PL) **I320 to I383** (fourth PL)
Output	O	**O0 to O30** (MC 422) **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	Set counter: **C0 to C47** Counter contents: **C48 to C95** Counter pulse release: **C96 to C143**
Timer	T	Timer start: **T0 to T47** Timer is running: **T48 to T95 and T96 to T999**
Byte Word Double word	B W D	**B0 to B9999** (8 bits) **B0 to B127** are free. They are deleted only after entering the code number 531210, not during a reset (nonvolatile area). The range can be reduce in the *.CFG file of the PLC compiler. **B128 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 S15**

8.6.2 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).

8.6.3 Timers

The PLC has over 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.

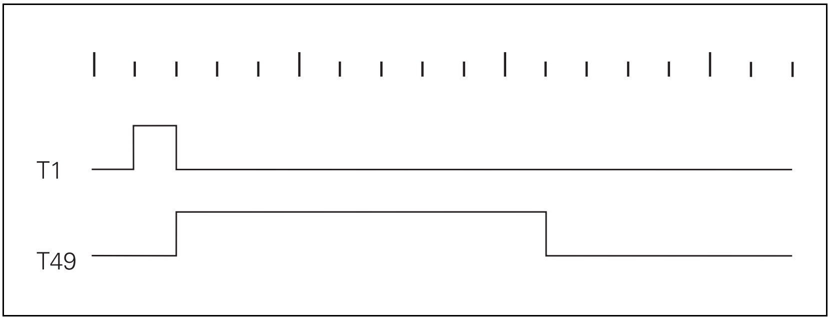
You can start the first 48 timers by setting one of the timers T0 to T47 for at most one PLC scan (otherwise the iTNC restarts the timer with the negative edge for each additional scan). The iTNC reserves the timer with the duration defined in MP4110.x, and sets the markers T48 to T95 (timer is running) until the defined duration has expired.

You can also set and start the timers T0 to T47 with Module 9006. Timers T96 to T999 can be started only through Module 9006.

Module 9197 can define and start cyclic timers (> T96). They are reset for a PLC cycle and are then automatically restarted.

Example:

Start of timer 1
Run time in MP4110.1 = 9 (PLC cycles)



Timer starts	**Timer is running**	**Machine parameters**
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

Timer starts	Timer is running	Machine parameters
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

With Module 9006 you can set the cycle 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 detection:

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 detection:

Marker	Value	Meaning
M4203	0	Timer started
	1	Error. See W1022.
W1022	1	Excessive run time
	3	Invalid timer number

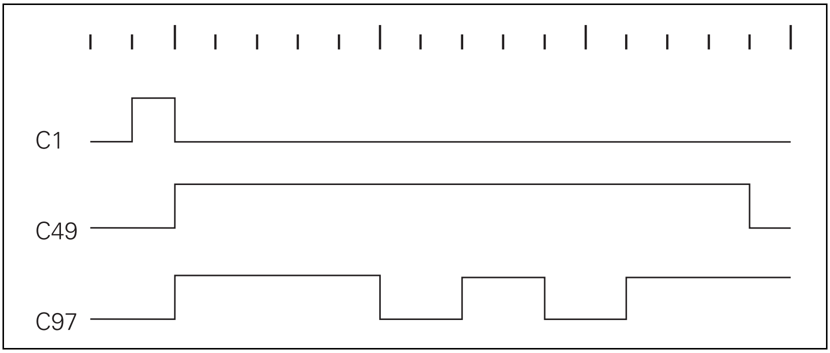
8.6.4 Counters

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)



Counter is set	Counter is running	Counter is started	Machine parameters
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
C15	C63	C111	MP4120.15
C16	C64	C112	MP4120.16
C17	C65	C113	MP4120.17

Counter is set	Counter is running	Counter is started	Machine parameters
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]

8.6.5 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 **Numerical designation for fast PLC inputs**
Input: 0 to 255 [no. of the PLC input]
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

Warning

Only the PLC inputs of the MC 422 can be defined as fast PLC inputs, and not the inputs on a PL.

8.7 Program Creation

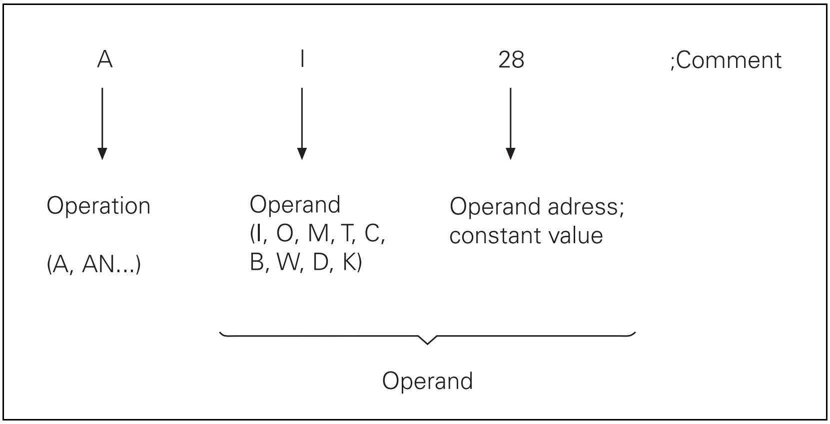
8.7.1 ASCII Editor

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.

8.7.2 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 says 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).

8.7.3 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 or double word occupies four bytes on the stack; a marker, input, output, timer or counter occupies two bytes.

Call:

```
CM      9019
PL      B/W/D      <Number bytes on processing stack>
```

8.7.4 Logical Names for Files

You can enter a logical name instead of a fixed file name. Logical names make the work easier for you, especially when you transfer file names to PLC modules.

