HEIDENHAIN

Technical Manual

TNC 426 CB/PB/M
TNC 430 CA/PA/M

NC Software	280 470-12
	280 471-12
	280 472-14
	280 473-14
	280 474-19
	280 475-19
	280 476-21
	280 477-21

December 2001

343 734-21 · 10 · 12/2001 · Bi · Printed in Germany · Subject to change without notice
(343 734-E6)

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 is perfect. Documentation is alive. It 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 No. 6

1.1 Releases

The following NC software was released:

NC software 280 474-18 and 280 475-18	June 2001
NC software 280 474-19 and 280 475-19	August 2001
NC software 280 476-15 and 280 477-15	November 2000
NC software 280 476-16 and 280 477-16	December 2000
NC software 280 476-17 and 280 477-17	March 2001
NC software 280 476-18 and 280 477-18	April 2001
NC software 280 476-19 and 280 477-19	July 2001
NC software 280 476-20 and 280 477-20	August 2001
NC software 280 476-21 and 280 476-21	November 2001

1.2 NC software 280 474-xx

NC software	Setup Disks	Release
280 474-18	286 195-21	06/2001
Export version:		
280 475-18	286 195-21	06/2001

NC Software	Setup Disks	Release
280 474-19	286 195-22	08/2001
Export version:		
280 475-19	286 195-22	08/2001

1.3 NC software 280 476-xx

NC software	Setup Disks	Release
280 476-15	286 197-17	11/2000
Export version:		
280 477-15	286 197-17	11/2000

Machine parameters

- MP7441 has been expanded:
  Bit 1: Reserved, enter 0
  Bit 2: Error message **"Enter depth as negative"** when a positive depth was programmed in the fixed cycles.
  = 0: Error message is suppressed
  = 1: Error message is not suppressed
- MP7682 has been expanded:
  Bit 3: Reserved, enter 0
  Bit 4: Reserved, enter 0

Machine integration

- Before a probing block, the control checks whether M4501 is set. If it is, the probing block start is delayed by 1 second.

NC software	Setup Disks	Release
280 476-16	286 197-18	12/2000
Export version:		
280 477-16	286 197-18	12/2000

Machine parameters

- MP2221 has been expanded:
  All HEIDENHAIN inverters except the UE 2xx compact inverter provide the error signal $\overline{\text{ERR-IZ}}$.
  Bit 2 = 0: Do not monitor the $\overline{\text{ERR-IZ}}$ signal, or the inverter doesn't supply this signal (SIEMENS and INDRAMAT inverters, and HEIDENHAIN UE 2xx compact inverters).
  Bit 2 = 1: Monitor the $\overline{\text{ERR-IZ}}$ signal (HEIDENHAIN inverters except UE 2xx compact inverters).
- The minimum input value of MP7430 was changed from 0.1 to 0.001.

PLC programming

- FN18: SYSREAD has been expanded:
  You can ascertain the angle of misalignment between the spindle and the tilted coordinate system.
  ID210
  NR8
- FN18: SYSREAD has been expanded:
  You can find the measurement results of touch probe cycles 0 and 1 without probe radius and length compensation.
  ID360
  NR3

NC Software	Setup Disks	Release
280 476-17	286 197-19	03/2001
Export version:		
280 477-17	286 197-19	03/2001

Machine parameters

- The PLC can change machine parameter MP2397.x.
- MP7683 has been expanded:
  Bit 7: AUTOSTART function of an NC program through the PLC
  0: AUTOSTART function of an NC program is performed by the NC.
  1: AUTOSTART function of an NC program is performed by the PLC.
  The NC does not trigger an NC start.

Machine integration

- M4182 is new:
  The marker indicates whether the autostart function was activated.
- M4183 is new:
  The marker indicates whether the time from the autostart function has expired.
- If an NC program block is interrupted in **`Single block mode`** or by a STOP block and the positions of NC axes are changed, the NC program can be restarted at the changed positions. If **`STRICTREPOS = YES`** in OEM.SYS, the function for restoring the position is activated.

Miscellaneous

- 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 NC software or setup files from the hard disk.
- When the control starts, the data of the fixed cycles and touch probe cycles are checked and the resulting information is saved in the log.

NC software	Setup Disks	Release
280 476-17	286 197-20	03/2001
Export version:		
280 477-17	286 197-20	03/2001

NC Software	Setup Disks	Release
280 476-18	286 197-21	04/2001
Export version:		
280 477-18	286 197-21	04/2001

Machine parameters

- The maximum input values of MP1060.x and MP1070.x were extended to 30 [m/s^2 or 1000°/s^2].
- MP7682 has been expanded:
  Bit 4: Tolerance of rotary axes with M128
  = 0: With consideration of head dimensions
  = 1: Without consideration of head dimensions

Machine integration

- Since no speed encoder is used during volts-per-hertz (U/f) control mode, W322 = 0. Module 9164 supplies the actual speed value while the spindle is active, and not during the acceleration and braking phase.

NC Software	Setup Disks	Release
280 476-18	286 197-22	05/2001
Export version:		
280 477-18	286 197-22	05/2001

NC Software	Setup Disks	Release
280 476-18	286 197-23	05/2001
Export version:		
280 477-18	286 197-23	05/2001

NC Software	Setup Disks	Release
280 476-19	286 197-24	07/2001
Export version:		
280 477-19	286 197-24	07/2001

Machine parameters

- MP2180 has been expanded, MP2181 has been added:
  MP2180 has been expanded into MP2180.0 to MP2180.8.
  In MP2180.0 to MP2180.8 and in MP2181, the same value must be entered.
- The maximum input value of MP2600.x was increased to 30 000 [A/(rev/s^2)].
- MP7160 has been expanded:
  Bit 3 = 0: IPC and acceleration feedforward control active
  Bit 3 = 1: IPC and acceleration feedforward control not active

Machine integration

- 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.
- A maximum of 16 variables can be used in the **`MP7530`** column and in the **`TEMPCOMP`** column of the description tables for the swivel axis geometry.

PLC modules

Module 9120 Starting a PLC axis

Module 9120 was expanded by an error code:
6= Feed rate not permitted

Module 9123 Traversing the reference marks of PLC axes

Module 9123 was expanded by an error code:
6= Feed rate not permitted

Miscellaneous

- If **`REMOTE.PLCPASSWORDFORCED = YES`** in OEM.SYS, machine backup, full backup and setup are only possible with the code word defined in **`PLCPASSWORD =`**.

NC Software	Setup Disks	Release
280 476-20	286 197-25	08/2001
Export version:		
280 477-20	286 197-25	08/2001

NC Software	Setup Disks	Release
280 476-20	286 197-26	09/2001
Export version:		
280 477-20	286 197-26	09/2001

NC software	Setup disks	Release
280 476-21	286 197-27	11/2001
Export version:		
280 477-21	286 197-27	11/2001

Machine parameters

- MP1152 is new:
  0: I3 (control-is-ready signal acknowledgement) is passed on directly to the NC
  1: I3 is processed by the PLC before being passed on to the NC

Machine integration

- The machining plane position indication (MP7500 bit 1 = 1) has been expanded by the following swivel-axis combination:
  Swivel head and rotary table: axis sequence B variable, A variable (tool axis Z)
- An internal EMERGENCY STOP can be simulated with the code number FAILTEST in order to check the 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.

- The power module table was expanded by the following columns: I-N-DC, T-DC, F-DC, T-AC, F-AC, T-IGBT, I-N-AC-3333, I-N-AC-4000, I-N-AC-5000, I-N-AC-6666, I-N-AC-8000 and I-N-AC-10000. The columns currently have no function.
- The motor table was expanded by the following columns T-DC, F-DC, T-AC and F-AC. The columns currently have no function.

Miscellaneous

- In the **`Machine-parameter programming`** mode, the DELETE ALL and SELECT soft keys after the DEL/SEL SETUP soft key were switched.

1.4 Use of Speed Encoders with EnDat Interface

Regardless of the **`Type of encoder`** in the motor table, the control attempted to communicate with a speed encoder with EnDat interface. If this did not succeed, a speed encoder with Z1 track was assumed.
If an error occurred during communication with the EnDat encoder, the control assumed that it was dealing with an encoder with a Z1 track. This was not the case, however, since encoders with EnDat interface do not have a Z1 track. This resulted in the error message **`C310 Z1 track error.`**
As of NC software 280 476-18 (in conjunction with the setup 286 197-22), the control uses the **`Type of encoder`** entry in the 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

If you use the motor table motor.sn instead of motor.mot, the control attempts to communicate with an encoder with EnDat interface. If this fails, due to an error or because no EnDat encoder is connected, the control assumes that it is dealing with an encoder with Z1 track and tries to read it. If this fails, the error message **`C310 Z1 track error`** appears.

1.5 Tool-Oriented Machining

As of NC software 280 476-17, "tool-oriented" pallet table machining is also possible. For more information, please refer to the User's Manual.

A special tool-change macro is required for tool-oriented pallet machining. This is defined through the keyword **TCTOOLMODE=** in **NCMACRO.SYS.**

This macro is called for tool oriented machining instead of the tool-change macro. If this macro is not defined in **NCMACRO.SYS,** a HEIDENHAIN standard macro is run.

The HEIDENHAIN standard macro performs the following functions:

- Positioning to clearance height
- Execution of M146
- Tool change through **TOOL CALL.** The existing tool-change macro is called.

The following new functions are available for interrogating whether a clearance height was programmed in the pallet table:

- FN18: SYSREAD ID510 NR5 IDX(axis)
  This function can ascertain whether a clearance height was programmed for the corresponding axis.
- FN18: SYSREAD ID510 NR6 IDX(axis)
  This function can ascertain the clearance height for the corresponding axis.

With the M function M146 the current geometry information is saved in a temporary file.

An NC macro can be defined through the keyword **CLAMP=** in **NCMACRO.SYS.** The macro is called when a loaded fixture (**FIX**) is called.

In addition to the standard prototype for pallet tables, the COPY SAMPLE FILES soft key copies the prototype for tool-oriented pallet table machining into the **PLC:\PROTO** directory. Both prototypes are offered when you create a new pallet table. If you do not want this, delete a prototype from the **PLC:\PROTO** directory. The existing prototype is then used automatically.

- **Prototyp.P** = standard prototype
- **Proto_to.P** = prototype for tool-oriented machining

1.6 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. This is remedied by NC software 280 476-13 with the new FIELD ORIENTATION function on the LE 426 M/30 000 rpm and the LE 430 M. When put into service, the control automatically finds the assignment between the encoder and the rotor magnets and saves this information on the hard disk. From this time on the assignment is available to the servo controller.

1.7 New Motor Table

As of NC software 280 476-13, the motor table motor.mot is used as the standard table instead of the previous standard motor tables motor.asn and motor.sn. Synchronous and asynchronous motors are registered in motor.mot. If the new motor table motor.mot is not available, the control looks for motor.asn and motor.sn.
You can transfer asynchronous motors from motor.asn into motor.mot. To transfer synchronous motors into motor.mot, please contact HEIDENHAIN.

1.8 Hardware

New receiver units

The EA 550 and EA 552 receiver units and the APE 511 interface electronics for the connection of two EA 552 to the LE have been superseded by new units.

Old units	New units
EA 550 (Id. Nr. 262 904-xx)	EA 632 (Id. Nr. 346 322-xx)
EA 552 (Id. Nr. 339 317-xx)	EA 652 (Id. Nr. 346 323-xx)
APE 511 (Id. Nr. 275 759-xx)	APE 652 (Id. Nr. 354 656-01)

LE 430 M/9 axes with flash EPROMs

On the LE 430/9 axes with flash EPROMs, only the speed encoder inputs and the position encoder inputs X1 to X6 (but not X35 to X38) are equipped with EnDat interface.

1.9 Replacing Instructions

Page	Change	Remove Page	Insert Page
Title	New software	December 2000	December 2001
Chapter 1	Update information	–	Update Info. 6
Chapter 2	Errors corrected, some descriptions changed and updated	Entire chapter	Entire chapter
Chapter 3	Errors corrected, some descriptions changed and updated	Entire chapter	Entire chapter
Chapter 4	Machine parameter list updated	Entire chapter	Entire chapter
Chapter 5	Module, marker and word lists updated	Entire chapter	Entire chapter
Chapter 6	Errors corrected, some descriptions changed, updated and expanded	Entire chapter	Entire chapter
Chapter 7	Errors corrected, some descriptions changed and updated	Entire chapter	Entire chapter
Chapter 8	Errors corrected, some descriptions changed and updated, new descriptions added	Entire chapter	Entire chapter
Chapter 9	No changes	Entire chapter	Entire chapter
Chapter 10	Errors corrected	Entire chapter	Entire chapter
Chapter 11	Index updated	Entire chapter	Entire chapter

2 Introduction

2 Introduction

2.1 General Information

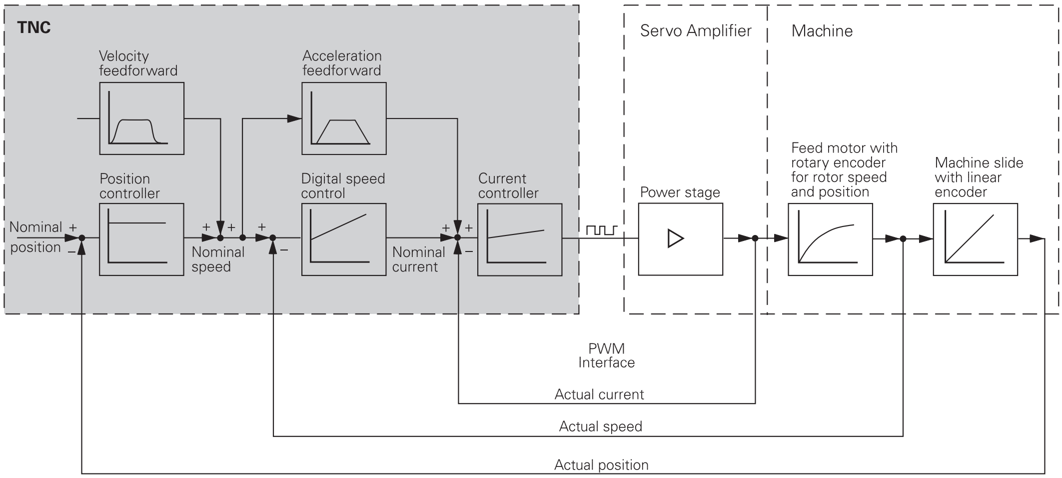
HEIDENHAIN contouring controls are designed for use with milling, drilling and boring machines as well as machining centers.

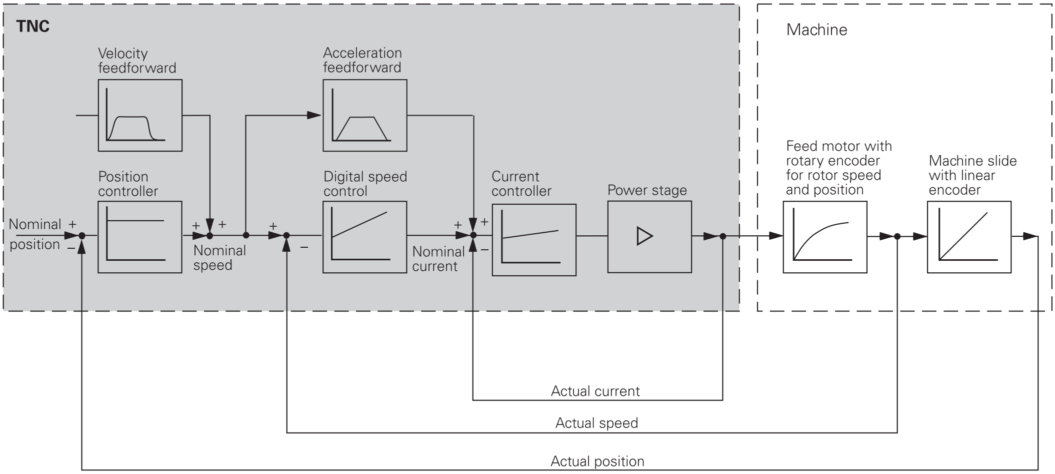
The **TNC 426 PB/M, TNC 430 PA/M** features integral digital drive control and controls the power stages via PWM signals.

Integration of the drive controller in the TNC 426 PB/M, TNC 430 PA/M 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.

With the **TNC 426 PB,** up to five axes and spindle speeds of up to 12 000 rpm can be controlled digitally (option: 30 000 rpm).
The **TNC 430 PA** supports up to 6 digitally controlled NC axes, 3 analog controlled secondary axes, and digitally controlled spindle speeds up to 30 000 rpm.

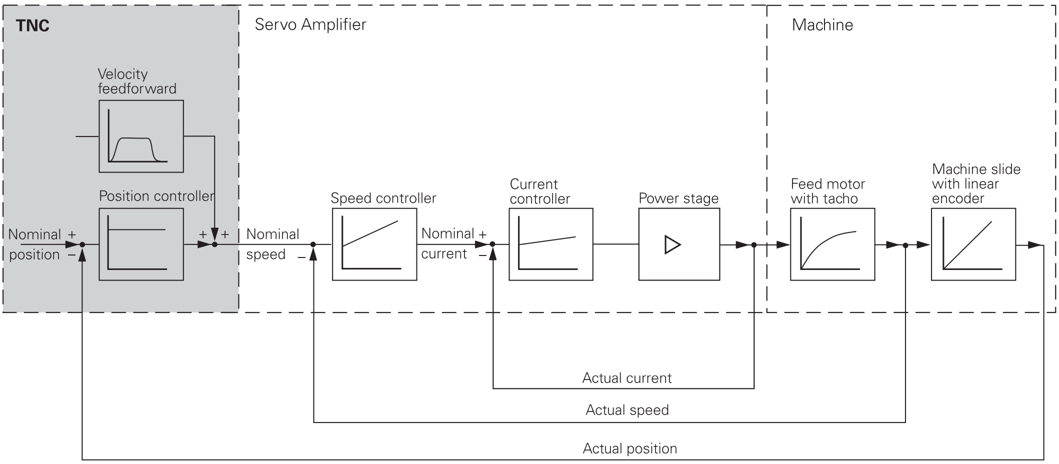




The **TNC 426 M** offers digital control for up to 5 axes and spindle speeds up to 12 000 rpm (option: 30 000 rpm). The **TNC 430 M** offers digital control for up to six or nine axes and spindle speeds up to 30 000 rpm.

The **TNC 426 M, TNC 430 M** is designed for connection of a compact or modular inverter system. Thus, together with HEIDENHAIN motors, a complete control package including servo drive can be offered (see Technical Manual “Inverter Systems and Motors”).

The **TNC 426 CB** is the version of the TNC 426 that is equipped with analog speed command interface and can control machines with up to five axes plus spindle. The **TNC 430 CA** also has an analog speed command interface for machines with up to eight axes plus spindle. A ninth axis can be controlled with an additional PCB.



2.2 Overview of Components

LE 426 CB, LE 430 CA Logic Unit	

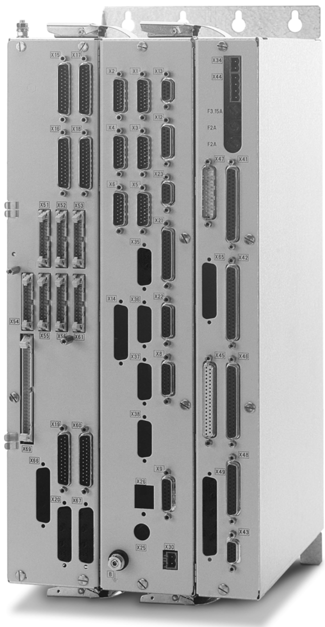
Logic unit	Signal inputs	ID numbers of LE for BC 120 display unit	ID numbers of LE for BF 120 display unit
LE 426 CB			
5 position inputs 1 spindle position input	1 V_{PP} (350 kHz) 1 V_{PP} (350 kHz)	312 001-xx	313 524-xx
5 position inputs 1 spindle position input	1 V_{PP} (50 kHz) 1 V_{PP} (350 kHz)	326 415-xx	326 419-xx
5 position inputs 1 spindle position input	11 μA_{PP} (50 kHz) 1 V_{PP} (350 kHz)	312 002-xx	313 525-xx
LE 430 CA			
8 position inputs 1 spindle position input	1 V_{PP} (350 kHz) 1 V_{PP} (350 kHz)	311 050-xx	313 523-xx
5 position inputs 3 position inputs 1 spindle position input	1 V_{PP} (50 kHz) 1 V_{PP} (350 kHz) 1 V_{PP} (350 kHz)	326 418-xx	326 424-xx

LE 426 PB, LE 430 PA Logic Unit

Logic unit	Signal inputs	ID numbers of LE for BC 120 display unit	ID numbers of LE for BF 120 display unit
LE 426 PB			
5 position inputs 1 spindle position input 6 speed inputs	1 V_{PP} (350 kHz) 1 V_{PP} (350 kHz) 1 V_{PP}		
Spindle up to 12 000 rpm		312 000-xx	313 527-xx
Spindle up to 30 000 rpm		315 475-xx	318 178-xx
5 position inputs 1 spindle position input 6 speed inputs	1 V_{PP} (50 kHz) 1 V_{PP} (350 kHz) 1 V_{PP}		
Spindle up to 12 000 rpm		326 414-xx	326 421-xx
Spindle up to 30 000 rpm		326 416-xx	326 420-xx
5 position inputs 1 spindle position input 6 speed inputs	11 μA_{PP} (50 kHz) 1 V_{PP} (350 kHz) 1 V_{PP}		
Spindle up to 12 000 rpm		311 999-xx	313 526-xx
Spindle up to 30 000 rpm		317 349-xx	318 177-xx

Logic unit	Signal inputs	ID numbers of LE for BC 120 display unit	ID numbers of LE for BF 120 display unit
LE 430 PA			
5 position inputs 1 spindle position input 7 speed inputs	1 V_{PP} (350 kHz) 1 V_{PP} (350 kHz) 1 V_{PP}	311 049-xx	313 521-xx
5 position inputs 1 spindle position input 7 speed inputs	1 V_{PP} (50 kHz) 1 V_{PP} (350 kHz) 1 V_{PP}	326 417-xx	325 716-xx

LE 426 M, LE 430 M Logic Unit



Logic unit	Signal inputs	ID numbers of LE for BC 120 display unit	ID numbers of LE for BF 120 display unit
LE 426 M with EPROMs			
6 position inputs 6 speed inputs	11 µA$_{PP}$/1 V$_{PP}$ (2 EnDats) 1 V$_{PP}$ (2 EnDats)		
Spindle up to 12 000 rpm		324 990-xx	324 991-xx
Spindle up to 30 000 rpm		324 994-xx	324 995-xx
LE 430 M with EPROMs			
6 position inputs 7 speed inputs	11 µA$_{PP}$/1 V$_{PP}$ (2 EnDats) 1 V$_{PP}$ (2 EnDats)	324 992-xx	324 993-xx
10 position inputs 10 speed inputs	11 µA$_{PP}$/1 V$_{PP}$ (6 EnDats) 1 V$_{PP}$ (10 EnDats)	324 996-xx	324 997-xx

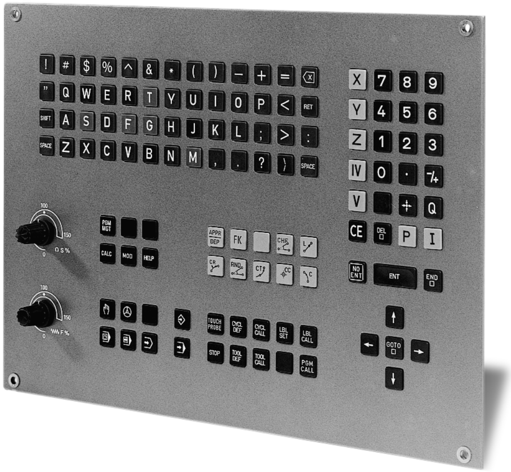
Note

A BC 120 can also be attached to the LE for the BF 120 at the same time.

Logic unit	Signal inputs	ID numbers of LE for BC 120 display unit	ID numbers of LE for BF 120 display unit
LE 426 M with flash EPROMs			
6 position inputs 6 speed inputs	11 µA$_{PP}$ /1 V$_{PP}$ (6 EnDats) 1 V$_{PP}$/EnDat		
Spindle up to 12 000 rpm		344 958-xx	344 959-xx
Spindle up to 30 000 rpm		344 962-xx	344 963-xx
LE 430 M with flash EPROMs			
6 position inputs 7 speed inputs	11 µA$_{PP}$ /1 V$_{PP}$ (6 EnDats) 1 V$_{PP}$/EnDat	344 960-xx	344 961-xx
10 position inputs 10 speed inputs	11 µA$_{PP}$ /1 V$_{PP}$ (6 EnDats) 1 V$_{PP}$/EnDat	344 964-xx	344 965-xx

Note

A BC 120 can also be attached to the LE for the BF 120 at the same time.

TE 420 Operating Panel The IV and V keys are snap-ons, and can be switched. Id. Nr. 313 038-11 You can find an overview of the available key symbols on Page 2 – 15. Horizontal rows to match the design of the flat-panel display Id. Nr. 316 343-01	
BC 120 Visual Display Unit 15-inch color screen (640 x 480 pixels) Id. Nr. 313 037-02	
BF 120 Visual Display Unit 10.4-inch color flat panel display (640 x 480 pixels) Id. Nr. 313 506-02	

BTS 1x0 Monitor/Keyboard Switch

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

Id. Nr. 317 292-01	BTS 110 (2 x BC 120) (see figure at right)
Id. Nr. 329 965-02	BTS 120 (2 x BF 120)

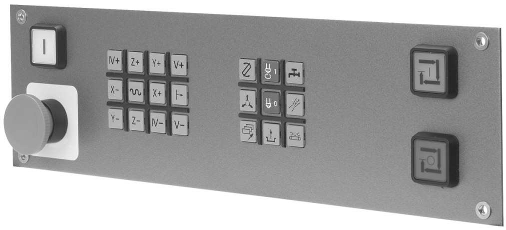
MB 420 Machine Operating Panel

Machine operating panel with snap-on (switchable) keys. You can find an overview of the available key symbols on Page 2 – 15.

Key assignment:

- Emergency stop
- Machine control voltage
- NC start, NC stop
- Five axis keys
- Rapid traverse
- Coolant
- Spindle start, spindle stop
- 7 keys for machine functions
  - Standard assignment: FN 1 to FN 5
  - Assignment for HEIDENHAIN basic PLC program: Retract axis, Tool change, Unlock tool, Menu selection →, Unlock door, Rinse water jet, Chip removal

Id. Nr. 293 757-33	Standard
Id. Nr. 293 757-45	Basic PLC program

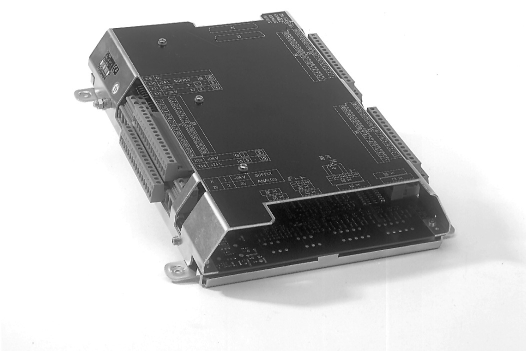


PL 410 B PLC Input/Output Unit
For the expansion of PLC inputs and outputs

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



HR 410 Handwheel

Portable handwheel with snap-on (switchable) keys. You can find an overview of the available key symbols on Page 2 – 15.

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 with PLC)
  - Spindle right, Spindle left, Spindle stop
  - NC start, NC stop, Spindle start (for 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 LE
Id. Nr. 281 429-xx	Extension to adapter cable
Id. Nr. 271 958-03	Dummy plug for emergency-stop circuit

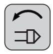
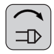
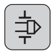
HR 130 Handwheel Panel-mounted handwheel Id. Nr. 254 040-05 With ergonomic control knob, radial cable outlet	

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, radial cable outlet
Id. Nr. 270 908-01	Handwheel selection switch

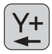
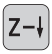
Key symbols for the spindle

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

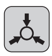
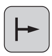
Axis direction keys for the principle axes

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

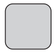
Key symbols for axis direction keys for rotary and secondary linear axes

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		

TS 220 Touch Probe

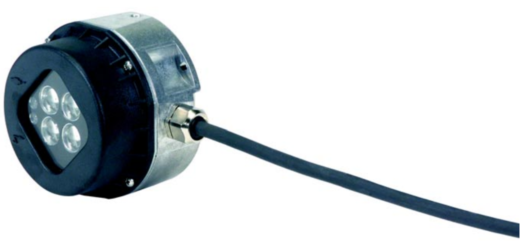
Touch-trigger probe with cable connection for workpiece setup, measurement during machining, and digitizing.

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

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 LE
Id. Nr. 310 197-xx	Adapter cable for connecting the EA 632 or the APE 652 with the LE



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 LE

Further components	Id. Nr.
Options for TNC 426 CB/PB, TNC 430 CA/PA	
Additional position input for a 6th or 9th axis (350 kHz)	311 537-51
Only TNC 430 PA: Position input for 3 additional axes with nominal speed command interface (350 kHz)	294 130-51
Ethernet interface	293 890-xx
Digitizing with triggering 3-D touch probe	286 405-01
Digitizing with a measuring 3-D touch probe (SP 2/1)	311 647-51
Options for TNC 426 M, TNC 430 M	
Ethernet interface	293 890-xx
Digitizing with triggering 3-D touch probe	286 405-01
Digitizing with a measuring 3-D touch probe (SP 2/1)	325 818-51
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 LE 426 M, LE 430 M (as of xxx xxx-3x)	317 505-05
11 μA_{PP} /1 V_{PP}	313 119-01

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 TNC 426 M and the TNC 430 M are described in the Technical Manual "Inverter Systems and Motors."

The components required for operating the TNC 426 and TNC 430 with non-HEIDENHAIN inverter systems are described in the "Technical Information for Operation of SIMODRIVE and POWER DRIVE inverter systems."

You will receive a set of supplementary pages every time changes are made to this manual.

2.3 Brief Description

<table>
<tr><th>Specifications</th><th>TNC 426</th><th>TNC 430</th></tr>
<tr><td colspan="3">Basic version with integrated motor control</td></tr>
<tr><td rowspan="2"></td><td colspan="2">TNC 426 PB, TNC 430 PA:
All position and speed inputs 1 V_{PP}
TNC 426 M, TNC 430 M:
All position inputs 1 V_{PP} or EnDat
All speed inputs 1 V_{PP} or EnDat</td></tr>
<tr><td>TNC 426 PB:
■ 5 axes plus spindle (up to 12 000 rpm) with position and speed inputs
■ 5 axes plus spindle (up to 30 000 rpm) with position and speed inputs
TNC 426 M:
■ 5 axes plus spindle (up to 12 000 rpm) with position and speed inputs
■ 5 axes plus spindle (up to 30 000 rpm) with position and speed inputs</td><td>TNC 430 PA:
■ 5 axes plus spindle (up to 30 000 rpm) with position and speed inputs
■ 6th axis with speed input
TNC 430 M:
■ 6 axes plus spindle (up to 30 000 rpm) with 6 position and 7 speed inputs
■ 9 axes plus spindle (up to 30 000 rpm) with position and speed inputs</td></tr>
<tr><td colspan="3">Basic version with analog speed command interface</td></tr>
<tr><td rowspan="2"></td><td colspan="2">Position inputs 1 V_{PP}</td></tr>
<tr><td>TNC 426 CB:
■ 5 axes plus spindle</td><td>TNC 430 CA:
■ 8 axes plus spindle</td></tr>
<tr><td colspan="3">Options</td></tr>
<tr><td rowspan="2"></td><td></td><td>TNC 430 CA:
■ Position input for the 9th axis
TNC 430 PA:
■ Position inputs for 3 additional axes with analog speed interface
■ Position input for the 6th axis</td></tr>
<tr><td colspan="2">■ Digitizing with triggering touch probe
■ Digitizing with measuring touch probe
■ Ethernet interface</td></tr>
</table>

Specifications		**TNC 426**	**TNC 430**
Display			
		■ 15-inch CRT color screen ■ 10.4-inch TFT color flat-panel display	
Program memory			
		Hard disk with > 2 gigabytes	
Input resolution and display step			
		Up to 0.1 µm for linear axes Up to 0.0001° for angular axes	
Interpolation			
	Straight lines	5 of 5 axes	5 of 9 axes
	Circle	■ 2 of 5 axes ■ 3 of 5 axes with tilted working plane	■ 2 of 9 axes ■ 3 of 9 axes with tilted working plane
	Helices	Superimposition of circular and linear paths	
Interpolation			
	Spline	Cubical splines can be executed	
Block processing time			
		From the hard disk: 4 ms	

Machine Integration	TNC 426	TNC 430
Feedback control		
Integral drive control	TNC 426 PB, TNC 426 M	TNC 430 PA, TNC 430 M
Analog speed command interface	TNC 426 CB	TNC 430 CA
Position loop resolution	$\frac{\text{Signal period}}{1024}$	
Cycle time for path interpolation	TNC 426 CB/PB, TNC 430 CA/PA: 3 ms TNC 426 M, TNC 430 M: can be set between 3 ms and 6 ms	
Cycle time for speed	TNC 426 PB/M, TNC 430 PA/M: 0.6 ms	
Nominal feed rate	TNC 426 PB/M, TNC 430 PA/M: $\frac{24000}{\text{No. of pole pairs}} \cdot \text{srew pitch} \cdot \text{min}^{-1}$	
	TNC 426 PB/M, TNC 430 PA/M: ■ Up to 60 m/min for encoders with 20 µm grating period ■ Up to 300 m/min for encoders with 100 µm grating period	
Speed	TNC 426 PB (Standard), TNC 426 M/12 000 rpm: $\frac{24000}{\text{No. of pole pairs}} \cdot \text{min}^{-1}$	
	TNC 426 PB (option), TNC 426 M/30 000 rpm: $\frac{60000}{\text{No. of pole pairs}} \cdot \text{min}^{-1}$	TNC 430 PA, TNC 430 M: $\frac{60000}{\text{No. of pole pairs}} \cdot \text{min}^{-1}$
	Volts-per-hertz control mode TNC 426 M/12 000 rpm: $\frac{24000}{\text{No. of pole pairs}} \cdot \text{min}^{-1}$ TNC 426 M/30 000 rpm: $\frac{60000}{\text{No. of pole pairs}} \cdot \text{min}^{-1}$	Volts-per-hertz control mode TNC 430 M: $\frac{60000}{\text{No. of pole pairs}} \cdot \text{min}^{-1}$
	TNC 426 CB: 100000 rpm	TNC 430 CA: 100000 rpm
Error compensation		
	■ Linear and nonlinear axis error ■ Backlash ■ Reversal spikes during circular movements ■ Offset ■ Thermal expansion ■ Stiction ■ Sliding friction	

Machine Integration		TNC 426	TNC 430
Integral PLC			
	PLC memory	Hard disk	
	Main memory (RAM)	512 KB	
	PLC cycle time	TNC 426 CB/PB, TNC 430 CA/PA: 21 ms TNC 426 M, TNC 430 M: can be set between 21 ms and 120 ms	
	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	TNC 426 PB/M: 13	TNC 430 PA/M: 13 TNC 430 CA: 3 with 9 NC axes plus spindle TNC 430 CB: 7 with 5 NC axes plus spindle
	Inputs for thermistors	3 (additional inputs as option)	
Commissioning aids			
		■ Oscilloscope ■ Trace function ■ Table function ■ Logic diagram ■ Log	
Data Interfaces			
		■ One each RS-232-C/V.24 and RS-422/V.11 with max. 115 Kbps ■ Expanded data interface with LSV2 protocol for external operation of the TNC ■ Option: Ethernet interface approx. 200 kbps to 1 Mbps	

User functions	TNC 426 and TNC 430
Program entry	HEIDENHAIN conversational and ISO
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
Contour approach and departure	■ Via straight line: tangential or perpendicular (APPR/DEP) ■ Via circular arc (APPR/DEP) ■ Via rounding radius (RND)
Tool compensation	■ Tool radius in the working plane, tool length ■ Radius compensated contour look ahead for up to 99 blocks (M120)
Cutting data table	For automatic calculation of speed and feed rate from various definable combinations of tool and workpiece materials
Constant contour speed	■ With respect to the path of the tool center ■ With respect to the tool cutting edge (M109, M110, M111)
3-D machining	■ Reduced feed rate during plunging (M103) ■ 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 position of the swivel head with the electronic handwheel during program run. The position of the tool tip does not change. ■ Jerk reduction ■ Spline ■ Tool perpendicular to contour ■ Tool radius compensation perpendicular to traversing and tool direction
Machining with rotary tables	■ Programming a contour on a cylindrical surface as if on a plane ■ Feed rate in mm/min (M116)
FK free contour programming	FK free contour programming in HEIDENHAIN conversational format with graphic support for workpiece drawings not dimensioned for NC
Subprogramming	Program section repeats, subprograms, program calls
Background programming	Creating or editing a program while another program is being run — also with graphical support
Fixed cycles	■ Peck drilling, tapping with or without a floating tap holder, reaming, boring, hole patterns, slot milling, rectangular and circular pocket milling, stud finishing, face milling of plane surfaces ■ OEM cycles (special cycles developed by the machine tool builder) can also be integrated ■ Contour pockets — also contour parallel ■ Contour train

User functions	TNC 426 and TNC 430
Coordinate transformation	■ Datum shift, rotation, mirroring ■ Scaling factor (axis specific) ■ Tilting the working plane
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 ■ Digitizing cycles
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
Programming aids	■ Pocket calculator ■ Structuring of part programs ■ Graphic support for the programming of cycles
Actual position capture	Actual positions can be transferred directly into the part program
Test graphics — display modes	Graphical simulation before a program run: ■ Plan view ■ Projection 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 while they are being entered (2-D pencil-trace graphics)
Program run graphics — display modes	Graphic simulation during real-time machining: ■ Plan view ■ Projection in three planes ■ 3-D view
Machining time	■ Calculation of approximate 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, each with 254 datums
Pallet tables	Multiple pallet tables with any number of entries for selection of pallets, part programs and datums

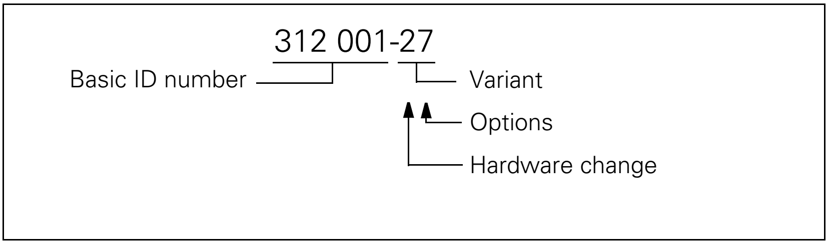
Export versions	TNC 426 CF, TNC 426 PF, TNC 426 ME	TNC 430 CE, TNC 430 PE, TNC 430 ME
Linear interpolation	4 of 5 axes	4 of 9 axes

Accessories	TNC 426 and TNC 430
Electronic handwheels	■ One portable HR 410 handwheel, or ■ One panel-mounted HR 130 handwheel, or ■ Up to 3 HR 150 panel-mounted handwheels via the HRA 110 handwheel adapter
	Superimpose handwheel positioning during program run (M118)
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
Digitizing of 3-D surfaces	■ With the TS 220 triggering 3-D touch probe and software module for the TNC (option) ■ Adapter kit for measuring 3-D touch probe (option) ■ PC evaluation software for digitized data: SUSA
Data transfer software	TNCremoNT, TNCremo
PLC development software	PLCdesign
Software for generating cycle structure	CycleDesign
PLC input/output unit	Up to four PL 410B or one PL 405B
	PL 410B Version 1: Additional 64 PLC inputs and 31 PLC outputs per PL PL 410B Version 2: Additional 64 PLC inputs and 31 PLC outputs as well as 4 analog inputs ± 10 V and 4 inputs for thermistors per PL PL 405 B: Additional 32 PLC inputs and 15 PLC outputs per PL

2.4 Hardware

2.4.1 Designation of the Logic Unit

ID number of the logic unit:



The basic ID number indicates hardware differences.
This first digit of the variant number indicates hardware changes.

The second digit of the variant number specifies the option:

Option number	Meaning
3	Export version with “digitizing with triggering touch probe” option
4	Standard version with “digitizing with triggering touch probe” option
7	Standard version with “digitizing with measuring and triggering touch probe” option
8	Export version without option
9	Standard version without option

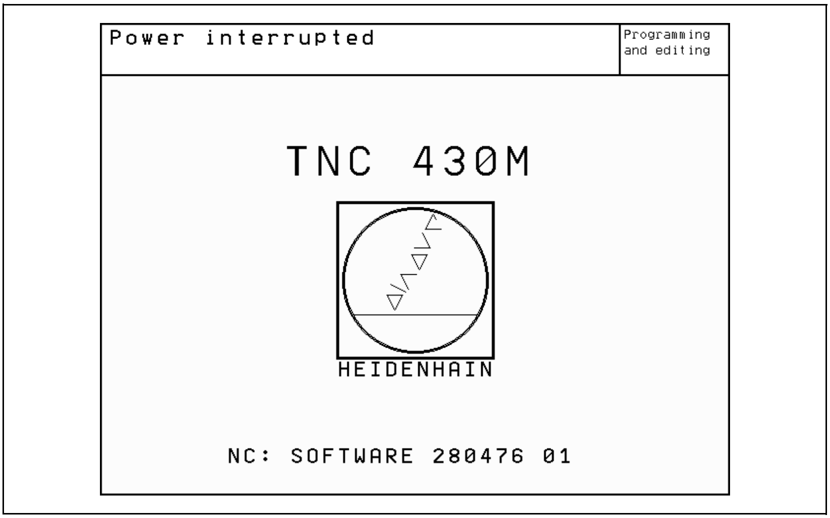
Variant	Changes to LE 426 C/P, LE 430 C/P
xxx xxx-2x	Initial version
xxx xxx-3x	Stronger rectifier in power supply
xxx xxx-4x	4 MB RAM; 3-row VGA connection; 3-phase current controller
xxx xxx-5x	Power supply unit with higher performance

Variant	Changes to LE 426 M, LE 430 M 324 990-xx to 324 997-xx	Changes to LE 426 M, LE 430 M 344 958-xx to 344 965-xx
xxx xxx-1x	–	flash EPROMs, all encoder inputs with EnDat interface
xxx xxx-2x	Initial version	–
xxx xxx-3x	New hard disk suspension, certain position encoder inputs with EnDat interface	–

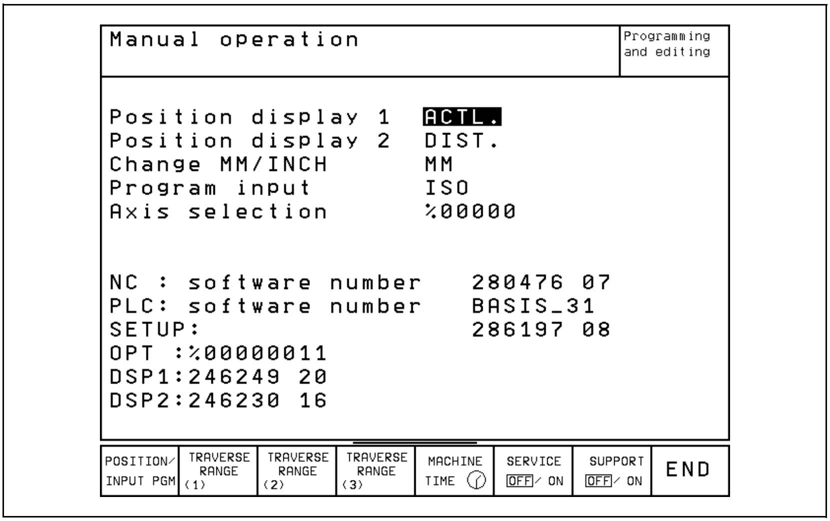
2.5 Software

2.5.1 Designation of the Software

The logic unit features a separate software for the NC and the PLC. The NC software is identified with an eight-digit number. The ID number is displayed briefly after the TNC is switched on:



If you press the MOD key in any operating mode, you can display the ID numbers of the NC software, the DSP software and the setup disks. If the hardware contains flash EPROMs, the letter “F” is displayed before the NC software Id. Nr.



Software type

The following software versions are available for the TNC 426 and TNC 430 controls:

NC software version	Setup disks	Export version
4 EPROMs 280 470-xx	1 disk 280 640-xx	4 EPROMs 280 471-xx
4 EPROMs 280 472-xx	3 disks 280 641-xx (to 280 641-05 2 disks)	4 EPROMs 280 473-xx
4 EPROMs 280 474-xx	3 disks 286 195-xx	4 EPROMs 280 475-xx
6 EPROMs 280 476-xx (Delivered software)	4 disks 286 197-xx (to 286 197-03 3 disks)	6 EPROMs 280 477-xx

NC software version	Data record for controls with flash EPROMs	
	Standard	Export
280 476-xx	340 436-xx	340 437-xx

Due to restrictions on the export of the TNC, HEIDENHAIN can also supply a special export version. This export version differs from the standard control though the installed NC software version. HEIDENHAIN releases a new NC software version whenever it introduces extensive new functions.

Certain software versions do not run on all hardware versions. Please consult the following table, which assigns each NC software version to a hardware version:

Hardware version	LE 426 C/P, LE 430 C/P	LE 426 M, LE 430 M 324 990-xx to 324 997-xx (EPROMs)	LE 426 M, LE 430 M 344 958-xx to 344 965-xx (flash EPROMs)
xxx xxx-1x	–	–	as of 280 476-07
xxx xxx-2x	280 470-xx 280 472-xx	280 474-xx 280 476-xx	–
xxx xxx-3x	280 470-xx 280 472-xx	280 474-xx 280 476-xx	–
xxx xxx-4x	280 470-xx 280 472-xx 280 474-xx 280 476-xx	–	–
xxx xxx-5x	280 470-xx 280 472-xx 280 474-xx 280 476-xx	–	–

2.5.2 Software Option

The following software options are available for the TNC 426 and TNC 430:

- Digitizing with triggering touch probe
- Digitizing with triggering and measuring touch probes

If you have ordered a TNC with an option, the software number will be supplemented by an option number displayed after the TNC is switched on.

The TNC can also be retrofitted with one of the options. Please contact HEIDENHAIN for further information.

Option	Option number	Id. Nr. of the adapter kit	Id. Nr. of the software module
Digitizing with triggering touch probe	1	286 405-01	246 051-01
Digitizing with the SP 2/1 triggering and measuring touch probe	11	TNC 426 CB/PB, TNC 430 CA/PA: 311 647-51 TNC 426 M, TNC 430 M: 325 818-51	–

2.5.3 PLC Software

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

2.5.4 NC Software Exchange

The following controls are equipped with EPROMs (**not** flash EPROMs):

- TNC 426 CB/PB
- TNC430 CA/PA
- TNC 426 M (324 990-xx, 324 991-xx, 324 994-xx, 324 995-xx)
- TNC 430 M (324 992-xx, 324 993-xx, 324 996-xx, 324 997-xx)

The following controls are equipped with flash EPROMs (**not** EPROMs):

- TNC 426 M (344 958-xx, 344 959-xx, 344 962-xx, 344 963-xx)
- TNC 430 M (344 960-xx, 344 961-xx, 344 964-xx, 344 965-xx)

The NC software and the English conversational language are stored in EPROMs. Other conversational languages are stored on the hard disk. If no current conversational languages are on the hard disk, load the English language through machine parameter MP 7230.x. If a software exchange becomes necessary, HEIDENHAIN provides new EPROMs and setup disks, or a new complete setup for controls with flash EPROMs.

Information about the cycles

Change the OEM cycles into binary format before reconversion, otherwise the TNC 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 provided on the setup disks of the NC software. For further information, refer to the User's Manual or the help texts for CycleDesign.

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	Directory cannot be opened
REMANENT PLC DATA NOT RESTORED	The file PLCMEM.A cannot be accessed
NOT ENOUGH SPACE	Not enough free memory on the hard disk
CONVERSION BIN ASC FAILED	A binary file has an incorrect format (e.g., binary format from an old NC software)
CONVERSION ASC BIN FAILED	An ASCII file on the hard disk is incorrect

NC software switch procedure on controls with EPROMs

The software must be exchanged only by trained personnel.

The READ_MP.A file on the provided floppy disks contains information on machine parameters. The README.TXT file provides notes on the software exchange.

Warning

Any contact with statically charged objects or handling without MOS protection can destroy the EPROMs!

- In the PROGRAMMING AND EDITING mode of operation, press the MOD key.
- Enter the code number 95148 and confirm with ENT.
- In the machine-parameter-editing mode of operation, press the MOD key.
- Ensure that the free space on the hard disk 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 PC software TNCBACK.EXE, TNCremo or TNCremoNT.
- Press the UPDATE DATA and CONVERT BIN -> ASC soft keys to change 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%

- The name and path of a log file can be entered after `Path =` in the header.

Danger

Danger of electrical shock! Disconnect the power before opening the unit.

- After conversion, switch the control off.
- Exchange the EPROMs on the processor board with the IC extraction/insertion tool: expansion slots, see graphic.
- After exchanging the EPROMs, switch the TNC on again.
- Complete or erase the machine parameters. Information about the machine parameters can be found on the READ_MP.A file on the first provided setup disk, or on the TNC in the directory PLC:\JH\.
- Exit the machine parameter editor: Press the END key. The message `LANGUAGE LOAD ERROR` appears.
- Connect the TNC to a PC through a serial data interface or by Ethernet.

- On the PC, enter the command SETUP or SETUP32 to copy the NC dialogs, HEIDENHAIN cycles etc. from the provided setup disks. After setup the control carries out a RESET.
  - DOS and Windows in the DOS window: Use the SETUP command, followed by the number of the PC's serial port (e.g., SETUP 2 for the COM2 port).
  - Windows 95, 98, NT: Use the SETUP32 command, followed by the number of the PC's serial port (e.g., SETUP32 2 for the COM2 port) or the TNC's IP address (e.g., SETUP32 160.1.180.21).

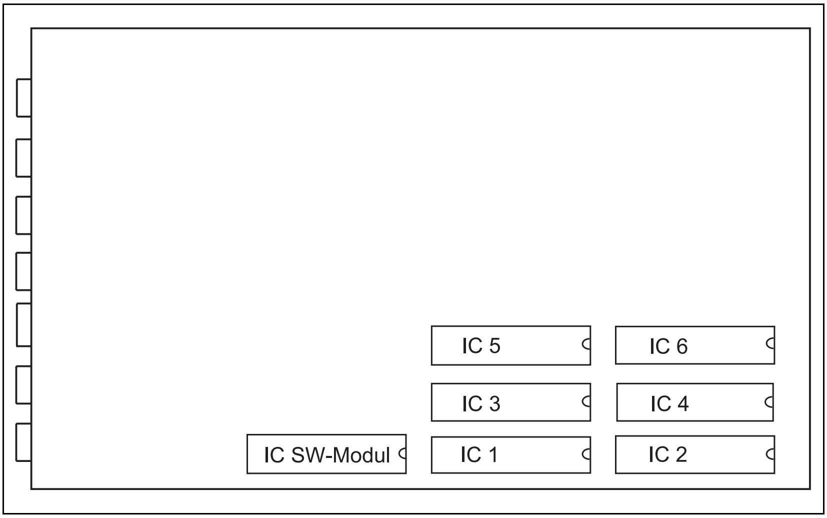
Note

As of NC software 280 476-17, 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 "Activating and deleting already existing NC software" on page 2 – 39.)

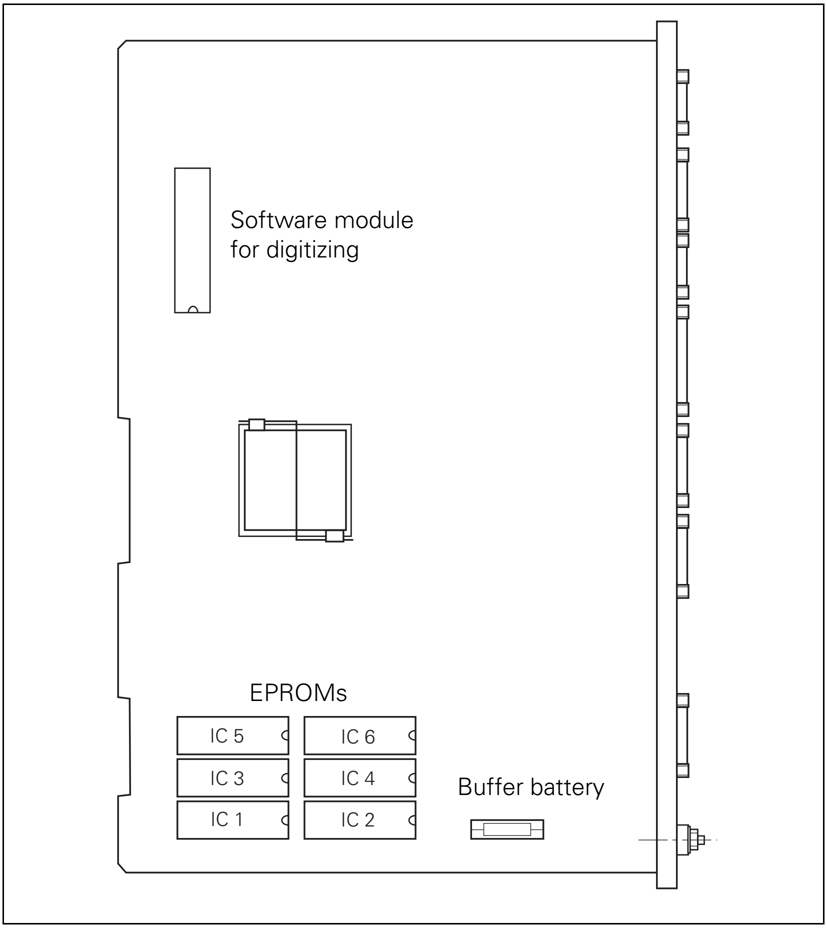
- On the TNC, switch to the PROGRAMMING AND EDITING mode of operation and press the MOD key.
- Enter the code number 95148 and confirm with ENT.
- In the machine-parameter-editing mode of operation, press the MOD key.
- Convert updated data to binary format: Press the soft key UPDATE DATA and CONVERT ASC -> BIN. The name of a log file may also be entered.
- Read-in files which you had saved to a PC.
- End of the NC software switch.
- With the COPY SAMPLE FILES soft key, the cutting data tables, the tables for tilted-axis geometry, and the table of M-function macros can be copied into the corresponding directory.

Position of EPROMs

LE 426 CB/PB and LE 430 CA/PA



LE 426 M and LE 430 M



NC software switch procedure on controls with flash EPROMs

The software must be exchanged only by trained personnel.

The READ_MP.A file on the provided data record contains information on machine parameters. The README.TXT file provides notes on the software exchange.

- In the PROGRAMMING AND EDITING mode of operation, press the MOD key.
- Enter the code number 95148 and confirm with ENT.
- In the machine-parameter-editing mode of operation, press the MOD key.
- Ensure that the free space on the hard disk 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 PC software TNCBACK.EXE, TNCremo or TNCremoNT.
- Press the UPDATE DATA and COVERT BIN -> ASC soft keys to change 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%

- The name and path of a log file can be entered after `Path =` in the header.
- Connect the TNC to a PC through a serial data interface or by Ethernet.
- Enter the SETUP or SETUP32 command on the PC to read-in the new NC software. After setup the control carries out a RESET.
  - DOS and Windows in the DOS window: Use the SETUP command, followed by the number of the PC's serial port (e.g., SETUP 2 for the COM2 port).
  - Windows 95, 98, NT: Use the SETUP32 command, followed by the number of the PC's serial port (e.g., SETUP32 2 for the COM2 port) or the TNC's IP address (e.g., SETUP32 160.1.180.21).

Note

As of NC software 280 476-17, 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 NC software from the hard disk. (See "Activating and deleting already existing NC software" on page 2 – 39.)

- Complete or erase the machine parameters. Information about the machine parameters can be found on the READ_MP.A file on the first provided setup disk, or on the TNC in the directory PLC:\JH\.
- Exit the machine parameter editor: Press the END key. The message **LANGUAGE LOAD ERROR** appears.
- On the TNC, switch to the PROGRAMMING AND EDITING mode of operation and press the MOD key.
- Enter the code number 95148 and confirm with ENT.
- In the machine-parameter-editing mode of operation, press the MOD key.
- Convert updated data to binary format: Press the soft key UPDATE DATA and CONVERT ASC -> BIN. The name of a log file may also be entered.
- Read-in files which you had saved to a PC.
- End of the NC software switch.
- With the COPY SAMPLE FILES soft key, the cutting data tables, the tables for tilted-axis geometry, and the table of M-function macros can be copied into the corresponding directory.

Activating and deleting already existing NC software

For controls with flash EPROMs, each new NC software is saved to its own directory in the SYS partition.

Note

For controls with EPROMs, only the contents of the setup disks are copied to the appropriate directories. The NC software automatically chooses the correct setup.
You can also delete setups that are no longer needed with the procedure detailed below:

To activate already existing NC software:

- While in the **Programming and Editing** operating mode, press the MOD key.
- Enter the code number 95148.
- Press the MOD key.
- Press the UPDATE DATA soft key.
- Press the DEL/SEL SETUP soft key.
- Use the arrow keys to select the desired NC software.
- Press the SELECT soft key.

To delete already existing NC software from the hard disk:

- While in the **Programming and Editing** operating mode, press the MOD key.
- Enter the code number 95148.
- Press the MOD key.
- Press the UPDATE DATA soft key.
- Press the DEL/SEL SETUP soft key.
- Use the arrow keys to select the desired NC software.
- Press the DELETE ALL soft key.
- Confirm the confirmation question with the YES soft key.

2.5.5 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 TNC at regular intervals. Data backup is described in detail in the “Readme” file, which is included on the disk.

2.6 Software Releases

2.6.1 NC Software 280 470-xx and 280 471-xx

NC software 280 470-01 (export version 280 471-01)

Release: 05/96

Initial version

NC software 280 470-02 (export version 280 471-02)

Release: 06/96

Improvements:

- M132 with TIME parameter
- Module 9035 parameter 21: control type
- M118, M120 also in ISO
- Cycle 27 cylindrical surface also in tilted working plane
- MP7680 bit 9 new
- MP2423, MP2425, MP2427, MP2433, MP2451, MP2451, MP7245, MP7250 removed
- MP2402 changed: current gain at maximum speed

NC software 280 470-03 (export version 280 471-03)

Release: 08/96

Improvements:

- New:

`GROSS POSITIONING ERROR`

NC software 280 470-04 (export version 280 471-04)

Release: 09/96

Improvements:

- MP6500 bit 4 and bit 5 new
- FN18: group numbers 350 and 500 new
- FN17: group number 500 new
- Cycles for tool measurement (31 to 33) expanded by the entry of a Q parameter in which the result of measurement is saved.

NC software 280 470-05 (export version 280 471-05)

Release: 12/96

Improvements:

- Rotary axes can be synchronized
- MP7682 bit 1 new
- In the compensation value tables .CMA and .COM the numbers of the axes are given instead of the names.
- Threshold for PLC: Time Out increased from 200 % to 300 %
- MP6500 bit 5, bit 6 and bit 8 new
- FN18: group numbers 51 and 52 new
- FN17: group number 210 new
- Input range of MP2500 and MP2501 increased to 1000
- The maximum number of points of all compensation value tables was increased to 1280.
- Coded NC error messages are displayed in plain language

NC software 280 470-06 (export version 280 471-06)

Release: 2/97

Improvements:

- The datum is set with the keys A B, C, X, Y, Z, U, V, W, a, b, c, x, y, z, u, v, w
- The software also runs on the special hardware of the LE 426 PB with spindle speeds up to 24 000 rpm.
- The NC software also runs on the new hardware of the LE 426 B and LE 430 with the Id. Nr. xxx xxx 4x.

NC software 280 470-07 (export version 280 471-07)

Release: 03/97

Improvements:

- MP2541 and MP2551 (band-rejection filter for spindle) new. Input is same as for MP2540 and MP2551 for the axes.
- Reversal spike compensation for circular movements with MP711.x to MP716.x was improved.

NC software 280 470-08 (export version 280 471-08)

Release: 5/97

Improvements:

- Hungarian conversational language new
- D760 new (offset for tilted axes, touch probe center misalignment)
- MP750 and MP752 new (backlash compensation)
- MP3143 expanded: 3 = same as input value 1, except that the second reference mark is evaluated.

NC software 280 470-09 (export version 280 471-09)

Release: 06/97

Improvements:

- MP6500 expanded: bit 10 and bit 11

NC software 280 470-10 (export version 280 471-10)

Release: 07/97

NC software 280 470-11 (export version 280 471-11)

Release: 06/98

Improvements:

- Input range of MP2510 and MP2511 extended to 30 000
- MP6500 bit 12: Consider PLC datum shift during tool measurement
- MP7500 bit 3: Displace datum with rotary tables in connection with "Tilt working plane" function

NC software 280 470-12 (export version 280 471-12)

Release: 03/99

Improvements

- Cycle 201, 202, 203: The default value in Q208 was increased to 30 000 mm or 1200 inches
- Cycle 210, 211: The starting point for the finishing cut is approached at the machining feed rate
- Cycle 212, 214: The contour is approached at the programmed machining feed rate

2.6.2 NC Software 280 472-xx and 280 473-xx

NC software 280 472-01 (export version 280 473-01)

Release: 04/97

Improvements:

- New function “Fast contour milling”: Cycle 32 or G62 and MP1096
- Automatic calculation of cutting data
- TCPM (Tool Center Point Management): With M128 you can superimpose manual axial machine movements during program run, whereby the offsets of the tilting axes are automatically compensated.
- Additional information with the HELP key
- Input menu for fixed input values is selected with the GOTO key.
- New pallet management
- Freely definable tables
- NC blocks can be transferred in spline format.
- More memory on the hard disk (1.5 GB)
- The MOD function PGM MGT enables the user to choose between standard and extended file management.
- In the status display the positions of all nine axes are shown. The spindle position overwrites the ninth axis.
- The progress of the copying process is shown in a pop-up window.
- The number of Q parameters was increased from 299 to 399.
- Q parameters are also permitted in FK blocks
- M110 is also effective in the contour pocket cycle
- Cycle 204: Back boring
- With MP7682 bit 2 you can define whether rotary axes should always be positioned by the shortest path.
- It is now possible to enter a chamfering feed rate in an NC block for chamfering (CHF).
- Cycle 19 “Working plane” has been extended with parameters for feed rate and safety clearance. (This applies only if the cycle positions tilted axes, which is defined in MP7500.)
- M114 can also be used with non-controlled axes or PLC axes.
- Hungarian conversational language added
- All soft keys appear in the defined conversational language.
- Language-dependent soft keys for OEM cycles
- The soft keys for FK programming do not appear until you have pressed the FK key.
- Soft key F for feed rate in the manual operating modes
- New soft key: JOG INCREMENT OFF/ON
- New soft key: HIDE TOOLS OFF/ON: In the tool table, only the tools in the tool magazine are displayed
- New soft keys for copying fields in the tool table
- PLC soft keys can be appended to NC soft-key rows.
- Ethernet: It is possible to enter the name of a network printer.
- The probe results of the manual probing function can be transferred immediately to the datum tables.
- MP6170, MP6171: multiple measurement with measuring tolerance
- Separate set of calibration data for TS and TT for every traverse range

- With MP6500 bit 4 you can define whether speed should be limited to 1000 rpm during tool measurement with TT.
- MP6500 bit 9: Automatic determination of the basic rotation for tool measurement with the cubical probe contact
- W760: Angular misalignment of the tilting axes for automatic adjustment of touch probe center misalignment
- The calibration data of the TS can be saved in the tool table by soft key.
- Cycle 31 to 33 (tool measurement) were expanded by the input field "Q parameter for result."
- With MP6500 bit 5 and bit 6 you can define the reaction to tool breakage.
- FN17, FN18 ID990 NR1 behavior during programmed probing
- FN17: ID210 NR6 Tilting the working plane during program run active/inactive
- FN17: ID50 Overwrite tool table
- FN17: ID210 Overwrite basic rotation
- FN18 ID350 Extended touch probe data
- FN23: CDATA Calculating the circle center from three probe points
- FN24: CDATA Calculating the circle center from four probe points
- FN25: Setting the datum
- ISO: Cycles with numbers greater than 200 can be programmed with graphic support (also OEM cycles)
- ISO: Cycles G75 and G76 (rectangular pocket) now include an input box for corner radius
- ISO: Parameter H (limit angle) can be entered after M112
- ISO: G60 Running digitized data new
- MP2000 removed. Digital axes can be defined in MP120
- In the compensation value tables COM and CMA you select the columns for the desired axes with soft keys.
- Nonlinear axis-error compensation: maximum number of compensation points increased from 640 to 1280
- A formula can be entered in MP2020 (distance per motor revolution).
- MP2541, MP2451: Band filter for spindle
- The number of tools in the tool table was increased from 254 to 32 767.
- M4019: Reversing the counting direction of the position encoder on the spindle
- Cooperative multitasking in the PLC (SPAWN command)
- Automatic tool recognition (BIS)
- String operand S#Axx new
- Module 9019: Check program stack
- Module 9035: Expanded by parameters 3, 1000, 1001
- Module 9038: Read axis information
- Module 9096: Erasing a line in the tool table
- Module 9112: Transmitting ASCII characters via RS-232-C
- Module 9113: Receiving ASCII characters via RS-232-C
- Module 9151: Select traverse range and axis designation

- Module 9200/9201: Expanded (PLC soft keys can be appended to NC soft-key rows)
- Module 9215: PLC pop-up window
- Module 9270: Read from OEM.SYS
- Module 9271: Write to OEM.SYS
- Automatic offset compensation of encoder signals
- Oscilloscope records can be saved in a file.
- MP7365.5: Selected channel in oscilloscope (input $00000FF)

NC software 280 472-02 (export version 280 473-02)

Release: 07/97

Improvements:

- Cycle 32 changed to “tolerance”
- M134 new
- System file TNC.SYS new
- MP6500 expanded: Bit 10 probing routine, bit 11 tool checking and changing the tool table
- MP7500 expanded: Bit 3 Setting the datum in a tilted coordinate system
- Editor for creating the format of freely definable tables
- FN18: ID200 and ID270 new
- FN17: ID350 new
- M4161 new
- PLC commands BTX, BCX and BSX new

NC software 280 472-03 (export version 280 473-03)

Release: 08/97

Improvements:

- Spline blocks also in tilted working plane
- MP7680, bit 10 new (spline curve at compensated outside corners)
- The software also runs on the old hardware of the LE 426 CB/PA and LE 430 CA/PA, but with reduced functions.
- Cycle 19: “Setup clearance” input box new

NC software 280 472-04 (export version 280 473-04)

Release: 10/97

Improvements:

- Code words LSV2TIME0 to LSV2TIME2 new
- Module 9038 expanded by transfer value –1
- DR2 can be defined in TOOL CALL block
- Spindle DSP limits max. torque to 2.5 · rated torque

NC software 280 472-05 (export version 280 473-05)

Release: 11/97

Improvements:

- Maximum spindle speed without spindle DSP was increased to 12 000 rpm.
- Maximum spindle speed with spindle DSP was increased to 24 000 rpm.
- MP6180.x, MP6181.x, MP6182.x: Approximate position of ring gauge center for probing cycle CALIBRATE TS
- MP6185: Distance below top surface for probing during calibration
- MP7471: Maximum speed of the linear axes for compensatory traverse by positioning the angular axes with M128.
- FN18: ID505 Datum table selected?
- FN18: ID1010 Does MP exist?
- Module 9135: Switch on infrared probe
- D364: Nominal speed new
- D368: Actual speed new
- Language-dependent text blocks in the print masks of the probing cycles
- New code word: LOGBOOK

NC software 280 472-06 (export version 280 473-06)

Release: 12/97

Improvements:

- FN17: ID990 NR2 Switch sensor monitoring on or off
- FN17: ID990 NR3 Transfer sensor data to tool table
- FN17: ID990 NR4 Coordinate transformation
- New timers T96 to T143
- M4065: All workpiece dimensions are OK
- M4066: Workpiece must be remachined
- M4067: Workpiece to be scrapped
- New touch probe cycles

NC software 280 472-07 (export version 280 473-07)

Release: 02/98

Improvements:

- MP2180: PWM frequency
- MP6500 bit 12: Consider PLC datum shift during tool measurement
- MP6120, MP6350, MP6360, MP6520: Minimum input value reduced to 1
- MP7260: Maximum input value reduced to 30 000
- MP7683 bit 3: Behavior at reaching end of pallet table
- FN18 ID50: Read data from tool table
- FN18 ID220 NR2: Read current datum shift
- FN18 ID220 NR4: Read current PLC datum shift
- Module 9008: Read certain inputs of PLC input/output unit
- Module 9009: Set certain outputs of PLC input/output unit
- Module 9145: Automatic actual and nominal value transfer

NC software 280 472-08 (export version 280 473-08)

Release: 07/98

Improvements:

- MP2510 and MP2511: Maximum input value reduced to 30 000
- MP2191: Decelerating spindle at EMERGENCY STOP
- MP6161: M function for orienting infrared probe before measurement
- MP6162: Orientation angle
- MP6163: Minimum difference before executing an oriented spindle stop
- New strings S4 to S7
- FN17 ID50: Overwrite data in tool table
- FN17 ID420: Effectiveness of cycles 7, 8, 10, 11, 26 and 19
- Module 9032: “No system memory” error code
- Module 9071, 9082, 9210: Transfer of an immediate string
- “Shut down” soft key to shut down the control
- Automatic tool compensation with the touch-probe cycles 421 to 426
- ISO: With G36 (spindle orientation) it is possible to program a Q parameter as an angular value.

NC software 280 472-09 (export version 280 473-09)

Release: 10/98

Improvements:

- MP6500 bit 13: Tilted system in which the tool is measured
- Module 9210: Path and name of the screen mask as an immediate string
- Cycle 420: Determine angle in the touch probe axis
- Cycle 427: Automatic length and radius compensation

NC software 280 472-10 (export version 280 473-10)

Release: 11/98

Improvements:

- The version of the setup disks is also displayed via the MOD key.

NC software 280 472-11 (export version 280 473-11)

Release: 11/98

Improvements:

- MP334: Number of grating periods between the zero pulses on encoders with distance-coded reference marks

NC software 280 472-12 (export version 280 473-12)

Release: 12/98

Improvements:

- Cycle 201, 202, 203: The default value in Q208 was increased to 30 000 mm or 1200 inches
- Cycle 212, 214: The contour is approached at the programmed machining feed rate
- Cycle 210, 211: The starting point for the finishing cut is approached at the machining feed rate
- Cycle 421, 422: A basic rotation or a rotation via cycle 10 is considered for the probing direction

NC software 280 472-13 (export version 280 473-13)

Release: 06/99

NC software 280 472-14 (export version 280 473-14)

Release: 12/99

Improvements:

- When the control is shut down, the hard disk is put into sleep mode.
- MP2600: Input range changed

2.6.3 NC Software 280 474-xx and 280 475-xx

NC software 280 474-01 (export version 280 475-01)

Release: 12/97

Improvements:

- MP115: Signals at the encoder inputs
- MP120, MP121: Assignment of the nominal speed value outputs by indicating connector
- MP7500 bit 4: Compensation of offset by exchanging spindle head
- MP7550: Angular coordination of tilting element
- Rapid traverse speed can be reduced with the F MAX soft key
- The radius compensation RR/RL in conjunction with M128 is effective in the plane perpendicular to the tool axis programmed in TOOL CALL.
- With the cycles 6, 7, 8, 16, 17 and 18 only one line can be digitized.
- Q150 to Q167 and Q180 to Q182 are used for the measuring results of the touch-probe cycles.
- New touch-probe cycles for workpiece measurement, datum setting and probe calibration.
- F AUTO can be programmed in OEM cycles.
- In the creation of OEM cycles, INCH programs are supported with a separate standard value entry.

NC software 280 474-02 (export version 280 475-02)

Release: 01/98

NC software 280 474-03 (export version 280 475-03)

Release: 04/98

Improvements:

- MP111: Position encoder input for first and second spindle
- MP112: Speed encoder inputs for the axes
- MP113: Speed encoder input for first and second spindle
- MP121: Nominal speed value output for first and second spindle
- MP2180: PWM frequency
- MP2191: Decelerating spindle at EMERGENCY STOP
- MP2510, MP2511: Maximum input value reduced to 30 000
- MP2900, MP2910, MP2920, MP2930: Reserved
- MP4020 bit 5: Single or double spindle operation
- MP6120, MP6350, MP6360, MP6520: Minimum input value reduced to 1
- MP6540: Safety clearance to probe contact of TT130 during tool measurement
- MP7262: Maximum index number for indexed tools
- MP7683 bit 3: Behavior at reaching end of pallet table
- MP13010 to MP13520: Machine parameters for second spindle
- FN18 ID50: Read data from tool table
- FN18 ID220: Read current PLC datum shift
- M4065: All workpiece dimensions are OK
- M4066: Workpiece must be remachined
- M4067: Workpiece to be scrapped

- New timers T96 to T143
- New strings S4 to S7
- W266: Index number of a programmed indexed tool
- D604: Maximum possible spindle speed
- Module 9008: Read certain inputs of PLC input/output unit
- Module 9009: Set certain outputs of PLC input/output unit
- Module 9088: Display M functions in status window
- Module 9091: Determine line number of tool in tool table
- Module 9145: Automatic actual and nominal value transfer
- Module 9175: Activate spindle
- Module 9202: A PLC window can be activated if the table editor is active
- Two spindles can be operated alternately
- The active M functions can be displayed with the STATUS M FUNCT. soft key.
- Indexed tools can be entered
- Program sections can be marked, deleted and copied in the NC editor
- ISO: Tool offsets can be entered when the tool number is programmed
- With M128 a feed rate can be entered for the maximum speed of the compensation traverse for the linear axes
- Cycle 205: Universal pecking
- Cycle 206: Tapping
- Cycle 207: Rigid tapping
- Cycle 208: Helical finish milling
- Cycle 200: Dwell time at bottom of hole
- Cycle 203: Retraction path at chip breaking
- Cycle 220: Approach clearance height during machining
- Cycle 210, 211, 212, 213, 214, 215, 220, 221: Preposition with positioning logic
- Cycle 410 to 418: Set datum not equal to zero

NC software 280 474-04 (export version 280 475-04)

Release: 05/98

Improvements:

- MP860: Define axis as torque-master-slave axis
- MP2900: Torque bias of the torque-master-slave control
- MP2910: Gain in the torque-master-slave control
- MP2920: Torque distribution of the torque-master-slave control
- MP2930: Speed rating factor of the torque-master-slave control
- MP7160 bit 1: Limiting the spindle speed during rigid tapping
- MP7263 bit 0: Display POCKET TABLE soft key
- MP7440 bit 6: Activating M134
- MP7441 bit 0: Suppress error message when machining cycle is called without M3 or M4
- MP7683 bit 4: Edit pallet table
- Indication of position of machining plane expanded
- FN17 ID50: Overwrite data in tool table
- Module 9092, 9093: Element numbers for tool number and index
- Module 9147: Assign new Ref value
- Module 9155: Switch axis from controlled/uncontrolled state
- Module 9156: Switch axis from uncontrolled/controlled state
- Module 9220: Traversing the reference marks in each operating mode and with PLC axes
- Module 9225: Compensation value for zero pulse
- Master-slave torque control
- ISO: With G36 (spindle orientation) it is possible to program a Q parameter as an angular value.
- ISO: In cycle G80, a feed rate and a safety clearance can be programmed as an option.
- Cycle 202, 204: Input parameters for angle in oriented spindle stop
- Several pocket tables possible
- M104 reactivates the manually set datum of all axes
- The ID number of the DSP software is shown in the system info
- Automatic tool compensation and monitoring of the breakage tolerance in the touch-probe cycles 421 to 426
- With the soft key EDIT PALLET, the current pallet table can be edited in pallet operation.
- Face milling: With M128 in connection with LN blocks, the tool is held perpendicular to the contour.
- Peripheral milling: With M128 in connection with RR/RL, the negative radius oversize (DR) is compensated perpendicular to the machining and tool direction.

NC software 280 474-05 (export version 280 475-05)

Release: 06/98

Improvements:

- MP6150, MP6361, MP6550: Maximum input value increased to 20 000
- MP7266.27: PLC value
- MP7600.0: Reserved
- MP7600.1: PLC cycle time
- Indication of position of machining plane expanded
- FN17 ID50: Write data to the PLC value column of the tool table
- FN18 ID50: Read data from PLC value column
- FN18 ID360: Read last datum of a manual touch-probe cycle or last touch point of the touch-probe cycle 0
- Module 9092, 9093, 9094: Element number 26 (PLC-VAL)
- Module 9145: With PLC axes, a strobe or M4176 = 1 is not necessary
- C-axis operation
- PLC axes can be operated with velocity feedforward control
- If the axis is moved between opening and closing the position control loop, the function "Approach position" is activated
- New column "PLC value" in the tool table
- In the pallet editor, you can transfer actual or reference values as well as the values from the previous manual touch-probe cycle with the key "Actual position capture"
- ISO: Cycle G128 (Cylinder surface slot)

NC software 280 474-06 (export version 280 475-06)

Release: 07/98

NC software 280 474-07 (export version 280 475-07)

Release: 08/98

Improvements:

- MP1094: Cutoff frequency for HSC filter
- MP1220 Removed
- MP1390 Removed
- MP1392: Axis-specific switching between operation with following error or velocity feedforward control
- MP2560, MP2560: Low-pass filter in speed control loop
- MP4020 bit 6: Reserved
- MP4020 bit 7: Transferring the values of the Pt100 inputs
- MP6161: M function for orienting infrared probe before measurement
- MP6162: Orientation angle
- MP6163: Minimum difference before executing an oriented spindle stop
- MP7267.5: Tool name
- MP7367.x: Colors for the large PLC window
- MP7500 bit 5: Tilting axis settings during datum setting
- MP7600.0: Position controller cycle time
- MP7650: Axis-specific count direction for handwheels
- FN17 ID420: Effectiveness of cycles 7, 8, 10, 11, 26 and 19
- FN17 ID960: For touch-probe cycle 3, it is possible to switch between X12 and X13

- FN26, FN27, FN28: Open, describe and read numeric cells for freely-definable tables
- W1062: Disable axis-specific handwheel pulses
- Module 9153: Define probe axis for manual measurement
- Module 9211: Assign identifiers for fields of the large PLC window
- Module 9280: Start NC macro (Run pallet entry)
- The gear range from W256 is output when the spindle speed is 0
- New column TNAME in the pocket table
- New column LOCK in the pallet table
- Format instructions for the large PLC window expanded
- New code words MPPASSWORD and MPLOCKFILE for OEM.SYS
- New code word PALEPILOG for NCMAKRO.SYS
- During start-up of the control, a customer-specific company logo can be displayed
- "Shut down" soft key to shut down the control
- The cycle structure can consist of up to 9 cycle trees
- M117: M116 is switched off
- M136, M137: Switching the contour feed from mm/min to mm/spindle revolution and vice versa
- M138: Axes with M114, M128 and "Tilt the working plane," but without M116, can be indicated.
- With touch-probe cycle 3, you can probe without retraction
- The block number for mid-program startup or the block number for Test Run is input in a pop-up window.

NC software 280 474-08 (export version 280 475-08)

Release: 09/98

Improvements:

- ISO: New soft key for ordering block numbers with constant increment

NC software 280 474-09 (export version 280 475-09)

Release: 10/98

Improvements:

- MP334: Number of grating periods between the zero pulses on encoders with distance-coded reference marks
- MP7289: Display step for spindle position
- FN18 ID200: Determine current tool radius with sign
- FN17 ID510: Determine current line of a pallet table
- FN17 ID510: Determine last line of the NC program of the current pallet
- Module 9210: Transfer of an immediate string
- Module 9247: All tables can be searched for contents in a field
- Module 9290: Select NC program
- Cycle 405: Correct workpiece angular tilt via C axis
- Cycle 420: Determine angle in the touch probe axis

NC software 280 474-10 (export version 280 475-10)

Release: 12/98

Improvements:

- MP1521: Transient response during acceleration and deceleration
- MP2360, MP2361: Time constant for deceleration of spindle at EMERGENCY STOP
- MP2391: Max. braking performance of spindle at EMERGENCY STOP
- MP6500 bit 13: Tilted system in which the tool is measured
- MP7160 bit 2: Spindle in position control loop during rigid tapping
- MP7245: Disable auxiliary cycles
- FN18 ID20: Determine index of active tool
- FN18 ID990: Program run in mid-program startup or automatic mode?
- FN18 ID280: M128 active/not active
- FN18 ID280: Determine programmed feed rate with M128
- Module 9035: Transfer values for M128 active and status of pallet processing
- Module 9159: Interrogate which drives are to be switched off in 200 ms (only TNC 426 M, TNC 430 M)
- Module 9169: Define axes for which I32 does not cause drive switch-off (only TNC 426M / TNC 430M).
- Module 9120, 9123, 9221, 9223: New error code when programming a non-controlled axis.
- The control type, NC software, files selected and control-in-operation symbol status are recorded in the log book.
- Format instructions for the large PLC window expanded
- The fast PLC input defined in MP4130.0 no longer causes automatic switch-off of the drive.
- The version of the setup disks is also displayed via the MOD key.
- With the manual touch-probe cycles, the datum must be set by soft key.
- Cycle 427: Automatic length and radius compensation
- Cycle 210, 211: The starting point for the finishing cut is approached at the machining feed rate
- The default value in Q208 of the machining cycles 201, 202 and 203 was increased to 30 000 mm/min or 1200 inch/min.
- Mid-program startup in a radius-compensated NC program is only necessary if the program was interrupted at a transitional spline.
- Current *.CDF and CONSTCYC.CDC file on the setup disks
- With manual touch-probe cycles, the measuring result can be transferred to the datum table or become the datum via soft key

NC software 280 474-11 (export version 280 475-11)

Release: 12/98

Improvements:

- MP1090: Input range expanded to 0.1 to 1000.0
- Cycle 421, 422: A basic rotation or a rotation via cycle 10 is considered for the probing direction

NC software 280 474-12 (export version 280 475-12)

Release: 03/99

Improvements:

- MP2221: Monitoring the reference pulse of the spindle-speed encoder
- M4181: NC program selected
- Input value of the rated frequency in the motor table for asynchronous motors increased from 999.9 Hz to 2000.0 Hz
- Module 9175: Error code approach position active
- Module 9281: Set cursor on a line in the selected pallet table
- Cycle 207, 17, 18: Maximum input value from Q239 expanded to 99.9999 mm
- New error number 1068 (Datum table?) for the function FN14: ERROR

NC software 280 474-13 (export version 280 475-13)

Release: 06/99

Improvements:

- MP340: External interpolation
- In the manual operating modes, the highest axis feed is stored in D388
- In the format instructions for the large PLC window, special characters can be entered with /xYY.
- With C-axis operation, the bit for the spindle or the axis can be transferred to modules 9161 and 9162.
- Module 9035: Axis-specific handwheel superpositioning (M118)
- Module 9040 and 9041: Reference values with calculated backlash compensation
- Module 9044: Spindle coordinates in the form 0.0001°
- Module 9122: Interrogate whether PLC axis has reached target position
- Module 9281: Datum shift and datum set can still be carried out, even if the line is disabled by an entry in LOCK.
- If a customer-specific company logo is displayed, the control description is no longer shown.
- Cycle 17, 18: If an interruption occurs during a tilted working plane, the soft key MANUAL TRAVERSE appears.

NC software 280 474-14 (export version 280 475-14)

Release: 07/99

Improvements:

- MP7500: Tilting-axis positioning during datum setting
- FN17 ID501: Write to REF-referenced datum table
- FN18 ID501: Read from REF-referenced datum table
- Module 9189: Shut down the control

NC software 280 474-15 (export version 280 475-15)

Release: 10/99

Improvements:

- MP2221: Monitoring of rotational direction for spindle with integral DSP
- Input range of the STR column in the motor tables expanded
- When operating two spindles, the speed encoder may be disconnected and reconnected
- Direction monitoring of digitally controlled spindles
- M4179: Control is shut down
- M4220: Error from PET table/F stop active
- M4221: Error from PET table/NC stop active
- M4222: Error from PET table/EM. STOP active
- Before the PLC program is converted, the PLC outputs are reset.
- Module 9002, 9003, 9005, 9008, 9009: Error code 24 (Module was called in the submit job or spawn job) added
- Module 9130: New error codes 1 (Invalid analog output) and 2 (Disabled analog output) added
- Module 9189: The drive ready signal is removed
- ISO: The Q key of the numerical keypad or the ASCII keyboard may be used
- When the control is shut down, the hard disk is put into sleep mode.
- Improved search function
- Cycles 400 to 403: An error message appears if a measurement in a tilted plane is to be carried out
- Cycle 204: A value with digits after the decimal point may be entered in Q255

NC software 280 474-16 (export version 280 475-16)

Release: 03/00

Improvements:

- MP420, MP430: Changes no longer lead to a control reset
- MP2600: Input range changed
- MP6510: Second measuring error added
- MP7640, MP7641, MP7645 are no longer shown through code number 123
- A German and an English machine parameter description, as well as the configuration files for CycleDesign, are saved in PLC:\JH.
- M141: Suppress touch probe monitoring

NC software 280 474-17 (export version 280 475-17)	Release: 09/00
NC software 280 474-18 (export version 280 475-18)	Release: 06/01
NC software 280 474-19 (export version 280 475-19)	Release: 08/01

2.6.4 NC Software 280 476-xx and 280 477-xx

NC software 280 476-01 (export version 280 477-01)

Release: 05/99

Improvements:

- MP340: External interpolation
- MP1097, MP1098: Input range expanded from 1 to 1000 [m/s^3] to 0.1 to 1000.0 [m/s^3]
- MP1150 expanded: MP1150.1 Time duration for which monitoring functions must remain switched off; MP1150.2 Time duration for which monitoring functions must remain active
- MP2221: Monitoring the reference pulse of the spindle-speed encoder
- MP2360, MP2361: Time constant for deceleration of spindle at EMERGENCY STOP
- MP2391: Max. braking performance of spindle at EMERGENCY STOP
- MP2393: Performance limit of spindle motor
- MP4030: Assignment physical/logical PL
- MP6585, MP6586: Monitoring the position of the rotary and additional linear axes during the tool measurement cycles
- MP7245: Disable auxiliary cycles
- MP7266.28: Center offset of sensor in main axis
- MP7266.29: Center offset of sensor in secondary axis
- MP7266.30: Spindle angle during calibration
- MP7411: One block/several blocks of sensor calibration data
- MP7500: Cycle 19 with spatial angle C not equal to 0
- MP7530: Input of a formula for temperature compensation with swivel heads and tilting tables
- MP73xx: New default values for color settings
- HSC filter: The jerk value in MP1097.x is for curvature changes
- If the fast PLC input defined in MP4130.0 is set, the drive is no longer switched off automatically.
- New tilting axis combinations for spatial angle
- M4181: NC program selected
- Input value of the rated frequency in the motor table for asynchronous motors increased from 999.9 Hz to 2000.0 Hz
- Several blocks of touch probe calibration data in the tool table
- In the manual operating modes, the highest axis feed is stored in D388
- Maximum spindle speed with spindle DSP was increased to 30 000 rpm.
- After the drive is switched on, the control waits 1.2 seconds for the “ready” signal.
- FN17 ID230: Overwrite software limit switches
- FN18 ID990: Number of axes that are programmed in the current datum table
- Module 9035: Axis-specific handwheel superpositioning (M118)
- Module 9040 and 9041: Reference values with calculated backlash compensation

- Module 9122: Interrogate whether PLC axis has reached target position
- Module 9151: The axis designations remain the same even if other machine parameters are edited
- Module 9152: Selection of axis display, designation and traverse range
- Module 9158: Limiting the maximum torque of an axis
- Module 9169: Define axes for which I32 does not cause drive switch-off (only TNC 426M / TNC 430M).
- Module 9175: Error code approach position active
- Module 9275: Write ASCII data to the log book
- Module 9276: Write contents of operands to the log book
- Module 9281: Set cursor on a line in the selected pallet table
- Cycle 33: Thread on taper
- The values from the offset adjusting via code number 75368 are displayed in a pop-up window
- With the machine parameters MP910.x to MP951.x, the coordinates can be transferred with the "Actual position capture" key.
- M136: Switching from mm/min to mm/spindle revolution
- Feed rate with three decimal places
- Number of transfer parameters for OEM cycles increased to 32
- The last input value is suggested as default value with the "F" and "F MAX" soft keys.
- M91, M92: Linear movements in the machine coordinate system with tilted working plane
- File access in LSV2 protocol via Ethernet with TNCremo
- New error number 1068 (Datum table?) for the function FN14: ERROR
- Cycle 207, 17, 18: Maximum input value from Q239 expanded to 99.9999 mm
- DIN/ISO: Cycles for tool measurement with G480 to G483
- Cycle 247: Set datum during program run via datum table
- Traversing the reference marks is carried out with a new screen layout
- Soft key "Optional stop": Stop program where M01 is programmed
- Soft key "Edit datum table": Edit datum table
- Error messages for the first spindle are marked with "S1," and for the second spindle with "S2"
- DSP error messages were classified
- M112: Feed rate limit so that jerk is not exceeded, exception: F MAX is programmed

NC software 280 476-02 (export version 280 477-02)

Release: 05/99

Improvements:

- In the format instructions for the large PLC window, special characters can be entered with /xYY.
- Module 9281: Datum shift and datum set can still be carried out, even if the line is disabled by an entry in LOCK.
- In ASCII element fields, line breaks occur after whole words.

NC software 280 476-03 (export version 280 477-03)

Release: 08/99

Improvements:

- MP21: Encoder monitoring for first and second spindle
- MP1089: Axis-specific jerk for Pass Over Reference Point mode
- MP2590, MP2591: Steepness of the braking ramp during an emergency stop
- MP4020: Automatically activate Return to Contour after an external emergency stop
- MP7281: Depiction of the NC program
- MP7500: Tilting-axis positioning during datum setting
- MP7683: Display the AUTOSTART soft key
- Numerical range for M functions from 0 to 999
- Assignment of NC macros to M functions through a table
- Description of tilting-axis geometry in tables
- M4586: Autostart
- NCMACRO.SYS: Executing an NC macro in case of an error or after an external/internal stop
- NCMACRO.SYS: Executing a macro after leaving Pass Over Reference Point mode
- NCMACRO.SYS: Executing a macro when a block scan has been interrupted
- After reestablishing machine status, the status set by the PLC is checked to see if it agrees with the status calculated by the NC.
- PGM CALL and CYCLE CALL (cycles greater than 68) are calculated automatically with the look-ahead function and run without exact stop (FN20: WAIT FOR SYNC).
- The integrated oscilloscope can record the jerk and the acceleration of the axes.
- W1018: Number of files opened by the PLC
- W1020: Number of open files
- M4546: Second tool age has expired

- FN17 ID290: Activate tilting-axis geometry description
- FN18 ID290: Ascertain tilting-axis geometry description
- FN17 ID210: Tilted working plane in Manual mode
- FN18 ID210: Is tilted working plane in Manual mode active?
- FN17 ID230: Turn limit switch monitoring on and off
- FN17 ID501: Write to REF-referenced datum table
- FN18 ID501: Read from REF-referenced datum table
- FN17 ID20: Switchover spindle 1 / spindle 2
- FN31: Switch range of traverse, axis assignment, axis display
- FN32: Setting the datum
- Module 9044: Reading the spindle coordinates in 0.0001°
- Module 9097: Selecting the geometry description
- Module 9161, 9162: The bit for the spindle or the axis can be transferred with C-axis operation
- Module 9189: Shut down the control
- Module 9291: Calling an NC macro
- Strings can be shown in the TABLE function
- Before the PLC program is converted, the PLC outputs are reset.
- Various LSV2 telegrams are entered in the log book.
- Two M functions in one NC block
- New NC blocks: SEL TABLE, SEL PATTERN and CYCL CALL PAT
- Cycle 208: Parameter Q342
- Cycle 247: Coordinates always interpreted in reference coordinates
- Cycles 262, 263, 264, 265, 267: Thread milling
- Cycle 209: Tapping with chip breaking
- Cycles 410 to 418: Also function when a basic rotation is active.
- Sample tables are copied into the appropriate directories with COPY SAMPLE FILES.
- Multiple empty coordinate fields in NC programs are skipped.
- If a company logo is displayed, the control description is no longer shown.

NC software 280 476-04 (export version 280 476-04)

Release: 09/99

NC software 280 476-05 (export version 280 477-05)

Release: 10/99

Improvements:

- MP2602, MP2604: Integral Phase Compensation - IPC
- Temperature compensation with M128 etc. through the description tables of the tilting-axis geometry
- The selected geometry description is indicated in the program management.
- Input range of the STR column in the motor tables expanded
- A DSP error message appears if the ready signal is missing from the speed encoder inputs for vertical axes.
- Direction monitoring of digitally controlled spindles
- M4179: Control is shut down
- M4230: NC start via LSV2
- M4231: NC stop via LSV2
- M4220: Error from PET table/F stop active
- M4221: Error from PET table/NC stop active
- M4222: Error from PET table/EM. STOP active
- Module 9002, 9003, 9005, 9008, 9009: Error code 24 (Module was called in the submit job or spawn job) added
- Module 9007: Diagnostic information of the PL
- Module 9040, 9041: Coordinate type 8 (temperature compensation from description tables) added
- Module 9087: Status of PLC error message
- Module 9098: Ascertain the active geometry description
- Module 9130: New error codes 1 (Invalid analog output) and 2 (Disabled analog output) added
- Module 9158: The torque of the active spindle can also be limited
- Module 9189: The drive ready signal is removed
- Module 9196: Finding the PLC cycle time
- Improved search function
- ISO: A Q parameter can be entered for the feed rate after M128 or G80
- ISO: The Q key of the numerical keypad or the ASCII keyboard may be used
- When the control is shut down, the hard disk is put into sleep mode.
- Cycle 440: Heat compensation

NC software 280 476-06 (export version 280 477-06)

Release: 12/99

Improvements:

- MP420, MP430: Changes no longer lead to a control reset
- MP2221: Monitoring of rotational direction for spindle with integral DSP
- MP2600: Input range changed
- MP6510: Second measuring error added
- MP7640, MP7641, MP7645 are no longer shown through code number 123
- All machine parameters that can be changed by the PLC can be changed in a running NC program.
- Overflows from multiturn encoders are entered in the NCDATA.SYS system file.
- Before the PLC program is converted , the memories of the PLC outputs are reset.
- Pointers to error messages are entered in the log book.
- M4547: T and G strobes with TOOL CALL
- FN17 ID990: Orient spindle
- FN18 ID990: Find spindle angle
- Hard disks larger than 3.25 MB are supported
- Module 9035: Transfer value 26 (jog increment limiting) added
- Module 9060: Status of M functions
- Module 9186: Transfer value 6 (jog increment) added
- Module 9300: Locking/releasing the pocket table
- Module 9305: Pocket exchange in the pocket table
- Module 9310: Read the machine parameter from the run-time memory
- Module 9320: Status of NC program end
- M140: Depart contour in the current tool-axis direction
- M141: Suppress touch probe monitoring
- New LSV2 telegrams
- Cycle 204: A value with digits after the decimal point may be entered in Q255
- A German and an English machine parameter description, as well as the configuration files for CycleDesign, are saved in PLC:\JH.
- Every NC setup is saved in its own directory and can be selected.
- M functions M0 to M999 are possible in Manual mode; M100 to M299 result in error messages

NC software 280 476-07 (export version 280 477-07)

Release: 02/00

Improvements:

- MP812: Software limit switches also with M94, modulo display and encoders with EnDat interface
- MP2606: Influence following error during the jerk phase
- MP3030: Zero spindle speed when switching to another gear range
- MP6500: Tool measurement with stationary spindle
- MP7261: Pockets in tool magazines 1 through 4
- MP7267: Position of the comment from the tool table in the pocket table
- MP7620: Feed-rate smoothing
- MP7683: Display of pallet table and NC program
- Up to 4 tool magazines may be managed in the pocket table.
- Encoders with EnDat interface can be used as position encoders for rotary axes.
- Monitoring of the number of grating periods between the reference end position and the first reference mark
- In a tilted working plane in Manual mode, more than one axis can be moved simultaneously, and the start key can be used to traverse the reference marks.
- In the WAIT column of the table of M-function macros, you can determine if the NC should wait for acknowledgement from the PLC.
- With the aid of the "machine parameter subfile" column in the assignment table of the tilting-axis geometry description, a function for changing milling heads can be created.
- 16 strings
- After the drive is switched on, the control waits 2 s for the "ready" signal.
- Module 9097: Error codes 9 (Error in the MPFILE column), 10 (Error in the MP7500 column) and 11 (Error in the machine parameter subfile) added
- Module 9031: Find the number of an entry in the pocket table
- Module 9032: Look for a free pocket in the tool magazine
- Module 9036: Switch tools between tool magazines
- Cylindrical surface interpolation in a tilted working plane
- SUPPORT soft key for remote diagnosis with TeleService
- A comment can be entered in the DOC column of the datum tables.
- M142: Delete modal program information
- M143: Delete programmed basic rotation
- Display "TNC 426/430 x" when control is switched on
- M109 or M110 are automatically cancelled when an OEM cycle ends or is cancelled.

NC software 280 476-07 (export version 280 477-07)

Release: 02/00

Improvements:

- MP812: Software limit switches also with M94, modulo display and encoders with EnDat interface

NC software 280 476-08 (export version 280 477-08)

Release: 05/00

NC software 280 476-09 (export version 280 477-09)

Release: 05/00

Improvements:

- MP10: Change no longer results in a reset
- MP1095: Nominal position value filter for manual and automatic operating modes
- MP1396: Feedback control with velocity semifeedforward
- MP1516: k_V factor for velocity feedforward
- MP2170: Waiting time between switching on the drive and the drive's standby signal
- MP2391: Max. braking performance of spindle during a powerfail
- MP4020: Pass on simulated and disabled keys to NC and PLC windows
- MP6572: Limit spindle speed in tool measurement cycles
- MP7230: Conversational language Russian
- MP7442: Number of the M function for the spindle orientation in cycles
- MP7502: Functionality of M144/M145
- A maximum of 10 variables can be used in the description tables for the tilting-axis geometry.
- Numbers in hexadecimal and binary form in freely definable tables
- New error messages for FN14
- M4587: Feed rate limit not effective
- During powerfail: Attempt to reset PLC outputs and to maintain readiness for 3 seconds
- FN17 ID600: Factor for velocity semifeedforward
- FN17 ID600: Factor from MP1396 for velocity semifeedforward
- FN18 ID20: Coordination between logical and geometrical axis
- FN18 ID310: M144 active
- Module 9035: Status information of active range of traverse
- Module 9061: Status of non-modal M functions
- Module 9066: Status of HEIDENHAIN inverter
- Module 9133: Temperature of the LE
- Module 9279: Control reset
- M144: When an axis is moved, the kinematics are recorded in the display, but without compensatory motions.
- Increased protection of the controls from unauthorized data accessing
- If the hardware contains flash EPROMs, the letter "F" is displayed before the NC software Id. Nr.
- When transferring an actual position capture to a datum table, it can be referenced to either the workpiece datum or the machine datum.
- When transferring measured values from the manual touch probe cycles to a datum table, they can be referenced to either the workpiece datum or the machine datum.

NC Software 280 476-10 (export version 280 477-10)

Release: 06/00

- MP6165: New probing function for the TS 632
- New password AXISNUMBER for OEM.SYS, to limit the number of machine parameter indexes

NC software 280 476-11 (export version 280 477-11)

Release: 07/00

- MP1087.x: Axis-specific jerk for Manual mode
- MP7680 bit 12: Behavior of Cycle 28
- New error message for FN14
- FN17 ID990: Axis shown on top of other axis due to tilting motion?

NC software 280 476-12 (export version 280 477-12)

Release: 07/00

NC software 280 476-13 (export version 280 477-13)

Release: 10/00

- MP110.x and MP111.x no longer result in a RESET
- Maximum input ranges for MP2391.x, MP2393.x, MP2395.x, MP2500.x, MP2501.x, MP2510.x and MP2511.x, MP6160, MP6161, MP6560, MP7442 expanded
- MP7500 bit 8: Recalculate the datum to the home position of the tilting element
- EnDat multiturn encoders as position encoders
- Actual-to-nominal value transfer only occurs in the affected axis during an erasable positioning error
- New function: Field orientation for synchronous spindles with encoders without a Z1 track, or nonaligned encoders with EnDat interface
- New motor table motor.mot
- FN18 ID290: Determine the value of the bits in MP7500
- Module 9146: Saving and reestablishing actual position values
- Module 9147: Bit from W1032 is reset
- Module 9157: Drive controller status
- Behavior during calibration of the touch-trigger probe with infrared transmission changed
- Tool-by-tool execution of pallet tables
- Cycle 431: Establish spatial angle
- Cycle 247: Set a datum from a datum table as the datum in the REF system

NC software 280 476-14 (export version 280 477-14)

Release: 11/00

- FN18 ID214: Ascertain tolerance in Cycle 32 or MP1096

NC software 280 476-15 (export version 280 477-15)

Release: 11/00

- MP7441 bit 1: Reserved
- MP7441 bit 2: Error message when a positive depth is programmed in machining cycles
- MP7682 bit 3 and bit 4: Reserved
- Was M4051 set before starting the probe block?

NC software 280 476-16 (export version 280 477-16)

Release: 12/00

- MP2221: Monitor $\overline{\text{ERR-IZ}}$ signal
- MP7430: Input range changed
- FN18 ID210 NR8: Angle of misalignment between the spindle and the tilted coordinate system
- FN18 ID360 NR3: Measurement results of touch probe cycles 0 and 1 without probe radius and length compensation

NC software 280 476-17 (export version 280 477-17)

Release: 03/01

- MP2397: Can be changed by the PLC
- MP7683 bit 7: Execute AUTOSTART function by PLC
- M4182: AUTOSTART function activated
- M4183: Time for AUTOSTART function expired
- `STRICTREPOS=` in OEM.SYS can be used to activate the function for restoring the position when an NC program is interrupted during Single Block mode or by a STOP block and the positions of the NC axes are changed.
- Check the available memory on the hard disk when the control is started
- Check the cycle data when the control is started and enter information into the log file
- Tool-oriented machining of pallet tables

NC software 280 476-18 (export version 280 477-18)

Release: 04/01

- MP1060.x and MP1070.x: Input range expanded
- In volts-per-hertz control mode, W322 = 0
- Improved behavior when using a speed encoder with EnDat interface

NC software 280 476-19 (export version 280 477-19)

Release: 07/01

- MP2180 has been expanded into MP2180.0 to MP2180.8.
- MP2600.x: Input range expanded
- MP7160 bit 3: IPC and acceleration feedforward for Cycle 17 and 18
- New error message **`Actual position saved <axis>`** when a calling of Module 9146 is not permitted
- In the **`MP7530`** column and in the **`TEMPCOMP`** column of the description tables for the swivel axis geometry, a maximum of 16 variables can be used.
- Module 9120 and 9123: Error code 6 (feed rate not permitted) added
- New code word **`REMOTE.PLCPASSWORDFORCED =`** for OEM.SYS, in order to permit machine backup, full backup and setup only with the code word defined with **`PLCPASSWORD =`**.

NC software 280 476-20 (export version 280 477-20)

Release: 08/01

NC software 280 476-21 (export version 280 477-21)

Release: 11/01

- MP1152: I3 (control-is-ready signal acknowledgement) is passed on directly to the NC, or is first processed by the PLC
- The machining plane position indication (MP7500 bit 1 = 1) has been expanded by the following swivel-axis combination:
  Swivel head and rotary table: axis sequence B variable, A variable (tool axis Z)
- FAILTEST code number for testing an internal EMERGENCY STOP
- The power module table was expanded by the following columns: I-N-DC, T-DC, F-DC, T-AC, F-AC, T-IGBT, I-N-AC-3333, I-N-AC-4000, I-N-AC-5000, I-N-AC-6666, I-N-AC-8000 and I-N-AC-10000. The columns currently have no function.
- The motor table was expanded by the following columns T-DC, F-DC, T-AC and F-AC. The columns currently have no function.
- In the **`Machine-parameter programming`** mode, the DELETE ALL and SELECT soft keys after the DEL/SEL SETUP soft key were switched.

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
- Handwheel

3.1.2 Electromagnetic Compatibility

Intended area of application

This device complies with EN 61800-3, and fulfills the requirements of an industrial low-voltage main power line. It is not designed to be used on a public low-voltage main power line from which households are supplied. This device can cause high-frequency disturbances.

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

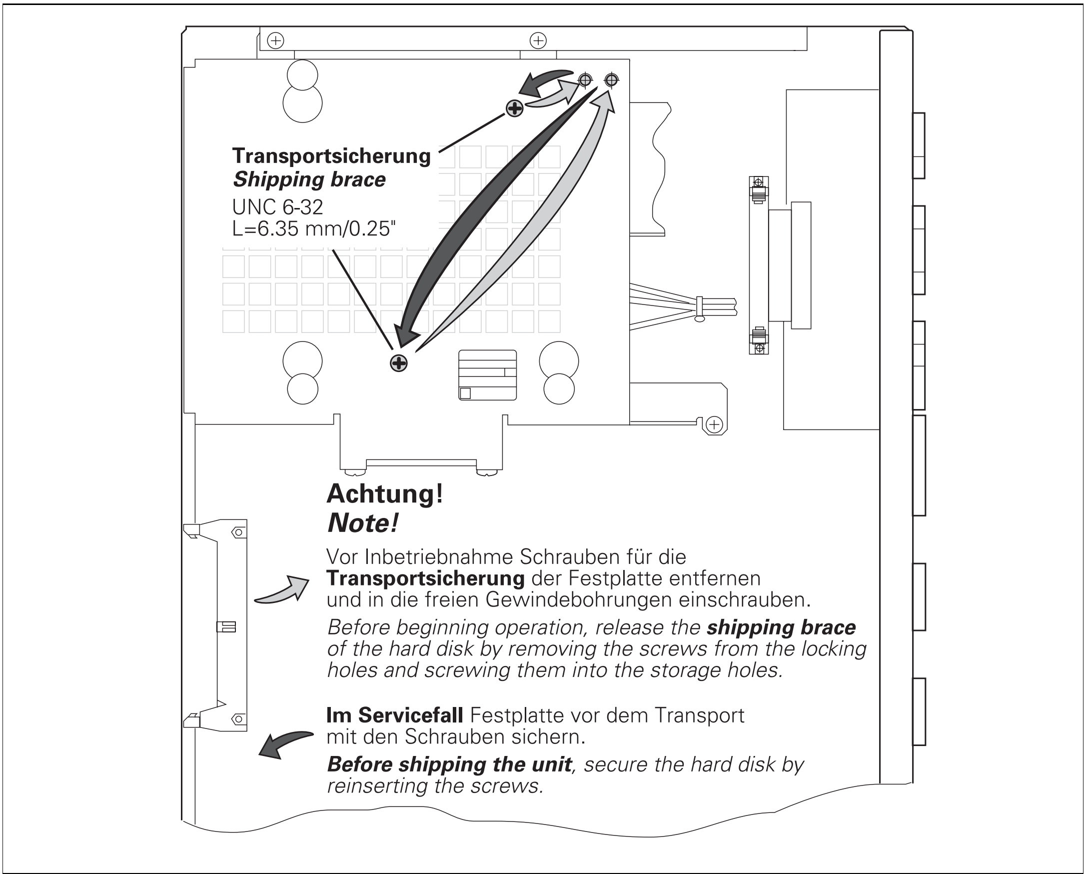
- A minimum distance of 20 cm from the logic unit and its leads to interfering equipment
- A minimum distance of 10 cm from the logic unit and its leads to cables that carry interference signals. For cables in metallic ducting, adequate decoupling can be achieved by using a grounded separation shield.
- Shielding according to EN 50 178
- Potential compensating lines Ø 6 mm^2 (see Grounding Diagram at end of chapter)
- Use only genuine HEIDENHAIN cables, connectors and couplings

3.1.3 Shipping Brace for Hard Disk (LE 426 M, LE 430 M)

The hard disks of the LE 426 M and the LE 430 M 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 LE. 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 released 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 LE is being shipped on its own), the hard disk must be secured with the shipping brace.



3.2 Ambient conditions

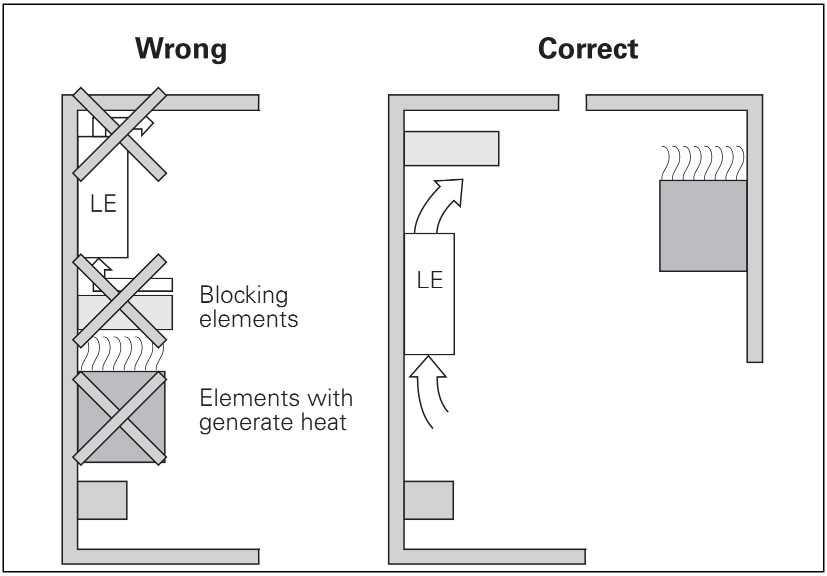
3.2.1 Heat Generation and Cooling

The permissible ambient temperature in operation is between 0 °C and 45 °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 must extract the warm air from the logic unit. 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 figure 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.



3.2.2 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 TNC not be switched off, so as to avoid condensation on the circuit boards.

3.2.3 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

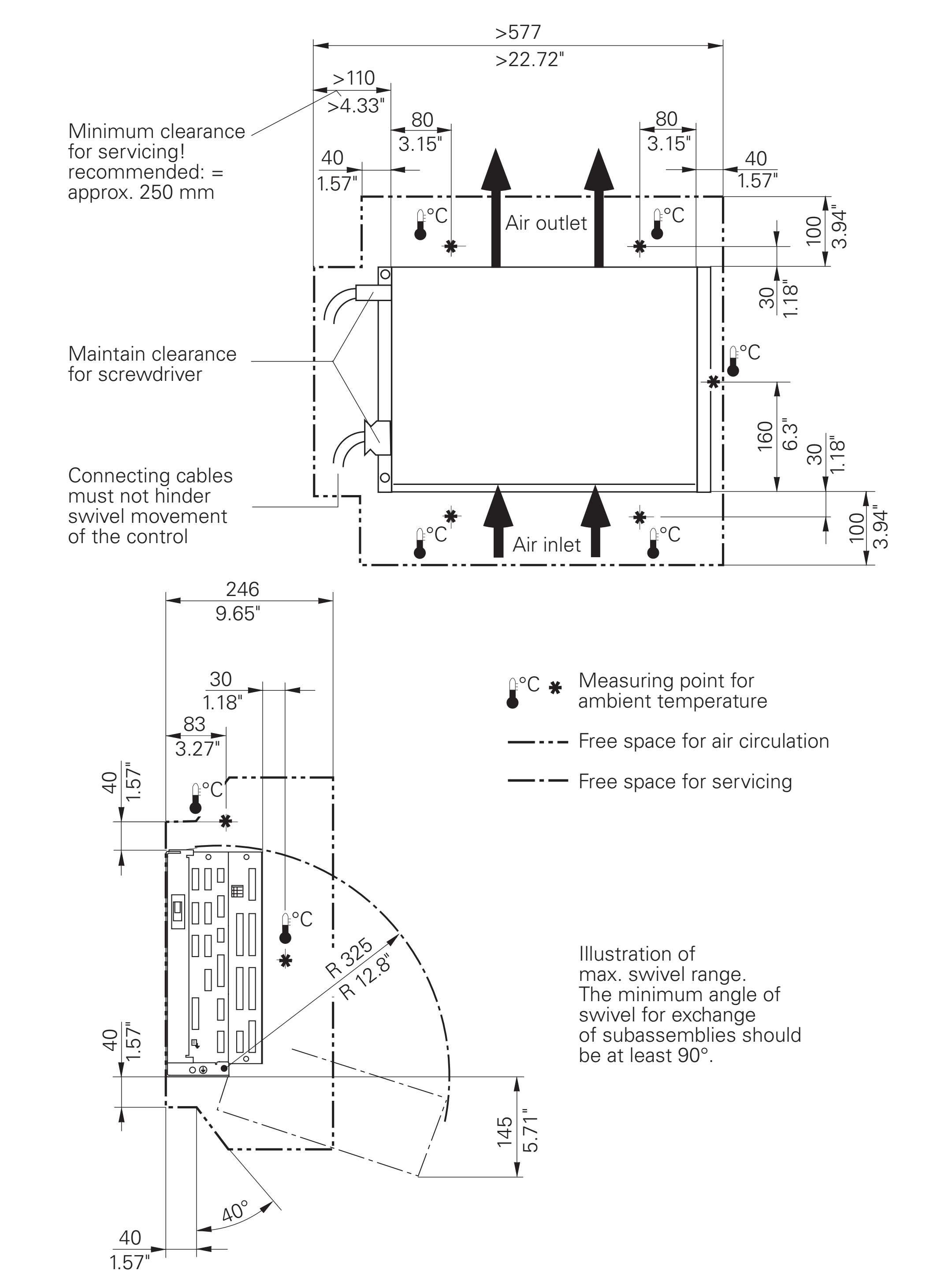
Permissible shock when hard disk shipping brace is used (only LE 426 M, LE 430 M): 300 m/s^2, 11 ms

3.2.4 Mounting Position

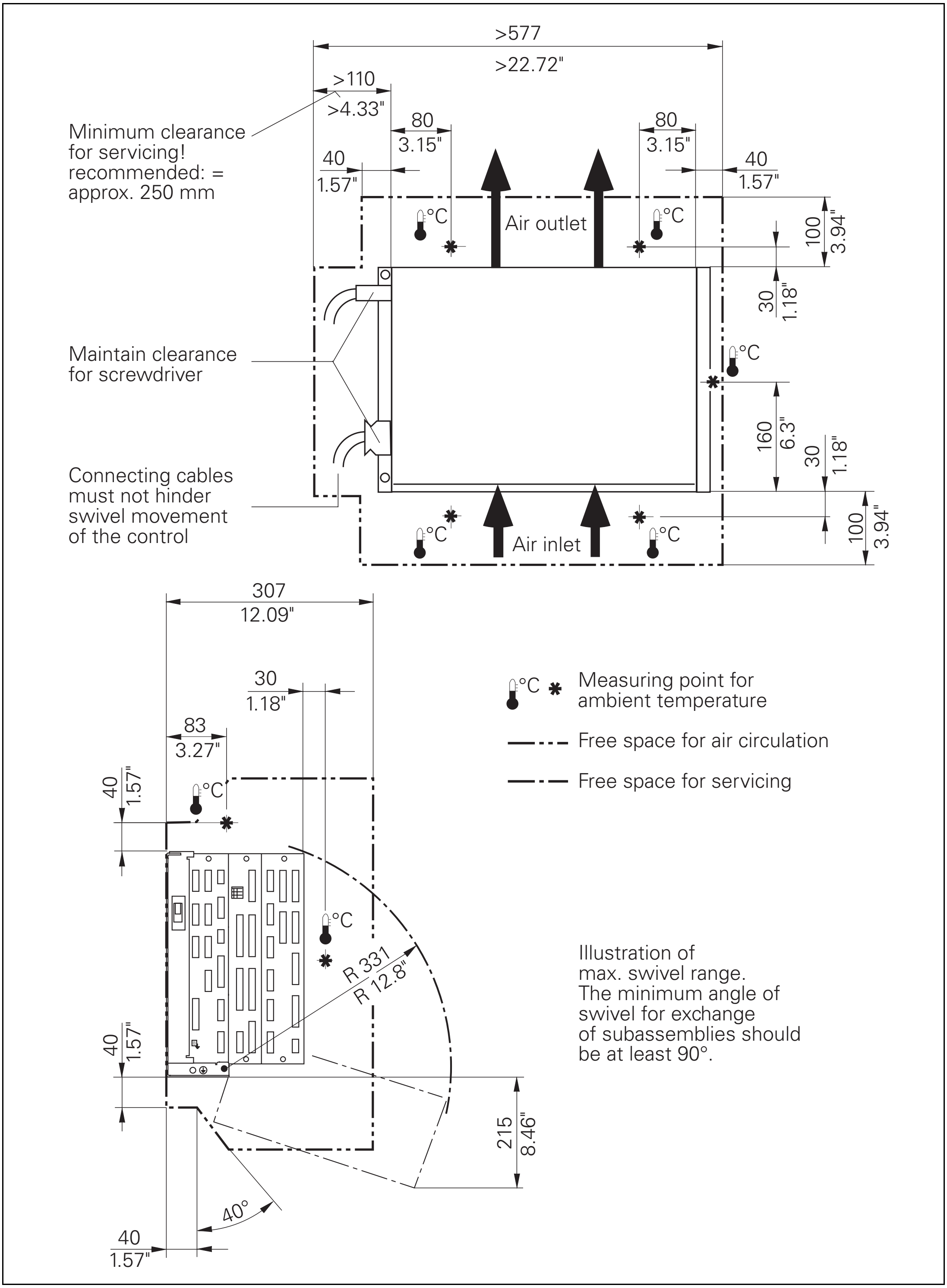
Warning

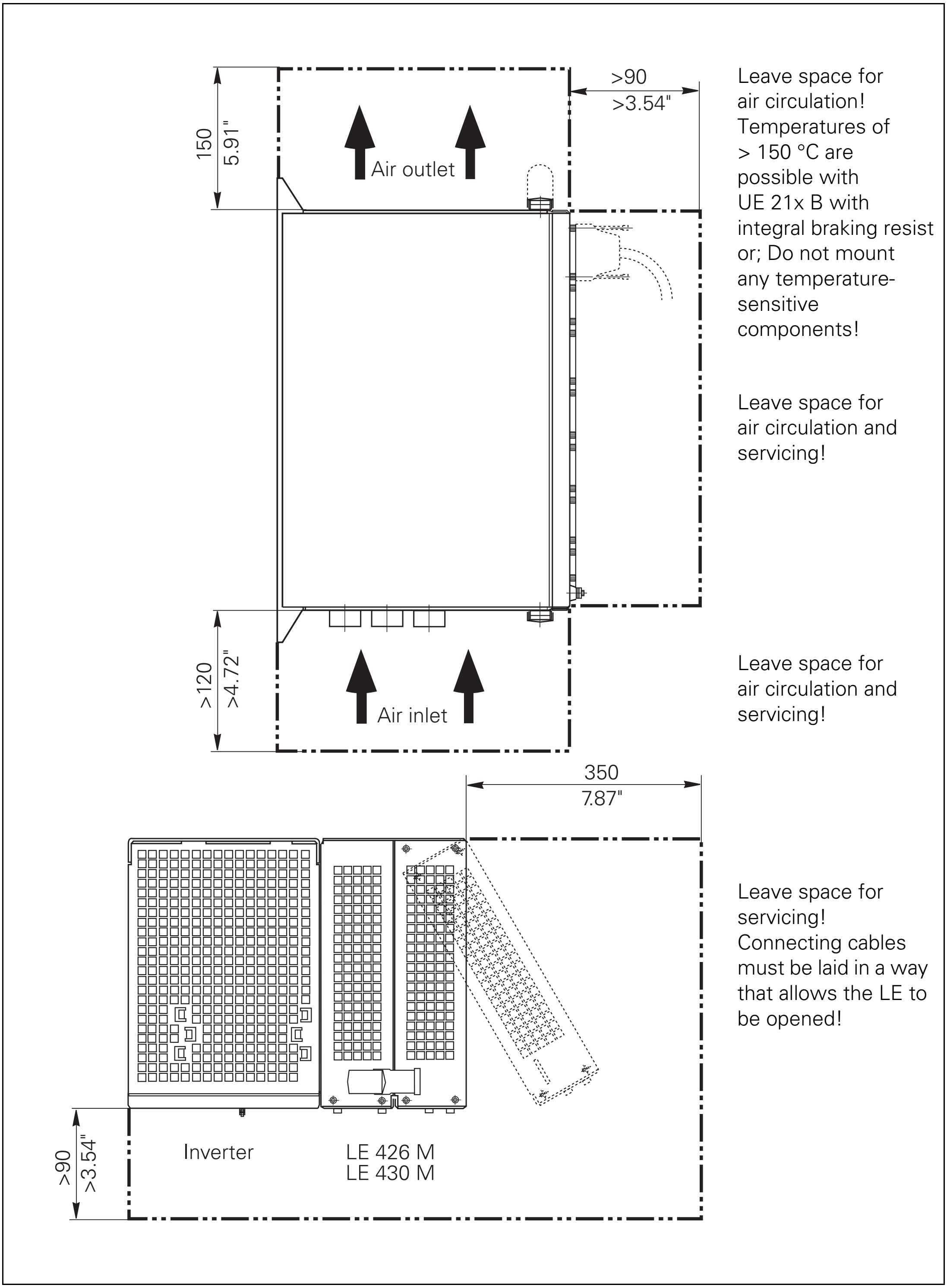
When mounting, please observe proper minimum clearance, space requirements, and length of connecting cable.

Logic unit
LE 426 CB,
LE 430 CA



**Logic unit
LE 426 PB,
LE 430 PA**




>90
>3.54"
150
5.91"
Air outlet
Leave space for air circulation! Temperatures of > 150 °C are possible with UE 21x B with integral braking resistor; Do not mount any temperature-sensitive components!
Leave space for air circulation and servicing!
>120
>4.72"
Air inlet
Leave space for air circulation and servicing!
350
7.87"
Leave space for servicing! Connecting cables must be laid in a way that allows the LE to be opened!
>90
>3.54"
Inverter
LE 426 M
LE 430 M

BC 120 visual display unit

Strong electromagnetic or magnetic fields can lead to a slight distortion of the picture on the BC 120.

To prevent distortion:

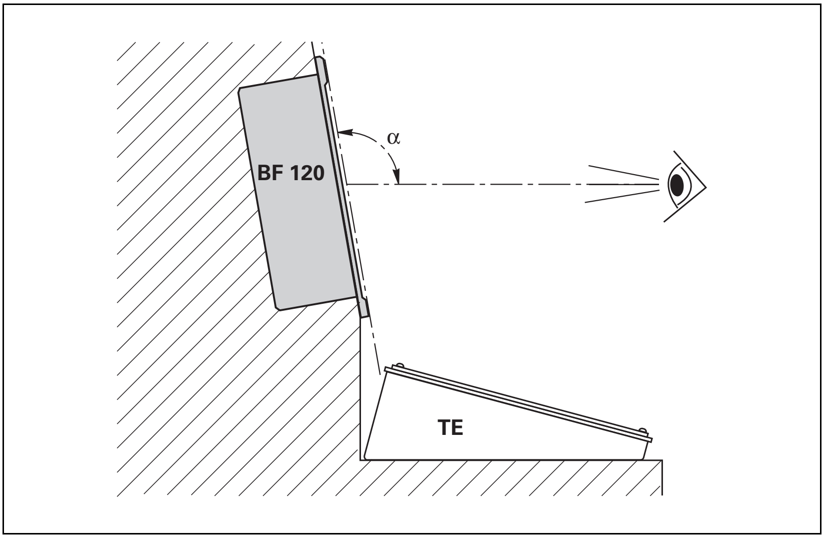
▶ Maintain a minimum clearance of 0.5 m between the screen housing and sources of interference. Possible sources of interference include permanent magnets, motors and transformers.

Space requirements for air circulation are shown in the dimension drawing at the end of this chapter.

BF 120 visual display unit

The BF 120 flat-panel display must be viewed with a slight backward slant.

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



PLC input/ output unit

A maximum of four PL 410B or one PL 405B can be connected to the TNC.

TNC 426 CB, TNC 430 CA:

- You can mount one PL on the logic unit.
- Additional PLs must be mounted separately in the electrical cabinet.

TNC 426 PB/M, TNC 430 PA/M:

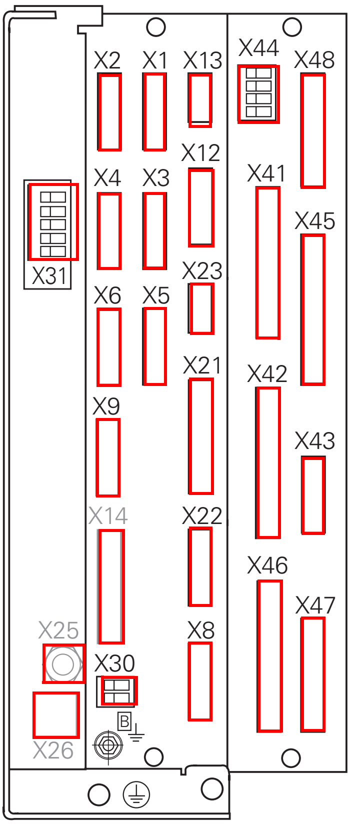
- The PLs must be mounted separately in the electrical cabinet.

3.3 Connection Overview

3.3.1 LE 426 CB

Warning

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

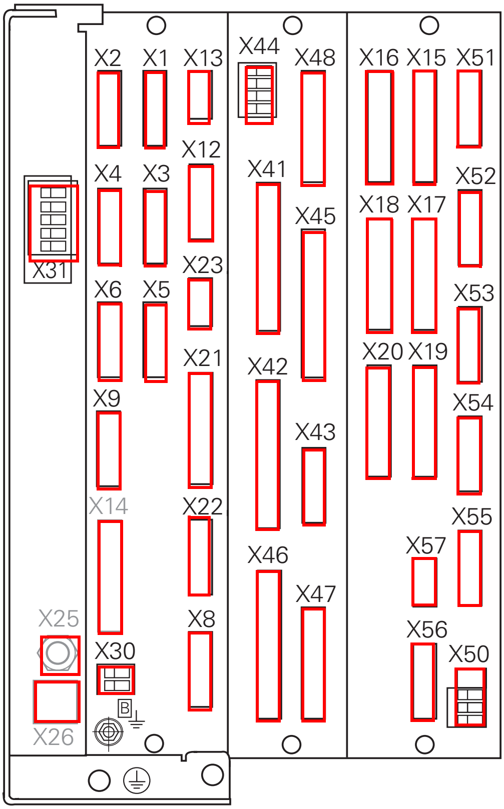


X1 to X6	Position encoder
X8	Analog output 1 to 6
X9	Analog output 7 to 13
X12	Triggering touch probe for workpiece measurement
X13	Triggering touch probe for tool measurement
X14	Measuring touch probe (option)
X21	RS-232-C/V.24 data interface
X22	RS-422/V.11 data interface
X23	Handwheel input
X25	Ethernet interface (option)
X26	Ethernet interface (option)
X30	Reference signal for spindle
X31	NC power supply
X41	PLC output
X42	PLC input
X43	CRT screen or
X49	Color flat-panel display
X44	PLC supply voltage
X45	TNC keyboard unit
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
B	Signal ground
⏚	Equipment ground (YL/GN)

3.3.2 LE 426 PB (Spindle up to 12 000 rpm)

Warning

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



X1 to
X6 Position encoder

X8 Analog output 1 to 6

X9 Analog output 7 to 13

X12 Triggering touch probe for workpiece measurement

X13 Triggering touch probe for tool measurement

X14 Measuring touch probe (option)

X15 to
X20 Encoder for speed

X21 RS-232-C/V.24 data interface

X22 RS-422/V.11 data interface

X23 Handwheel input

X25 Ethernet interface (option)

X26 Ethernet interface (option)

X30 Reference signal for spindle

X31 NC power supply

X41 PLC output

X42 PLC input

X43 CRT screen or

X49 Color flat-panel display

X44 PLC supply voltage

X45 TNC keyboard unit

X46 Machine operating panel

X47 PLC expansion

X48 PLC analog input

X50 Input for drive motor enabling

X51 to
X56 Output to axes motor power module

X57 Reserved

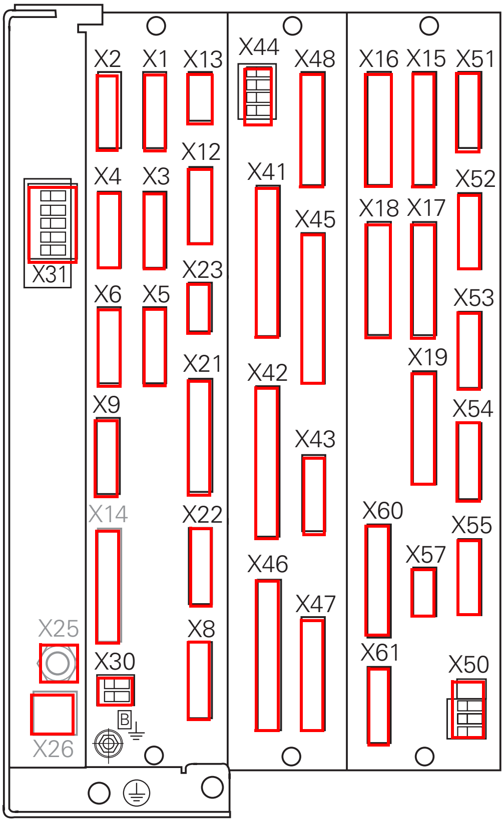
B Signal ground

Equipment ground (YL/GN)

3.3.3 LE 426 PB (Spindle up to 30 000 rpm)

Warning

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

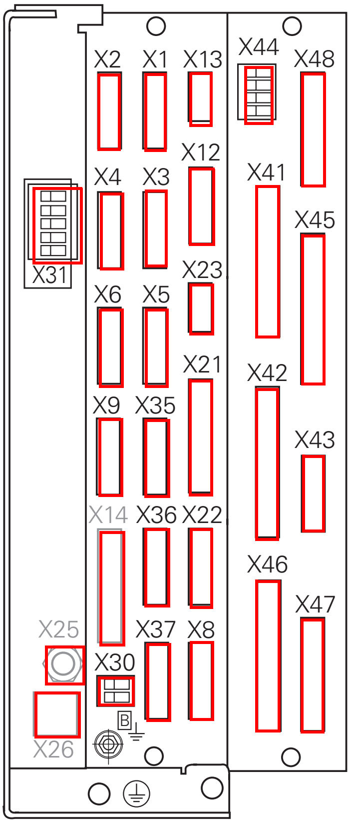


X1 to X6	Position encoder
X8	Analog output 1 to 6
X9	Analog output 7 to 13
X12	Triggering touch probe for workpiece measurement
X13	Triggering touch probe for tool measurement
X14	Measuring touch probe (option)
X15 to X19	Encoder for speed
X21	RS-232-C/V.24 data interface
X22	RS-422/V.11 data interface
X23	Handwheel input
X25	Ethernet interface (option)
X26	Ethernet interface (option)
X30	Reference signal for spindle
X31	NC power supply
X41	PLC output
X42	PLC input
X43	CRT screen or
X49	Color flat-panel display
X44	PLC supply voltage
X45	TNC keyboard unit
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
X50	Input for drive motor enabling
X51 to X55	Output to axes motor power module
X57	Reserved
X60	Encoder for spindle speed
X61	Output to motor power module of the spindle
B	Signal ground
⏚	Equipment ground (YL/GN)

3.3.4 LE 430 CA

Warning

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

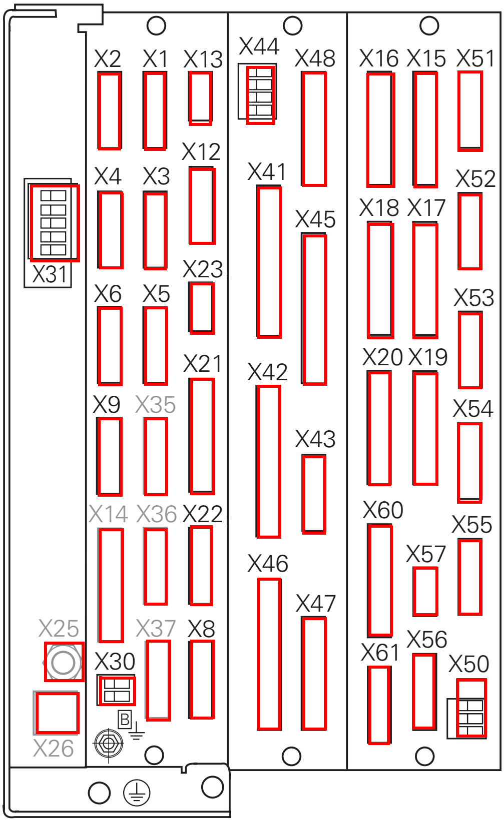


X1 to X6	Position encoder
X8	Analog output 1 to 6
X9	Analog output 7 to 13
X12	Triggering touch probe for workpiece measurement
X13	Triggering touch probe for tool measurement
X14	Measuring touch probe (option) or
X38	Additional position encoder input (option)
X21	RS-232-C/V.24 data interface
X22	RS-422/V.11 data interface
X23	Handwheel input
X25	Ethernet interface (option)
X26	Ethernet interface (option)
X30	Reference signal for spindle
X31	NC power supply
X35 to X37	Position encoder
X41	PLC output
X42	PLC input
X43	CRT screen or
X49	Color flat-panel display
X44	PLC supply voltage
X45	TNC keyboard unit
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
B	Signal ground
⏚	Equipment ground (YL/GN)

3.3.5 LE 430 PA

Warning

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



X1 to
X6 Position encoder

X8 Analog output 1 to 6

X9 Analog output 7 to 13

X12 Triggering touch probe for workpiece measurement

X13 Triggering touch probe for tool measurement

X14 Measuring touch probe (option) or

X38 Additional position encoder input (option)

X15 to
X20 Encoder for axes speed

X21 RS-232-C/V.24 data interface

X22 RS-422/V.11 data interface

X23 Handwheel input

X25 Ethernet interface (option)

X26 Ethernet interface (option)

X30 Reference signal for spindle

X31 NC power supply

X35 to Position encoder for 3 axes with analog
X37 speed command interface (option)

X41 PLC output

X42 PLC input

X43 CRT screen or

X49 Color flat-panel display

X44 PLC supply voltage

X45 TNC keyboard unit

X46 Machine operating panel

X47 PLC expansion

X48 PLC analog input

X50 Input for drive motor enabling

X51 to
X56 Output to motor power module

X57 Reserved

X60 Encoder for spindle speed

X61 Output to motor power module of the spindle

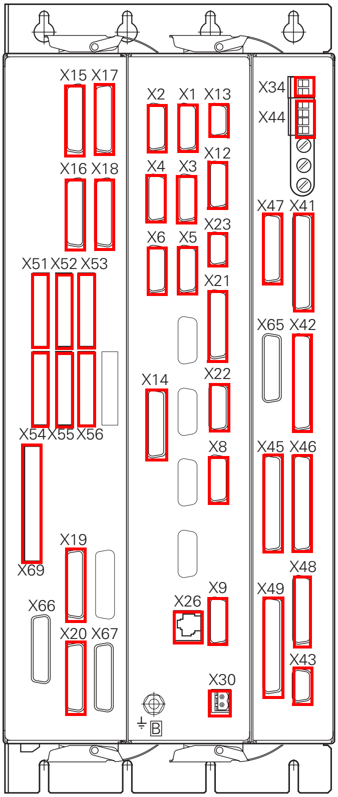
B Signal ground

⏚ Equipment ground (YL/GN)

3.3.6 LE 426 M (Spindle up to 12 000 rpm)

Warning

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



X1 to
X6 Position encoder

X8 Analog output 1 to 6

X9 Analog output 7 to 13

X12 Triggering touch probe for workpiece measurement

X13 Triggering touch probe for tool measurement

X14 Measuring touch probe (option)

X15 to
X20 Encoder for speed

X21 RS-232-C/V.24 data interface

X22 RS-422/V.11 data interface

X23 Handwheel input

X26 Ethernet interface (option)

X30 Reference signal for spindle

X34 Input 24 V for "Control-is ready"

X41 PLC output

X42 PLC input

X43 CRT screen or

X49 Color flat-panel display

X44 PLC supply voltage

X45 TNC keyboard unit

X46 Machine operating panel

X47 PLC expansion

X48 PLC analog input

X51 to
X56 Output to motor power module

X65 to
X67 Reserved

X69 Power supply

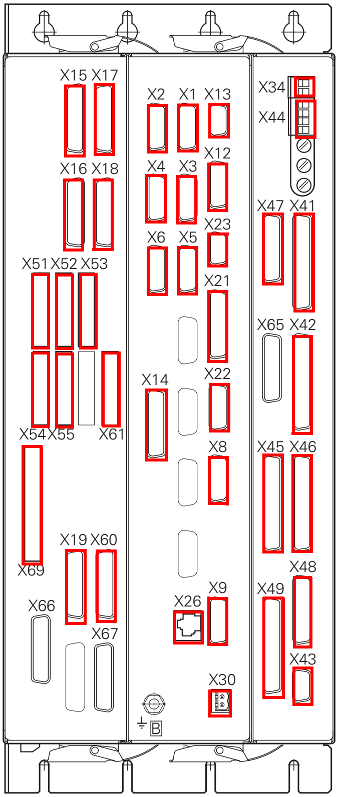
B Signal ground

⏚ Equipment ground (YL/GN)

3.3.7 LE 426 M (Spindle up to 30 000 rpm)

Warning

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

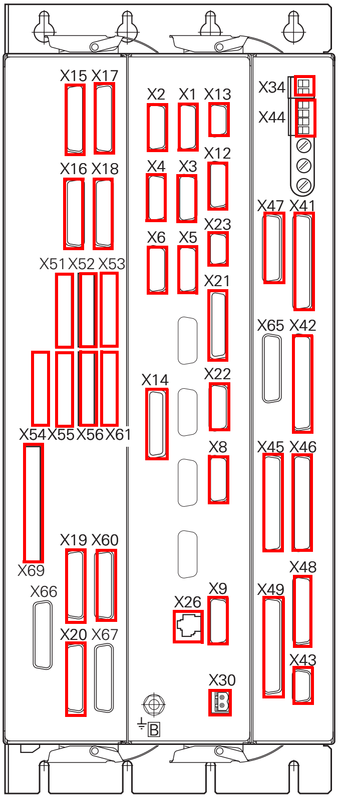


X1 to X6	Position encoder
X8	Analog output 1 to 6
X9	Analog output 7 to 13
X12	Triggering touch probe for workpiece measurement
X13	Triggering touch probe for tool measurement
X14	Measuring touch probe (option)
X15 to X19	Encoder for axes speed
X21	RS-232-C/V.24 data interface
X22	RS-422/V.11 data interface
X23	Handwheel input
X26	Ethernet interface (option)
X30	Reference signal for spindle
X34	Input 24 V for "Control-is ready"
X41	PLC output
X42	PLC input
X43	CRT screen or
X49	Color flat-panel display
X44	PLC supply voltage
X45	TNC keyboard unit
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
X51 to X55	Output to axes motor power module
X60	Encoder for spindle speed
X61	Output to spindle motor power module
X65 to X67	Reserved
X69	Power supply
B	Signal ground
⏚	Equipment ground (YL/GN)

3.3.8 LE 430 M/6 Axes

Warning

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

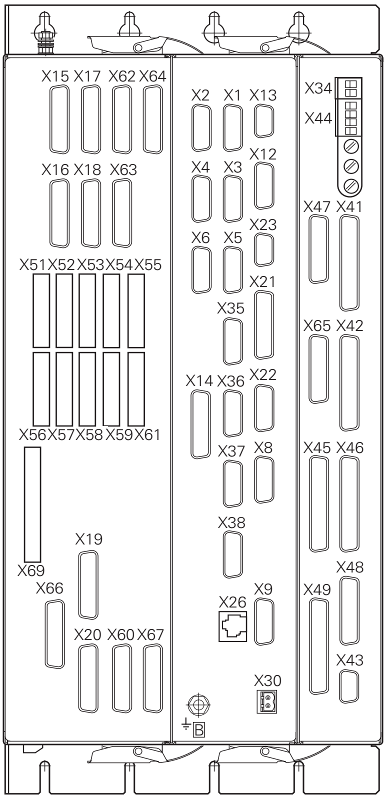


X1 to X6	Position encoder
X8	Analog output 1 to 6
X9	Analog output 7 to 13
X12	Triggering touch probe for workpiece measurement
X13	Triggering touch probe for tool measurement
X14	Measuring touch probe (option)
X15 to X20	Encoder for axes speed
X21	RS-232-C/V.24 data interface
X22	RS-422/V.11 data interface
X23	Handwheel input
X26	Ethernet interface (option)
X30	Reference signal for spindle
X34	Input 24 V for output "Control is ready"
X41	PLC output
X42	PLC input
X43	CRT screen or
X49	Color flat-panel display
X44	PLC supply voltage
X45	TNC keyboard unit
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
X51 to X56	Output to motor power module
X60	Encoder for spindle speed
X61	Output to spindle motor power module
X65 to X67	Reserved
X69	Power supply
B	Signal ground
⏚	Equipment ground (YL/GN)

3.3.9 LE 430 M/9 Axes

Warning

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



X1 to X6	Position encoder
X8	Analog output 1 to 6
X9	Analog output 7 to 13
X12	Triggering touch probe for workpiece measurement
X13	Triggering touch probe for tool measurement
X14	Measuring touch probe (option)
X15 to X20	Encoder for axes speed
X21	RS-232-C/V.24 data interface
X22	RS-422/V.11 data interface
X23	Handwheel input
X26	Ethernet interface (option)
X30	Reference signal for spindle
X34	Input 24 V for “Control-is ready”
X35 to X38	Position encoder
X41	PLC output
X42	PLC input
X43	CRT screen or
X49	Color flat-panel display
X44	PLC supply voltage
X45	TNC keyboard unit
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
X51 to X59	Output to motor power module
X60	Encoder for spindle speed
X61	Output to spindle motor power module
X62 to X64	Encoder for axes speed
X65 to X67	Reserved
X69	Power supply
B	Signal ground
⏚	Equipment ground (YL/GN)

3.4 Power Supply

3.4.1 NC Power Supply for LE 426 CB/PB, LE 430 CA/PA

Danger

The dc-link power supply must be opened only by HEIDENHAIN service personnel!

X31: NC supply voltage

Pin layout:

Cnnctn. terminals	Assignment	LE 426 PB, LE 430 PA	LE 426 CB, LE 430 CA
⏚	Equipment ground (YL/GY)		
U_1	Phase 1	400 Vac ± 10 % 50 to 60 Hz via isolating transformer	
U_2	Phase 2		
$–U_Z$	DC-link voltage –	385 Vdc to 660 Vdc	–
$+U_Z$	DC-link voltage +	Other voltage ranges on request	

Power consumption of the logic unit is approx. 55 W.

LE 426 CB, LE 430 CA

Ensure compliance with the European standard EN 55022 for "electromagnetic interference"

- Connect the LE to the ac line power through an isolating transformer or a line filter. If the **inverter** is connected to the ac line power through a line filter, the LE can be connected to this power supply.

You can switch off the power supply with Module 9167. (See "Overview of Modules" on page 5 – 3).

LE 426 PB, LE 430 PA

Power must be supplied via an isolating transformer (200 VA, basic insulation in accordance with EN 50178 or VDE 0550).

Ensure a reliable supply voltage for the drive control

- Connect the dc-link voltage of the servo amplifier to the terminals $+U_Z$ and $–U_Z$ (385 Vdc to 660 Vdc).

The LE monitors the rectified voltage:

- An overvoltage up to 720 V is permissible for 5 seconds.
- Over 720 V, the NC prevents a pulse release for the IGBT of the power module: the motors coast to a non-controlled stop and there can be no energy recovery to the dc link.
- Below 385 Vdc (powerfail) all drives are brought to a controlled stop; the control must be switched off and on again.
- Below 155 Vdc, the control is reset (RESET).
- At 135 Vdc, the dc-link power supply switches off.

3.4.2 NC Power Supply for LE 426 M, LE 430 M

X69: NC supply voltage and control signals

The LE 426 M, LE 430 M is supplied via X69. With lengths of 600 mm and longer, the ribbon cable is led doubled to the LE to increase the line cross section.

Ribbon connector, 50-pin	Assignment
1a to 5b	+5 V
6a to 7b	+12 V
8a	+5 V (low-voltage separation)
8b	0 V (low-voltage separation)
9a	+15 V
9b	-15 V
10a	UZAN
10b	0 V
11a	IZAN
11b	0 V
12a	$\overline{\text{RES.PS}}$
12b	0 V
13a	$\overline{\text{PF.PS.ZK}}$
13b	GND
14a	$\overline{\text{ERR.UZ.GR}}$
14b	GND
15a	$\overline{\text{ERR.IZ.GR}}$
15b	GND
16a	$\overline{\text{ERR.TMP}}$
16b	GND
17a	RDY.PS
17b	GND
18a	$\overline{\text{ERR.ILEAK}}$
18b	GND
19a	$\overline{\text{PF.PS.AC}}$ (not with UV 130, UE 2xx, UE 2xxB)
19b	GND
20a	Do not assign
20b	GND
21a	Do not assign
21b	GND
22a	Do not assign
22b	GND
23a	Reserved (SDA)
23b	GND
24a	Reserved (SLC)
24b	GND
25a	$\overline{\text{RES.LE}}$
25b	GND

The LE monitors the rectified voltage:

- An overvoltage up to 720 V is permissible for 5 seconds.
- Over 720 V, the NC prevents a pulse release for the IGBT of the power module: the motors coast to a non-controlled stop and there can be no energy recovery to the dc link.
- Below 385 Vdc (powerfail) all drives are brought to a controlled stop; the control must be switched off and on again.
- Below 155 Vdc, the control is reset (RESET).
- At 135 Vdc, the dc-link power supply switches off.

3.4.3 Buffer Battery for the Logic Unit

Danger

When exchanging the buffer battery, remember:

- Switch off the machine and the TNC!
- The buffer batteries may be exchanged only by trained personnel!

The buffer battery powers the RAM memory when the TNC is switched off. The batteries may be removed for exchanging for a time period not exceeding 24 hours.

LE 426 CB/PB, LE 430 CA/PA

Battery type: Three AA-size cells, leak-proof, IEC designation "LR6"

The batteries must be exchanged if the TNC displays the message **EXCHANGE BUFFER BATTERY**.

- Open the LE. The buffer batteries are located next to the power supply unit.
- To open the battery case, remove the cover with a screwdriver and one-fourth of a revolution counterclockwise.
- Exchange the batteries and ensure that the battery box is closed properly.

LE 426 M, LE 430 M

Battery type: 1 lithium battery, type CR 2450N (Renata), Id. Nr. 315 878-01

The batteries must be exchanged if the TNC displays the message **EXCHANGE BUFFER BATTERY**.

- Open the LE. The buffer battery may be found to the right beside the EPROMs of the NC software (see Position of EPROMs in Chapter 2).
- Exchange the battery; the new battery can be inserted in only one position.

3.4.4 PLC Power Supply

The PLC of the LE 426 CB/PB/M and LE 430 CA/PA/M logic units as well as the PL 410B/PL 405B 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 e.g. 15 A, this corresponds to a capacity of 2250 µF.

EN 61 131-2:1994 permits the following values:

- 5% alternating component
- 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:

PL 410B: approx. 460 W
PL 405B: approx. 235 W

Nominal operating current per output

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

X44: PLC supply voltage

Pin layout of the logic unit:

Terminal	Assignment	PLC outputs
1	+24 V cannot be switched off via EMERGENCY STOP	O24 to O30 control is ready
2	+24 V can be switched off via EMERGENCY STOP	O16 to O23
3		O0 to O15
4	0 V	

If you are using the TNC as a programming station, the PLC power supply must be connected to terminals 1 and 2.

3.4.5 Power Supply for the PL 4xxB

X9 to X14: Power supply

Pin layout on the PL 410B:

Terminal	Assignment	PL 1	PL 2	PL 3	PL 5
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 405B:

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 410B

The PL 410B 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

3.4.6 Supply Voltage for Control-is-Ready Signal (LE 426 M, LE 430 M)

X34: Power supply for control-is-ready signal

Pin layout:

Connecting terminal X34	Assignment	Connection when using a HEIDENHAIN inverter
1	+24 V	X72/1
2	0 V	X72/2

If you are using a non-HEIDENHAIN inverter system, X34 must be wired in accordance with the basic circuit diagrams at the end of the chapter.

3.4.7 Power Supply for Visual Display Units

BC 120

Connection to line power via Euro connector.

Supply voltage and power consumption	
Line voltage	100 V to 240 V
Frequency range	50 Hz to 60 Hz
Power consumption	80 W

Connecting terminal	Assignment
L1	L1 (BK)
N	MP (BL)
⏚	Equipment ground (YL/GY)

BF 120

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

Connecting terminal X1	Assignment
1	+24 V
2	0 V

Power consumption: 15 W

3.5 Encoders

3.5.1 Prerequisites

HEIDENHAIN contouring controls are designed for use with incremental linear and angular encoders as measuring systems. The 1 V_{PP} and 11 μA_{PP} signals are interpolated by a factor of 1024.
Encoders with one or more reference marks are permissible. HEIDENHAIN recommends the use of encoders with distance-coded reference marks because they greatly reduce the traverse distance required to establish the absolute position.
In addition, depending on the LE, encoders with EnDat interface can be connected:

- LE 426 M, LE 430 M with EPROMs: only certain inputs
- LE 426 M, LE 430 M with flash EPROMs:
  all speed encoder inputs, 6 position encoder inputs

Please use only HEIDENHAIN encoder cables, connectors and couplings. For maximum cable lengths, see “Cable Overview” at the end of this chapter.

Current consumption of encoders

Maximum current consumption:

- 200 mA per speed encoder input
- 100 mA per position encoder input
- 300 mA per EnDat encoder input

3.5.2 Position Encoder

LEs with power inputs (11 μA_{PP})

Input	Id. Nr. of LEs with 11-μA_{PP} input	Max. input frequency
LE 426 PB		
	311 999-xx, 313 526-xx, 317 349-xx, 318 177-xx	50 kHz
LE 426 CB		
	312 002-xx, 313 525-xx	50 kHz

LEs with voltage inputs (1 V_{PP})

Input	Id. Nr. of LEs with 1-V_{PP} input	Max. input frequency
LE 426 PB		
X6	311 999-xx, 312 000-xx, 313 526-xx, 315 527-xx, 315 475-xx, 317 349-xx, 318 177-xx, 318 178-xx, 326 414-xx, 326 416-xx, 326 420-xx, 326 421-xx	350 kHz
X1 to X5	312 000-xx, 315 527-xx, 315 475-xx, 318 178-xx	350 kHz
	326 414-xx, 326 416-xx, 326 420-xx, 326 421-xx	50 kHz
LE 426 CB		
X6	312 001-xx, 312 002-xx, 313 524-xx, 313 525-xx, 326 415-xx, 326 419-xx	350 kHz
X1 to X5	312 001-xx, 315 524-xx	350 kHz
	326 415-xx, 326 419-xx	50 kHz
LE 430 PA		
X6	311 049-xx, 313 521-xx, 326 417-xx, 325 716-xx	350 kHz
X1 to X5	311 049-xx, 313 521-xx	350 kHz
	325 716-xx, 326 417-xx	50 kHz
X35 to X38	311 049-xx, 313 521-xx, 326 417-xx, 325 716-xx	350 kHz
LE 430 CA		
X6	311 050-xx, 313 523-xx, 326 418-xx, 326 424-xx	350 kHz
X1 to X5	311 050-xx, 313 523-xx	350 kHz
	326 418-xx, 326 424-xx	50 kHz
X35 to X38	311 050-xx, 313 523-xx, 326 418-xx, 326 424-xx	350 kHz
LE 426 M, LE 430 M		
	All LEs 1 V_{PP} or EnDat	With 1 V_{PP}: 50 kHz/350 kHz switchable

Position encoder inputs with EnDat interface

EnDat interface is available at the following position encoder inputs:

LE	Input with EnDat interface
LE 426 M / 12 000 rpm with EPROMs	X5, X6
LE 426 M / 30 000 rpm with EPROMs	X5, X6
LE 430 M/6 axes with EPROMs	X5, X6
LE 430 M/9 axes with EPROMs	X1 to X6
LE 426 M / 12 000 rpm with flash EPROMs	X1 to X6
LE 426 M / 30 000 rpm with flash EPROMs	X1 to X6
LE 430 M/6 axes with flash EPROMs	X1 to X6
LE 430 M/9 axes with flash EPROMs	X1 to X6

X1 to X6, X35 to X38: Inputs for encoders with 11 µA$_{PP}$

Note

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

Pin layout for:

- LE 426 CB/PB: All inputs
- LE 430 CA/PA: All inputs
- LE 426 M, LE 430 M (xxx xxx-2x): All inputs
- LE 426 M, LE 430 M/6 axes (from xxx xxx-3x): X1 to X4
- LE 430 M/9 axes (from xxx xxx-3x): X35 to X38

Logic unit		Encoder cable	
D-sub connection (male) 15-pin	**Assignment**	**D-sub cnnctr. (female) 15-pin**	
1	+5 V	1	Brown
2	0 V	2	White
3	I_1 +	3	Green
4	I_1 –	4	Yellow
5	0 V	5	White/Brown (internal shield)
6	I_2 +	6	Blue
7	I_2 –	7	Red
8	0 V	8	
9	+5 V	9	
10	I_0 +	10	Gray
11	0 V	11	
12	I_0 –	12	Pink
13	0 V	13	
14	Do not assign	14	
15	Do not assign	15	
Housing	External shield	Housing	External shield

Pin layout for:

- LE 426 M, LE 430/6 axes with EPROMs (as of xxx xxx-3x): X5, X6
- LE 430 M/9 axes with EPROMs (from xxx xxx-3x): X1 to X6
- LE 426 M, LE 430 M with flash EPROMs: X1 to X6

Logic unit		**Adapter connection (317 505-05)**			**Encoder cable**	
D-sub connection (male) 15-pin	**Assignment**	**Female**		**Male**	**D-sub cnnctr. (female) 15-pin**	
1	+5 V	1		1	1	Brown
2	0 V	2		2	2	White
3	I_1 +	3		3	3	Green
4	I_1 –	4		4	4	Yellow
5	Do not assign	5		5	5	White/Brown (internal shield)
6	I_2 +	6		6	6	Blue
7	I_2 –	7		7	7	Red
8	0 V	8		8	8	
9	+5 V	9		9	9	
10	I_0 +	10		10	10	Gray
11	0 V	11		11	11	
12	I_0 –	12		12	12	Pink
13	0 V	13		13	13	
14	Do not assign	14		14	14	
15	Do not assign	15		15	15	
Housing	External shield	Housing			Housing	External shield

X1 to X6, X35 to X38: Inputs for encoders 1 to 10 with 1 V_{PP}

Pin layout:

Note

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

LE		**AK 309 783-xx AK 310 199-xx**			**Encoder**	
Male	**Assignment**	**Female**	**Color**	**Female**	**Male**	**Color**
1	+5 V (U_P)	1	Brown/Green	12	12	Brown/Green
2	0 V (U_N)	2	White/Green	10	10	White/Green
3	A+	3	Brown	5	5	Brown
4	A–	4	Green	6	6	Green
5	Do not assign	5				
6	B+	6	Gray	8	8	Gray
7	B–	7	Pink	1	1	Pink
8	Do not assign	8				
9	+5 V (sensor line)	9	Blue	2	2	Blue
10	R+	10	Red	3	3	Red
11	0 V (sensor line)	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

X1 to X6, X35 to X38: Inputs for encoders 1 to 10 with EnDat interface

Pin layout:

Note

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

LE		AK 332 115-xx			VB 323 897-xx			AK 313 791-xx		
Male	**Assgnmnt.**	**Female**	**Color**	**Female**	**Male**	**Color**	**Female**	**Male**	**Color**	**Female**
1	+5 V (U_P)	1	Brown/ Green	7	7	Brown/ Green	7	7	Brown/ Green	5b
2	0 V (U_N)	2	White/ Green	10	10	White/ Green	10	10	White/ Green	6a
3	A+	3	Green/ Black	15	15	Green/ Black	15	15	Green/ Black	2a
4	A–	4	Yellow/ Black	16	16	Yellow/ Black	16	16	Yellow/ Black	2b
5	Data	5	Gray	14	14	Gray	14	14	Gray	3b
6	B+	6	Blue/ Black	12	12	Blue/ Black	12	12	Blue/ Black	1a
7	B–	7	Red/ Black	13	13	Red/ Black	13	13	Red/ Black	1b
8	$\overline{\text{Data}}$	8	Pink	17	17	Pink	17	17	Pink	3a
9	+5 V (sensor 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.	External shield	Hsg.		Externa l shield		Hsg.	External shield	

3.5.3 Speed Encoder (LE 426 PB/M, LE 430 PA/M)

Maximum input frequency

Maximum input frequency of speed encoder inputs:

Input	Max. input frequency
X15 to X20	350 kHz
X62 to X64	
X60	410 kHz to NC software 280 470-xx: 350 kHz

Note

Keep in mind the line count of the speed encoder when choosing the spindle motor:

$$x = \frac{f \cdot 60 \cdot 1000}{n}$$

x = line count of the speed encoder
f = maximum input frequency
n = maximum spindle speed

Example:
f = 410 kHz; n = 24 000 rpm

$$x = \frac{410 \cdot 60 \cdot 1000}{24000} \approx 1024$$

Inputs with EnDat interface

EnDat interface is available at the following speed encoder inputs:

LE	Input with EnDat interface
LE 426 M / 12 000 rpm with EPROMs	X19, X20
LE 426 M / 30 000 rpm with EPROMs	X19, X60
LE 430 M/6 axes with EPROMs	X19, X20
LE 430 M/9 axes with EPROMs	X15 to X20, X62 to X64, X60
LE 426 M / 12 000 rpm with flash EPROMs	X15 to X20
LE 426 M / 30 000 rpm with flash EPROMs	X15 to X19, X60
LE 430 M/6 axes with flash EPROMs	X15 to X20, X60
LE 430 M/9 axes with flash EPROMs	X15 to X20, X62 to X64, X60

X15 to X20, X62 to X64, X60: Inputs for 1 V_{PP}

Pin layout:

Note

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

LE		AK 289 440-xx			VB 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.

X15 to X20, X62 to X64, X60: Inputs with EnDat interface

Pin layout:

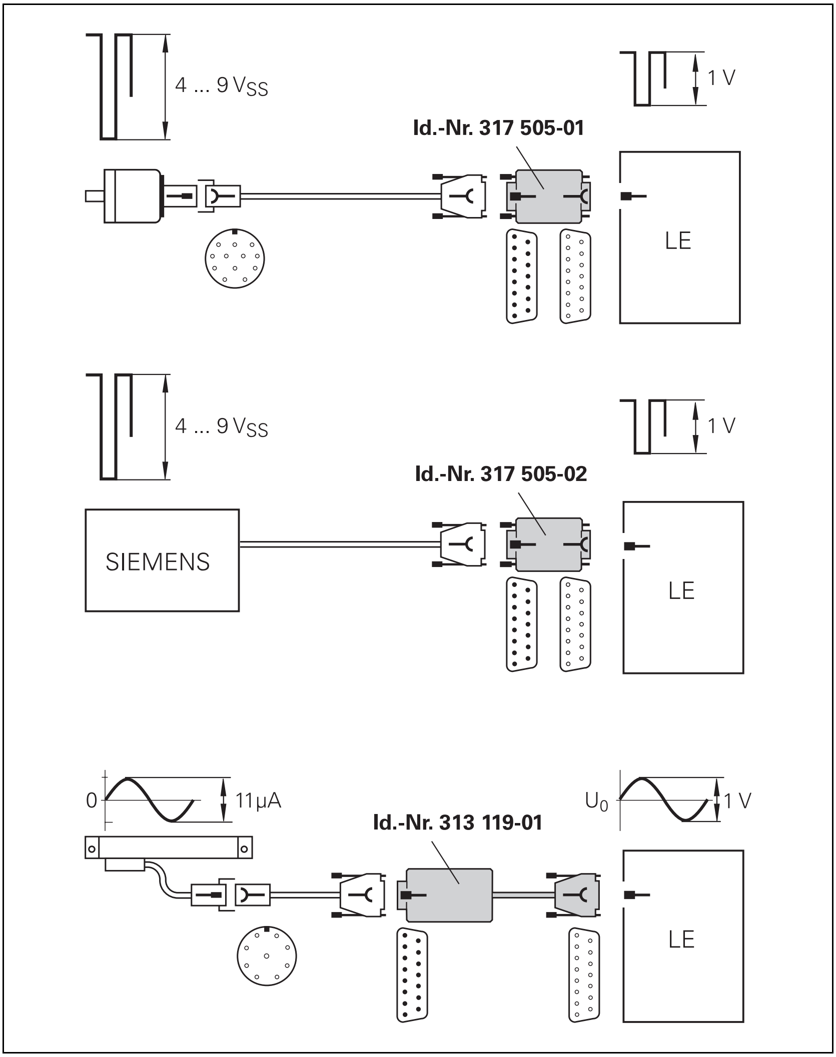
Note

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

LE		**AK 336 376-xx**				**VB 340 302-xx**		
Male	**Assignment**	**Female**	**Color**	**Female**		**Male**	**Color**	**Female**
1	+5 V (U_P)	1	Brown/Green	10	Line drop compensator 336 397-01, 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.	External shield	Hsg.		Hsg.	External shield	Hsg.

3.5.4 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 connctr. (female) 15-pin	Assignment	D-sub connctn. (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 connctr. (female) 15-pin	Assignment	D-sub connctn. (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 connctr. (female) 15-pin	Assignment	D-sub connctn. (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

3.6 Connecting the Motor Power Module

The LE 426 M and the LE 430 M can be operated with HEIDENHAIN and with non-HEIDENHAIN inverter systems. The LE 426 PB and the LE 430 PA can only be operated with non-HEIDENHAIN inverter systems.

For a description of the HEIDENHAIN inverter systems, refer to the Technical Manual "Inverter Systems and Motors." The components required for operation of the LE with non-HEIDENHAIN inverter systems are described in the manual "Technical Information for the Operation of SIMODRIVE and POWER DRIVE Inverter Systems."

X51 to X59, X61: Outputs to the motor power module

The following applies for the outputs to the motor power module:

Logic level: 5 V
Analog signals I_{ACT}: ±7.5 V
PWM frequency X51 to X59: Can be set from 3 kHz to 7 kHz via MP2180
PWM frequency X61: 5 kHz

Output	Speed
X51 to X59	Axis 1 to 9, or spindle (TNC 426 PB/M with 12 000 rpm)
X61	Spindle (TNC 430 PA/M and TNC 426 PB/M with 30 000 rpm)

Pin layout of logic unit, connecting cable, and expansion board:

Note

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

Logic unit		**Connecting cable Id. Nr. 289 208-xx**			**Expansion board Id. Nr. 324 952-xx**	
D-sub conninctn. (female) 15-pin	**Assignment**	**D-sub connctr. (male) 15-pin**		**D-sub connctr. (female) 15-pin**	**X1, X2 D-sub connection (female) 15-pin**	
1	Do not assign	1	Black	1	1	
2	PWM U_1	2	Blue	2	2	
3	PWM U_2	3	Gray	3	3	
4	PWM U_3	4	White	4	4	
5	$\overline{\text{Reset}}$	5	Green	5	5	
6	Ready	6	White/Pink	6	6	
7	$-I_{act2}$ –	7	Gray/Pink	7	7	
8	$-I_{act1}$ –	8	Black	8	8	
9	0 V U_1	9	Red	9	9	
10	0 V U_2	10	Pink	10	10	
11	0 V U_3	11	Brown	11	11	
12	0 V (analog)	12	Brown/Green	12	12	
13	$\overline{\text{Temp. warn.}}$	13	Red/Green	13	13	
14	$+I_{act2}$	14	Red/Blue	14	14	
15	$+I_{act1}$	15	Violet	15	15	
Housing	External shield	Housing	External shield	Housing	Housing	

Pin layout for LE 426 M, LE 430 M:

Ribbon connector, 20-pin	**Assignment**
1a	PWM U_1
1b	0 V U_1
2a	PWM U_2
2b	0 V U_2
3a	PWM U_3
3b	0 V U_3
4a	$\overline{\text{SH2}}$
4b	0 V (–SH2)
5a	$\overline{\text{SH1B}}$
5b	0 V ($\overline{\text{SH1B}}$)
6a	$+I_{actl\ 1}$
6b	$-I_{actl\ 1}$
7a	0 V (analog)
7b	$+I_{actl\ 2}$
8a	$-I_{actl\ 2}$
8b	0 V (analog)
9a	Do not assign
9b	Do not assign
10a	$\overline{\text{Temp. warning}}$
10b	Ready

3.7 Analog Input

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

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

The current values of the inputs can be requested with Module 9003.
(See “Overview of Modules” on page 5 – 3).

	Analog inputs (±10 V)	**Inputs for Pt 100 thermistors**
Logic unit	3	3
PL 405B	–	–
PL 410B	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 (LE) 100 mV (PL 410 B)
Internal value range:	–100 to +100, at a resolution of 100 mV -10 to +10, 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 (LE) 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 logic unit

Pin layout:

Warning

Remember to connect the analog inputs with the correct polarity!

Note

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

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

X15 to X18: Analog input on the PL 410B

Pin layout:

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

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

Pin layout:

Note

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

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

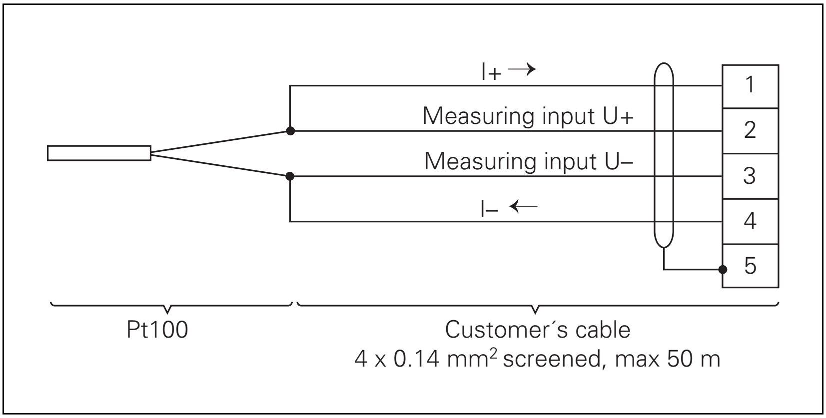
Connection to the analog inputs

Characteristics of the connecting cable:

- Shielding
- 2 conductors with 0.14 mm^2 each
- Maximum length 50 meters

Connection to the inputs for Pt 100 thermistors

▶ Configure the thermistor connection as a “four-conductor circuit.”



3.8 Analog Output

Output:	±10 V
Maximum load of outputs:	2 mA
Maximum capacity:	2 nF

13 analog outputs are available:

- Connection X8: Analog output 1 to 6
- Connection X9: Analog output 7 to 13

PLC analog output

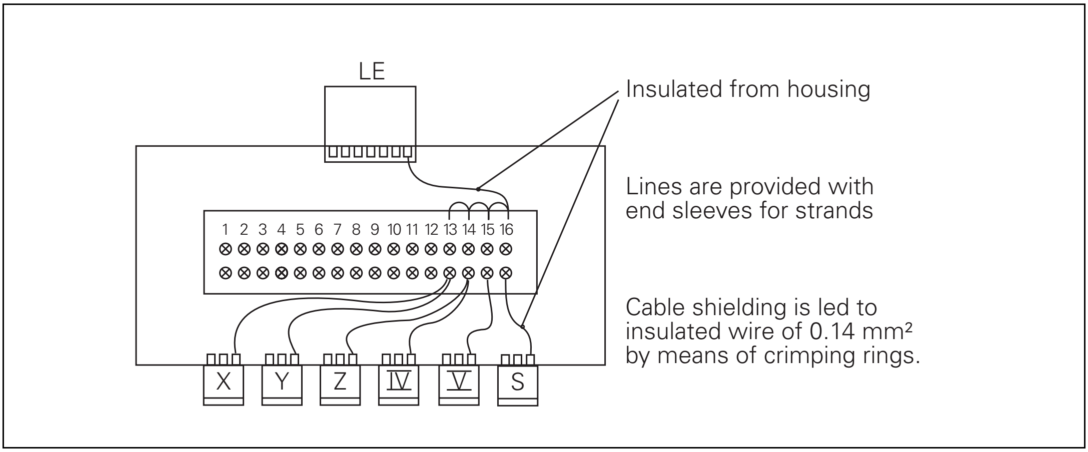
The PLC analog outputs can be controlled through Module 9130.
(See "Overview of Modules" on page 5 – 3).

3.8.1 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	Nominal value in spindle	±10 V
12		0 V
13 to 16	Shield connection	

X8: Analog output

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

Pin layout on logic unit and connecting cable:

Logic unit		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 end of chapter.

Pin layout on logic unit and connecting cable:

Logic unit		**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	Analog output 13: ±10 V	2	Brown/Green
3	Analog output 8: ±10 V	3	Yellow
4	Analog output 11: ±10 V	4	Red/Blue
5	Analog output 9: ±10 V	5	Pink
6	Analog output 11: 0 V	6	Gray/Pink
7	Analog output 10: ±10 V	7	Red
8	Analog output 12: ±10 V	8	Violet
9	Analog output 7: 0 V	9	White
10	Analog output 13: 0 V	10	White/Gray
11	Analog output 8: 0 V	11	Green
12	Do not assign	12	
13	Analog output 9: 0 V	13	Gray
14	Analog output 10: 0 V	14	Blue
15	Analog output 12: 0 V	15	Black
Housing	External shield	Housing	External shield

3.9 Touch Probe Systems

The following touch probes can be connected to the TNC:

- TS 220, a touch-trigger probe with cable connection for digitizing, 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 tool presetting
- Measuring touch probe

For suitable connecting cables, see "Cable Overview" at end of chapter.

3.9.1 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:

LE		**AK 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 550:

LE		AK 310 197-xx			EA 550 262 904-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 552 can be connected to the LE via the APE 511.
This is necessary on large machines or on machines with swivel heads, for example.

Pin layout for TS 632 with two EA 552 via the APE 511:

LE	AK 310 197-xx	APE 511 275 759-xx		VB 336 157-xx			EA 552 339 317-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.		

3.9.2 Triggering Touch Probe for Tool Measurement

X13: Connection of the touch probe

Pin layout of the logic unit:

Note

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

Pin layout on adapter cable and touch probe:

LE		**AK 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.

3.9.3 Measuring Touch Probe (Option)

The X14 connection for the measuring touch probe is not included. The “digitizing with measuring touch probe” option is available as an adapter kit that includes connection X14. With the TNC 430 CA/PA you can use either the adapter kit for the measuring touch probe input X14, or the adapter kit for an additional position encoder input X38.

X14: Measuring touch probe SP 2/1

Pin layouts on logic unit, adapter cable, connecting cable, and encoder:

Note

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

Logic unit (mounting set)			**Adapter cable Id. Nr. 296 839-xx**			**Renishaw VB A-1016-6640**			**Reni-shaw SP 2/1**
D-sub cnnctr. (female) 25-pin	**Assgnmnt.**		**D-sub cnnctr. (male) 25-pin**		**Cplng. on mntng. base (female) 21-pin**	**Cnnctr. (male) 21-pin**			
3	$\overline{Ua2}$	Axis X	3	Pink	7				
4	$\overline{Ua1}$		4	Yellow	5				
16	Ua2		16	Gray	6				
17	Ua1		17	Green	4				
7	$\overline{Ua2}$	Y axis	7	Brown/Blue	11				
8	$\overline{Ua1}$		8	Red	9				
20	Ua2		20	White/Blue	10				
21	Ua1		21	Blue	8				
11	$\overline{Ua2}$	Z axis	11	Violet	19				
12	$\overline{Ua1}$		12	Red/Blue	13				
24	Ua2		24	Black	18				
25	Ua1		25	Gray/Pink	12				
1	0 V		1	White	1				
5	+12 V		5	Brown	3				
9	Overtravel 1		9	White/ Green	15				
13	0 V		13						
14	Overtravel 2		14	Brown/ Green	21				
18	ERROR		18	White/Gray	14				
22	$\overline{\text{SWITCH}}$		22	Gray/Brown	20				
2, 6, 10, 15, 19, 23	Do not assign								
Housing	Ext. shield		Housing	Ext. shield	Housing	Housing	Ext. shield	Hsg.	Hsg.

3.10 Data Interface

The TNC features three interfaces:

- RS-232-C/V.24
- RS-422/V.11
- Ethernet connection

The interfaces can be used simultaneously. The user selects the interfaces he wishes to use. (See “Data Interfaces” on page 8 – 3).

3.10.1 RS-232-C/V.24 Data Interface

Please note:

- Maximum cable length 20 m.
- To connect a peripheral device you must install an adapter cable either in the electrical cabinet or on the operating panel. See also “Dimensions” at the end of this chapter.
- For connecting cables, see “Cable Overview” at the end of this chapter.