Syntax: **>Group.name**

```
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 PLC 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

8.8 PLC Commands

8.8.1 Overview

The following table provides an overview of all commands explained in this chapter:

Group of functions	Syntax	Function
Loading and saving instructions		
	L	Load
	LN	Load NOT
	L–	Load two’s complement
	LB	Load BYTE
	LW	Load WORD
	LD	Load double word
	=	Assign
	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 instructions		
	+	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
	<	Less than
	>	Greater than
	<=	Less than or equal
	>=	Greater than or equal
	<>	Not equal
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 [ ]
	<[ ]	Less than [ ]
	>[ ]	Greater than [ ]
	<=[ ]	Less than or equal [ ]
	>=[ ]	Greater than or equal [ ]
	<>[ ]	Equal [ ]
Shifting instructions		
	<<	Shift left
	>>	Shift right
Bit commands		
	BS	Bit set
	BC	Bit reset
	BT	Bit test

Group of functions	Syntax	Function
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

8.8.2 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 accu.	Operand content
Load the operand content into the logic accumulator.	L I4	Logic accu. = 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 execution 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**	**Accu. 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

8.8.3 LOAD NOT (LN)

Logic processing with the LOAD NOT command

Syntax: LN (LOAD NOT)

Operands: M, I, O, T, C

Action:

Load the 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	Accu. 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	Accu. 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

8.8.4 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 execution.

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	Accu. 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

8.8.5 LOAD BYTE (LB)

Syntax: LB (LOAD BYTE)

Operands: M, I, O, T, C

Action:

Copy 8 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 (8th) operand becomes 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	Accu. 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**	1 1 1 0 0 1 1 0	0	1	1	0	0	1	1	1
			7	6	5	4	3	2	1	0
Assign accumulator contents to byte 8.	**= B8**	1 1 1 0 0 1 1 0	1	1	1	0	0	1	1	0

8.8.6 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 becomes 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.

8.8.7 LOAD DOUBLE WORD (LD)

Syntax: LD (LOAD DOUBLE WORD)

Operands: M, I, O, T, C

Action:

Copy 32 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 (32nd) operand becomes 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.

8.8.8 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	Accu. content	Operand content
Load the operand content into the logic accu.	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 execution, 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	Accu. 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

8.8.9 ASSIGN BYTE (B=)

Syntax: B= (STORE BYTE)

Operands: M, I, O, T, C

Action:

Assign 8 markers from the word accumulator to 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 the example for the command W=. The command B= is used in the same manner as the command W=, except that the iTNC processes eight operands.

8.8.10 ASSIGN WORD (W=)

Syntax: W= (STORE WORD)

Operands: M, I, O, T, C

Action:

Assign 16 markers from the word accumulator to 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	Accu. content	Operand content
Load content of word W8 into the word accumulator.	**L W8**	36FF	
			O16 ... O1
Assign accumulator content to outputs O5 to O20.	**W= O1**	36FF	0 1 1 0 1 1 0 1 1 1 1 1 1 1 1

8.8.11 ASSIGN DOUBLE WORD (D=)

Syntax: D= (STORE DOUBLE WORD)

Operands: M, I, O, T, C

Action:

Assign 32 markers from the word accumulator to 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 the example for the command W=. The command D= is used in the same manner as the command W=, except that the iTNC processes 32 operands.

8.8.12 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 procedure, see the example for the command ASSIGN (=).

Word processing

Syntax: =N (STORE NOT)

Operands: B, W, D

Action:

Assign the complement of the word accumulator to the addressed operand. For procedure, see the example for the command ASSIGN (=).

8.8.13 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 procedure, see the example for the command ASSIGN (=).

8.8.14 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 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 1, then set output O2 and marker M500.
Initial state:

Input	I4	= 1
Input	I5	= 0
Output	O2	= ?
Marker	M500	= ?

Function	STL	Accu. content	Operand content
Load the operand content into the logic accu.	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

8.8.15 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 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 1, then reset output O2 and marker M500.
Initial state:
Input I4 = 1
Input I5 = 0
Output O2 = ?
Marker M500 = ?

Function	STL	Accu. content	Operand content
Load the operand content into the logic accu.	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

8.8.16 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	Accu. content	Operand content
Load the operand content into the logic accu.	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

8.8.17 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	Accu. content	Operand content
Load the operand content into the logic accu.	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

8.8.18 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 begin 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	Accu. content	Operand content
Load the operand content into the logic accu.	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 content 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	Accu. 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

8.8.19 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. However, 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	Accu. content	Operand content
Load the operand content into the logic accu.	L I4	1	1
Gate the content of logic accumulator and input I5 with AND NOT.	AN I5	1	1
Assign the gating result to output O2.	= O2	1	1

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	Accu. content	Operand content
Load W6 into the word accumulator.	L W6	3C36	3C36
Gate the content of word accumulator and word W4 with AND NOT.	AN W4	814	36AA
Assign the gating result to word W8.	= W8	814	814

8.8.20 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. However, 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	Accu. content	Operand content
Load the operand content into the logic accu.	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 ODER 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	Accu. 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

8.8.21 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. However, 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	Accu. content	Operand content
Load the operand content into the logic accu.	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	Accu. content	Operand content
Load W6 into the word accumulator.	L W6	3C36	3C36
Gate the content of word accumulator and word W4 with OR NOT.	ON W4	814	36AA
Assign the gating result to word W8.	= W8	814	814

8.8.22 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. However, 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	Accu. content	Operand content
Load the operand content into the logic accu.	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 content of byte S B5 and 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	Accu. 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

8.8.23 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. However, 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	Accu. content	Operand content
Load the operand content into the logic accu.	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	Accu. 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	FFFFF563	36AA
Assign the gating result to word W8.	= W8	FFFFF563	FFFFF563

8.8.24 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	Accu. 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

8.8.25 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	Accu. 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

8.8.26 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	Accu. 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

8.8.27 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	Accu. 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

8.8.28 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	Accu. 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

8.8.29 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.

8.8.30 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.

8.8.31 EQUAL TO (==)

Syntax: == (EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Gate 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	Accu. content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Gate 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

8.8.32 LESS THAN (<)

Syntax: < (LESS THAN)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Gate 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	Accu. 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

8.8.33 GREATER THAN (>)

Syntax: > (GREATER THAN)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Gate 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	Accu. 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

8.8.34 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. Gate 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	Accu. 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

8.8.35 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. Gate 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	Accu. 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

8.8.36 NOT EQUAL (<>)

Syntax: <> (NOT EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Gate 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	Accu. 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

8.8.37 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.

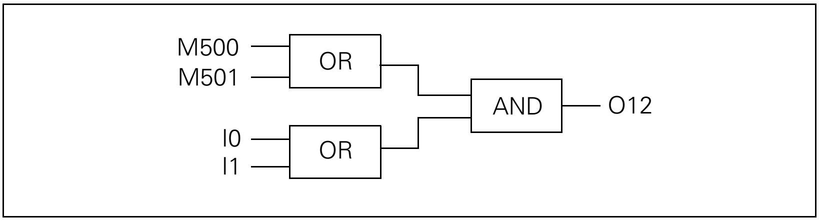
Please note:

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**	**Accu. 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

8.8.38 AND NOT [ ] (AN[ ])

Syntax: AN[ ] (AND NOT [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

8.8.39 OR [ ] (O[ ])

Syntax: O[ ] (OR [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

8.8.40 OR NOT [ ] (ON[ ])

Syntax: ON[ ] (OR NOT [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

8.8.41 EXCLUSIVE OR [ ] (XO[ ])

Syntax: XO[ ] (EXCL: OR [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

8.8.42 EXCLUSIVE OR NOT [ ] (XON[ ])

Syntax: XON[ ] (EXCL: OR NOT [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

8.8.43 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 wordD100 = ?

Function	STL	Accu. content	Operand content
Load the double word D12 into the word accu.	L D12	15000	15000
Opening parenthesis: Buffer the accumulator content onto the program stack.	+[		
Load the constant K 1000 into the word accumulator.	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

8.8.44 SUBTRACTION [ ] (–[ ])

Syntax: –[ ] (MINUS –[ ])

Operands: None

Action:

See example for ADDITION [ ]

8.8.45 MULTIPLICATION [ ] (X[ ])

Syntax: x[ ] (MULTIPLY [ ])

Operands: None

Action:

See example for ADDITION [ ]

8.8.46 DIVISION [ ] (/[ ])

Syntax: /[ ] (DIVIDE [ ])

Operands: None

Action:

See example for ADDITION [ ]

8.8.47 REMAINDER [ ] (MOD[ ])

Syntax: MOD[ ] (MODULO [ ])

Operands: None

Action:

See example for ADDITION [ ]

8.8.48 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.

See next page for example.

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**	**Accu. 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 W36.	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