X21: RS-232-C/V.24 data interface

Pin layouts on logic unit, connecting cables, and adapter block:

Note

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

Logic unit		**Connecting cable Id. Nr. 239 760-xx**			**AB Id. Nr. 310 085-01**		**Connecting cable Id. Nr. 274 545-01**		
D-sub cnnctr. (female) 25-pin	**Assign-ment**	**D-sub cnnctr. (male) 25-pin**		**D-sub cnnctr. (female) 25-pin**	**D-sub cnnctr. (male) 25-pin**	**D-sub cnnctr. (female) 25-pin**	**D-sub cnnctr. (male) 25-pin**		**D-sub cnnctr. (female) 25-pin**
1	GND	1	WH/BN Ext. shield	1	1	1	1	WH/BN External shield	1
2	RXD	2	Green	3	3	3	3	Yellow	2
3	TXD	3	Yellow	2	2	2	2	Green	3
4	CTS	4	Gray	5	5	5	5	Pink	4
5	RTS	5	Pink	4	4	4	4	Gray	5
6	DTR	6	Blue	20	20	20	20	Brown	6
7	Signal GND	7	Red	7	7	7	7	Red	7
20	DSR	20	Brown	6	6	6	6	Blue	20
8 to 19, 21 to 25	Do not assign			8	8	8	8		8
Housing	Ext. shield	Housing	Ext. shield	Housing	Housing	Housing	Housing	Ext. shield	Housing

3.10.2 RS-422/V.11 Data Interface

Please note:

- Maximum cable length 1000 m.
- To connect a peripheral device you must install an adapter cable either in the electrical cabinet or on the operating panel. See also "Dimensions" at the end of this chapter.
- For connecting cables, see "Cable Overview" at the end of this chapter.

X22: RS-422/V.11 data interface

Pin layouts on logic unit, connecting cables, and adapter block:

Logic unit		Connecting cable Id. Nr. 289 208-xx			PL Id. Nr. 310 086-01	
D-sub cnnctr. (female) 15-pin	**Assignment**	**D-sub connctr. (male) 15-pin**		**D-sub cnnctr. (female) 15-pin**	**D-sub connctn. (male) 15-pin**	**D-sub cnnctr. (female) 15-pin**
1	Chassis GND	1	black external shield	1	1	1
2	RXD	2	Blue	2	2	2
3	CTS	3	Gray	3	3	3
4	TXD	4	White	4	4	4
5	RTS	5	Green	5	5	5
6	DSR	6	White/Green	6	6	6
7	DTR	7	Green/Pink	7	7	7
8	Signal GND	8	Black	8	8	8
9	$\overline{\text{RXD}}$	9	Red	9	9	9
10	$\overline{\text{CTS}}$	10	Pink	10	10	10
11	$\overline{\text{TXD}}$	11	Brown	11	11	11
12	$\overline{\text{RTS}}$	12	Yellow	12	12	12
13	$\overline{\text{DSR}}$	13	Brown/Green	13	13	13
14	$\overline{\text{DTR}}$	14	Red/Blue	14	14	14
15	Do not assign	15	Violet	15	15	15
Housing	External shield	Housing	External shield	Housing	Housing	Housing

3.10.3 Ethernet Interface (Option)

Maximum data transfer rate: 200 Kbps to 1 Mbps

X26: RJ45 connection

Please note the maximum cable length:

- Unshielded 100 m
- Shielded 400 m

Note

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

RJ45 connection (female) 8-pin	Assignment
1	TX+
2	TX–
3	REC+
4	Do not assign
5	Do not assign
6	REC –
7	Do not assign
8	Do not assign

3.11 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 of the logic unit:

Note

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

D-sub connection (female) 9-pin	Assignment
1	CTS
2	0 V
3	RTS
4	+12 V ± 0.6 V (U_V)
5	Do not assign
6	DTR
7	$\overline{\text{TxD}}$
8	$\overline{\text{RxD}}$
9	DSR
Housing	External shield

3.11.1 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

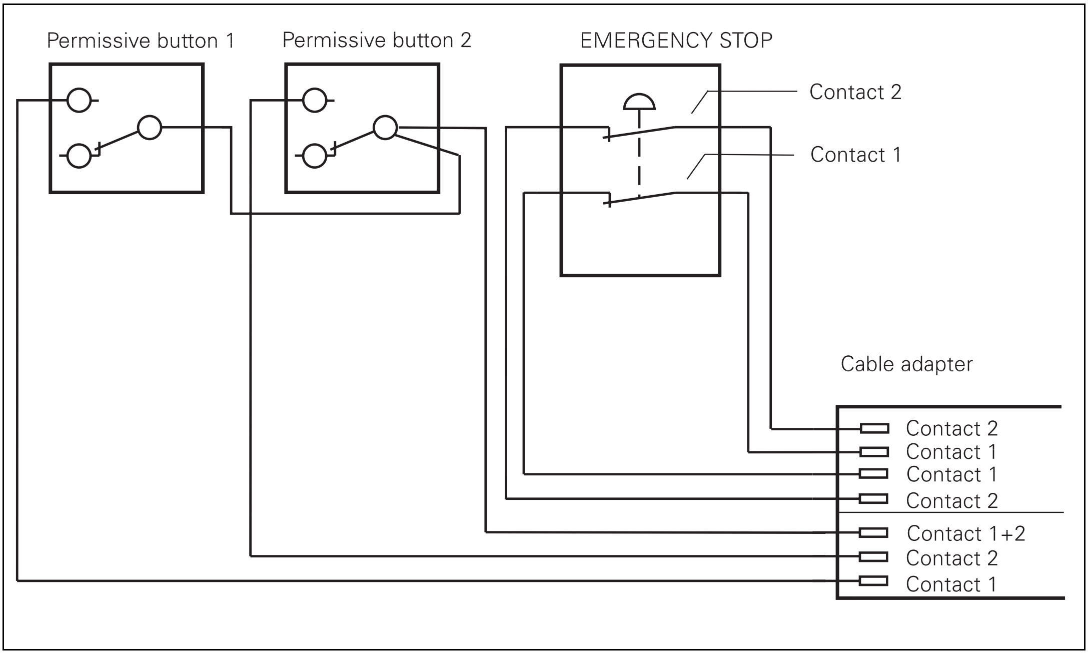
Pin layout for the various extension cables, adapter cables, connecting cables, and the handwheel:

Extension cable Id. Nr. 281 429-xx			**Adapter cable Id. Nr. 296 466-xx**			**Connecting cable Id. Nr. see next page**			**HR 410 Id. Nr. 296 469-xx**	
D-sub cnnctr. (male) 9-pin		**D-sub cnnctr. female 9-pin**	**D-sub cnnctr. (male) 9-pin**		**Cplng. on mntng. base (fem.) (5+7)-pin**	**Cnnctr. (male) (5+7)-pin**		**Cnnctr. (female) (5+7)-pin**	**Cnnctr. (male) (5+7)-pin**	
Housing	Shield	Housing	Hsg.	Shield	Housing	Housing	Shield	Housing	Hsg.	Shield
2	White	2	2	White	E	E	White	E	E	
4	Brown	4	4	Brown	D	D	Brown	D	D	
6	Yellow	6	6	Yellow	B	B	Yellow	B	B	
7	Gray	7	7	Gray	A	A	Gray	A	A	
8	Green	8	8	Green	C	C	Green	C	C	
					6	6	BK	6	6	
					7	7	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 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

3.11.2 HR 130 Panel-Mounted Handwheel

Standard cable length for the HR 130 is 1 meter.

Pin layout for extension cable and handwheel:

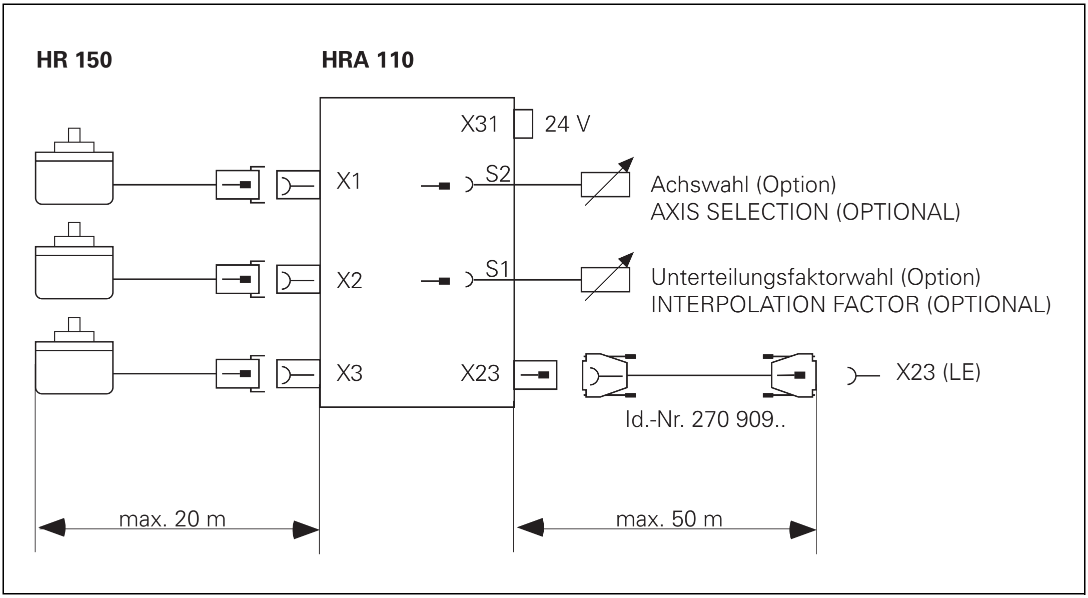
Extension cable Id. Nr. 281 429-xx			HR 130 Id. Nr. 254 040-xx	
D-sub cnnctr. (male) 9-pin	**Color**	**D-sub connctr. (female) 9-pin**	**D-sub cnnctr. (male) 9-pin**	**Color**
Housing	Shield	Housing	Housing	Shield
2	White	2	2	White
4	Brown	4	4	Brown
6	Yellow	6	6	Yellow
8	Green	8	8	Green
7	Gray	7		

3.11.3 HRA 110 Handwheel Adapter

With the handwheel adapter you can connect two or three panel-mounted HR 150 handwheels to the TNC.

The first and second handwheels are assigned to the X and Y axes. The third handwheel can be assigned to axes Z, IV or V either through a selection switch (option) or with MP7645. (See "Machine Parameters" on page 4 – 3).

An additional switch enables you to select, e.g., 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. (See "Overview of Modules" on page 5 – 3).



X1, X2, X3: Inputs for HR 150 handwheels

Pin layout on the HRA 110 for the HR 150:

HRA 110 **261 097-xx**	
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 the logic unit

Pin layout on the HRA 110:

HRA 110 **261 097-xx**	
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: Power supply

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 **261 097-xx**	
Connecting terminal	**Assignment**
1	+ 24 Vdc as per IEC 742 (VDE 551)
2	0 V

Maximum current consumption 200 mA.

3.12 Input: Spindle Reference Signal

If you have mounted a HEIDENHAIN rotary encoder directly on the spindle, i.e., without a mechanical transmission ratio, you must not connect this input.

If you use the X30 input for evaluation of the reference signal, then adjust this function with MP3143. (See "Machine Parameters" on page 4 – 3).

X30: Reference signal for spindle

Pin layout:

Connecting terminal	Assignment
1	+24 V
2	0 V

3.13 Input: Drive Motor Enabling (LE 426 PB, LE 430 PA)

A power supply of 24 Vdc must be available at X50 to enable the drive motors.

X50: Drive enabling

Pin layout for logic units up to Id. Nr. xxx xxx-3x:

Connecting terminal	Assignment
1	+24 Vdc
2	Do not assign
3	0 V

Pin layout for logic units beginning with Id. Nr. xxx xxx-4x:

Connecting terminal	Assignment
1	+24 Vdc
2 to 6	Do not assign
7	0 V

3.14 Switching Inputs 24 Vdc (PLC)

3.14.1 Input Signals and Addresses

Input signals of the switching inputs on the LE and the PL 4xxB:

Voltage range	Logic unit	PL 4xxB
“1” signal: U_i	13 V to 30.2 V	
“0” signal: U_i	-20 V to 3.2 V	

Current ranges	Logic unit	PL 4xxB
“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	Logic unit X42 (PLC input)
I128 to I152	25	Logic unit X 46 (machine operating panel)
I64 to I127 I64 to I95	64 32	First PLC input/output board PL 410B First PLC input/output board PL 405B
I192 to I255 I192 to I223	64 32	Second PLC I/O board PL 410B Second PLC input/output board PL 405B
I256 to I319 I256 to I287	64 32	Third PLC input/output board PL 410B Third PLC input/output board PL 405B
I320 to I383 I320 to I351	64 32	Fourth PLC input/output board PL 410B Fourth PLC input/output board PL 405B

X42: PLC input on the logic unit

Pin layout on the LE:

Logic unit		Connecting cable Id. Nr. 244 005-xx, Id. Nr. 263 954-xx	
D-sub connection (female) 37-pin	**Assignment**	**D-sub connection (male) 37-pin**	
1	I0	1	Gray/Red
2	I1	2	Brown/Black
3	I2	3	White/Black
4	I3 Control-is-ready signal acknowledgement	4	Green/Black
5	I4	5	Brown/Red
6	I5	6	White/Red
7	I6	7	White/Green
8	I7	8	Red/Blue
9	I8	9	Yellow/Red
10	I9	10	Gray/Pink
11	I10	11	Black
12	I11	12	Pink/Brown

Logic unit		Connecting cable Id. Nr. 244 005-xx, Id. Nr. 263 954-xx	
D-sub connection (female) 37-pin	**Assignment**	**D-sub connection (male) 37-pin**	
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	LE 426 CB/PB, LE 430 CA/PA: Do not use LE 426 M, LE 430M: I32 Drive enabling	33	Pink/Green
34	Do not assign	34	Brown
35	0 V (PLC) Test output; do not assign	35	Yellow/Pink
36	0 V (PLC) Test output; do not assign	36	Violet
37	0 V (PLC) Test output; do not assign	37	White
Housing	External shield	Housing	External shield

3.14.2 PLC Inputs on the PL 410B

X3, X4, X5, X6: PLC input

Pin layout on the PL:

X3				
Terminal	**Assignment**			
	1st PL	**2nd PL**	**3rd PL**	**4th PL**
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**			
	1st PL	**2nd PL**	**3rd PL**	**4th PL**
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**			
	1st PL	**2nd PL**	**3rd PL**	**4th PL**
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**			
	1st PL	**2nd PL**	**3rd PL**	**4th PL**
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

3.14.3 PLC Inputs on the PL 405B

X3, X4:
PLC input

Pin layout on the PL:

X3				
Terminal	**Assignment**			
	1st PL	**2nd PL**	**3rd PL**	**4th PL**
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**			
	1st PL	**2nd PL**	**3rd PL**	**4th PL**
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

3.15 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:

	Logic unit	**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	Logic unit X41 (PLC output)
O0 to O7	8	Logic unit X46 (machine operating panel)
O32 to O62	31	First PLC input/output unit
O64 to O94	31	Second PLC input/output unit
O128 to O158	31	Third PLC input/output unit
O160 to O190	31	Fourth PLC input/output unit

X41: PLC output on the logic unit

Pin layout on the LE:

Logic unit		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/Blue
15	O14	15	Yellow
16	O15	16	Red
Supply via X44, pin 2; disconnectable by EM. 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; **not** disconnectable with EM. STOP			
25	O24	25	Yellow/Brown
26	O25	26	Gray/Brown
27	O26	27	Yellow/Brown
28	O27	28	White/Yellow
29	O28	29	Gray/White
30	O29	30	Pink/Blue
31	O30	31	Pink/Red

Logic unit		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**	
32	Test output; do not assign	32	Brown/Blue
33	Test output; do not assign	33	Pink/Green
34	Control is ready	34	Brown
35	Test output; do not assign	35	Yellow/Pink
36	Test output; do not assign	36	Violet
37	Test output; do not assign	37	White
Housing	External shield	Housing	External shield

X7, X8: PLC output on the PL 410B

Pin layout on the PL:

X7				
Terminal	**Assignment**			
	1st PL	**2nd PL**	**3rd PL**	**4th PL**
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**			
	1st PL	**2nd PL**	**3rd PL**	**4th PL**
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 output on the PL 405B

X8				
Terminal	**Assignment**			
	1st PL	**2nd PL**	**3rd PL**	**4th PL**
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			

3.16 Connecting the PL 4xxB Input/Output Unit

Up to four PL 4xxB can be connected to the TNC.

TNC 426 CB, TNC 430 CA:
One PL can be mounted in the logic unit, additional PLs are mounted in the electrical cabinet.

TNC 426 PB/M, TNC 430 PA/M:
The PLs must be mounted separately in the electrical cabinet.

The PL 410B 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 410B	263 371-xx	64	31	–	–
PL 410B	263 371-xx	64	31	4	4
PL 405B	263 371-xx	32	15	–	–

No more than one PL 405B may be used. If connecting through PL 410B, the PL 405B must be connected last.

X47: PLC expansion on the LE

Pin layout of logic unit, connecting cable, and PL:

Logic unit		Connecting cable Id. Nr. 289 111-xx			First PL 4xx B	
D-sub cnnctr. (male) 25-pin	Assignment	D-sub cnnctr. (female) 25-pin		D-sub cnnctr. (male) 25-pin	D-sub cnnctn. (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
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 4xxB on the PL 410B

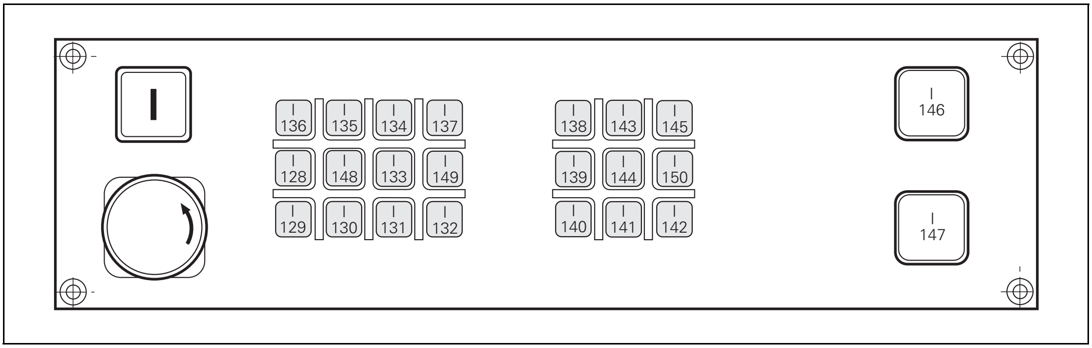
Pin layout of logic unit, connecting cable, and PL:

PL 410B		Connecting cable Id. Nr. 289 111-xx			PL 4xxB	
D-sub cnnctr. (male) 25-pin	**Assignment**	**D-sub cnnctr. (female) 25-pin**		**D-sub cnnctr. (male) 25-pin**	**D-sub cnnctn. (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	External shield	Housing	External shield	Housing	Housing	External shield

3.17 Machine Operating Panel

For machines with up to four axes, HEIDENHAIN offers the MB 420 machine operating panel. It is installed below the operating panel. There are two versions of the MB 420 available, one with a standard set of keys and the other with a set of keys to be selected and placed by the machine tool builder. An assortment of approx. 40 placeable keys is supplied with the latter version. 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:



X46: PLC inputs and outputs

PLC inputs I128 to I152 and the PLC outputs O0 to O7 are located at X46 of the machine operating panel. The reference potential (PLC) for outputs O0 to O7 is connected to pins 34 and 35.

Pin layout of logic unit, connecting cable, and machine operating panel:

Warning

PLC inputs I128 to I152 must be driven only with the power supply from pins 36 and 37, because this power supply is internally protected (PLC power supply from X44 connection 2).

Logic unit		Connecting cable Id. Nr. 263 954-xx			MB 420; 293 757-xx	
D-sub cnnctr. (female) 37-pin	Assignment	D-sub cnnctr. (male) 37-pin		D-sub cnnctr. (female) 37-pin	D-sub cnnctr. (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	FN1
12	I139	12	Pink/Brown	12	12	FN2
13	I140	13	Yellow/Blue	13	13	FN3
14	I141	14	Green/Blue	14	14	FN4
15	I142	15	Yellow	15	15	FN5
16	I143	16	Red	16	16	Spindle on
17	I144	17	Gray	17	17	Spindle off
18	I145	18	Blue	18	18	Coolant On/Off
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	Black
23	I150	23	White/Pink	23	23	Black
24	I151	24	Gray/Green	24	24	Via X3
25	I152	25	Yellow/Brown	25	25	Via X3
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	0 V (PLC)	34	Brown	34	34	
35	0 V (PLC)	35	Yellow/Pink	35	35	
36	+24 V (PLC)	36	Violet	36	36	
37	+24 V (PLC)	37	White	37	37	
Housing	External shield	Housing	External shield	Housing	Housing	

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

3.18 TNC Keyboard Unit

The TNC keyboard is connected by cable with the logic unit, and by ribbon cable to the soft keys of the visual display unit. The ribbon cable is included with the visual display unit.

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

Pin layout of the TNC keyboard:

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

X45: TNC keyboard unit

Pin layout of logic unit, connecting cable, and TNC keyboard unit:

Logic unit		Connecting cable Id. Nr. 263 954-xx			TE 420 313 038-xx
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

Logic unit		Connecting cable Id. Nr. 263 954-xx			TE 420 313 038-xx
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**
17	RL16	17	Gray	17	17
18	RL17	18	Blue	18	18
19	RL18	19	Pink	19	19
20	SL0	20	White/Gray	20	20
21	SL1	21	Yellow/Gray	21	21
22	SL2	22	Green/Red	22	22
23	SL3	23	White/Pink	23	23
24	SL4	24	Gray/Green	24	24
25	SL5	25	Yellow/ Brown	25	25
26	SL6	26	Gray/Brown	26	26
27	SL7	26	Yellow/Black	27	27
28	RL19	28	White/Yellow	28	28
29	RL20	29	Gray/Blue	29	29
30	Do not assign	30	Pink/Blue	30	30
31	RL21	31	Pink/Red	31	31
32	RL22	32	Brown/Blue	32	32
33	RL23	33	Pink/Green	33	33
34	Spindle override (wiper)	34	Brown	34	34
35	Feed rate override (wiper)	35	Yellow/Pink	35	35
36	+5 V override potentiometer	36	Violet	36	36
37	0 V override potentiometer	37	White	37	37
Housing	External shield	Housing	External shield	Housing	Housing

3.19 Visual Display Unit

Two display units are available:

- BC 120, 15-inch color screen
- BF 120, TFT color flat-panel display

When ordering, make sure that you also order the fitting logic unit for the screen.

Depending on type of display unit (BC 120 or BC 110B) and LE, an adapter may be necessary for connecting the screen:

LE	BC 120	BC 110B
LE 426 CB/PB, LE 430 CA/PA to hardware xxx xxx-3x	Adapter two-row/three-row Id. Nr.: 313 434-02	No adapter required
LE 426 CB/PB, LE 430 CA/PA from hardware xxx xxx-4x	No adapter required	Adapter three-row/two-row Id. Nr.: 313 434-01
LE 426 M, LE 430 M as of hardware xxx xxx-2x	No adapter required	Adapter three-row/two-row Id. Nr.: 313 434-01

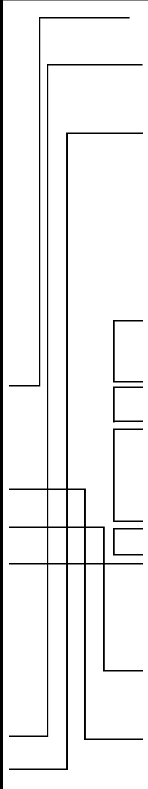
The ribbon cable for connecting the display unit soft keys with the TE 420 is included with the display unit.

X43: Visual Display Unit BC 120

Pin layout for the LE 426 CB/PB, LE 430 CA/PA with Id. Nr. xxx xxx-3x, the adapter, the connecting cable, and the visual display unit:

Note

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

Logic unit Id. Nr. xxx xxx-3x		**Adapter 313 434-02**	**VL 312 878-xx**			**BC 120**
D-sub conncton. (female) 15-pin 2-row	**Assignment**	**2-row/ 3-row**	**D-sub connctr. (male) 15-pin 3-row**		**D-sub connctr. (female) 15-pin 3-row**	**D-sub connctr. (male) 15-pin 3-row**
1	GND		1	Coax I red	1	1
2	Do not assign		2	Coax I green	2	2
3	Do not assign		3	Coax I blue	3	3
4	Do not assign		4		4	4
5	Do not assign		5		5	5
6	Do not assign		6	Coax S red	6	6
7	R		7	Coax S GN	7	7
8	Do not assign		8	Coax S blue	8	8
9	HSYNC		9		9	9
10	VSYNC		10	Gray	10	10
11	GND		11	Green	11	11
12	Do not assign		12		12	12
13	Do not assign		13	Pink	13	13
14	G		14	Yellow	14	14
15	B		15		15	15
Housing	External shield	Housing	Housing	External shield	Housing	Housing

X43: Visual Display Unit BC 120

Pin layout for the LE 426 CB/PB, LE 430 CA/PA with Id. Nr. xxx xxx-4x and an LE 426 M, LE 430 M, the connecting cable, and the visual display unit:

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

Logic unit Id. Nr. xxx xxx-4x		VB 312 878-xx			BC 120
D-sub connctn. (female) 15-pin 3-row	Assignment	D-sub cnnctr. (male) 15-pin 3-row		D-sub connctr. (female) 15-pin 3-row	D-sub cnnctr. (male) 15-pin 3-row
1	R	1	Coax I red	1	1
2	G	2	Coax I green	2	2
3	B	3	Coax I blue	3	3
4	Do not assign	4		4	4
5	Do not assign	5		5	5
6	GND	6	Coax S red	6	6
7	GND	7	Coax S GN	7	7
8	GND	8	Coax S blue	8	8
9	Do not assign	9		9	9
10	GND	10	Gray	10	10
11	GND	11	Green	11	11
12	Do not assign	12		12	12
13	HSYNC	13	Pink	13	13
14	VSYNC	14	Yellow	14	14
15	Do not assign	15		15	15
Housing	External shield	Housing	External shield	Housing	Housing

X43: Visual display unit BC 110B

Pin layout for the LE 426 CB/PB, LE 430 CA/PA with Id. Nr. xxx xxx-3x and an LE 426 M, LE 430 M, the connecting cable, and the visual display unit:

Logic unit Id. Nr. xxx xxx-3x		**VB 250 477-xx**			**BC 110B**
D-sub connctn. (female) 15-pin 2-row	**Assignment**	**D-sub cnnctr. (male) 15-pin 2-row**		**D-sub connctr. (female) 15-pin 2-row**	**D-sub cnnctr. (male) 15-pin 2-row**
1	GND	1		1	1
2	Do not assign	2		2	2
3	Do not assign	3		3	3
4	Do not assign	4		4	4
5	Do not assign	5		5	5
6	Do not assign	6		6	6
7	R	7	Coax red	7	7
8	Do not assign	8		8	8
9	HSYNC	9	Yellow	9	9
10	VSYNC	10	Pink	10	10
11	GND	11	Black	11	11
12	Do not assign	12		12	12
13	Do not assign	13		13	13
14	G	14	Coax green	14	14
15	B	15	Coax blue	15	15
Housing	External shield	Housing	External shield	Housing	Housing

Note

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

X43: Visual display unit BC 110B

Pin layout for a logic unit with Id. Nr. xxx xxx-4x, the connecting cable, and the visual display unit:

Logic unit Id. Nr. xxx xxx-4x		**VL 312 878-xx**			**Adapter 313 434-01**	**BC 110B**
D-sub conntn. (female) 15-pin 3-row	**Assignment**	**D-sub conntr. (male) 15-pin 3-row**		**D-sub conntr. (female) 15-pin 3-row**	**3-row/ 2-row**	**D-sub conntr. (male) 15-pin 2-row**
1	R	1	Coax I red	1		1
2	G	2	Coax I green	2		2
3	B	3	Coax I blue	3		3
4	Do not assign	4		4		4
5	Do not assign	5		5		5
6	GND	6	Coax S red	6		6
7	GND	7	Coax S GN	7		7
8	GND	8	Coax S blue	8		8
9	Do not assign	9		9		9
10	GND	10	Gray	10		10
11	GND	11	Green	11		11
12	Do not assign	12		12		12
13	HSYNC	13	Pink	13		13
14	VSYNC	14	Yellow	14		14
15	Do not assign	15		15		15
Housing	External shield	Housing	External shield	Housing	Housing	Housing

Note

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

X49: BF 120 visual display unit

Pin layout for the logic unit, the connecting cable, and the visual display unit:

Note

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

Logic unit		VB 312 876-xx			VB 312 875-xx			BF 120
D-sub cnnctr. (female) 62-pin	**Assign-ment**	**D-sub cnnctr. (male) 62-pin**		**D-sub cnnctr. (female) 62-pin**	**D-sub cnnctr. (male) 62-pin**		**D-sub cnnctr. (female) 62-pin**	**D-sub cnnctr. (male) 62-pin**
1	0 V	1	Gray/ Black	1	1	Gray/ Black	1	1
2	CLK.P	2	Brown/ Black	2	2	Brown/ Black	2	2
3	HSYNC	3	Green/ Black	3	3	Green/ Black	3	3
4	$\overline{\text{BLANK}}$	4	Orange/ Black	4	4	Orange/ Black	4	4
5	VSYNC	5	Blue/ Black	5	5	Blue/ Black	5	5
6	0 V	6	Green/ White	6	6	Green/ White	6	6
7	R0	7	Orange/ White	7	7	Orange/ White	7	7
8	R1	8	Brown/ White	8	8	Brown/ White	8	8
9	R2	9	Gray/ White	9	9	Gray/ White	9	9
10	R3	10	Blue/ White	10	10	Blue/ White	10	10
11	0 V	11	Violet/ White	11	11	Violet/ White	11	11
12	G0	12	Violet/ Brown	12	12	Violet/ Brown	12	12
13	G1	13	Violet/ Green	13	13	Violet/ Green	13	13
14	G2	14	Violet/ Orange	14	14	Violet/ Orange	14	14
15	G3	15	Violet/ Blue	15	15	Violet/ Blue	15	15
16	0 V	16	Red/Gray	16	16	Red/Gray	16	16
17	B0	17	Red/ Brown	17	17	Red/ Brown	17	17
18	B1	18	Yellow/ Gray	18	18	Yellow/ Gray	18	18
19	B2	19	Yellow/ Brown	19	19	Yellow/ Brown	19	19

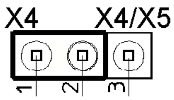
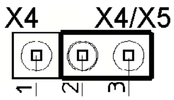
Logic unit		VB 312 876-xx			VB 312 875-xx			BF 120
D-sub cnnctr. (female) 62-pin	Assign-ment	D-sub cnnctr. (male) 62-pin		D-sub cnnctr. (female) 62-pin	D-sub cnnctr. (male) 62-pin		D-sub cnnctr. (female) 62-pin	D-sub cnnctr. (male) 62-pin
20	B3	20	Yellow/ Green	20	20	Yellow/ Green	20	20
21	0 V	21	Free	21	21	Free	21	21
22	0 V	22	Black/ Gray	22	22	Black/ Gray	22	22
23	$\overline{\text{CLP.P}}$	23	Black/ Brown	23	23	Black/ Brown	23	23
24	$\overline{\text{HSYNC}}$	24	Black/ Green	24	24	Black/ Green	24	24
25	BLANK	25	Black/ Orange	25	25	Black/ Orange	25	25
26	$\overline{\text{VSYNC}}$	26	Black/ Blue	26	26	Black/ Blue	26	26
27	0 V	27	White/ Green	27	27	White/ Green	27	27
28	$\overline{\text{R0}}$	28	White/ Orange	28	28	White/ Orange	28	28
29	$\overline{\text{R1}}$	29	White/ Brown	29	29	White/ Brown	29	29
30	$\overline{\text{R2}}$	30	White/ Gray	30	30	White/ Gray	30	30
31	$\overline{\text{R3}}$	31	White/ Blue	31	31	White/ Blue	31	31
32	0 V	32	Gray/ Violet	32	32	Gray/ Violet	32	32
33	$\overline{\text{G0}}$	33	Brown/ Violet	33	33	Brown/ Violet	33	33
34	$\overline{\text{G1}}$	34	Green/ Violet	34	34	Green/ Violet	34	34
35	$\overline{\text{G2}}$	35	Orange/ Violet	35	35	Orange/ Violet	35	35
36	$\overline{\text{G3}}$	36	Blue/ Violet	36	36	Blue/ Violet	36	36
37	0 V	37	Gray/Red	37	37	Gray/Red	37	37
38	$\overline{\text{B0}}$	38	Brown/ Red	38	38	Brown/ Red	38	38
39	$\overline{\text{B1}}$	39	Gray/ Yellow	39	39	Gray/ Yellow	39	39
40	$\overline{\text{B2}}$	40	Brown/ Yellow	40	40	Brown/ Yellow	40	40
41	$\overline{\text{B3}}$	41	Green/ Yellow	41	41	Green/ Yellow	41	41
42	0 V	42	Free	42	42	Free	42	42

Logic unit		VB 312 876-xx			VB 312 875-xx			BF 120
D-sub cnnctr. (female) 62-pin	**Assign-ment**	**D-sub cnnctr. (male) 62-pin**		**D-sub cnnctr. (female) 62-pin**	**D-sub cnnctr. (male) 62-pin**		**D-sub cnnctr. (female) 62-pin**	**D-sub cnnctr. (male) 62-pin**
43	$\overline{\text{DISP. LOW}}$	43	Red/Blue	43	43	Red/Blue	43	43
44	DISP. LOW	44	Blue/Red	44	44	Blue/Red	44	44
45	$\overline{\text{DISP.ON}}$	45	Red/ Orange	45	45	Red/ Orange	45	45
46	DISP.ON	46	Orange/ Red	46	46	Orange/ Red	46	46
47	C0	47	Green/ Red	47	47	Green/ Red	47	47
48	C1	48	Red/ Green	48	48	Red/ Green	48	48
49	C2	49	Orange/ Yellow	49	49	Orange/ Yellow	49	49
50	C3	50	Yellow/ Orange	50	50	Yellow/ Orange	50	50
51	C4	51	Yellow/ Blue	51	51	Yellow/ Blue	51	51
52	C5	52	Blue/ Yellow	52	52	Blue/ Yellow	52	52
53 to 56	Do not assign	53 to 56	Free	53 to 56	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	Free	57 to 62	57 to 62
Housing		Housing		Housing	Housing		Housing	Housing

3.20 BTS 1x0 Monitor/Keyboard Switch

Two monitors (BTS 110: 2 x BC 120, BTS 120: 2 x BF 120) and two TE 420 keyboards can be connected to an LE 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.
With the BTS 110, the potentiometers of the current keyboard are active. With the BTS 120, a jumper on the PCB is used to determine which potentiometers 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

X1, X2, X4, X5 to X7: Monitor and keyboard connections

Refer to the Sections "TNC Operating Panel" and "Visual Display Unit" for the pin layouts of the individual connections.

Connection designation	Monitor/Keyboard
X1	Input BC 120 or BF 120
X2	Input TE 420
X4	1st output TE 420
X5	2nd output TE 420
X6	1st output BC 120 or BF 120
X7	2nd output BC 120 or BF 120

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

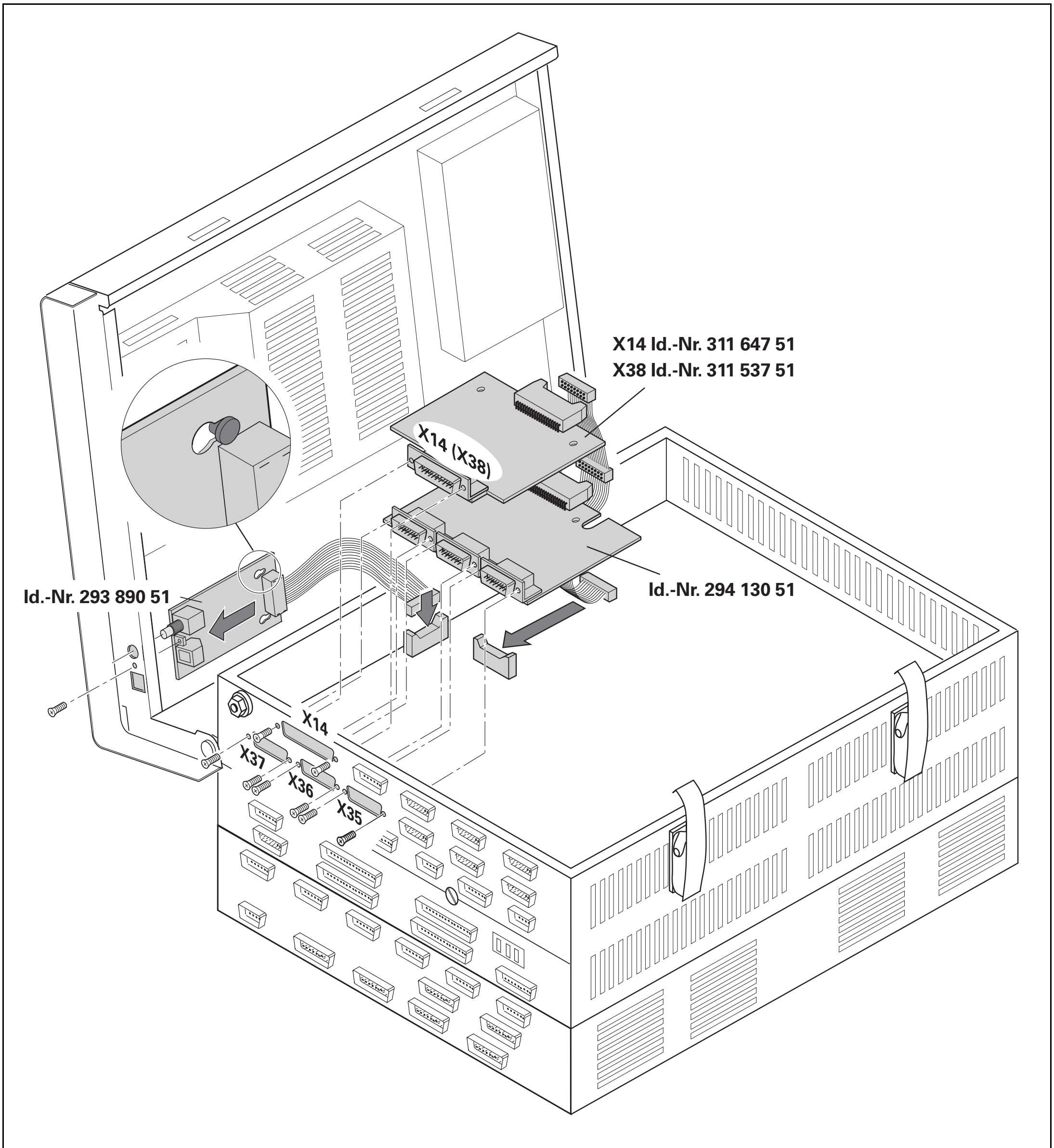
X8: Supply voltage for BTS 120

Pin layout on the BTS 120:

Connecting terminal	Assignment
1	+24 V
2	0 V

3.21 Mounting the Optional PCBs in the LE 426 CB/PB, LE 430 CA/PA

HEIDENHAIN supplies the optional PCBs separately.



Danger

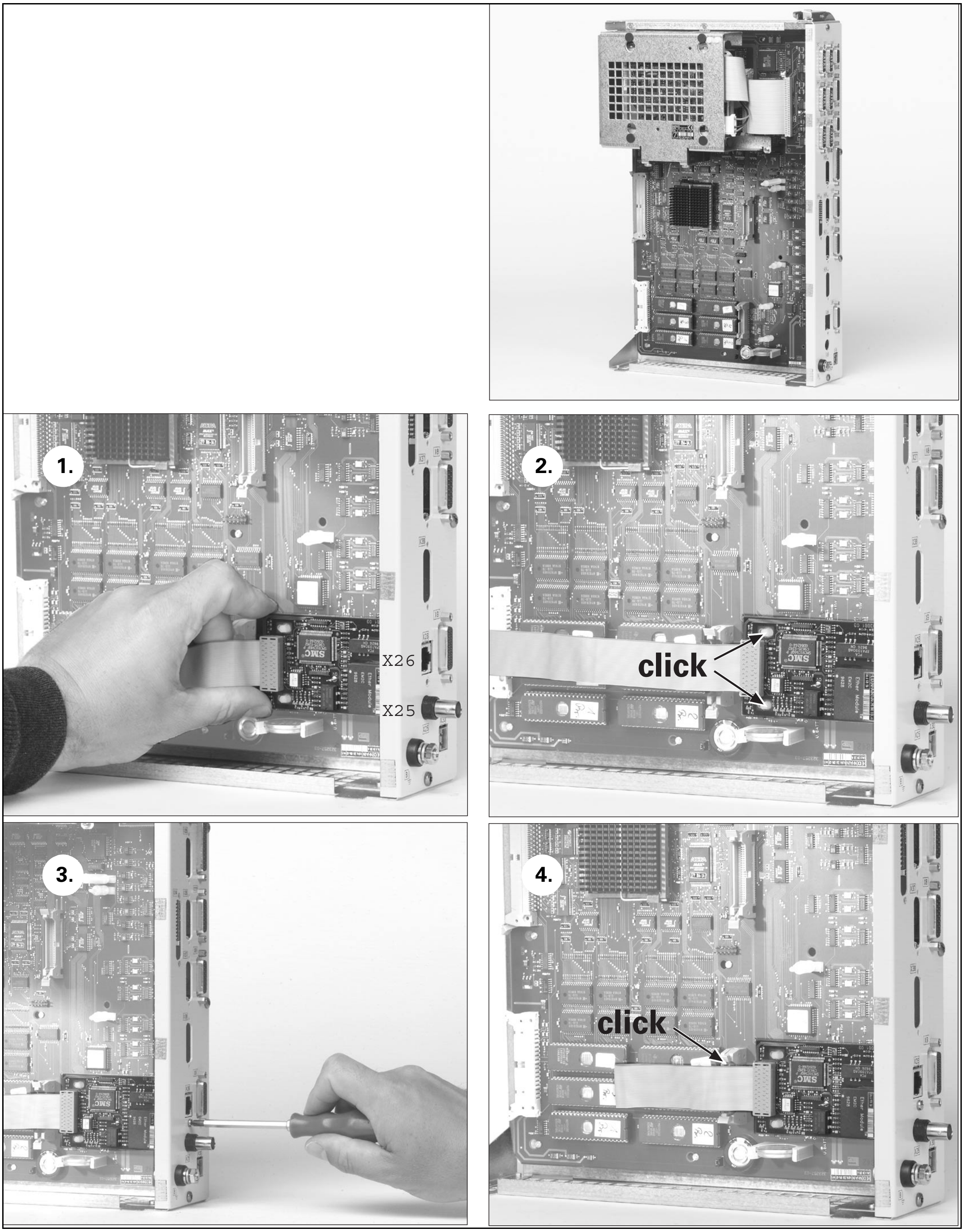
The installation must be performed only by trained personnel.

3.22 Mounting the Optional PCBs in the LE 426 M, LE 430 M

3.22.1 Ethernet Interface

Warning

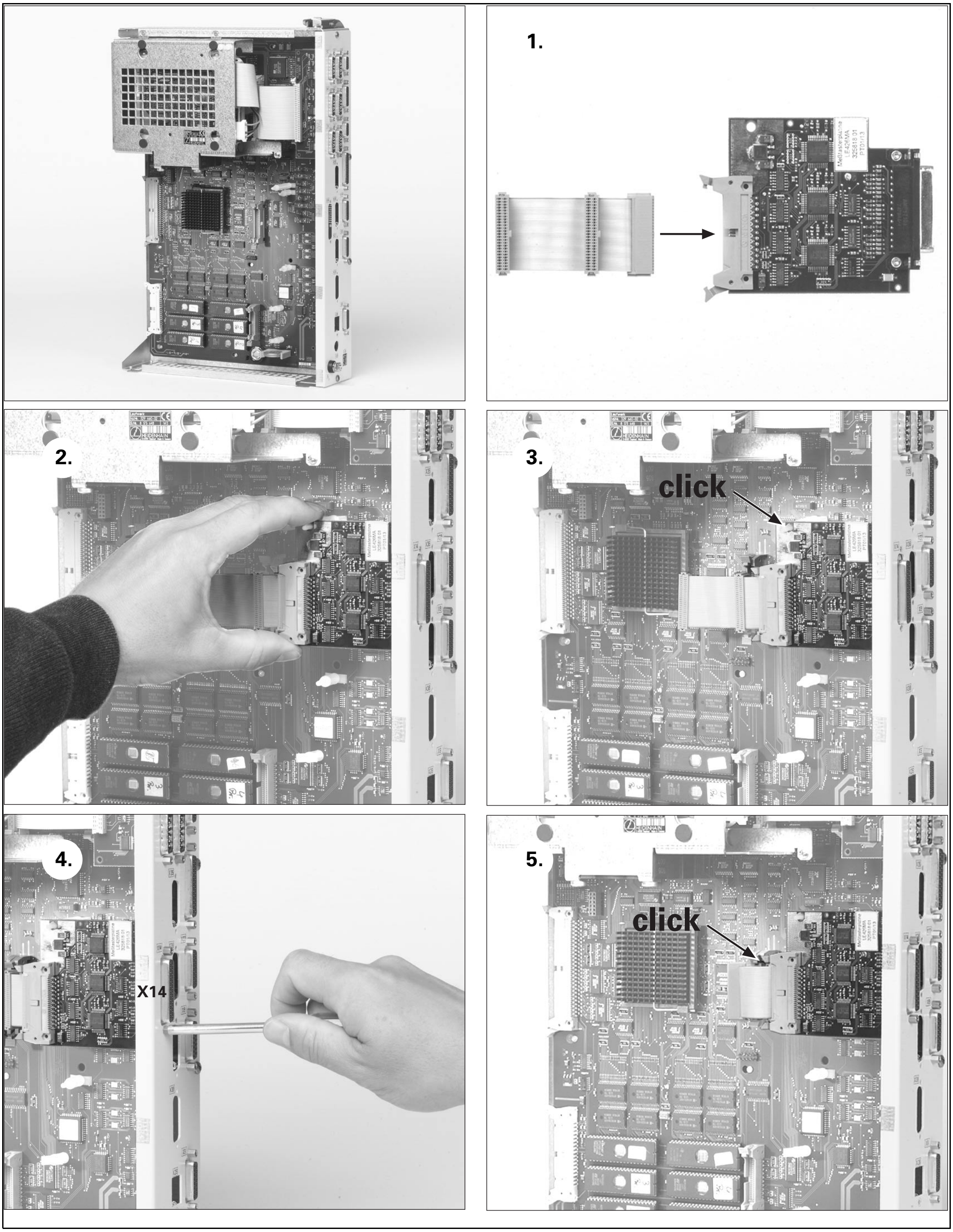
The installation must be performed only by trained personnel.



3.22.2 Digitizing with the Measuring 3-D Touch Probe

Warning

The installation must be performed only by trained personnel.



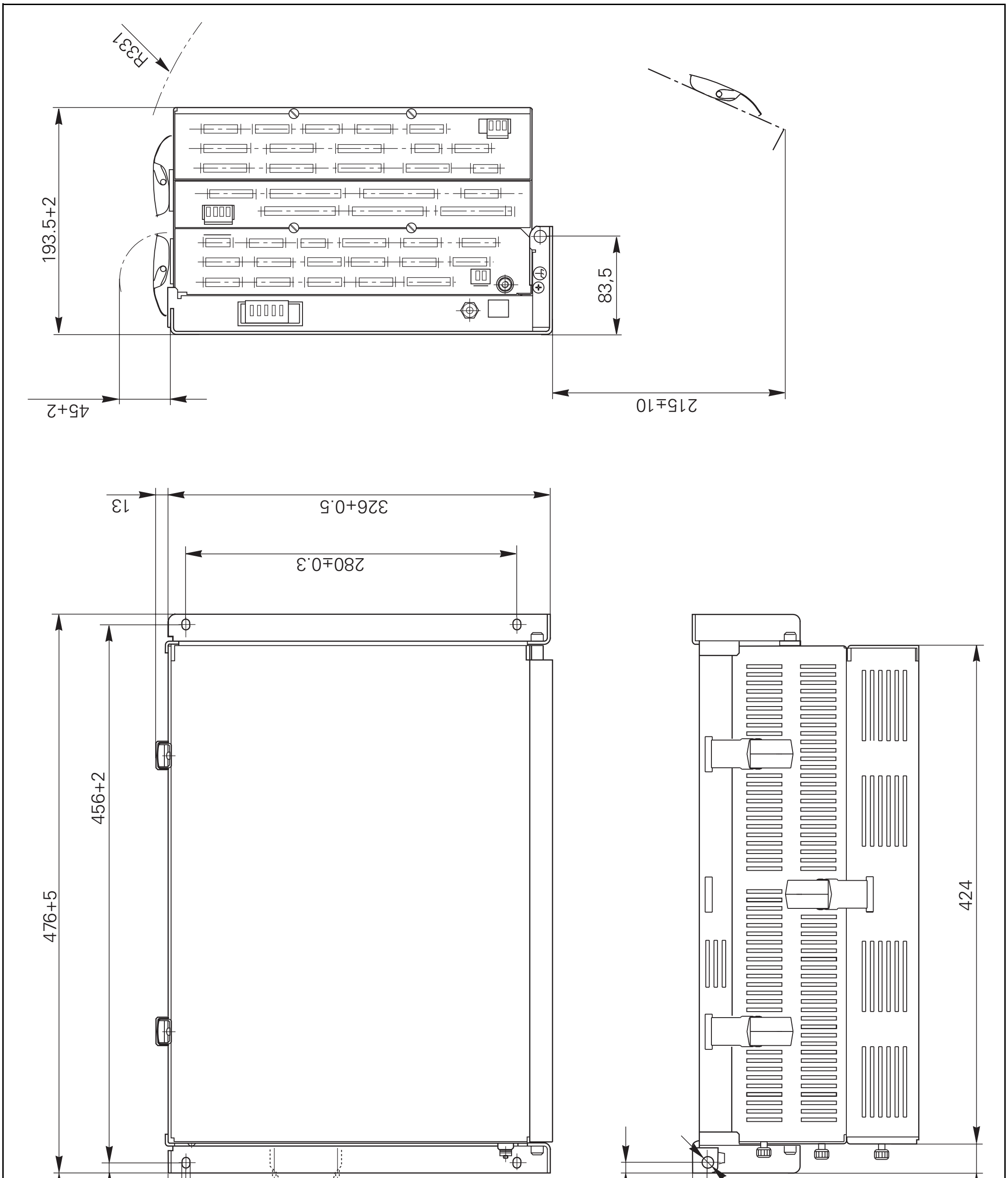
3.23 Dimensions

Note

All dimensions in [mm].

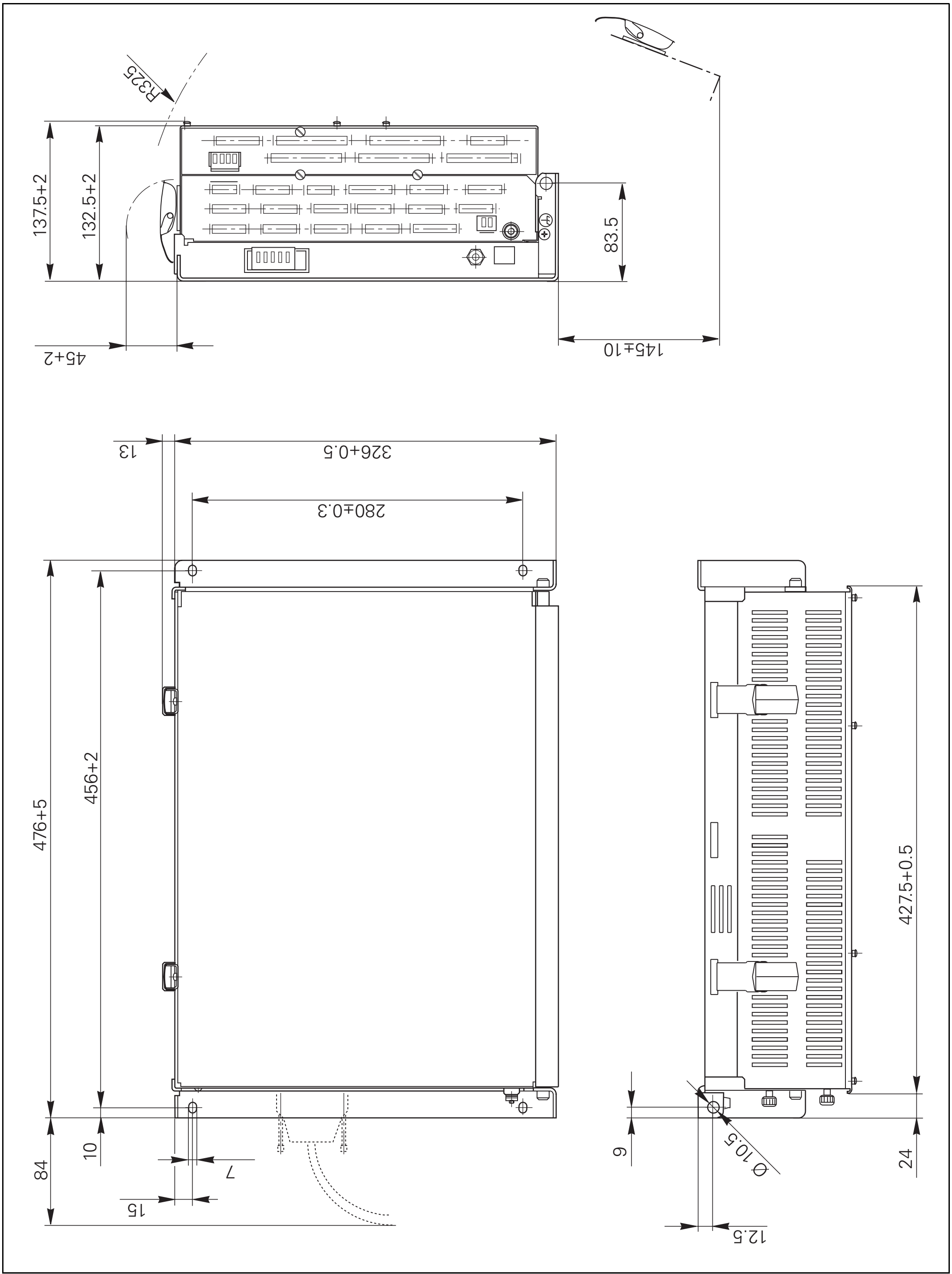
3.23.1 LE 426 PB, LE 430 PA

Weight: 8.8 kg



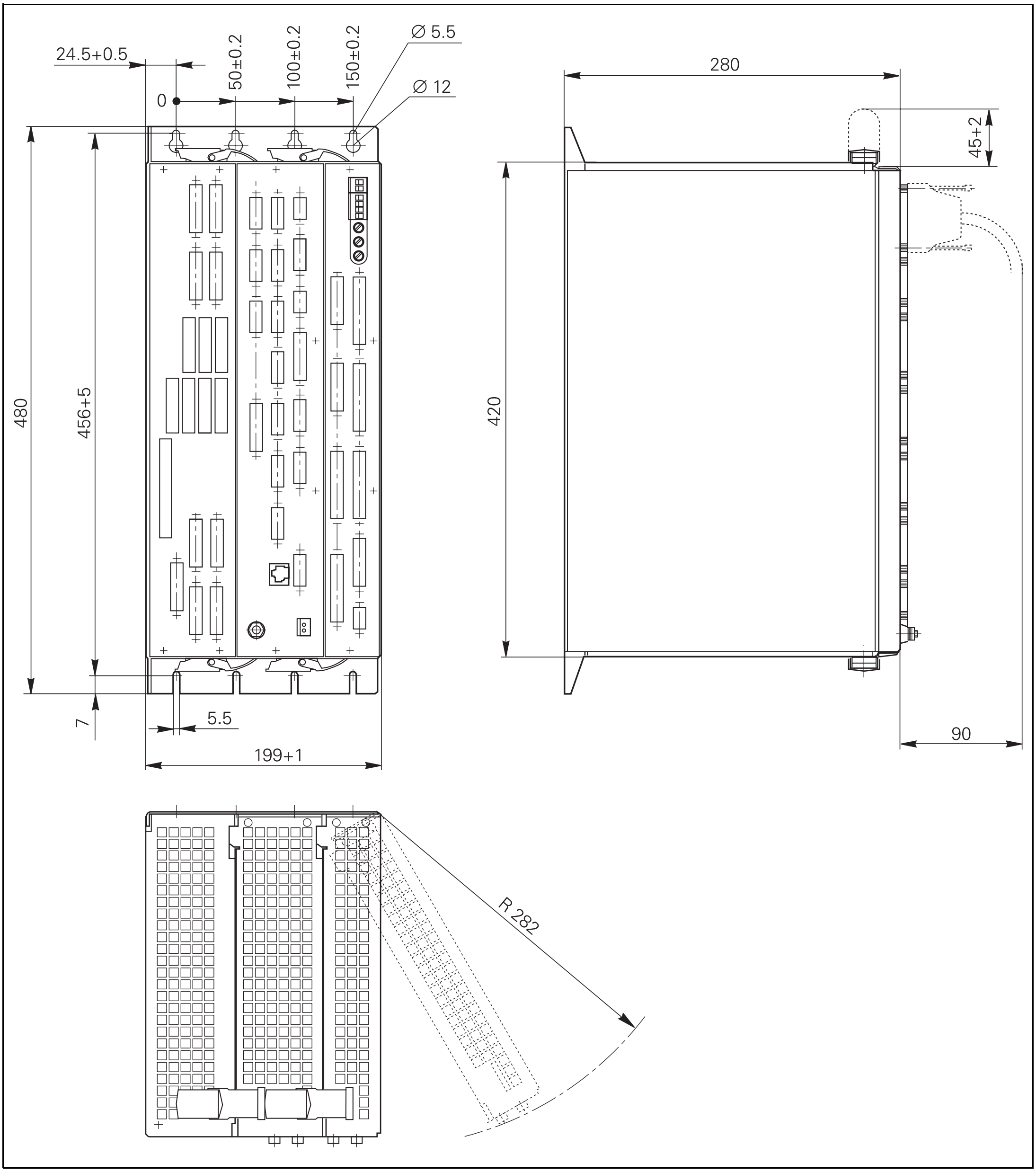
3.23.2 LE 426 CB, LE 430 CA

Weight: 8.8 kg



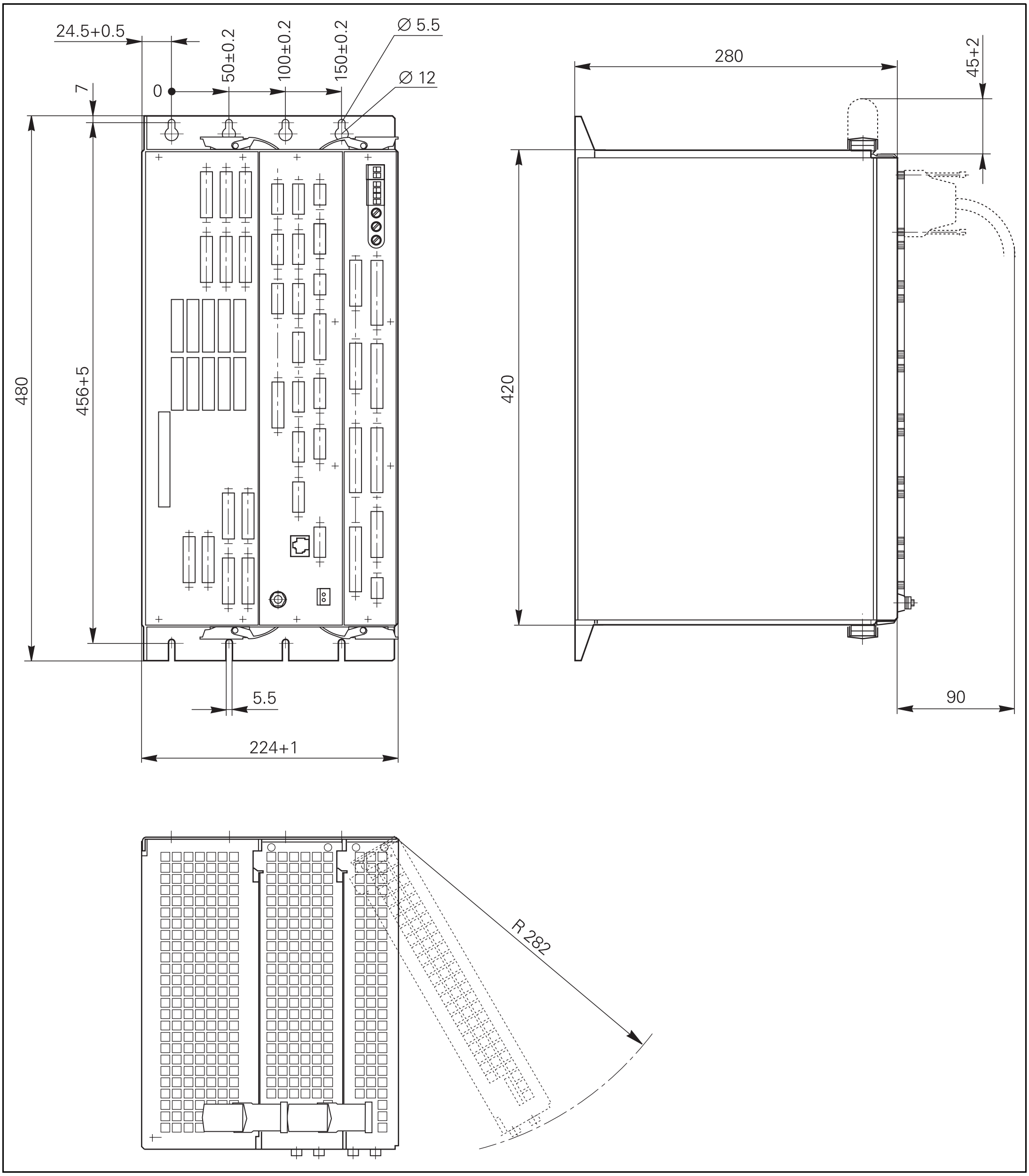
3.23.3 LE 426 M, LE 430 M (Max. 6 Axes)

Weight: approx. 8.5 kg



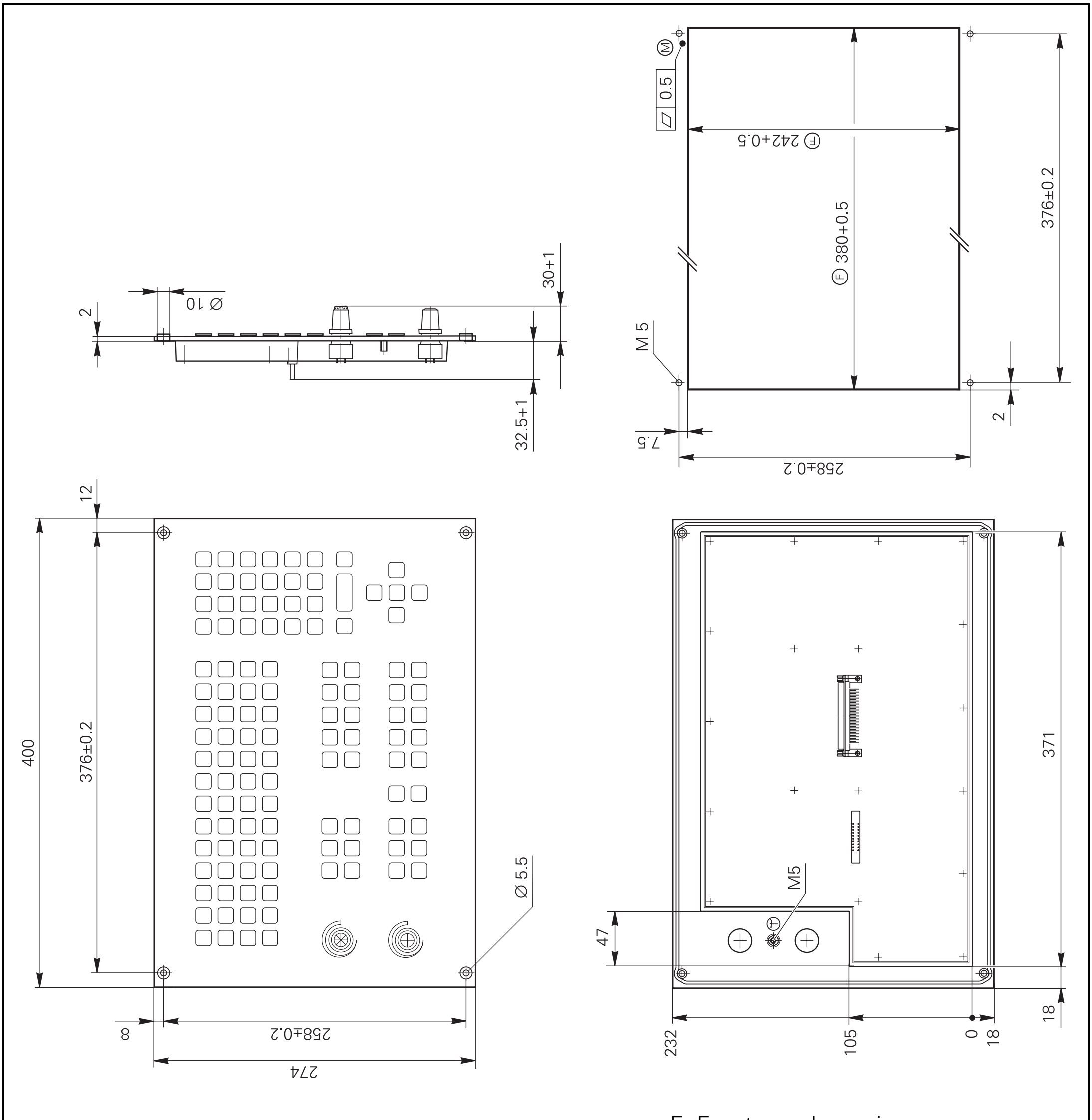
3.23.4 LE 430 M (Max. 9 Axes)

Weight: approx. 9 kg



3.23.5 TE 420

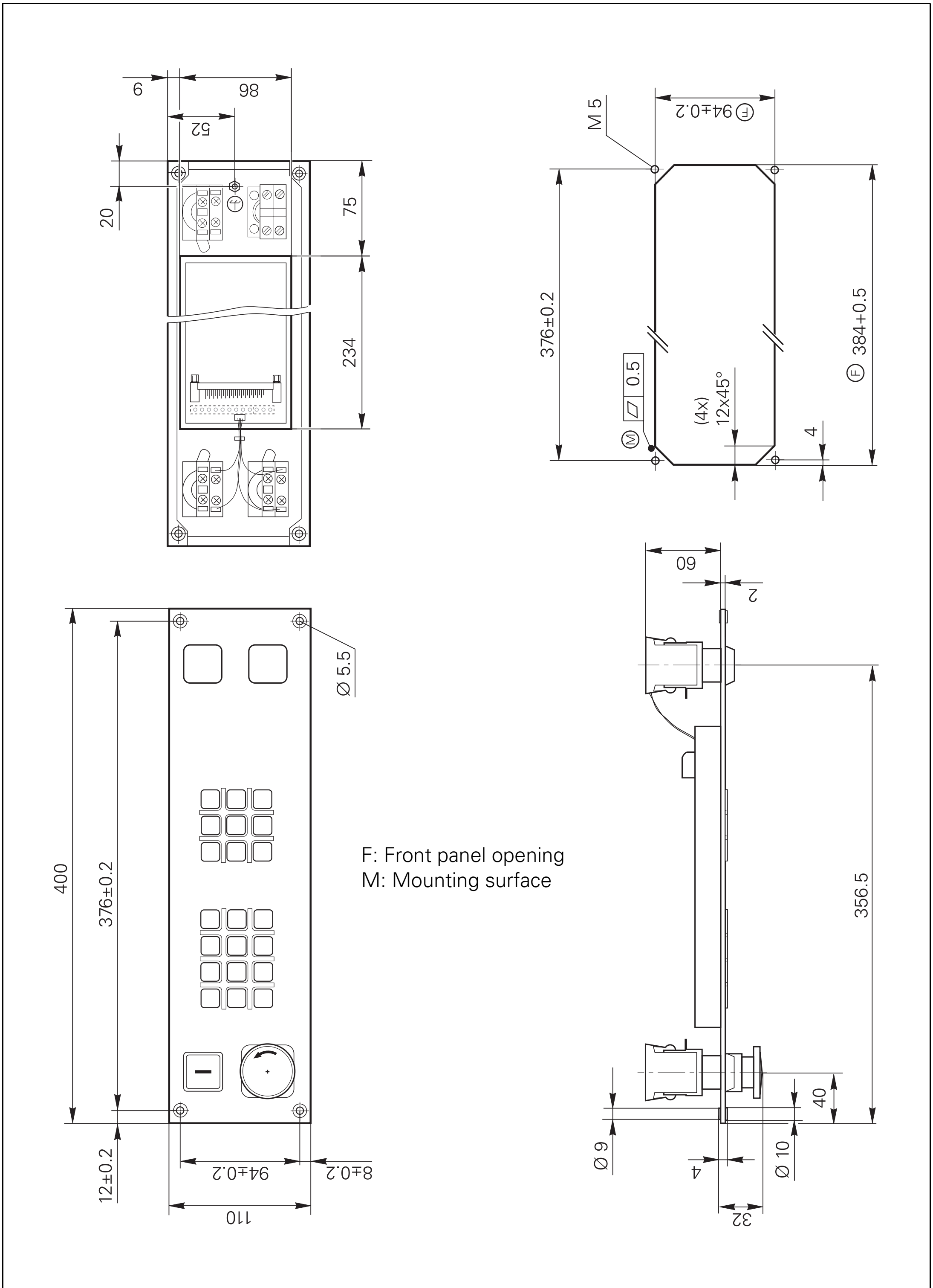
Weight: 2.4 kg



F: Front panel opening
M: Mounting surface

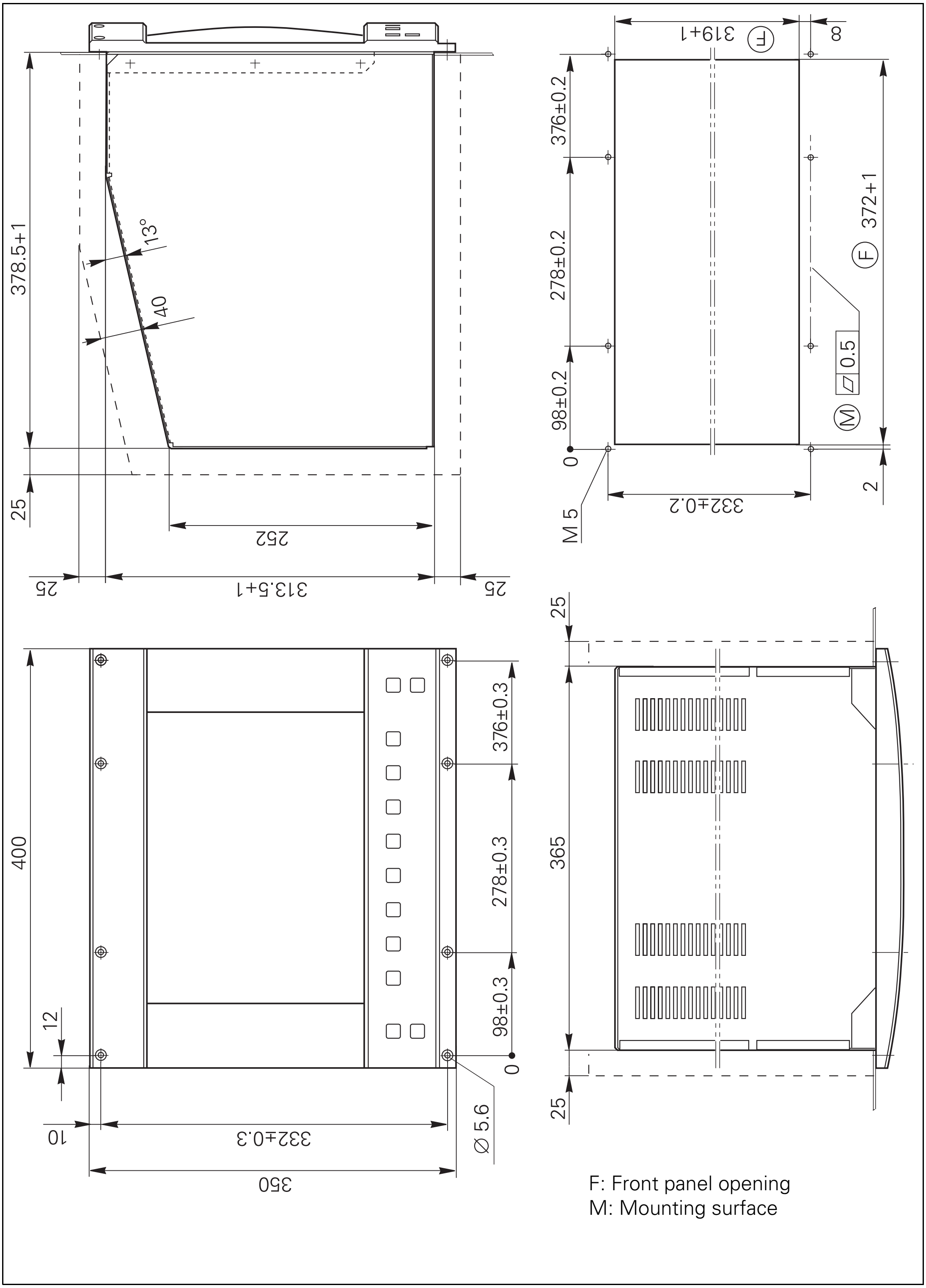
3.23.6 MB 420

Weight: 0.9 kg



3.23.7 BC 120

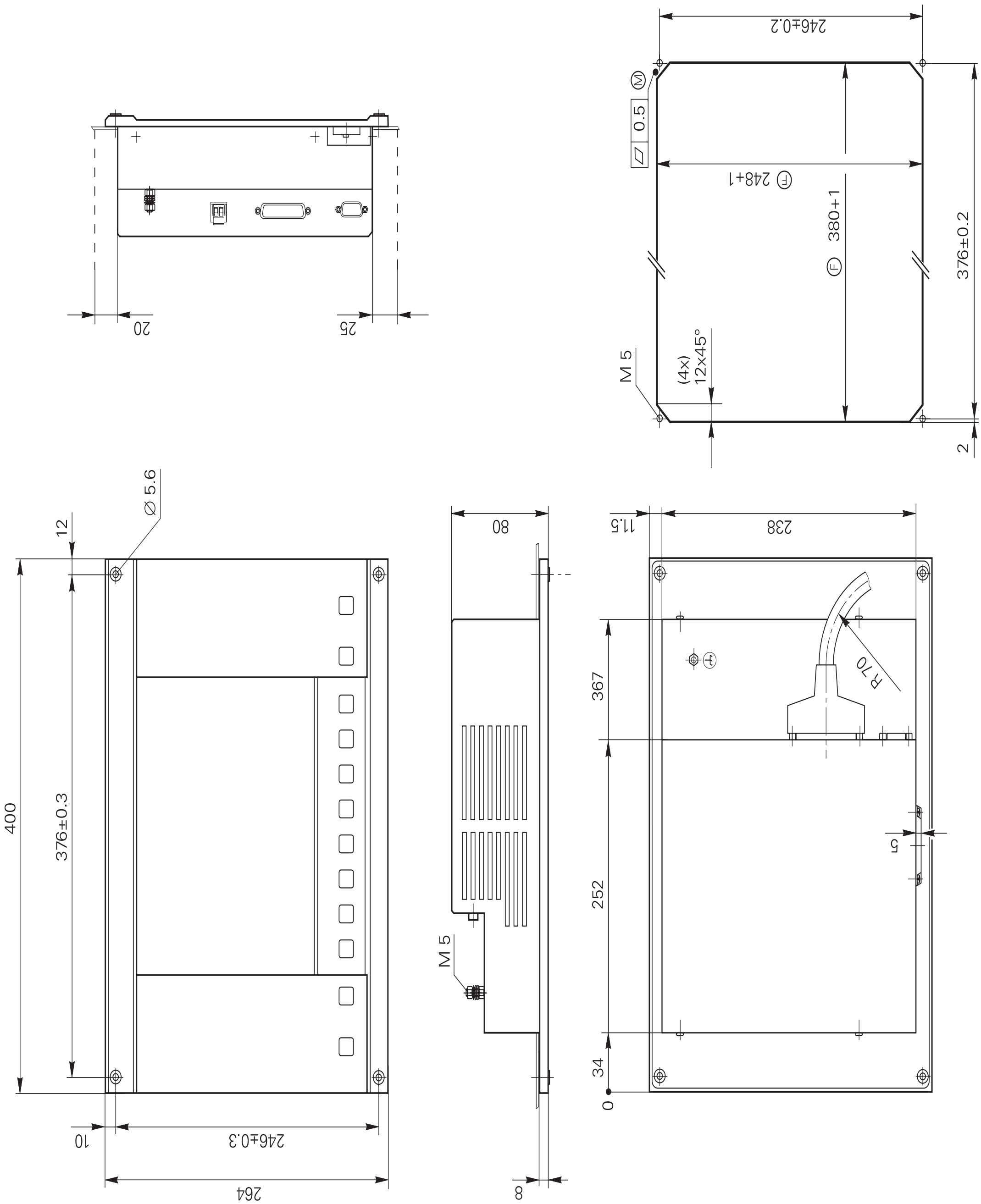
Weight: 14 kg



F: Front panel opening
M: Mounting surface

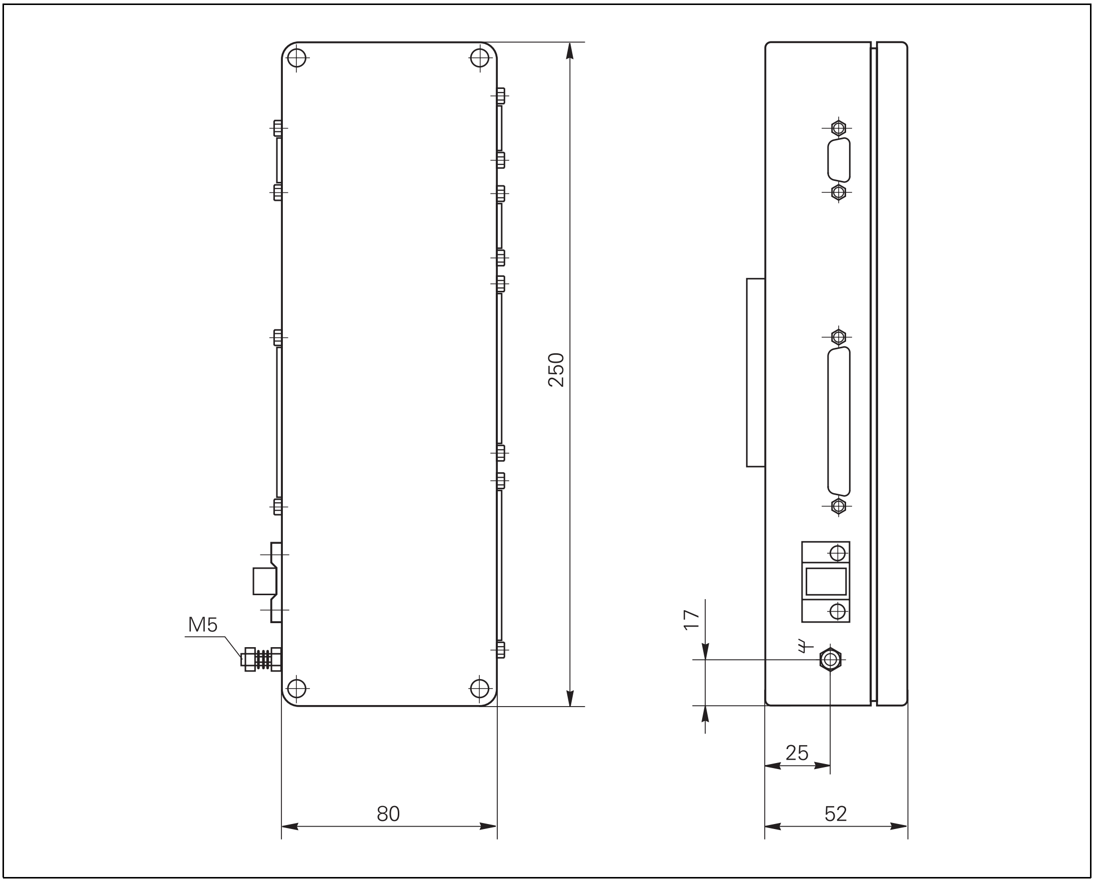
3.23.8 BF 120

Weight: 3 kg

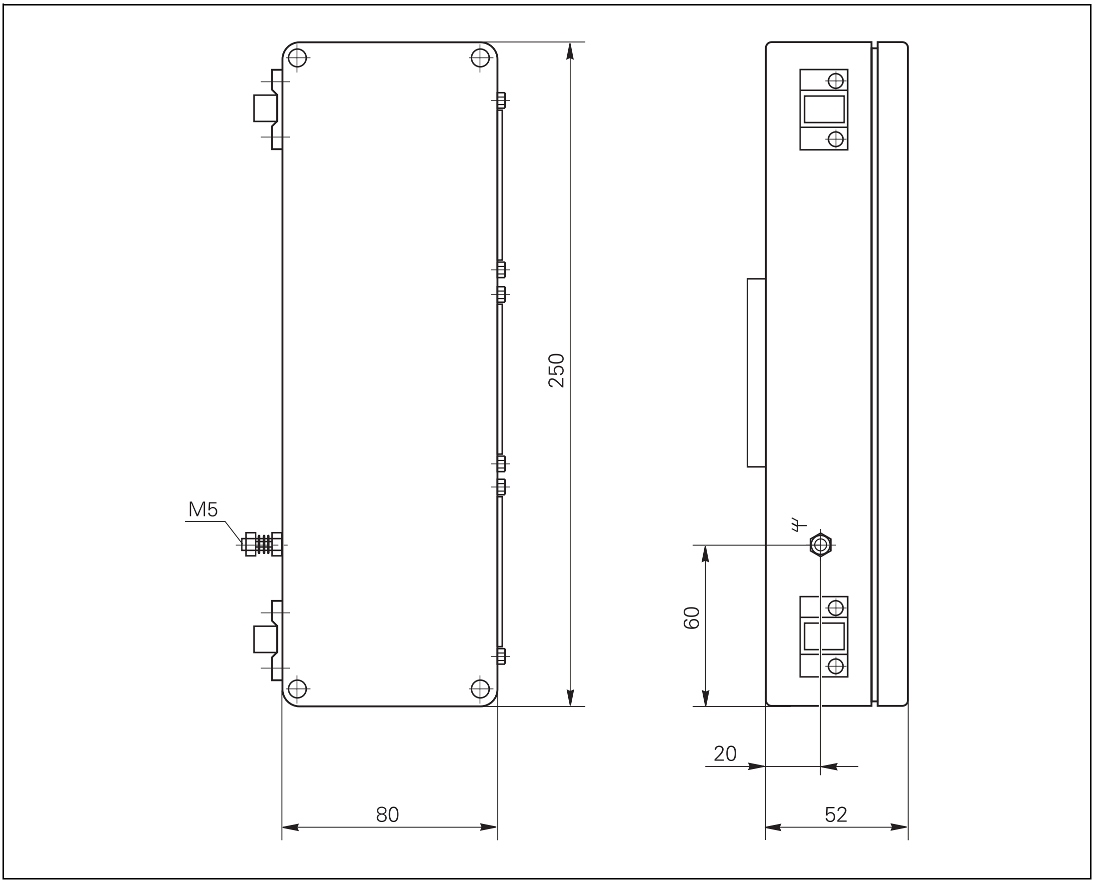


F: Front panel opening
M: Mounting surface

3.23.9 BTS 110

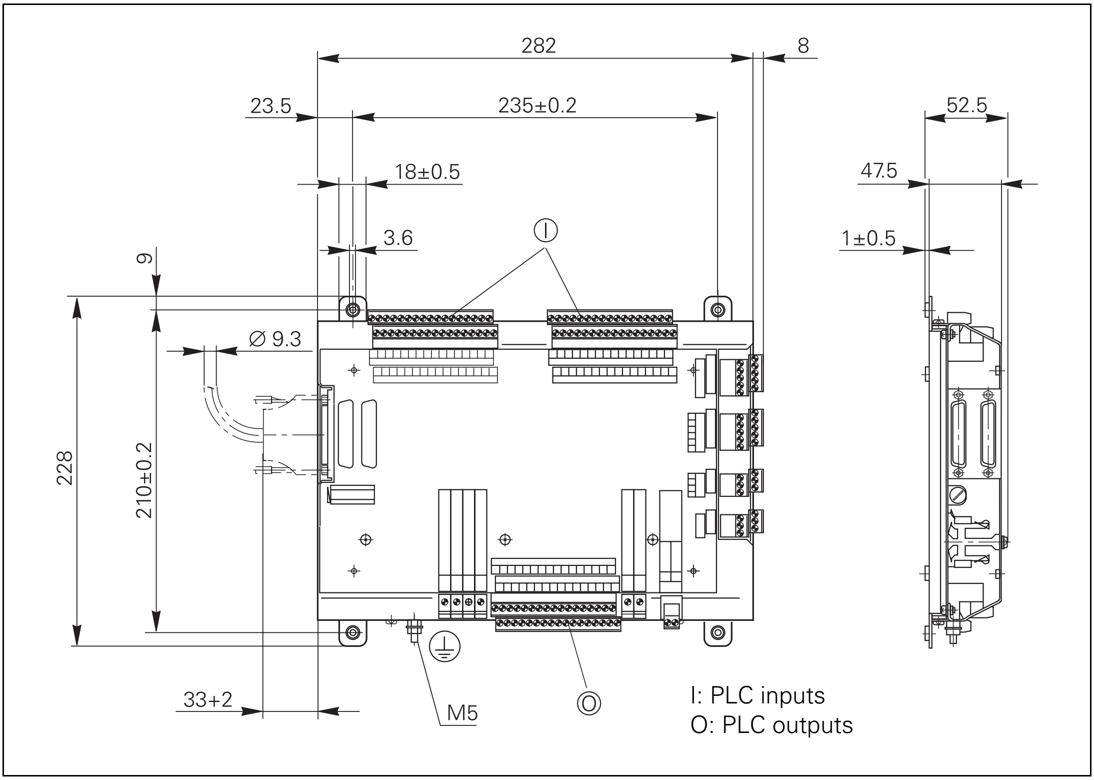


3.23.10 BTS 120



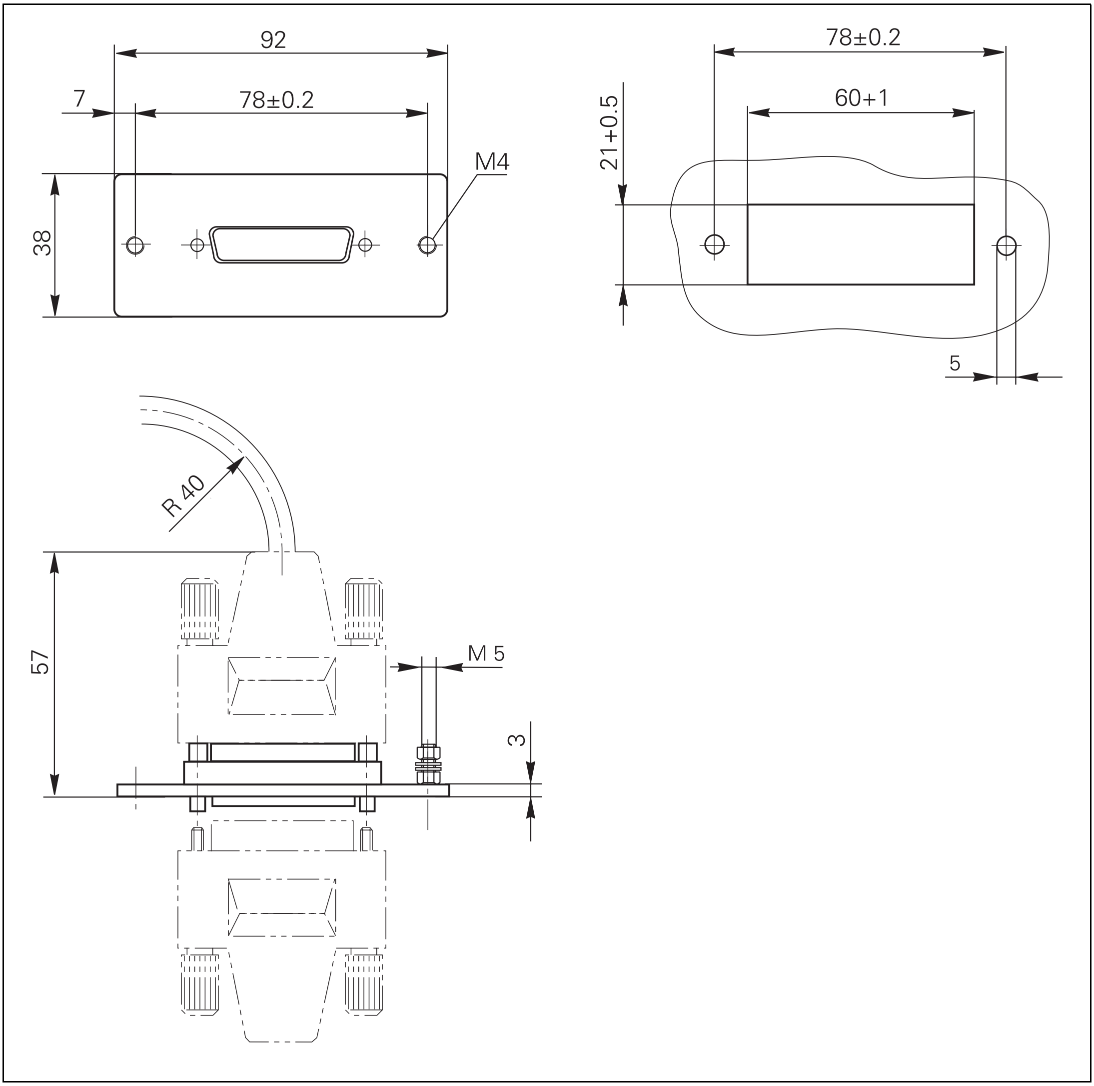
3.23.11 PL 4xx B

Weight: 1.5 kg



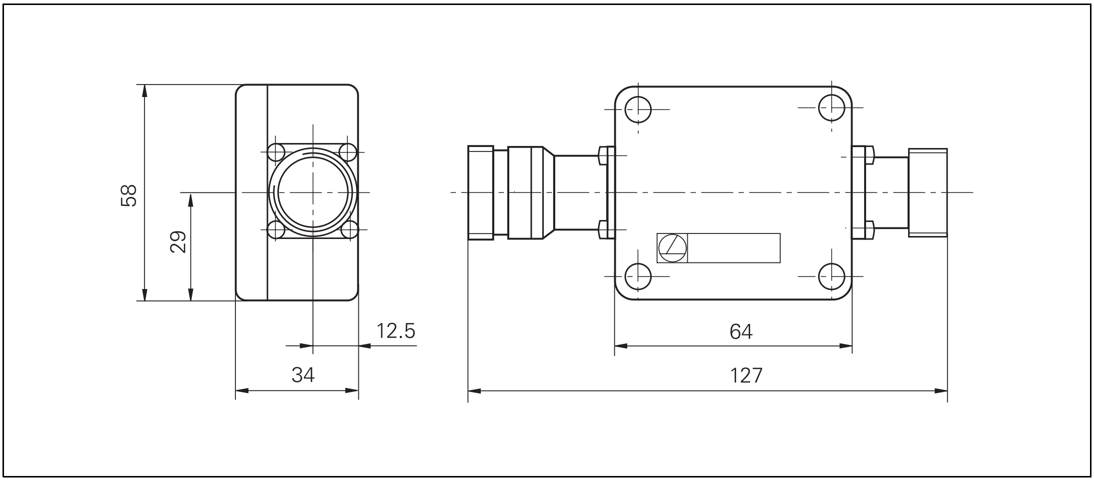
3.23.12 Adapter Block for the Data Interface