8.8.49 LESS THAN [ ] (<[ ])

Syntax: <[ ] (LESS THAN [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

8.8.50 GREATER THAN [ ] (>[ ])

Syntax: >[ ] (GREATER THAN [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

8.8.51 LESS THAN OR EQUAL TO [ ] (<=[ ])

Syntax: <=[ ] (LESS EQUAL [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

8.8.52 GREATER THAN OR EQUAL TO [ ] (>=[ ])

Syntax: >=[ ] (GREATER EQUAL [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

8.8.53 NOT EQUAL [ ] (<>[ ])

Syntax: <>[ ] (NOT EQUAL [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

8.8.54 SHIFT LEFT (<<)

Syntax: << (SHIFT LEFT)

Operands: B, W, D, K

Action:

A SHIFT LEFT instruction multiplies the content of the word accumulator by two. This is done by simply shifting the bits 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 instruction 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 12.
Initial state:
Double word D8 = 3E 80 (hex)
Double word D12 = ?

Function	STL	Accu. content	Operand content
Load the double word D8 into the word accu.	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.

8.8.55 SHIFT RIGHT (>>)

Syntax: >> (SHIFT RIGHT)

Operands: B, W, D, K

Action:

A SHIFT RIGHT instruction 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 instruction 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	Accu. content	Operand content
Load the double word D8 into the word accu.	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.

8.8.56 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	Accu. content	Operand content
Load the double word D8 into the word accu.	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

8.8.57 BIT RESET (BC)

Syntax: BC (BIT CLEAR)

Operands: B, W, D, K, X

Action:

With the BIT RESET 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	Accu. content	Operand content
Load the double word D8 into the word accu.	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

8.8.58 BIT TEST (BT)

Syntax: BT (BIT TEST)

Operands: B, W, D, K, X

Action:

With the BIT TEST 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	Accu. content	Operand content
Load the double word D8 into the word accu.	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

8.8.59 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.

Memory assignment in the 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 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.

8.8.60 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.

8.8.61 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.

Memory assignment in the 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.

8.8.62 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 contents prior to the module call: 1A 44 3E 18

Function	**STL**	**Accu. 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

8.8.63 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.

8.8.64 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.

8.8.65 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. JP interrupts a logic sequence.

Example:

See JPT command.

8.8.66 JUMP IF LOGIC ACCUMULATOR = 1 (JPF)

Syntax: JPT (JUMP IF TRUE)

Operands: Label (LBL)

Action:

The JPT command is a conditional jump instruction. 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. JPT interrupts a logic sequence.

Example:

This example also applies to the commands JP and JPF.

Depending on the state of the input I5, skip a certain program section.
Initial state:
Input I5 = 1

Function	STL	Accu. content	Operand content
Load the operand content into the logic accu.	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

8.8.67 JUMP IF LOGIC ACCUMULATOR = 0 (JPF)

Syntax: JPT (JUMP IF FALSE)

Operands: Label (LBL)

Action:

The JPF command is a conditional jump instruction. 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. JPF interrupts a logic sequence.

Example:

See JPT command.

8.8.68 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. CM interrupts a logic sequence.

Example:

See command CMF.

8.8.69 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. CMT interrupts a logic sequence.

Example:

See command CMF.

8.8.70 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. CMF interrupts a logic sequence.

Example:

This example also applies to the commands CM and CMT.

Depending on the state of the input I5, call the Module 10. Initial state:
Input I5 = 0

Function	STL	Accu. content	Operand content
Load the operand content into the logic accu.	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 the 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		

8.8.71 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 command EM. 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 command EM as program end. The iTNC can reach the subsequent program instructions only through a jump instruction.

8.8.72 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.

8.8.73 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.

8.8.74 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 instruction.

8.9 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. Go into 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^X Substring 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 accu 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)	

8.10 Commands for String Processing

String processing enables you use the PLC program to generate and manipulate any texts. With Module 9082 you can display these texts in the PLC window of the screen and delete them again with Module 9080. The iTNC features one string accumulator and 16 string memories (S0 to S15), 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

Example

String accumulator (characters)	
1	128
C O O L A N T O 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 with 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 S15).
- 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 999)
  `PLC-DIALOG: S#Dxx xx:` Line number from the PLC dialog file (0 to 999).
  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 no longer checks 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 999), but not the actual text as in an immediate string.

The processing times depend on the length of the strings.

8.10.1 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 command OVWR.

8.10.2 ADD STRING (+)

Syntax: + (PLUS)

Operands: S <arg>

Action:

Attach 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 command OVWR.

8.10.3 SAVING A 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 15 (String memory S0 to S15). See also "Operand declaration."

Example:

See command OVWR.

8.10.4 OVERWRITING A 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 15 (string memory S0 to S15). 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		= **HYDRAUL.**
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 string into the string accu.	**L S "HYDRAUL."**	H Y D R A U 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 state:

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	
...	...	

8.10.5 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 <>.

8.10.6 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 <>.

8.10.7 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 <>.

8.10.8 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 <>.

8.10.9 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 <>.

8.10.10 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 string into the string accu.	**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

8.10.11 Modules for String Processing

Module 9070 Copy a number from a string

The iTNC searches a selectable string memory (S0 to S15) 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 a period or 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 detection:

Marker	Value	Meaning
M4203	0	Number was 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 [bits]
	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 S15).

Call:
PS K/B/W/D/S<String no. or string>
CM 9071
PL B/W/D <Length of the string>

Error detection:

Marker	Value	Meaning
M4203	0	String length was ascertained
	1	Error. See W1022.
W1022	2	Invalid immediate strings, address of the source or target string is out of range (S0 to S15), string memory was searched but no string end was found

8.11 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. 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 you can edit the same markers as those in the PLC sequential program without interfering in its execution. No exact times can be stated for the commands for managing the submit queue.

8.11.1 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 command CAN.

8.11.2 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 command CAN.

8.11.3 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 commands SUBM and RPLY.

Depending on input I10, submit the subroutine with the label LBL 300 to the NC for processing. In addition, check the execution 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 state of input I10 into the logic accu.	**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 accu.	**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 the 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.

8.12 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.

8.12.1 STARTING A PARALLEL PROCESS (SPAWN)

Syntax: SPAWN <label>

Operands: D

Action:

In the specified double word, the iTNC returns the identifier. See page 8 – 132. 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 eight parallel processes are permitted). If a process ends with EM, the iTNC removes it from the memory to provide space.

8.12.2 Control of Events

The parallel processes can make events available to one another. This saves computing 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 8 – 28). 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.

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 detection:

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 detection:

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 detection:

Marker	Value	Meaning
M4203	0	Context was changed
	1	Error code in W1022
W1022	30	Module was not called in a spawn job 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 detection:

Marker	Value	Meaning
M4203	0	Process waited
	1	Error code in W1022
W1022	30	Module was not called in a spawn job or submit job

8.13 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 the 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>.

8.14 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 instructions IFT, IFF, WHILET, WHILEF, ENDW, UNTILT and UNTILF require a valid gating result in the logic accumulator. You conclude the sequence of gating operations. The instructions ELSE, ENDI and REPEAT require that all previous operations sequences have been concluded.

8.14.1 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**

8.14.2 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**

8.14.3 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 instruction 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 instruction!

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**

8.14.4 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 command ENDC.

End of indexed module call (ENDC)

Syntax: ENDC (ENDCASE)

Operands: None

Action:

You 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**

8.15 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 instructions must be located at the beginning of your PLC program—i.e., before the first PLC command. With the USES instruction you link another file into the program. The GLOBAL instruction supplies a label from its own file for an entry that can be used by all other files. The EXTERN instruction 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. If you use more than one file, the main program must have the status flag "M" in the directory. This is done in the RAM by using the PLC program function "COMPILE" once and selecting the main program in the file window.

8.15.1 USES INSTRUCTION (USES)

Syntax: USES <file name>

Operands: None