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



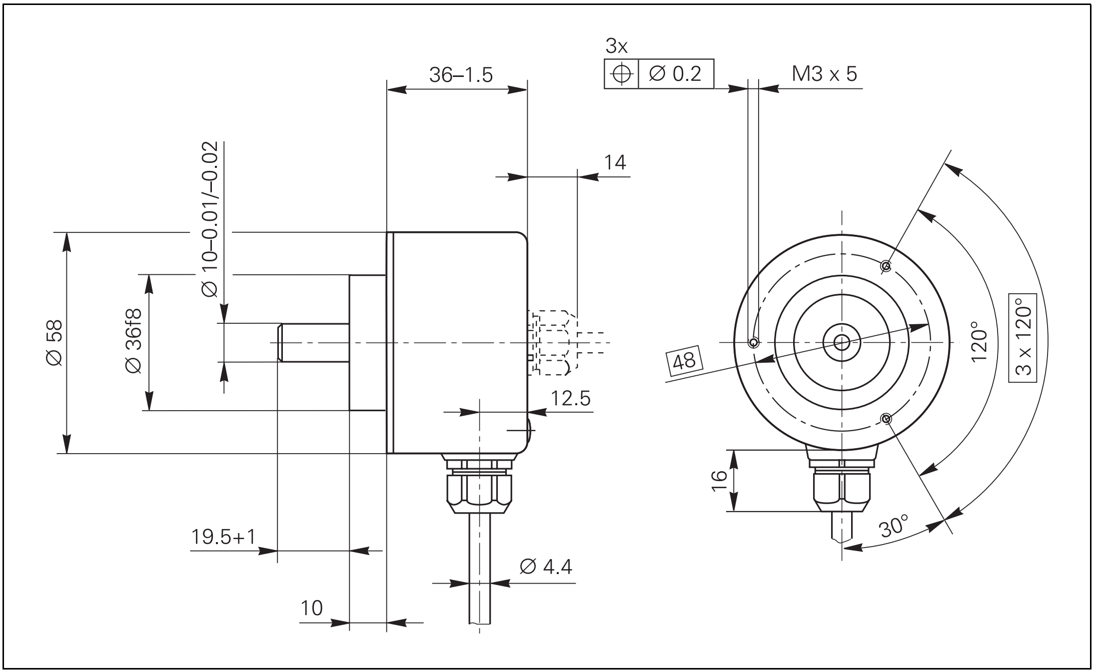
3.23.13 Voltage Controller

Voltage controller for encoders with EnDat interface

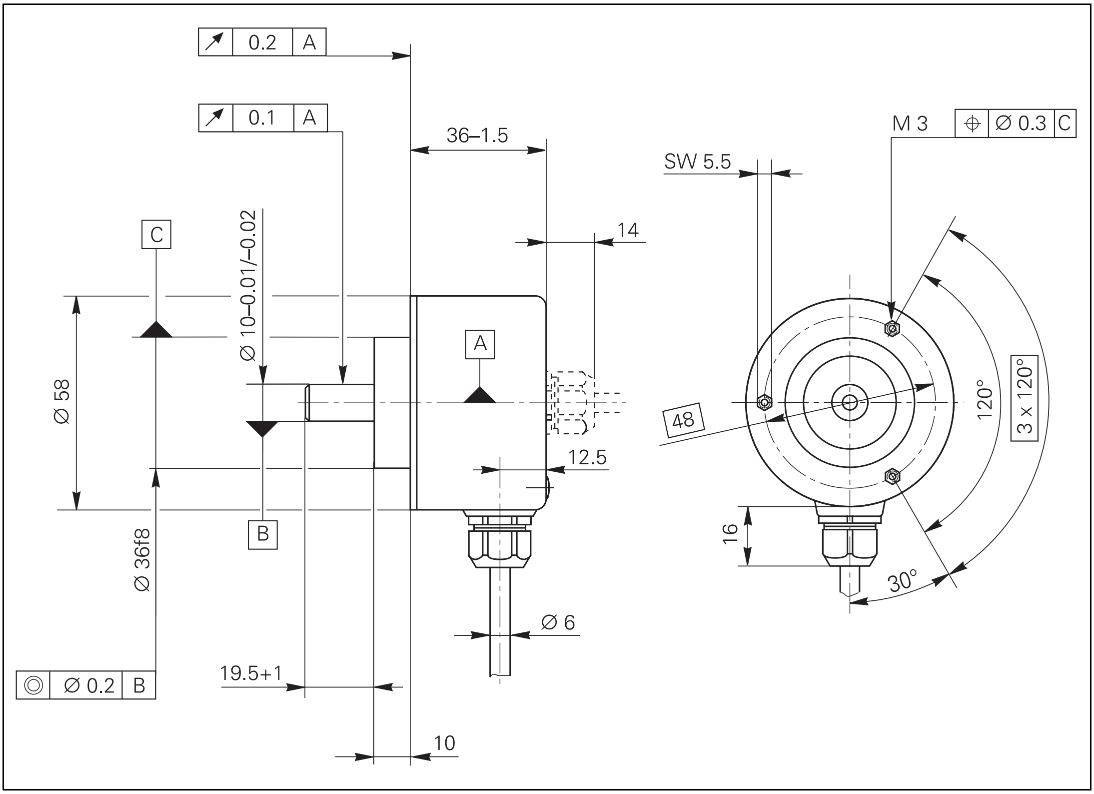


3.23.14 Handwheels

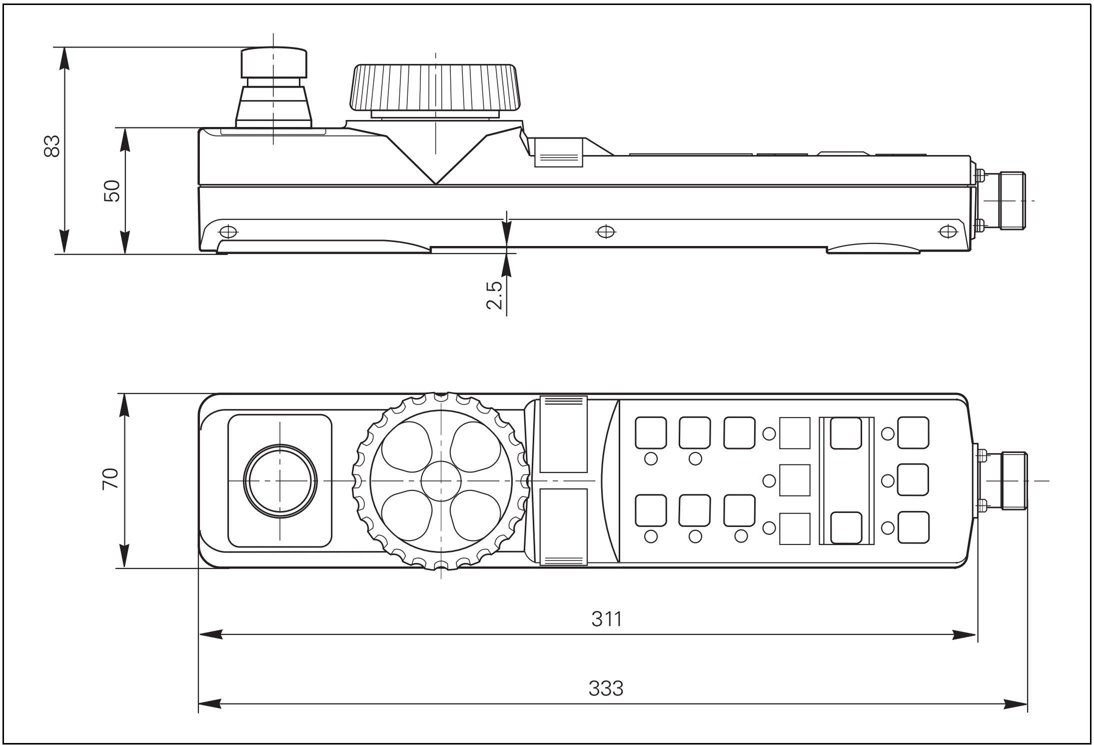
HR 130 Weight: approx. 0.7 kg



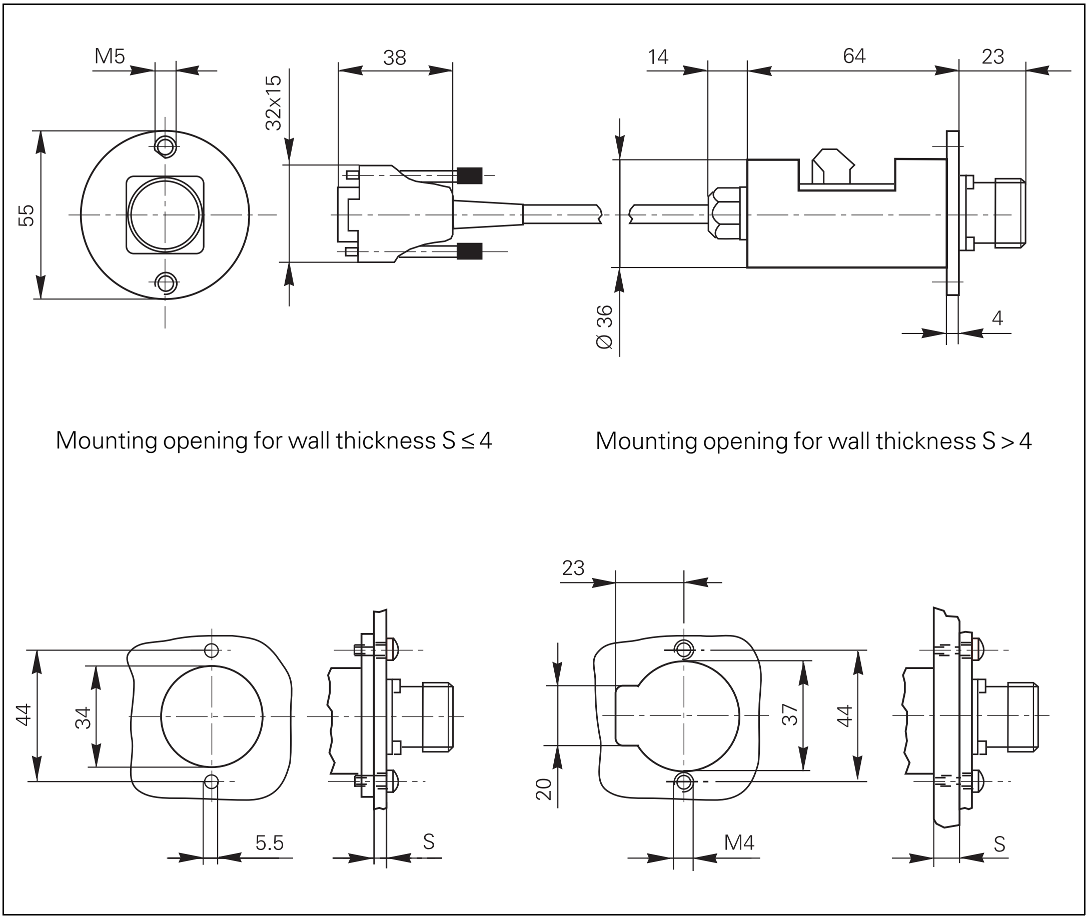
HR 150 Weight: approx. 0.7 kg



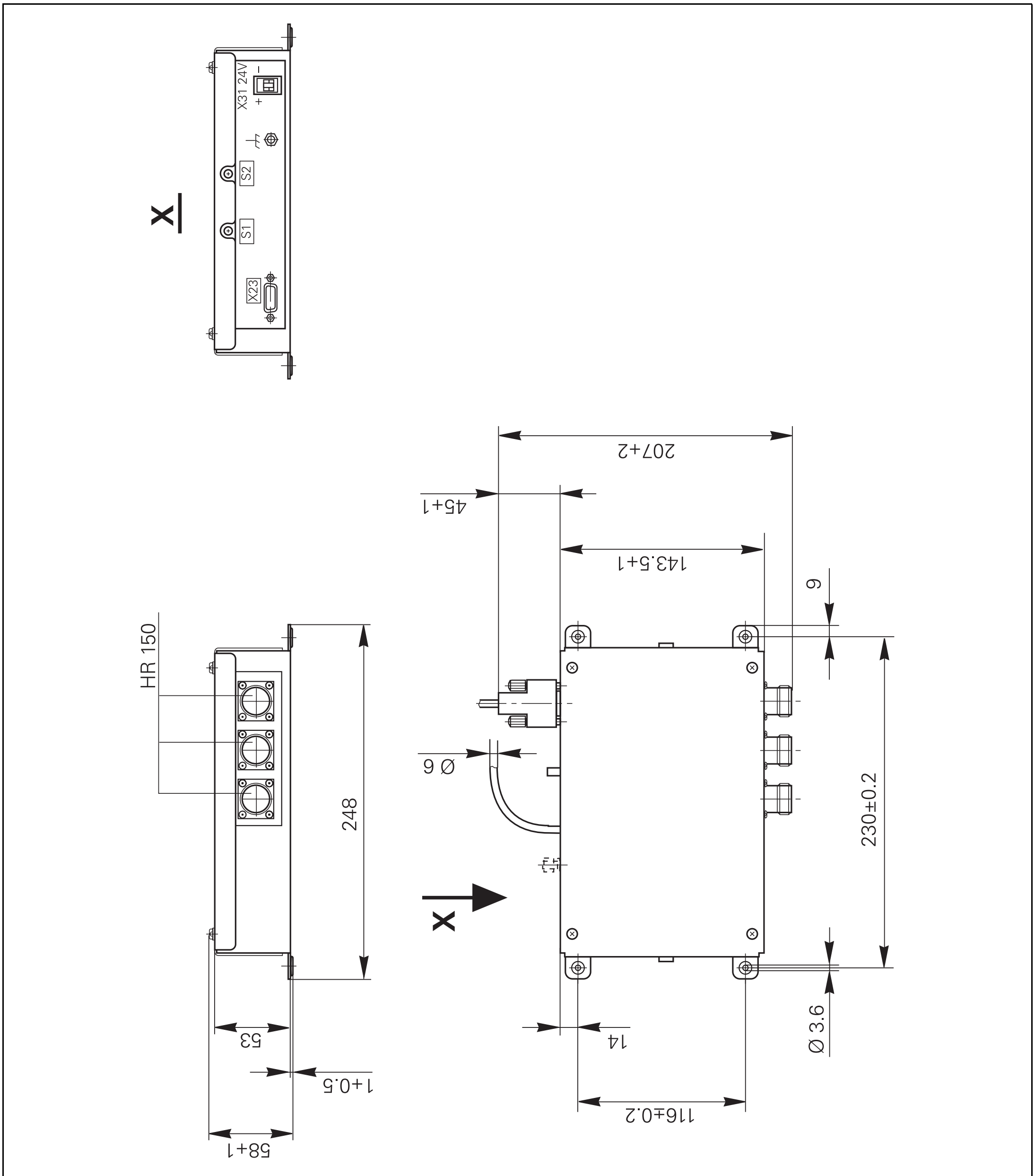
HR 410



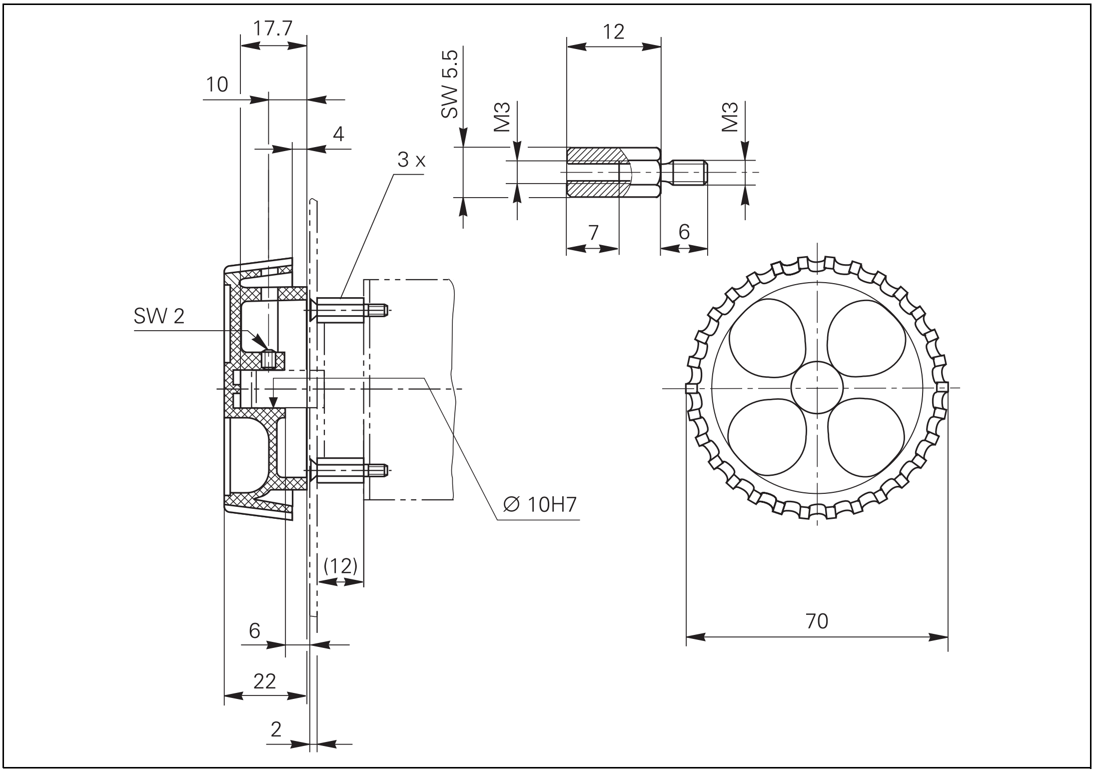
Adapter cables



HRA 110 Weight: approx. 1.5 kg

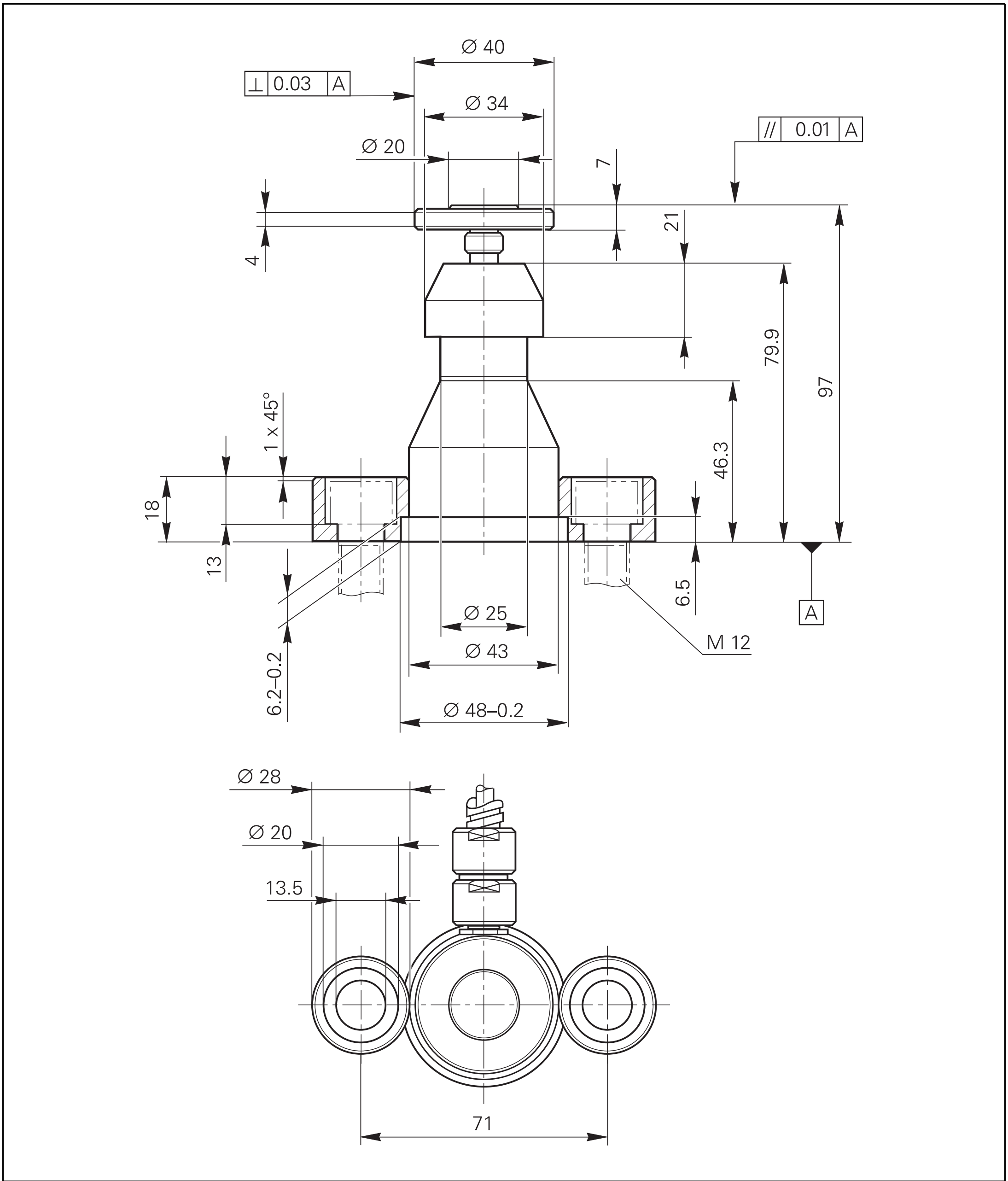


Control knob for HR 130 and HR 150

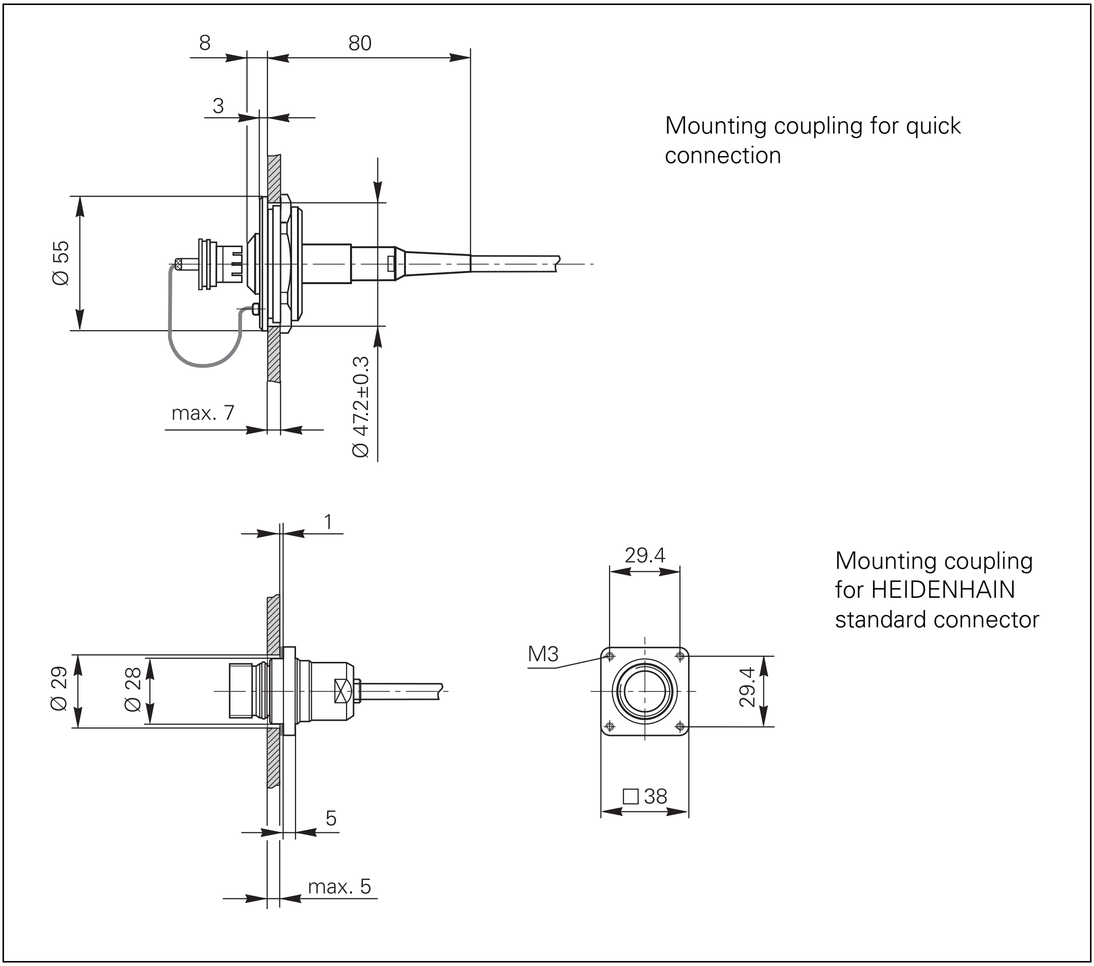


3.23.15 Touch probe systems

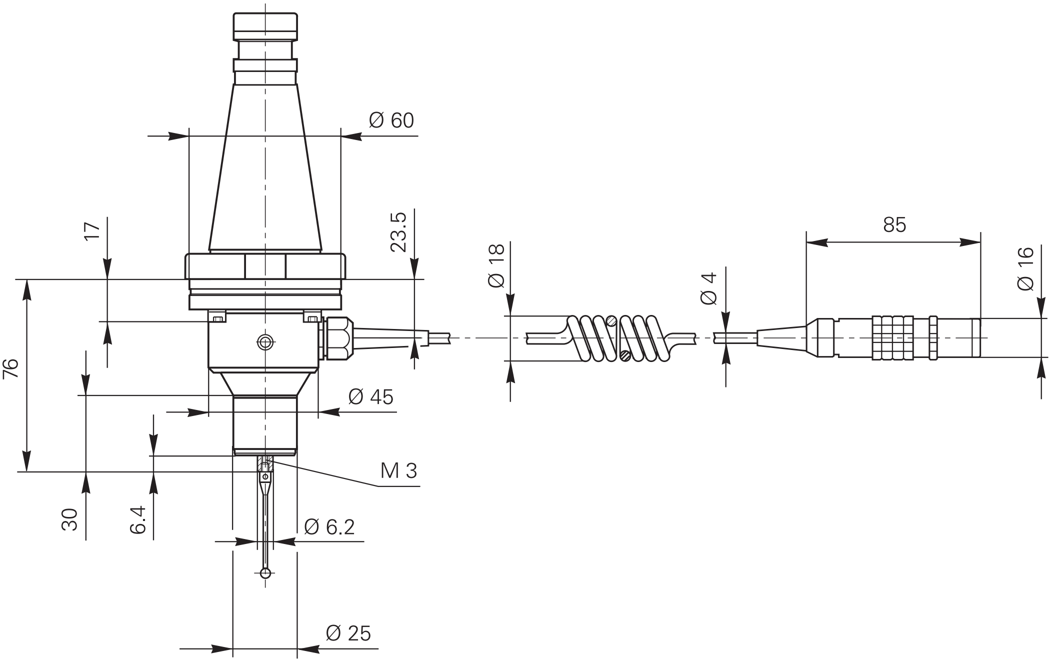
TT 130



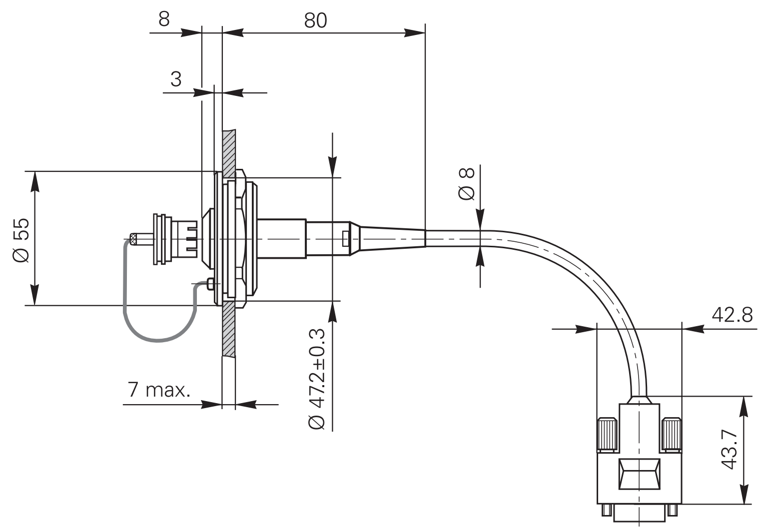
Adapter cable for TT and TS



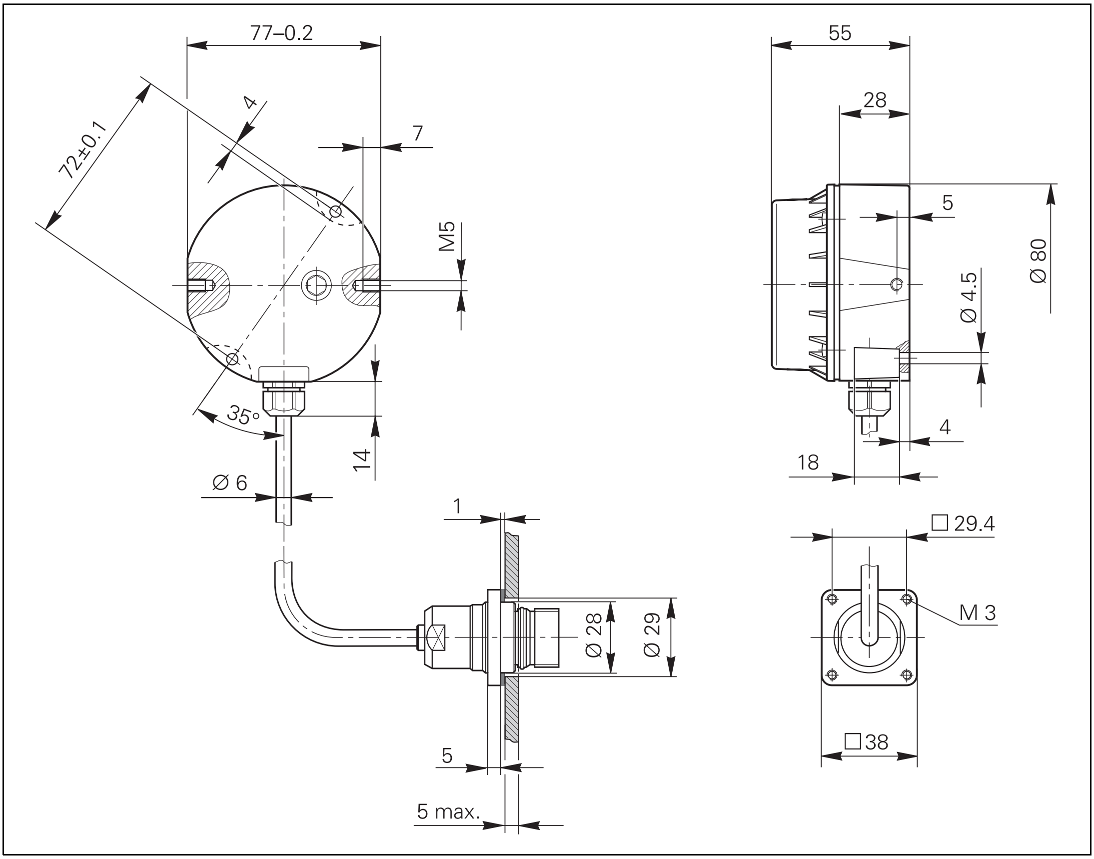
TS 220



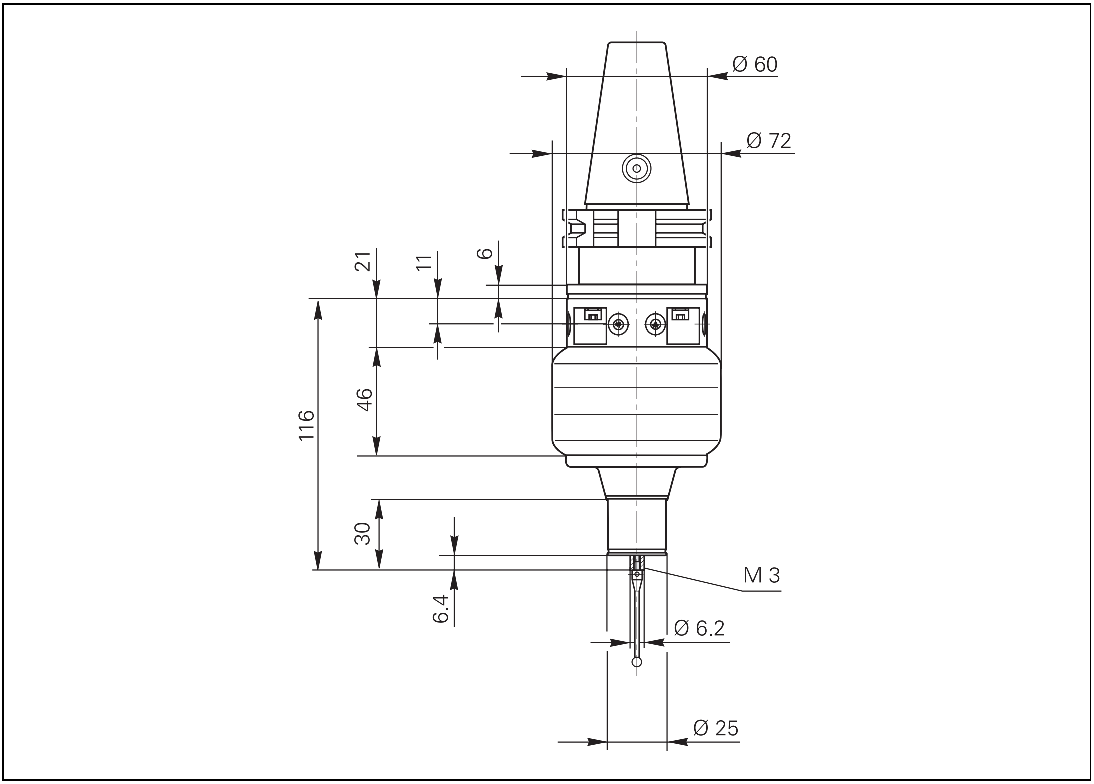
Adapter cable for TS 120/TS 220



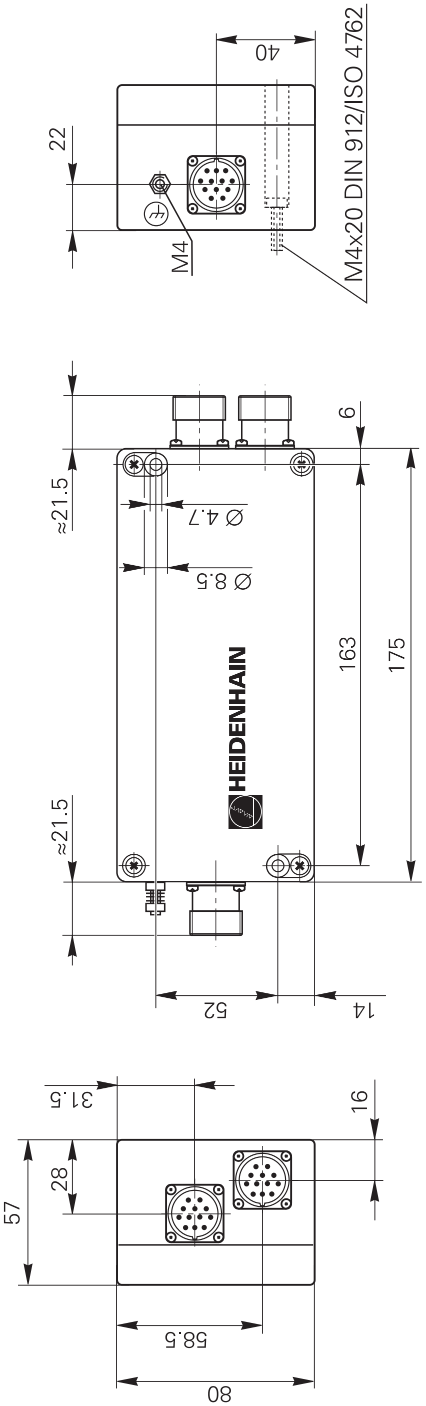
EA 6x2 Receiver Unit



TS 632

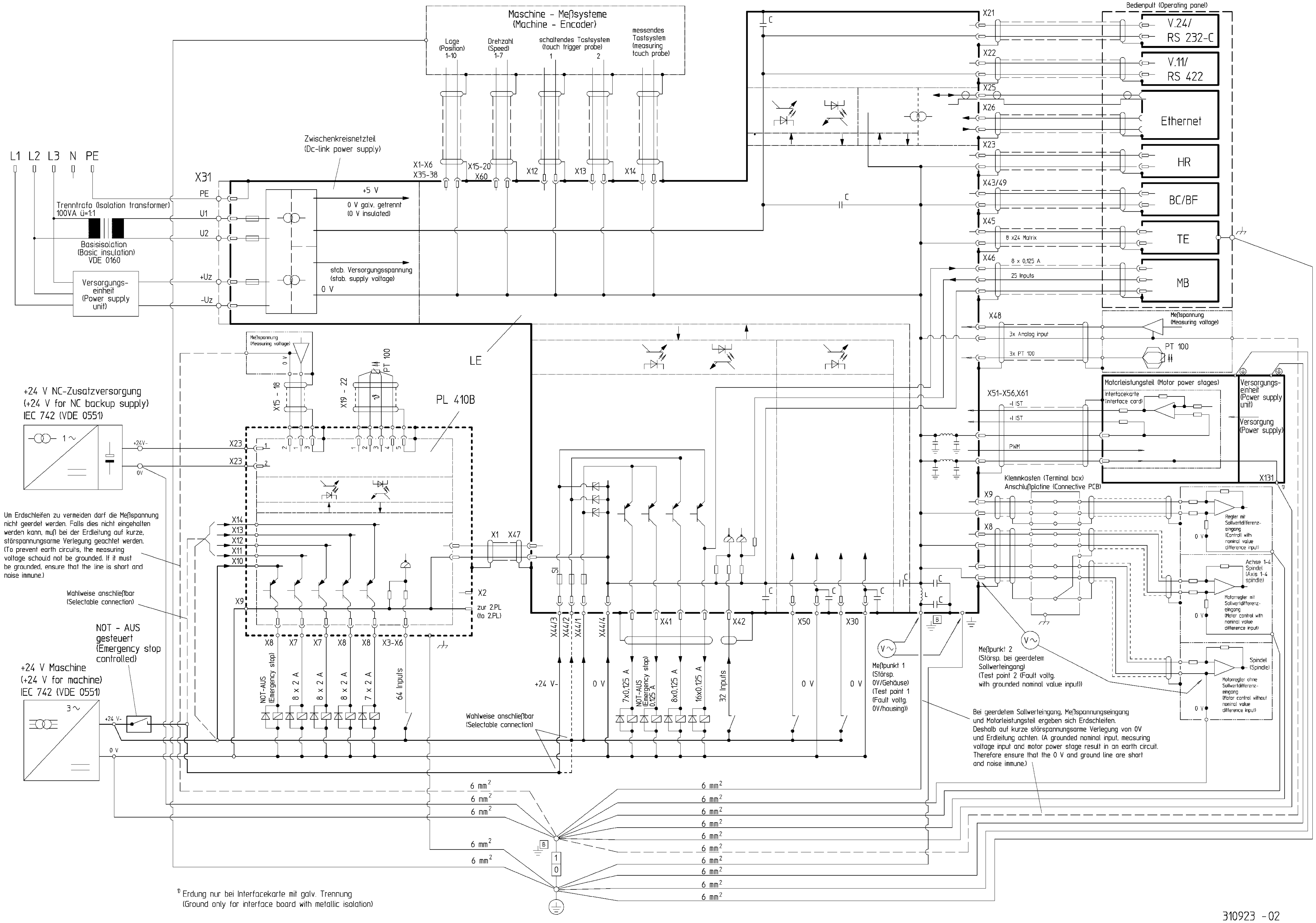


APE 652

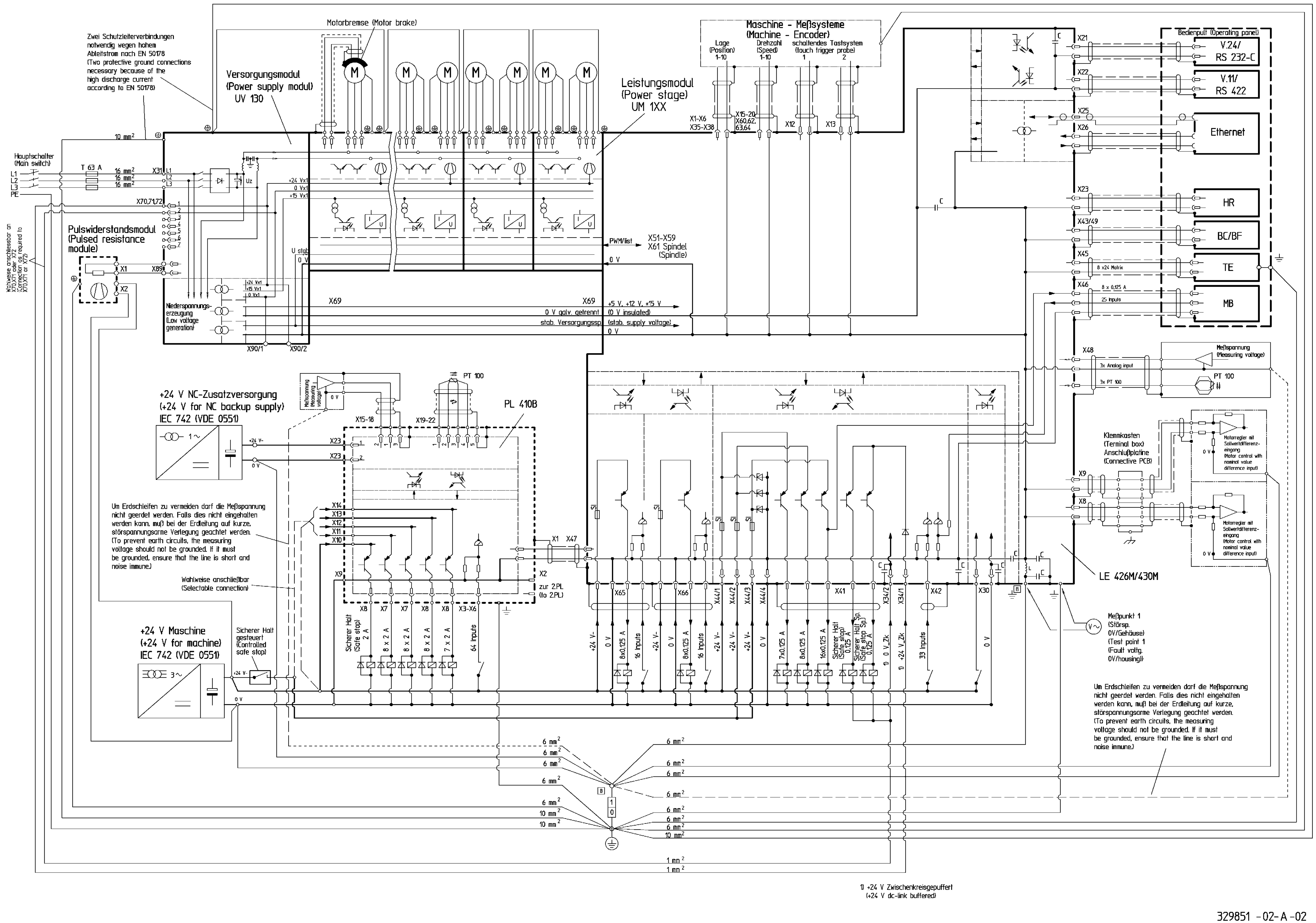

40
22
M4
M4x20 DIN 912/ISO 4762
6
≈21.5
Ø 4.7
Ø 8.5
163
175
HEIDENHAIN
≈21.5
52
14
31.5
16
28
57
58.5
80

3.24 Grounding Diagrams

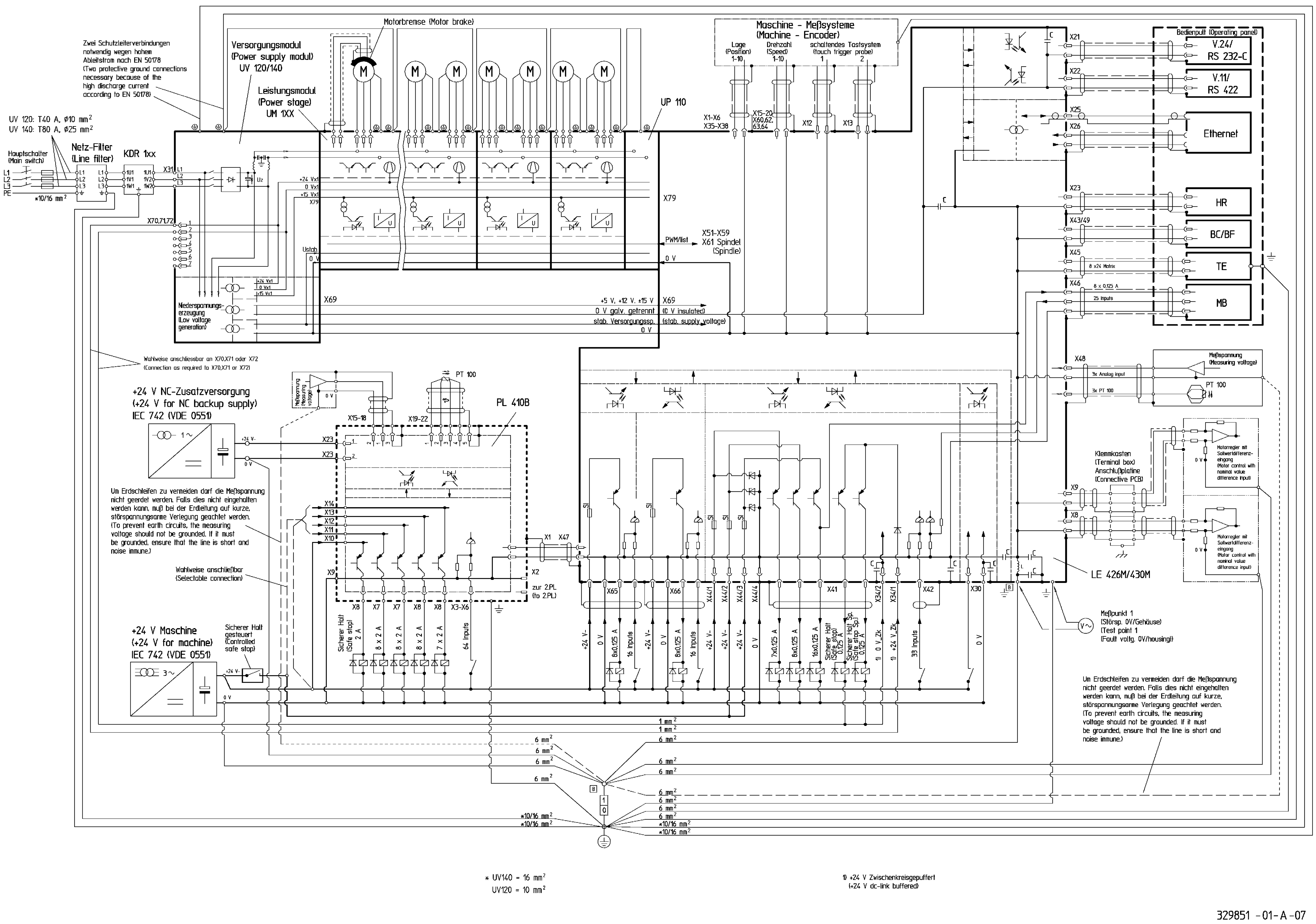
3.24.1 Grounding diagram TNC 426 CB/PB, TNC 430 CA/PA



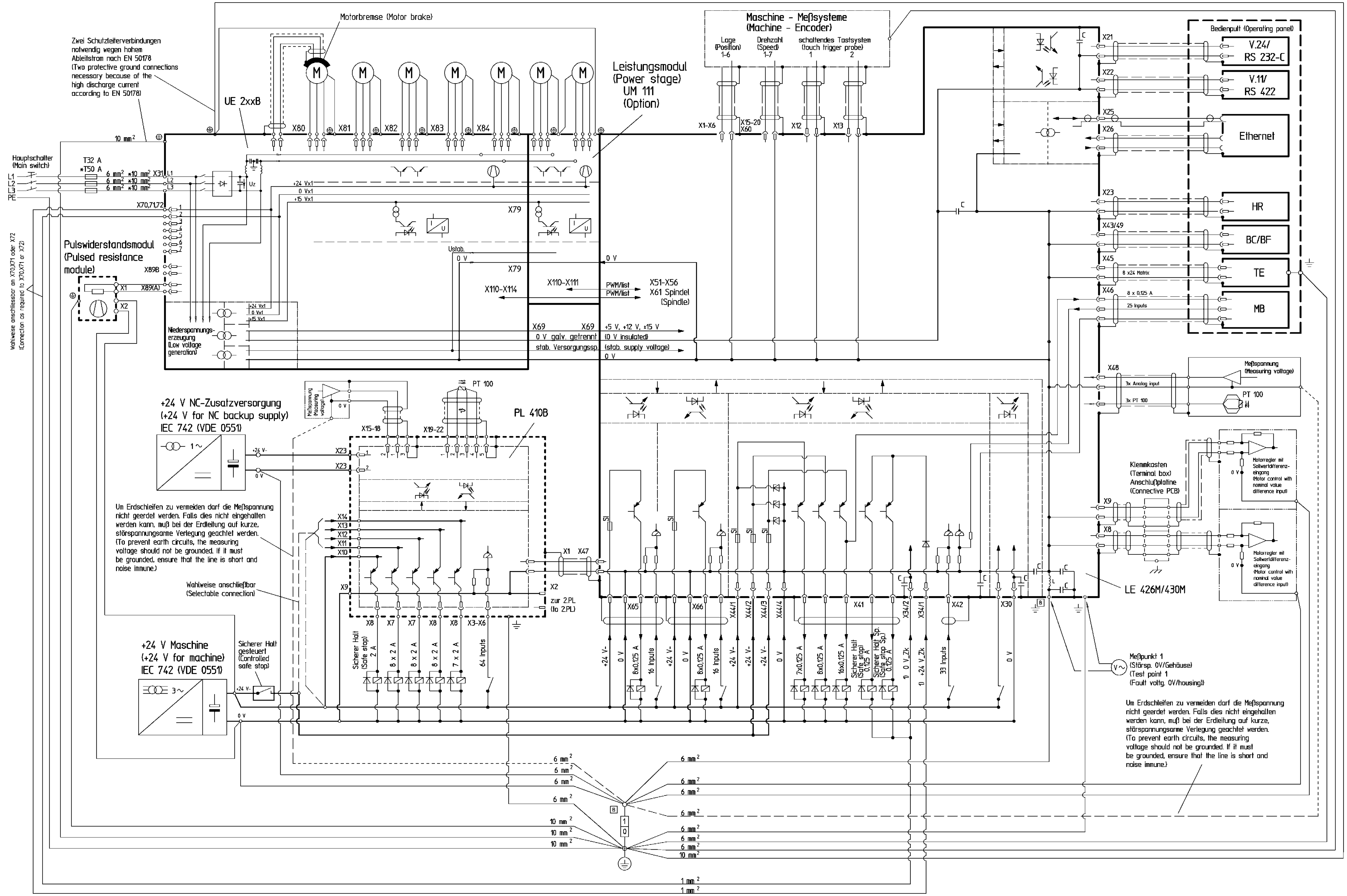
3.24.2 Grounding Diagram for TNC 426 M, TNC 430 M with Modular Nonregenerative HEIDENHAIN Inverter



3.24.3 Grounding Diagram for TNC 426 M, TNC 430 M with Modular Energy Recovery HEIDENHAIN Inverter



3.24.4 Grounding Diagram for TNC 426 M, TNC 430 M with HEIDENHAIN Compact Inverter UE 2xx B

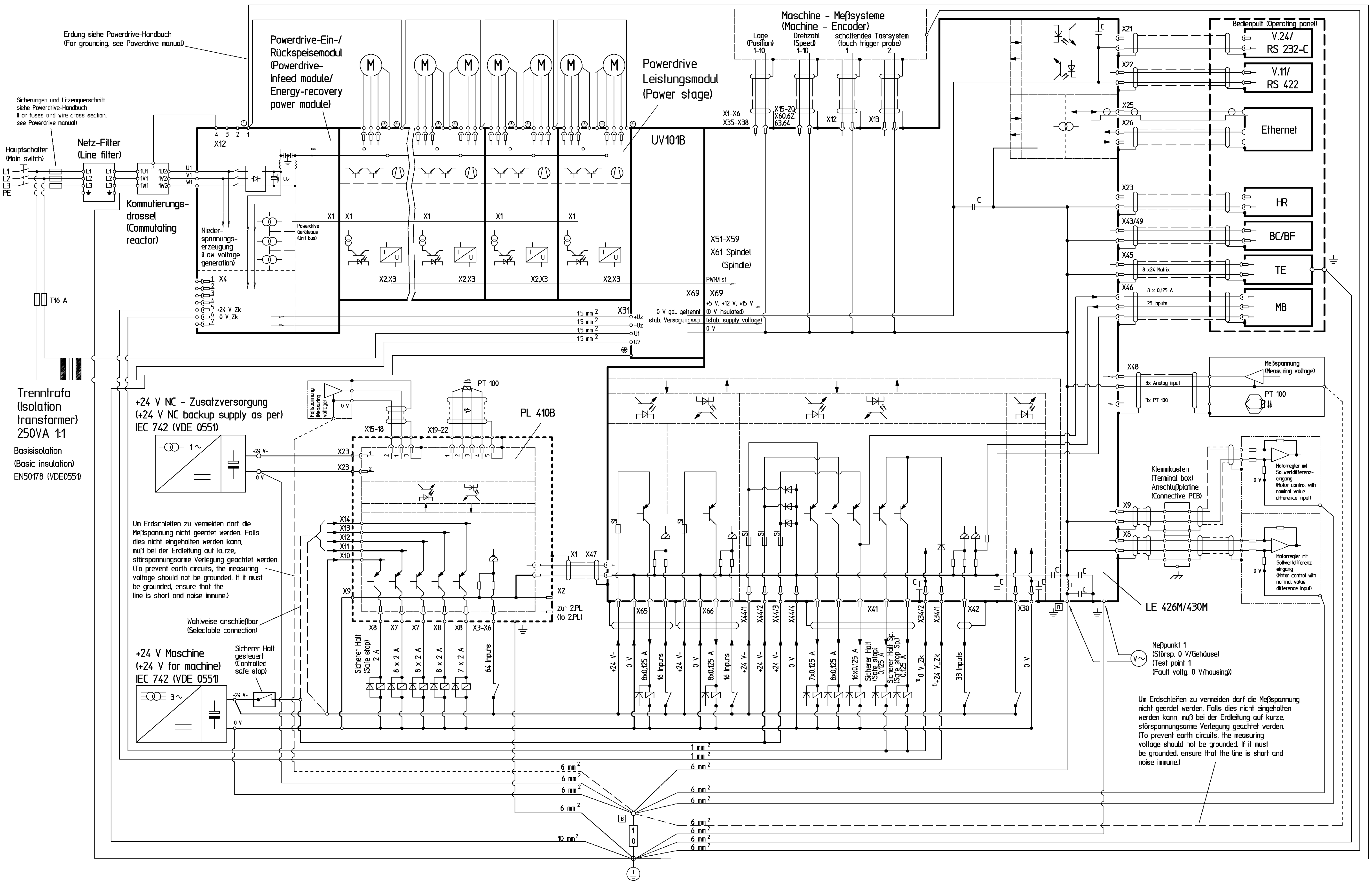


* gültig für UE230B,240B,242B
(Applies to UE230B,240B,242B)
X89B bei UE230B,UE240B,UE242B nicht vorhanden
(X89B does not exist with UE230B,UE240B,UE242B)

1) +24 V Zwischenkreisgepuffert
(+24 V dc-link buffered)

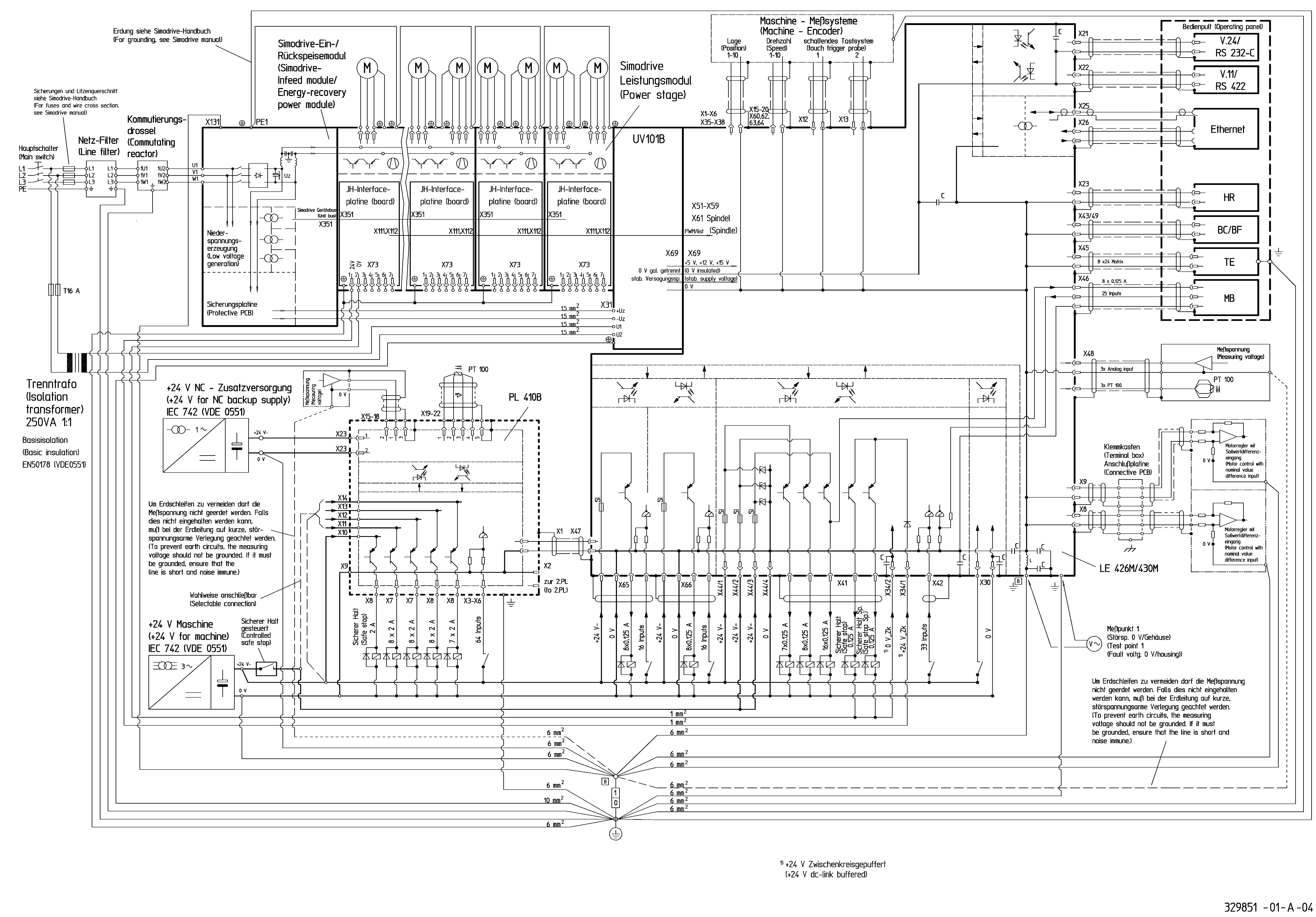
329851 -02-A-03

3.24.5 Grounding plan TNC 426 M, TNC 430 M with POWERDRIVE Inverter System

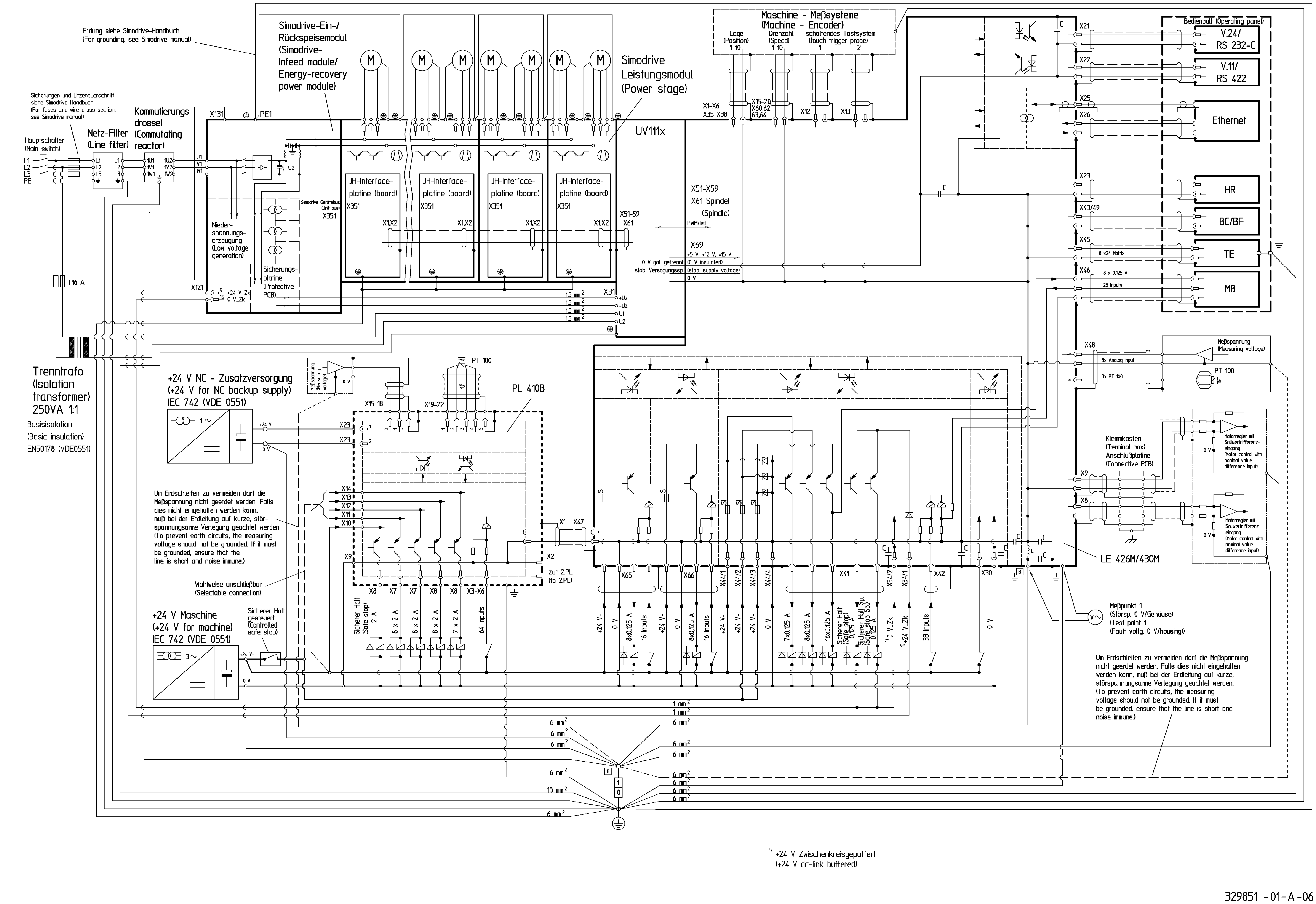


329851 -01- A -05

3.24.6 Grounding plan TNC 426 M, TNC 430 M with SIMODRIVE Inverter System in single-row configuration

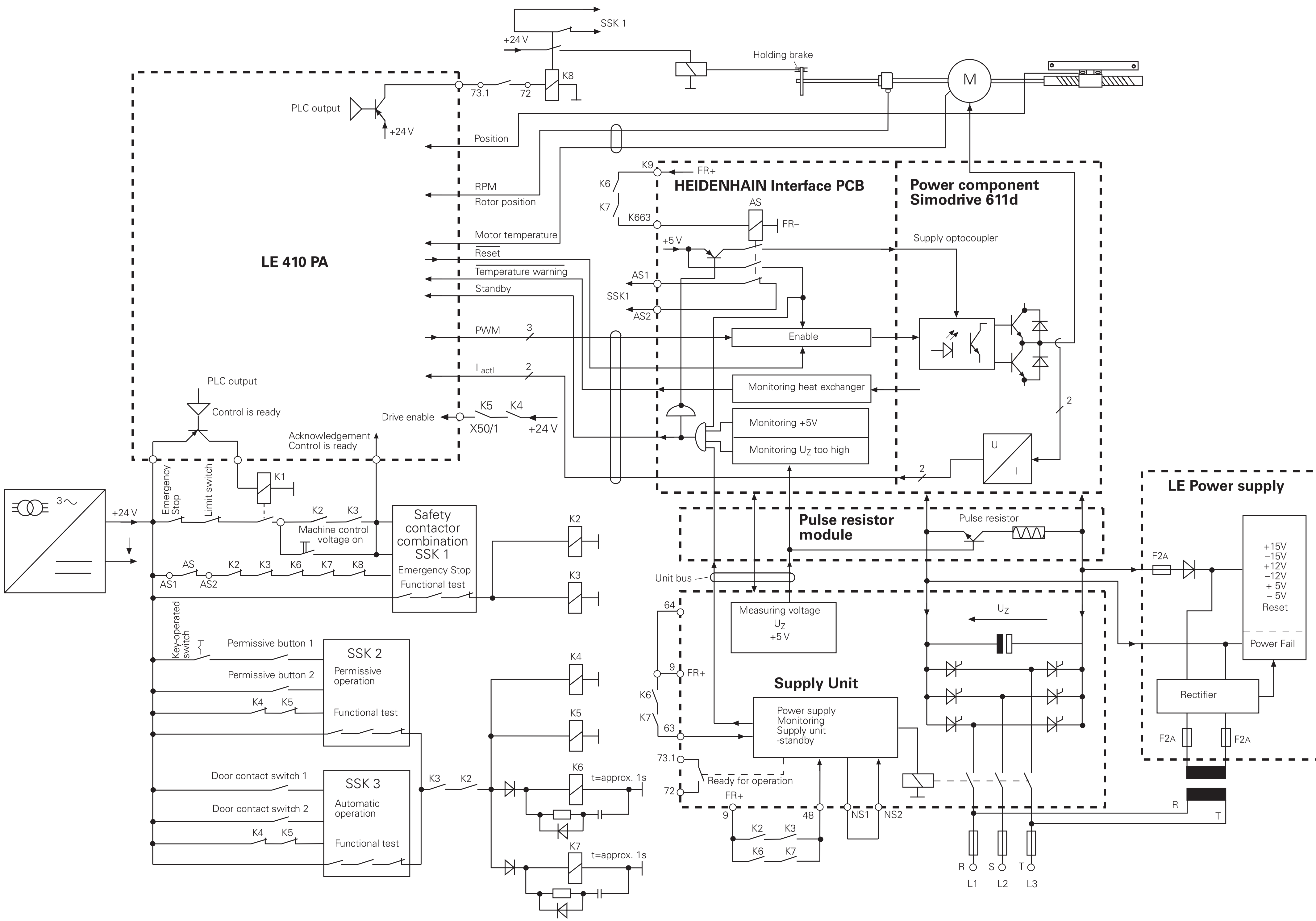


3.24.7 Grounding plan TNC 426 M, TNC 430 M with SIMODRIVE Inverter System in double-row configuration

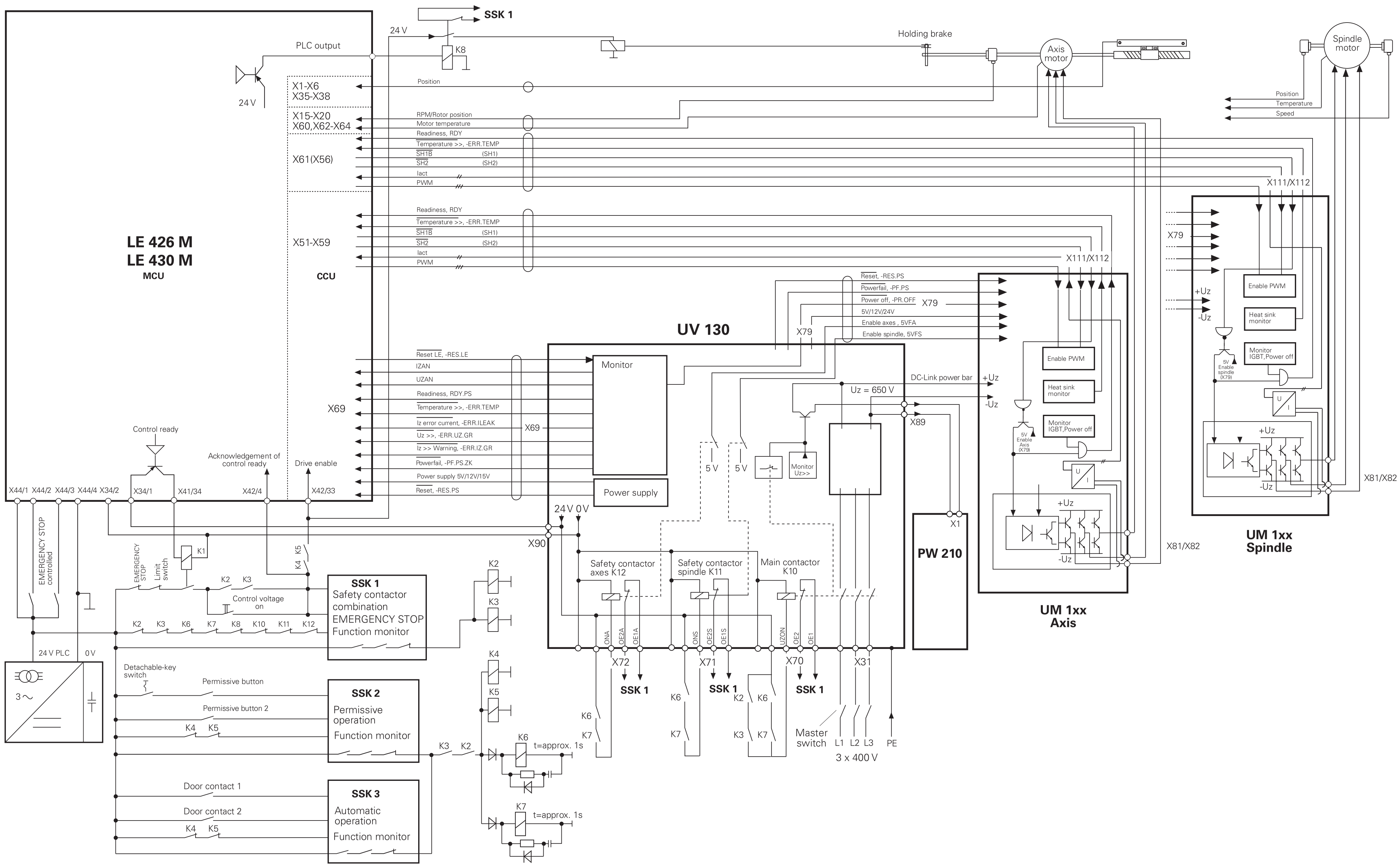


3.25 Basic Circuit Diagrams

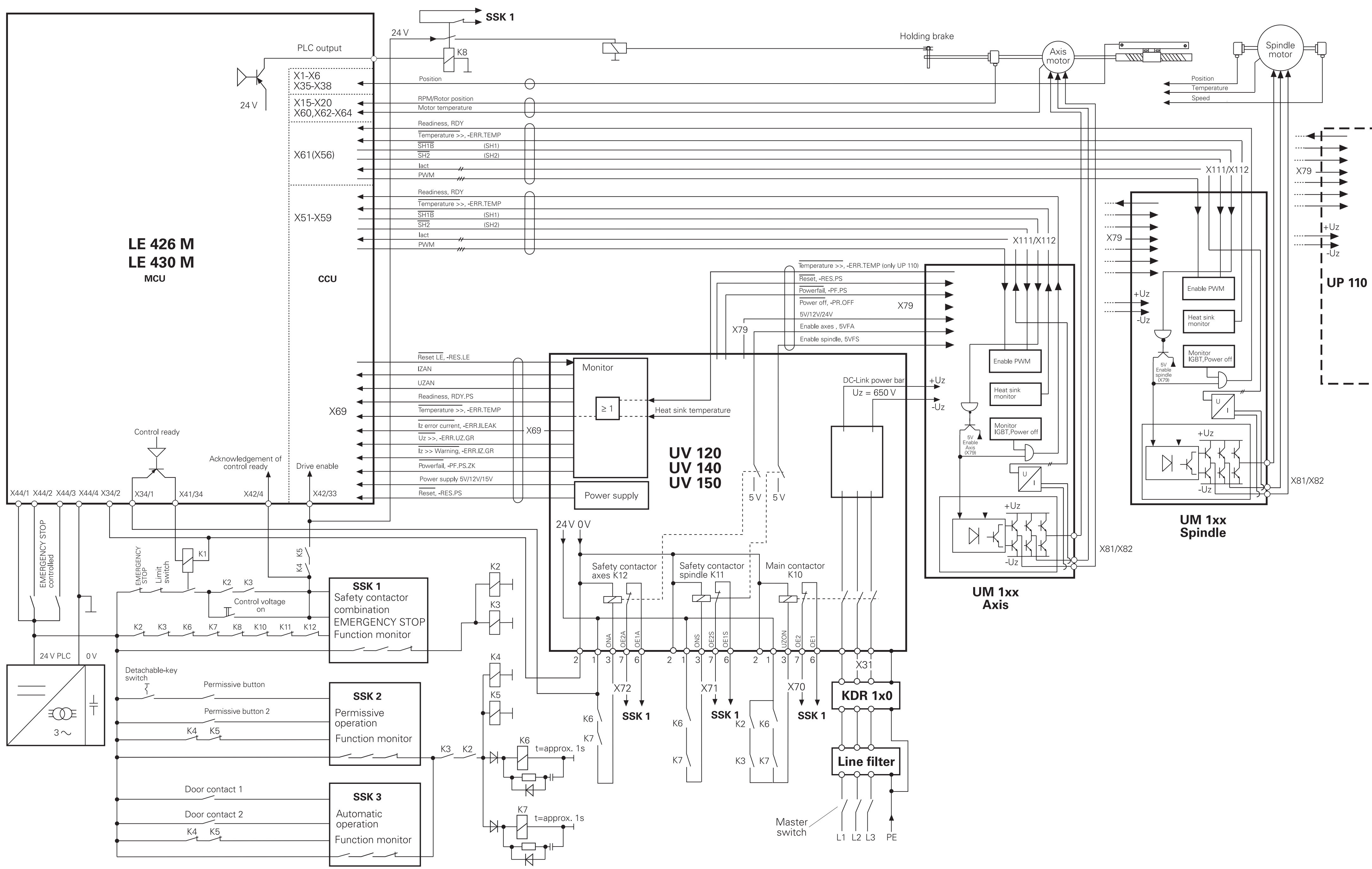
3.25.1 Basic Circuit Diagram TNC 426 PB, TNC 430 PA



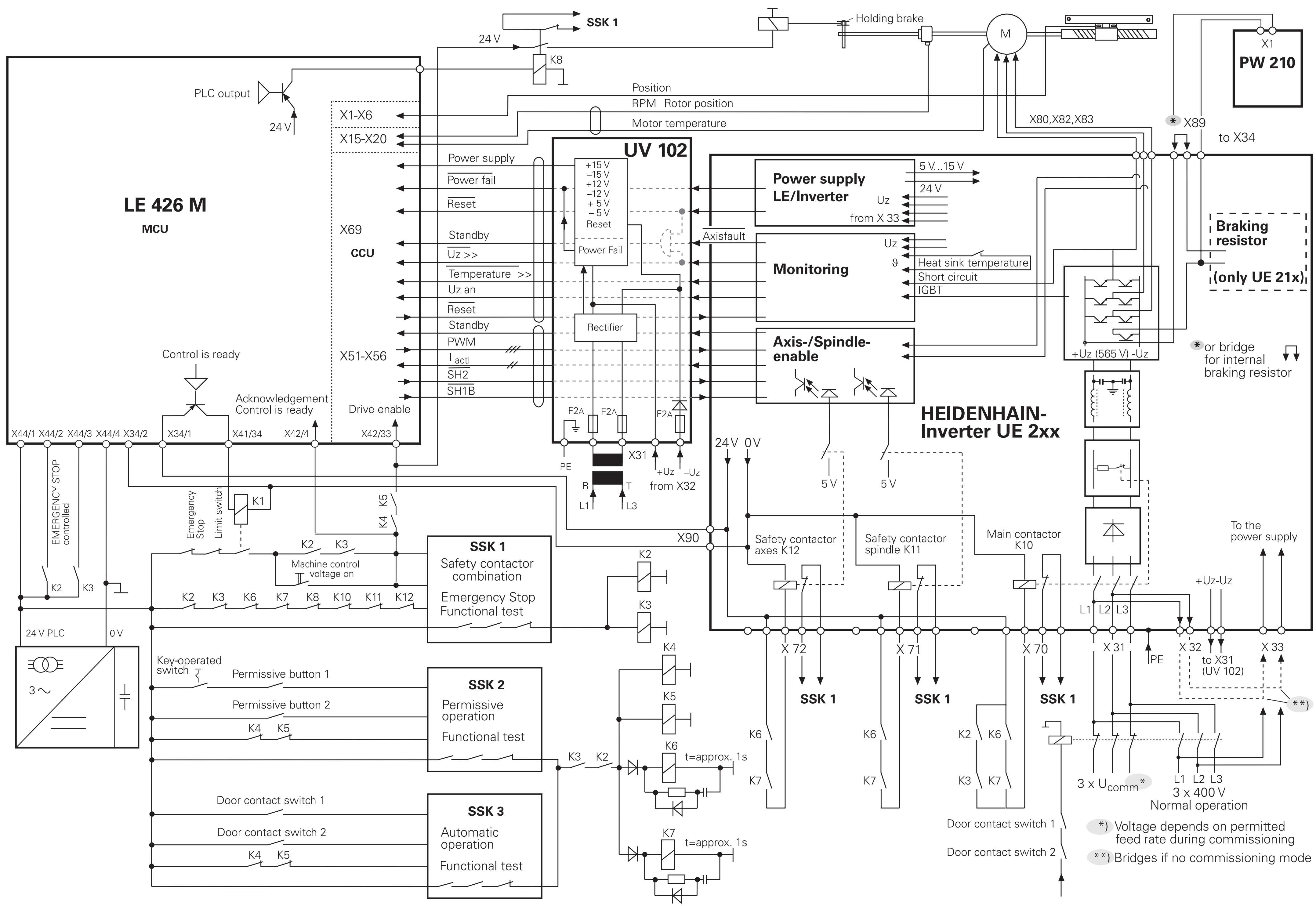
3.25.2 Basic Circuit Diagram for TNC 426 M, TNC 430 M with Modular Nonregenerative HEIDENHAIN Inverter



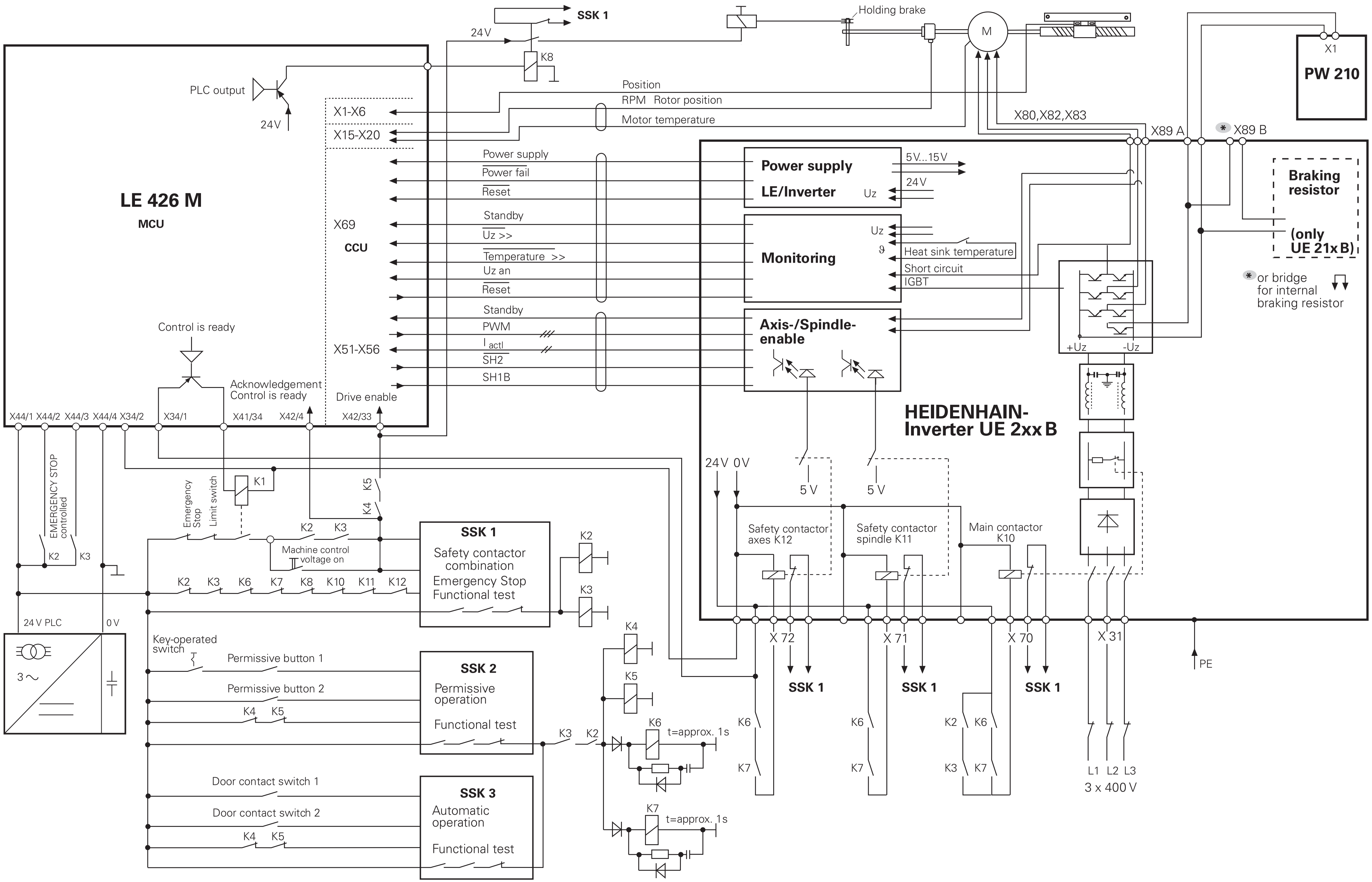
3.25.3 Basic Circuit Diagram for TNC 426 M, TNC 430 M with Modular Energy-Recovery HEIDENHAIN Inverter



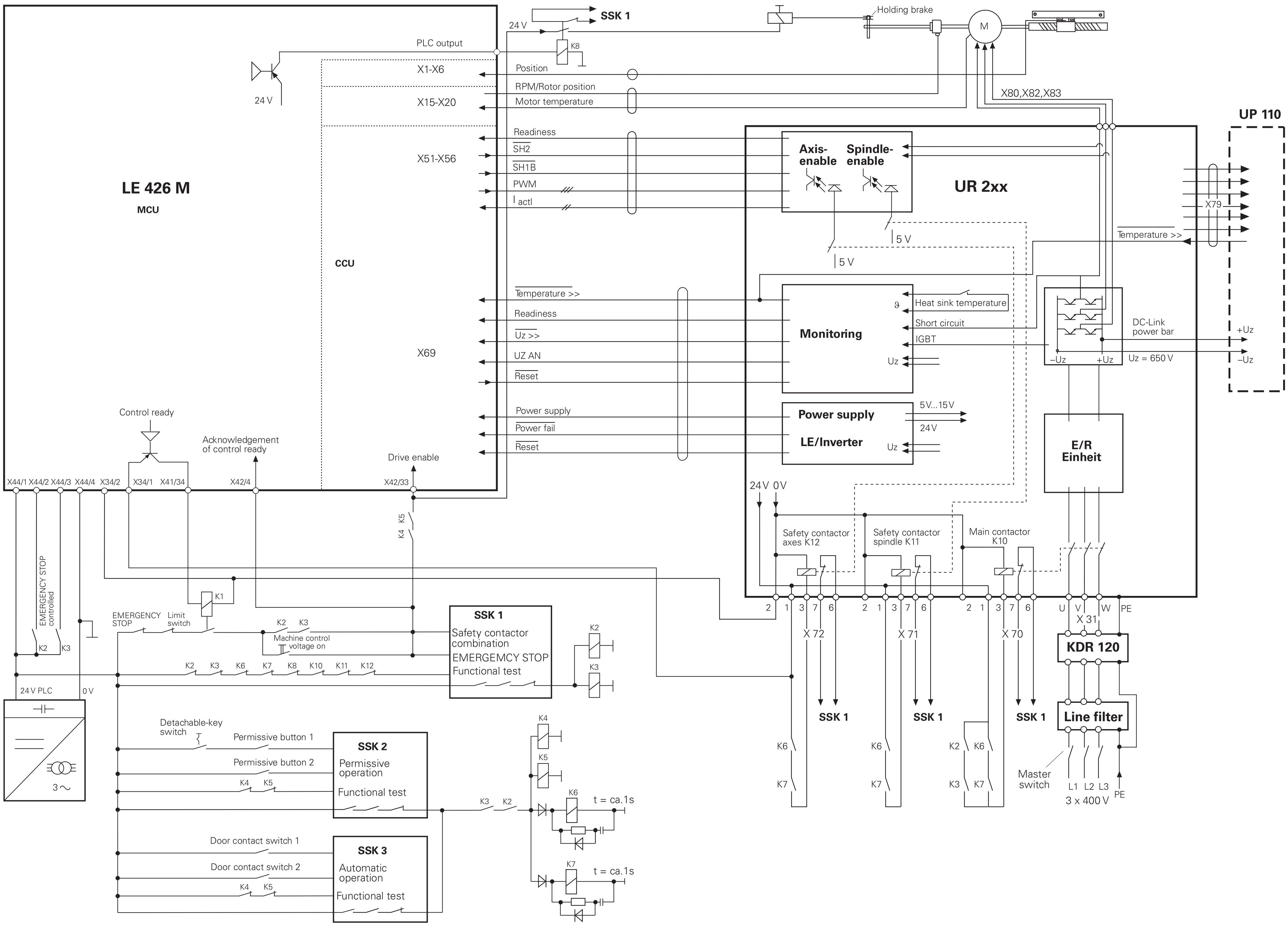
3.25.4 Basic Circuit Diagram for TNC 426 M with HEIDENHAIN Compact Inverter UE 2xx with UV 102



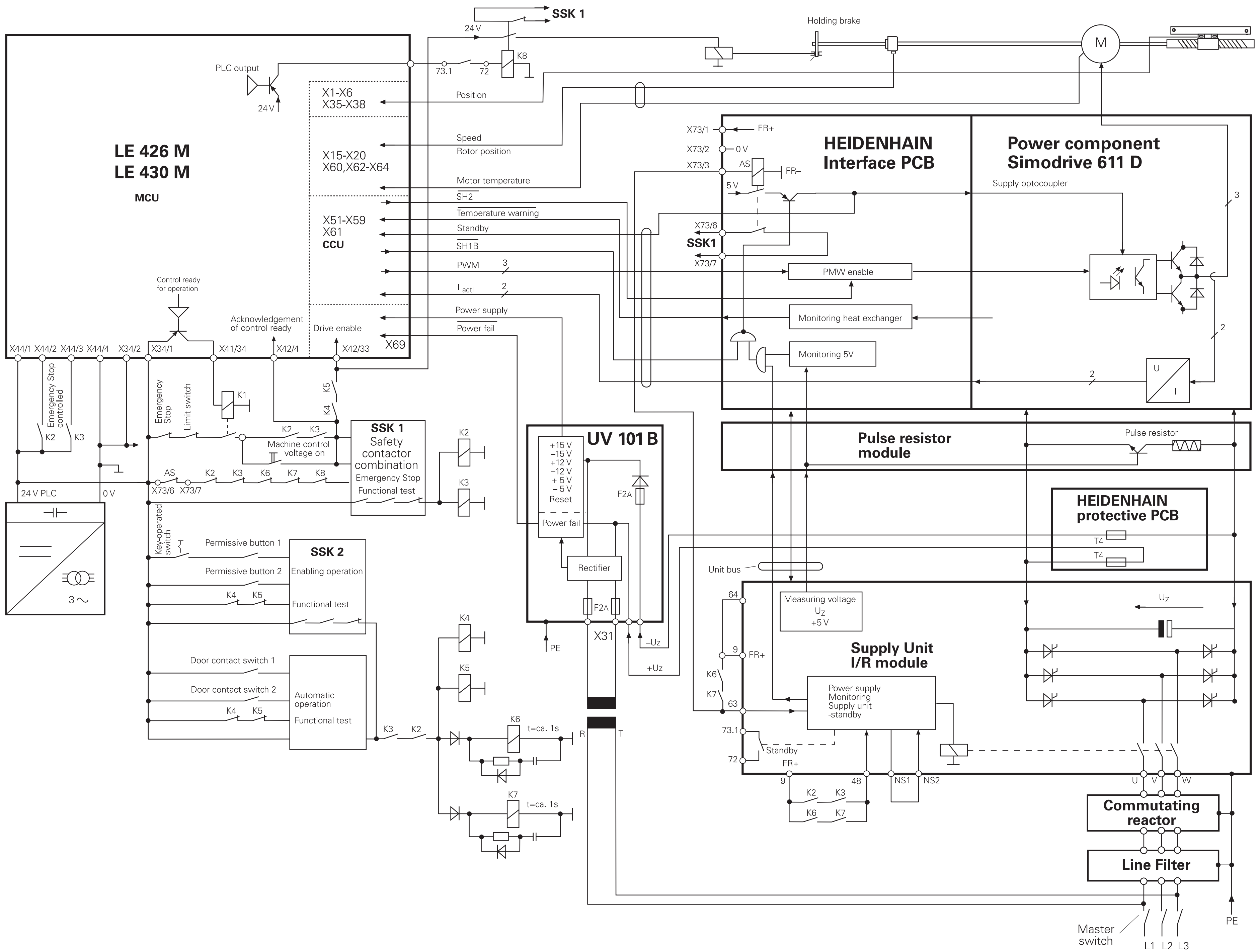
3.25.5 Basic Circuit Diagram for TNC 426 M with UE 2xxB Nonregenerative HEIDENHAIN Compact Inverter



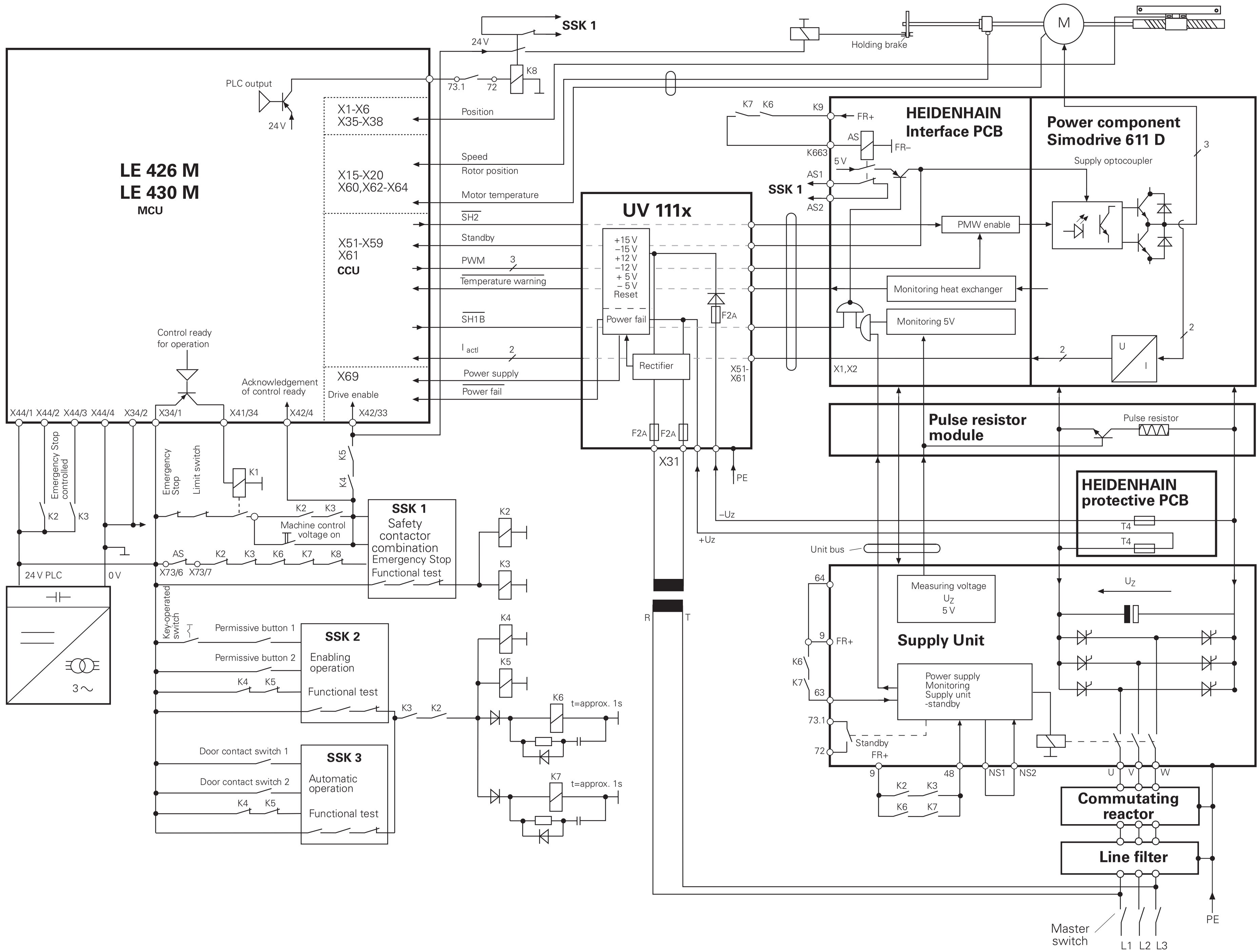
3.25.6 Basic Circuit Diagram for TNC 426 M with UR 2xxB Regenerative HEIDENHAIN Compact Inverter



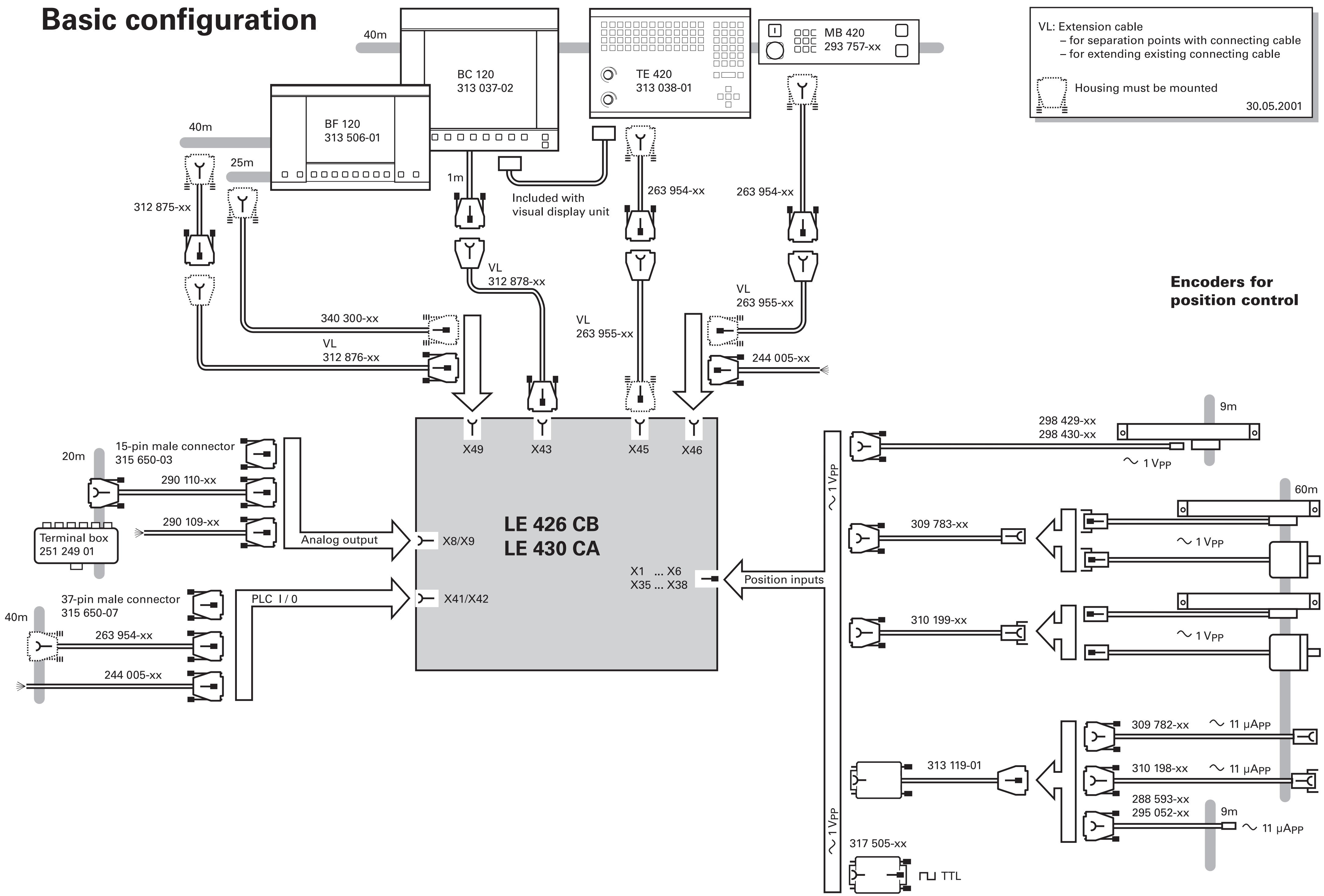
3.25.7 Basic Circuit Diagram TNC 426 M, TNC 430 M with SIMODRIVE 611D in Single Row



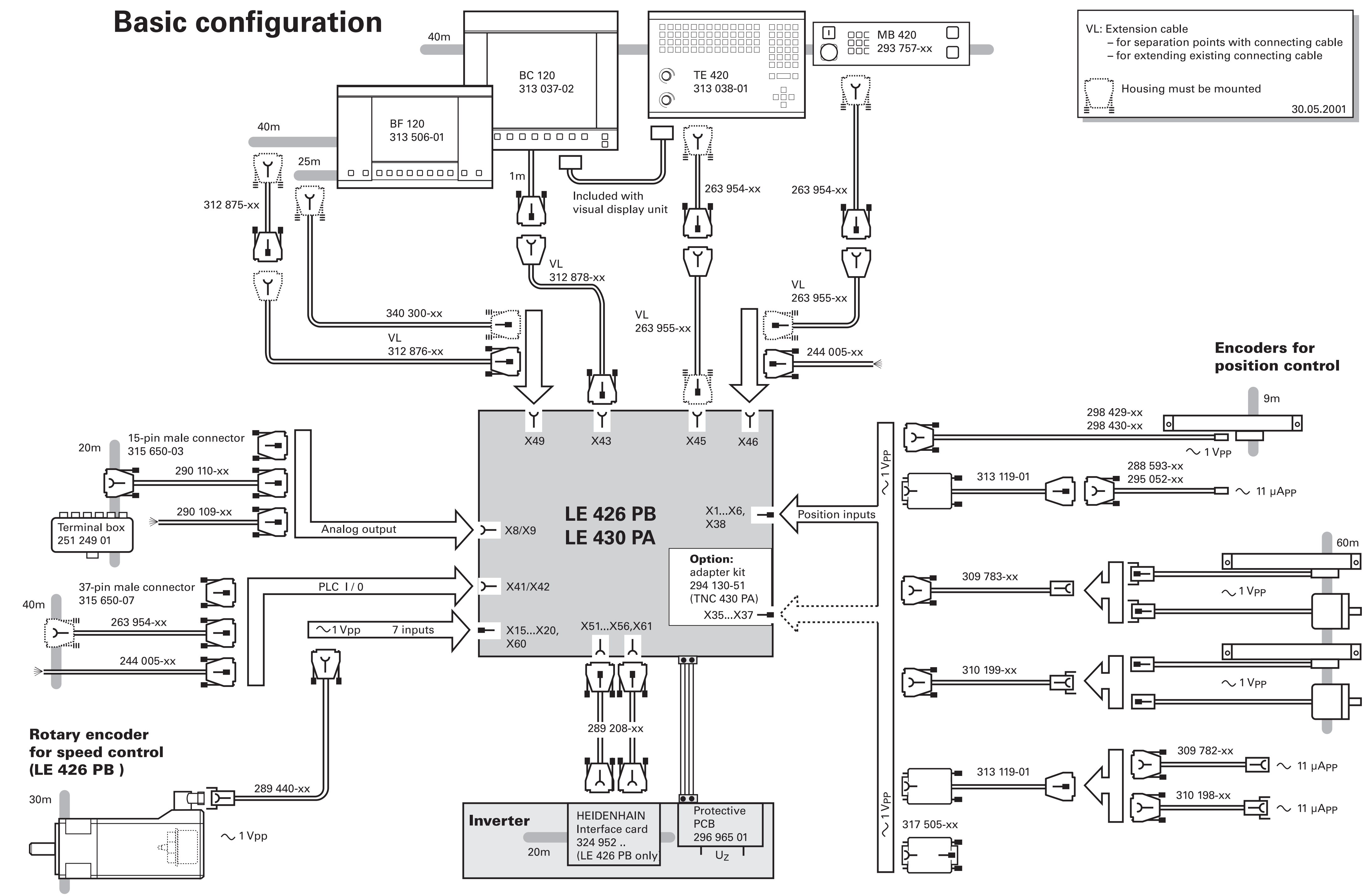
3.25.8 Basic Circuit Diagram TNC 426 M, TNC 430 M with SIMODRIVE 611D in Double Row

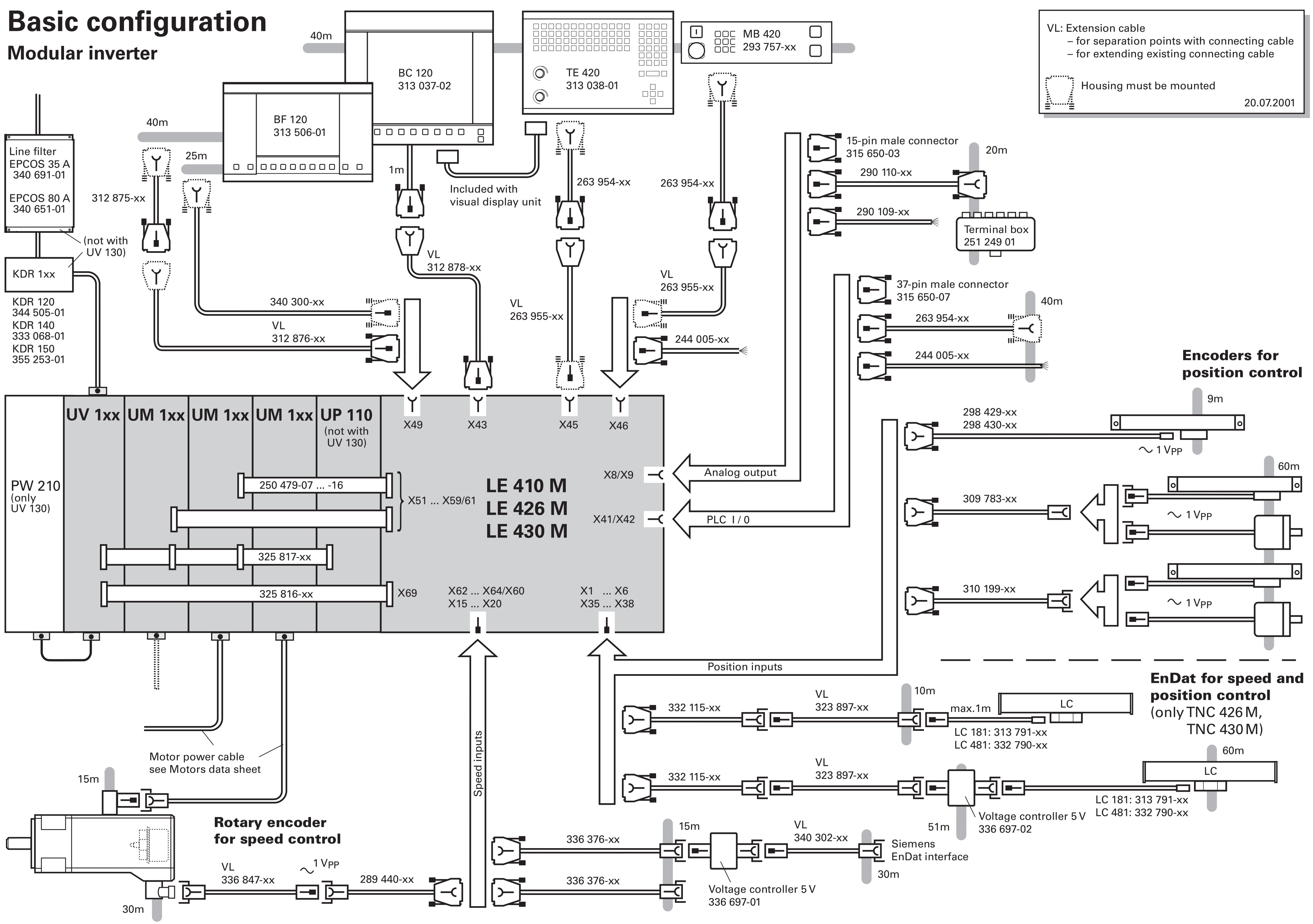


3.26.1 Cable Overview TNC 426 CB, TNC 430 CA - Basic Configuration



Basic configuration



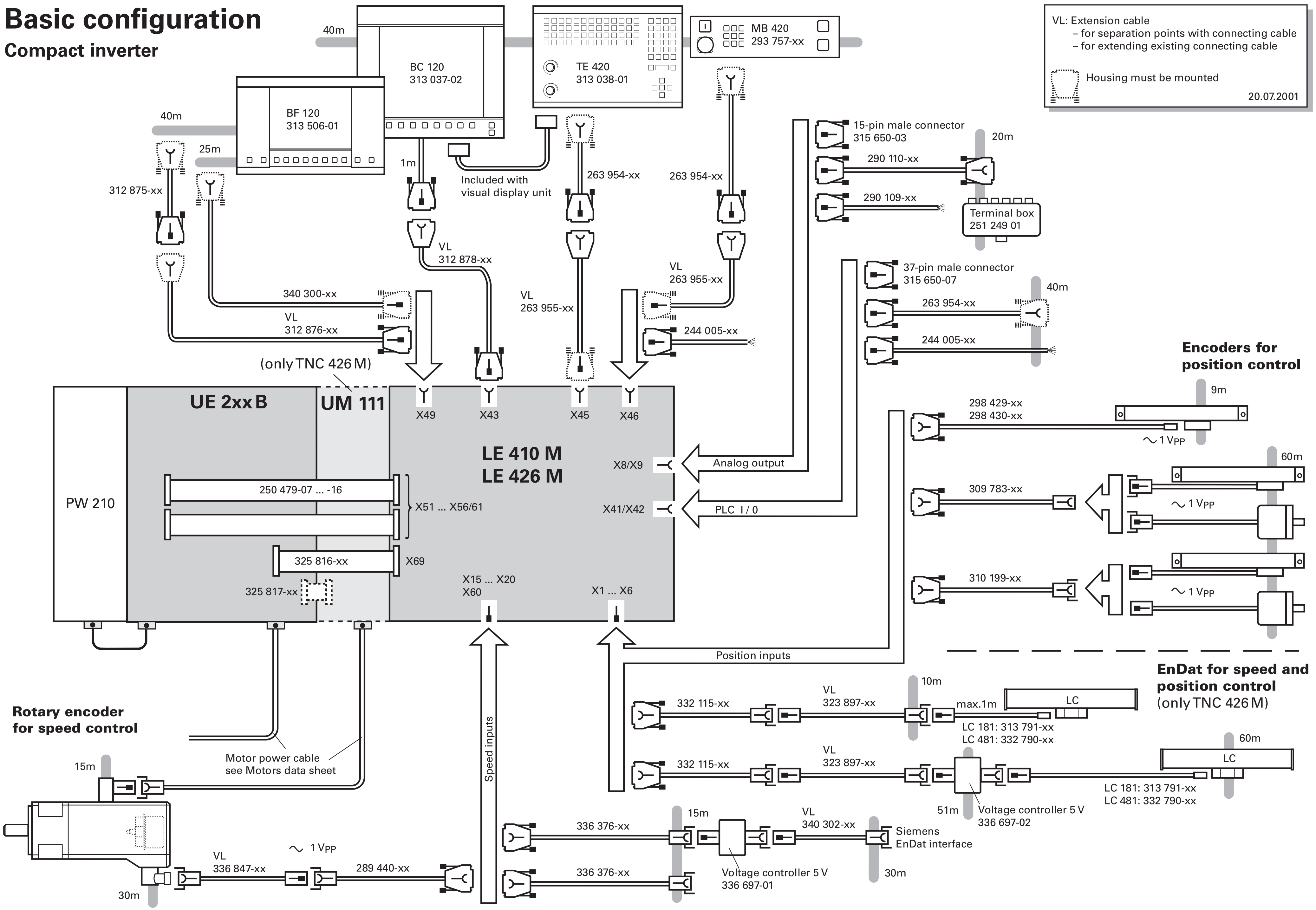


Basic configuration
Modular inverter
VL: Extension cable
– for separation points with connecting cable
– for extending existing connecting cable
Housing must be mounted
20.07.2001
Line filter
EPCOS 35 A
340 691-01
EPCOS 80 A
340 651-01
(not with UV 130)
KDR 1xx
KDR 120
344 505-01
KDR 140
333 068-01
KDR 150
355 253-01
BC 120
313 037-02
BF 120
313 506-01
TE 420
313 038-01
MB 420
293 757-xx
40m
40m
25m
1m
312 875-xx
340 300-xx
VL
312 876-xx
VL
312 878-xx
Included with visual display unit
263 954-xx
VL
263 955-xx
263 954-xx
VL
263 955-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
Encoders for position control
9m
298 429-xx
298 430-xx
1 VPP
60m
309 783-xx
1 VPP
310 199-xx
1 VPP
PW 210
(only UV 130)
UV 1xx
UM 1xx
UM 1xx
UM 1xx
UP 110
(not with UV 130)
X49
X43
X45
X46
250 479-07 ... -16
X51 ... X59/61
325 817-xx
325 816-xx
X69
LE 410 M
LE 426 M
LE 430 M
X8/X9
Analog output
X41/X42
PLC I / 0
X62 ... X64/X60
X15 ... X20
X1 ... X6
X35 ... X38
Position inputs
Speed inputs
EnDat for speed and position control
(only TNC 426 M, TNC 430 M)
332 115-xx
VL
323 897-xx
10m
max.1m
LC
LC 181: 313 791-xx
LC 481: 332 790-xx
332 115-xx
VL
323 897-xx
60m
LC
LC 181: 313 791-xx
LC 481: 332 790-xx
51m
Voltage controller 5 V
336 697-02
Motor power cable
see Motors data sheet
15m
Rotary encoder for speed control
VL
336 847-xx
1 VPP
289 440-xx
30m
336 376-xx
336 376-xx
15m
Voltage controller 5 V
336 697-01
VL
340 302-xx
Siemens
EnDat interface
30m

3.26.4 Cable Overview for TNC 426 M, TNC 430 M with HEIDENHAIN Nonregenerative Compact Inverter UE 2xxB - Basic Configuration

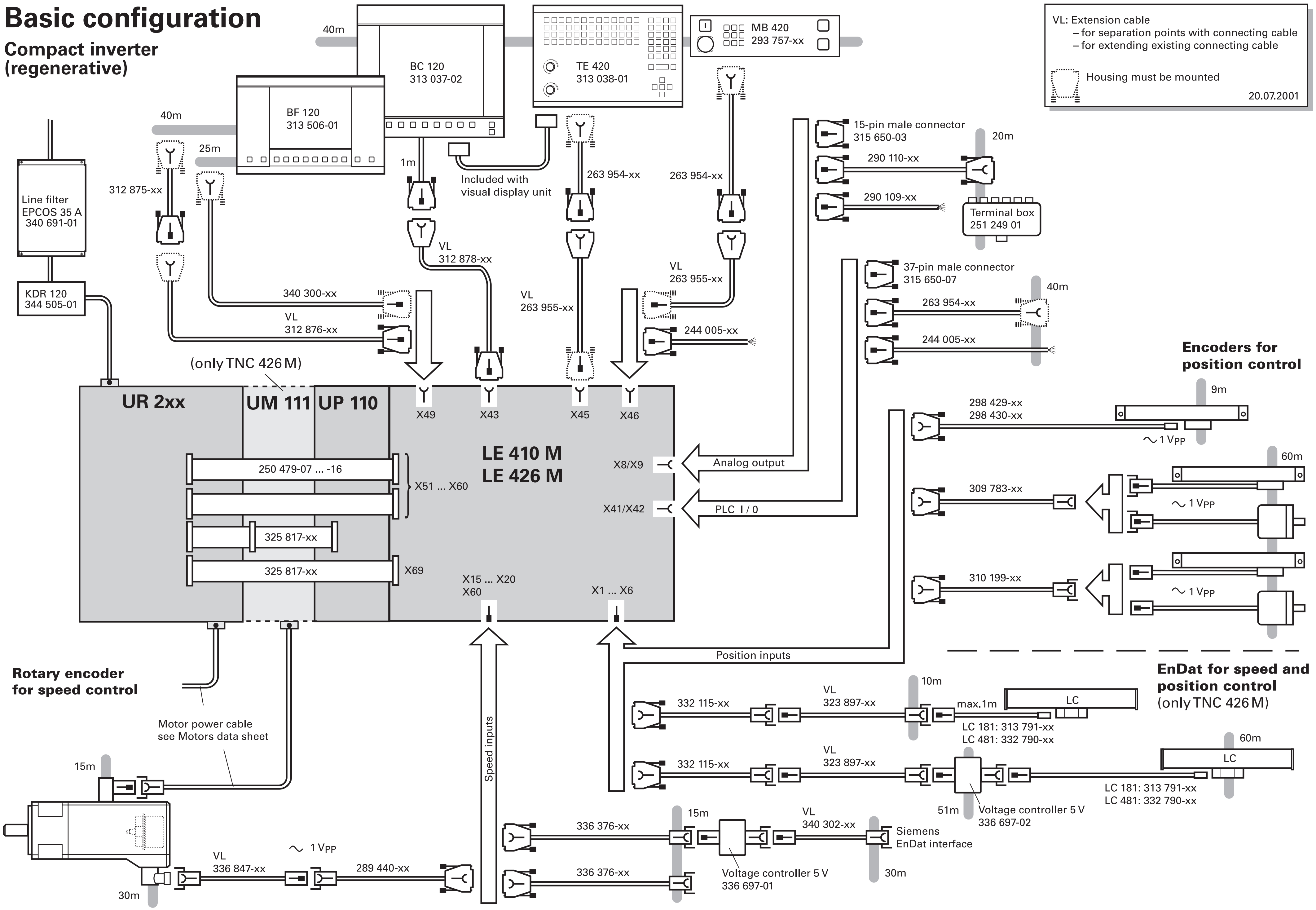
Basic configuration

Compact inverter

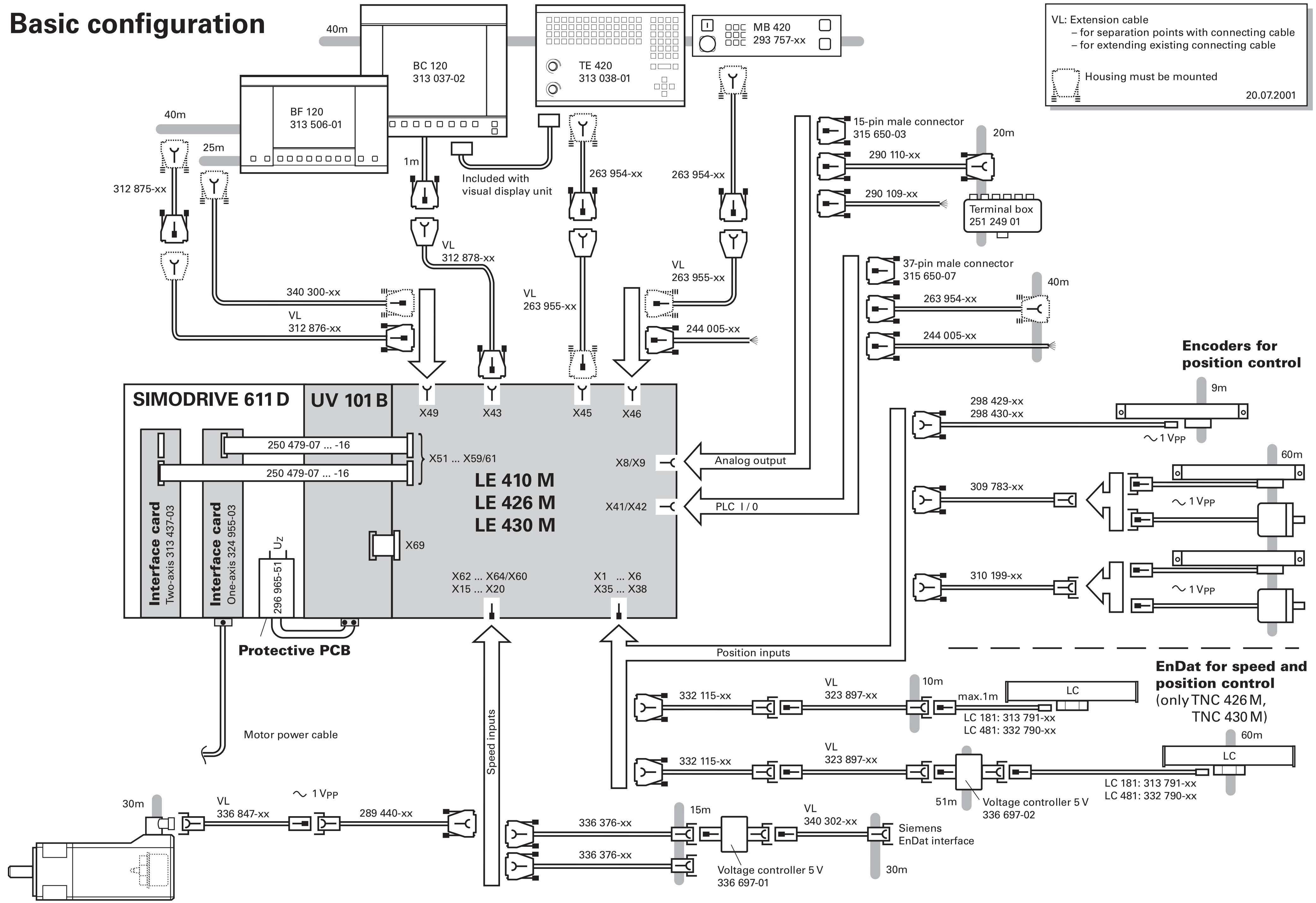


Basic configuration

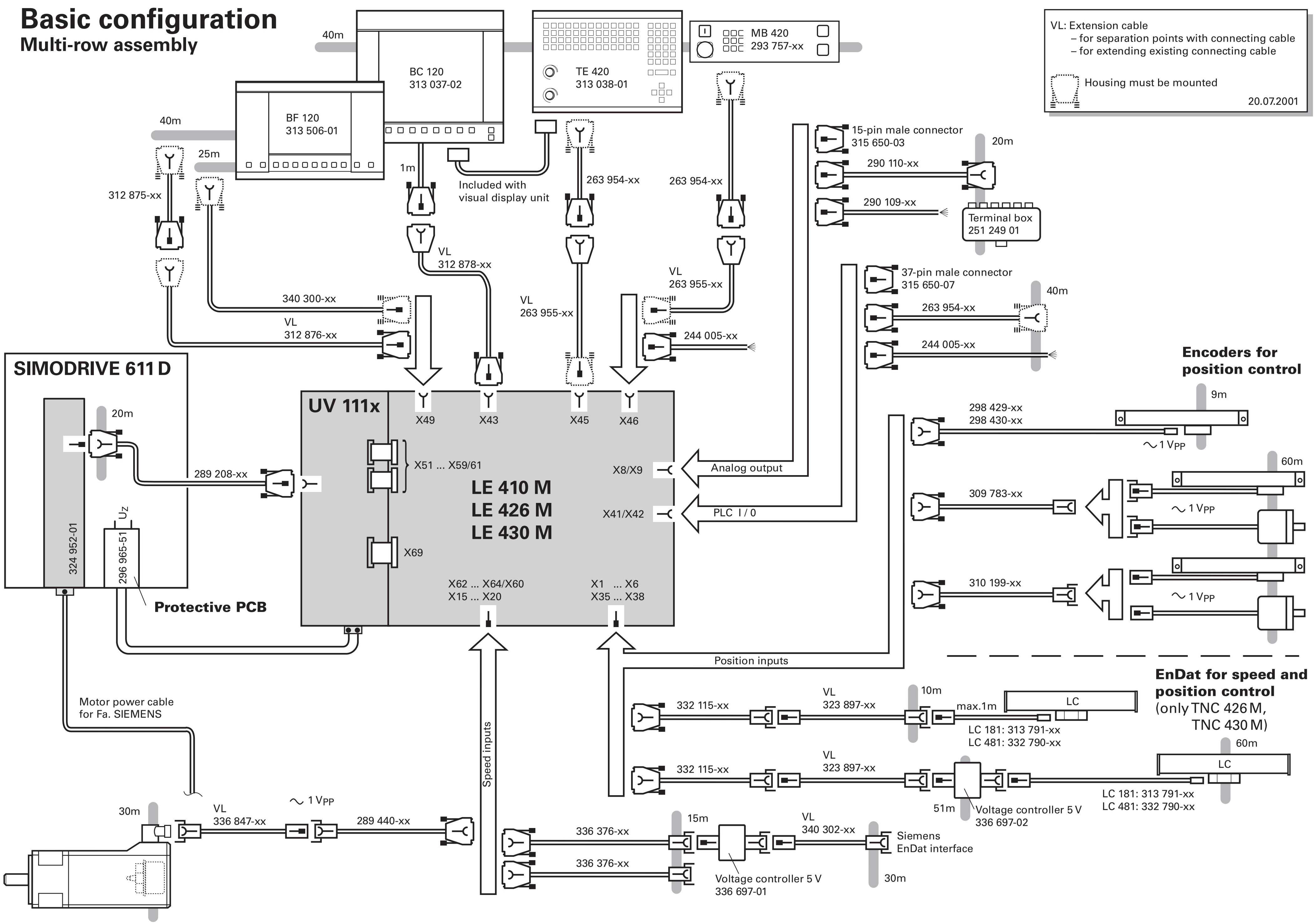
Compact inverter (regenerative)



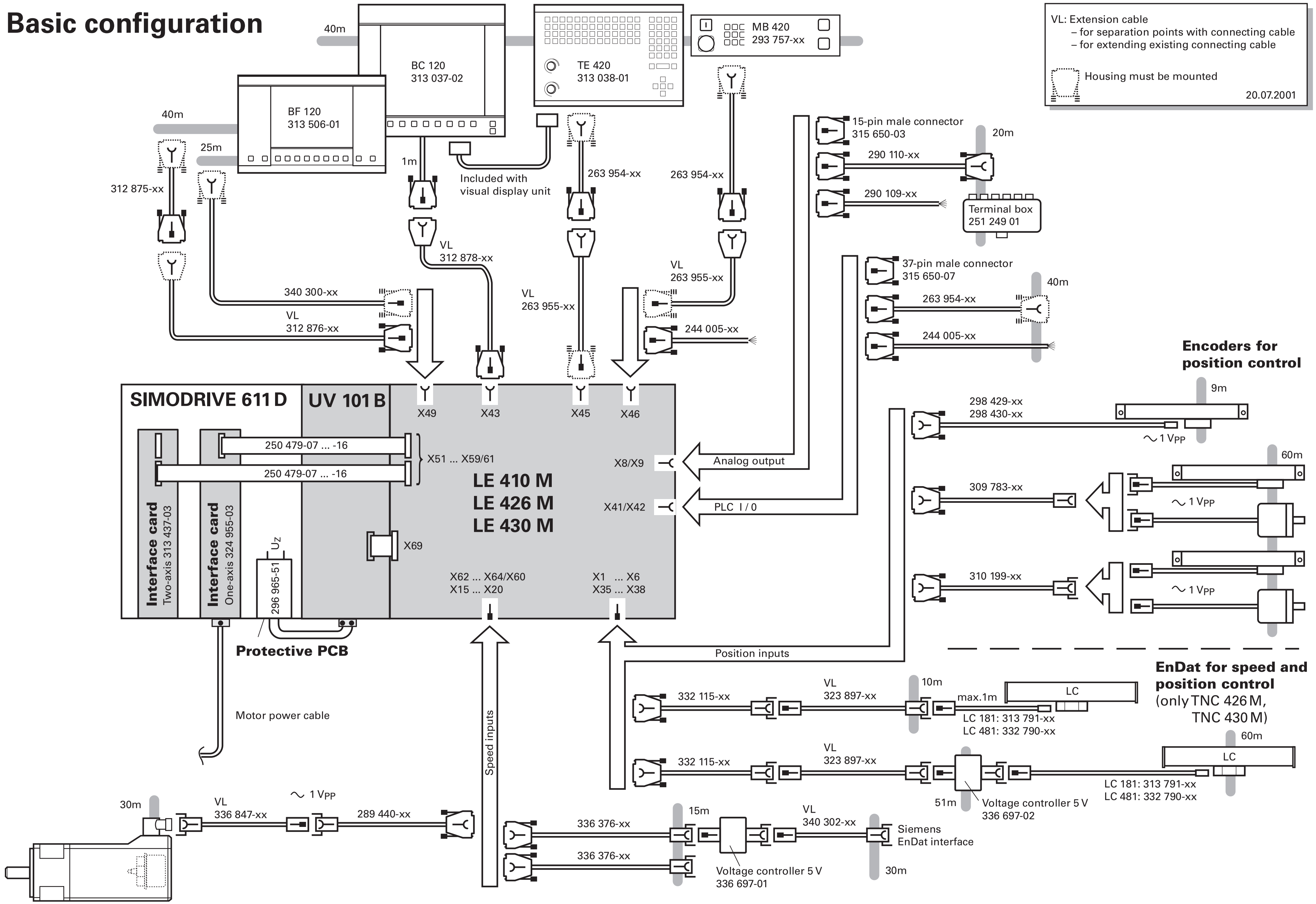
Basic configuration



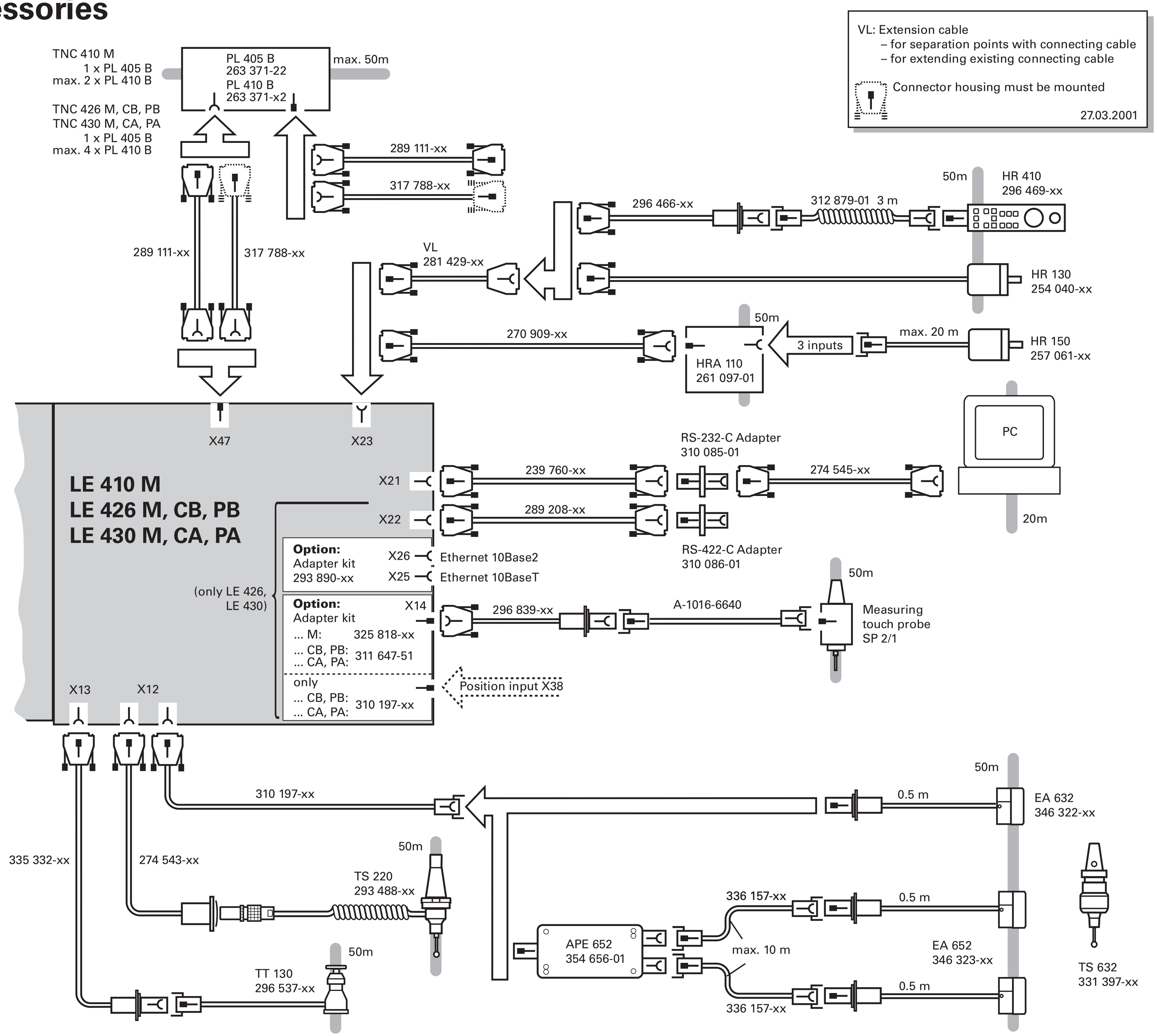
Basic configuration
Multi-row assembly



Basic configuration



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
6200 to 6299	Digitizing with triggering touch probe
6300 to 6399	Digitizing with measuring 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 TNC 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 be 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 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 (As of NC software 280 472-01)

Only for MP2020.x (linear distance of one motor revolution); starting with NC Software 280 476-01 also 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 TNC generates a standard machine parameter list (MP NAME). In this list the TNC 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 TNC:

- 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 by the PLC. Starting in NC Software 280 476-06, this MP can also be changed in a running NC program during a strobe output.
RUN	The MP can also be changed while a 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 (see "Display and Operation").
- 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 = [code number], 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:

Example: MP910.0 = 100.12 [mm] Transfer value: 1001200 (4 decimal places)

The value in the run-time memory is changed. The value from the editable machine parameter file does not change. The old value becomes valid again after the machine parameter file is edited and exited.

For non-indexed machine parameters, zero must be transferred as the index. Once the NC program has started, the module operates only during the output of an M/S/T/Q strobe.

Call only in a submit job.

Call:

```
PS    B/W/D/K   <MP number>
PS    B/W/D/K   <MP index>
PS    B/W/D     <MP value>
CM    9031
PL    B/W/D     <Error code>
                0: No error
                1: MP does not exist / is not changeable / is not changeable
                during a running program
                2: MP value out of range
                3: Error while saving (fatal error)
                4: Call was not in a submit or spawn job
                5: Call during running program without strobe
```

Error recognition:

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

Module 9032 Read machine 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 recognition:

Marker	Value	Meaning
M4203	0	MP was read
	1	MP could not be read from the table

Module 9310 Read the machine parameter from the run-time memory

With this module you can 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 recognition:

Marker	Value	Meaning
M4203	0	MP was read
	1	Error code in W1022
W1022	20	Module was not called in a spawn job or submit job

Module 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 respect any existing safety problems 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>
    0 to 15
CM 9033
    Note: If a new file is selected, program execution ends here.
PS 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: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Axis not active 1: Axis active	RESET	6 – 5
MP20	Monitoring functions for the axes Format: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Monitoring not active 1: Monitoring active	PLC RUN	6 – 11
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 – 11
MP21.0	No function		
MP21.1	Amplitude of encoder signals		
MP21.2	Edge separation of encoder signals		
MP100	Designation of axes Format: XYZABCUVWxyzabcuvw- Input: Characters 1 to 9 correspond to axes 1 to 9	PLC RUN	6 – 5, 6 – 28
MP100.0	Traverse range 1		
MP100.1	Traverse range 2		
MP100.2	Traverse range 3		
MP110.0-8	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		6 – 16

MP	Function and input	Software version and behavior	Page
MP111	Position encoder input for the spindle(s) Input: 0: No position encoder input 1 to 6: Position encoder inputs X1 to X6 35 to 38: Position encoder inputs X35 to X38	280 474-03	6 – 18, 6 – 181
MP111.0	Position encoder input for the first spindle		
MP111.1	Position encoder input for the second spindle		
MP112.0-8	Assignment of speed encoder inputs to the axes Input: 0: No speed encoder input 15 to 20: Speed encoder inputs X15 to X20 62 to 64: Speed encoder inputs X62 to X64	280 474-03 RESET	6 – 16
MP113	Speed encoder for the spindle(s) Input: 0: No speed encoder input 15 to 20: Speed encoder inputs X15 to X20 60: Speed encoder input X60 (only LE with integral spindle DSP) 62 to 64: Speed encoder inputs X62 to X64	280 474-03 RESET	6 – 18, 6 – 183
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: %xxxxxxxxxx Input: Bit 0 to bit 5: Position encoder inputs X1 to X6 Bit 6 to bit 9: Position encoder inputs X35 to X38 0: 1 V_{PP} 1: 11 μA_{PP}	280 474-01 RESET	6 – 9
MP115.1	Reserved Format: %xxxxxxxxxx Input: Enter %0000000000		
MP115.2	Input frequency of the position encoder inputs Format: %xxxxxxxxxx Input: Bit 0 to bit 5: Position encoder inputs X1 to X6 Bit 6 to bit 9: Position encoder inputs X35 to X38 With 1 V_{PP}: 0: 50 kHz 1: 350 kHz With 11 μA_{PP}: 0: 50 kHz 1: 150 kHz		

MP	Function and input	Software version and behavior	Page
MP120.0-8	Assignment of speed encoder outputs to the axes	280 474-01	6 – 16
	Input: 0: No servo-controlled axis 1 to 6: Analog output 1 to 6 at terminal X8 7 to 13: Analog output 7 to 13 at terminal X9 51 to 59: Digital output X51 to X59	RESET	
MP120.0-8	Assignment of speed encoder outputs to the axes	RESET	
	Input: 0: No servo-controlled axis A1 to A6: Analog outputs 1 to 6 terminal X8 A 7 to A13: Analog outputs 7 to 13 at terminal X9 D1 to D6: Digital axes 1 to 6		
MP121	Nominal speed command output of the spindle(s)		6 – 18
MP121	Nominal speed command output of the spindle	RESET	
	Input: 0: No servo-controlled axis A1 to A6 or 1 to 6: Analog outputs 1 to 6 at terminal X8 A7 to A13 or 7 to 13: Analog outputs 7 to 13 at terminal X9 S1: Digital spindles		
MP121.0	Nominal speed command output of the first spindle	280 474-01	
	Input: 0: No servo-controlled axis 1 to 6: Analog output 1 to 6 at terminal X8 7 to 13: Analog output 7 to 13 at terminal X9 51 to 59, 61: Digital outputs X51 to X59, X61	RESET	
MP121.1	Nominal speed command output of the second spindle	280 474-01	
	Input: 0: No servo-controlled axis 1 to 6: Analog output 1 to 6 at terminal X8 7 to 13: Analog output 7 to 13 at terminal X9 51 to 59: Digital output X51 to X59	RESET	
MP210	Counting direction of position encoder output signals	RESET	6 – 10
	Format: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Positive 1: Negative		

MP	Function and input	Software version and behavior	Page
MP331.0-8	Distance for the number of signal periods in MP332 Input: 0.0001 to 99.999 999 9 [mm] or [°]	PLC REF	6 – 9
MP332.0-8	Number of signal periods for the distance in MP331 Input: 1 to 16 777 215	PLC REF	6 – 9
MP334.0-8	Nominal increment between two fixed reference marks on encoders with distance-coded reference marks Input: 1 to 65 535 0: 1 000	280 474-09 PLC	6 – 9
MP340.0-8	Interpolation factor for external interpolation Input: 0 to 99 0 = 1: No external interpolation	280 474-13 RESET	6 – 9
MP410	Assignment of axis keys IV and V Input: Axis designation XYZABCUVWxyzabcuvw–	PLC RESET	6 – 5
MP410.3	Axis key IV		
MP410.4	Axis key V		
MP420.0-8	Hirth coupling for axes 1 to 9 Input: 0: No Hirth coupling 1: Hirth coupling	PLC	6 – 398
MP430.0-8	Prescribed increment for Hirth coupling Input: 0.0000 to 30.0000 [°]	PLC	6 – 398
MP710.1-8	Backlash compensation for axes 1 to 9 Input: -1.0000 to +1.0000 [mm] or [°]	PLC	6 – 36
MP711.0-8	Height of the spikes during circular movement (only analog) for axes 1 to 9 Input: -1.0000 000 to +1.0000 999 [mm] (digital: 0)	PLC RUN	6 – 47
MP712.0-8	Compensation value per control loop cycle time for axes 1 to 9 Input: 0.000 000 to 99.999 999 [mm] (digital: 0)	PLC RUN	6 – 47
MP715.0-8	With M105, height of the spikes during circular movement (only analog) for axes 1 to 9 Input: -1.0000 000 to +1.0000 999 [mm] (digital: 0)	PLC RUN	6 – 47
MP716.0-8	With M105, compensation value per control loop cycle time for axes 1 to 9 Input: 0.000 000 to 99.999 999 [mm] (digital: 0)	PLC RUN	6 – 47
MP720.0-8	Linear axis error compensation for axes 1 to 9 Input: -1 000 to +1.000 [mm/m]	PLC	6 – 38

MP	Function and input	Software version and behavior	Page
MP730	Selection of linear/nonlinear axis error compensation Format: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Linear axis error compensation 1: Nonlinear axis error compensation	PLC	6 – 38, 6 – 43
MP750.0-8	Backlash in axes 1 to 9 Input: -1.0000 to +1.0000 [mm] or [°]	PLC	6 – 37
MP752.0-8	Compensation time for backlash in axes 1 to 9 Input: 0 to 1000 [ms]	PLC	6 – 37
MP810.0-8	Display mode for rotary axes and PLC auxiliary axes in axes 1 to 9 Input: 0.0000 to 99 999.9999 [°] 0: Display +/-99 999.9999 1: Modulo value for display	PLC REF	6 – 232
MP812	Activate software limit switches for tilting axes with modulo display, M94 and encoders with EnDat interface Input: %xxxxxxxxx 0: Software limit switch not active 1: Software limit switch active	280 476-07 RESET	6 – 232
MP850.0-8	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	6 – 87
MP855.0-8	Synchronization monitoring for axes 1 to 9 Input: 0 to 100.0000 [mm] 0: Monitoring not active	PLC	6 – 89
MP860.0-8	Datum for synchronization control for axes 1 to 9 Input: 0: Datum at position after switch-on 1: Datum at reference marks 2: Axis is torque slave axis	280 474-04 PLC	6 – 89, 6 – 94

MP	Function and input	Software version and behavior	Page
MP910.0-8	Positive software limit switches, traverse range 1 (default setting after power on) Input: -99 999.9999 to +99 999 [mm] or [°]	PLC	6 – 25
MP911.0-8	Positive software limit switches, traverse range 2 Input: -99 999.9999 to +99 999 [mm] or [°]	PLC	6 – 25
MP912.0-8	Positive software limit switches, traverse range 3 Input: -99 999.9999 to +99 999 [mm] or [°]	PLC	6 – 25
MP920.0-8	Negative software limit switches, traverse range 1 (default setting after power on) Input: -99 999.9999 to +99 999 [mm] or [°]	PLC	6 – 25
MP921.0-8	Negative software limit switches, traverse range 2 Input: -99 999.9999 to +99 999 [mm] or [°]	PLC	6 – 25
MP922.0-8	Negative software limit switches, traverse range 3 Input: -99 999.9999 to +99 999 [mm] or [°]	PLC	6 – 25
MP950.0-8	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	6 – 224
MP951.0-8	Simulating tool change position for TOOL-CALL during block scan for axes 1 to 9 Input: -99 999.9999 to +99 999.9999 [mm] or [°]	PLC RUN	6 – 295
MP960.0-8	Machine datum for axes 1 to 9 Input: -99 999.9999 to +99 999.999 [mm] or [°] Values with respect to the scale reference point	PLC REF	6 – 102, 6 – 224

4.3.2 Positioning

MP	Function and input	Software version and behavior	Page
MP1010.0-8	Rapid traverse in axes 1 to 9 Input: 10 to 300 000 [mm/min]	PLC	6 – 128
MP1020.0-8	Manual feed rate for axes 1 to 9 Input: 10 to 300 000 [mm/min]	PLC	6 – 128
MP1030.0-8	Positioning window Input: 0.0001 to 2.0000 [mm]	PLC	6 – 162
MP1040	Analog axes: Polarity of nominal value voltage Digital axes: Algebraic sign of the nominal speed value Format: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Positive 1: Negative		6 – 10
MP1050.0-8	Analog axes: Analog voltage for rapid traverse in axes 1 to 9 Input: 1 000 to 9 000 [V] Digital axes: without function Input: 1	PLC	6 – 128
MP1060.0-8	Acceleration for axes 1 to 9 Input: 0.001 to 30.000 [m/s or 1000°/s]	PLC	6 – 118
MP1070	Radial acceleration Input: 0.001 to 30.000 [m/s or 1000°/s]	PLC RUN	6 – 152
MP1080.0-8	Analog axes: Integral factor for offset adjustment for axes 1 to 9 Input: Enter 0 to 65 535 Input: 0	PLC RUN	6 – 151
MP1087.0-8	Max. permissible axis-specific jerk for Manual mode Input: 0.1 to 1000.0 [m/s or 1000°/s]	280 476-10 PLC RUN	6 – 118
MP1089.0-8	Max. permissible axis-specific jerk for Pass Over Reference Point mode Input: 0.1 to 1000.0 [m/s or 1000°/s]	280 476-03 PLC RUN	6 – 118

MP	Function and input	Software version and behavior	Page
MP1090	Maximum permissible jerk on the tool path Input: 0.1 to 1000.0 [m/s or 1000°/s]	PLC RUN	6 – 118
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 – 118
MP1094	HSC filter Input: 0: HSC filter inactive 0.1 to 166.0: Cutoff frequency for HSC filter	280 474-07	6 – 118
MP1095	Nominal position value filter Input: 0: Single filter 1: Double filter	PLC RUN	6 – 118
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 – 118, 6 – 153
MP1097.0-8	Max. permissible axis-specific jerk (single/HSC filter) Input: 0.1 to 1000.0 [m/s or 1000°/s]	PLC RUN	6 – 118
MP1098.0-8	Max. permissible axis-specific jerk (double/HSC filter) Input: 0.1 to 1000.0 [m/s or 1000°/s]	PLC RUN	6 – 118
MP1099	Minimum filter order Input: 0 to 20	PLC RUN	6 – 118
MP1099.0	Minimum filter configuration for single filter (MP1095 = 0)		
MP1099.1	Minimum filter configuration for double filter (MP1095 = 1)		
MP1110.0-8	Standstill monitoring for axes 1 to 9 Input: 0.0010 to 30.0000 [mm]	PLC	6 – 162
MP1140.0-8	Threshold from which movement monitoring is effective for axes 1 to 9 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 – 161

MP	Function and input	Software version and behavior	Page
MP1150	Delay time for erasing the nominal velocity value with the erasable error message: **EXCESSIVE SERVO LAG IN <AXIS>** Input: 0 to 65.535 [s] Recommended: 0 s	PLC RUN	6 – 130, 6 – 157, 6 – 160
MP1150.0	Delay time for erasing the nominal velocity value with the erasable error message: **EXCESSIVE SERVO LAG IN <AXIS>** Input: 0 to 65.535 [s] Recommended: 0	280 476-01	
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 off 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]		
MP1152	Interrogation of I3 "Control-is-ready signal acknowledgement" Input: 0: I3 is passed on directly to the NC 1: I3 is processed by the PLC before being passed on to the NC	280 476-21	6 – 173
MP1220	Analog axes: automatic cyclic offset adjustment Input: 0 to 65 536 [s] 0: No automatic adjustment	from 280 474-07 no longer PLC RUN	6 – 151
MP1320	Direction for traversing the reference marks Format: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Positive 1: Negative	PLC	6 – 102
MP1330.0-8	Velocity for traversing the reference marks for axes 1 to 9 Input: 80 to 300 000 [mm/min]	PLC RUN	6 – 102
MP1331.0-8	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 – 102

MP	Function and input	Software version and behavior	Page
MP1340.0-8	Sequence for traversing the reference marks Input: 0: No evaluation of reference marks 1: Axis X 2: Axis Y 3: Axis Z 4: Axis 4 5: Axis 5 6: Axis 6 7: Axis 7 8: Axis 8 9: Axis 9	PLC REF	6 – 102
MP1350.0-8	Type of reference mark traverse Input: 0: Position encoder with distance-coded reference marks (old routine) 1: Position encoder with one reference mark 2: Special type (length measurement with ROD) 3: Position encoder with distance-coded reference marks (new routine) 4: Same as 3 except that two additional reference pulses are evaluated 5: Encoder with EnDat interface	PLC REF	6 – 102
MP1390	Velocity feedforward in the POSITIONING WITH MANUAL DATA INPUT, PROGRAM RUN SINGLE BLOCK and PROGRAM RUN FULL SEQUENCE operating modes Input: 0: Operation with velocity feedforward control 1: Operation with following error (lag)	from 280 474-07 no longer PLC	6 – 121
MP1391	Velocity feedforward control in the MANUAL and HANDWHEEL operating modes Format: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Operation with following error (lag) 1: Operation with velocity feedforward control	PLC RUN	6 – 49, 6 – 122
MP1392	Velocity feedforward in the POSITIONING WITH MANUAL DATA INPUT, PROGRAM RUN SINGLE BLOCK and PROGRAM RUN FULL SEQUENCE operating modes Format: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Operation with following error (lag) 1: Operation with velocity feedforward control	280 474-07 PLC RUN	6 – 121
MP1396.0-8	Feedback control with velocity semifeedforward for axes 1 to 9 Input: 0.001 to 0.999 1: Velocity feedforward control	280 476-09 PLC RUN	6 – 127

4.3.3 Operation with Velocity Feedforward Control

MP	Function and input	Software version and behavior	Page
MP1410.0-8	Position monitoring for operation with velocity feedforward (erasable) for axes 1 to 9 Input: 0.0010 to 30.0000 [mm] Recommended: 0.5 mm	PLC	6 – 159
MP1420.0-8	Position monitoring for operation with velocity feedforward (EMERGENCY STOP) for axes 1 to 9 Input: 0.0010 to 30.0000 [mm] Recommended: 2 mm	PLC	6 – 159
MP1510.0-8	k_V factor for velocity feedforward for axes 1 to 9 Input: 0.100 to 1000.000 [(m/min)/mm]	PLC RUN	6 – 125
MP1511.0-8	Factor for static friction compensation for axes 1 to 9 Input: 0 to 16 777 215 [s]	PLC RUN	6 – 49
MP1512.0-8	Limit to the amount of static friction compensation for axes 1 to 9 Input: 0 to 16 777 215 [counting steps]	PLC RUN	6 – 49
MP1513.0-8	Feed-rate limitation for static friction compensation for axes 1 to 9 Input: 0 to 300 000 [mm/min]	PLC RUN	6 – 49
MP1515.0-8	k_V factor for velocity feedforward effective after M105 for axes 1 to 9 Input: 0.100 to 20.000 [(m/min)/mm]	PLC RUN	6 – 125
MP1516.0-8	k_V factor for velocity semifeedforward for axes 1 to 9 Input: 0.100 to 20.000 [(m/min)/mm]	PLC RUN	6 – 127
MP1521	Transient response during acceleration and deceleration Input: 1 to 255 [ms] 0: Function inactive	PLC	6 – 118

4.3.4 Operation with Following Error (Servo Lag)

MP	Function and input	Software version and behavior	Page
MP1710.0-8	Position monitoring for operation with following error (erasable) for axes 1 to 9 Input: 0.0000 to 300.0000 [mm] Recommended: 1.2 · following error	PLC	6 – 159
MP1720.0-8	Position monitoring for operation with following error (EMERGENCY STOP) for axes 1 to 9 Input: 0.0000 to 300.0000 [mm] Recommended: 1.4 · following error	PLC	6 – 159
MP1810.0-8	k_V factor for operation with following error for axes 1 to 9 Input: 0.100 to 20.000 [(m/min)/mm]	PLC	6 – 123
MP1815.0-8	k_V factor for operation with following error effective after M105 for axes 1 to 9 Input: 0.100 to 20.000 [(m/min)/mm]	PLC	6 – 123
MP1820.0-8	Multiplier for the k_V factor for axes 1 to 9 Input: 0.001 to 1.00000	PLC	6 – 129
MP1830.0-8	Kink point for axes 1 to 9 Input: 0.000 to 100.000 [%]	PLC	6 – 129

4.3.5 Integrated Speed and Current Control

MP	Function and input	Software version and behavior	Page
MP2000	Type of drive Input: 0: Output of nominal speed value (analog axis) 1: Output of current pulse (digital axis)	RESET	6 – 462
MP2001	Type of drive for spindle Input: 0: Output of nominal speed value (analog spindle) 1: Output of current pulses (digital spindle)		6 – 462
MP2020.0-8	Traverse per motor revolution for axes 1 to 9 Input: Analog axes: without function Digital axes: 0 to 100.000 [mm] or [°]		6 – 161
MP2100.0-8	Type of power module for axes 1 to 9 Input: Name of the selected power module (entered by the TNC)		6 – 465
MP2101	Model of power module for the spindle Input: Name of the selected power stage (is entered by the TNC)		6 – 465
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]	280 476-09	6 – 144
MP2180.0-8	PWM frequency of the axes Input: 3000 to 7000 [Hz] 0 = 5000 Hz (for HEIDENHAIN inverters)	280 472-07	6 – 467
MP2181	PWM frequency of the spindle Input: 3000 to 7000 [Hz] 0 = 5000 Hz (for HEIDENHAIN inverters)	280 476-19	6 – 467
MP2190	dc link voltage U_Z Input: 0 to 10 000 [V] HEIDENHAIN inverters: UE 2xx, UE 2xxB, UV 130: 565 V UV 120, UV 140, UV 150, UR 2xx: 650 V		6 – 467
MP2191	Braking the first spindle for an Emergency Stop Input: 0: With monitoring of the maximum braking current 1: Without monitoring of the maximum braking current	280 474-03	6 – 198
MP2200.0-8	Motor model for axes 1 to 9 Input: Name of the selected motor (is entered by the TNC)		6 – 465
MP2201	Motor model for the spindle Input: Name of the selected motor (is entered by the TNC)		6 – 465

MP	Function and input	Software version and behavior	Page
MP2221	Current and speed controller monitoring functions Format: %xxx Input: Bit 0 – Monitoring the reference mark 0: Monitoring active 1: Monitoring inactive Bit 1 – Monitoring the rotational direction (only with spindle DSP) 0: Monitoring active 1: Monitoring inactive Bit 2 – 0: Do not monitor $\overline{\text{ERR-IZ}}$ signal, or inverter does not supply this signal 1: Monitor $\overline{\text{ERR-IZ}}$ signal	280 474-12 PLC	6 – 183
MP2302.0-8	Reference value for I^2t monitoring of feed motors for axes 1 to 9 Input: 0 to 1000.000 [· rated current of motor] 0: I^2t monitoring of feed motors turned off 1: Rated current of motor as reference value		6 – 167
MP2303	Reference value for I^2t monitoring of spindle motor Input: 0 to 1000.000 [· rated current of motor] 0: I^2t monitoring of spindle motors turned off 1: Rated current of motor as reference value		6 – 167
MP2312.0-8	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 – 170
MP2313	Reference value for utilization display of the spindle motor Input: 0 to 1000.000 [· rated current of motor] 0 or 1: Reference value is rated current of motor		6 – 170
MP2340.0-8	Speed starting from which the field angle begins to shift on synchronous motors for the axes 1 to 9 Input: 0 to 100 000 min 0: No field angle offset		6 – 466
MP2350.0-8	Field-angle offset on synchronous motors for axes 1 to 9 Input: 0 to 60 [°]		6 – 466
MP2360.0-8	Time constant for braking axes 1 to 8 or the second spindle in an emergency stop Input: 0.01 to 5.00 [s] 0: Function inactive	280 474-10	6 – 198

MP	Function and input	Software version and behavior	Page
MP2361	Time constant for braking the first spindle in an emergency stop Input: 0.01 to 5.00 [s] 0: Function inactive	280 474-10	6 – 198
MP2391 MP2391.0 MP2391.1	Maximum power for braking the first spindle in an emergency stop Input: 0.1 to 3000.000 [kW] 0: Braking power is not limited Wye connection Delta connection	280 474-10	6 – 200
MP2393 MP2393.0 MP2393.1	Power limiting of spindle motor Input: 0: No power limit 0.1 to 3000.000 [kW] Wye connection Delta connection	280 476-01	6 – 192
MP2395 MP2395.0 MP2395.1	Maximum power for braking the first spindle in a power failure Input: 0.1 to 3000.000 [kW] 0: Braking power is not limited Wye connection Delta connection	280 476-09	6 – 200
MP2397 MP2397.0 MP2397.1	Maximum torque of the spindle motor Input: 0: No torque limiting 0.1 to 3000.000 [Nm] Wye connection Delta connection	280 476-13 PLC	6 – 192
MP2400.0-8	Gain for current controller at standstill for axes 1 to 9 Input: 0.00 to 9 999.00 [V/A] 0: Controller disable		6 – 148
MP2401	Gain for the spindle current controller at standstill Input: 0.00 to 9999.99 [V/A] 0: Controller disable		6 – 213
MP2402.0-8	Gain for current controller at maximum speed for axes 1 to 9 Input: 0.00 to 9999.99 [V/A] 0: Value from MP2400.x		6 – 148
MP2403	Gain for the spindle current controller at maximum speed Input: 0.00 to 9999.99 [V/A] 0: Value from MP2401		6 – 213

MP	Function and input	Software version and behavior	Page
MP2421.0-1	Proportional factor of the spindle current controller for wye and delta connection Input: 0.00 to 9999.99 [VA]		6 – 213
MP2431.0-1	Integral factor of the spindle current controller for wye and delta connection Input: 0.00 to 9999.99 [V/As]		6 – 213
MP2500.0-8	Proportional factor of the speed controller for axes 1 to 9 Input: 0 to 1 000 000.000 [As]	PLC	6 – 135
MP2501.0-1	Proportional factor of the spindle speed controller for wye and delta connection Input: 0 to 100 000 000.000 [As]	PLC	6 – 212
MP2510.0-8	Integral factor of the speed controller for axes 1 to 9 Input: 0 to 100 000 000 [A]	PLC	6 – 135
MP2511.0-1	Integral-action factor of the spindle speed controller for wye and delta connection Input: 0 to 100 000 000 [A]	PLC	6 – 212
MP2512.0-8	Limiting the integral-action component of the speed controller for axes 1 to 9 Input: 0.000 to 30.000 [s] (realistically: 0.1 to 2.0)	PLC	6 – 49, 6 – 139
MP2520.0-8	Differential factor of the speed controller for axes 1 to 8 Input: 0 to 1.0000 [As]	PLC	6 – 136
MP2521.0-1	Differential factor of the spindle speed controller for wye and delta connection Input: 0 to 1.0000 [As]	PLC	6 – 212
MP2530.0-8	PT_2 element of the speed controller (2nd-order delay) for axes 1 to 8 Input: 0 to 1.0000 [s]	PLC	6 – 137
MP2531.0-1	PT_2 second-order time delay element of the speed controller for the first spindle for wye and delta connection Input: 0 to 1.0000 [s] 0 = 0.001 s	PLC	6 – 212
MP2540.0-8	Band-rejection filter damping for axes 1 to 8 Input: 0.0 to 18.0 [dB]	PLC	6 – 137
MP2541	Band-rejection filter damping of the spindle Input: 0.0 to 18.0 [dB]	PLC	6 – 212

MP	Function and input	Software version and behavior	Page
MP2550.0-8	Band-rejection filter center frequency for axes 1 to 8 Input: 0.0 to 999.9 [Hz]	PLC	6 – 137
MP2551	Band-rejection filter central frequency of the spindle Input: 0.0 to 999.9 [Hz]	PLC	6 – 212
MP2560.0-8	Low-pass filter for axes 1 to 9 Input: 0: No low-pass filter 1: 1st-order low-pass filter 2: 2nd-order low-pass filter	280 474-07 PLC	6 – 136
MP2561	Low-pass filter spindle Input: 0: No low-pass filter 1: 1st-order low-pass filter 2: 2nd-order low-pass filter	280 474-07 PLC	6 – 212
MP2590.0-8	Braking ramp for axes 1 to 8 or the second spindle in an emergency stop Input: 0.1 to 999.9 [rpm/ms] 0: Function inactive	280 476-03 PLC RUN	6 – 149, 6 – 198
MP2591	Braking ramp for the first spindle in an emergency stop Input: 0.1 to 999.9 [rpm/ms] 0: Function inactive	280 476-03 PLC RUN	6 – 198
MP2600.0-8	Acceleration feedforward for axes 1 to 9 Input: 0 to 30.000 [A/(rev/s)]	PLC	6 – 139
MP2602.0-8	IPC time constant T_1 for axes 1 to 9 Input: 0.0001 to 1.0000 [s] 0: IPC inactive	280 476-05 PLC RUN	6 – 141
MP2604.0-8	IPC time constant T_2 for axes 1 to 9 Input: 0.0001 to 1.0000 [s] 0: IPC inactive	280 476-05 PLC RUN	6 – 141
MP2606.0-8	Following error during the jerk phase for axes 1 to 9 Input: 0.000 to 10 000	280 476-07 PLC RUN	6 – 141
MP2610.0-8	Friction compensation at rated speed for axes 1 to 9 (effective only with velocity feedforward control) Input: 0 to 30.0000 [A] 0: No friction compensation (or analog axis)	PLC	6 – 50

MP	Function and input	Software version and behavior	Page
MP2612.0-8	Delay of friction compensation at rated speed for axes 1 to 9 (effective only with velocity feedforward control) Input: 0.0000 to 1.0000 [s] (typically: 0.015 s) 0: No friction compensation (or analog axis)	PLC	6 – 50
MP2620.0-8	Friction compensation at rated speed for axes 1 to 9 Input: 0 to 30.0000 [A] 0: No friction compensation (or analog axis)	PLC	6 – 50
MP2630.0-8	Holding current for axes 1 to 9 Input: -30.000 to +30.000 [A]	PLC	6 – 142
MP2800.0-8	Movement monitoring for position and speed for axes 1 to 9 Input: Analog axes: without function Digital axes: 0 to 99 999.999 [mm] 0: No monitoring	PLC	6 – 161
MP2900.0-8	Tensioning torque between master and slave for master-slave torque control (entry for the slave axis) Input: -100.00 to +100.00 [Nm]	280 474-04 PLC	6 – 97
MP2910.0-8	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)]	280 474-04 PLC	6 – 97
MP2920.0-8	Factor for variable torque distribution of the torque-master-slave control (entry for the slave axis) Input: 0.000 to 100.000 1: Master and slave axes have identical motors	280 474-04 PLC	6 – 97
MP2930.0-8	Speed rating factor of the torque-master-slave control (entry for the slave axis) Input: -100.00 to +100.00 [%]	280 474-04 PLC	6 – 97

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 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	PLC RESET	6 – 180
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)	RESET	6 – 452
MP3012	Feed rate from output of an analog voltage of 10 V, MP3011 = 1 Input: 0 to 300 000 [mm/min]		6 – 452
MP3013.x	Characteristic curve kink points (velocity) for output of the analog voltage with M202 Input: 10 to 300 000 [mm/min]	PLC RUN	6 – 453
MP3014.x	Characteristic curve kink points (voltage) for output of the analog voltage with M202 Input: 0.000 to 9.999 [V]	PLC RUN	6 – 453
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	6 – 194

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	6 – 188, 6 – 455
MP3120	Zero speed permitted Input: 0: S = 0 allowed 1: S = 0 not allowed	PLC	6 – 187
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 – 186
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 – 186
MP3142	Line count of the spindle position encoder Input: 100 to 9 999 [lines]	PLC RUN	6 – 181
MP3143	Mounting configuration of the spindle position encoder Input: 0: Position encoder directly on the first spindle 1: Position encoder via transmission (transmission in MP3450.x and MP3451.x) X30 pin 1: reference pulse 2: Position encoder via transmission (transmission 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 – 181
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 – 187

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 – 187, 6 – 188
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	Limitation for spindle speed override Input: 0 to 150 [%]	PLC RUN	6 – 191
MP3310.0	Upper limit		
MP3310.1	Lower limit		
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 – 185
MP3412	Multiplication factor for MP3411.x Input: 0.000 to 1.999	PLC RUN	6 – 186, 6 – 202, 6 – 207, 6 – 211
MP3412.0	With M05		
MP3412.1	With oriented spindle stop		
MP3412.2	With tapping with floating tap holder		
MP3412.3	With rigid tapping		
MP3415	Overshoot behavior of the spindle with M03, M04 and M05 Input: 0 to 1000 [ms]	PLC RUN	6 – 185, 6 – 202, 6 – 207, 6 – 211
MP3415.0	With M03, M04 and M05		
MP3415.1	For oriented spindle stop		
MP3415.2	With tapping		
MP3415.3	With rigid tapping		
MP3420	Spindle positioning window Input: 0 to 360.0000 [°]	PLC RUN	6 – 202
MP3430	Deviation of the reference mark from the desired position (spindle preset) Input: 0 to 360 [°]	PLC RUN	6 – 202

MP	Function and input	Software version and behavior	Page
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 – 202
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 – 181
MP3451.0-7	Number of spindle revolutions for gear ranges 1 to 8 Input: 0 to 65 535 0: No transmission	PLC RUN	6 – 181
MP3510.0-7	Rated speed for the gear ranges 1 to 8 Input: 0 to 99 999.999 [rpm]	PLC	6 – 187
MP3515.0-7	Maximum spindle speed for gear ranges 1 to 8 Input: 0 to 99 999.999 [rpm]	PLC	6 – 191
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 – 205 6 – 202

4.3.7 Integral PLC

MP	Function and input	Software version and behavior	Page
MP4020	PLC compatibility with TNC 415 / TNC 425 Format: %xxxxxxxxxxx Input: Bit 0 = Change words (W1024 and subsequent) into markers Bit 1 = Change markers (4000 and subsequent) into markers (2000 and subsequent) Bit 2 = Change configuration bits from MP4310 into markers (M2192 to M2239 and M3200 to M3263) Bit 3 = Error markers are available Bit 4 = Nonvolatile markers in the range M1000 to M1999 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: Values with a change rate of 1 K/s are accepted 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, before being transferred to the NC, is first processed by an active PLC window Bit 10 – Behavior of a disabled key 0: Disabled key only works on the active PLC window 1: Disabled key works on neither the active PLC window, nor on the NC	RESET	6 – 394, 6 – 130, 6 – 216, 6 – 338
MP4030	Assignment of physical to logical PL Input: 0: First logical PL 1: Second logical PL 2: Third logical PL 3: Fourth logical PL	280 476-01 PLC	6 – 388
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
MP4060.0-8	Traverse distance for lubrication of axes 1 to 9 Input: 0 to 99 999.999 [mm] or [°]	PLC RUN	6 – 26
MP4070	Compensation amount per PLC cycle for lagged-tracking axis error compensation Input: 0.0001 to 0.005 [mm]	PLC RUN	6 – 44
MP4110.0-47	Timer preset value T0 to T47 Input: 0 to 65 535 [PLC cycle times]		7 – 17
MP4120.0-31	Counter preset for C0 to C31 Input: 0 to 65 535 [PLC cycles]		7 – 20
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] TNC 426 M, TNC 430 M: If you use I32, enter the following values: up to 280 474-11: MP4130.0 = 159 as of 280 474-12: MP4130.0 = 32 as of 280 476-01: MP4130.0 = 32		6 – 157 7 – 21
MP4131.0 MP4131.1 MP4131.2-5	Activation criterion for fast PLC input for switching off the monitoring functions Reserved Condition for activating fast PLC inputs Input: 0: Activation at low level 1: Activation at high level		6 – 157 7 – 21
MP4210.0-47	Setting a number in the PLC (D768 to D956) Input: -99 999.9999 to +99 999.9999		6 – 205
MP4220.0-4	Setting a number in the PLC (W960 to W968) Input: 10 to 30 000		7 – 37
MP4230.0-31	Setting a number in the PLC (Module 9032) Input: -99 999.9999 to +99 999.9999		7 – 37
MP4231.0-31	Setting a number in the PLC (Module 9032) Input: -99 999.9999 to +99 999.9999		7 – 37
MP4310.0-6	Setting a number in the PLC (W976 to W988, M4300 to M4411) Input: 10 to 30 000		7 – 37

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	8 – 15
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: 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 bits Bit 6 = 1, Bit 7 = 1: Length of the stop: 1 bits	PLC, CN123 RUN	8 – 18
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, CN123 RUN	8 – 18
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	8 – 30

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	6 – 346
MP6120	Probing feed rate (triggering touch probe) Input: 1 to 3000 [mm/min]	PLC, CN123 RUN	6 – 349
MP6130	Maximum measuring range Input: 0.001 to 99 999.9999 [mm]	PLC, CN123 RUN	6 – 349
MP6140	Setup clearance over measuring point Input: 0.001 to 99 999.9999 [mm]	PLC, CN123 RUN	6 – 349
MP6150	Rapid traverse in probing cycle (triggering touch probe) Input: 10 to 20 000 [mm/min]	PLC, CN123 RUN	6 – 349
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 by the PLC	PLC, CN123 RUN	6 – 352
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	280 474-08 PLC, CN123 RUN	6 – 350
MP6162	Orientation angle Input: 0 to 359.9999 [°]	280 474-08 PLC, CN123 RUN	6 – 350
MP6163	Minimum difference between the current spindle angle and MP6162 before executing an oriented spindle stop Input: 0 to 3.0000 [°]	280 474-08 PLC, CN123 RUN	6 – 350
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	280 476-10 PLC, CN123 RUN	6 – 350
MP6170	Number of measurements in a programmed measurement (touch probe block) Input: 1 to 3	PLC, CN123 RUN	6 – 353

MP	Function and input	Software version and behavior	Page
MP6171	Confidence range for programmed measurement (MP6170 > 1) Input: 0.002 to 0.999 [mm]	PLC, CN123 RUN	6 – 353
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]	280 472-05 PLC, CN123	6 – 352
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]	280 472-05 PLC, CN123	6 – 352
MP6181.0	X coordinate		
MP6181.1	Y coordinate		
MP6181.2	Z coordinate		
MP6182	Coordinates 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]	280 472-05 PLC, CN123	6 – 353
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]	280 472-05 PLC, CN123	6 – 353

4.3.10 Digitizing with TS (option)

MP	Function and input	Software version and behavior	Page
MP6200	Selection of triggering or measuring touch probe (only with "digitizing with measuring touch probe" option) Input: 0: Triggering touch probe (e.g. TS 220) 1: Measuring touch probe	PLC, CN123	6 – 346, 6 – 368
MP6210	Number of oscillations in normal direction per second Input: 0 to 65.535 [1/s]	PLC, CN123 RUN	6 – 361
MP6220	Traverse distance for lubrication of the touch probe axis at line end Input: 0 to 999.999 [mm]	PLC, CN123 RUN	6 – 363
MP6221	Time after which the probe axis must be lubricated Input: 0 to 65 535 [mm]	PLC, CN123 RUN	6 – 363
MP6230	Feed rate in normal direction Input: 0 to 1 000 [mm/min]	PLC, CN123 RUN	6 – 361
MP6240	Maximum deflection of the stylus Input: 0 to 10 000 [mm]	PLC, CN123 RUN	6 – 361
MP6260	Output of M90 in NC blocks with digitized data Input: 0: No output of M90 1: Output of M90 in every NC block	PLC, CN123 RUN	6 – 361
MP6270	Rounding of decimal places Input: 0: Output in 0.001-mm steps (1 µm) 0: Output in 0.01-mm steps (10 µm) 2: Output in 0.0001-mm steps (0.1 µm)	PLC, CN123 RUN	6 – 361