Action:

You can use the USES instruction in the main program to link other files. Files that are linked with USES can themselves also use the instruction to link further files. It is also permissible to use the USES instruction to link a single file to several other files. The code for this file is generated only once. The USES instruction requires a file name as an argument. The USES instruction only links a file; it does not run the file's program code. The USES instruction cannot be compared with a CM instruction. The linked files must therefore contain individual modules that you can then call with the CM instruction.

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	⋮

8.15.2 GLOBAL INSTRUCTION (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 instruction 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.

8.15.3 EXTERN INSTRUCTION (USES)

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 instruction 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 instruction in a CASE branch

The name of the external label cannot be used again as a local label in the same file.

8.16 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 page5 – 2)

If the iTNC runs a module unsuccessfully, it sets marker 4203. You then can evaluate this marker to display an error message.

8.16.1 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 detection:

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 destination 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 detection:

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 to the data stack.	**PS B10**	35	35
Read byte B35 and save to 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 detection:

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 to 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 in byte B35.	**CM 9020**	120	

8.16.2 Number Conversion

Module 9050 Conversion of binary numbers → ASCII

Module 9050 converts a binary numerical value consisting of a mantissa and exponent to base 10 into a 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 detection:

Marker	Value	Meaning
M4203	0	Number was converted
	1	For error see W1022
W1022	2	Invalid string address or invalid exponent

Module 9051 Conversion of binary numbers → 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-justified 1: Sign left-justified, number right-justified 10: Sign and number right-justified 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 detection:

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 of ASCII numbers → 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 TNC saves the ASCII-coded decimal number>
CM	9052	
PL	B/W/D	<Numerical value>
PL	B/W/D	<Exponent to the base of 10 of a value>

Error detection:

Marker	Value	Meaning
M4203	0	Number was converted
	1	For error see W1022
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 detection:

Marker	Value	Meaning
M4203	0	Number was converted
	1	For error see W1022
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 detection:

Marker	Value	Meaning
M4203	0	Number was converted
	1	For error see W1022
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), or logic accu. [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 in the binary number 99.	**CM 9054**	01100011

9 Data Interfaces

9 Data Interfaces

9.1 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 floppy-disk drives and hard disks
- Other computer systems

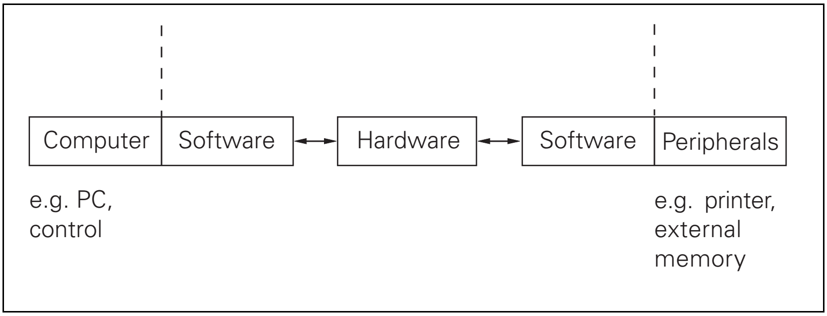
A data interface makes it possible for the CPU and its peripheral devices to communicate.

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:

- 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 construction
- Pin layout
- Electrical characteristics

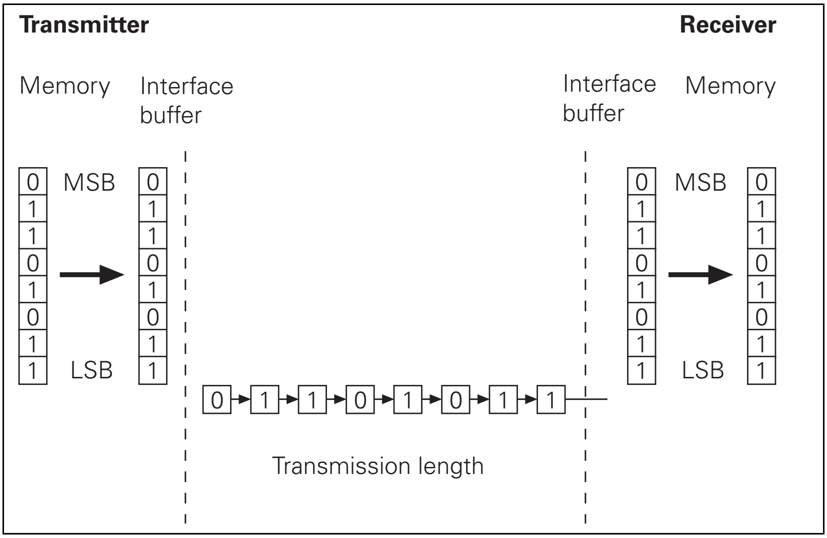
The software is the operating software, which includes, for example, the drivers for the output modules.

9.1.1 Principles of Data Transfer

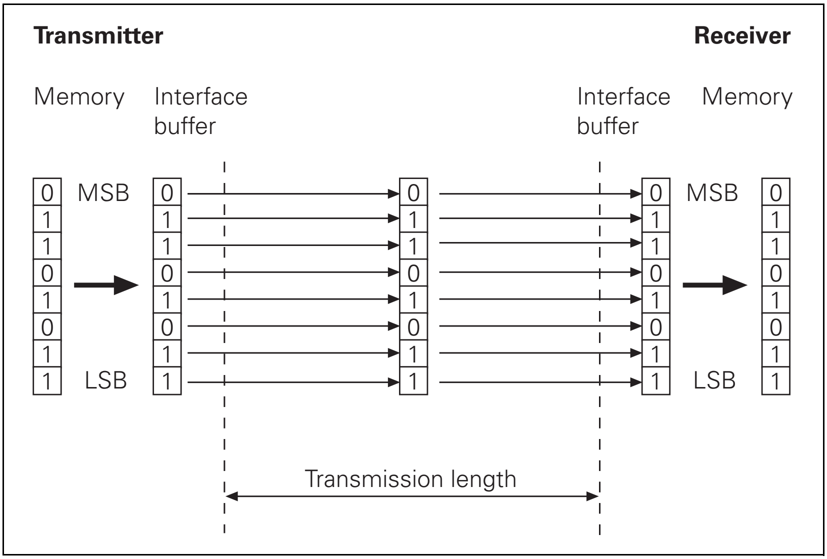
The term "data" is used to describe all of the information that the computer is capable of collecting and processing.

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:

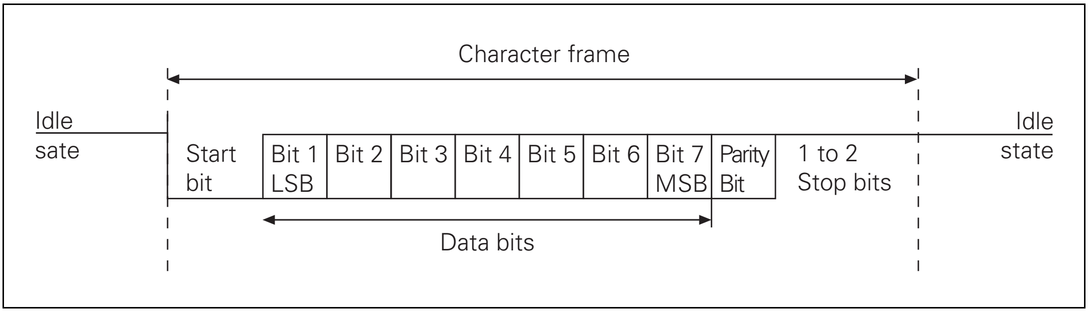
1 1 1 1 0 1 0 = 122 dec = 7A hex

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: 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

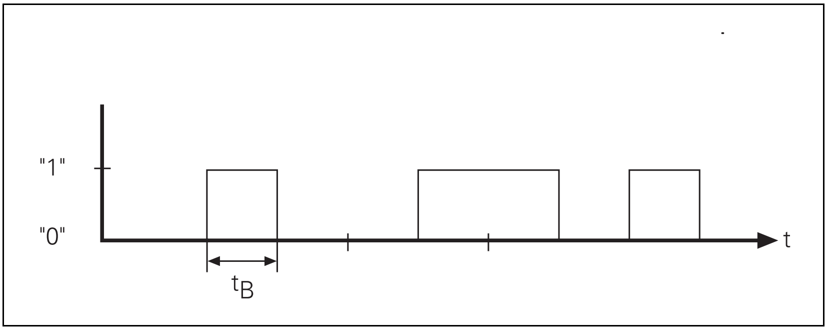
Data transfer rate

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, 115 200 bps

The time taken to transmit one bit (t_B) can be calculated from the transfer rate:



$$t_B = \frac{1}{\text{Baud rate (Bit/s)}}$$

For example, a transfer rate of 19 200 bps will have a bit duration of t_B = 52.083 µs.

$$t_B = \frac{1}{19200\ \text{(Bit/s)}} = 52{,}083\ \mu s$$

The number of characters transmitted per second can be calculated from the transfer rate and the transmission format:

$$\text{Characters per second} = \frac{\text{Baud rate (Bit/s)}}{\text{Number of bits per character}}$$

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.

$$\text{Characters per second} = \frac{300\ \text{(Bit/s)}}{1 + 7 + 2} = 30$$

9.1.2 Data Transfer Check: Handshaking

By handshaking, two devices control data transfer between them. A distinction is drawn between “software” 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 (switching on transmitting unit)
- 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

9.2 iTNC Serial Data Interfaces

9.2.1 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 Transmission Protocol
- Transmission protocol with Block Check Character (BCC)
- LSV2 transmission protocol

9.2.2 RS-232-C/V.24 Interface

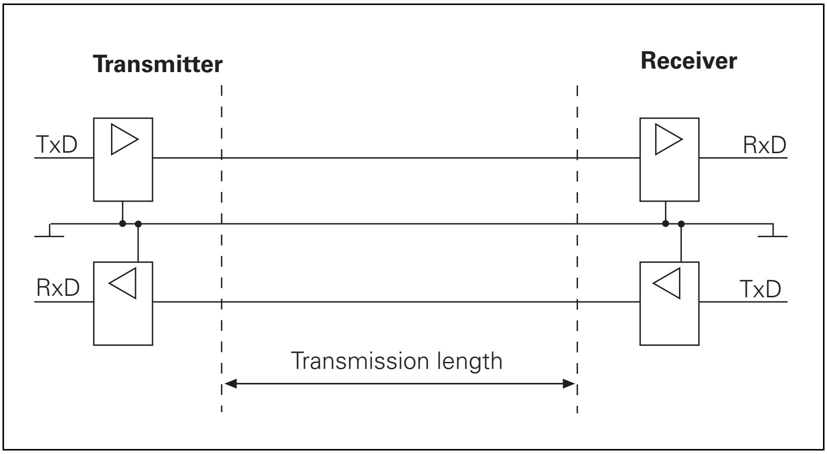
RS-232-C/V.24 is the designation for a serial interface. Data transfer is executed asynchronously, with a start bit before each character and one or two stop bits after each character.

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.

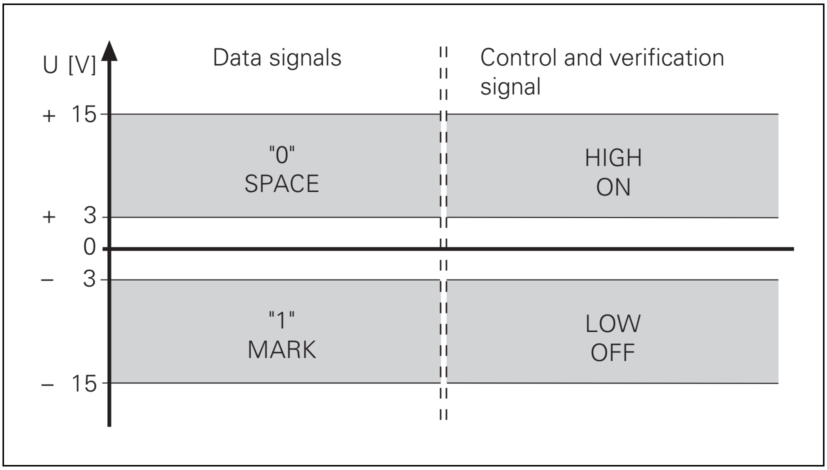
Physical connections:



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 designations

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 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 layouts

Keep in mind that there might be a difference between the pin layout of the MC 422 and the adapter block.

9.2.3 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

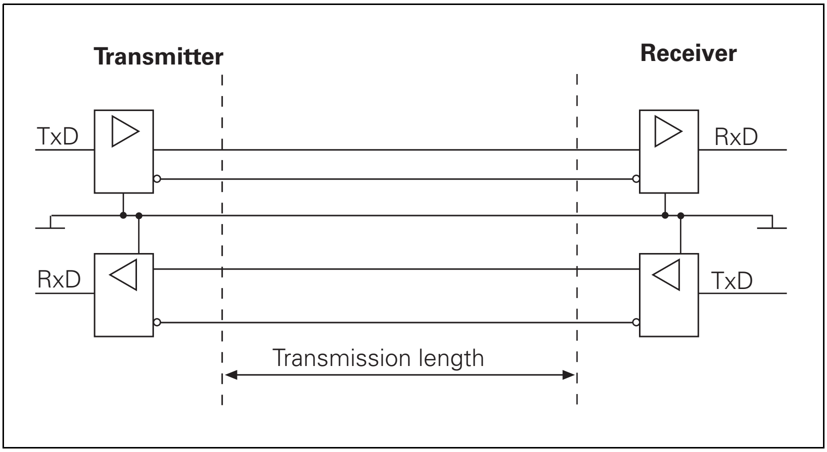
Hardware

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

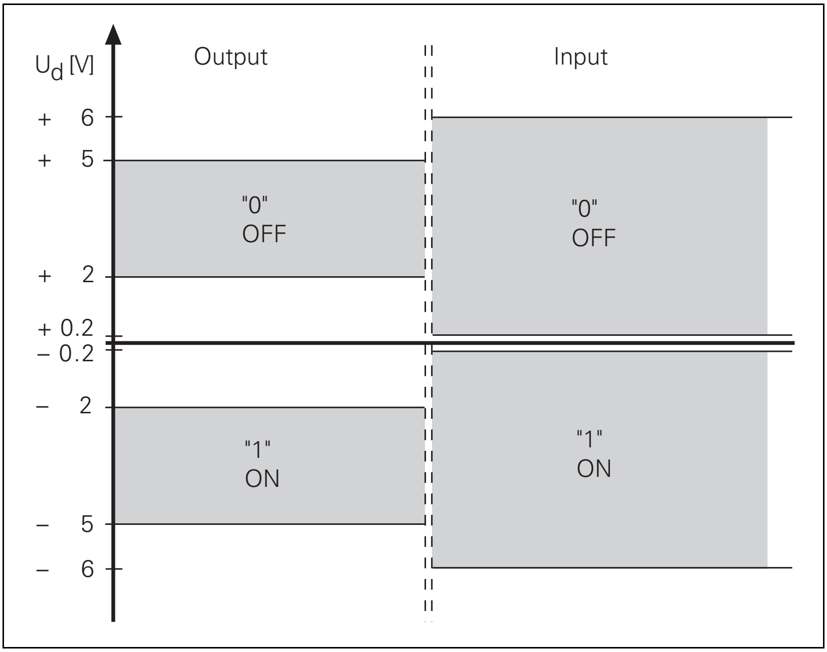
The signals are both transmitted and received as differential voltage.

A positive differential voltage corresponds to logic zero (OFF).

A negative differential voltage corresponds to logic one (ON).

V_{dmin} = 2 V and V_{dmax} = 5 V

The control unit detects the differential voltages between V_{dmin} = 0.2 and V_{dmax} = 6 V as a logically defined level.



Signal designations

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 layouts

The MC 422 and adapter block have the same pin layout.

9.3 Configuration of Interfaces

9.3.1 Control Characters

Overview of control characters specific to HEIDENHAIN

Character	Designation	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 transferred without error.
NAK	Negative Acknowledgment	Transmitted by the receiver when a data block has transferred with an error. The transmitter must re-transmit the data block.

9.3.2 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

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).

9.3.3 Configuration of Interfaces

To configure data format and the type of handshake in the EXT1/EXT2/EXT3 operating modes (EXT3 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)

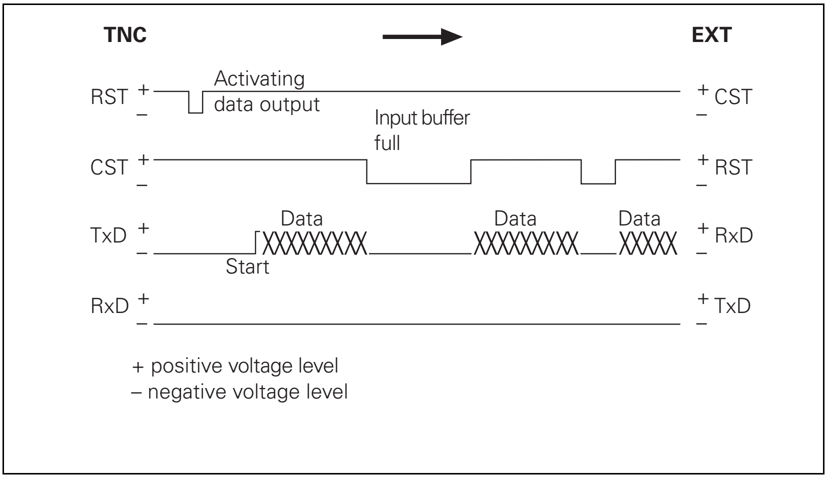
With **bit 1** you can ensure that the BCC is not interpreted as a control character.

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.

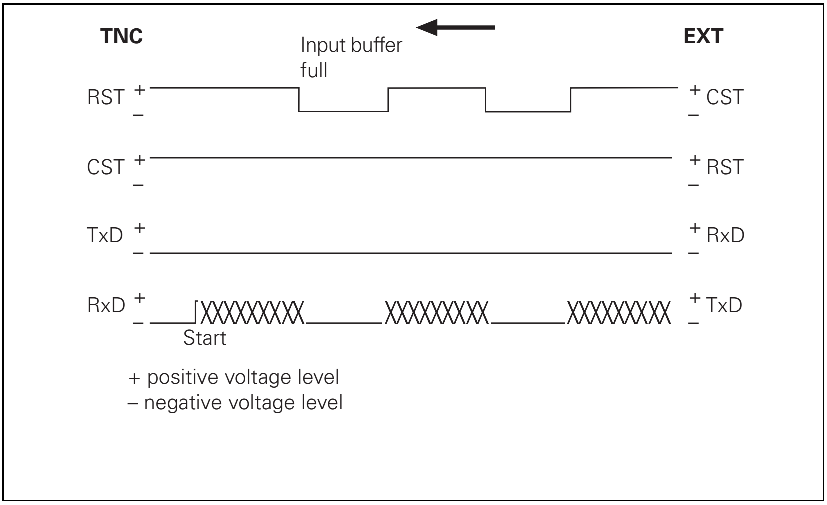
Hardware handshaking

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 logic 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 handshake 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.

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

MP5020.0 Operating mode EXT1
MP5020.1 Operating mode EXT2
MP5020.2 Operating mode EXT3 (PLC)

Transmission protocol

The transmission protocol for operating modes EXT1/EXT2/EXT3 is defined with MP5030:

MP5030 **Data transfer 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)

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 parameter	**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.

9.4 Data Transmission Protocols

9.4.1 Selection of Transmission Protocols

The operating modes are assigned the following transmission protocols:

Operating modes	Transmission protocol
FE1	Select a protocol with BCC and with fixed control characters 1 start bit, 7 data bits, 1 stop bit
EXT1, EXT2, EXT3	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 error message **Protected file!**:

▶ 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.

9.4.2 Standard Transmission Protocol

General Information

To set the standard data transmission protocol in the operating modes EXT1/EXT2/EXT3:

▶ Select MP5030.0-2 = 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 transfer 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>	End of program
...	...
<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>... 1st line of file <CR> <LF>... 5th line of file <CR> <LF>	 Transmission stop: <DC3> Resume transmission: <DC1>
6th line of file <CR> <LF>... Last line of file <CR> <LF>	...

Read-in selected file

The EXT1 operating mode is set with software handshake.

To read-in 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> 1st 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.

9.4.3 Transmission 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 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.

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 dialog (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. With <NAK>, 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 and EXT3 modes:

- Set MP5020 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 letters><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.

Read-in 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>	

9.4.4 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 computer.
- Diagnosis of iTNC error messages and keystrokes for service purposes. The last 1000 events are stored in the iTNC.

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 (standard 3 s)
- LSV2TIME1 = Timeout for acknowledging ENQ or check sum (standard 3 s)
- LSV2TIME2 = Timeout when sending DLE 0, DLE 1 or NAK until a valid character is received (standard 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 standard values are used.

9.5 Saving/Reading Files

The table lists all the files that can be saved to external memory and read back in from them.

File	File extension	File code
NC program in HEIDENHAIN dialog	.H	H
NC program in ISO format	.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 HEIDENHAIN-Software **TNCremo** the file code has no significance. The files are saved on the PC with the same extension as on the TNC.

Note

Files that have no code (-) can only be transmitted with the LSV2 protocol of TNCremo.

9.6 Data Transfer by PLC

9.6.1 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 422 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 in EXT3 mode.

MP5040	**Data transfer rate in operating mode EXT3 (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: 115 200 bps

9.6.2 PLC Modules

With the following PLC modules you can operate 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 receive 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 transmit and receive buffers each hold 63 bytes.

Module 9100 Assign data interface
Module 9100 assigns an interface to the PLC and configures the transfer parameters. It initializes the interface, thereby erasing any errors that may have occurred. The interface is switched to receive mode.

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
CM    9100
```