4.3.11 Digitizing with Measuring Touch Probe (option)

MP	Function and input	Software version and behavior	Page
MP6310	Deflection depth of the stylus (measuring touch probe) Input: 0.1000 to 2.0000 [mm]	PLC, CN123	6 – 368
MP6320	Counting direction of encoder output signals (measuring touch probe) Format: %xxx Input: Bits 0 to 2 represent axes X to Z 0: Positive 1: Negative	CN123	6 – 368
MP6321	Measuring the center offset while calibrating the measuring touch probe Input: 0: Calibration with measurement of the Center offset 1: Calibration without measuring the Center offset	CN123	6 – 353, 6 – 369
MP6322.0-2	Assignment of the touch probe axes to the machine axes X, Y and Z Input: 0: Touch probe axis X 1: Touch probe axis Y 2: Touch probe axis Z	CN123	6 – 369
MP6330	Maximum deflection of the stylus (measuring touch probe) Input: 0.1 to 4.000 [mm]	CN123	6 – 369
MP6350	Feed rate for positioning to the MIN point and approaching the contour (measuring touch probe) Input: 1 to 3000 [mm/min]	CN123	6 – 369
MP6360	Probing feed rate (measuring touch probe) Input: 1 to 3000 [mm/min]	CN123	6 – 349, 6 – 369
MP6361	Rapid traverse in probing cycle (measuring touch probe) Input: 10 to 10 000 [mm/min]	CN123	6 – 350, 6 – 369
MP6362	Feed rate reduction, if the stylus of the measuring touch probe is deflected to the side Input: 0: Feed rate reduction not active 1: Feed rate reduction active	PLC, CN123	6 – 369
MP6370	Radial acceleration when digitizing with measuring touch probe Input: 0.001 to 3.000 [m/s] Recommended input value: 0.1	PLC, CN123 RUN	6 – 369
MP6390	Target window for contour line Input: 0.1000 to 4.0000 [mm]	PLC, CN123	6 – 369

4.3.12 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: Disabled 1: Not disabled 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 stationary spindle (bit 1=1) 1: Tool length measurement with rotating spindle, only if in the tool table a Tool offset for radius (TT: R-OFFS) is entered 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, when exceeding the breaking tolerance, is not stopped 1: When exceeding the breaking tolerance, the NC program is stopped and the error message “tool broken” is displayed Bit 6 – NC stop during “tool measurement” 0: The NC program, when exceeding the breaking tolerance, is not stopped 1: When exceeding the breaking tolerance, the NC program is stopped and the error message “touch point inaccessible” is displayed	PLC	6 – 370, 6 – 371, 6 – 373, 6 – 375, 6 – 377

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 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 principle 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 a different tilt position Bit 14 – Tool measurement with number of teeth 0 0: Tool measurement with rotating spindle 1: Tool measurement with stationary spindle	PLC	6 – 370, 6 – 371, 6 – 373, 6 – 375, 6 – 377

MP	Function and input	Software version and behavior	Page
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 in the angle reference axis (0° axis) 3: Negative probing direction in the +90° axis	PLC, CN123	6 – 372
MP6505.0	Traverse range 1		
MP6505.1	Traverse range 2		
MP6505.2	Traverse range 3		
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, CN123	6 – 375
MP6510	Permissible measuring error for tool measurement with rotating tool Input: 0.002 to 0.999 [mm]	PLC, CN123	6 – 375
MP6510.0	First measurement error		
MP6510.1	Second measurement error		
MP6520	Probing feed rate for tool measurement with non-rotating tool Input: 1 to 3000 [mm/min]	PLC, CN123 RUN	6 – 376
MP6530	Distance from the tool end to the top of the probe contact during tool radius measurement for 3 traverse ranges Input: 0.001 to 99.9999 [mm]	PLC, CN123	6 – 372
MP6530.0	Traverse range 1		
MP6530.1	Traverse range 2		
MP6530.2	Traverse range 3		
MP6531	Diameter or edge length of the TT 130 probe contact for 3 traverse ranges Input: 0.001 to 99.9999 [mm]	PLC	6 – 373
MP6531.0	Traverse range 1		
MP6531.1	Traverse range 2		
MP6531.2	Traverse range 3		

MP	Function and input	Software version and behavior	Page
MP6540	Safety zone around the probe contact of the TT 130 for pre-positioning Input: 0.001 to 99 999.9999 [mm]	PLC, CN123	6 – 372
MP6540.0	Safety clearance in tool axis direction	280 474-03	
MP6540.1	Safety clearance in the plane perpendicular to the tool axis	280 474-03	
MP6550	Rapid traverse in probing cycle for TT 130 Input: 10 to 20 000 [mm/min]		6 – 372
MP6560	M function for spindle orientation during individual tooth measurement Input: -1: Spindle orientation directly by NC 0: Function inactive 1 to 999: Number of the M function for spindle orientation by the PLC	PLC, CN123 RUN	6 – 371

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, CN123	6 – 376
MP6572	Maximum permissible speed during tool measurement Input: 1 to 1000 [rpm] 0: 1000 [rpm]	280 476-09 PLC, CN123 RUN	6 – 376
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, CN123	6 – 373
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, CN123	6 – 373
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, CN123	6 – 373
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	280 476-01 PLC, CN123	6 – 376
MP6586 MP6586.0-5	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 °] Axes A to W	280 476-01 PLC, CN123	6 – 376

4.3.13 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 – 207
MP7110.1	Maximum for feed rate override during tapping Input: 0 to 150 [%]		6 – 207
MP7120.0	Dwell time for reversal of spindle rotational direction Input: 0 to 65.535 [s]	PLC RUN	6 – 207
MP7120.1	Advanced switching time of the spindle during tapping with coded spindle-speed output Input: 0 to 65.535 [s]		6 – 208
MP7120.2	Spindle slow-down time after reaching the hole depth Input: 0 to 65.535 [s]		6 – 207
MP7130	Run-in behavior of the spindle during rigid tapping Input: 0.001 to 10 [°/min]	PLC	6 – 211
MP7150	Positioning window of the tool axis during rigid tapping Input: 0.0001 to 2 [mm]	PLC	6 – 211
MP7160	Spindle response during Cycle 17 and 18 Format: %xxxx Input: Bit 0 – Oriented spindle stop with Cycle 17 0: Before execution of Cycle 17 spindle orientation 1: Before execution of Cycle 17 no spindle orientation Bit 1 – Spindle speed 0: Spindle speed is not limited 1: Spindle speed is limited so that about 1/3 of the time the spindle runs at constant speed Bit 2 – Spindle in position feedback control 0: Spindle operated without position feedback control 1: Spindle operated with position feedback control Bit 3 – IPC and acceleration feedforward control 0: Active 1: Not active	PLC, CN123 RUN	6 – 211

4.3.14 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	6 – 324
MP7212	Power interrupted message Input: 0: Acknowledge message with CE key 1: Message does not appear	PLC, CN123 RUN	6 – 327
MP7220	Block number increment for ISO programs Input: 0 to 250	PLC, CN123 RUN	6 – 301
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, CN123 RUN	6 – 300
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		6 – 300

MP	Function and input	Software version and behavior	Page
MP7226.0	Size of the pallet table Input: 0 to 255 [lines]	PLC, CN123 RUN	6 – 309
MP7226.1	Size of the datum table Input: 0 to 255 [lines]		6 – 302
MP7229	Depiction of the NC program	PLC, CN123	6 – 245
MP7229.0	Line number for program testing Input: 100 to 9999	RUN	
MP7229.1	Program length to which FK blocks are allowed Input: 100 to 9999		
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	PLC, CN123 RUN	6 – 326
MP7230.0	NC conversational language		
MP7230.1	PLC conversational language (user parameters), soft keys for OEM cycles		
MP7230.2	Write PLC error messages		
MP7230.3	Help files		
MP7235	Time difference to Universal Time (Greenwich Mean Time) Input: -23 to +23 [hours] 0: Universal Time (Greenwich Mean Time) 1: Central European Time (CET) 2: Central European daylight-saving time	PLC, CN123 RUN	6 – 332

MP	Function and input	Software version and behavior	Page
MP7237 MP7237.0 MP7237.1 MP7237.2	Displaying and resetting 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 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 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	PLC RUN	6 – 329
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	6 – 329
MP7245	Disabling auxiliary cycles Input: 0: Auxiliary cycles disabled 1: Auxiliary cycles permitted	280 474-09 PLC RUN	6 – 287
MP7246	Disabling paraxial positioning blocks Input: 0: Paraxial positioning block enabled 1: Paraxial positioning block disabled	PLC RUN	6 – 327
MP7260	Number of tools in the tool table Input: 0 to 30 000	CN123	6 – 402
MP7261.0-3	Number of pockets in the tool magazine 1 to 4 Input: 0 to 254	CN123	6 – 402
MP7262	Maximum tool index number for indexed tools Input: 0 to 9	280 474-03 CN123	6 – 417
MP7263	Hiding/showing the POCKET TABLE soft key Format: %x Input: Bit 0 – 0: POCKET TABLE soft key is shown 1: POCKET TABLE soft key is hidden	280 474-04 CN123	6 – 402
MP7266	Elements of the tool table Input: 0: No display 1 to 99: Position in the tool table	CN123	6 – 402

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	6 – 404
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)		
MP7270	Feed-rate display in the MANUAL OPERATION and ELECTRICAL HANDWHEEL operating modes Input: 0: Display of feed rate by pressing an axis direction key (axis-specific feed rate from MP1020) 1: Display of the feed rate also before pressing an axis direction key (smallest value from MP1020 for all axes)	PLC, CN123 RUN	6 – 237
MP7280	Decimal character Input: 0: Decimal comma 1: Decimal point	PLC, CN123 RUN	6 – 326
MP7281	Depiction of the NC program Input: 0: All blocks completely 1: Current block all at once, others line by line 2: All blocks line by line; block when editing all at once	280 476-03 PLC RUN	6 – 245
MP7285	Tool length offset in the tool-axis position display Input: 0: Tool length is not offset 1: Tool length is offset	PLC, CN123 RUN	6 – 231
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°	280 474-09 PLC, CN123 RUN	6 – 231

MP	Function and input	Software version and behavior	Page
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, CN123 RUN	6 – 231
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.	PLC RUN	6 – 5
MP7291.0	Display in traverse range 1		
MP7291.1	Display in traverse range 2		
MP7291.2	Display in traverse range 3		
MP7295	Disabling "datum setting" Format: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Not disabled 1: Disabled	PLC, CN123 RUN	6 – 224
MP7296	"Datum setting" through axis keys Input: 0: Datum can be set by axis keys and by soft keys 1: Datum can be set only by soft key	PLC, CN123 RUN	6 – 224
MP7300	Erasing the status display and Q parameters Input: 0: Status display, Q parameters and tool data are deleted when the program is selected. 1: Status display, Q parameters and tool data are deleted with M02, M30, END PGM, or when a program is selected. 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 or with M02, M30, or END PGM. 4: Status display and Q parameters are deleted when a program is selected. 5: Status display and Q parameters are deleted when a program is selected or with M02, M30, or END PGM. 6: Erase the status display when a program is selected, or with M02, M30, or END PGM. 7: Erase the status display when a program is selected, or with M02, M30, or END PGM.	PLC, CN123 RUN	6 – 241

MP	Function and input	Software version and behavior	Page
MP7310	Graphic display mode Format: %xxxx Input: Bit 0 – Projection in three planes: 0: German-preferred projection 1: US-preferred projection Bit 1 – Rotation of 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	PLC, CN123 RUN	6 – 230
MP7315	Tool radius for graphic simulation without TOOL CALL Input: 0.0000 to 99 999.9999 [mm]	PLC, CN123 RUN	6 – 454
MP7316	Penetration depth of the tool Input: 0.0000 to 99 999.9999 [mm]	PLC, CN123 RUN	6 – 454
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, CN123 RUN	6 – 454
MP7330.0-15	Specifying the user parameters 1 to 16 Input: 0 to 9999.00 (no. of the user parameter)	PLC RUN	6 – 323
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	6 – 323

4.3.15 Colors

MP	Function and input	Software version and behavior	Page
MP7350	Window frames	PLC RUN	6 – 225
MP7351	Error messages	PLC RUN	6 – 225
MP7352 MP7352.0 MP7352.1 MP7352.2	"Machine" operating mode display Background Text for operating mode Dialog	PLC RUN	6 – 225
MP7353 MP7353.0 MP7353.1 MP7353.2	"Programming" operating mode display Background Text for operating mode Dialog	PLC RUN	6 – 225
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	6 – 225
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	6 – 225
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	6 – 226
MP7357 MP7357.0 MP7357.1	"Machine" soft-key display Background Symbols	PLC RUN	6 – 226
MP7358 MP7358.0 MP7358.1	"Programming" soft-key display Background Symbols	PLC RUN	6 – 226
MP7360 MP7360.0 MP7360.1 MP7360.2 MP7360.3 MP7360.4	Graphics: 3-D view Background Top surface Front face Text display in the graphics window Lateral face	PLC RUN	6 – 226

MP	Function and input	Software version and behavior	Page
MP7361 MP7361.0 MP7361.1 MP7361.2 MP7361.3 MP7361.4	Graphics: Projection in three planes Background Top view Front and side view Axis cross and text in the graphic display Cursor	PLC RUN	6 – 226
MP7362 MP7362.0 MP7362.1 MP7362.2 MP7362.3	Additional text display in the graphic window and pocket calculator Background of graphic window and pocket calculator Background of status display and keys of the pocket calculator Status symbols and pocket calculator symbols (c in “cos”) Status values and texts of the pocket calculator (os in “cos”)	PLC RUN	6 – 226
MP7363 MP7363.0 MP7363.1 MP7363.2 MP7363.3 MP7363.4	Programming graphics Background Resolved contour Subprograms and frame for zooming Alternative solutions Unresolved contour	PLC RUN	6 – 226
MP7364 MP7364.0-6 MP7364.7 MP7364.8 MP7364.9	Color of the help illustrations for cycles Colors 1 to 7 of the graphic program used Line color (color 8 of the graphic program) Color for highlighted graphic elements if defined in the help illustration Background	PLC RUN	6 – 226
MP7365 MP7365.0 MP7365.1 MP7365.2 MP7365.3 MP7365.4 MP7365.5 MP7365.6 MP7365.7	Oscilloscope Background Channel 1 Channel 2 Channel 3 Channel 4 Selected channel Grid Cursor and text	PLC RUN	6 – 226

MP	Function and input	Software version and behavior	Page
MP7366	Pop-up window (HELP key, pop-up menus etc.)	PLC	6 – 227
MP7366.0	Background	RUN	
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	PLC	6 – 227
MP7367.0	Background	RUN	
MP7367.1	Color 1		
MP7367.2	Color 2		
MP7367.3	Color 3		
MP7367.4	Color 4		
MP7367.5	Color 5		
MP7367.6-14	Colors 6 to 14		
MP7392	Screen saver Input: 1 to 99 [min] 0: No screen saver	PLC, CN123 RUN	6 – 227

4.3.16 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 principal axes 1: Scaling cycle is effective only in the working plane	PLC, CN123 RUN	6 – 293
MP7411	Tool data in the touch probe block Format: %xx Input: Bit 0 – 0: The calibrated data of the touch probe is used. 1: The current tool data from the last TOOL CALL is used Bit 1 – 0: Only one set of touch probe calibration data 1: More than one set of touch probe calibration data are managed in the tool table	280 476-01 PLC, CN123 RUN	6 – 350
MP7420	Cycles for milling pockets with combined contours Format: %xxxxx Input: Bit 0 – Milling direction for channel milling: 0: Channel milling of the contours for pockets counterclockwise, clockwise for islands 1: Channel milling of the contours for pockets clockwise, 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 only if the programmed contours intersect Bit 3 – Rough-out and channel milling to pocket depth or for every infeed 0: Rough-out and channel milling uninterrupted to pocket depth 1: For each process: first channel milling, then rough-out depending on bit 1 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	PLC, CN123 RUN	6 – 293
MP7430	Overlap factor for pocket milling Input: 0.001 to 1.414	PLC, CN123 RUN	6 – 291

MP	Function and input	Software version and behavior	Page
MP7431	Arc end-point tolerance Input: 0.0001 to 0.016 [mm]	PLC, CN123 RUN	6 – 326
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 acknowledgement of the M function 1: No program stop. Acknowledgement is not waited for. Bit 3 – Switching of k_V factors with M105/M106 0: Function is not in effect 1: Function is in effect Bit 4 – Reduced feed rate in the tool axis with M103 0: Function is not in effect 1: Function is in effect 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, CN123 RUN	6 – 47, 6 – 123, 6 – 125, 6 – 286, 6 – 455
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 suppressed	280 474-04 PLC, CN123 RUN	6 – 286
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: Spindle orientation by the NC	280 476-09 PLC, CN123 RUN	6 – 201

MP	Function and input	Software version and behavior	Page
MP7450	Offsetting the tool change position from MP951.x in block scan Format: %xxxxxxxxx Input: Bits 0 to 8 correspond to axes 1 to 9 0: Do not offset 1: Offset	PLC RUN	6 – 295
MP7451.0-8	Feed rate for returning to the contour for axes 1 to 9 Input: 10 to 300 000 [mm/min]	PLC	6 – 295
MP7460	Angle for constant contour speed at corners Input: 0.0001 to 179.9999 [°]	PLC, CN123 RUN	6 – 155
MP7470	Maximum contouring tool feed rate at 100% override Input: 0 to 300 000 [mm/min] 0: No limitation	PLC, CN123 RUN	–
MP7471	Maximum velocity of the principle axes during compensating movements through M128 Input: 0 to 300 000 [mm/min]	280 472-05 PLC, CN123 RUN	6 – 84
MP7475	Reference for datum table Input: 0: Reference is workpiece datum 1: Reference is machine datum (MP960.x)	PLC, CN123 RUN	6 – 302

MP	Function and input	Software version and behavior	Page
MP7480 MP7480.0	Output of the tool or pocket number With TOOL CALL block Input: 0: No output 1: Output of the tool number only when tool number changes 2: Output of the tool number with every TOOL CALL block 3: Output of the tool and pocket number only when tool number changes 4: Output of the tool and pocket number output with every TOOL CALL block 5: Output of the tool and pocket number only when tool number changes. Pocket table is not changed. 6: Output of the tool and pocket number output with every TOOL CALL block. Pocket table is not changed.	PLC RUN	6 – 423
MP7480.1	With TOOL DEF block Input: 0: No output 1: Output of the tool number only when tool number changes 2: Output of the tool number with every TOOL DEF block 3: Output of the tool and pocket number only when tool number changes 4: Output of the tool and pocket number output with every TOOL DEF block		
MP7490	Functions for traverse ranges Format: %xxxx Input: Bit 0 – 0: Display one traverse range via MOD 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) 1: One datum for all traverse ranges Bit 2 – Calibration data: touch probe for workpiece measurement: 0: One set of calibrating 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 calibrating data for all traverse ranges 1: Every traverse range has its own set of calibration data	PLC RUN	6 – 25, 6 – 350, 6 – 370

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 TNC calculates the position of the tilting axes of the head/table) Bit 2 – 0: With Cycle 19 the tilting axes are not positioned 1: With Cycle 19 the tilting axes are positioned Bit 3 – 0: The active tilting-axis position, with respect to the machine datum, is included. 1: The first axis assumes a 0° position Bit 4 – 0: The mechanical offset when changing the spindle head when calling M128, M114 or "tilted working plane" is compensated for 1: Mechanical offset during PLC datum shift is compensated for	PLC	6 – 81

MP	Function and input	Software version and behavior	Page
MP7500	Tilting working plane Format: %xxxxxxxxx Bit 5 – 0: The active tilting-axis position, with respect to the machine datum, is included. 1: The tilting-axis position, which is entered with the 3D ROT soft key, is used. Bit 6 – 0: Spatial angle C is realized through a rotation of the coordinate system 1: Spatial angle C is realized through a rotation of the table Bit 7 – 0: The active tilting-axis position, with respect to the machine datum, is included. 1: The active tilting-axis position, in case a) Manual tilting is active, is derived from the tilting angles in the 3D ROT window; b) Manual tilting is inactive, is derived from the reference coordinates. Bit 8 – 0: The active tilting-axis position, depending on bits 3, 5 and 7, is included. 1: In case manual tilting is active, the datum to be set for the principle axes X, Y and Z is recalculated to the home position of the tilting element.	PLC	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 operating modes 0: M144/M145 active 1: M144 is activated automatically at the start of an NC program. Deactivation only with M145 during an NC program. Bit 2 – M144/M145 in the manual operating modes 0: M144/M145 not active 1: M144/M145 active	280 476-09 PLC RUN	6 – 85

MP	Function and input	Software version and behavior	Page
MP7510	Transformed axis	PLC	6 – 82
	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	RUN	
MP7510.0-14	Transformation 1 to transformation 15		
MP7520	Additional code for transformation	PLC	6 – 82
	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, relative to the machine datum for the tilting table	RUN	
MP7520.0-14	Transformation 1 to transformation 15		
MP7530	Type of dimension for transformation	PLC	6 – 82
	Input: -99 999.9999 to +99 999.9999 0: Free tilting axis	RUN	
MP7530.0-14	Transformation 1 to transformation 15		
MP7550	Home position of the tilting element	280 474-01	6 – 82
	Input: -99 999.9999 to +99 999.9999	PLC	
MP7550.0	A Axis	RUN	
MP7550.1	B Axis		
MP7550.2	C Axis		

4.3.17 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: 5)	280 474-07 RESET	6 – 121
MP7600.1	PLC cycle time = position controller cycle time · MP7600.1 Input: 1 to 20 Proposed input value: 7 (= 21 ms)		6 – 121, 7 – 3
MP7620	Feed-rate override and spindle speed override Format: %xxxxxxx Input: Bit 0 – Feed rate override if rapid traverse key in Program Run mode is pressed 0: Override not effective 1: Override effective Bit 1 – Non-functional Bit 2 – Feed rate override if rapid traverse key and machine direction button in Manual operating mode are pressed 0: Override not effective 1: Override effective Bit 3 – Feed rate override and spindle speed override in 1% steps or along a nonlinear curve 0: 1% steps 1: Nonlinear characteristic curve Bit 4 – Non-functional Bit 5 – Reserved Bit 6 – Feed-rate smoothing 0: Not active 1: Active	PLC RUN	6 – 119, 6 – 191, 6 – 237
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	6 – 381
MP7641	Entry of the interpolation factor Input: 0: Through TNC keyboard 1: Through PLC Module 9036	PLC RUN	6 – 381

MP	Function and input	Software version and behavior	Page
MP7645	Initializing parameter for handwheel	PLC	6 – 385
MP7645.0	Layout of the handwheel keypad for HR 410 Input: 0: Evaluation of the keys by the NC, including LEDs 1: Evaluation of the keys by PLC	RUN	
MP7645.0	Assignment of a third handwheel via axis selector switch S2, when MP7645.2 = 0 Input: 0: Switch position 1 (at the left stop) 3rd handwheel axis Z Switch position 2 3rd handwheel axis IV Switch position 3 3rd handwheel axis V 1: Switch position 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		
MP7645.1	Fixed assignment of third handwheel if MP7645.2 = 1 Input: 4: Axis Z 8: Axis IV (MP410.3) 16: Axis V (MP410.4)		
MP7645.2	Assignment of a third handwheel via axis selector switch or MP7645.1 Input: 0: Assignment by axis selection switch according to MP7645.0 1: Assignment by MP7645.1		
MP7645.3-7	No function		

MP	Function and input	Software version and behavior	Page
MP7650	Counting direction for handwheel Format: %xxxxxxxxx Input: 0: Negative counting direction 1: Positive counting direction	from 280 474-07 bit-coded PLC RUN	6 – 381
MP7660	Threshold sensitivity for electronic handwheel Input: 0 to 65 535 [increments]	PLC RUN	6 – 381
MP7670	Interpolation factor for handwheel Input: 0 to 10	PLC RUN	6 – 381, 6 – 384
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]	PLC RUN	6 – 384
MP7671.0	Low speed		
MP7671.1	Medium speed (only HR 410)		
MP7671.2	High speed (only HR 410)		

MP	Function and input	Software version and behavior	Page
MP7680	Machine parameter with multiple function Format: %xxxxxxxxxxxxx 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 – 153, 6 – 155, 6 – 293, 6 – 295, 6 – 344, 6 – 416

MP	Function and input	Software version and behavior	Page
MP7680	Machine parameter with multiple function Format: %xxxxxxxxxxxxx Bit 8 – Inserting a rounding arc or cubic spline 0: Rounding arc is inserted 1: Instead of the rounding arc a cubic spline is inserted 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 – Reserved Bit 12 – Behavior of Cycle 28 0: Standard behavior 1: The slot wall is tangentially approached and departed; at the beginning and end of the slot a rounding arc with a diameter equal to the slot is inserted.	PLC RUN	6 – 153, 6 – 155, 6 – 293, 6 – 295, 6 – 344, 6 – 416
MP7681	M/S/T/Q transfer to the PLC during block scan Format: %xxxx Input: Bit 0 – 0: During block scan, transfer M functions to the PLC 1: Collect M functions, and after the block scan, to the PLC 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: During block scan, transfer S or G code to the PLC 1: After block scan, transfer S or G code to the PLC Bit 3 – 0: During block scan, transfer FN19 outputs to the PLC 1: After block scan, transfer last FN19 outputs to the PLC	PLC	6 – 297

MP	Function and input	Software version and behavior	Page
MP7682	Machine parameter with multiple function Format: %xxx 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 – 119, 6 – 231, 6 – 232,
MP7683	Pallet tables, executing Format: %xxxxx Input: Bit 0 – Operating mode PROGRAM RUN, SINGLE BLOCK 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 – Operating mode PROGRAM RUN, FULL SEQUENCE 0: During the start, a complete NC program is run. 1: At the start all NC programs are executed up to the 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 pallet table is reached, the process begins again with the first line 0: Function is not in effect 1: Function is in effect (bit 2 = 1) Bit 4 – Editing the active pallet table 0: Active pallet table cannot be edited 1: In the operating modes PROGRAM RUN, FULL SEQUENCE and PROGRAM RUN, SINGLE BLOCK, the active pallet table can be edited	PLC RUN	6 – 247, 6 – 306

MP	Function and input	Software version and behavior	Page
	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 by PLC 0: AUTOSTART function performed by the NC 1: AUTOSTART function of an NC program is performed by the PLC. The NC does not trigger an NC start.		
MP7690	MEMORY TEST during switch-on Format: %xxx Input: 1: No MEMORY TEST during switch-on 0: MEMORY TEST during switch-on Bit 0 – Test the RAM Bit 1 – Test the EPROM Bit 2 – Test the hard disk		6 – 326

4.3.18 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.	280 474-03	6 – 216

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		7 – 177
9002	Reading all inputs of a PLC input/output unit		6 – 390
9003	Reading in analog inputs		6 – 393
9004	Edges of PLC inputs		6 – 392
9005	Update all outputs of a PLC input/output unit		6 – 391
9007	Diagnostic information of the PL	280 476-05	6 – 389
9008	Reading specific inputs of a PLC input/ output unit	280 472-07	6 – 391
9009	Update certain outputs of a PLC input/ output unit	280 472-07	6 – 392
9010/ 9011/ 9012	Read in the word range		7 – 178
9019	Size of the processing stack		7 – 47
9020/ 9021/ 9022	Write in the word range		7 – 179
9031	Overwrite machine parameters		4 – 8
9032	Read machine parameters		4 – 9, 7 – 37
9033	Select machine parameter file		4 – 11
9034	Load a machine parameter subfile		4 – 12
9035	Reading status information		6 – 22, 6 – 43, 6 – 242
9036	Writing status information		6 – 382, 6 – 396
9038	Reading general axis information		6 – 20
9040	Reading of axis coordinates (Format 0.001 mm)		6 – 233
9041	Reading of axis coordinates (Format 0.0001 mm)		6 – 233
9042	Reading the spindle coordinates (Format 0.001°)		6 – 182

Module	Function	SW Vers.	Page
9044	Reading the spindle coordinates (Format 0.0001°)	280 476-03	6 – 182
9050	Conversion of binary numbers → ASCII		7 – 180
9051	Conversion of binary numbers → ASCII		7 – 181
9052	Conversion of ASCII → Binary		7 – 182
9053	Conversion from binary → ASCII/ hexadecimal		7 – 182
9054	Conversion from ASCII/hexadecimal → binary		7 – 183
9055	Local time		6 – 332
9060	Status of M functions	280 476-06	6 – 284
9061	Status of non-modal M functions	280 476-09	6 – 284
9066	Status of HEIDENHAIN inverter	280 476-09	6 – 172
9070	Copy a number from a string		7 – 157
9071	Find the string length		7 – 158
9080	Clearing the small PLC window		6 – 251
9081	Interrogating the status of the small PLC window		6 – 251
9082	Showing a string in the small PLC window		6 – 252
9083	Showing a moving-bar diagram in the small PLC window		6 – 254
9085	Display PLC error messages		6 – 289
9086	Erase PLC error messages		6 – 290
9087	Status of PLC error message		6 – 290
9088	Displaying the M functions	280 474-03	6 – 239
9089	Control in operation		6 – 240
9090	Selection of a line in the pallet table		6 – 306, 6 – 309
9091	Finding the line number of a tool in the tool table	280 474-03	6 – 417
9092	Searching for an entry in the tables selected for execution (.T/.D/.TCH)		6 – 405
9093	Read data from tables selected for program (.T/.D/.TCH)		6 – 407
9094	Writing data into a tool and datum table		6 – 408
9095	Select active line in configuration file		6 – 43
9096	Erasing a line in the tool table		6 – 409
9097	Selecting the geometry description	280 476-03	6 – 66
9098	Finding the active geometry description	280 476-05	6 – 67

Module	Function	SW Vers.	Page
9100	Assign data interface		8 – 31
9101	Release data interface		8 – 32
9102	Status of data interface		8 – 32
9103	Transmit string through data interface		8 – 33
9104	Receive string through data interface		8 – 34
9105	Transmit binary data through data interface		8 – 35
9106	Receive binary data through data interface		8 – 36
9107	Read from receiving buffer		8 – 37
9110	Transmit a message via LSV2		8 – 38
9111	Receive a message via LSV2		8 – 39
9112	Transmit ASCII characters via data interface		8 – 40
9113	Receive ASCII characters via data interface		8 – 41
9120	Starting a PLC axis		6 – 29
9121	Stopping a PLC axis		6 – 29
9122	Status of PLC axis		6 – 30
9123	Traversing the reference marks of PLC axes		6 – 31
9124	Feed rate override for PLC axis		6 – 31
9125	Stop PLC axis at next Hirth grid position		6 – 32
9130	Output of an analog voltage		6 – 394
9133	Internal temperature of the LE	280 476-09	6 – 165
9135	Switch on 3-D touch probe	280 472-05	6 – 346
9145	Actual-to-nominal value transfer	280 472-07	6 – 132
9146	Saving and reestablishing actual position values	280 476-13	6 – 219
9147	Assigning the reference value to an axis	280 474-04	6 – 100
9151	Selecting traverse range and axis designation		6 – 23
9152	Selecting traverse range, axis display and axis designation	280 476-01	6 – 24
9153	Switching the touch probe axis	280 474-07	6 – 350
9155	Axis switchover from closed loop to open loop	280 474-04	6 – 219
9156	Axis switchover from open loop to closed loop	280 474-04	6 – 220
9157	Drive controller status	280 476-13	6 – 144
9158	Maximum torque	280 476-01	6 – 171
9159	Drives that are switched off in 200 ms	280 474-10	6 – 144

Module	Function	SW Vers.	Page
9160	Status request for temperature monitoring and I^2t monitoring		6 – 168
9161	Enabling the drive controller		6 – 145
9162	Status request of the drive controller		6 – 145
9163	Wye/delta connection switchover		6 – 214
9164	Reading the actual speed value of the motor		6 – 135
9165	Sampling the current motor temperature		6 – 165
9166	Momentary utilization of the drive motor		6 – 169
9167	Supply voltage monitoring		6 – 164
9168	Interrogating the commissioning status		6 – 472
9169	Axes for which I32 does not switch off the drives	280 474-10	6 – 145
9171	Oriented spindle stop		6 – 203
9175	Spindle switchover	280 474-03	6 – 216
9180	Simulation of NC keys		6 – 339
9181	Disabling individual NC keys		6 – 339
9182	Re-enabling individual NC keys		6 – 340
9183	Disabling groups of NC keys		6 – 340
9184	Re-enabling groups of NC keys		6 – 341
9186	Call a soft-key function		6 – 341
9187	Status of a soft-key function call		6 – 342
9189	Shutting down the control	280 476-03	6 – 249
9190	Starting the operating times		6 – 329
9191	Stopping the operating times		6 – 329
9192	Reading the operating times		6 – 330
9193	Setting the operating times		6 – 330
9194	Alarm when operating time exceeded		6 – 331
9195	System time		6 – 331
9196	Finding the PLC cycle time	280 476-05	7 – 3
9200	Display/delete PLC soft-key row		6 – 274
9201	Display/delete PLC soft key		6 – 275
9202	Select/deselect PLC soft keys and PLC windows		6 – 275
9210	Opening or erasing screen mask for the PLC window		6 – 266
9211	Status of the large PLC window		6 – 267
9215	Activating a PLC pop-up window		6 – 279
9220	Renewed traversing of the reference marks		6 – 103
9221	Starting a PLC positioning movement		6 – 34

Module	Function	SW Vers.	Page
9222	Status request of PLC positioning movement		6 – 34
9223	Free rotation		6 – 234
9225	Compensation value for the reference mark	280 474-04	6 – 99
9230	Datum shift		6 – 399
9231	Compensation of thermal expansion		6 – 45
9240	Opening a file		6 – 319
9241	Closing a file		6 – 320
9242	Positioning in a file		6 – 320
9243	Reading from a file line by line		6 – 321
9244	Writing to a file line by line		6 – 322
9245	Reading a field out of a table		6 – 311
9246	Writing to a field in a table		6 – 313
9247	Searching for a condition in a table		6 – 315
9250	Starting the PLC editor for tables		6 – 317
9251	Ending the PLC editor for tables		6 – 318
9252	Positioning the cursor in the PLC editor		6 – 318
9255	Reading a field out of a table		6 – 312
9256	Writing to a field in a table		6 – 314
9260	Receiving and waiting for events		7 – 166
9261	Sending events		7 – 167
9262	Context change between spawn processes		7 – 168
9263	Interrupting a spawn process for a defined time		7 – 168
9270	Reading a code word		7 – 43
9271	Writing a code word		7 – 43
9275	Writing ASCII data into the log	280 476-01	6 – 336
9276	Writing operand contents into the log	280 476-01	6 – 337
9279	Control reset	280 476-09	6 – 249
9280	Start the NC macro (Run pallet entry)	280 474-07	6 – 308
9281	Selection of a line in the pallet table	280 476-01	6 – 307
9290	Selecting a file	280 474-09	6 – 301
9291	Calling an NC macro	280 476-03	7 – 45
9300	Locking/releasing the pocket table	280 476-06	6 – 410

Module	Function	SW Vers.	Page
9301	Find the number of an entry in the pocket table	280 476-07	6 – 412
9302	Search for a free pocket in the tool magazine	280 476-07	6 – 412
9305	Tool exchange in the pocket table	280 476-06	6 – 411
9306	Exchange tools between tool magazines	280 476-07	6 – 413
9310	Read the machine parameter from the run-time memory	280 476-06	4 – 10
9320	Status of the NC program end	280 476-06	6 – 246

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		6 – 283
M	4000	Spindle in position	NC	NC		6 – 203
M	4001	Nominal speed command signal of the spindle not in the ramp	NC	NC		6 – 186
M	4002	Nominal speed value = 0	NC	NC		6 – 186
M	4003	Nominal speed value output analog or digital (MP3010 = 3 to 8)	NC	NC		6 – 184
M	4004	Impermissible speed was programmed	NC	NC		6 – 187
M	4005	Status display and nominal speed value output for M03	PLC	PLC		6 – 186, 6 – 238
M	4006	Status display and nominal speed value output for M04	PLC	PLC		6 – 186, 6 – 238
M	4007	Status display M05 and spindle stop	PLC	PLC		6 – 186, 6 – 238
M	4008	Disable speed output for spindle	PLC	PLC		6 – 187, 6 – 238
M	4009	Counterclockwise spindle rotation (for gear change)	PLC	PLC		6 – 189
M	4010	Clockwise spindle rotation (for gear change)	PLC	PLC		6 – 189
M	4011	Activate rotational speed MP3520.0 and direction of rotation from M4013	PLC	PLC		6 – 205
M	4012	Opening the spindle control loop	PLC	PLC		6 – 203, 6 – 352
M	4013	Direction for spindle orientation from a standstill (M03 = 0; M04 = 1)	PLC	PLC		6 – 205
M	4014	Reverse the direction of spindle rotation	PLC	PLC		6 – 186
M	4015	Renewed evaluation of the spindle reference mark	PLC	NC		6 – 203
M	4016	Cycle 13 is executed	NC	PLC		6 – 205
M	4017	Spindle moving in feedback control	NC	NC		6 – 203
M	4018	Reference mark for spindle not yet traversed	NC	NC		6 – 203
M	4019	Reversing the counting direction of the position encoder on the spindle	PLC	PLC		6 – 186
M	4030	Cycle 2 or Cycle 17 active	NC	NC		6 – 207, 6 – 211
M	4031	Cycle 17 or Cycle 18 active	NC	NC		6 – 211

	Marker	Description	Set	Reset	SW Vers.	Page
M	4040	Status display M07, M08, and M09 highlighted	PLC	PLC		6 – 238
M	4041	Status display M07, M08, M09, MK	PLC	PLC		6 – 238
M	4042	Status display M07, M08, M09, MK	PLC	PLC		6 – 238
M	4050	Touch probe not ready, ready signal is missing	NC	NC		6 – 349
M	4051	Stylus deflected before start of probing cycle	NC	NC		6 – 349
M	4052	Stylus is deflected, probing process is completed	NC	PLC		6 – 349
M	4053	Probing process has been ended or canceled	NC	NC		6 – 349
M	4054	Battery voltage too low (battery warning at touch probe connection); evaluated only during the probing process	NC	NC		6 – 349
M	4055	Enable the probing process	NC	PLC		6 – 349
M	4056	NC stop in all operating modes if stylus is deflected	PLC	PLC		6 – 349
M	4060	Cycle for tool measurement started	NC	NC		6 – 378
M	4061	0: Measure the tool 1: Check the tool	NC	NC		6 – 378
M	4062	0: Wear tolerance not exceeded 1: Wear tolerance exceeded	NC	NC/ PLC		6 – 378
M	4063	0: Breakage tolerance not exceeded 1: Breakage tolerance exceeded	NC	NC/ PLC		6 – 378
M	4065	Workpiece dimensions are OK	NC	PLC	280 474-03	6 – 353
M	4066	Workpiece must be reworked	NC	PLC	280 474-03	6 – 353
M	4067	Workpiece is scrap	NC	PLC	280 474-03	6 – 353
M	4070	Strobe signal for gear code	NC	NC		6 – 189
M	4071	Strobe signal for S code	NC	NC		6 – 194
M	4072	Strobe signal for M functions	NC	NC		6 – 283
M	4073	Strobe signal T code (P code) with TOOL CALL	NC	NC		6 – 424, 6 – 440
M	4074	Strobe signal T code (P code) with TOOL DEF	NC	NC		6 – 424, 6 – 440
M	4075	Transfer active with FN19	NC	NC		7 – 22
M	4090	Acknowledgment of ”gear change completed”	PLC	PLC		6 – 189
M	4091	Acknowledgment of S code	PLC	PLC		6 – 194
M	4092	Acknowledgment of M functions	PLC	PLC		6 – 283
M	4093	Acknowledgment of T code (P code) with TOOL CALL	PLC	PLC		6 – 424, 6 – 440
M	4094	Acknowledgment of T code (P code) with TOOL DEF	PLC	PLC		6 – 424, 6 – 440
M	4095	Acknowledgement of transfer with FN19	PLC	PLC		7 – 22

	Marker	Description	Set	Reset	SW Vers.	Page
M	4120 - 4128	PLC positioning axis 1 to 9 active	NC/ PLC	NC/ PLC		6 – 35
M	4130	Activation of spindle orientation, or spindle orientation has been started with Module 9171	NC/ PLC	NC		6 – 205
M	4131	Activation of Q-parameter transfer to the NC; data from D258, Q number from W516	PLC	NC		7 – 22
M	4132	Activate datum shift from D528 to D544, or call Module 9230	PLC	NC		6 – 399
M	4133	Starting and stopping the free rotation function	PLC	NC		6 – 235
M	4134	Activation of a gear range and speed through the PLC	PLC	NC		6 – 189
M	4135	Strobe marker for selecting the traverse range	PLC	NC		6 – 23
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		6 – 295
M	4157	Returning to the contour (APPROACH POSITION) is active	NC	NC		6 – 295
M	4158	Block scan active	NC	NC		6 – 295
M	4159	PLC editor: END key or soft key pressed	NC	NC/ PLC		6 – 317
M	4160	Pallet table selected	NC	NC		6 – 306
M	4161	M/S/T/Q transfer after block scan	NC	NC		6 – 297
M	4170	END PGM, M02 or M30 was executed	NC	NC		6 – 300
M	4172	1. PLC scan after power on	NC	NC		–
M	4173	1. PLC scan after interruption of the PLC program	NC	NC		–
M	4174	1. 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 blinks	NC	NC		6 – 240
M	4176	Control is in operation, control-in-operation symbol is on or is blinking	NC	NC		6 – 240
M	4177	Erasable error message is displayed	NC	NC		6 – 173
M	4178	Error message EMERGENCY STOP is displayed	NC	NC		6 – 173
M	4179	Control is shut down	NC	NC	280 474-15	6 – 249

	Marker	Description	Set	Reset	SW Vers.	Page
M	4180	Rapid traverse programmed (FMAX)	NC	NC		6 – 237
M	4181	NC program selected	NC	PLC	280 476-01	6 – 245
M	4182	AUTOSTART active	NC	NC	280 476-17	6 – 247
M	4183	Time from AUTOSTART expired	NC	NC	280 476-17	6 – 247
M	4200	Overflow during multiplication	NC	PLC		7 – 116, 7 – 130, 7 – 159
M	4201	Division by 0	NC	PLC		7 – 117, 7 – 130, 7 – 159
M	4202	Incorrectly executed modulo	NC	PLC		7 – 118, 7 – 130, 7 – 159
M	4203	Error status for PLC module	NC	NC/ PLC		7 – 130, 7 – 159
M	4204	Reserved for errors that the PLC programmer would like to catch	NC	NC		7 – 159
M	4220	Error from PET table with F stop active	NC	NC	280 474-15	6 – 278
M	4221	Error from PET table with NC stop active	NC	NC	280 474-15	6 – 278
M	4222	Error from PET table with EM. STOP active	NC	NC	280 474-15	6 – 278
M	4230	NC start via LSV2	NC	NC	280 476-05	6 – 344
M	4231	NC stop via LSV2	NC	NC	280 476-05	6 – 344
M	4300 to 4315	Value from MP4310.0	NC	NC		7 – 36
M	4316 to 4331	Value from MP4310.1	NC	NC		7 – 36
M	4332 to 4347	Value from MP4310.2	NC	NC		7 – 36
M	4348 to 4363	Value from MP4310.3	NC	NC		7 – 36
M	4364 to 4379	Value from MP4310.4	NC	NC		7 – 36
M	4380 to 4395	Value from MP4310.5	NC	NC		7 – 36
M	4396 to M4411	Value from MP4310.6	NC	NC		7 – 36
M	4520	Additional T code (P code) follows with TOOL CALL	NC	NC		6 – 425, 6 – 440
M	4521	Tool number zero programmed	NC	NC		6 – 424
M	4522	Tool with pocket number programmed is in effect with MP7480.0 = 3 or 4 and TOOL CALL	NC	NC		6 – 425
M	4523	Tool without pocket number programmed is in effect with MP7480.0 = 3 or 4 and TOOL CALL	NC	NC		6 – 425

	Marker	Description	Set	Reset	SW Vers.	Page
M	4524	Special tool called, TOOL CALL	NC	NC		6 – 425, 6 – 440
M	4525	TOOL CALL after expiration of tool life	NC	NC		6 – 425
M	4526 - 4534	Axis 1 to Axis 9 is the tool axis	NC	NC		6 – 21
M	4538	Geometry of the tool from W264	PLC	NC		6 – 295, 6 – 424
M	4540	Sequence of tool number or pocket number transfer (M4520 = 1)	PLC	PLC		6 – 425, 6 – 440
M	4541	Special tool in original pocket in spite of variable pocket coding	PLC	PLC		6 – 425, 6 – 414, 6 – 440
M	4542	Do not update pocket number in the pocket table	PLC	PLC		6 – 295, 6 – 425
M	4543	Tool life 1 expired (TIME1 in the tool table)	NC	NC/ PLC		6 – 416
M	4546	Tool life 2 expired (TIME2 in the tool table)	NC	NC/ PLC	280 476-03	6 – 416
M	4547	T and G strobes with TOOL CALL	NC	NC	280 476-06	6 – 424, 6 – 189
M	4560	NC stop (0: Stop)	PLC	PLC		6 – 344
M	4561	Rapid traverse	PLC	PLC		6 – 344
M	4562	Memory function for axis direction keys (MP7680 Bit 0 = 1)	PLC	PLC		6 – 344
M	4563	Feed-rate enable for all axes	PLC	PLC		6 – 131
M	4564	NC start	PLC	PLC		6 – 344
M	4570	Unit of measure for transfer with FN19	NC	NC		7 – 22
M	4571	Activation of decoded M-code transfer in M1900 to M1999	PLC	PLC		6 – 283
M	4572	Enabling the incremental jog positioning	PLC	PLC		6 – 395
M	4574	Select the traverse range (with M4575)	PLC	PLC		6 – 23, 6 – 349
M	4575	Select the traverse range (with M4574)	PLC	PLC		6 – 23, 6 – 349
M	4576	Locking the handwheel	PLC	PLC		6 – 381
M	4577	Disabled key was pressed	NC	PLC		6 – 338
M	4579	INCREMENT OFF/ON soft key	NC	NC		6 – 395
M	4580	Suppress EMERGENCY STOP, open all position control loops, NC stop	PLC	PLC		6 – 130, 6 – 173
M	4581	Open all position control loops, NC stop, activate “Approach position”	PLC	PLC		6 – 130
M	4586	Enable AUTOSTART	PLC	NC/ PLC	280 476-03	6 – 247
M	4587	Feed rate limit exceeded F MAX	PLC	PLC	280 476-09	6 – 128

	Marker	Description	Set	Reset	SW Vers.	Page
M	4590	Status fast PLC input from MP4130.2	NC	PLC		7 – 21
M	4591	Status fast PLC input from MP4130.3	NC	PLC		7 – 21
M	4592	Status fast PLC input from MP4130.4	NC	PLC		7 – 21
M	4593	Status fast PLC input from MP4130.5	NC	PLC		7 – 21

	Marker	Description	Set	Reset	SW Vers.	Page
W	256	Gear code	NC/ PLC	NC/ PLC		6 – 189
W	258	S code	NC	NC		6 – 194
W	260	Code for M functions	NC	NC		6 – 283
W	262	Tool pocket number	NC	NC		6 – 424, 6 – 440
W	264	Tool number	NC	NC		6 – 424, 6 – 440
W	266	Index number of a programmed indexed tool	NC	NC	280 474-03	6 – 417
W	268	Tool magazine number	NC	NC		6 – 411
W	270	Line number in help file	NC	NC		6 – 277
W	272	Operating mode	NC	NC		6 – 102
W	274	Code of the depressed key	NC	NC		6 – 338
D	276	Code of the code number last entered via MOD	NC	NC		6 – 324
D	280	1. integer value from FN19	NC	NC		6 – 22
D	284	2. integer value from FN19	NC	NC		7 – 22
W	302	Number of the PLC soft key that was pressed	NC	NC		6 – 273
W	320	Nominal speed value [rpm]	NC	NC		6 – 184
W	322	Actual speed value [rpm]	NC	NC		6 – 184
D	356	Programmed speed [0.001 rpm]	NC	NC		6 – 184, 6 – 189
D	360	Programmed feed rate	NC	NC		6 – 128
D	364	Nominal speed value [rpm]	NC	NC	280 472-05	6 – 184
D	368	Actual speed value [rpm]	NC	NC	280 472-05	6 – 184
D	388	Current tool feed rate [mm/min]	NC	NC		6 – 128
W	480-484	Analog input at X48 [0.1 V]	NC	NC		6 – 393
W	486 - 490	Temperature input at X48 [0.5 °C]	NC	NC		6 – 44, 6 – 393
W	492	Percentage for spindle override (NC to PLC)	NC	NC		6 – 191
W	494	Percentage for feed rate override (NC to PLC)	NC	NC		6 – 237
W	516	Q No. 0-7 for numerical data transfer PLC to NC	PLC	PLC		7 – 22
B	518	Defining the free rotation function	PLC	PLC		6 – 235
B	519	Traverse direction for free rotation	PLC	PLC		6 – 235
W	522	Enabling the high-speed PLC inputs	PLC	PLC		6 – 157
D	528	Double word with multiple function, here data for transfer from PLC to NC	PLC	PLC		6 – 399

	Marker	Description	Set	Reset	SW Vers.	Page
D	528-544	Target position for PLC positioning	PLC	PLC		6 – 35
D	528	Datum shift for axis 1	PLC	PLC		6 – 399
D	532	Datum shift for axis 2	PLC	PLC		6 – 399
D	536	Datum shift for axis 3	PLC	PLC		6 – 399
D	540	Datum shift for axis 4	PLC	PLC		6 – 399
D	544	Datum shift for axis 5	PLC	PLC		6 – 399
W	560 - 568	Feed rate for PLC positioning	PLC	PLC		6 – 35, 6 – 235
W	576 - 584	Lag-tracking axis error compensation	PLC	PLC		6 – 44
D	592	Nominal position for spindle orientation	PLC	PLC		6 – 205
D	596	Max. feed rate from PLC [mm/min]	NC/ PLC	PLC		6 – 128
D	604	Maximum possible spindle speed	PLC	NC/ PLC	280 474-03	6 – 184
W	754	% function for feed-rate override for free rotation	PLC	PLC		6 – 235
D	756	Programmed rotational speed or rotational speed of the PLC [0.001 rpm]	NC/ PLC	NC/ PLC		6 – 189
D	760	Offset in tilting axes touch probe center offset [1/10 000°]	PLC	PLC		6 – 352
W	764	Percentage for spindle override (PLC to NC)	NC/ PLC	NC/ PLC		6 – 191
W	766	Percentage for feed rate override (PLC to NC)	NC/ PLC	NC/ PLC		6 – 237
D	768 - 956	Values from MP4210.0 to MP4210.47	NC	NC		7 – 35
W	960 - 968	Value from MP4220.0 to MP4310.4	NC	NC		7 – 36
W	976 - 988	Value from MP4310.3 to MP4310.6	NC	NC		7 – 36
W	1008	S code for minimum speed	NC	NC		6 – 194
W	1018	Number of files opened by the PLC	NC	NC	280 476-03	6 – 301
W	1020	Number of open files	NC	NC	280 476-03	6 – 301
W	1022	Error status of the module last called	NC	NC		–
W	1024	Axis release	NC	NC		6 – 130
W	1026	Axes in position	NC	NC		6 – 162
W	1028	Axes in motion	NC	NC		6 – 163
W	1030	Current direction of traverse	NC	NC		6 – 10
W	1032	Reference marks not yet traversed	NC	NC		6 – 102
W	1034	Positive software limit switch was traversed	NC	NC		6 – 25
W	1036	Negative software limit switch was traversed	NC	NC		6 – 25
W	1038	Preparing opening of the position control loop	PLC	PLC		6 – 131
W	1040	Axis-specific opening of the position control loop	PLC	PLC		6 – 131
W	1042	Deactivation of monitoring functions	PLC	PLC		6 – 156

	Marker	Description	Set	Reset	SW Vers.	Page
W	1044	Actual-to-nominal value transfer	PLC	PLC		6 – 132
W	1046	Manual traverse in positive direction	PLC	PLC		6 – 344
W	1048	Manual traverse in negative direction	PLC	PLC		6 – 344
W	1050	Incremental jog positioning in positive direction	PLC	PLC		6 – 395
W	1052	Incremental jog positioning in negative direction	PLC	PLC		6 – 395
W	1054	Reference end position	PLC	PLC		6 – 102
W	1056	Lubrication pulse: Value in MP4060 exceeded	NC	NC		6 – 26
W	1058	Resetting the accumulated distance	PLC	PLC		6 – 26
W	1060	Axis-specific feed-rate enable	PLC	PLC		6 – 131
W	1062	Lock the handwheel for specific axes	PLC	PLC	280 474-07	6 – 381

6 Machine Integration

6 Machine Integration

6.1 Machine Axes

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.

Note

Axis 9 may only be used as a PLC axis.

MP10	**Active axes**
Format:	%xxxxxxxxx
Input:	Bits 0 to 8 correspond to axes 1 to 9 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:	XYZABCUVWxyzabcuvw-
Input:	Characters 1 to 9 correspond to 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 the 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.

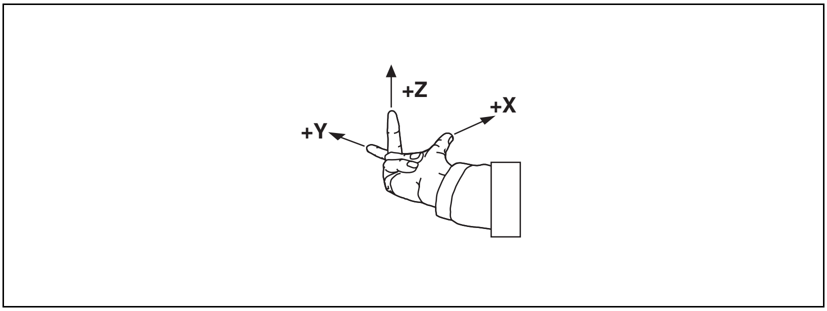
MP410	**Assignment of axis keys IV and V**
Input:	Axis designation XYZABCUVWxyzabcuvw-
MP410.3	Axis key IV
MP410.4	Axis key V

6.1.2 Axis Designation

Principal axes X, Y, Z

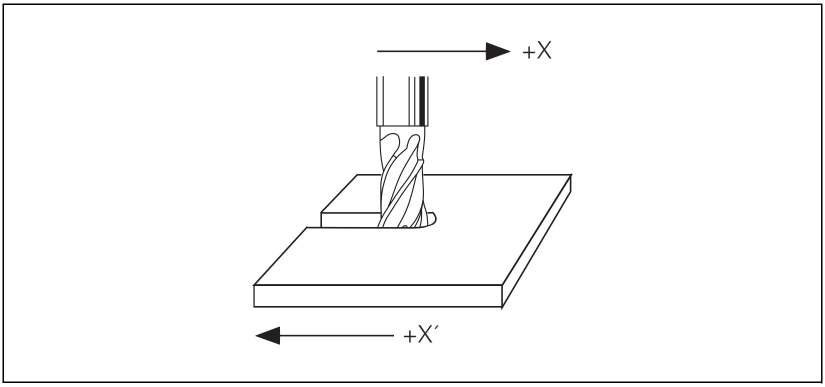
The coordinate axes and their directions of motion are defined in the international standard ISO 841.

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



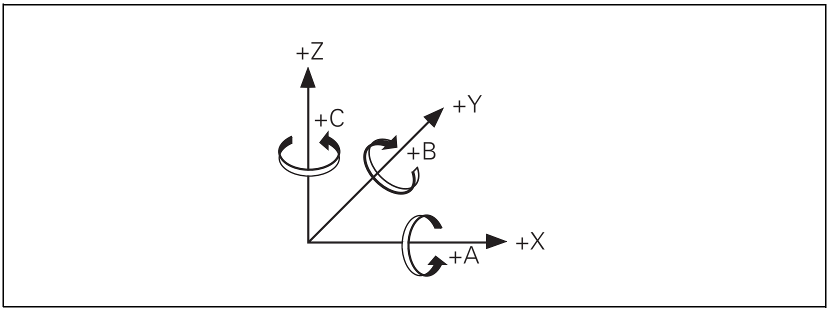
Algebraic signs of the axes

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´:



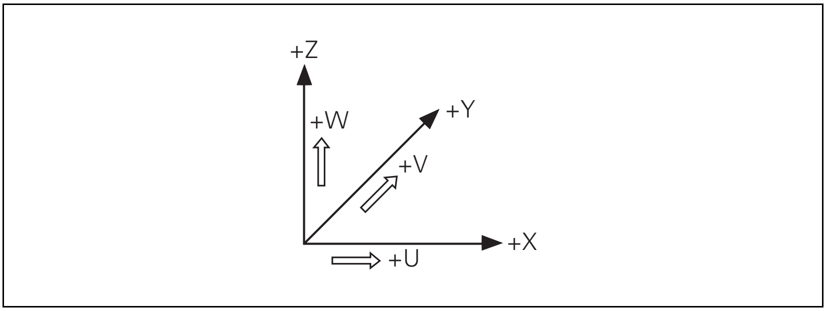
Rotary axes A, B, C

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. HEIDENHAIN contouring controls work with incremental position encoders. The TNC 426 M and TNC 430 M contouring controls are also compatible with encoders with an EnDat interface.

Signal period

For any given distance the position encoder outputs a fixed number of signal periods. The signal is subdivided by 1024.

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 these data the TNC 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 shaft speed encoder must be entered in MP331.x and MP332.x. This also applies to speed encoders with an EnDat interface, since the incremental track of the speed encoder is used for position feedback control.

HEIDENHAIN offers linear encoders with **distance-coded reference marks.** The nominal increment between two fixed reference marks depends on the encoder being used:

- In MP334.x, enter for each axis the nominal increments between two fixed reference marks.

If the number of grating periods between the reference end position and the first reference mark exceeds the value from MP334.x, the error message **`Ref mark <axis>: incorrect spacing`** appears. This monitoring is turned off with MP334.x = 0.

Example:
LS 486C:
Incremental linear encoder with distance-coded reference marks, grating period 20 µm (= one signal period covers 0.02 mm), nominal increment between reference marks is 20 mm.

MP331.x = 0.02
MP332.x = 1

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

MP331.0-8 Distance for the number of signal periods in MP332
Input: 0.0001 to 99 999.9999 [mm] or [°]

MP332.0-8 Number of signal periods for the distance in MP331
Input: 1 to 16 777 215

MP334.0-8 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.

MP340.0-8 Interpolation factor for external interpolation
As of software version:280 476-01
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 with the LE 426 M and LE 430 M. With MP115.0 you select the signal at position encoder inputs 1 to 10, and with MP115.2 you select the maximum input frequency at the position encoder inputs 1 to 10.

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: %xxxxxxxxxx
Input: Bit 0 to bit 5: Position encoder inputs X1 to X6
Bit 6 to bit 9: Position encoder inputs X35 to X38
0: 1 V_{PP}
1: 11 μA_{PP}

MP115.1 Reserved
Format: %xxxxxxxxxx
Input: Enter %0000000000

MP115.2 Input frequency of the position encoder inputs
Format: %xxxxxxxxxx
Input: Bit 0 to bit 5: Position encoder inputs X1 to X6
Bit 6 to bit 9: Position encoder inputs X35 to X38
For 1 V_{PP}: 0: 50 kHz
1: 350 kHz
For 11 μA_{PP}: 0: 50 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: %xxxxxxxxx
Input: Bits 0 to 8 correspond to axes 1 to 9
0: Positive
1: Negative

MP1040 **Analog axes: Polarity of nominal value voltage**
Digital axes: Algebraic sign of the nominal speed value
Format: %xxxxxxxxx
Input: Bits 0 to 8 correspond to axes 1 to 9
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 value.

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: %xxxxxxxxx
Input: Bits 0 to 8 correspond to axes 1 to 9
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. Has no function
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

Position encoders:
In the event of a disturbance, the error message
`EnDat defective <error codes> <axis>` will appear.

Meaning of the error codes up to NC software 280 476-12:

Error code (to 280 476-12)	Meaning
00001	Light source defective
00010	Signal amplitude too small
00100	Incorrect position value

The error codes may also appear combined, in which case they add themselves together.

A disturbance in communication between the EnDat position encoder and the control (e.g. defective cable) results in the error message
`EnDat defective 11111 <axis>.`

Meaning of the error codes as of NC software 280 476-13:

The error code is shown in hexadecimal notation. The 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 of 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”

Speed encoders:

Up to NC software 280 476-17:
Regardless of the **Type of encoder** in the motor table motor.mot, the TNC is attempting to communicate with a speed encoder with EnDat interface. If this does not succeed, a speed encoder with Z1 track will be assumed.
If an error occurs during communication with the EnDat encoder, the control assumes that it is dealing with an encoder with Z1 track. However, it will not find the track, since encoders with EnDat interface do not have a Z1 track. This results in the error message **C310 Z1 track error.**

As of NC software 280 476-18:
The TNC 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 must 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

If you use the motor table motor.sn instead of motor.mot, the control attempts to communicate with an encoder with EnDat interface. The control then assumes that it is dealing with an encoder with a Z1 track, and tries to read the track. If this is not possible, for example because it is actually an encoder with EnDat interface, the error message **C3F0 DSP error in axis <axis>** appears.

6.1.4 Assignment

Axes

With the following machine parameters you assign the position and speed encoder inputs or speed command outputs to the individual axes.

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.

The following are the only possible assignments of connector designations to axes:

LE	Axis	9	8	7	6	5	4	3	2	1
LE 426 PB/M, 12 000 rpm	Designation (MP100.x)	–	–	–	–	?	?	?	?	?
	Position (MP110.x)	–	–	–	–	X1 to X6				
	Speed (MP112.x)	–	–	–	–	X15 to X20				
	Nominal value (MP120.x)	–	–	–	–	X51 to X56[a]				
LE 426 PB/M, 30 000 rpm	Designation (MP100.x)	–	–	–	–	?	?	?	?	?
	Position (MP110.x)	–	–	–	–	X1 to X6				
	Speed (MP112.x)	–	–	–	–	X15 to X19				
	Nominal value (MP120.x)	–	–	–	–	X51 to X55[a]				
LE 430 PA	Designation (MP100.x)	?	?	?	?	?	?	?	?	?
	Position (MP110.x)	X1 to X6 and X35 to X38								
	Speed (MP112.x)	–	–	–	X15 to X20					
	Nominal value (MP120.x)	–	–	–	X51 to X56[a]					
LE 430 M/ 6 axes	Designation (MP100.x)	–	–	–	–	?	?	?	?	?
	Position (MP110.x)	–	–	–	–	X1 to X6				
	Speed (MP112.x)	–	–	–	X15 to X20					
	Nominal value (MP120.x)	–	–	–	X51 to X56[a]					
LE 430 M/ 9 axes	Designation (MP100.x)	?	?	?	?	?	?	?	?	?
	Position (MP110.x)	X1 to X6 and X35 to X38								
	Speed (MP112.x)	X62 to X64			X15 to X20					
	Nominal value (MP120.x)	X57 to X59[a]			X51 to X56[a]					

a. digital, also possible with analog nominal speed value

MP110.0-8 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

MP112.0-8 Assignment of speed encoder inputs to the axes

Input: 0: No speed encoder input
15 to 20: Speed encoder inputs X15 to X20
62 to 64: Speed encoder inputs X62 to X64

MP120.0-8 Assignment of speed encoder outputs to the axes

As of software version:280 474-01

Input: 0: No servo-controlled axis
1 to 6: Analog outputs 1 to 6 at terminal X8
7 to 13: Analog outputs 7 to 13 at terminal X9
51 to 59: Digital outputs X51 to X59

Note

When assigning the speed command signal outputs of the LE 426 M and LE 430 M, please note arrangement of power modules.

MP120.0-8 Assignment of speed encoder outputs to the axes

As of software version:280 470-01

Input: 0: No servo-controlled axis
A1 to A6: Analog outputs 1 to 6 at terminal X8
A7 to A13: Analog outputs 7 to 13 at terminal X9
D1 to D6: Digital axes 1 to 6

Note

NC software 280 470-xx:
Only the values from 0 to A13 are permissible. Digital axes are entered in MP2000.

Spindles

All position encoder inputs may be used for the spindle/spindles. The possible groupings of speed encoder inputs and nominal speed value outputs can be seen in the following tables:

LE 426 PB/M, 12 000 rpm

First spindle		Second spindle	
Speed	**Nominal value**	**Speed**	**Nominal value**
X15 to X20	Digital: X51 to X56	–	Analog: 1 to 13
–	Analog: 1 to 13	–	Analog: 1 to 13

LE 426 PB/M, 30 000 rpm

First spindle		Second spindle	
Speed	**Nominal value**	**Speed**	**Nominal value**
X60	Digital: X61	X15 to X19	Digital: X51 to X56
X60	Digital: X61	–	Analog: 1 to 13
–	Analog: 1 to 13	–	Analog: 1 to 13

LE 430 PA, LE 430 M/6 Axes

First spindle		Second spindle	
Speed	**Nominal value**	**Speed**	**Nominal value**
X60	Digital: X61	X15 to X20	Digital: X51 to X56
X60	Digital: X61	–	Analog: 1 to 13
–	Analog: 1 to 13	–	Analog: 1 to 13

LE 430 M/9 Axes

First spindle		Second spindle	
Speed	**Nominal value**	**Speed**	**Nominal value**
X60	Digital: X61	X15 to X20, X62 to X64	Digital: X51 to X59
X60	Digital: X61	–	Analog: 1 to 13
–	Analog: 1 to 13	–	Analog: 1 to 13

MP111 **Position encoder input for the spindle(s)**

As of software version:280 474-03

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(s)**

As of software version:280 474-03

MP113.0 Speed encoder for the first spindle

Input: 0: No speed encoder input
15 to 20: Speed encoder inputs X15 to X20
60: Speed encoder input X60 (only on LE with integral spindle DSP)
62 to 64: Speed encoder inputs X62 to X64

MP113.1 Speed encoder for the second spindle

Input: 0: No speed encoder input
15 to 20: Speed encoder inputs X15 to X20
62 to 64: Speed encoder inputs X62 to X64

MP121 **Nominal speed command output of the spindle(s)**

As of software version:280 474-01

MP121.0 Nominal speed command output of the first spindle

Input: 0: No servo-controlled spindle
1 to 6: Analog outputs 1 to 6 at terminal X8
7 to 13: Analog outputs 7 to 13 at terminal X9
51 to 59, 61: Digital output X51 to X59, X61

MP121.1 Nominal speed command output of the second spindle

Input: 0: No servo-controlled spindle
1 to 6: Analog outputs 1 to 6 at terminal X8
7 to 13: Analog outputs 7 to 13 at terminal X9
51 to 59: Digital outputs X51 to X59

Note

Remember the arrangement of power modules when assigning the speed command signal outputs of the LE 426 M, LE 430 M.

MP121 **Nominal speed command output of the spindle**

As of software version:280 470-01

Input: 0: No servo-controlled spindle
A1 to A6 or 1 to 6: Analog outputs 1 to 6 at terminal X8
A7 to A13 or 7 to 13: Analog outputs 7 to 13 at terminal X9
S1: Digital spindle

Note

Only the values from 0 to A13 are permissible. Enter the digital spindle in MP2001.

6.1.5 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 the axes 1 to 9 and bit 15 represents the spindle. If status information is read for only one axis, only bit 0 is changed. The following table shows the meanings of the return codes:

Status information	Meaning
0	0: Axis (spindle) not active (MP10 or MP3010 or no encoder) 1: Axis (spindle) active
1	Depending on the current traverse range: 0: NC axis or not active 1: PLC axis
2	0: No 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-coded output for all axes: –1
PS    B/W/D/K   <Status information>
                See table above
CM    9038
PL    B/W/D     <Information>
```

Error recognition:

Marker	Value	Meaning
M4203	0	Information was read
	1	Error code in W1022
W1022	1	Status information not available on this TNC
	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

6.1.6 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. (See “Tool Changer” on page 6 – 400).

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 recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid
	20	Call was not in a submit or spawn job

Selecting the traverse range

You can switch the range of traverse in two ways:

▶ Select the traverse range with Module 9151 or 9152. With Module 9152 you can change the axis display at the same time.

or

▶ Select the traverse range with M4574 and M4575 according to the table below.
▶ In all operating modes you must activate the traverse range with strobe marker M4135.

You may only use the traverse range switching function via M4574 and M4575 during an M/S/T/Q strobe in all operating modes (except for **Manual Operation** and **El. Handwheel**).

M4574	M4575	Traverse range/Datum
0	0	Area 1
1	0	Area 2
0	1	Area 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 Selecting traverse range and axis designation

As of NC software: 280 472-01

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 correspond to axes 1 to 9
               With –1 the axis designations from M100.x are valid
PS   B/W/D/K   <Traverse range>
               0 to 2: Range of traverse
               –1: Do not change range of traverse
CM   9151
```

Error recognition:

Marker	Value	Meaning
M4203	0	Traverse range/axis designation switched over
	1	Error code in W1022
W1022	2	Invalid value for traverse range
	3	For the axis assignment, neither a string nor –1 was transferred
	21	The module was called during a running program or without an M/S/T/Q strobe

Module 9152 Selecting traverse range, axis display and axis designation

As of NC software: 280 476-01

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 correspond to axes 1 to 9
With –1 the axis designations from M100.x are valid

PS B/W/D/K <Traverse range>
0 to 2: Range of traverse
–1: Do not change range of traverse

CM 9152

Error recognition:

Marker	Value	Meaning
M4203	0	Traverse range, axis designation and axis display are switched
	1	Error code in W1022
W1022	2	Invalid value for traverse range, or string for axis configuration, axis display or key configuration is too long
	3	For the axis assignment, axis display or key configuration, neither a string nor –1 was transferred
	21	The module was called during a running program or without an M/S/T/T2/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

As of NC software 280 476-01 you can save the values for MP910.x, MP911.x, MP912.x, MP920.x, MP921.x and MP922.x with the actual-position-capture key.

MP910.0-8 **Positive software limit switches, traverse range 1 (default setting after power on)**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP911.0-8 **Positive software limit switches, traverse range 2**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP912.0-8 **Positive software limit switches, traverse range 3**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP920.0-8 **Negative software limit switches, traverse range 1 (default setting after power on)**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP921.0-8 **Negative software limit switches, traverse range 2**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]

MP922.0-8 **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 in words W1034 and W1036 to the PLC:

		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

6.1.7 Lubrication Pulse

You can define the traverse distance for each axis after which the PLC commands lubrication:

- In MP4060.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.

Note

After the TNC has been reset, the accumulated distance is reset.

MP4060.0-8 **Traverse distance for lubrication of axes 1 to 9**
Input: 0 to 99 999.9999 [mm] or [°]

		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 correspond to 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 positioning 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.
- The 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 correspond to 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 correspond to 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 correspond to 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 correspond to 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 correspond to 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 correspond to 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 and also the main spindle directly through the PLC. ((See “Spindle” on page 6 – 180).)

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 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 – 28)

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, for example the feed rate, during positioning.

The following conditions apply to a PLC positioning command:

- If more than one axis is moved simultaneously, then the axes will be interpolated.
- If you start another axis during a PLC positioning movement,
  - then 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 graphic.

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 correspond to 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 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 running 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 correspond to 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 attained
		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. 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 TNC can compensate the following mechanical axis errors:

- Backlash
- Linear axis errors
- Nonlinear axis errors
- Thermal expansion
- Reversal spikes during circular movements
- Stiction

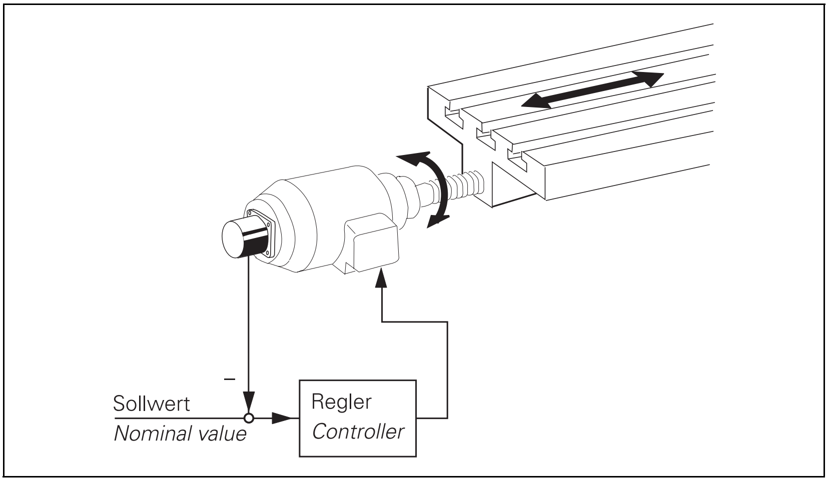
Per axis you can activate either the linear or the nonlinear axis error compensation. All other types of compensation are nonexclusive.

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:

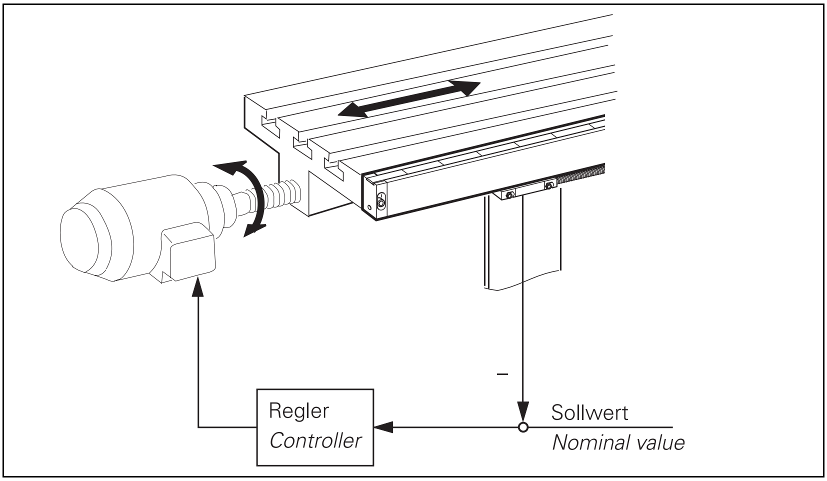
- In MP710, enter the value that the TNC should add to or subtract from the encoder signal after a reversal in direction.

MP710.1-8 Backlash compensation for axes 1 to 9
Input: -1.0000 to +1.0000 [mm] or [°]

Cause within the control loop

> **Note**
>
> Available as of NC software 280 470-08 and 280 472-01!

If axis movement is measured with a linear encoder, the TNC 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, enter the reversal error in mm.
- In MP752, enter the time in which the distance to be compensated should be traversed.

MP750.0-8 **Backlash in axes 1 to 9**
Input: -1.0000 to +1.0000 [mm] or [°]

MP752.0-8 **Compensation time for backlash in axes 1 to 9**
Input: 0 to 1000 [ms]
Example:

MP750: 0.03 mm
MP752: 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:

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

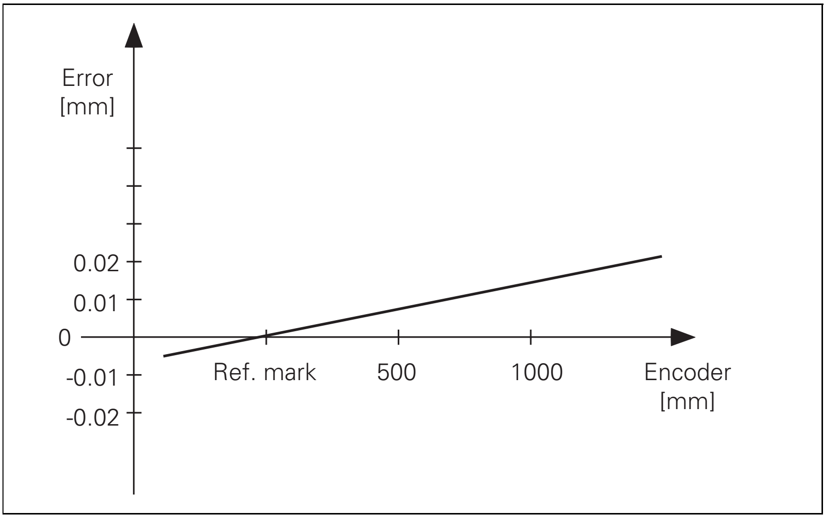
6.4.2 Linear Axis Error Compensation

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 long.
Negative linear axis error: The table moves too short.



Compensation:

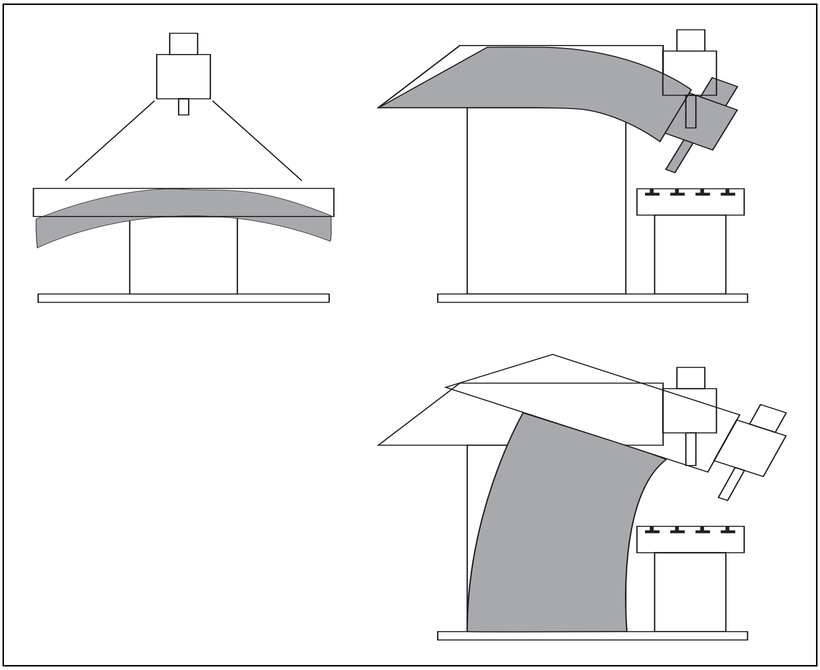
- In MP720, enter the axis error in [mm/m].
- With MP730, activate the linear axis error compensation.

MP720.0-8	**Linear axis error compensation for axes 1 to 9**
Input:	–1.000 to +1.000 [mm/m]

MP730	**Selection of linear/nonlinear axis error compensation**
Format:	%xxxxxxxxx
Input:	Bits 0 to 8 correspond to axes 1 to 9 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:

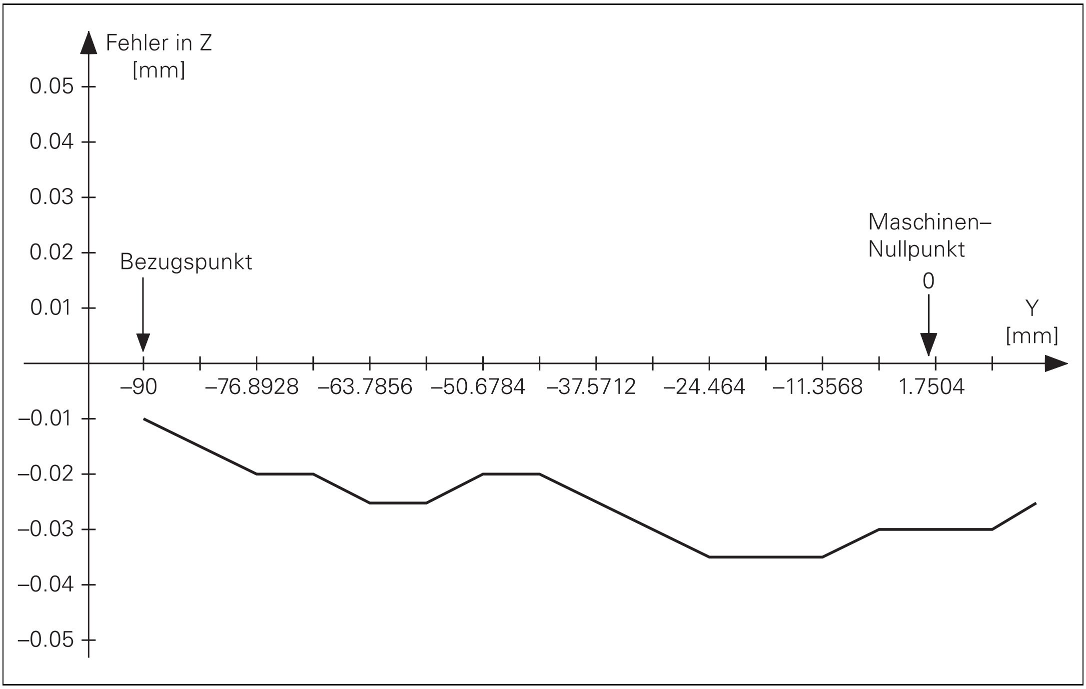


The best way to measure nonlinear axis error is with a comparator measuring system such as the HEIDENHAIN VM 101.

Note

The TNC 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 TNC:

▶ 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 TNC interprets 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:

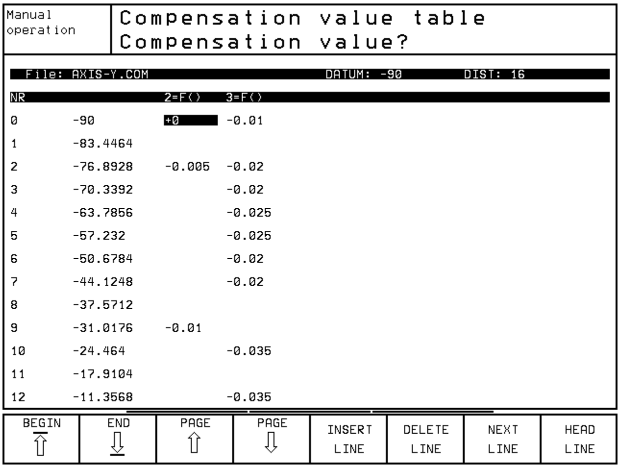
Input values for the spacing of the compensation points:
possible powers of 2^{16} = 6.5536 mm

Number of compensation points:

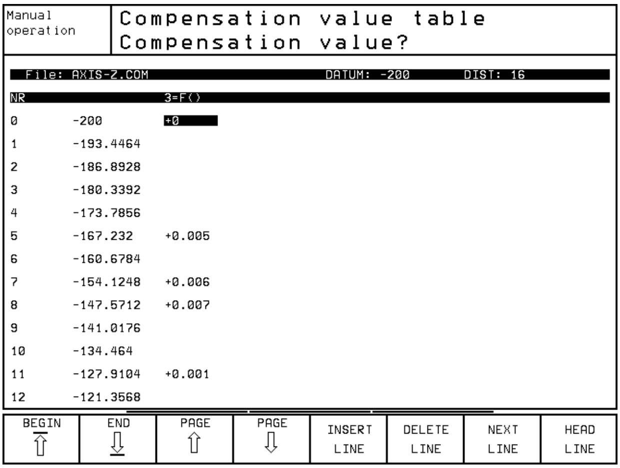
$$\frac{500\text{ mm}}{6.5536\text{ mm}} = 77\text{ compensation points in Y}$$

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

Entries:



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

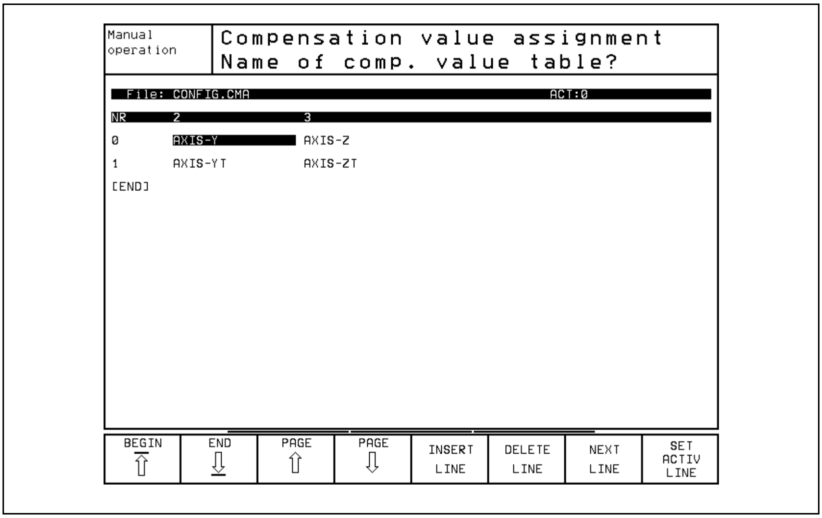
The appropriate machine parameter must be set for nonlinear axis error compensation, and the compensation value table must be registered in a configuration file:

- 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 a .CMA extension.
- 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 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 .CMA extension 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: %xxxxxxxxx

Input: Bits 0 to 8 correspond to axes 1 to 9
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 in 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 recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid
	20	Call was not in a submit or spawn job

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 on 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” on page 6 – 75).

The actual value display does not change during the 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 0.005 [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:

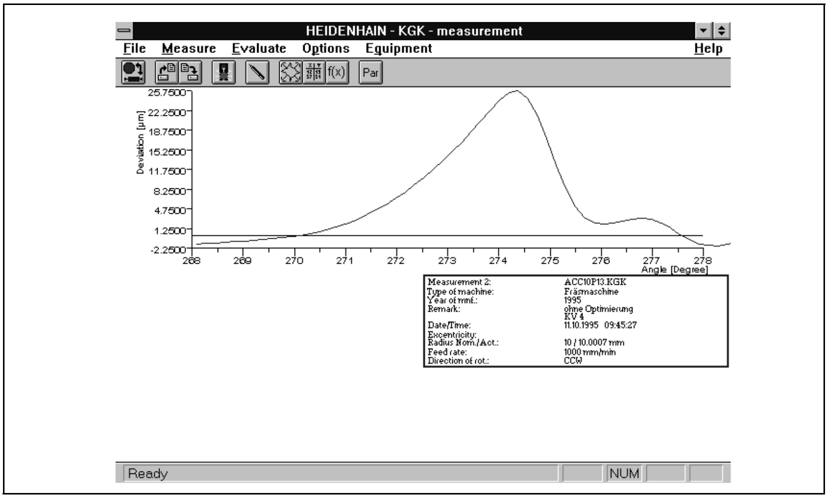
PS	B/W/D/K	<Axis> Axis 0 to 8
PS	B/W/D/K	<Compensation value> Range: –30 000 to +30 000 [1/10μ]
CM	9231	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Invalid axis number
	2	Invalid compensation value
	24	The module was called in a spawn job or submit job

6.4.5 Compensation of Reversal Spikes during Circular Traverse

The 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:

$$\text{Comp. } [mm] = \frac{\text{Reversal spikes } [\mu m] \cdot \text{ control loop cycle time } [s] \cdot 10^{-3}}{0{,}5 \cdot t_{SpD}[s]}$$

The compensation value is entered in MP712.x.

Compensation

Digital axes:

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 – 50).

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).

MP711.0-8 Height of the spikes during circular movement (only analog) for axes 1 to 9

Input: -1.0000 to +1.0000 [mm] (digital: 0)

MP712.0-8 Compensation value per control loop cycle time for axes 1 to 9

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

MP715.0-8 With M105, height of the spikes during circular movement (only analog) for axes 1 to 9

Input: -1.0000 to +1.0000 [mm] (digital: 0)

MP716.0-8 With M105, compensation value per control loop cycle time for axes 1 to 9

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 TNC. You can measure following error by using, for example, the integrated oscilloscope of the TNC.

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 at a 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.

MP1511.0-8 Factor for static friction compensation for axes 1 to 9

Input: 0 to 16 777 215 [µs]

MP1512.0-8 Limit to the amount of static friction compensation for axes 1 to 9

Input: 0 to 16 777 215 [counting steps]

MP1513.0-8 Feed-rate limitation for static friction compensation for axes 1 to 9

Input: 0 to 300 000 [mm/min]

MP1391 Velocity feedforward control in the MANUAL and HANDWHEEL operating modes

Format: %xxxxxxxxx

Input: Bits 0 to 8 correspond to axes 1 to 9
0: Operation with following error (lag)
1: Operation with velocity feedforward control

Digital axes: Limit to 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.

MP2512.0-8 Limiting the integral-action component of the speed controller for axes 1 to 9

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 TNC, 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 (MP2210) 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.

MP2610.0-8	**Friction compensation at low speed for axes 1 to 9 (effective only with velocity feedforward control)**
Input:	0 to 30.0000 [A] 0: No friction compensation (or axis is analog)
MP2612.0-8	**Delay of friction compensation for axes 1 to 9 (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)
MP2620.0-8	**Friction compensation at rated speed for axes 1 to 9**
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 TNC using the **tilt working plane** function. The user programs the part program in the X/Y plane and the TNC 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.

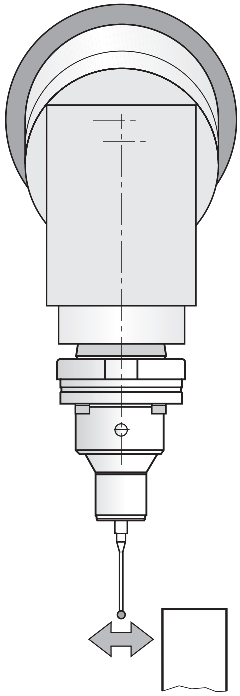
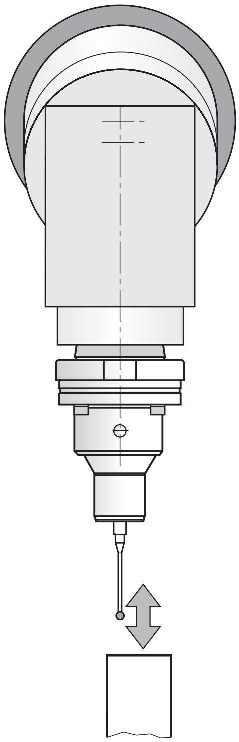
> **Note**
>
> The 3-D ROT function must be inactive during the entire measuring process.

Double swivel head 45°

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

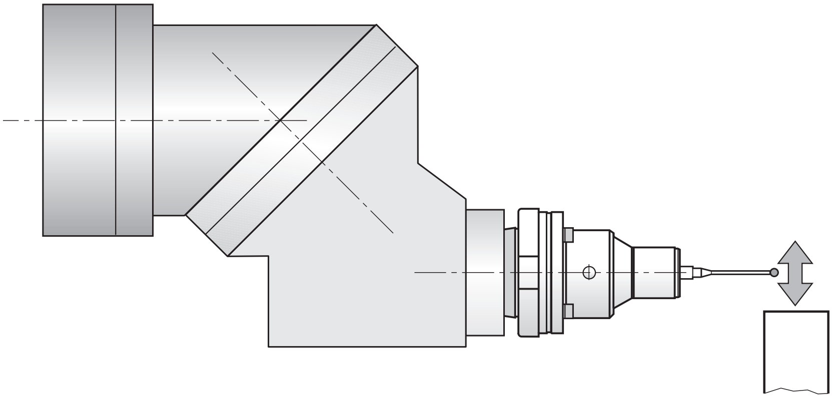
Input values for the machine parameters:

- 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 ■ Bring tilting axes B and C into 0° position ■ Probe surface X1 ■ Set X = 0	
Step 1b ■ Probe surface Z ■ Set Z = 0	

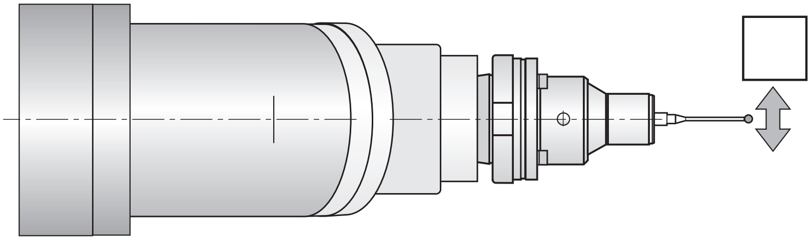
Step 2

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



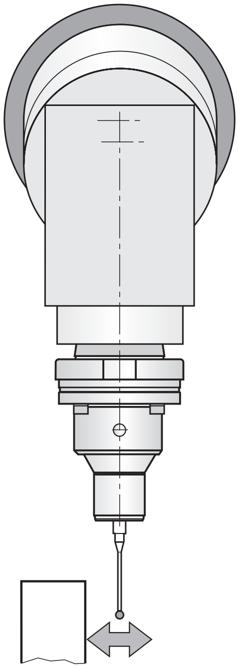
Step 3

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



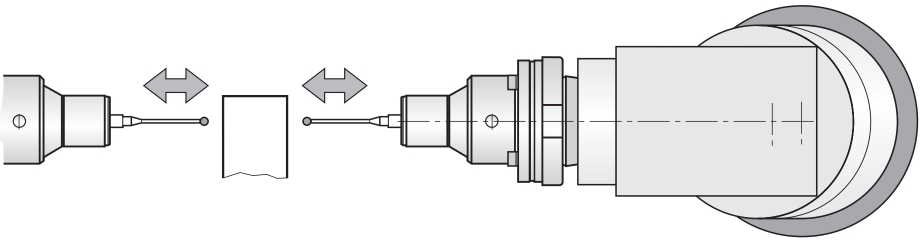
Step 4a

- Position C = 0
- Probe surface X2
- L = determined value



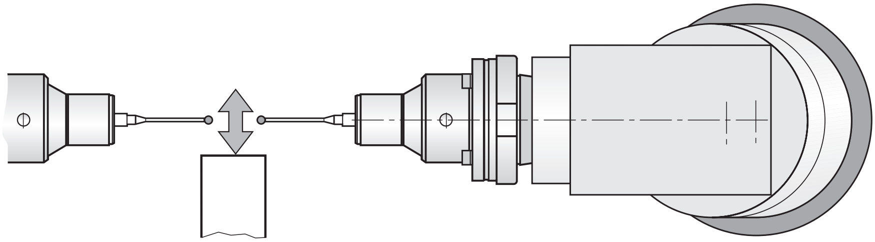
Step 4b

- 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

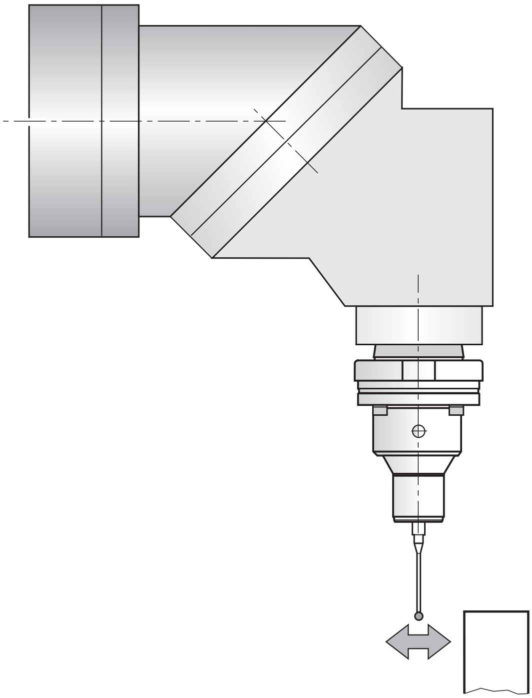


Step 5

- 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.

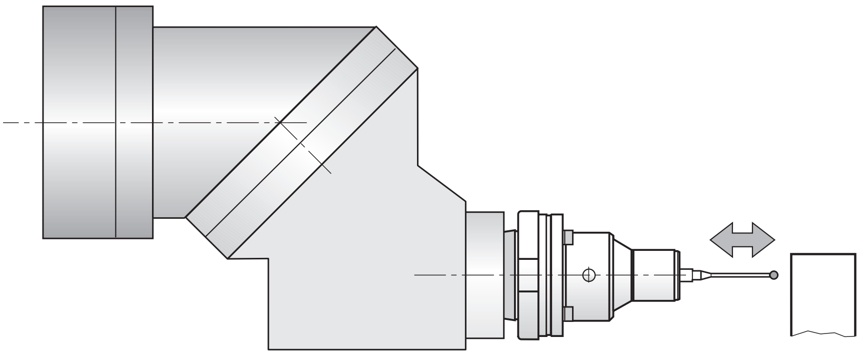
Step 6

- Probe surface Y2
- Set Y = 0



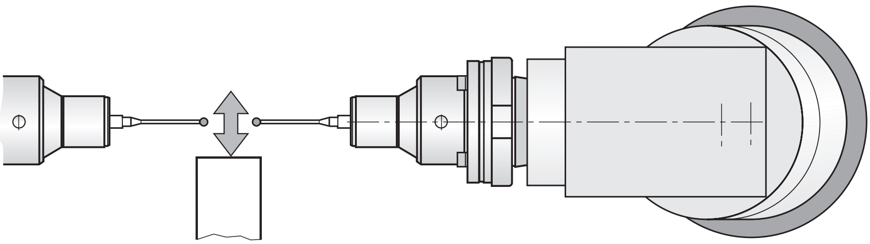
Step 7

- 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)

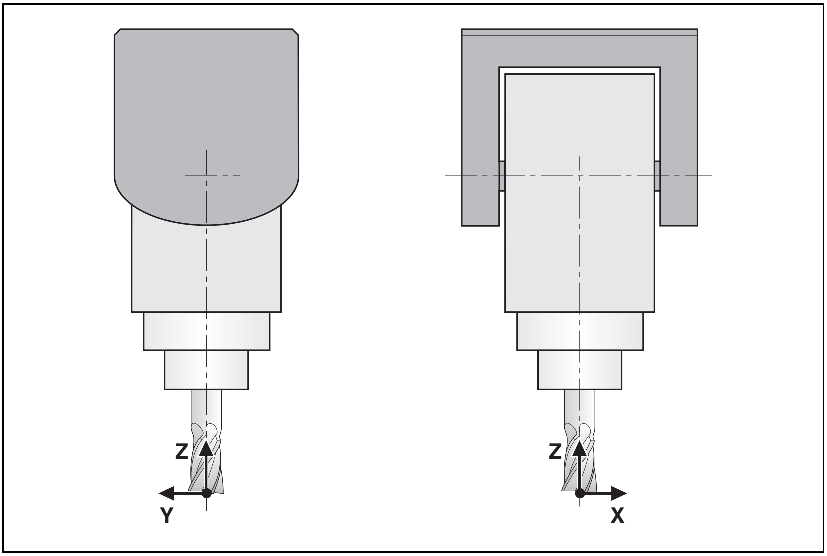


Step 8

- 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)

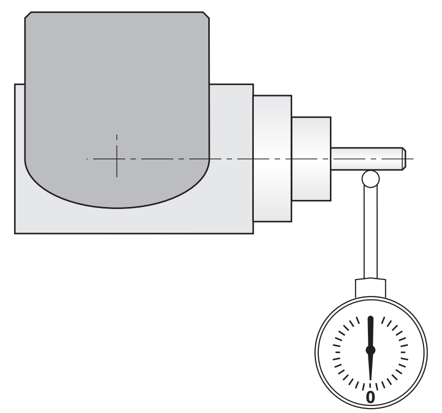


Forked swivel head In this example, the mechanical offset of a forked swivel head is determined with a dial indicator and a cylinder with a known diameter.



Temporary input values for the machine parameters:

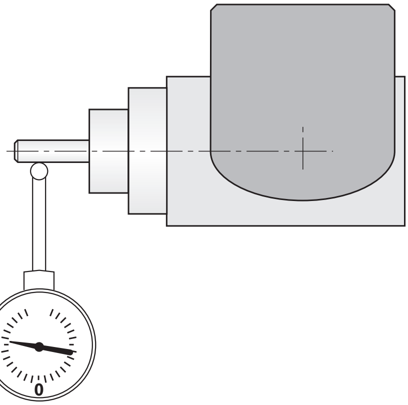
- MP7500 = %xxxx101
- MP7510.0 = %000100 ; Shift in Z axis
- MP7510.1 = %000010 ; Shift in Y axis
- MP7510.2 = %001000 ; Free tilting axis A
- MP7510.3 = %000001 ; Shift in X axis
- MP7510.4 = %000010 ; Shift in Y axis
- MP7510.5 = %100000 ; Free tilting axis C
- MP7510.6 = %000000 ; End transformation

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

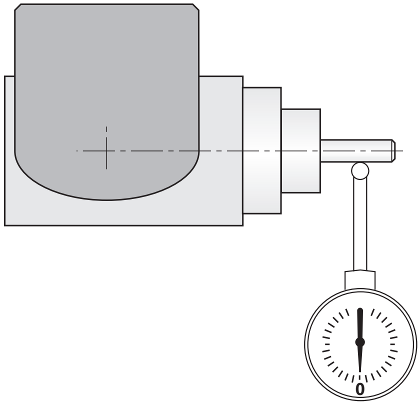
Checking the settings:

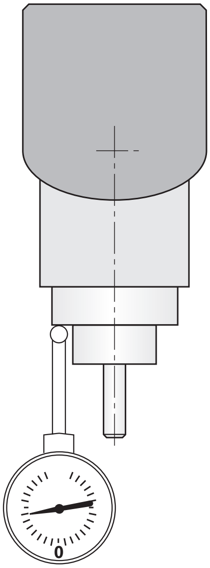
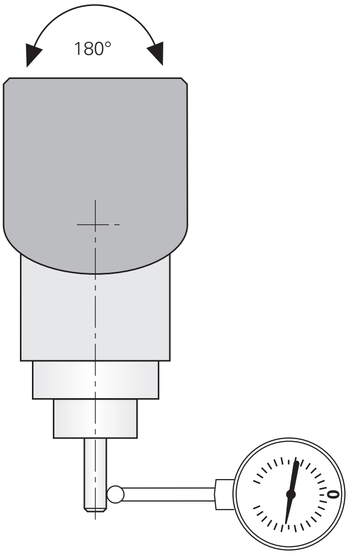
- Activate 3-D ROT
- Position A = +90
- Set dial indicator to 0
- Datum setting
- Position A = –90
- Probe same position again
- Display and dial indicator must read 0

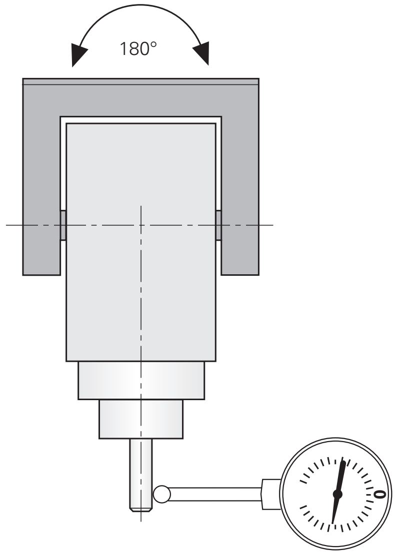
Step 2a

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

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.

Compensation of mechanical offset when exchanging the spindle head:

- With MP7500 bit 4 = 0, the mechanical offset is only compensated when M128, M114 or “tilted working plane” is called.
- With MP7500 bit 4 = 1 you must compensate the mechanical offset by means of a PLC datum shift. This allows the mechanical offset to be compensated during all tilting axis movements, and not just when M128, M114 or “tilted working plane” is called. You can also use functions M144 or M145; (See “Miscellaneous function M144/M145” on page 6 – 84).

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.

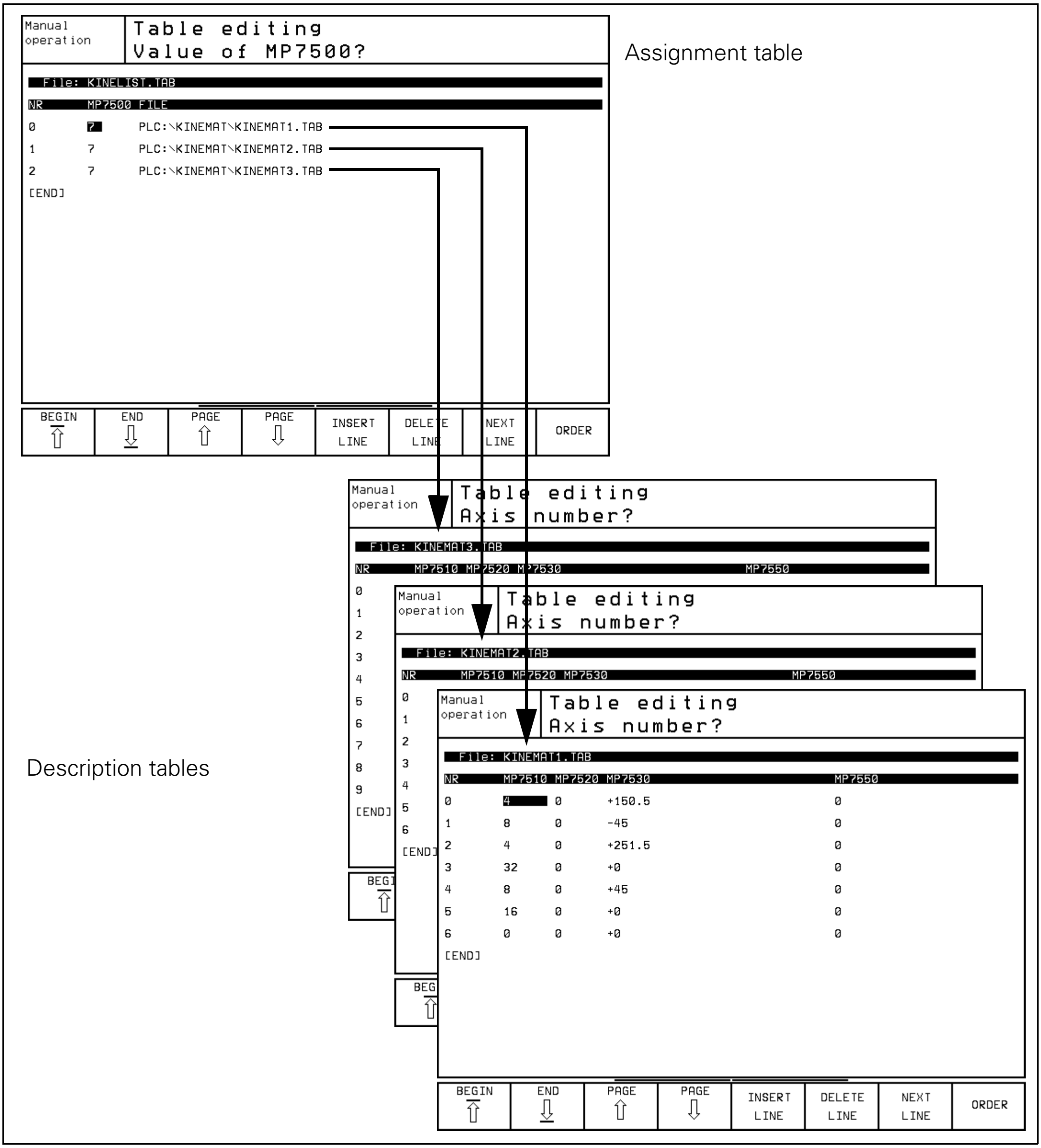
Description tables
The description table contains the contents of machine parameters MP7510.x, MP7520.x, MP7530.x and MP7550.x. The index x corresponds to the line number.
Of course the MP7530 column may also contain formulas, such as temperature compensation with M128, etc. (See "Temperature compensation" on page 6 – 75). A formula for a permanently effective temperature compensation may be entered in the **TEMPCOMP** column. (See "Permanent temperature compensation" on page 6 – 75).

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, 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, 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.



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
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.

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 recognition:

Marker	Value	Meaning
M4203	0	Geometry description was selected
	1	Error code in W1022
W1022	2	Invalid mode; or line was not found in the assignment table; or description table was not defined, does not exist or is incomplete; or there is no KINEMATIC= entry in the OEM.SYS file.
	9	Error in the MPFILE column
	10	Error in the MP7500 column
	11	Error in the machine parameter subfile
	20	Module was not called in a spawn job or submit job
	21	Call was made during a running NC program without a strobe

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.

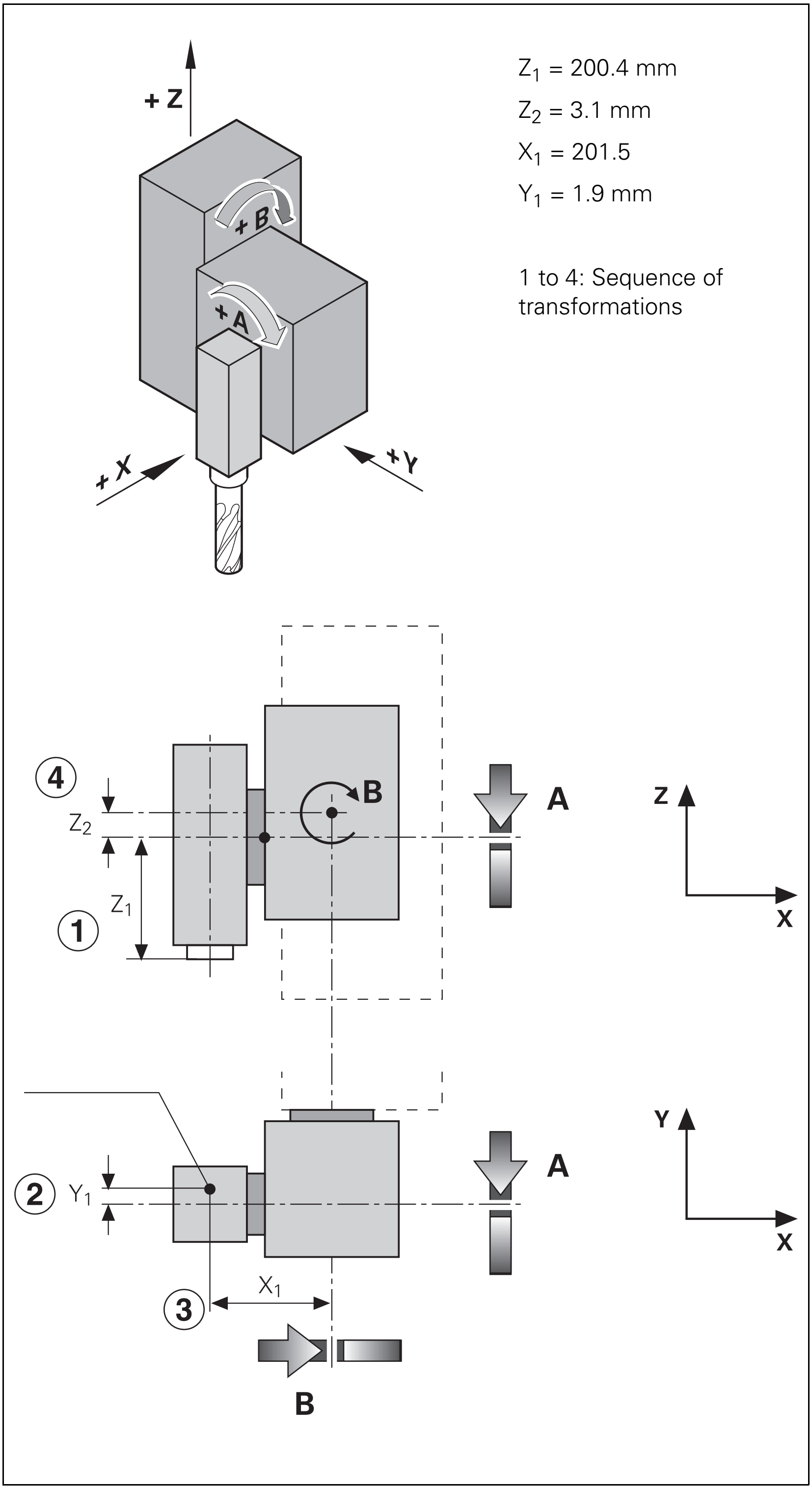
Call:

PS	B/W/D/K	<String number for table name> 0 to 7: String number (line number is also found) –1: Find only line number, no name
CM	9098	
PL	B/W/D	<Line number in the assignment table> –1: Line number not found

Error recognition:

Marker	Value	Meaning
M4203	0	Name and/or line number 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

Example 1: Rectangular double swivel head

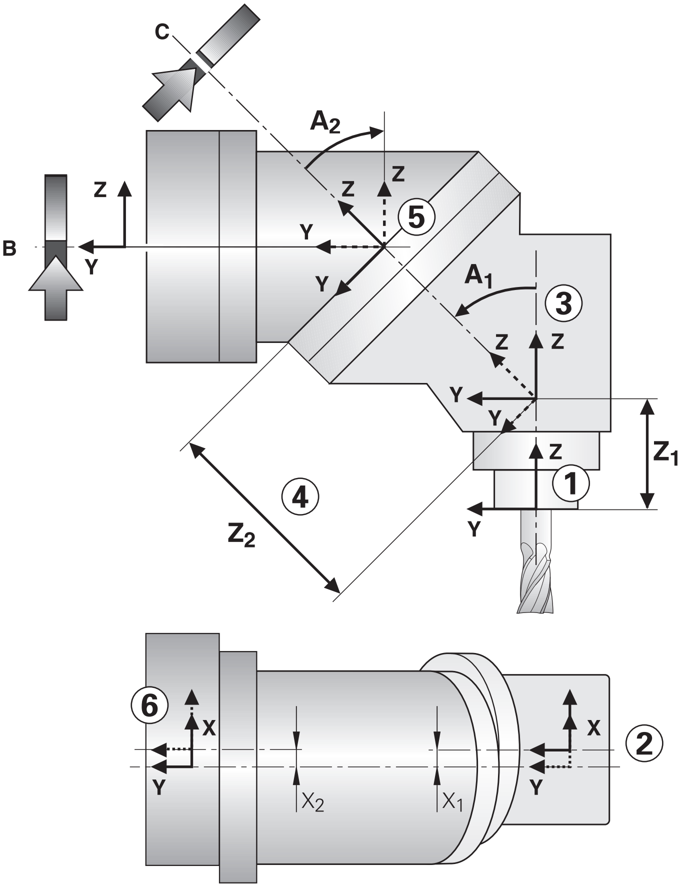


```
MP 7510.0    : %000100    ;Shift in Z axis (Z1)
MP 7510.1    : %000010    ;Shift in Y axis (Y1)
MP 7510.2    : %001000    ;Free tilting axis A
MP 7510.3    : %000001    ;Shift in X axis (X1)
MP 7510.4    : %000100    ;Shift in Z axis (Z2)
MP 7510.5    : %010000    ;Free tilting axis B
MP 7510.6 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

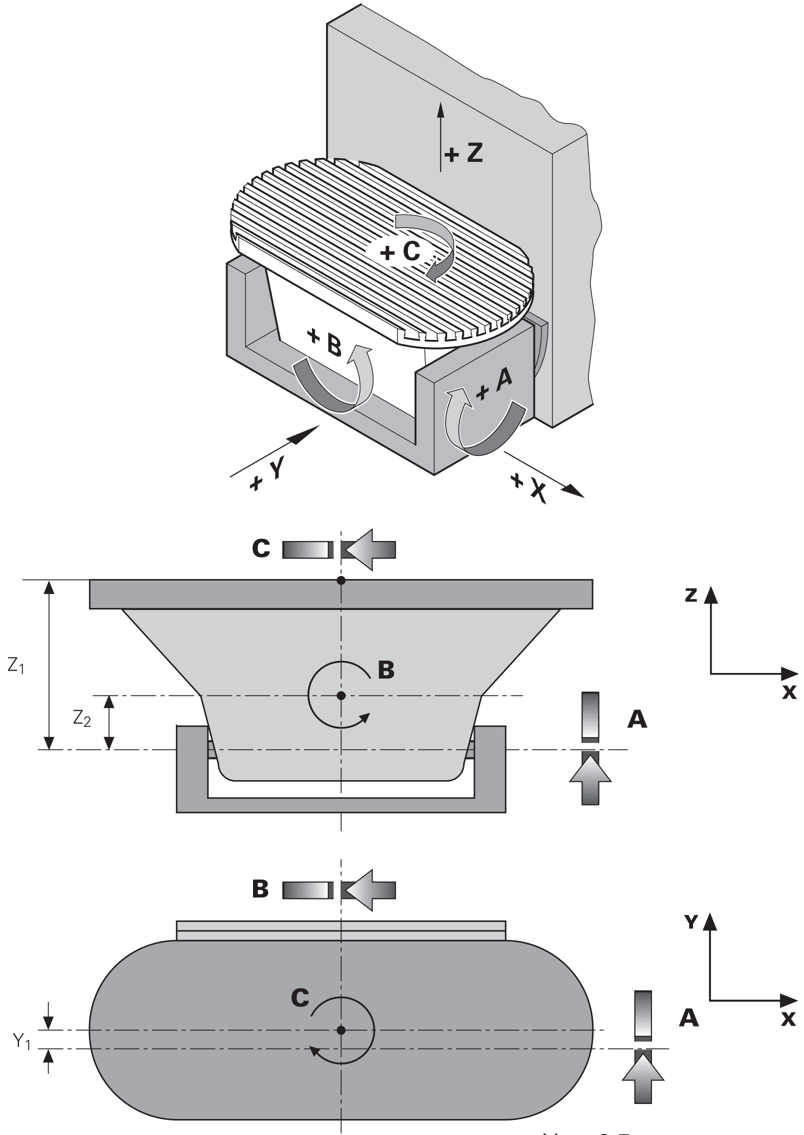
1 to 6: Sequence of transformations

```
MP 7510.0    : %000100    ;Shift in Z axis (Z1)
MP 7510.1    : %000001    ;Shift in X axis (X1)
MP 7510.2    : %001000    ;Rotate the 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 the 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.

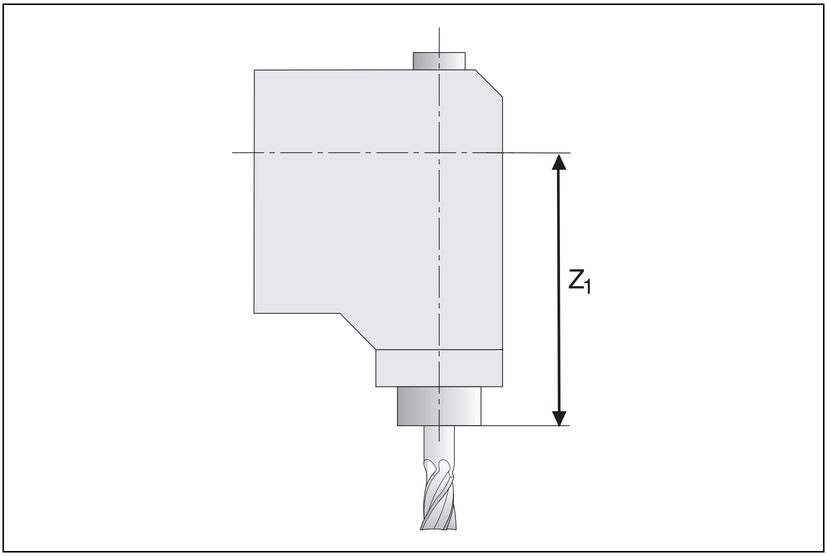
There are two possibilities for temperature compensation:

- 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 10log10
  - 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 double swivel head and a rotary table:

NR	MP7510	MP7520	MP7530	TEMPCOMP	MP7550
0	4	0	–100	–0.605e–3*W486	0
1	1	0	0	0	0
2	8	0	45	45	0
3	16	0	0	0	
4	8	0	–45	–45	
5	1	3	400.6		
6	2	3	–479.8		
7	32	1	0		
8	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 TNC 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 TNC 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 principle axis is shown on top of another principle 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 TNC 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:

The behavior of servo-controlled axes during “datum setting” depends on MP7500 bit 3, bit 5, bit 7 and bit 8:

MP7500 bit 3 = 0
During “datum setting” in X, Y, and Z, the TNC saves the reference coordinates of the tilting element needed for calculating the offset when “tilted working plane” is **active.**

MP7500 bit 3 = 1
To 280 474-06:
During “datum setting” in X, Y, and Z, the datum for a rotary table C (tool axis Z) is assumed to be 0, meaning the workpiece was aligned and datum = 0 has been set. “Tilt working plane” refers to this 0° position. A datum cannot be set when “tilted working plane” is **active.**
From 280 474-07:
During “datum setting” in X, Y, and Z, the reference coordinates of the tilting element are assumed. If “tilted working plane” is **active**, datum = 0 is assumed.
From 280 474-12 and 280 476-01:
Same as up to 280 474-06:
From 280 474-14 and 280 476-03:
Datum = 0 is assumed only for the first rotary table axis (only this axis can align a workpiece). For all other axes, “datum setting” in X, Y, and Z is possible with “tilted working plane” **active.**

MP7500 bit 5 = 0
See MP7500 bit 3 = 0

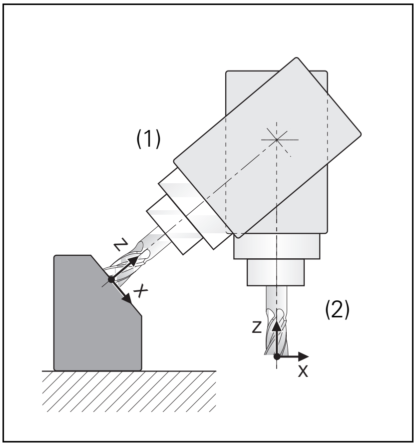
MP7500 bit 5 = 1
It can happen with titling elements with Hirth couplings that by locking the Hirth coupling, the actual value of the encoder will no longer exactly agree with the mechanical position of the tilting element. If this happens, the nominal values should be used to calculate the various datums (MP7682 bit 1).
If problems continue to occur, MP7500 bit 5 should be set to 1. The tilting angles entered in 3-D ROT are used to calculate the datums for X, Y and Z.

MP7500 bit 7 = 0
See MP7500 bit 3 = 0

MP7500 bit 7 = 1 (recommended)
During “datum setting” for X, Y and Z, the tilting angles entered in 3-D ROT are used to calculate the datum if “tilted working plane” is **active.**
During “datum setting” for X, Y and Z, the reference points of the tilting axes are used to calculate the datum if “tilted working plane” is **inactive.** This allows a workpiece to be aligned, a datum to be set, “tilt working plane” to be activated, and a new datum to be set in the “tilted working plane.”

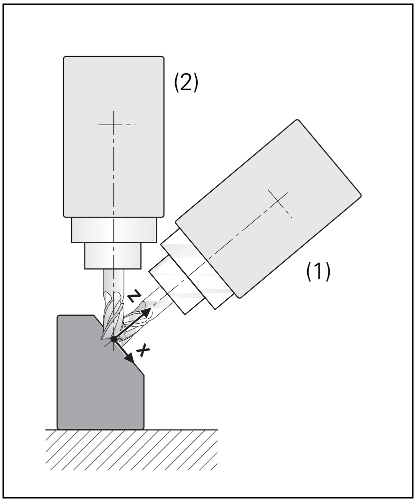
MP7500 bit 8 = 0

During "datum setting" for X, Y and Z, the position of the tilting axes depending on bit 3, bit 5 and bit 7 are considered when "tilted working plane" is **active** (1).
The reference coordinates of the linear axes are maintained when "tilted working plane" is **inactive** and the tilting element is in its home position (2).



MP7500 bit 8 = 1

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.**
Bit 7 must equal 1.



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 3, bit 5, bit 7, bit 8).

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 $\neq$ 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, C. Each designation can be used only once.
- Starting with NC software 280 476-xx, it is possible to position with M91 or M92 when Cycle 19 "tilted working plane" is active.

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 as of NC software 280 472-01: Axis sequence A variable; B variable
- Universal swivel head as of NC software 280 476-01: axis sequence A fixed; B –90°; A variable; B +90°; A fixed; C variable
- Swivel head and rotary table as of NC software 280 476-21: axis sequence B variable, A variable

With tool axis Y:

- Rotary and swivel table as of NC software 280 474-04: axis sequence B variable; A variable
- Double swivel head 45° and rotary table as of NC software 280 474-05: axis sequence A fixed; C variable; A fixed; B variable
- Rotary and swivel table as of NC software 280 474-05: axis sequence C variable; A variable

With tool axis X:

- Universal swivel head as of NC software 280 476-01:
  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 TNC calculates the position of the tilted axes of the head/table) Bit 2 – 0: The tilting axes are not positioned with Cycle 19 1: The tilting axes are positioned with Cycle 19 Bit 3 – 0: The current tilting-axis position is taken into account with respect to the machine datum 1: The 0° position is assumed for the first tilting axis Bit 4 – 0: Compensate mechanical offset during exchange of the spindle head when calling M128, M114 or “tilted working plane” 1: Compensate mechanical offset during PLC datum shift Bit 5 – 0: The current tilting-axis position is taken into account with respect to the machine datum 1: The tilting-axis position that was entered with the 3-D ROT soft key applies Bit 6 – 0: Spatial angle C is realized through a rotation of the coordinate system 1: Spatial angle C is realized through a rotation of the table Bit 7 – 0: The current tilting-axis position is taken into account with respect to the machine datum 1: The active tilting-axis position is a) derived from the tilting angles in the 3-D ROT window if manual tilting is active. b) derived from the reference coordinates of the rotary axes if tilting is inactive. Bit 8 – 0: The tilting axis positioning is considered depending on bit 3, bit 5 and bit 7 1: If manual tilting is active, the datum to be set for the principal axes X, Y and Z is recalculated back to the home position of the tilting element

MP7510 **Transformed axis**
Format: %xxxxxx
Input: 0: End of the transformation sequence
Bit 0 corresponds to axis X
Bit 1 corresponds to axis Y
Bit 2 corresponds to axis Z
Bit 3 corresponds to axis A
Bit 4 corresponds to axis B
Bit 5 corresponds to axis C
MP7510.0–14 Transformation 1 to transformation 15

MP7520 **Additional code for transformation**
Format: %xx
Input: Bit 0 – Tilting axis
0: Swivel head
1: Tilting table
Bit 1 – Type of dimension in MP7530
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

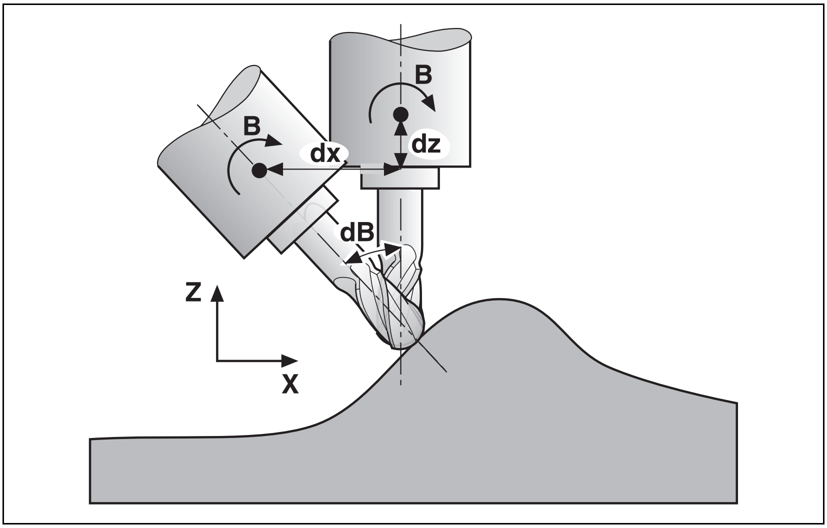
MP7530 cannot be overwritten with Module 9031 (overwrite machine parameters), since the MP contains a string, but the module transfers an integer value.

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 TNC compensates the offset of the tool that results from tilting the axes. The tool tip is always located on the programmed nominal- coordinates.



The TNC can perform a 3-D length compensation; the radius compensation must be performed by the CAD system or the postprocessor. If the TNC 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 TNC **does not take this into account.**
- As of NC software 280-472-xx: 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 TNC **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 TNC automatically performs the compensating movements in the principle axes.

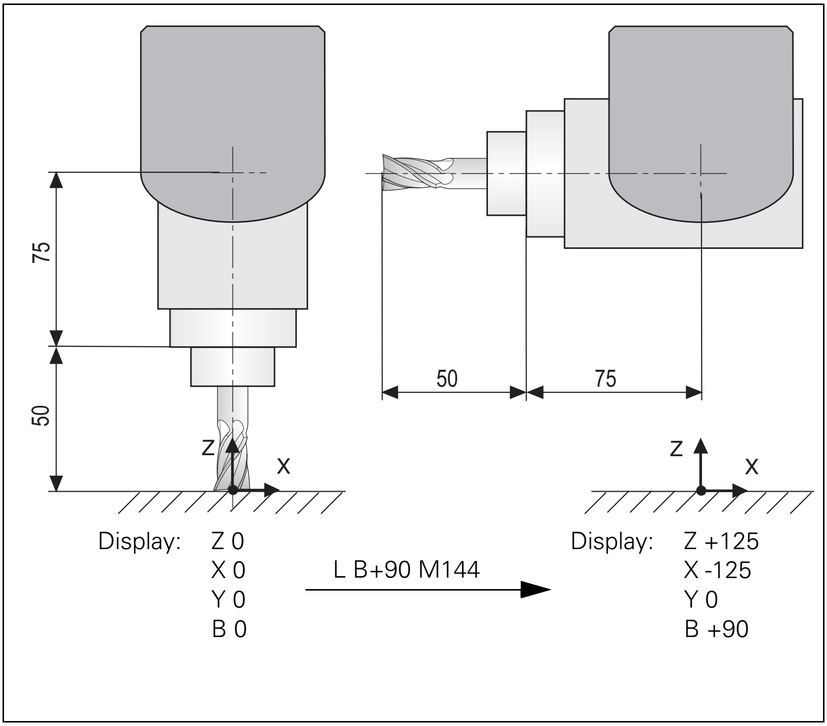
When M128 is used, the principal axes make compensating movements:

▶ In MP7471, define the maximum velocity of the principal axes during compensating movements.

MP7471	**Maximum velocity of the principal axes during compensating movements through M128**
Input:	0 to 300 000 [mm/min]

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 move only 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.

MP850.0-8 **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 TNC 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 TNC 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 you cause an offset 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.
- One axis cannot be both master and slave.
- 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.
- Up to NC software 280 470-04: Master and slave axes must be linear.

- 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.

MP855.0-8 Synchronization monitoring for axes 1 to 9
Input: 0 to 100.0000 [mm]
0: Monitoring not active

MP860.0-8 Datum for synchronization control for axes 1 to 9
Input: 0: Datum at position after switch-on
1: Datum at reference marks

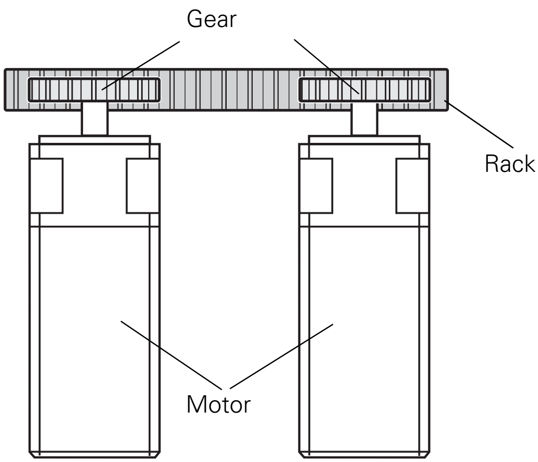
6.6.2 Master-Slave Torque Control

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.

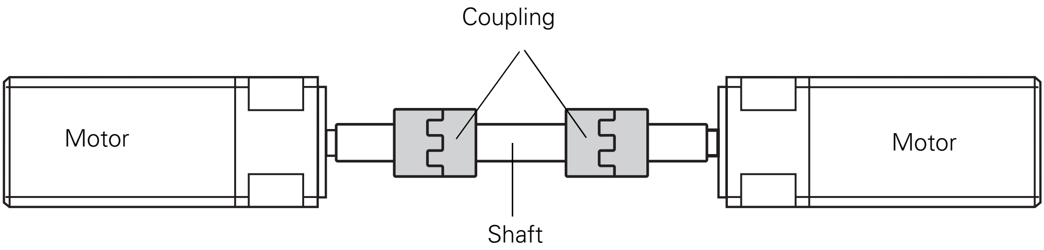
From a maximum of nine controlled axes, four times two axes can be controlled in the torque-master-slave-control, whereby you must keep in mind that the master and slave axis are on the same speed controller PCB.
First speed controller PCB: X15 to X20
Second speed controller PCB: X62 to X64

In principle there are two applications:

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

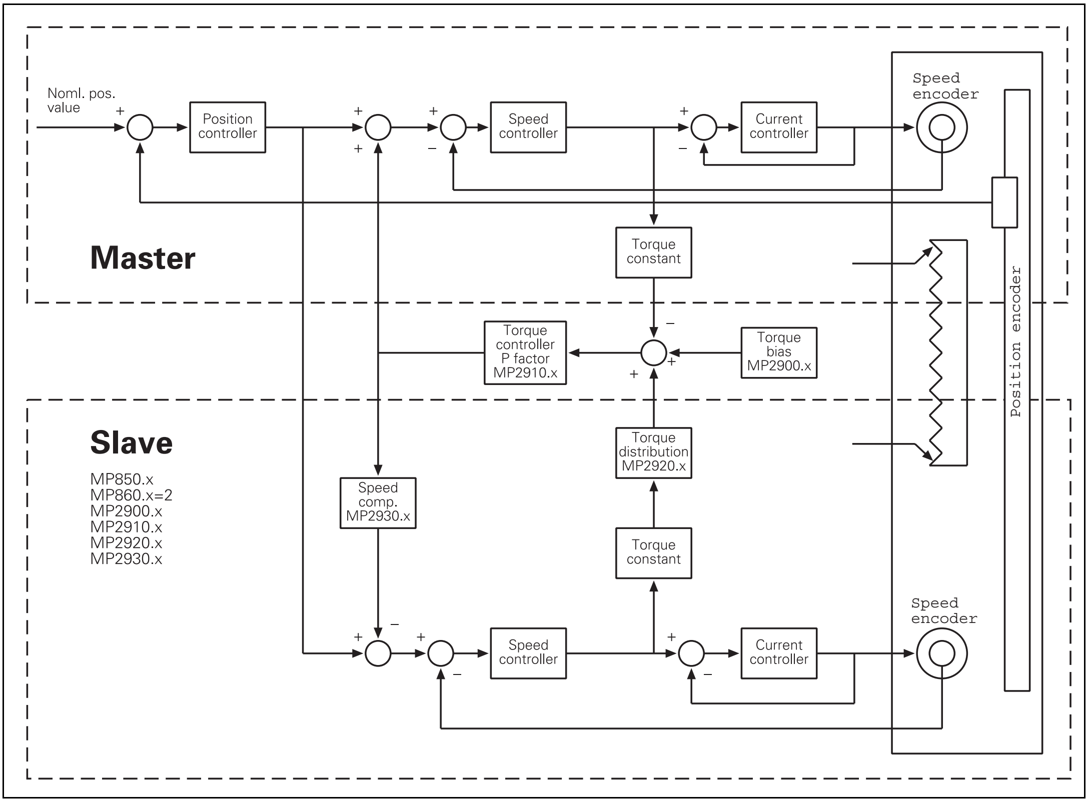


Minimization of mechanical play through mutual tensioning



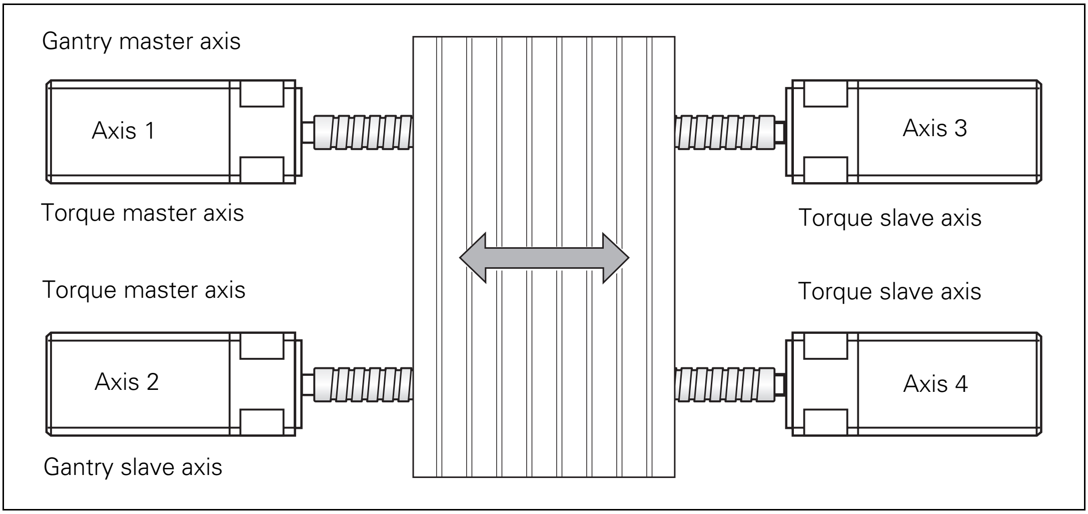
Distribution of torque with a rigid coupling

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.

MP860.0-8	**Datum for synchronization control for axes 1 to 9**
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 – 462).
- 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 – 462). 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

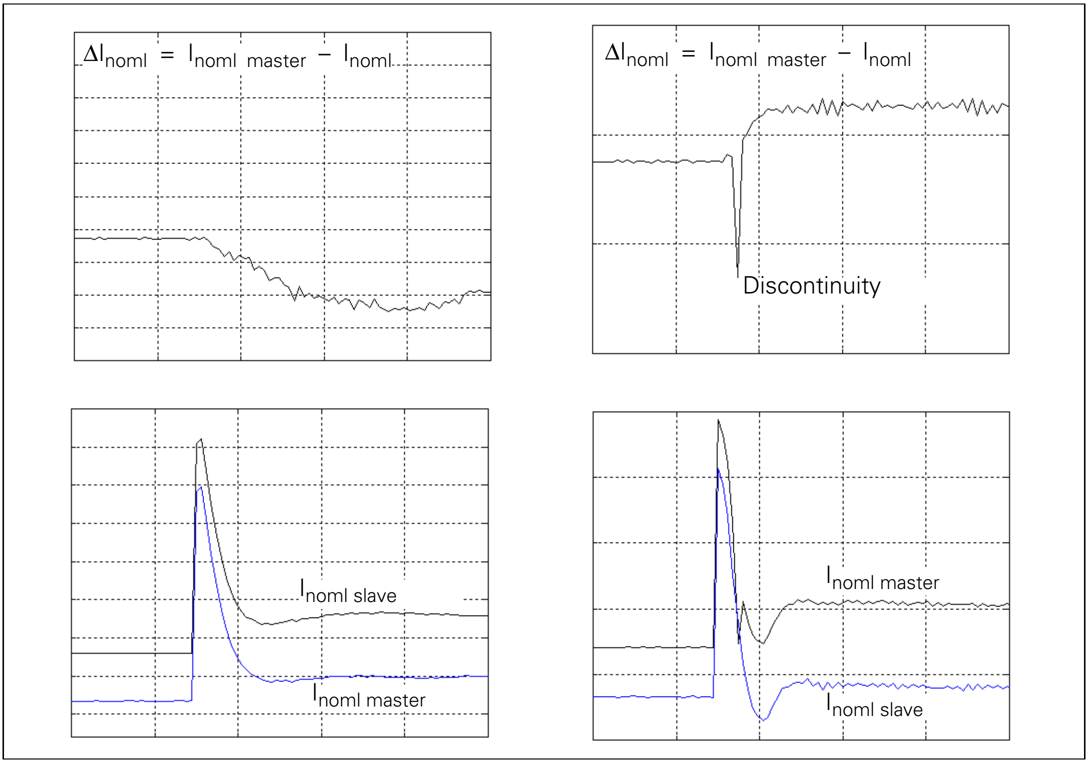
For low-frequency oscillations (< approx. 200 Hz) you should not use a filter, because it may have a negative influence on the dynamics of the control. For the mid-range frequency (approx. 200 Hz to approx. 400 Hz) ensure that you do not excite any low-frequency oscillation.
The higher the frequency 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 an 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 is the required tensioning torque (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 – 462).
- 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, 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.
- 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 – 462).
- 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.

MP2900.0-8 Tensioning torque between master and slave for master-slave torque control (entry for the slave axis)

Input: -100.00 to +100.00 [Nm]

MP2910.0-8 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)]

MP2920.0-8 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

MP2930.0-8 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

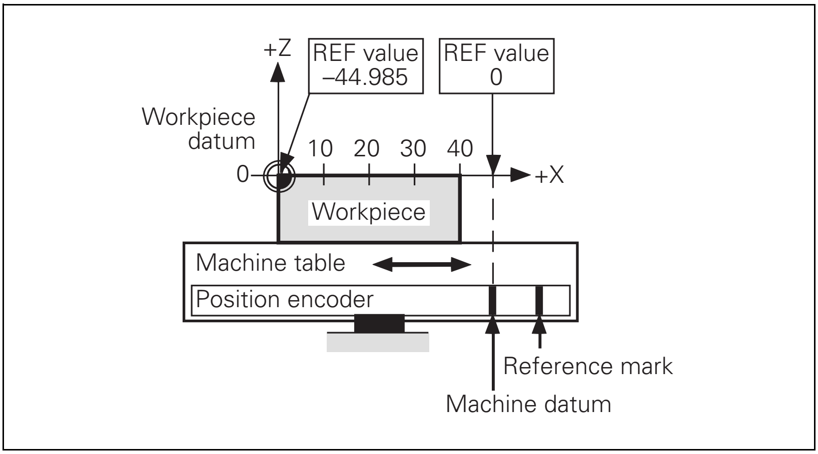
The position value (coordinates) of an axis position is defined with respect to a freely selectable datum. When the axes are moved, the ACTUAL position is calculated incrementally. If there is an interruption in power, the reference between the axis position and the position value is lost.

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 recognition:

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 the reference value to an axis
If a new reference value is assigned to an axis, the corresponding bit is reset in W1032.

Call:

PS	B/W/D/K	<Axis number> 0 to 8: axes 1 to 9
PS	B/W/D/K	<New reference value in 0.1 µm>
CM	9147	

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing strobe in M4176 = 1
	24	Module was called in a spawn 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.

"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.
As of NC software 280 476-03: 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 word 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, (See "NCMACRO.SYS" on page 7 – 44).

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 LE 426 M and LE 430 M. 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 TNC automatically attempts to communicate with the encoder.

When connecting a speed encoder with an EnDat interface as a position encoder:

- Enter MP1350 = 5.
- In MP110.x, enter 0 for the axis with EnDat interface of the speed encoder.

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.

MP960.0-8 **Machine datum for axes 1 to 9**
Input: -99 999.9999 to +99 999.999 [mm] or [°]
Values with respect to the scale reference point

MP1320 **Direction for traversing the reference marks**
Format: %xxxxxxxxx
Input: Bits 0 to 8 correspond to axes 1 to 9
0: Positive
1: Negative

MP1330.0-8 **Velocity for traversing the reference marks for axes 1 to 9**
Input: 80 to 300 000 [mm/min]

MP1331.0-8 **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]

MP1340.0-8 **Sequence for traversing the reference marks**
Input: 0: No evaluation of reference marks
1: Axis X
2: Axis Y
3: Axis Z
4: Axis 4
5: Axis 5
6: Axis 6
7: Axis 7
8: Axis 8

MP1350.0-8 **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

		Set	Reset
W272	**Operating mode** 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 correspond to axes 1 to 9	NC	NC
W1054	**Reference end position** Bits 0 to 8 correspond to 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 to traverse 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:

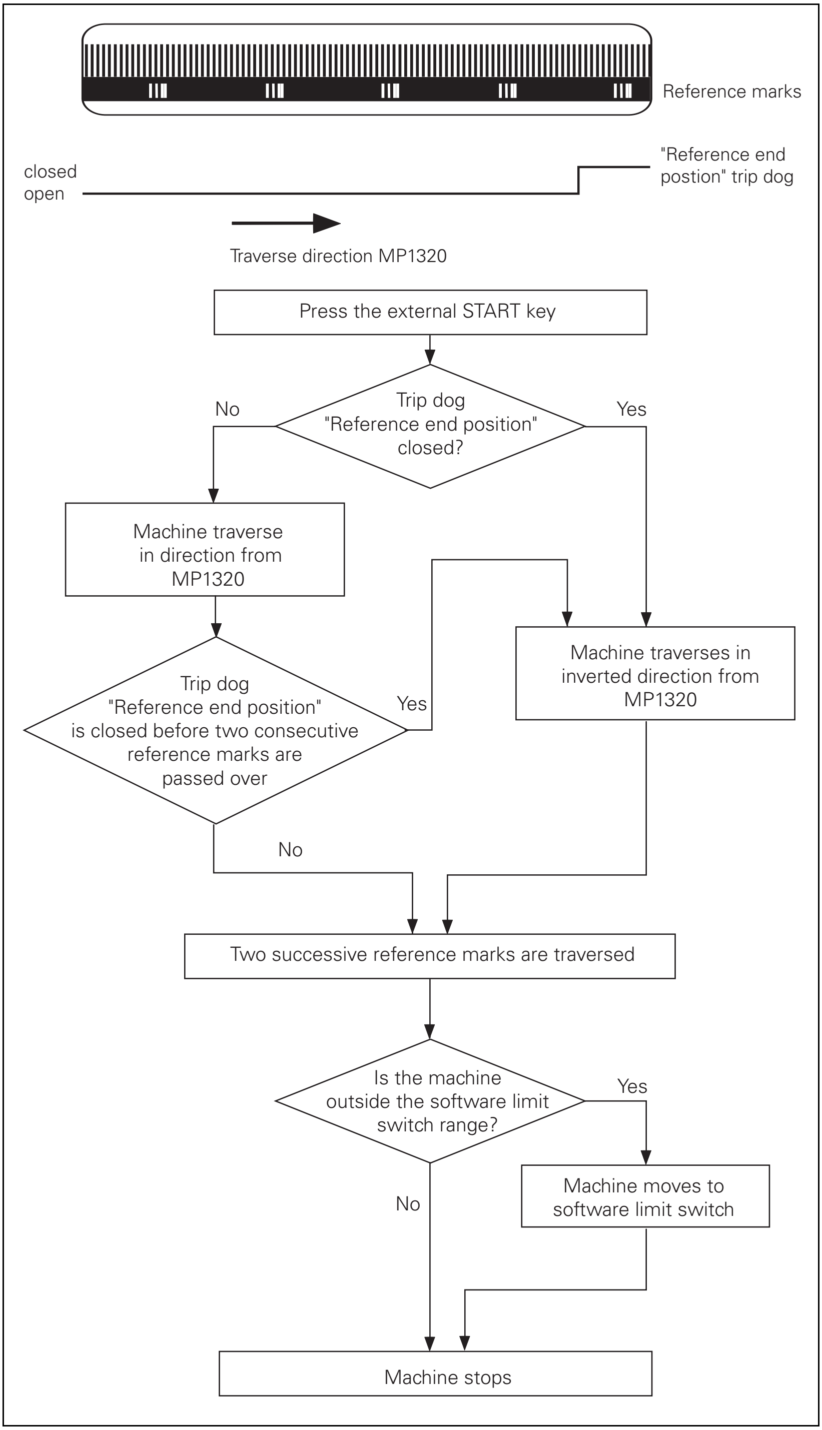
- The speed for traversing the reference mark is defined in the module.
- The spindle must be started from a standstill to traverse the reference mark.
- If the spindle is started for reference mark traverse, marker M4018 is set.

Call:

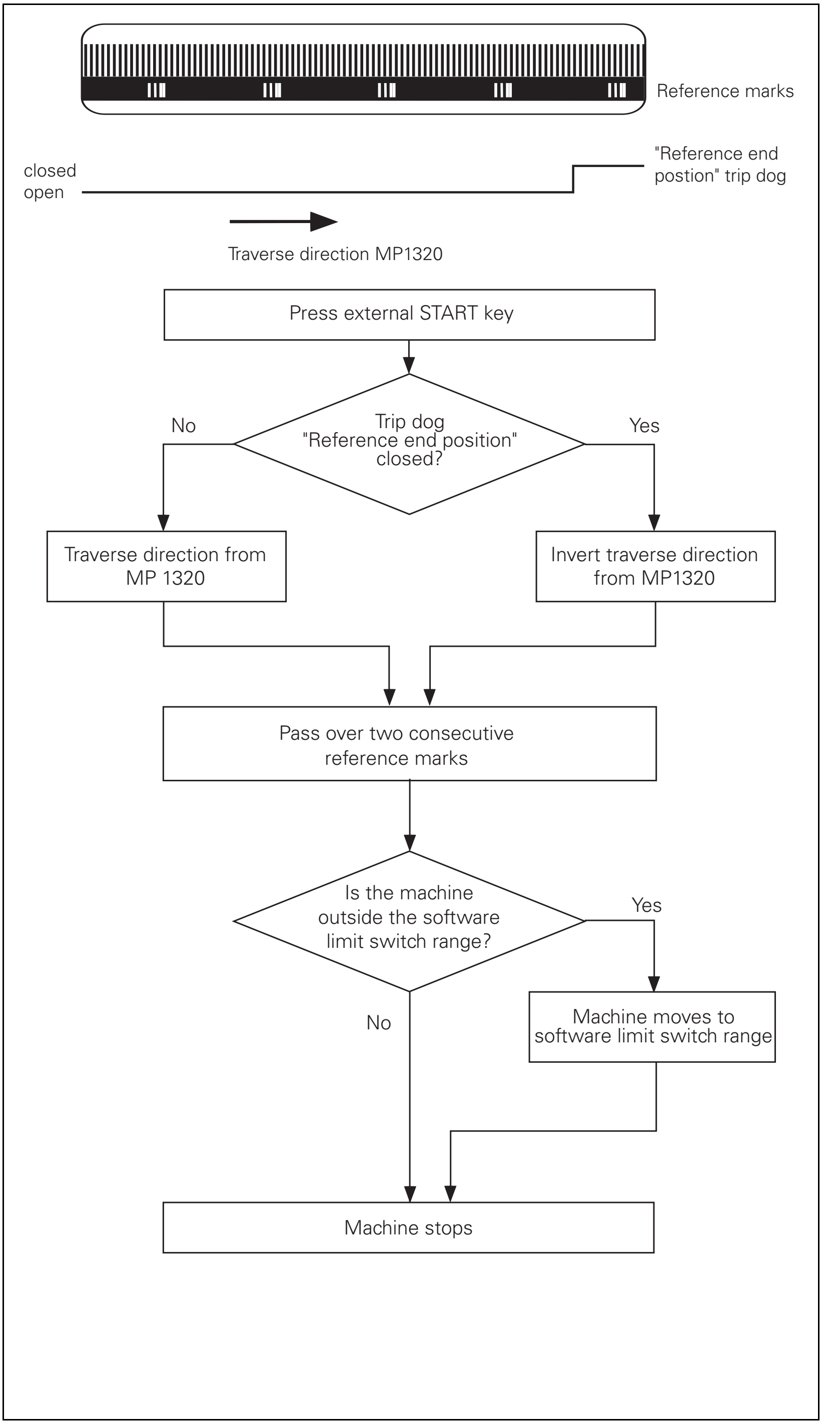
PS	B/W/D/K	<Axis/spindle> 0 to 8: Axes 1 to 9 15: Spindle
PS	B/W/D/K	<Feed rate/shaft speed> 0: Feed rate MP1330.x >0: Feed rate in mm/min or shaft speed in 1/1000 rpm
PS	B/W/D/K	<Direction of traverse> –1: Negative direction 0: Direction from MP1320.x 1: Positive direction
CM	9220	
PL	B/W/D	<Error code> 0: Reference mark traverse is commanded 1: Axis does not exist or is not a servo-controlled spindle 2: Inadmissible values for the feed rate / direction 3: Incorrect operating mode (up to 280 474-04) 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



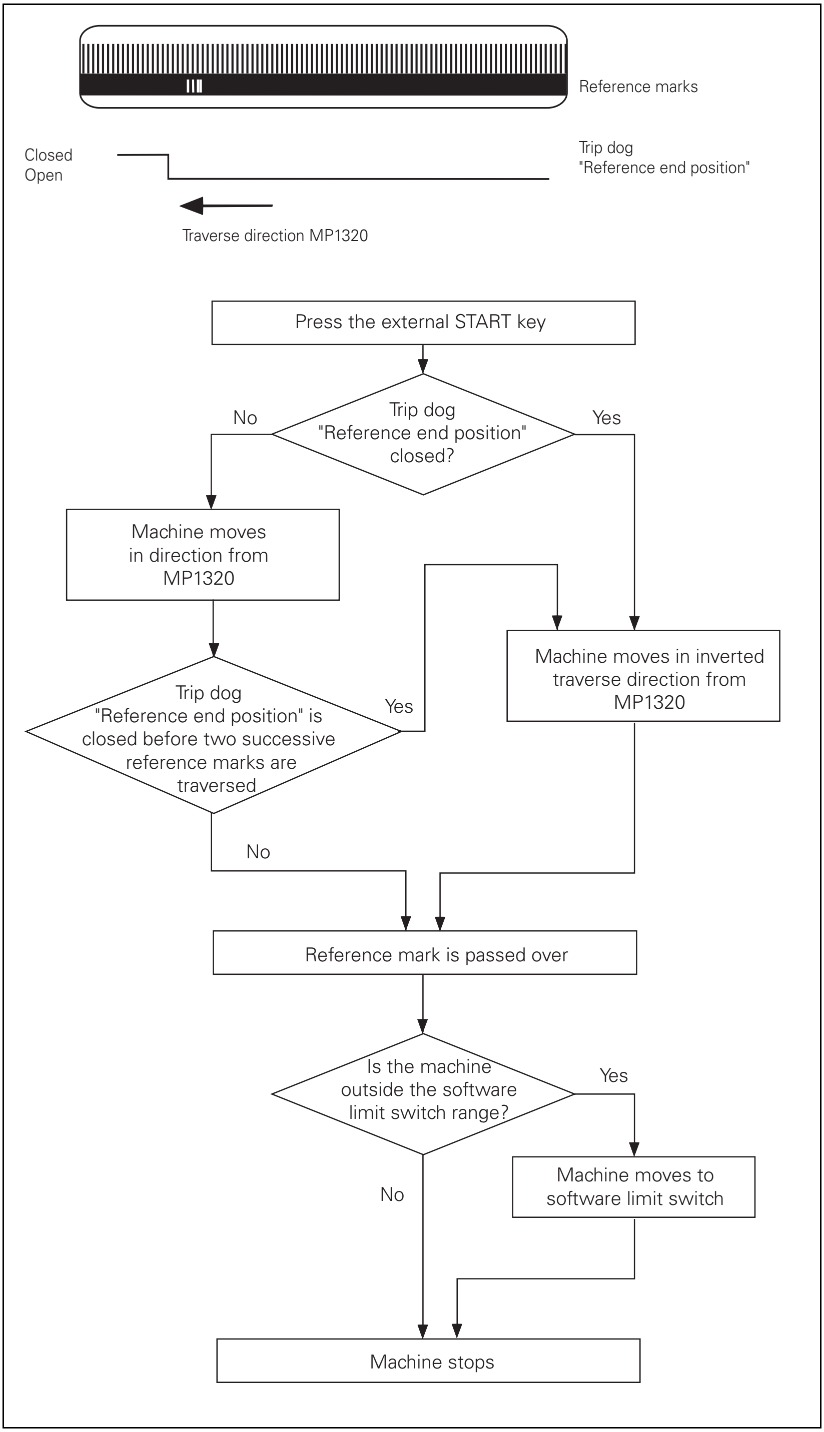
Function when MP1350.x = 0. This setting is used only to ensure compatibility. Do not use for new installations.



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

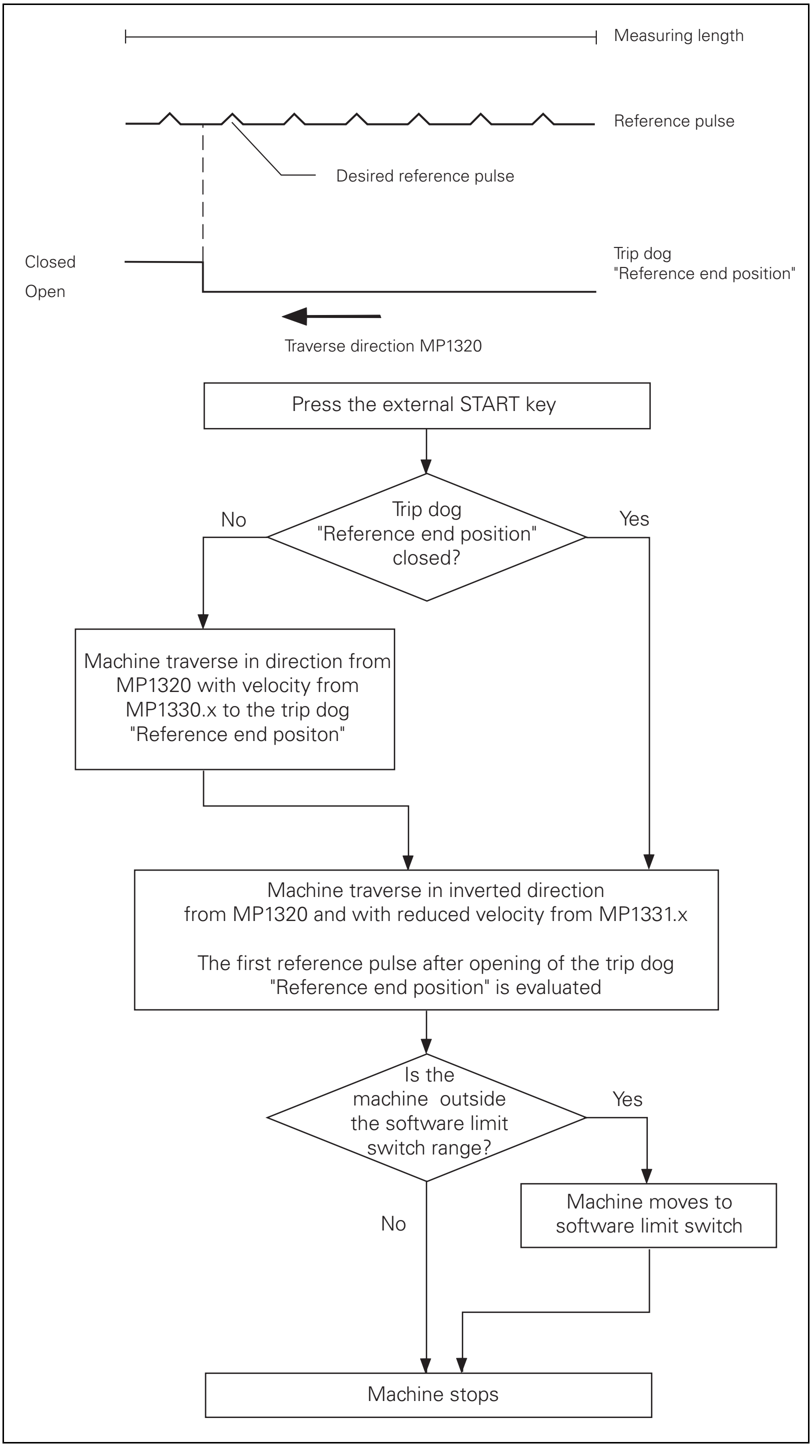
Function when MP1350.x = 1



Linear measurement through rotary encoder

Function when MP1350.x = 2

For linear measurement using a rotary encoder, a reference pulse is produced on 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.

TNC 426 CB, TNC 430 CA

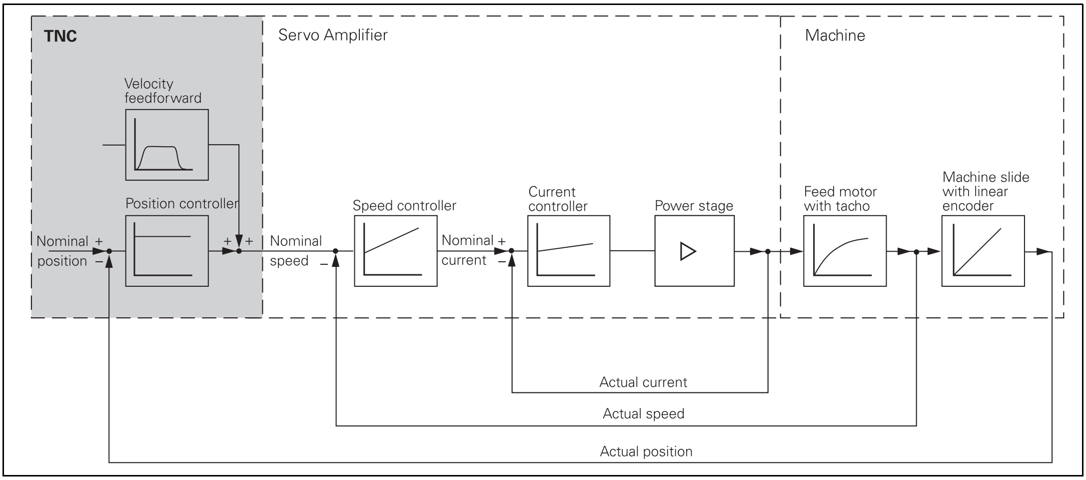
The position control loop is integrated in the TNC.

The speed controller, current controller and power supply unit are located in the servo amplifier.

The nominal speed command signal is sent by the TNC to the servo amplifier through an analog ±10-V interface.

The TNC 426 CB has an analog nominal speed command interface and controls machines with up to 5 axes plus spindle.

The TNC 430 CA has an analog nominal speed command interface for machines with up to 8 axes plus spindle. A further axis can be controlled with an additional PCB.



Note

In the TNC 426 CB, TNC 430 CA there are no machine parameters for speed and current controllers.

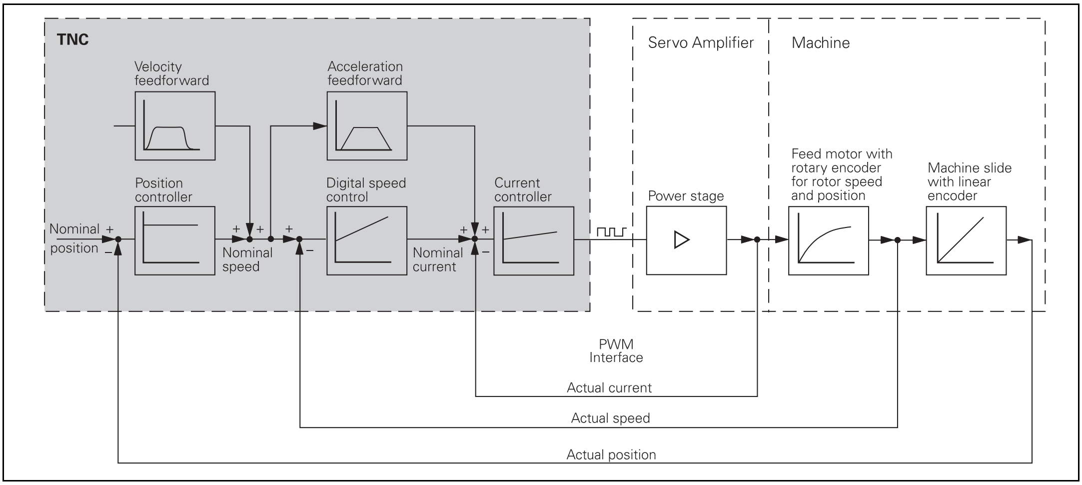
For instructions on adjusting these controllers, refer to the description of your servo amplifier.

TNC 426 PB,
TNC 430 PA

The position, speed, and current controllers are located in the TNC. The power supply unit is located in the servo amplifier. The power supply is driven by the TNC through PWM signals.
(PWM = pulse-width modulated)

The TNC 426 PB controls machines with up to 5 axes and with spindle speeds up to 12 000 rpm.
Option: 30 000 rpm for motors with two pole pairs.

The TNC 430 PA controls machines with up to 6 digitally controlled NC axes, 2 analog controlled axes and one digitally controlled spindle with a speed of up to 30 000 rpm for motors with two pole pairs.

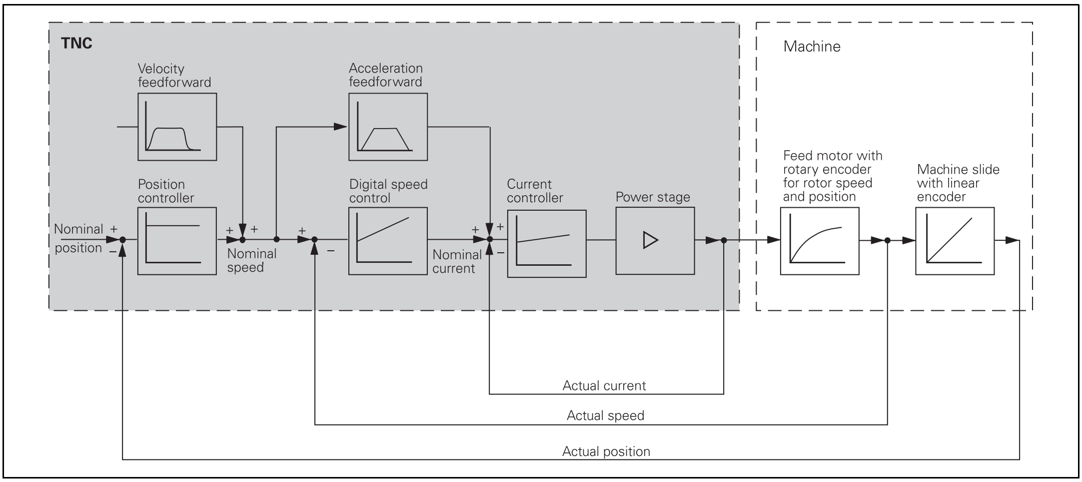


TNC 426 M,
TNC 430 M

The position controller, speed controller, current controller and the power module are located in the TNC.

The TNC 426 M controls machines with up to 5 axes and with spindle speeds up to 12 000 rpm.
Option: 30 000 rpm for motors with two pole pairs.

The TNC 430 M controls machines with up to 9 axes and with spindle speeds up to 30 000 rpm.



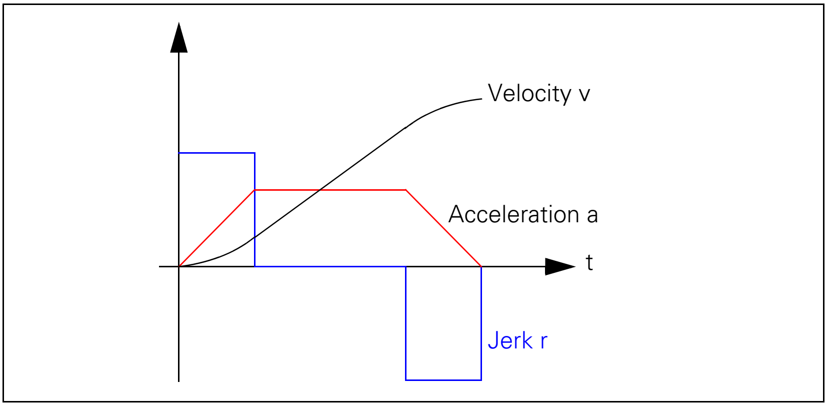
6.8.1 Relation Between Jerk, Acceleration, Velocity and Distance

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}.

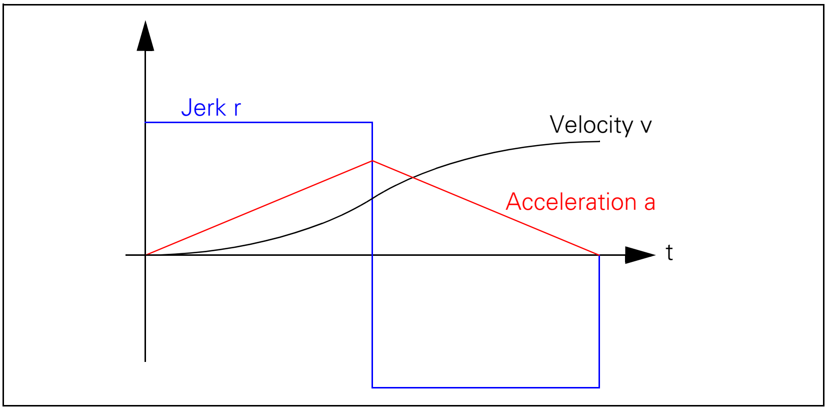
Maximum acceleration

Taking into account the motor and the power stage, 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.

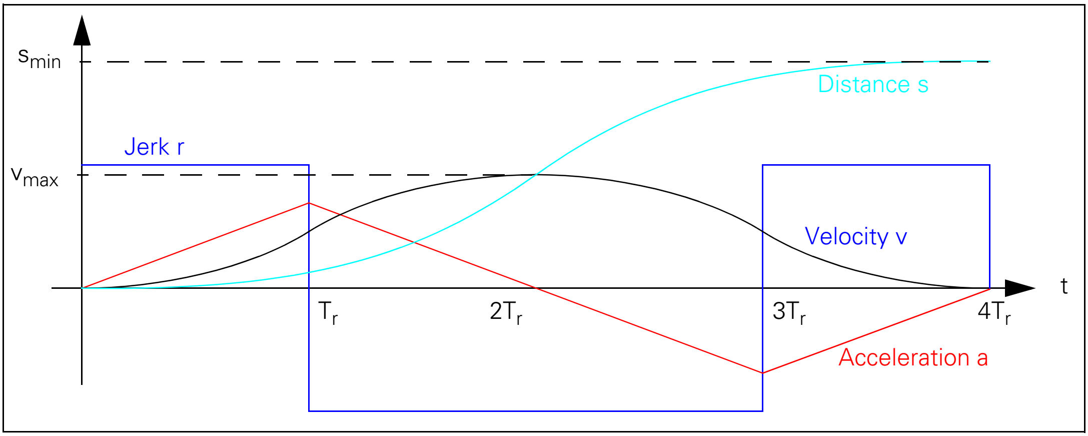
$$a_{max} = \sqrt{v_{max} \cdot r_{max}}$$



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:

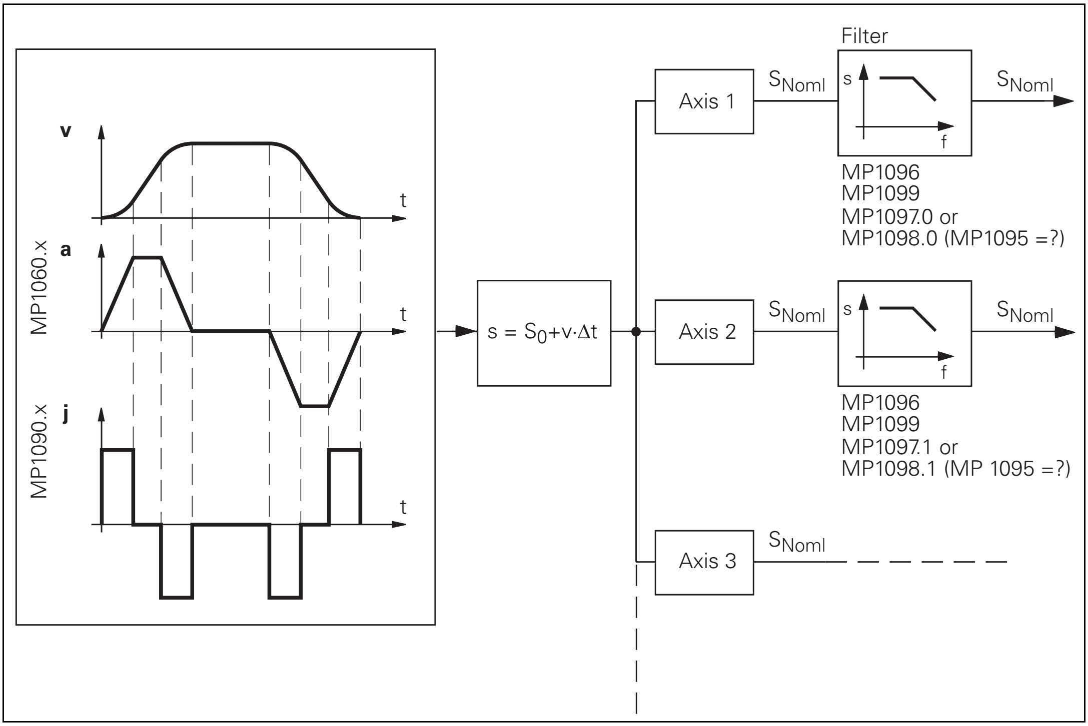
$$s_{min} = 2 \cdot v_{max} \cdot \sqrt{\frac{v_{max}}{r_{max}}} = 2 \cdot 0.5\,\frac{m}{s} \cdot \sqrt{\frac{0.5\,\frac{m}{s}}{70\,\frac{m}{s^3}}} = 0.085\text{ m} = 85\text{ mm}$$

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 The Interpolator

Schematic of the Interpolator:



The interpolator calculates a velocity every 3 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 smallest acceleration value applies.