Error detection:

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 already assigned 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 detection:

Marker	Value	Meaning
M4203	0	Interface was released
	1	Error code in W1022
W1022	1	Incorrect interface
	14	Interface not assigned
	20	Module was not called in a submit job or spawn 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 PLC
Bit 2: Interface is ready (see above)
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 still contains characters

Error detection:

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 detection:

Marker	Value	Meaning
M4203	0	String was transmitted
	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 detection:

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 detection:

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 submit job or spawn 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 to 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 detection:

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 receive 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	<Read binary value>

Error detection:

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 detection:

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 detection:

Marker	Value	Meaning
M4203	0	Message was 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 MP5030.2 = 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 detection:

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 sent 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 MP5030.2 < 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 a word at least 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       <Read ASCII character
                [0 to 255 = ASCII characters; –1 = error>
```

Error detection:

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	Receiver queue full

9.7 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.

9.8 Ethernet Interface

General Information

The iTNC requires an NFS server (Network File System) or a Windows PC as the remote station. It must work according to the TCP/IP protocol principle.

OSI 7-layer model		**iTNC**
7	Application layer	NFS
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.

iTNC settings

▶ Press the MOD key in the **`Programming and Editing`** operating mode and enter the code number NET123. With the soft keys you can select the required network options:

Soft key	Option	Meaning
DEFINE NET	Settings on the iTNC 530 for networking	
	`ADDRESS`	Network address of the iTNC: Enter as four decimal numbers separated by points (dotted-decimal notation). Your network supervisor of the iTNC can give you an internet address.
	`MASK`	Subnet mask; serves to distinguish the net and host ID of the network: Enter as four decimal numbers separated by points (dotted-decimal notation). Your network supervisor of the iTNC can give you a subnet-mask.
	`BROADCAST`	The broadcast address of the iTNC; is needed only if it is different from the standard setting: Enter as four decimal numbers separated by points (dotted-decimal notation). The standard setting is made of net and host ID, in which all bits are set to 1.
	`ROUTER`	Network address of default router: Enter as four decimal numbers separated by points (dotted-decimal notation). This entry is required only if your network consists of several subnetworks interconnected by router.
	`HOST`	Host name: This is the name used by the iTNC in the network. If you use a host-name server, you must enter the Fully Qualified Host Name here. If you leave this entry blank, the iTNC will use the so-called null authentication. If you work with null authentication, the entries under UID, GID, DCM and FCM will be ignored.
	`DOMAIN`	Enter the name and path of an ASCII file. In the ASCII file, names are assigned to the IP addresses of network drives. Then the names instead of the IP addresses can be used in the network settings. Example: **`DOMAIN = NET.A`** File NET.A: **`PC1 160.1.180.20`** **`PC2 160.1.180.21`** **`...`** In the column **`MOUNTDEVICE`** only the names **`PC1`**, **`PC2`** and so on are used.
	`NAMESERVER`	Network address of the domain server (at present without function)

Note

The protocol is not entered. The data transfer protocol as per RFC 894 is used.

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. Entry of the network address as four decimal numbers separated by points (dotted-decimal notation). When entering the path name, pay attention to capitalization. ■ 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 for data reception in bytes. Input range: 512 to 8192 **wsize**: Packet size for data transmission in bytes. 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 mounted automatically can be mounted at any time in the program management.

Soft key	Option	Meaning
	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	
	**TNC USER ID**	Definition of which user identification the end user uses to access files in the network. Your network specialist determines this value.
	OEM USER ID	Definition of which user identification the machine tool builder uses to access files in the network. Your network specialist determines this value.
	TNC GROUP ID	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.
	UID for mount	Definition of the user identification with which the logon process is executed. **USER**: The logon is with the USER identification. **ROOT:** The logon is with the identification of the ROOT user, value = 0.

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.

9.9 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 codeword **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 codenumber 807667. To permit this access only with the codenumber defined in OEM.SYS under **PLCPASSWORD =** (no longer with 807667):

- Enter the codeword **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 codenumber defined in OEM.SYS under **PLCPASSWORD =** (no longer with 807667):

- Enter the codeword **REMOTE.PLCPASSWORDFORCED = YES** in OEM.SYS.

10 Error Messages

10 Error Messages

10.1 DSP Error Messages

Error message	Cause	Corrective action	As of NC software
8010 Error in LSV2 transfer	■ Error in data transfer by LSV2 protocol.	■ Press the CE key to acknowledge the error. ■ Error does not impair the control functions. ■ Inform your service agency.	340 420-01 340 422-01
8040 Heat-sink temperature UV 1xx	■ Temperature of UV 1xx power supply unit's heat sink too high. ■ If the heat-sink temperature continues to increase, the unit will be switched off.	■ Stop the machine and let it cool down. ■ Continue working with lower power (reduce the feed rate).	340 420-01 340 422-01
8041 Iz of UV 1xx too high	■ DC-link current of UV 1xx too high.	■ Continue working with lower power (reduce the feed rate).	340 420-01 340 422-01
8042 Leakage current of UV 1xx too high	■ Isolation problem (e.g. defective motor).	■ Inform your service agency. • Check the motor. • Check the wiring.	340 420-01 340 422-01
8043 No inverter-ready signal	■ Readiness signal of the inverter (supply unit) is inactive after the feedback control starts. ■ Master contactor has opened. ■ Error in PLC program. ■ Inverter defective.	■ Try restarting the inverter. If the error recurs: ■ Inform your service agency. • Check the wiring (master contactor). • Check the PLC program. • Exchange the inverter (supply unit).	340 420-07 340 422-01
8080 Uz of UV 1xx too high	■ Excessive DC-link voltage of the power supply unit.	■ Inform your service agency. • Check the machine parameter for braking of the spindle. • If required, check the braking resistor. • Exchange the power supply unit.	340 420-01 340 422-01
8086 Probing already active	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
8092 Pos. contr. cyc. time error	■ MC is outputting erroneous cycle time for CC position controller. ■ A hardware error has occurred.	■ Inform your service agency. • Check machine parameter MP7600.x. • Exchange drive control board.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
8130 DIR <axis> motor table OK?	■ DIR in motor table may be incorrect.	■ Change DIR in motor table.	340 420-01
8130 Wrong rotational direction <axis>	■ Entry in column DIR of the motor table incorrect. ■ Error in wiring.	■ Inform your service agency. • Change DIR in motor table. • Check the wiring.	340 420-02
8130 Motor brake defective <axis>	■ Motor brake defective.	■ Inform your service agency. • Check controls for motor brakes. • Exchange motor.	340 420-07 340 422-01
8140 Error <axis> field orientation	■ Field orientation impossible for mechanical reasons. ■ Incorrect relation between electrical field and mechanical motor movement. ■ Error in motor encoder signal. ■ Error in motor connection. ■ Mechanical brake not released.	■ Inform your service agency. • Check the machine parameters for number of signal periods and distance for the number of signal periods. • Check the machine parameter for the linear distance of one motor revolution. • For linear motors: check column STR of the motor table. • Check the speed encoder connection. • Check the motor connection. • Release brakes during orientation.	340 420-01 340 422-01
8400 No drive-on command for <axis>	■ Speed controller waiting for drive-on command; PLC has sent no drive-on command.	■ Check the PLC program. ■ Inform your service agency. • Check software version.	340 420-01 340 422-01
8410 I2T value is too high <axis>	■ The load of the drive is too high for the duration.	■ Reduce load or duration. ■ Inform your service agency. • Check the motor table (columns I-N, T-DC, F-DC, T-AC, F-AC). • Check the power module table (columns I-N, I-N-DC, T-DC, F-DC, T-AC, F-AC). • Check the machine parameter for reference value for I2t monitoring. • Check the ratio of I-N (motor) to I-N (power module).	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
8430 Load is too high <axis>	■ Drive has maximum current and cannot increase acceleration. ■ Excessive load (torque, power) on the drive.	■ Reduce load. ■ Inform your service agency. • Check the motor table (column I-MAX). • Check the check the power module table (column I-MAX). • Check the machine parameters for maximum braking power, power limiting and maximum torque. • Check the ratio of I-N (motor) to I-N (power module).	340 420-01 340 422-01
8440 Field orient. successful <axis>	■ Field orientation successfully completed.	■ Acknowledge the message with CE.	340 420-01 340 422-01
8450 I2T value of motor is too high	■ The load of the motor is too high over the duration.	■ Reduce the load or the duration. ■ Check motor table and machine parameters. ■ Check whether the motor is designed for the load. ■ Inform your service agency.	340 420-07 340 422-01
8460 I2T value of power module is too high	■ The load of the power module is too high over the duration.	■ Reduce the load or the duration. ■ Check motor table and machine parameters. ■ Check whether the power module is designed for the load. ■ Inform your service agency.	340 420-07 340 422-01
8800 Signal LT-RDY inactive <axis>	■ Inverter switch-off during closed-loop control of a vertical axis (cause = vertical axis).	■ Inform your service agency. • Check the PLC program. • Check the wiring of the inverter.	340 420-01 340 422-01
8810 Signal LT-RDY inactive <axis>	■ Inverter switch-off during closed-loop control of a vertical axis (cause = vertical axis).	■ Inform your service agency. • Check the PLC program. • Check the wiring of the inverter.	340 420-01 340 422-01
8820 Field angle unknown <axis>	■ Field angle of the motor on the reference point of the speed encoder has not yet been ascertained.	■ Inform your service agency. • Orient the field. • Check the motor table (column SYS).	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
8830 EnDat: No field angle <axis>	■ Field angle of the motor with unaligned speed encoder with EnDat interface was not yet ascertained. ■ The transferred EnDat serial number does not match the stored EnDat serial number.	■ Inform your service agency. • Orient the field. • Check the motor table (column SYS).	340 420-01 340 422-01
8840 Axis not available <axis>	■ Starting command for unavailable axis.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
8850 Drive still active <axis>	■ Position definition (Z1 track, EnDat encoder) was started, although the drive was still active.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
8B00 Zn track <axis> error	■ Contamination of the motor encoder (Zn track). ■ Motor encoder cable is defective. ■ Motor control board defective.	■ Inform your service agency. • Exchange the motor. • Check the motor encoder cable. • Exchange the motor drive control board.	340 420-01 340 422-01
8B10 Traverse direction <axis> incorrect	■ DIR entry in motor table is incorrect. ■ Incorrect motor power connection.	■ Inform your service agency. • Check the DIR entry in the motor table. • Check the motor power connection.	340 420-01 340 422-01
8B20 Error <axis> field orientation	■ Field orientation impossible for mechanical reasons. ■ Incorrect relation between electrical field and mechanical motor movement. ■ Error in motor encoder signal. ■ Error in motor connection. ■ Mechanical brake not released.	■ Inform your service agency. • Check the machine parameters for number of signal periods and distance for the number of signal periods. • Check the machine parameter for the linear distance of one motor revolution. • For linear motors: check the motor table (column STR). • Check the speed encoder connection. • Check the motor connection. • Release brakes during orientation.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
8B30 Motor temperature <axis> too high	■ Measured motor temperature too high. ■ No temperature sensor. ■ Motor encoder cable is defective (wire broken). ■ Entry in motor table is incorrect. ■ Incorrect or defective temperature sensor was installed.	■ Let the motor cool down. ■ Inform your service agency. • Check the motor encoder cable. • Check the entry in the motor table. • Measure the temperature sensor (2000 [Ohm] at 25 [°C]).	340 420-01 340 422-01
8B40 No drive release <axis>	■ Inverter is not ready for operation. ■ No pulse release for the power module. ■ Uz too large. ■ Power-fail signal is active. ■ If M control: NE2 input is active. ■ If P control: drive release at X50 is inactive. ■ Motor control board defective. ■ PWM cable defective. ■ Noise pulses.	■ Inform your service agency. • Check the signal and cabling of the pulse release. • Check Uz. • Check the emergency stop circuit. • If the power supply is not regenerative: Is the braking resistor connected? • If the power supply is regenerative: Is energy recovery activated? • Exchange the power module. • For P controls: Exchange the interface card. • Exchange the motor drive control board.	340 420-01 340 422-01
8B50 Axis module <axis> not ready	■ No pulse release for the power module. ■ U_Z too high. ■ 5 V power supply too weak. ■ Inverter is not ready for operation. ■ Motor control board defective. ■ PWM cable defective. ■ Noise pulses.	■ Inform your service agency. • Check the signal and cabling of the pulse release. • Check U_Z. • If the power supply is not regenerative: Is the braking resistor connected? • If the power supply is regenerative: Is energy recovery activated? • Check the grounding and shielding of the cable. • Exchange the power module. • For P controls: Exchange the interface card. • Exchange the motor drive control board.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
8B60 Error in axis module<axis>	■ Undervoltage, temperature, or short-circuit monitor of an IGBT in the inverter has responded.	■ Let the inverter cool down. ■ Inform your service agency. • Examine the motor for a short circuit in the windings. • Exchange the power module.	340 420-01 340 422-01
8B70 External drive lock <axis>	■ Drive switch-on is blocked by one or more external signals.	■ Inform your service agency. • Check external enabling signals (EM.STOP, PFAIL, NO). • Check the PLC program. • Check external wiring.	340 420-01 340 422-01
8B80 External drive stop <axis>	■ Drive switched off by external signal.	■ Inform your service agency. • Check external enabling signals (EM.STOP, PFAIL, NO). • Check the PLC program. • Check external wiring.	340 420-01 340 422-01
8B90 No field orientation <axis>	■ No field orientation was performed. ■ Serial number of encoder has changed. ■ Field orientation was not possible.	■ Inform your service agency. • Orient the field.	340 420-01
8B90 Current regulator lock <axis>	■ Wrong motor model. ■ Line count or rotary encoder type incorrect.	■ Inform your service agency. • Check the type of motor in the motor table. • If UASM (U/f mode), then line count must be 0. • If UASM (U/f mode), then encoder type must be 0.	340 420-07 340 422-01
8BA0 Incorrect line count <axis>	■ Incorrect entry in motor table. ■ Faulty reference signal. ■ Noise pulses. ■ Encoder cable is defective.	■ Inform your service agency. • Check the entry in the motor table. • Check the motor encoder cable. • Exchange the motor encoder cable. • Exchange motor.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
8BC0 Motor current <axis> too high	■ Incorrect motor or power module selected. ■ Incorrect current controller parameters. ■ Incorrect parameters in the motor table. ■ Power module defective. ■ Motor cable defective (short circuit). ■ Motor defective (short circuit, ground fault). ■ Motor control board defective.	■ Inform your service agency. • Are the correct motor and power module selected? • Check the current control adjustment. • Check the motor and motor cable for a short circuit. • Exchange the power module or drive control board.	340 420-01 340 422-01
8BD0 Excessive following error in <axis>	■ The following error of a moved axis is greater than the value entered in machine parameter MP1720 (for following error mode) or MP1420 (for feedforward mode). ■ The acceleration entered is too large. ■ The motor is not moving even though drive-on was given.	■ Reduce machining feed rate, increase speed. ■ Remove possible sources of vibration. ■ If this occurs frequently: Inform your service agency. ■ Inform your service agency. • Check MP1060.x. • The motor current must not be limited during acceleration.	340 420-01 340 422-01
9800 MCU command unknown	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01
9800 MC command unknown <command>	■ MC command not permitted for this hardware. ■ MC command not currently permitted. ■ 0 = Faulty command code > 255 1...255 = Incorrect or invalid command code ■ Internal software error.	■ Inform your service agency. • Check software version.	340 420-04
9800 CC<Board> MC command unknown <command>	■ MC command not permitted for this hardware. ■ MC command not currently permitted. ■ 0 = Faulty command code > 255 1...255 = Incorrect or invalid command code ■ Internal software error.	■ Inform your service agency. • Check software version.	340 420-07 340 422-01

Error message	Cause	Corrective action	As of NC software
A000 Error during T2 test	■ Error during the test of emergency-stop loop 2.	■ Inform your service agency. • Check the wiring. • Check the emergency-stop key. • Replace the hardware.	340 420-01 only until 340 420-07
A080 Op. state MCU not equal CCU	■ The automatic SRG, SBH, and SH operating states of the MCU and CCU are compared cyclically. If the states are unalike for over 200 ms, a stop 1 is output.	■ Press CE to acknowledge the error message. ■ Switch on the machine. ■ Inform your service agency. • Check software version.	340 420-01 only until 340 420-07
A080 Op. state MC not equal CC	■ The automatic SRG, SBH, and SH operating states of the MCU and CCU are compared cyclically. If the states are unalike for over 200 ms, a stop 1 is output.	■ Press CE to acknowledge the error message. ■ Switch on the machine. ■ Inform your service agency. • Check software version.	340 420-04
AC00 Mot. enc. amp. too high <axis>	■ Noise on motor encoder signal. ■ Short circuit in motor encoder cable. ■ Motor encoder signal amplitude too high.	■ Inform your service agency. • Check connection of motor encoder (ground connection). • Check the motor encoder.	340 420-01
AC00 CC amplitude too high <axis>	■ The encoder signal amplitude is too high or the contamination signal is active. ■ Noise on motor encoder signal. ■ Short circuit in motor encoder cable. ■ Motor encoder signal amplitude too high.	■ Inform your service agency. • Check connection of motor encoder (ground connection). • Check the motor encoder.	340 420-07 340 422-01
AC10 amplitude too low <axis>	■ Interruption in motor encoder cable. ■ Motor encoder signal amplitude missing.	■ Inform your service agency. • Check connection of motor encoder. • Check the motor encoder.	340 420-01
AC10 CC amplitude too low <axis>	■ The encoder signal amplitude is too low or the contamination signal is active. ■ Interruption in motor encoder cable. ■ Motor encoder signal amplitude missing.	■ Inform your service agency. • Check connection of motor encoder (ground connection). • Check the motor encoder. • Check encoder signal amplitude.	340 420-07 340 422-01
AC20 <Axis> frequency too high	■ Noise on motor encoder signal.	■ Inform your service agency. • Check connection of motor encoder. • Check the motor encoder.	340 420-01

Error message	Cause	Corrective action	As of NC software
AC20 CC <axis> frequency too high	■ The maximum input frequency was exceeded at an encoder input. ■ Noise on motor encoder signal.	■ Inform your service agency. • Check connection of motor encoder (ground connection). • Check the motor encoder. • Check encoder signal input frequency.	340 420-07 340 422-01
B800 Safe inputs <input> not equal	■ Wiring error X65, X66 (and X67). ■ Safety module defective.	■ Inform your service agency. • Check the wiring X65, X66 (and X67). • Exchange the safety module.	340 420-01
B800 CC<Board> safe inputs <input> not equal	■ Safety-oriented input of the CCU > 400 ms not equal to the safety-oriented input of the MCU. ■ Different levels at the safety module input: 0 = Axis configuration A 1 = Axis configuration B 2 = Safe stop of spindle 3 = Permissive button of machine operating panel 4 = Acknowledgement of switch-off 5 = NC stop 6 = Spindle stop 7 = Permissive button of handwheel 8 = Safe reduced velocity of axes 9 = Safe reduced velocity of spindle 10 = Safe reduced velocity of auxiliary axes 11 = Safe controlled stop of axes 12 = Safe controlled stop of spindle 13 = Safe controlled stop of auxiliary axes or permissive button on tool changer 14 = Machine key is active 15 = Machine on ■ Wiring error X65, X66 (and X67). ■ Safety module defective.	■ Inform your service agency. • Check the wiring X65, X66 (and X67). • Exchange the safety module.	340 420-07 340 422-01

Error message	Cause	Corrective action	As of NC software
B900 Error in supply voltage	■ The Vcc supply voltage was out of range. ■ Excessive load from external components (e.g. encoder). ■ The power supply unit (UVxxx) is defective.	■ Inform your service agency. ■ Measure the supply voltage. • If below range (< 4.75 V): Check the encoder connections. • If above range (> 5.40 V): Exchange the power supply unit (UVxxx).	340 420-01
B900 Supply voltage <code>	■ The Vcc supply voltage (x) was out of range. ■ +4 = Undervoltage Vcc (+5 V) Excessive load from external components (e.g. encoders). ■ +6 = Undervoltage Vcc (+5 V) The power supply unit is defective. ■ +13 = Undervoltage Vcc (+15 V) The power supply unit is defective. ■ +17 = Undervoltage Vcc (+15 V) The power supply unit is defective. ■ –13 = Undervoltage Vcc (–15 V) The power supply unit is defective. ■ –17 = Undervoltage Vcc (–15 V) The power supply unit is defective.	■ Measure supply voltage Vcc (x). ■ Vcc (+5 V) < +4.75 V Check encoder connections. ■ Vcc (+5 V) > +5.50 V Exchange power supply unit. ■ Vcc (+15 V) < +14.25 V Exchange power supply unit. ■ Vcc (+15 V) > +16.50 V Exchange power supply unit. ■ Vcc (–15 V) < –14.25 V Exchange power supply unit. ■ Vcc (–15 V) > –16.50 V Exchange power supply unit.	340 420-04

Error message	Cause	Corrective action	As of NC software
B900 CC<Board> supply voltage <code>	■ The Vcc supply voltage (x) was out of range. ■ +4 = Undervoltage Vcc (+5 V) Excessive load from external components (e.g. encoders). ■ +6 = Undervoltage Vcc (+5 V) The power supply unit is defective. ■ +14 = Undervoltage Vcc (+15 V) The power supply unit is defective. ■ +16 = Undervoltage Vcc (+15 V) The power supply unit is defective. ■ –14 = Undervoltage Vcc (–15 V) The power supply unit is defective. ■ –16 = Undervoltage Vcc (–15 V) The power supply unit is defective.	■ Inform your service agency. • Measure supply voltage Vcc (x). • Vcc (+5 V) < +4.75 V Check encoder connections. • Vcc (+5 V) > +5.50 V Exchange power supply unit. • Vcc (+15 V) < +14.25 V Exchange power supply unit. • Vcc (+15 V) > +16.50 V Exchange power supply unit. • Vcc (–15 V) < –14.25 V Exchange power supply unit. • Vcc (–15 V) > –16.50 V Exchange power supply unit.	340 420-07 340 422-01
BA00 Error in operating temperature	■ The permissible operating temperature was exceeded. ■ Temperature sensor on PCB is defective. ■ Insufficient ventilation of the electrical cabinet (fan defective). ■ Ambient temperature is too high.	■ Check the ventilation conditions. ■ Inform your service agency.	340 420-01
BA00 Operating temperature <Board>	■ Permissible operating temperature exceeded. ■ Temperature sensor on motor control board defective. ■ Insufficient ventilation of the electrical cabinet (fan defective). ■ Ambient temperature too high.	■ Check the ventilation conditions. ■ Inform your service agency. • Exchange the motor drive control board.	340 420-04

Error message	Cause	Corrective action	As of NC software
BA00 CC<Board> operating temperature <temperature>	■ The temperature inside the LE was out of the permissible range. (–128...0...+127 = Measured temperature value [°C]) ■ Temperature sensor on PCB is defective. ■ Insufficient ventilation of the electrical cabinet (fan defective). ■ The ambient temperature is too high or too low.	■ Check the ventilation conditions. ■ Inform your service agency.	340 420-07 340 422-01
C000 No data exchange with MCU	■ Communication with the MCU was interrupted. ■ An internal software error has occurred.	■ Inform your service agency.	340 420-01
C000 No data exchange with MC	■ Communication with the MC was interrupted. ■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-04 340 422-01
C001 Undefined error	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
C002 MCU command invalid	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01
C002 MC command invalid	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-04 340 422-01
C003 MCU/CCU system clock mismatch	■ Hardware error (quartz generator). ■ Software error.	■ Inform your service agency. • Exchange the drive control board or processor board. • Check software version.	340 420-01
C003 MC/CC system clock mismatch	■ Hardware error (quartz generator). ■ Software error.	■ Inform your service agency. • Exchange drive control board or CPU board. • Check software version.	340 420-04 340 422-01
C004 Undefined interrupt	■ Software error. ■ Hardware error: Disturbance results in internal interrupt.	■ Switch off the machine. ■ Switch on the machine. ■ Inform your service agency. • Check software version. • Check the grounding.	340 420-01 340 422-01
C005 Unknown hardware identifier	■ Software does not fit the hardware. ■ Hardware defective.	■ Inform your service agency. • Check software version. • Exchange drive control board.	340 420-01 340 422-01
C006 I-CTRL communication: TIME	■ Communication error between speed and current controllers.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
C007 DC-link voltage too low	■ Line power interrupted. ■ Inverter defective.	■ Check your line power supply. ■ Inform your service agency. • Check the inverter.	340 420-01 340 422-01
C008 I-CTRL communication: QUEUE	■ Communication error between speed and current controllers.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
C009 Stack overflow	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
C00A PWM triangular signal error	■ Hardware error: Triangular signal does not oscillate or it oscillates at the wrong frequency.	■ Inform your service agency. • Exchange drive control board.	340 420-01 340 422-01
C00B Too little main memory	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
C00C LSV2, incorrect number of data	■ The number of LSV2 data to be read is incorrect ■ A software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
C00D Program checksum error	■ An internal software or hardware error has occurred.	■ Inform your service agency. • Check software version. • Exchange drive control board.	340 420-01 340 422-01
C00E Controller software timeout	■ An internal software or hardware error has occurred.	■ Inform your service agency. • Check software version. • Exchange drive control board.	340 420-01 340 422-01
C00F Error in software timer	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
C010 Bus error in speed controller	■ Access violation on controller periphery.	■ Inform your service agency. • Exchange control board.	340 420-01 340 422-01
C011 Softw. synchronization err.	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
C012 Pos. control err. Cycle time	■ MC is outputting erroneous cycle time for CC position controller. ■ A hardware error has occurred.	■ Inform your service agency. • Check machine parameter MP7600.x. • Exchange drive control board.	340 420-01 340 422-01
C013 PWM frequency error	■ Entered PWM frequency in MP2180 lies outside the permissible input range.	■ Inform your service agency. • Check MP2180.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
C014 Interpolator, PWM invalid	■ Invalid relation between interpolator clock and PWM frequency.	■ Inform your service agency. • Change the relation between interpolator clock and PWM frequency. • See the Technical Manual for possible relations.	340 420-01 340 422-01
C015 Interpolator, PWM changed	■ Interpolator clock or PWM frequency was changed.	■ Restart the control.	340 420-01 340 422-01
C110 Unknown motor type <axis>	■ Entry in MP2200.x or in the motor table incorrect. ■ An internal software error has occurred.	■ Inform your service agency. • Check MP2200.x and motor table • Check software version.	340 420-01 340 422-01
C140 Pole pair no. too large <axis>	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	340 420-01 340 422-01
C150 Field current error <axis>	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	340 420-01 340 422-01
C160 Grating per. of motor encoder <axis>	■ Measured grating period does not agree with entry in the motor table.	■ Inform your service agency. • Check motor table (line count). • Check the motor.	340 420-01 340 422-01
C170 Rotor time constant err. <axis>	■ The rotor time constant calculated from the rotor table is invalid.	■ Inform your service agency. • Check the motor table.	340 420-01 340 422-01
C180 Rated speed error <axis>	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	340 420-01 340 422-01
C1D0 Current sensor voltage <axis>	■ Incorrect entry in power module table.	■ Inform your service agency. • Check the power module table.	340 420-01 340 422-01
C1E0 Imax of power module <axis>	■ Incorrect entry in power module table.	■ Inform your service agency. • Check the power module table.	340 420-01 340 422-01
C210 Tmax of motor table <axis>	■ Incorrect temperature entry in motor table.	■ Inform your service agency. • Check the motor table.	340 420-01 340 422-01
C240 Irated of power module <axis>	■ Incorrect entry in power module table.	■ Inform your service agency. • Check the power module table.	340 420-01 340 422-01
C250 Irated of motor <axis> incorrect	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	340 420-01 340 422-01
C260 Imax of motor <axis> incorrect	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	340 420-01 340 422-01
C270 Nmax of motor <axis> incorrect	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	340 420-01 340 422-01
C280 Field angle <axis> incorrect	■ Incorrect entry in MP2340.x or MP2350.x.	■ Inform your service agency. • Check entry in MP2340.x or MP2350.x.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
C290 Uz <axis> incorrect	■ Incorrect entry in MP2190 (dc-link voltage Uz).	■ Inform your service agency. • Check the entry in MP2190.	340 420-01 340 422-01
C2A0 Encoder input <axis>	■ Incorrect entry in MP112.x or MP113.x (speed encoder). ■ An internal software error has occurred.	■ Inform your service agency. • Check the entry in MP112.x or MP113.x. • Check software version.	340 420-01 340 422-01
C2B0 PWM output <axis>	■ Incorrect entry in MP120.x or MP121.x (nominal speed output). ■ An internal software error has occurred.	■ Inform your service agency. • Check the entry in MP120.x or MP121.x. • Check software version.	340 420-01 340 422-01
C2C0 Band filter parameter <axis>	■ Incorrect entry in MP2540.x, MP2541.x, MP2550.x or MP2551.x. ■ An internal software error has occurred.	■ Inform your service agency. • Check the entry in MP2540.x, MP2541.x, MP2550.x or MP2551.x. • Check software version.	340 420-01 340 422-01
C2D0 Encoder line count <axis>	■ Encoder line count was changed.	■ Restart the control.	340 420-01 340 422-01
C2E0 Motor pole-pair number <axis>	■ Number of motor pole pairs was changed.	■ Restart the control.	340 420-01 340 422-01
C2F0 DIR in motor table <axis>	■ DIR in motor table was changed.	■ Restart the control.	340 420-01 340 422-01
C300 Zn track <axis> error	■ Contamination of the motor encoder (Zn track). ■ Motor encoder cable is defective. ■ Motor control board defective.	■ Inform your service agency. • Exchange the motor. • Check the motor encoder cable. • Exchange the motor drive control board.	340 420-01 340 422-01
C310 Z1 track <axis> error	■ Contamination of the motor encoder (Z1 track). ■ Motor encoder cable is defective. ■ Motor control board defective.	■ Inform your service agency. • Exchange the motor. • Check the motor encoder cable. • Exchange the motor drive control board.	340 420-01 340 422-01
C330 Motor temp. too high <axis>	■ Measured motor temperature is too high. ■ No temperature sensor. ■ Motor encoder cable is defective. ■ Entry in motor table is incorrect. ■ Incorrect or defective temperature sensor was installed.	■ Let the motor cool down. ■ Inform your service agency. • Check the motor encoder cable. • Check the entry in the motor table. • Measure the temperature sensor (2000 [Ohm] at 25 [°C]).	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
C340 Unknown counter range <axis>	■ Hardware defective. ■ Incorrect software version.	■ Inform your service agency. • Check software version. • Exchange drive control board.	340 420-01 340 422-01
C350 Axis module <axis> not ready	■ No pulse release for the axis module. ■ Uz too large. ■ 5-V power supply too weak. ■ Inverter is not ready for operation. ■ Motor control board defective. ■ PWM cable defective. ■ Noise pulses.	■ Inform your service agency. • Check the signal and cabling of the pulse release. • Check Uz. • If the power supply is not regenerative: Is the braking resistor connected? • If the power supply is regenerative: Is energy recovery activated? • Check the grounding and shielding of the cable. • Exchange the power module. • For P controls: Exchange the interface card. • Exchange the motor drive control board.	340 420-01 340 422-01
C370 Angular deviation of motor encoder <axis>	■ Motor encoder defective. ■ Motor encoder cable is defective. ■ Drive control board defective.	■ Inform your service agency. • Check motor encoder and leads. • Exchange drive control board.	340 420-01 340 422-01
C380 Motor <axis> not controllable	■ Motor cable switched (e.g., X with Y). ■ Motor encoder cable switched. ■ Phases incorrectly connected to motor. ■ Motor encoder cable is defective. ■ Incorrect motor table entry (direction of rotation). ■ Motor defective. ■ I2t monitoring is responding.	■ Check motor cabling. ■ Inform your service agency. • Check motor and motor encoder cable. • Check motor table entry. • Check I2t monitoring (MP2302.x).	340 420-01 340 422-01
C390 Error 3-D touch probe system <axis>	■ Software error. ■ Hardware error: control board.	■ Inform your service agency. • Exchange the motor drive control board. • Check software version.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
C3A0 Incorrect Ref position <axis>	■ Incorrect motor selected (MP2200). ■ Ground error on the motor encoder cable (noise on Ref). ■ Motor encoder defective.	■ Inform your service agency. • Check motor selection (MP2200). • Check motor encoder cabling (ground). • Exchange the motor.	340 420-01 340 422-01
C3B0 Motor <axis> is not turning	■ Inverter is not ready. ■ Noise on the RDY input of the PWM output connector. ■ Motor jammed. ■ Inverter defective. ■ Motor defective. ■ Incorrect motor selected (MP2200.x). ■ Assignment of PWM outputs incorrectly entered in MP120.x. ■ Assignment of encoder inputs incorrectly entered in MP112.x. ■ Motor power cable switched. ■ Motor encoder cable has been switched. ■ Incorrect motor connection. ■ I2t monitoring has responded.	■ Inform your service agency. • Check the inverter. • Check motor and cabling. • Check machine parameters. • Check I2t monitoring (MP2302.x).	340 420-01 340 422-01
C3C0 Motor current <axis> too high	■ Incorrect current controller parameters. ■ Incorrect parameters in the motor table. ■ Power module defective. ■ Motor defective. ■ Motor control board defective.	■ Inform your service agency. • Is the correct motor and power module selected? • Check the current control adjustment. • Check the motor and motor cable for a short circuit. • Exchange power module or drive control board	340 420-01 340 422-01
C3D0 PWM component defect <axis>	■ An internal hardware error has occurred.	■ Inform your service agency. • Exchange drive control board.	340 420-01 340 422-01
C3E0 Incorrect rated U of motor <axis>	■ Rated motor voltage outside of the permitted input range.	■ Inform your service agency. • Check the entry in the motor table.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
C3F0 EnDat not found <axis>	■ EnDat communication is defective.	■ Inform your service agency. • Check the motor table (column SYS). • Exchange the motor drive control board. • Check speed encoder cables (defective or too long). • Check speed encoder. • Check the grounding and shielding of the cable.	340 420-01 340 422-01
C400 Encoder line count incorrect <axis>	■ The values for the encoder line count from the motor table do not match the downloaded values.	■ Inform your service agency. • Check the motor type in the machine parameters. • Check the motor table. • Check the mounted encoder.	340 420-01
C400 Line count incorrect <axis>	■ Line count from the motor table does not match the downloaded values.	■ Inform your service agency. • Check machine parameters for linear distance of one motor revolution and distance for the number of signal periods. • Check the motor table (columns TYPE and STR). • Check speed encoder.	340 420-02 340 422-01
C410 Rotor position <axis> undefined	■ Contamination of the speed encoder (Zn track). ■ Speed encoder cable defective. ■ Motor control board defective. ■ Speed encoder defective.	■ Inform your service agency. • Exchange the motor. • Check speed encoder cable. • Exchange the motor drive control board.	340 420-01 340 422-01
C420 V control <axis> is not possible	■ Incorrect speed controller parameters (kV, kI) ■ Incorrect speed controller filter parameters ■ Excessive acceleration.	■ Inform your service agency. • Check machine parameters for speed controller. • Check machine parameters for filter. • Check software version.	340 420-01 340 422-01
C430 No position module <axis>	■ Position module with position encoder does not exist. ■ Position module with position encoder is not connected correctly. ■ Position module with position encoder is defective.	■ Inform your service agency. • Install position module with position encoder. • Check the connection of the position module with position encoder. • Exchange the position module with position encoder.	340 420-01

Error message	Cause	Corrective action	As of NC software
C430 Error of position input <axis>	■ Position encoder input does not exist. ■ Position encoder input not connected correctly. ■ Position encoder input defective.	■ Inform your service agency. • Install position encoder input. • Check connection of the position encoder input. • Exchange position encoder input.	340 420-02 340 422-01
C440 PWM frequency <axis> incorrect	■ PWM frequency within a control group is incorrect.	■ Inform your service agency. • Check machine parameters for PWM frequency. • PWM frequency > 5000 Hz only with suitable hardware and only with PWM outputs X51, X52, X57 and X58. • PWM frequency <= 5000 Hz must be identical within the control group. • PWM frequency > 3200 Hz.	340 420-01 340 422-01
C450 Incorrect encoder <axis>	■ Incorrect encoder selected in the motor table, e.g. linear encoder instead of rotary encoder, EnDat encoder instead of encoder with Z1 track.	■ Inform your service agency. • Correct the encoder entry in the motor table. • Check the motor encoder cable. • Exchange the motor. • Exchange the motor drive control board.	340 420-01
C450 Incorrect encoder <axis>	■ Entry in column SYS of the motor table incorrect. ■ Speed encoder cable defective. ■ Speed encoder defective. ■ Motor control board defective.	■ Inform your service agency. • Check the motor table (column SYS). • Check speed encoder cable. • Exchange the motor. • Exchange the motor drive control board.	340 420-02 340 422-01
C460 Motor speed too high <axis>	■ Motor not controllable	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
C470 No nominal speed values <axis>	■ An internal software error has occurred. ■ Position controller cycle time too short.	■ Inform your service agency. • Check software version. • Check machine parameter MP7600.0.	340 422-01
D000 DP RAM area overlap	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01
D000 DP RAM area <area>	■ An internal software error has occurred. 1...255 = Area number	■ Inform your service agency. • Check software version.	340 420-04

Error message	Cause	Corrective action	As of NC software
D000 CC<Board> DP RAM area <area>	■ An internal software error has occurred. 1...255 = Area number	■ Inform your service agency. • Check software version.	340 420-07 340 422-01
D100 Software error	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	340 420-01
D100 Software error <code>	■ An internal software error has occurred. 0...255 = Code for faulty software module or routine	■ Inform your service agency. • Check software version.	340 420-04
D100 CC<Board> software error <code>	■ An internal software error has occurred. 0...255 = Code for faulty software module or routine	■ Inform your service agency. • Check software version.	340 420-07 340 422-01
E001 Status NR1/NR2 not equal	■ NR2 input incorrectly connected. ■ Software error.	■ Inform your service agency. • Check wiring. • Check software version.	340 420-01 340 422-01
E002 Status NE1/NE2 not equal	■ NE2 input incorrectly connected. ■ Software error.	■ Inform your service agency. • Check wiring. • Check software version.	340 420-01 340 422-01
E003 PLC module 9169 illegal	■ PLC Module 9169 in safety-oriented software (illegal). ■ Software error.	■ Inform your service agency. • Check the PLC program. • Check software version.	340 420-01 340 422-01
E008 SRG speed too high	■ Safe reduced rotational velocity (SRG) was exceeded. ■ No standstill in safe controlled stop (SBH) operating mode.	■ Inform your service agency.	340 420-01 340 422-01
E009 Incorrect gear range	■ Software error.	■ Inform your service agency. • Check software version.	340 420-01 340 422-01
E00A Safe machine parameter error	■ CRC checksum does not fit the entered safe MPs.	■ Inform your service agency. • Check the safe machine parameters.	340 420-01 340 422-01
E00B Cutout channels test error	■ Machine key depressed (ZT.HR, ZT.MB, MT signal).	■ Inform your service agency. • Check the wiring X65, X66 (and X67). • Check machine keys.	340 420-01 340 422-01
E00C Error in MP3210 transfer	■ MP3210.x incorrect. ■ Check the software version of the MC.	■ Inform your service agency. • Check MP3210.x. • Check software version.	340 420-01 340 422-01
E00D Error in MP3510 transfer	■ MP3510.x incorrect. ■ MC software error.	■ Inform your service agency. • Check MP3510.x. • Check software version.	340 420-01 340 422-01
E00E Error in MP2020 transfer	■ MP2020.x incorrect. ■ MC software error.	■ Inform your service agency. • Check MP2020.x. • Check software version.	340 420-01 340 422-01

Error message	Cause	Corrective action	As of NC software
E010 Error in CCU watchdog test	■ CCU watchdog signal does not switch to low level	■ Inform your service agency.	340 420-01
E010 Error in CC watchdog test	■ CC watchdog signal does not switch to low level	■ Inform your service agency. • Check software version. • Exchange the motor drive control board.	340 420-04 340 422-01
E011 Error in CCU watchdog test	■ CCU watchdog signal does not switch to high level	■ Inform your service agency.	340 420-01
E011 Error in CC watchdog test	■ CC watchdog signal does not switch to high level	■ Inform your service agency. • Check software version. • Exchange the motor drive control board.	340 420-04 340 422-01
E012 Error in CC switch-off test	■ CC switch-off signal –NO does not switch to low level.	■ Inform your service agency. • Hardware defective. • Check drive control board.	340 420-04 340 422-01
E013 Error in CC switch-off test	■ CC switch-off signal –NO does not switch to high level.	■ Inform your service agency. • Hardware defective. • Check drive control board.	340 420-04 340 422-01
E014 Error in CC switch-off test	■ CC switch-off signal –NO does not switch on both CCs.	■ Inform your service agency. • Hardware defective. • Check connection of both controller boards.	340 420-04 340 422-01
E120 Safe function call error	■ Software error.	■ Inform your service agency. • Check software version.	340 420-01
E120 CC<Board> Safe function call error	■ Software error.	■ Inform your service agency. • Check software version.	340 420-07 340 422-01
E140 Current to axis <axis> not equal 0	■ Motor current was determined during cutout channel test (24-hour test).	■ Inform your service agency. • Check the inverter.	340 420-01 340 422-01
E150 Inverter <axes> ready	■ RDY status of the inverter is HIGH instead of LOW.	■ Inform your service agency. • Check the inverter. • Check the cabling of the cutout channels.	340 420-01 340 422-01
E160 Inverter <axis> not ready	■ RDY status of the inverter is LOW instead of HIGH.	■ Inform your service agency. • Check the inverter. • Check the cabling of the cutout channels.	340 420-01 340 422-01
E170 Position deviation too large <axis>	■ MP640 too small. ■ Incorrect mounting of position encoder. ■ Incorrect temperature compensation, linear or nonlinear compensation, or reversal error.	■ Inform your service agency. • Correct MP640. • Check the encoder mounting. • Check the compensation.	340 420-01 340 422-01

10.2 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. input/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.

10.3 Error Messages of the File System

The following error messages can be displayed on the iTNC:

Error message	Error correction
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	

11 Index

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

X