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.
- Limit the axis-specific jerk with MP1097.x or MP1098.x. The machine-parameter block to be used depends on the value entered in MP1095.x. If MP1095.x = 0, MP1097.x is valid. If MP1095.x = 1, MP1098.x is valid.
- 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 3 ms from the calculated velocity. For linear interpolation:

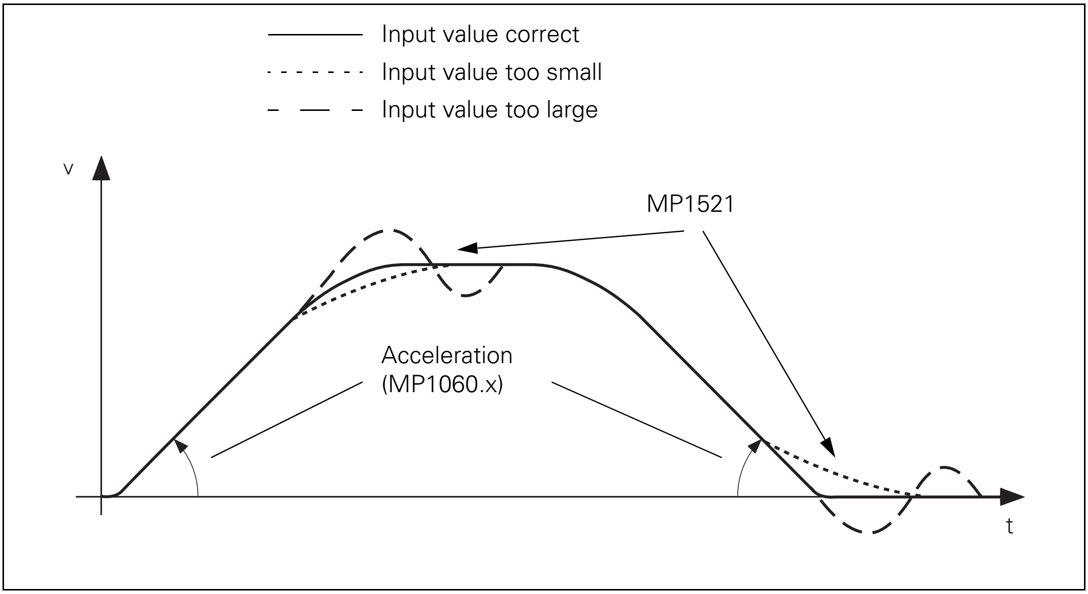
$$s = s_O + v \cdot \Delta t$$

s = nominal position value
s_O = previous nominal position value
v = calculated velocity
Δt = cycle time (3 ms)

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 TNC 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 TNC calculates the filter parameters from the permissible axis-specific jerk and the tolerance:

- 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), as of NC software 280 476-01 you can also enter for the HSC filter the permissible axis-specific jerk for curvature changes (e.g. tangential transitions from a line to an arc) in MP1097.x.
- 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.

Single filter (MP1099.0)

Damping [dB]	Frequency to be damped [Hz]										
	10	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**	**55**	**60**
3	10	–	–	–	3	2	2	–	–	–	1
4	12	7	5	4	–	–	–	2	–	–	–
5	13	8	6	–	–	3	–	–	2	–	–
6	14	9	–	5	4	–	–	–	–	2	–
7	15	10	7	–	–	–	3	–	–	–	–
8	16	–	–	6	–	4	–	3	–	–	2
9	17	11	8	–	5	–	–	–	–	–	–
10	18	–	–	–	–	–	–	–	–	–	–
11	19	12	–	–	–	–	4	–	3	–	–
12	–	–	9	7	–	–	–	–	–	–	–

Double filter (MP1099.1)

Damping [dB]	Frequency to be damped [Hz]										
	10	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**	**55**	**60**
3	7	4	3	2	–	–	1	1	–	–	–
4	8	5	–	–	2	–	–	–	1	–	–
5	9	6	4	3	–	2	–	–	–	1	–
6	10	–	–	–	–	–	–	–	–	–	1
7	11	7	5	–	3	–	2	–	–	–	–
8	–	–	–	4	–	–	–	–	–	–	–
9	12	8	–	–	–	–	–	2	–	–	–
10	13	–	6	–	–	3	–	–	–	–	–
11	–	–	–	–	–	–	–	–	2	–	–
12	14	9	–	5	4	–	–	–	–	–	–

HSC filter (MP1094)

Damping [dB]	Frequency to be damped [Hz]										
	10	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**	**55**	**60**
3	12	19	24	29	34	39	44	49	54	59	64
4	10	17	22	27	32	37	42	47	52	57	62
5	6	15	21	26	31	36	41	46	51	56	61
6	1	14	20	25	30	35	39	45	50	55	60
7	–	13	18	23	28	33	38	43	48	54	59
8	–	11	17	23	28	33	38	43	48	53	58
9	–	10	16	22	27	32	37	42	47	52	57
10	–	9	16	21	26	31	36	41	46	51	56
11	–	7	15	20	25	30	35	40	45	50	55
12	–	6	14	19	24	29	34	39	44	49	54

- 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 parameter	Single filter	Double filter	HSC filter
HSC filter	MP1094 = 0	MP1094 = 0	MP1094 = cutoff frequency
Single/double filter	MP1095.x = 0	MP1095.x = 1	MP1095.0 = nonfunctional MP1095.1 = 0 or 1
Tolerance for contour transitions	MP1096 = Tolerance (Cycle 32)		
Axis-specific jerk for single filter	MP1097.x = Jerk (at corners)	MP1097.x = nonfunctional	MP1097.x = Jerk (at curvature changes)
Axis-specific jerk for double filter	MP1098.x = nonfunctional	MP1098.x = Jerk (at corners)	MP1098.x = Jerk (at corners)
Minimum filter order	MP1099.0 = Filter order	MP1099.1 = Filter order	MP1099.x = nonfunctional

MP1060.0-8 Acceleration for axes 1 to 9
Input: 0.001 to 30.000 [m/s^2 or 1000°/s^2]

MP1087.0-8 Max. permissible axis-specific jerk for Manual mode
Input: 0.1 to 1000.0 [m/s^3 or 1000°/s^3]

MP1089.0-8 Max. permissible axis-specific jerk for Pass Over Reference Point mode
Input: 0.1 to 1000.0 [m/s^3 or 1000°/s^3]

MP1090 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

MP1097.0-8 Max. permissible axis-specific jerk (single/HSC filter)
Input: 0.1 to 1000.0 [m/s^3 or 1000°/s^3]

MP1098.0-8 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

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 = 5, the TNC has a minimum position controller cycle time of 3 ms. Particularly during applications that require intensive processing (e.g. M128) or several axes (5 to 9) the error message **PROCESSOR CHECK ERROR** may appear. In this case, increase the position control cycle time to 3.6 ms by entering the factor 6 in MP7600.0.
This increase simultaneously increases the PLC cycle time. To return to the previous PLC cycle time, enter the factor 6 in MP7600.1; the PLC cycle time is then 21.6 ms. For entries which lead to a PLC cycle time < 20 ms, the PLC cycle time is limited to 20 ms.

MP7600.0 Position controller cycle time = MP7600.0 · 0.6 ms
Input: 1 to 20
Proposed input value: 5 (= 3 ms)

MP7600.1 PLC cycle time = position controller cycle time · MP7600.1
Input: 1 to 20
Proposed input value: 7 (= 21 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 MP1390 or 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.

MP1390 Velocity feedforward in the POSITIONING WITH MANUAL DATA INPUT, PROGRAM RUN SINGLE BLOCK and PROGRAM RUN FULL SEQUENCE operating modes
As of software version:only before 280 474-07
Input: 0: Operation with velocity feedforward control
1: Operation with following error (lag)

MP1392 Velocity feedforward in the POSITIONING WITH MANUAL DATA INPUT, PROGRAM RUN SINGLE BLOCK and PROGRAM RUN FULL SEQUENCE operating modes
As of software version:280 474-07
Format: %xxxxxxxxx
Input: Bits 0 to 8 correspond to axes 1 to 9
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 = %000000000).

MP1391 **Velocity feedforward control in the MANUAL and HANDWHEEL operating modes**

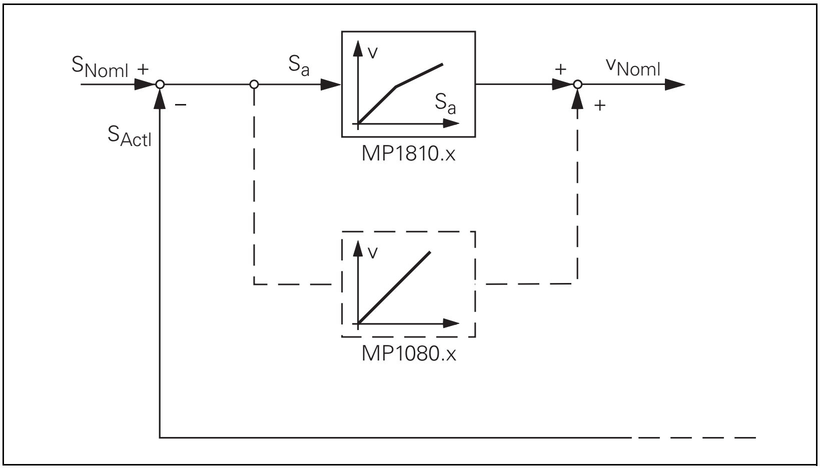
Format: %xxxxxxxxx

Input: Bits 0 to 8 correspond to axes 1 to 9
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:

$v = k_v \cdot s_a$

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. But 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]

MP1810.0-8	**k_V factor for operation with following error for axes 1 to 9**
Input:	0.100 to 20.000 [(m/min)/mm]
MP1815.0-8	**k_V factor for operation with following error effective after M105 for axes 1 to 9**
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.

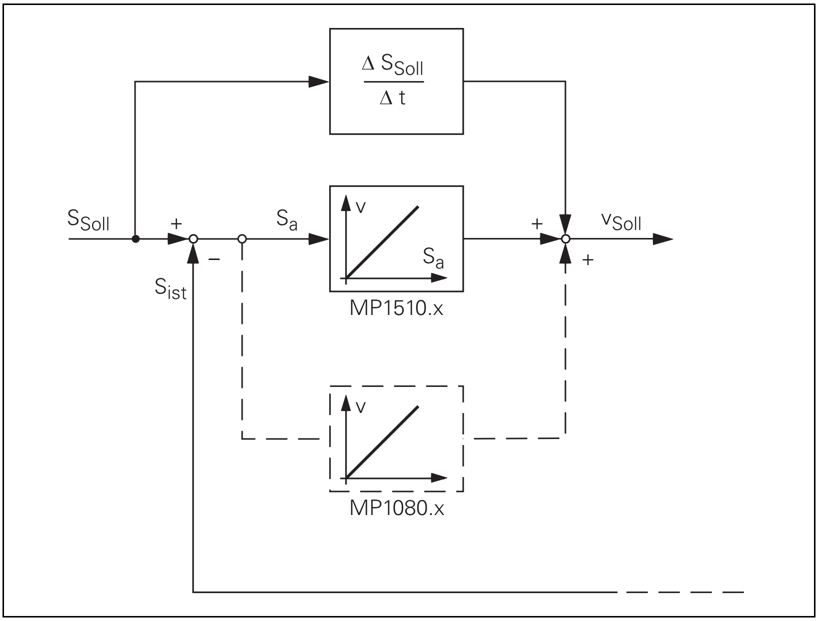
For the POSITIONING WITH MANUAL DATA INPUT; PROGRAM RUN, SINGLE BLOCK; and PROGRAM RUN, FULL SEQUENCE operating modes:

▶ Switch-on the velocity feedforward control with MP1390 or MP1392.

For the MANUAL and HANDWHEEL operating modes:

▶ Switch-on the velocity feedforward control with MP1391.

Block diagram:

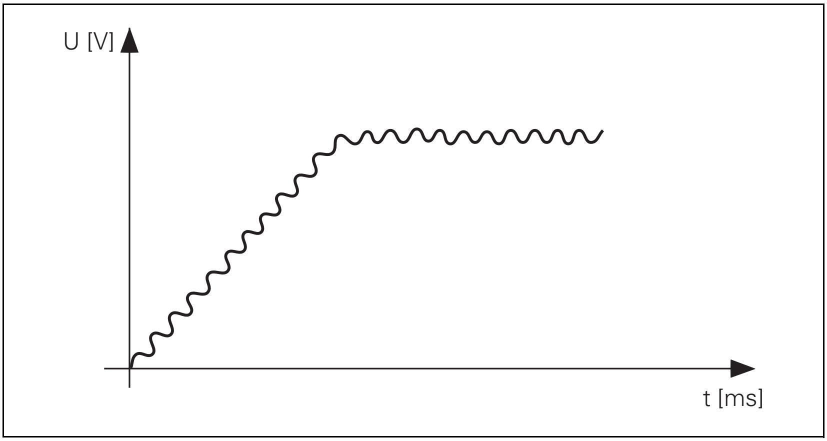


Analog axes:
For stationary axes, the integral factor has an additional effect (MP1080.x). It produces an offset adjustment.

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.

MP1510.0-8 **k_V factor for velocity feedforward for axes 1 to 9**
Input: 0.100 to 1000.000 [(m/min)/mm]

MP1515.0-8 **k_V factor for velocity feedforward effective after M105 for axes 1 to 9**
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. For example, velocity semifeedforward is activated 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.

MP1396.0-8 Feedback control with velocity semifeedforward for axes 1 to 9
Input: 0.001 to 0.999
1: Velocity feedforward control

MP1516.0-8 k_V factor for velocity semifeedforward for axes 1 to 9
Input: 0.100 to 20.000 [(m/min)/mm]

Rapid traverse

- In MP1010.x, define for each axis the rapid traverse of the machine.

You can reduce this value through the PLC:

- Enter the reduced value in D596.
  If the value in D596 is larger than MP1010.x, then MP1010.x 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 MP1010.x 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 F MAX soft key are effective again.

For manual operation the feed rate is significantly lower 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 rate of all axes is saved in D388 (as of NC software 280 476-01).

The maximum possible feed rate depends on the encoder being used.

v_{max} [mm/min] = P [mm] · f_i [kHz] · 60

v_{max} = Maximum traversing speed
P = Signal period of the encoder
f_i = Input frequency of the encoder input, (See “Encoders” on page 3 – 31) and (See "Encoder signals” on page 6 – 9).

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.

$$v_{max}[\text{mm/min}] = \frac{24\,000}{\text{No. of pole pairs}}[\text{1/min}] \cdot \text{Ball screw pitch [mm]}$$

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.

MP1010.0-8 Rapid traverse in axes 1 to 9
Input: 10 to 300 000 [mm/min]

MP1020.0-8 Manual feed rate for axes 1 to 9
Input: 10 to 300 000 [mm/min]

MP1050.0-8 Analog axes: Analog voltage for rapid traverse in axes 1 to 9
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.

$$\text{Pos. loop resolution [µm]} = \frac{\text{signal period [µm]}}{1024}$$

Analog axes

The TNC 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:

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

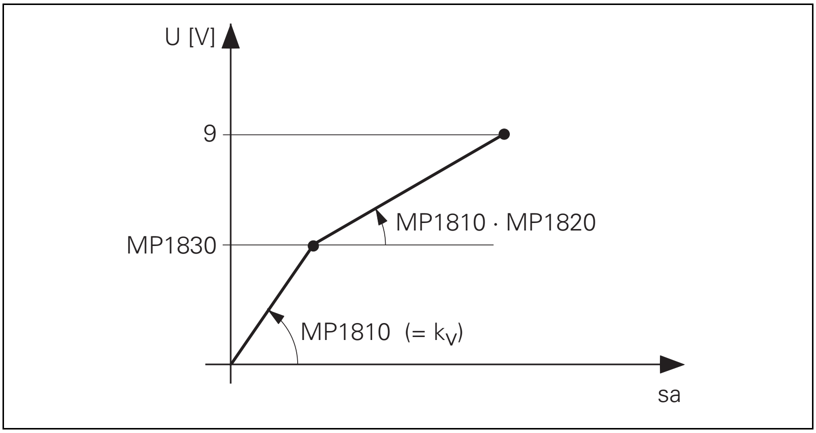
If ΔU is divided by the smallest possible voltage step (0.15 mV), the result is the number *n* of the possible voltage steps per position error.

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 following error:

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

MP1820.0-8 Multiplier for the k_v factor for axes 1 to 9
Input: 0.001 to 1.000 00

MP1830.0-8 Kink point for axes 1 to 9
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.

Warning

MP1150.1 = 0 switches position monitoring off! Safe machine operation is not possible if the position monitoring function is switched off. Uncontrolled machine movements will **not** be detected!

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, the **Approach position** function is automatically activated.

If M4580 has been set, an external EMERGENCY STOP (X42, pin 4 "control-is-ready signal acknowledgement") **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 correspond to 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 compatibility**

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 correspond to axes 1 to 9 0: Not active 1: Active	PLC	PLC
W1040	**Axis-specific opening of the position control loop** Bits 0 to 8 correspond to 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 axis, you must first enable the feed rate through the PLC. Until “feed-rate enable” is set, the nominal velocity value zero is output. In the status display, “F” is highlighted.

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 correspond to 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 correspond to 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.
For a transfer only of PLC axes, the module can always be called.
For a transfer via M strobe, MP7440 bit 2 must not be set. For a transfer via S/G strobe, MP3030 must not be set.

Call:
PS B/W/D/K <Axes bit-encoded>
CM 9145

Error recognition:

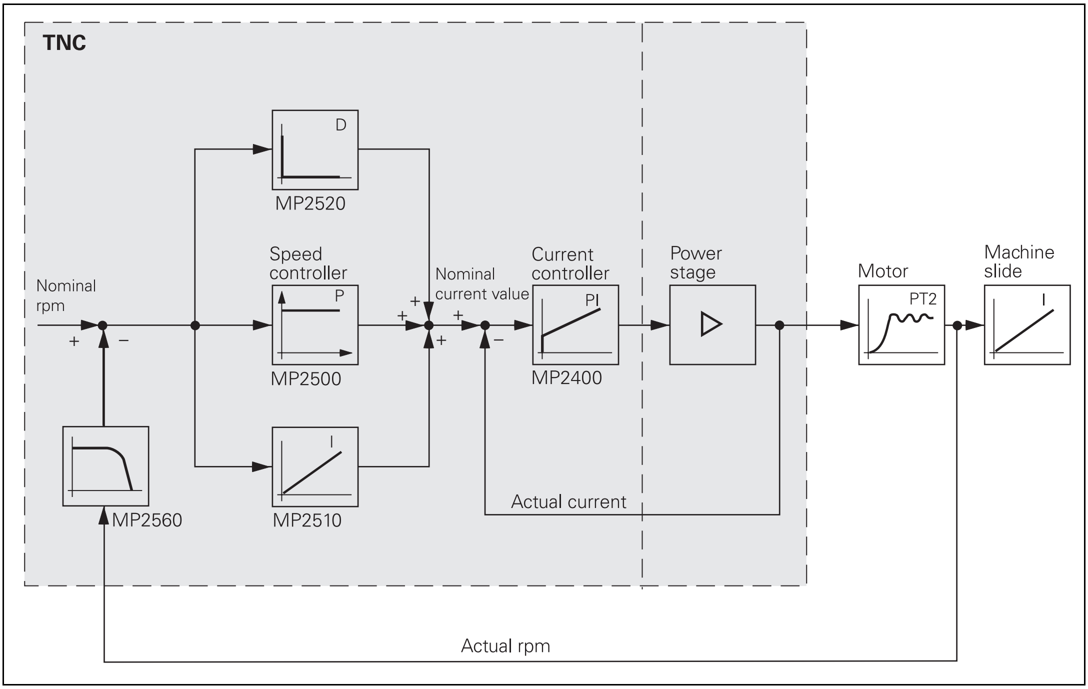
Marker	Value	Meaning
M4203	0	Actual value was assumed as nominal value
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing M/S/T/T2/G strobe in M4176 = 1
	24	Module was called in a spawn job or submit job

6.8.4 Speed Controller (Only TNC 426 PB/M, TNC 430 PA/M)

Digital speed controllers are integrated in the TNC 426 PB/M and TNC 430 PA/M:

- TNC 426 PB: digital speed encoder for 5 axes and 1 spindle
- TNC 430 PA: digital speed encoder for 6 axes and 1 spindle
- TNC 426 M: digital speed encoder for 5 axes and 1 spindle
- TNC 430 M: digital speed encoder for 6 or 9 axes and 1 spindle

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 – 462)

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

MP2500.0-8 Proportional factor of the speed controller for the axes 1 to 9

Input: 0 to 1 000 000.000 [As]

MP2510.0-8 Integral factor of the speed controller for axes 1 to 9

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:

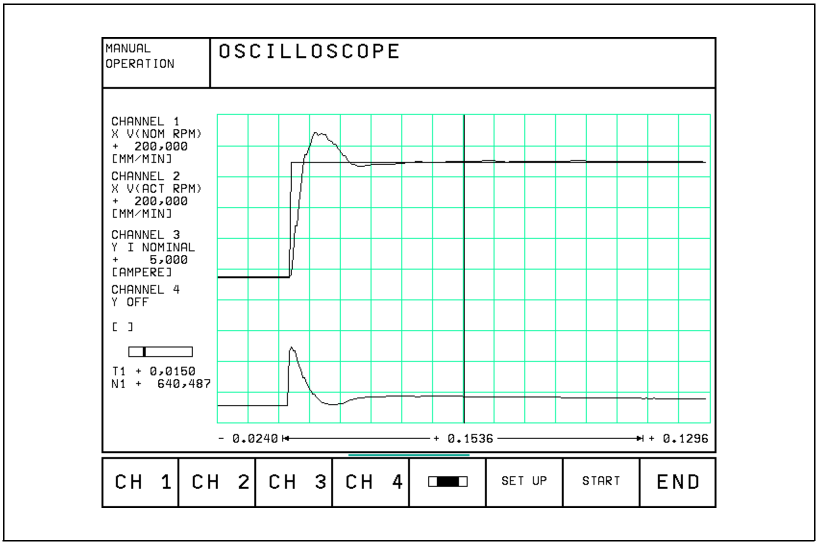
$$\text{Resolution} = \frac{1}{\text{Line count} \cdot 1024} \cdot 100\,000\ [\text{rpm}]$$

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 recognition:

Marker	Value	Meaning
M4203	0	Actual speed value was read
	1	Control has no integrated current controller



The step response illustrated above is idealized. In practice, interfering oscillations are superimposed on the step response.

You can reduce these interference oscillations with the differential factor, the PT_2 second-order time-delay element, the band rejection filter and the low-pass filter of the speed controller.

Differential factor

The differential factor can reduce low-frequency oscillations. However, it increases oscillations in the high frequency range.

- In MP2520.x, enter a differential factor.

Note

Ensure that the system is stable enough!

The differential factor is not recommended on machines with motors that have belt couplings. The influence of aging and temperature is too great.

Estimating the differential factor:

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

MP2520.x: Differential factor of the speed controller [As^2]
MP2500.x: Proportional factor of the speed controller
T: Period of the lowest interference frequency [s]

MP2520.0-8 Differential factor of the speed controller for axes 1 to 8
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 TNC.
- 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.)	1st order (MP2560.x = 1)
> 700 Hz (approx.)	2nd order (MP2560.x = 2)

If you cannot achieve satisfactory results with the low-pass filter, try the PT_2 element.

MP2560.0-8 Low-pass filter for axes 1 to 9
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

MP2530.0-8 PT_2 element of the speed controller (2nd-order delay) for axes 1 to 8
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 TNC, 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]

MP2540.0-8 Band-rejection filter damping for axes 1 to 8
Input: 0.0 to 18.0 [dB]

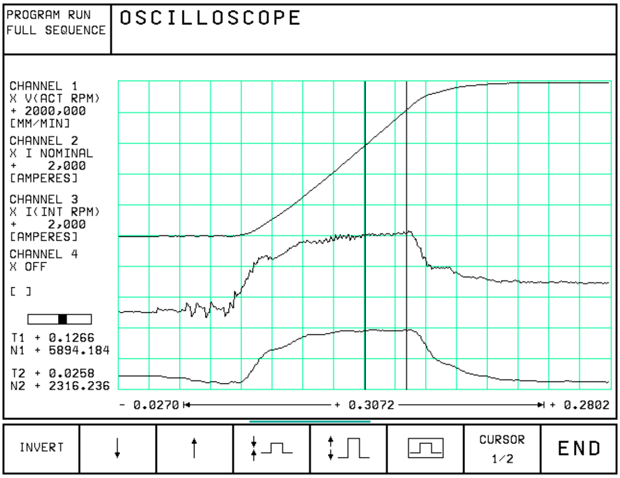
MP2550.0-8 Band-rejection filter center frequency for axes 1 to 8
Input: 0.0 to 999.9 [Hz]

Acceleration feedforward

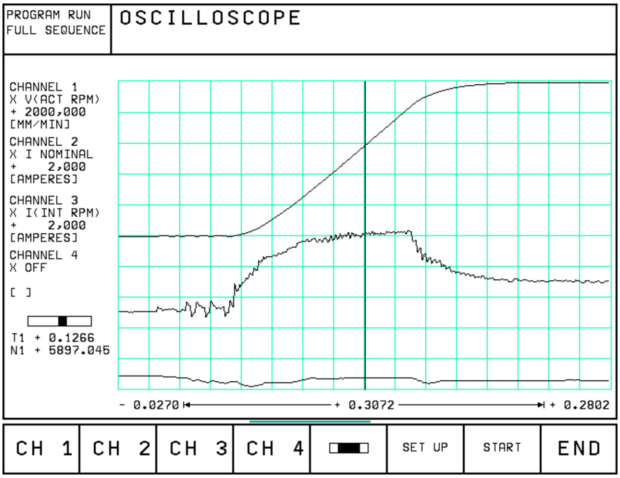
Acceleration feedforward functions only in velocity feedforward control in parallel with the speed controller.

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 (N INT) calculate the input value for MP2600.x.
- Adjust the acceleration feedforward control with MP2600.x.



MP2600 = 0



MP2600 = optimum

For calculation of the acceleration feedforward, the integral-action component of the nominal current value INTEG. RPM is recorded with the internal oscilloscope. The actual speed value V (ACT RPM) and nominal current value I NOMINAL are also recorded for better illustration.

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

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

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

ΔV (ACTUAL RPM) = actual speed value during change

MP2020.x = traverse distance per motor revolution

MP2600.0-8 Acceleration feedforward for axes 1 to 9
Input: 0 to 30.000 [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

MP2512.0-8 Limiting the integral-action component of the speed controller for axes 1 to 9
Input: 0.000 to 30.000 [s]

Integral Phase Compensation IPC

An I factor can be set in the speed controller of the TNC (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 (MP2600.x) feedforward 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.
- $\text{MP1510.x} = 0.65 \cdot \text{determined } k_V$

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.
- $\text{MP1510.x} = 0.65 \cdot \text{determined } k_V$

MP2602.0-8 IPC time constant T_1 for axes 1 to 9
Input: 0.0001 to 1.0000 [s]
0: IPC inactive

MP2604.0-8 IPC time constant T_2 for axes 1 to 9
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)

MP2606.0-8 Following error during the jerk phase for axes 1 to 9
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 TNC 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:

Note

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

$$\mathrm{MP2630} = \frac{\mathrm{I\ NOMINAL}_1 + \mathrm{I\ NOMINAL}_2}{2}$$

MP2630.0-8 Holding current for axes 1 to 9
Input: -30.000 to +30.000 [A]

Enabling the drive controller

TNC 426 PB/TNC 430 PA

- To enable the drive, assign 24 Vdc to connection X50 terminal 1.

If the ready signal is missing, or if there is no voltage on connection X50, the drive controller cannot be switched on.

If you disconnect the voltage for connection X50, all drive controllers are switched off. The drive controllers can be switched on with Module 9161 as soon as voltage is applied to X50. You can use, for example, the axis release W1024 as a criterion for drive enabling.

To enable the PLC to detect the disconnection of voltage to X50:

- Also apply the drive-releasing signal to a PLC input (see Basic Circuit Diagram at end of Chapter 3).
  To avoid contact problems, do not use the same relay contact as for drive enabling (X50/1).

When the drive controller is switched off, the axis is brought to a standstill. Then the speed controller and current controller are opened. Then the power supply unit is switched off with the reset signal.

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.

TNC 426 M/TNC 430 M

- To enable the drive, assign 24 Vdc to connection X42 pin 33 (I32).

If the ready signal is missing, or if there is no voltage on connection X42/33, the drive controller cannot be switched on.

If you disconnect the voltage for connection X42/33, all drive controllers are switched off. The drive controllers can be switched on with Module 9161, as soon as voltage is applied to X42/33. You can use, for example, the axis release W1024 as a criterion for drive enabling.

You can define axes for which the drives will not switch off if drive enabling (X42/33) is missing:

- 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.

Note

If drive enabling through X50 or X42/33 is missing, the error message `8B40 No drive release <axis>` appears.

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 TNC 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.

Note

From NC software 280 476-01 to 280 476-06, the waiting time is defined as 1.2 s (MP2170 is not available).
From NC software 280 476-07 to 280 476-08, the waiting time is defined as 2 s (MP2170 is not available).

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]

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: Reserved 3: Reserved
CM	9157	
PL	B/W/D	<Axis status information bit-encoded>

Error recognition:

Marker	Value	Meaning
M4203	0	Status information was ascertained
	1	Error code in W1022
W1022	2	Invalid status information was programmed
	24	Module was called in a spawn job or submit job

Module 9159 Drives that are switched off in 200 ms
This module functions only on the TNC 426 M/TNC 430 M controls.

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 recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Control has no current controller, or the call is in a spawn or submit job

Module 9162 Status request of the drive controller
Call:
CM 9162
PL B/W/D <Drive is ready>
Bit 15 0
Axis Sxxxxxx987654321
0: Not ready
1: Ready

Module 9169 Axes for which I32 does not switch off the drives
This module functions only on the TNC 426 M/TNC 430 M controls.

Call:
PS B/W/D/K <Axes bit-encoded>
CM 9169

6.8.5 Current Controller (Only TNC 426 PB/M, TNC 430 PA/M)

Analog current controllers are integrated in the TNC 426 PB, TNC 430 PA:

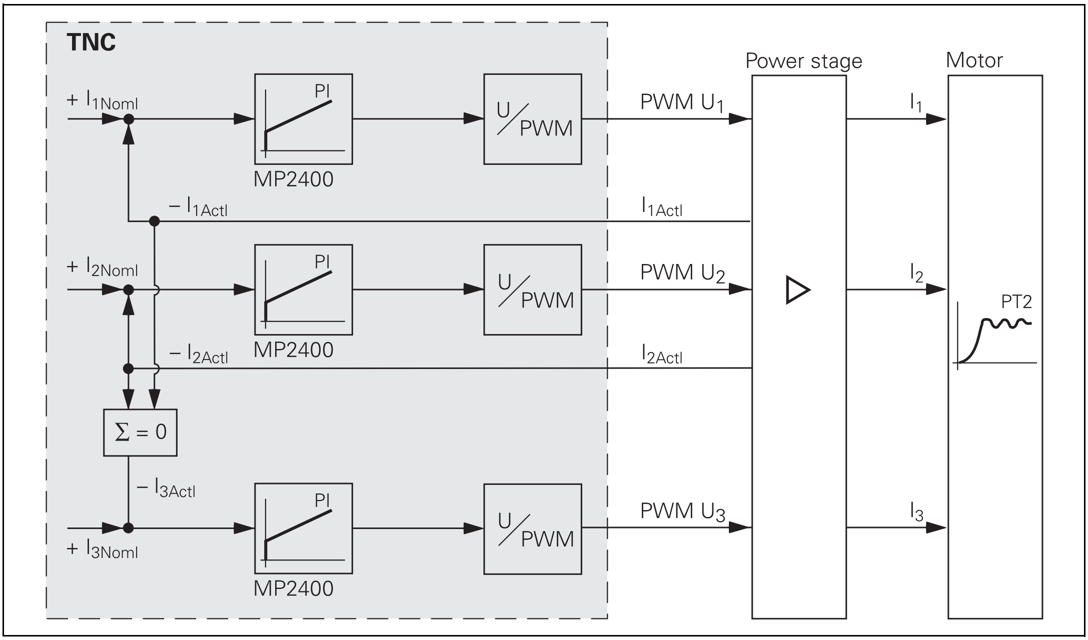
- TNC 426 PB: analog current controller for 5 axes and 1 spindle
- TNC 430 PA: analog current controller for 6 axes and 1 spindle
- TNC 426 M: analog current controller for 5 axes and 1 spindle
- TNC 430 M: analog current controller for 6 or 9 axes and 1 spindle

The phase currents I1 and I2 are servo controlled. The phase current I3 is calculated from I1 and I2. The sum of all phase current values is zero.

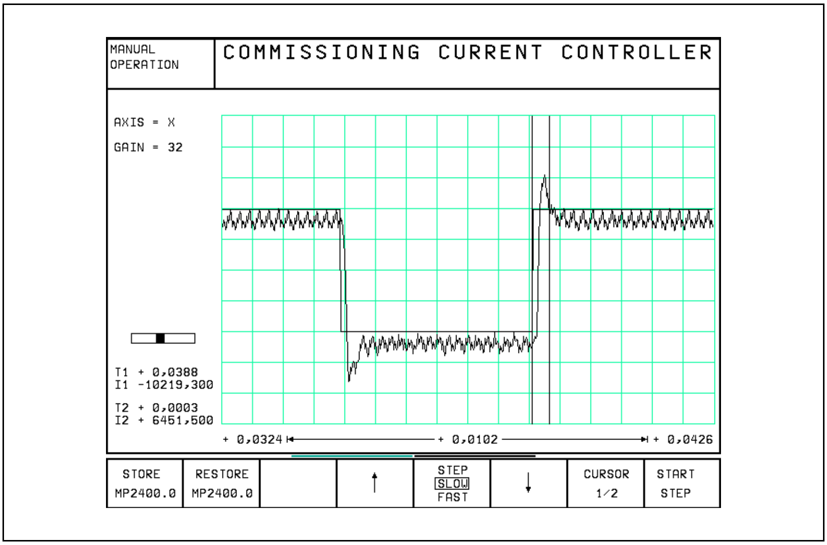
The actual current values for I1 and I2 are found from the motor power stage and are provided to X51 to X59 and X61 as voltage signals.

The phase currents I1, I2 and I3 are transferred as PWM signals to the motor power stage.

Circuit diagram:



You adjust the current controller to attain the optimum result, with the position and speed controller switched off.



The step response shows a light overshoot with a short rise time and settling time. The settling time t_{off} should be smaller than the cycle time of the speed controller (600 µs):

- In MP2400.x, define the current gain at standstill.

Some asynchronous motors run rough at maximum speed:

- In this case, enter in MP2402.x a current gain greater than that in MP2400.x. The current gain from MP2402.x is reached at maximum speed. It is interpolated linearly between standstill and maximum speed.
  If you enter the value zero in MP2402, the current gain from MP2400 is in effect for the entire speed range.

MP2400.0-8 Gain for current controller at standstill for axes 1 to 9
Input: 0.00 to 9 999.00 [V/A]
0: Controller disable

MP2402.0-8 Gain for current controller at maximum speed for axes 1 to 9
Input: 0.00 to 9 999.99 [V/A]
0: Value from MP2400.x

6.8.6 Braking the Spindle for an Emergency Stop

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.

Setting the axis brake ramp for an emergency stop:

- Enter as a minimum value in $MP2590.x = \frac{MP1060.x \cdot 60}{MP2020.x}$
- 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.

MP2590.0-8 Braking ramp for axes 1 to 8 or the second spindle in an emergency stop

Input: 0.1 to 999.9 [rpm/ms]
0: Function inactive

6.9 Offset Adjustment

Digital axes:

An offset adjustment at the output of the current controller is automatically compensated.

Analog axes:

An offset at the output of the current controller can be compensated in various ways:

- Offset adjustment by code number
- Automatic cyclic offset adjustment
- Offset adjustment with integral factor

Note

Automatic cyclic offset adjustment and offset adjustment by integral factor must not both be active at the same time!

The maximum permissible offset voltage in the control is 100 mV. If this voltage is exceeded, the error message **EXCESSIVE OFFSET IN <AXIS>** appears.

6.9.1 Offset Adjustment by Code Number

- Activate the offset adjustment with the code number 75 368.

The TNC displays the offset values of the analog axes in the dialog line. The values show the setting of the voltage in 0.15-mV steps.
Display value 10 means: 10 · 0.15 mV = 1.5 mV.
The displayed offset value consists of the offset values that are generated in the motor controller and in the control.

If the values are to be automatically compensated:

- Press the ENT key or the CONTINUE soft key. The control outputs a corresponding countervoltage.

If nothing is to be changed:

- Press the END soft key.

If the offset adjustment is to be switched off again:

- Enter the code number 75 368 and press the NO ENT key or the CANCEL soft key.

The offset values are saved in the control and remain safe in the event of power interruption. After a control is exchanged, the offset adjustment must be repeated by means of the code number.

6.9.2 Automatic Cyclic Offset Adjustment

The offset is adjusted automatically if the programmed time is expired and the following conditions are fulfilled:

- All axes are at standstill.
- The spindle is switched off.
- The axes are not clamped.

▶ In MP1220, program a time after which the offset is cyclically adjusted.

If the offset voltage is greater than 1 mV, the offset is compensated in each cycle by 1 mV.

If the offset voltage is less than 1 mV, the offset is compensated in voltage steps of 0.15 mV.

MP1220 Analog axes:
Automatic Cyclic Offset Adjustment

As of software version:only before 280 474-07

Input: 0 to 65 536 [s]
0: No automatic adjustment

6.9.3 Offset Adjustment with Integral Factor

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.

MP1080.0-8 Analog axes: Integral factor for offset adjustment for axes 1 to 9

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):

▶ Define the radial acceleration in MP1070.

MP1070 limits the feed rate during circular movement according to the formula:

$$v\ [m/s] = \sqrt{r\ [m] \cdot MP1070\ [m/s^2]}$$

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 30 000 [m/s^2 or $1000°/s^2$]

6.10.2 Contour Velocity at Corners

As of NC software 280 472-xx

To comply with a defined tolerance, the TNC can reduce the tool velocity before reaching machining corners, line-to-arc transitions and arc-to-arc transitions. The control can react to a potential violation velocity tolerance up to 128 blocks in advance. This feature is known as "look ahead":

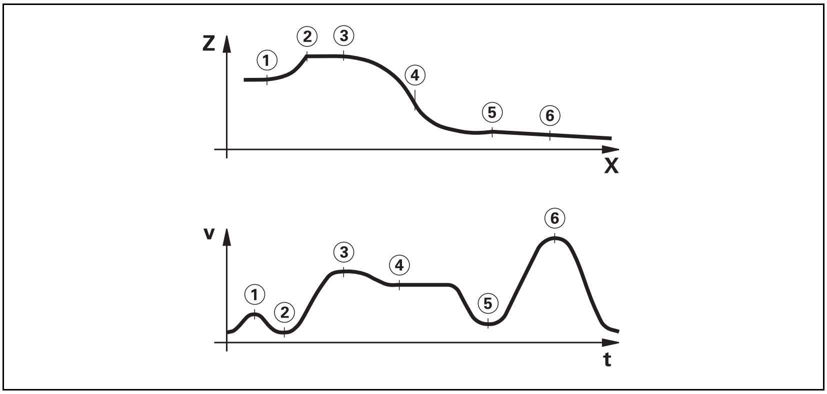
▶ Define the permissible tolerance for contour transitions in MP1096. The larger the tolerance, the greater the tool velocity.

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

Jerk limitation (See "The Interpolator" on page 6 – 113) and nominal-position-value filter enable the TNC 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

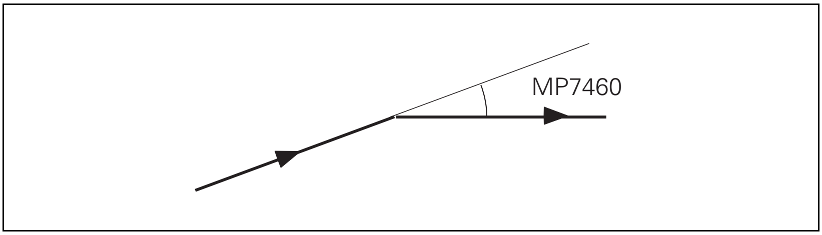
MP7680	**Machine parameter with multiple function**
Format:	%xxxxxxxxxxxxxx
Input:	Bit 10 – Cutter-radius-compensated outside corners: 0: Insertion of a circular arc 1: Insertion of a spline curve Proposed input value: %xx1xxxxxxxxxx Bit 11 – Reserved

Up to NC software 280 470-xx

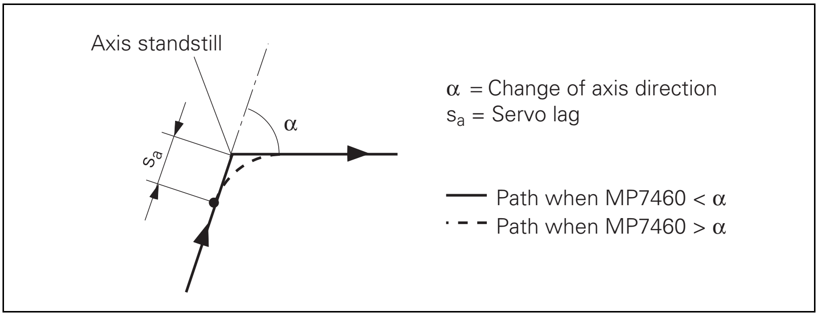
You can define an angle limit for traverse at constant velocity. The size of the angle depends on the machine drives:

- Enter the permissible angle in MP7460.
  Realistic input values: 5° to 15°

MP7460 functions without radius compensation at outside corners and with radius compensation at inside corners. The parameters apply for operation with both following error and feedforward control.



The contour is machined as shown here:



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.

- In MP7680 bit 7, specify whether the rounding arcs should always be inserted or only when 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 cubic spline produces an additional jerk reduction. But it takes more computing time than an inserted arc. As of 280 476-xx the feed rate for arcs and splines is reduced enough to prevent any excessive jerk. This does not apply if F MAX is programmed.

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.

If you program M132, you can reduce the jerk on the contour for changes of acceleration in individual axes. M133 switches M132 off.

MP7460	**Angle for constant contour speed at corners**
Input:	0.0001 to 179.9999 [°]

MP7680	**Machine parameter with multiple function**
Format:	%xxxxxxxxxxxxxx
Input:	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. 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

6.11 Monitoring Functions

The NC monitors the axis position 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 correspond to 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 (21 ms) for switching off the monitoring functions is not sufficient, you must use a high-speed PLC input. Fast PLC inputs are interrogated in the control loop cycle (3 ms).

- In MP4130.0, enter the number of the PLC input that must be sampled faster.

Note

The inputs of the PL 4xxB cannot be used as high-speed PLC inputs.

- Define in MP4131.0 the activation criterion for the input specified in MP4130.0.
- Enable MP4130.0 with W522 bit 0. As soon as the input is set, the monitoring functions and the drives are switched off. The axes are automatically stopped. 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] TNC 426 M, TNC 430 M: If you use I32, enter the following values: up to 280 474-11: MP4130.0 = 159 as of 280 474-12: MP4130.0 = 32 as of 280 476-01: MP4130.0 = 32 The inputs of the PL 4xxB 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 fast 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 TNC 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 all axes.
As of NC software 280 476-13, an actual-to-nominal value transfer is carried out only for the axis affected.

If the second limit is exceeded, the 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.

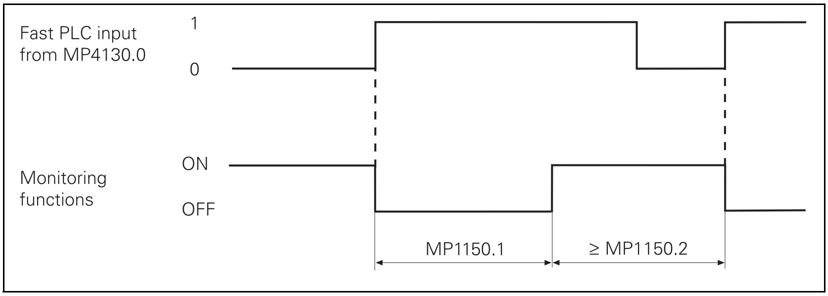
If blocked axes are the cause of the erasable error message **EXCESSIVE SERVO LAG IN <AXIS>,** a nominal velocity value may freeze, since the machine axes can no longer be moved:

- 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.

As of NC software 280 476-xx

The function of MP1150 has been expanded as of NC software 280 476-01:

- 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.
- As of NC software 280 476-01 the fast PLC input specified in MP4130.0 no longer automatically switches off the drive. 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).



MP1710.0-8 Position monitoring for operation with following error (erasable) for axes 1 to 9
Input: 0.0000 to 300.0000 [mm]
Recommended: 1.2 · following error

MP1720.0-8 Position monitoring for operation with following error (EMERGENCY STOP) for axes 1 to 9
Input: 0.0000 to 300.0000 [mm]
Recommended: 1.4 · following error

MP1410.0-8 Position monitoring for operation with velocity feedforward (erasable) for axes 1 to 9
Input: 0.0010 to 30.0000 [mm]
Recommended: 0.5 mm

MP1420.0-8 Position monitoring for operation with velocity feedforward (EMERGENCY STOP) for axes 1 to 9
Input: 0.0010 to 30.0000 [mm]
Recommended: 2 mm

MP1150	**Position monitoring**
As of software version:280 476-01	
MP1150.0	Delay time for erasing the nominal velocity value with the erasable 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] 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]

MP1150	**Delay time for erasing the nominal velocity value with the erasable error message EXCESSIVE SERVO LAG <AXIS>**
Input:	0 to 65.535 [s] Recommended: 0 s

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

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.

- Enter a minimum value in MP1140.x.

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 “Encoders” on page 6 – 8)).
- Enter the distance per motor revolution in MP2020.x. A formula can also be entered in MP2020.x (See “Special case: Entering a formula (As of NC software 280 472-01)” on page 4 – 5).
- In MP2800.x, enter a limit value for this position difference. If you are not using a position encoder, you must enter 0 in MP2800.x as the position difference.

If the difference is greater than the input value from MP2800.x, the error message **MOVEMENT MONITORING IN <AXIS> B** appears.

Warning

If you enter the maximum value in MP1140.x or MP2800.x, no movement monitoring is active.

Safe machine operation is not possible without the movement monitoring function.

MP1140.0-8 Threshold from which movement monitoring is effective for axes 1 to 9

Input:	Analog axes:	0.030 to 10.000 [V]
	Digital axes:	0.030 to 10.000 [1000 rpm]
		recommended: 0.030 [1000 rpm]

MP2020.0-8 Traverse per motor revolution for axes 1 to 9

Input: Analog axes: nonfunctional
Digital axes: 0 to 100.000 [mm] or [°]

MP2800.0-8 Movement monitoring for position and speed for axes 1 to 9

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.

MP1110.0-8 Standstill monitoring for axes 1 to 9
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.

MP1030.0-8 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.

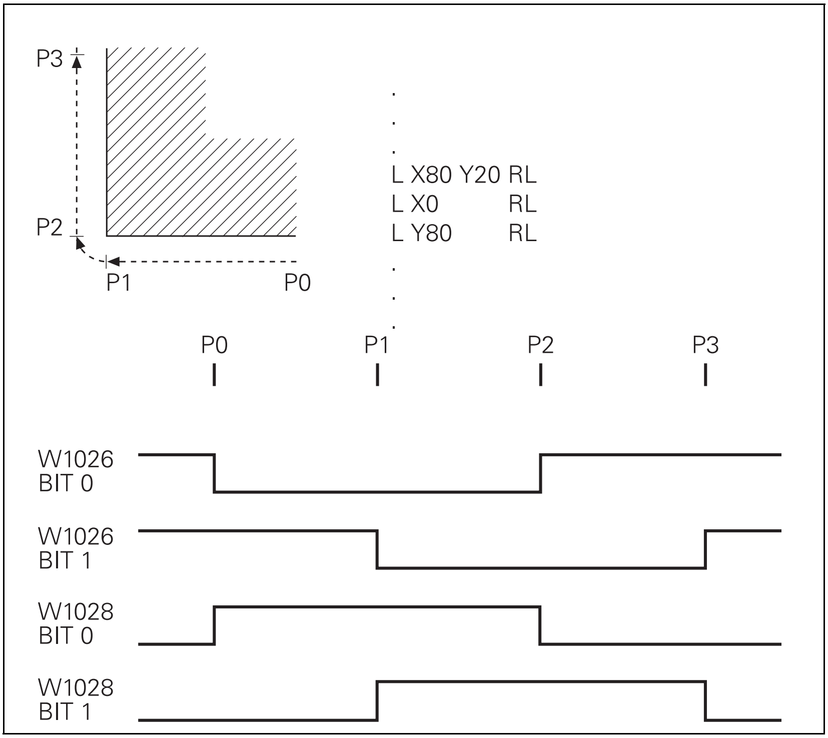
In the ELECTRONIC HANDWHEEL mode of operation the bit for the current handwheel axis is reset.

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

		Set	Reset
W1026	**Axes in position** Bits 0 to 8 correspond to 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 correspond to axes 1 to 9 0: Axis not in motion 1: Axis in motion	NC	NC



6.11.6 NC Supply Voltage Monitoring

The rectified voltage is monitored. Monitoring is worthwhile only with digital axes. The supply voltage must lie within the defined range (See “Supply Voltage for Control-is-Ready Signal (LE 426 M, LE 430 M)” on page 3 – 29). A transient overvoltage (approx. 5 s) up to 720 Vdc is permitted.

If 720 Vdc is exceeded, the NC revokes the pulse release (reset) for the IGBT of the power stage. The motors coast out of loop to a stop. No energy is returned to the dc link.

Below 385 Vdc, all drives are brought to a controlled stop. PLC outputs are switched off, and the control displays the error message **POWER FAIL**. The control must be turned off and on again.

Below 155 Vdc the control is reset.

Below 135 Vdc the dc-link power supply switches off.

Module 9167 Supply voltage monitoring
With this module you can switch the monitoring for supply voltage >385 Vdc 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: Supply voltage monitoring for >385 Vdc off
1: Supply voltage monitoring for >385 Vdc on
CM 9167
PL B/W/D <Error code>
0: Command executed
–1: Transferred parameter invalid

Error recognition:

Marker	Value	Meaning
M4203	0	Supply voltage monitoring on or off
	1	Error code in W1022
W1022	2	Transferred parameter invalid

6.11.7 Temperature Monitoring

Internal temperature of the logic unit

The internal temperature of the logic unit is monitored constantly. At approx. 70 °C, the error message **TNC OPERATING TEMP. EXCEEDED** appears.
The internal temperature of the LE can be read with module 9133.

Module 9133 Internal temperature of the LE

Call:

PS	B/W/D/K	<Code>
		0: Internal temperature of the LE
CM	9133	
PL	B/W/D	<Internal temperature of the LE>

Error recognition:

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

Motor temperature (only digital axes)

To measure the motor temperature, a KTY 84 must be connected at pins 13 and 25 of X15 to X20, X62 to X64 and at X60. 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 recognition:

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

Temperature of the power module's heat sink (only digital axes)

At X51 to X59 and X61 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 supply unit.

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 (Digital Axes Only)

The instantaneous motor current is limited to either the maximum current of the power supply unit, 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, I^2t is monitored:
For this purpose, the squares of the actual current values are integrated. For feed motors, the duration of integration is 10 s, for main spindle motors 150 s. The power module's rated current or the reference value for "I^2 t monitoring of the motor" is used as the I^2t limit value, whichever is smaller.

▶ Enter a reference value for I^2t monitoring for both

- MP2302.x for feed motors
- MP2303 for the spindle 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.

If the mean current value from the integral exceeds the I^2t limit value, the I^2t early warning signal responds.

I^2t limit

If the mean current value from the integral exceeds the 1.1-fold value of the I^2t limit value, the I^2t limit responds. Within 0.4 s the motor current is throttled to the 1.1-fold value of the I^2t limit.

If the calculated mean current value falls below the 1.1-fold of the mean current value, limiting is canceled.

Note

There is no I^2t limitation for spindle drives.

MP2302.0-8 Reference value for I^2t monitoring of feed motors for axes 1 to 9
Input: 0 to 1000.000 [· rated current of motor]
0: I^2t monitoring of feed motors switched off
1: Rated current of motor as reference value

MP2303 Reference value for I^2t monitoring of spindle motor
Input: 0 to 1000.000 [· rated current of motor]
0: I^2t monitoring of spindle motor switched off
1: Rated current of motor as reference value

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

Call:

CM	9160					
PL	D	<Temperature monitoring>				
		Bit	31	15	876543210	
		Axis:		Sxxxxx987654321		
PL	D	<I2t monitoring>				
		Bit	31	16	15	876543210
		Axis	xxxxxxx987654321		Sxxxxx987654321	
			I^2t limit		I^2t early warning	

Error recognition:

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

6.11.9 Monitoring of Power Supply Unit and DC-Link Voltage (digital axes only)

At X51 to X59 and X61 the ready signal is available at pin 6.

The signal is reset for the following reasons:

- The connection of K9 to K663 on the HEIDENHAIN interface PCB is not closed (with SIMODRIVE inverter system)
- Voltage from the unit bus (FR+) is missing (with SIMODRIVE inverter system)
- Error in the power supply unit (+5 V or U_z is missing or U_z is too large)

As soon as the readiness signal is reset, the drive controllers are switched off. Normally, the error message **MOVEMENT MONITORING IN <AXIS> A** is output through the position control loop.

Subsequently, the PWM signal release is switched off by the reset signal.

After the drive controller is switched on with Module 9161, you can use Module 9162 to interrogate the readiness of the drive controller ((See “Module 9162 Status request of the drive controller” on page 6 – 145)).

6.11.10 Current Utilization on the Drive Motors (Digital Axes Only)

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

Utilization means:

Speed range	n_{actl} < rated speed	$n_{actl} \geq$ rated speed
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} = <motor rated current>

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 or MP2313 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 recognition:

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

MP2312.0-8 **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

MP2313 **Reference value for utilization display of the spindle motor**
Input: 0 to 1000.000 [· rated current of motor]
0 or 1: Reference value is rated current of motor

Limiting the maximum torque

For **axes,** the torque is limited to the value taken from the list of either the power modules or the synchronous motors, whichever is lower.

With Module 9158 you can limit the torque. The torque-producing current required for the desired torque must be transferred to the module.
Formulas for calculating the torque-producing current:

Synchronous 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)

Asynchronous motor:
Two distinctions must be made with asynchronous motors:

- Anchored speed range (speed < threshold rpm for field weakening)

$$I_q = \frac{M \cdot n_N \cdot 2 \cdot \pi \cdot \sqrt{I_N^2 - I_0^2}}{P_N \cdot 60}$$

I_q: Torque-producing current
M: Desired torque
n_N: Rated speed (from motor table)
I_N: Rated current (from motor table)
I_0: No-load current (from motor table)
P_N: Rated power output (from motor table)

- Variable speed range (speed > threshold rpm for field weakening)

$$I_q = \frac{M \cdot n_N \cdot n \cdot 2 \cdot \pi \cdot \sqrt{I_N^2 - I_0^2}}{P_N \cdot n_{FS} \cdot 60}$$

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)

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.

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 recognition:

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

6.11.11 Status of HEIDENHAIN Inverters

All HEIDENHAIN inverters, except the UE 2xx compact inverters and non-HEIDENHAIN inverters, provide the error signal $\overline{\text{ERR-IZ}}$ in the event of an excessive dc-link current at X69.
For HEIDENHAIN inverters except the UE 2xx compact inverters:

- Activate the monitoring function for the $\overline{\text{ERR-IZ}}$ signal with MP2221 bit 2 = 1.

After the monitoring function has been activated and the signal has appeared,

- The torque-producing current for the spindle is limited to 70 % of the present current on the LE 426 M/12000 rpm

For HEIDENHAIN UE 2xx compact inverters and non-HEIDENHAIN inverters:

- Deactivate the monitoring function for the $\overline{\text{ERR-IZ}}$ signal with MP2221 bit 2 = 0.

After the monitoring function has been deactivated, the control continues working until the inverter switches off.

With module 9066, further status information on the HEIDENHAIN inverters can be read in the PLC.

MP2221 Current and speed controller monitoring functions
Input: Bit 2 –
0: Do not monitor $\overline{\text{ERR-IZ}}$ signal, or inverter does not supply this signal
1: Monitor $\overline{\text{ERR-IZ}}$ signal

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: No function
              Bit 1: dc-link voltage too high
              Bit 2: Heat sink temperature too high
              Bit 3: Short-circuit of a motor phase with UZ
              Bit 4: dc-link voltage too high
              Bit 5: Power supply unit not ready
              Bit 6: Leakage current too high
```

Error recognition:

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

6.11.12 EMERGENCY STOP Monitoring

On the control there are the PLC input I3 (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 TNC 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 acknowledgement" 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 after switching the machine control voltage back on.

- In MP1152, define the interrogation of the "control-is-ready signal acknowledgement" input. If MP1152 = 0, the input is interrogated directly by the NC (in the position controller cycle). If MP1152 = 1, the input is processed by the PLC (in the PLC cycle) before being passed on to the NC.

Resetting the "control-is-ready signal acknowledgement" 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.

Warning

MP1150.1 = 0 switches position monitoring off! Safe machine operation is not possible if the position monitoring function is switched off. Uncontrolled machine movements will **not** be detected!

If marker M4580 is set, then instead of the external emergency stop ("control-is-ready signal acknowledgement" input), the control loops of all axes and of the spindle are opened, and an NC stop is performed.

MP1152 **Interrogation of I3 "Control-is-ready signal acknowledgement"**
Input: 0: I3 is passed on directly to the NC
1: I3 is processed by the PLC before being passed on to the NC

		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.

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

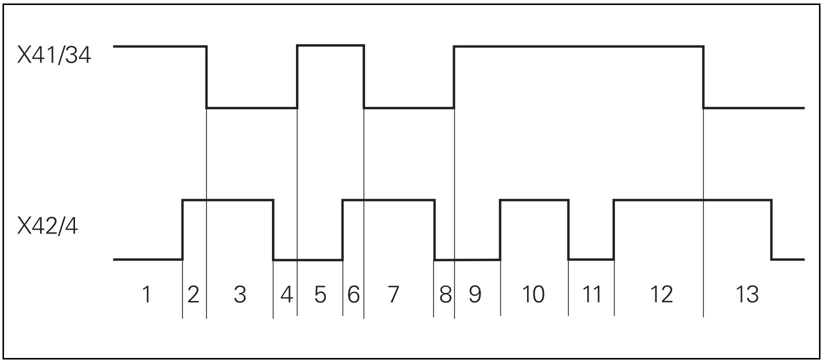
The circuitry recommended by HEIDENHAIN is illustrated in the Basic Circuit Diagram.

Ensure that the control-is-ready acknowledgment occurs within 380 ms.

Flowcharts

Flowchart for:

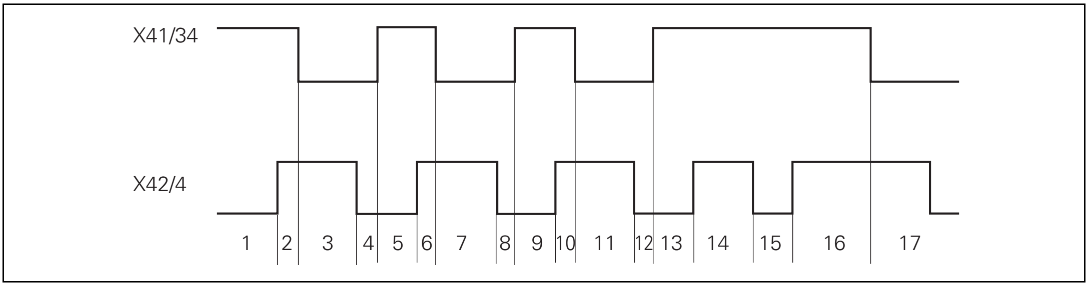
- TNC 426 CB/PB
- TNC 426 M/12 000 rpm (NC software < 280 476-06)



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 < 380 ms)	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	Recognition of the machine control voltage on X42/4 and switch-off of the control-is-ready signal on X41/34 by DSP (t < 120 ms)	
7	Maximum time within which the control-is-ready acknowledgment on X42/4 must go to zero (t < 380 ms)	If exceeded **EMERGENCY STOP DEFECTIVE**
8	Recognition of the acknowledgment and setting of X41/34 (t < 120 ms)	
9	Waiting for machine control voltage	**RELAY EXTERNAL DC VOLTAGE MISSING**
10	Normal control operation. Control-is-ready output and acknowledgment are high.	
11	Control voltage is switched off externally.	**EMERGENCY STOP**
12	After switching on again, the machine control voltage can be switched off, and then the control operates normally.	
13	After detecting a fault, the control switches off the control-is-ready output (X41/34).	Blinking error message

Flowchart for:

- TNC 430 CA/PA
- TNC 426 M/30 000 rpm (NC software < 280 476-06)
- TNC 430 M (NC software < 280 476-06)

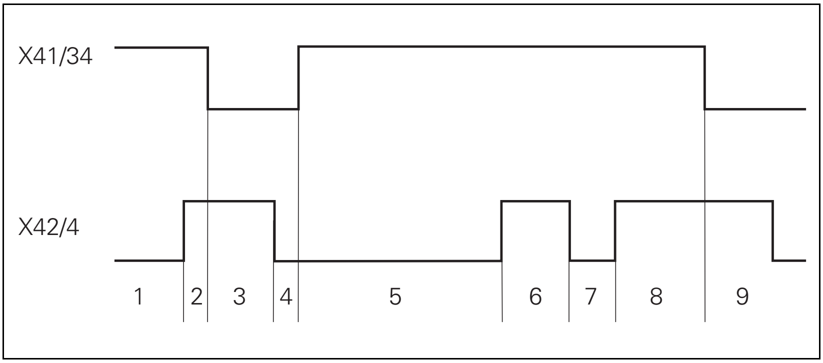


Step	Function	Screen display
1	Waiting for machine control voltage	**RELAY EXTERNAL DC VOLTAGE MISSING**
2	Recognition of the machine control voltage on X42/4 and switch-off of the control-is-ready signal on X41/34 by host computer (t < 66 ms)	
3	Maximum time within which the control-is-ready acknowledgment on X42/4 must go to zero (t < 380 ms)	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	Recognition of the machine control voltage on X42/4 and switch-off of the control-is-ready signal on X41/34 by DSP 1 (t < 120 ms)	
7	Maximum time within which the control-is-ready acknowledgment on X42/4 must go to zero (t < 380 ms)	If exceeded **EMERGENCY STOP DEFECTIVE**
8	Recognition of the acknowledgment and setting of X41/34 (t < 120 ms)	
9	Waiting for machine control voltage	**RELAY EXTERNAL DC VOLTAGE MISSING**
10	Recognition of the machine control voltage on X42/4 and switch-off of the control-is-ready signal on X41/34 by DSP 2 (t < 120 ms)	

Step	Function	Screen display
11	Maximum time within which the control-is-ready acknowledgment on X42/4 must go to zero (t < 380 ms)	If exceeded **EMERGENCY STOP DEFECTIVE**
12	Recognition of the acknowledgment and setting of X41/34 (t < 120 ms)	
13	Waiting for machine control voltage	**RELAY EXTERNAL DC VOLTAGE MISSING**
14	Normal control operation. Control-is-ready output and acknowledgment are high.	
15	Control voltage is switched off externally.	**EMERGENCY STOP**
16	After switching on the machine control voltage again, the error message can be cleared, and then the control operates normally.	
17	After detecting a fault, the control switches off the control-is-ready output (X41/34).	Blinking error message

Flowchart for:

- TNC 426 M (NC software > 280 476-06)
- TNC 430 M (NC software > 280 476-06)



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 < 380 ms)	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 "Operating a Second Spindle" on 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:
1st: Oriented spindle stop
2nd: Spindle jog
3rd: M3/M4
4th: 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 MP111.0.
  - If you have a digital spindle and would like to use the speed encoder also as a position encoder, then you must set MP111.0 = 0.

▶ Enter in MP3142 the line count of the rotary encoder to be used. 1-V_{PP} signals undergo 1024-fold subdivision.

▶ Enter in MP3142 the type of mounting of the position encoder 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 – 103).

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 the case the reference signal **must** be evaluated with Module 9220 ((See “Renewed traversing of the reference marks ” on page 6 – 103)).

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 9 999 [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 recognition:

Marker	Value	Meaning
M4203	0	Actual speed value was read
	1	Target address is too large or is not a double-word address

Module 9044 Reading the spindle coordinates (format 0.0001°)
Call:
SEE MODULE 9042.

6.12.2 Speed Encoder of the Spindle

Digital speed control requires a shaft speed encoder:

▶ Define the speed encoder input in MP113.x.

The TNC, with or without integral spindle DSP, 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 MP2221 bit 0 = 1.

The TNC, with or without integral spindle DSP, 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> not controllable`** appears.
TNC with integral spindle DSP:
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.

MP113	**Speed encoder for the spindle**
MP113.0	Speed encoder for the first spindle
Input:	0: No speed encoder input 15 to 20: Speed encoder inputs X15 to X20 60: Speed encoder input X60 (only on LE with integral spindle DSP) 62 to 64: Speed encoder inputs X62 to X64
MP113.1	Speed encoder for the second spindle
Input:	0: No speed encoder input 15 to 20: Speed encoder inputs X15 to X20 62 to 64: Speed encoder inputs X62 to X64

MP2221	**Current and speed controller monitoring functions**
Input:	%xx Bit 0 – Monitoring the reference mark 0: Monitoring active 1: Monitoring inactive Bit 1 – Monitoring the rotational direction (only with spindle DSP) 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:

TNC	Maximum spindle speed
TNC 426 CB, TNC 430 CA	100 000 rpm
TNC 426 PB, TNC 426 without spindle DSP (for motors with 2 pole pairs)	12 000 rpm
TNC 430 PA, TNC 426 M with spindle DSP, TNC 430 M (for motors with 2 pole pairs)	30 000 rpm

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.

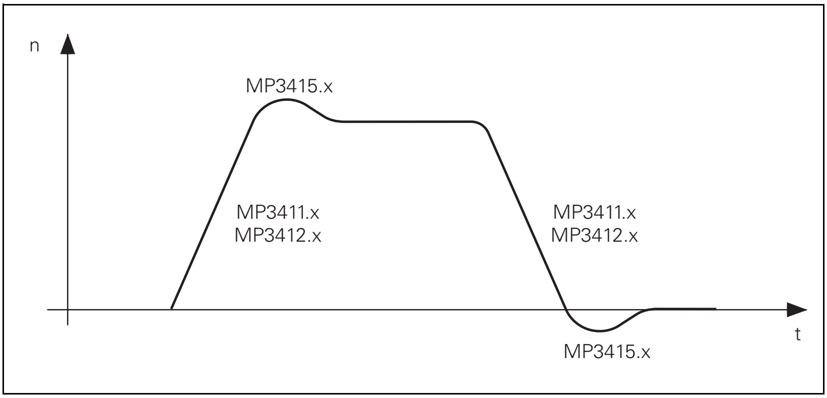
Digital spindles:
The nominal speed value is transferred to the internal speed controller.

		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 open-loop control

In the SPINDLE ORIENTATION mode of operation, the nominal speed value is controlled in a closed loop. In all other modes it is in an open loop.

The actual speed value of the spindle is not checked.



- 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 M03, M04 and M05 MP3411 so that the motor accelerates and brakes within the current limit.
- With MP3415, define the overshoot behavior for every operating mode. Set MP3415.0 so that only one overshoot is visible.

Note

As of NC software 280 476-05, MP3415.0 only influences the overshoot behavior when the spindle is switched on with M4011, but no longer with M4009 or M4010.

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 rev)/min · 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. (See "Position and Status Display” on page 6 – 231).

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:

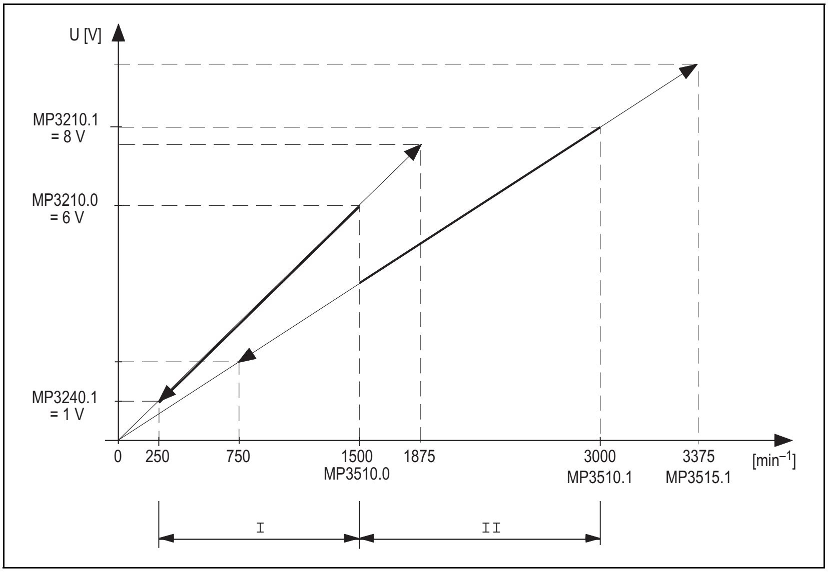
- With MP7620, bit 3, select the mode of the override.

Value range in W492 and W764:

- 1% steps: 1 to 150
- Nonlinear characteristic curve: In the lowest range, 0.01% steps are available. Beginning with a value of 2.5%, the step is 0.75%

Example: Two gear ranges for an analog spindle:

- Gear range I: 1500 rpm at 6 V (MP3210.0 = 6; MP3510.0 = 1500)
- Gear range II: 3000 rpm at 8 V (MP3210.1 = 8; 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 nominal value voltage: 1 V (MP3240.1 = 1)



MP3310	**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

Power limit of spindle

You can limit the power of your spindle motor to get wider gear ranges:

- In MP2393.x, enter the maximum power for wye and delta connection.

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.

The torque can be calculated for any speed:

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

M: Torque
P: Power
n: Speed

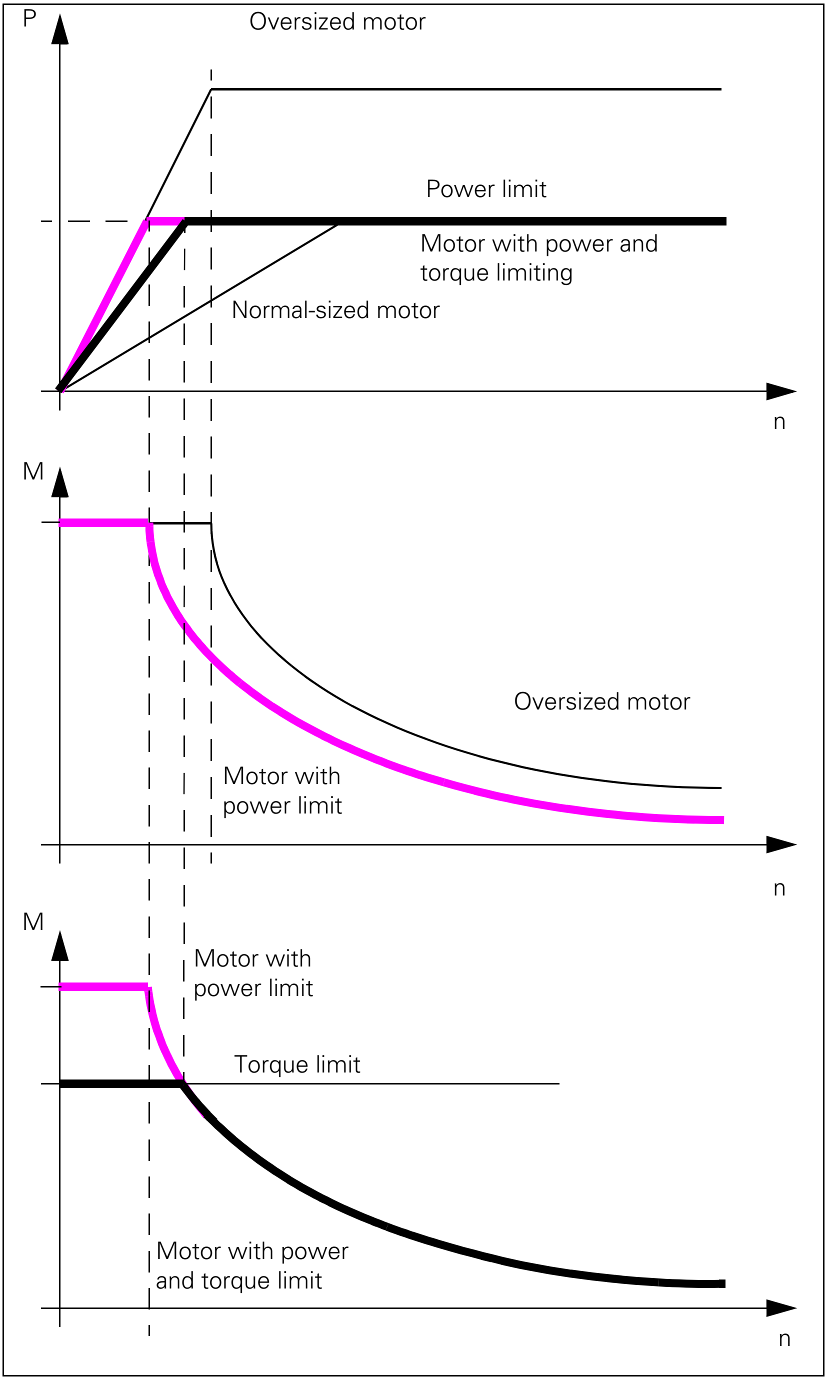
Note

MP2393.x can have an effect on the braking of the spindle in an emergency stop ((See “Spindle with integral DSP ” on page 6 – 199)).

- Enter the maximum power for the spindle in MP2393.x.
- Enter the maximum torque for the spindle in MP2397.x.

MP2393	**Power limiting of spindle motor**
Input:	0: No power limit 0.1 to 3000.000 [kW]
MP2393.0	Wye connection
MP2393.1	Delta connection

MP2397	**Maximum torque of the spindle motor**
Input:	0: No torque limiting 0.1 to 30 000 [Nm]
MP2397.0	Wye connection
MP2397.1	Delta connection


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

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 below). 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

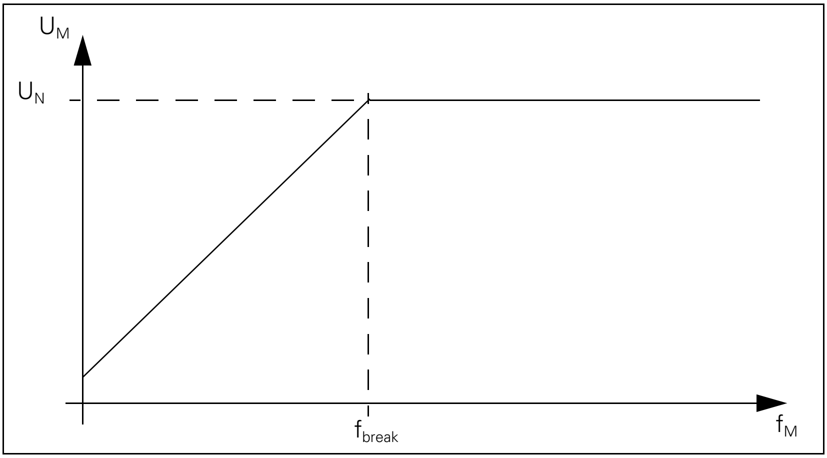
S code	rpm
S 00	0
S 01	0.112
S 02	0.125
S 03	0.14
S 04	0.16
S 05	0.18
S 06	0.2
S 07	0.224
S 08	0.25
S 09	0.28
S 10	0.315
S 11	0.355
S 12	0.4
S 13	0.45
S 14	0.5
S 15	0.56
S 16	0.63
S 17	0.71
S 18	0.8
S 19	0.9
S 20	1
S 21	1.12
S 22	1.25
S 23	1.4
S 24	1.6
S 25	1.8
S 26	2
S 27	2.24
S 28	2.5
S 29	2.8
S 30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



Maximum speed in U/f control mode:

- **with** an integral spindle DSP: $\frac{60000}{\text{No. pole pairs}}$ rpm
- **without** an integral spindle DSP: $\frac{24000}{\text{No. pole pairs}}$ rpm

Volts-per-hertz control mode is only possible on the TNC 426 M and TNC 430 M. The following PWM outputs may be used:

- TNC 426 M with spindle DSP, TNC 430: X61
- TNC 426 M without spindle DSP: X55, X56

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) and speed controller (MP25xx, MP 26xx) 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

- If the maximum current is exceeded, the inverter switches off and the spindle coasts to a stop.
- 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 Braking the Spindle for an Emergency Stop

For an emergency stop 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.
All of the following braking strategies are also effective when braking the spindle with M05, if the brake ramp in M05 is steeper than the brake ramp in the emergency-stop braking strategy.

Spindle without integral DSP

Controls without integral spindle DSP offer you three strategies for braking the spindle in an emergency stop. If you use a second digital spindle, all three strategies are available for it:

- Strategy 1: The maximum braking current is monitored and, if required, limited.
- Strategy 2: The regenerated power is influenced by a time constant.
- Strategy 3: The brake ramp can be entered.

The nominal velocity 0 is output for strategies 1 and 2.

The strategies are also effective during a power fail and with M05 if the brake ramp set in MP3411 and MP3412 is steeper than the brake ramp which would result from the strategies.

Note

The strategies are not mutually exclusive, meaning all strategies may be activated. In an emergency stop the first addressed strategy becomes effective.

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. Strategy 3 is recommended for both inverter systems.

Strategy 3:

- Enter MP236x = 0 and MP2191 = 1.
- Enter a high value in MP259x.
- Use the emergency stop to brake the spindle from the maximum speed.
- Decrease the value entered in MP259x until the braking time is as short as possible and the mechanics of the machine are not stressed too much.

Strategy 1:

- Enter MP236x = 0 and MP259x = 0.
- Enter MP2191 = 1.

Strategy 2:

- Enter MP2191 = 0 and MP259x = 0.
- In MP236x, enter a high value, e.g. 4.
- Use the emergency stop to brake the spindle from the maximum speed.

▶ Decrease the value entered in MP236x until the braking time is as short as possible and the mechanics of the machine are not stressed too much.

Note

Since the mass of the tool and the temperature of the braking resistor affect the braking power, MP236x should be determined with the heaviest tool and a "hot" braking resistor. To heat up the braking resistor, accelerate and decelerate the spindle several times at a quick pace.

MP2191 **Braking the first spindle in an emergency stop with monitoring of the maximum braking current**
Input: 0: Braking with monitoring of the maximum braking current
1: Braking without monitoring of the maximum braking current

MP2360.0-8 **Time constant for braking axes 1 to 8 or the second spindle in an emergency stop**
Input: 0.01 to 5.00 [s]
0: Function inactive

MP2361 **Time constant for braking the first spindle in an emergency stop**
Input: 0.01 to 5.00 [s]
0: Function inactive

MP2590.0-8 **Braking ramp for axes 1 to 8 or the second spindle in an emergency stop**
Input: 0.1 to 999.9 [rpm/ms]
0: Function inactive

MP2591 **Braking ramp for the first spindle in an emergency stop**
Input: 0.1 to 999.9 [rpm/ms]
0: Function inactive

Spindle with integral DSP

For a spindle with integral DSP, the maximum braking performance during spindle braking in an emergency stop can be entered in MP2391.x, and the maximum braking performance during a power fail can be entered in MP2395.x.

If the power limit (MP2393.x) is used in normal spindle operation, then the maximum braking performance is limited to the lower of the two values in MP2393 and MP2391.
Example:

Function	Case 1	Case 2
Power limit MP2393.x	10 kW	5 kW
Maximum braking performance MP2391	5 kW	10 kW
Limiting the braking performance to	5 kW (from MP2391)	5 kW (from MP2393)

Warning

After an emergency stop the spindle DSP automatically switches off after a fixed period of time. If the spindle needs more time, the drive will switch off and the spindle will coast to a stop.

- Up to 280 476-02: 5 seconds
- 280 476-03 and later: 10 seconds

Braking upon emergency stop

- For **inverters with regenerative power supply,** enter MP2391.x = 0 so as not to limit the braking power.
- Calculate for **inverters with braking resistors** the input value for MP2391.x from the following formula:

$$\text{MP2391.x} = \frac{U_Z^2}{R \cdot 1000}$$

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, UV 150, UR 2xx = 650 V))

Braking during power fail

For the LE 426 M with spindle DSP and the LE 430 M, during a power fail the "SH1B" signal at X6 is maintained for 3 seconds to allow the spindle to decelerate. 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 MP2395.x with the above formula.
- Calculate for **inverters with braking resistors** the input value for MP2395.x with the above formula.

Note

If after entry of a value in MP2391.x or MP2395.x the mechanics are overloaded by the braking process, lower the value in MP2391.x or MP2395.x until you have found an optimum between braking time and mechanical loading.

MP2391	**Maximum power for braking the first spindle in an emergency stop**
Input:	0.1 to 3000.000 [kW] 0: Braking power is not limited
MP2391.0	Wye connection
MP2391.1	Delta connection
MP2395	**Maximum power for braking the first spindle in a power failure**
Input:	0.1 to 3000.000 [kW] 0: Braking power is not limited
MP2395.0	Wye connection
MP2395.1	Delta connection

6.12.7 Oriented Spindle Stop

For spindle orientation the spindle must be in a closed control loop:

- Mount a position encoder for the spindle.
- With MP3010 (input value 6 to 8), specify whether the control provides for spindle orientation.

In the NC's touch probe cycles and rigid tapping cycle, the NC orients the spindle directly. In these cases, the NC sets M4017. You must reset M4012 in the PLC.

To orient the spindle to a specific angle in an NC program, use FN17: SYSWRITE ID990 NR8. The conditions above must be followed. The NC program resumes after the spindle is in position (M4000).
You can ascertain the current spindle angle with FN18: SYSWRITE ID990 NR8.

If the spindle orientation is started with an M function (e.g. M19), you must activate the oriented spindle stop in the PLC.
In MP7442, enter the number of the M function (e.g., 19) which will trigger the oriented spindle stop during the machining cycles. If MP7442 = 0 (no oriented spindle stop), the error message **`ORIENTATION not permitted`** appears when a cycle which uses oriented spindle stop is called.

The spindle orientation runs asynchronously to the NC positioning commands. You may only acknowledge the orientation once the spindle is in position (M4000).

The NC starts orienting the spindle only if the drive is switched on with Module 9161.

There are three ways to orient the spindle in the PLC:

- Module 9171
- Marker M4130
- Via initiator with marker M4011

MP7442	**Number of the M function for spindle orientation in the cycles**
Input:	1 to 999: Number of the M function 0: No oriented spindle stop –1: Oriented spindle stop by the NC

Process of spindle orientation with Marker M4130 or 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 free again.

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 the positioning movement lasts.

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.

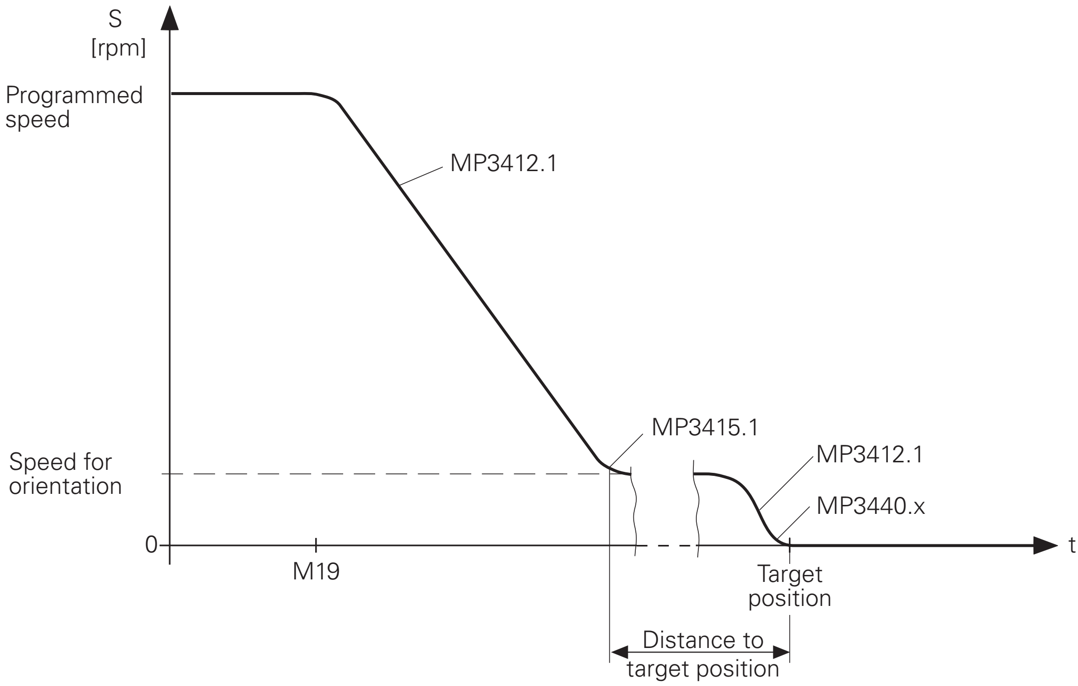
Module 9171 Oriented spindle stop
Can only be called from the sequential program.

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
```

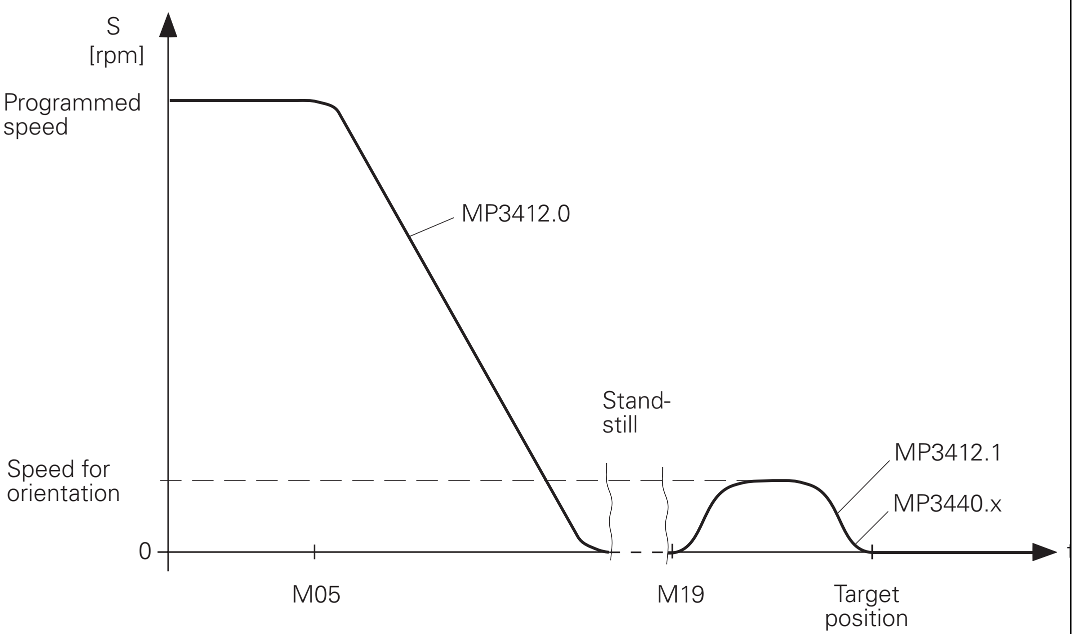
Error recognition:

Marker	Value	Meaning
M4203	0	Spindle is brought to an oriented stop
	1	Error code in W1022
W1022	1	Incorrect value for direction of rotation or rotational angle
	2	Incorrect speed
	19	No feedback-controlled spindle
	24	Call was in a submit or spawn 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.

6.12.8 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] = 1.8 [V]
MP3411.x = 0.05 [V/ms]

$$\frac{1.8\ [V]}{0.05\ [V/ms]} = 36\ ms$$

In this example the spindle was braked 36 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 also diagram below.)

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:

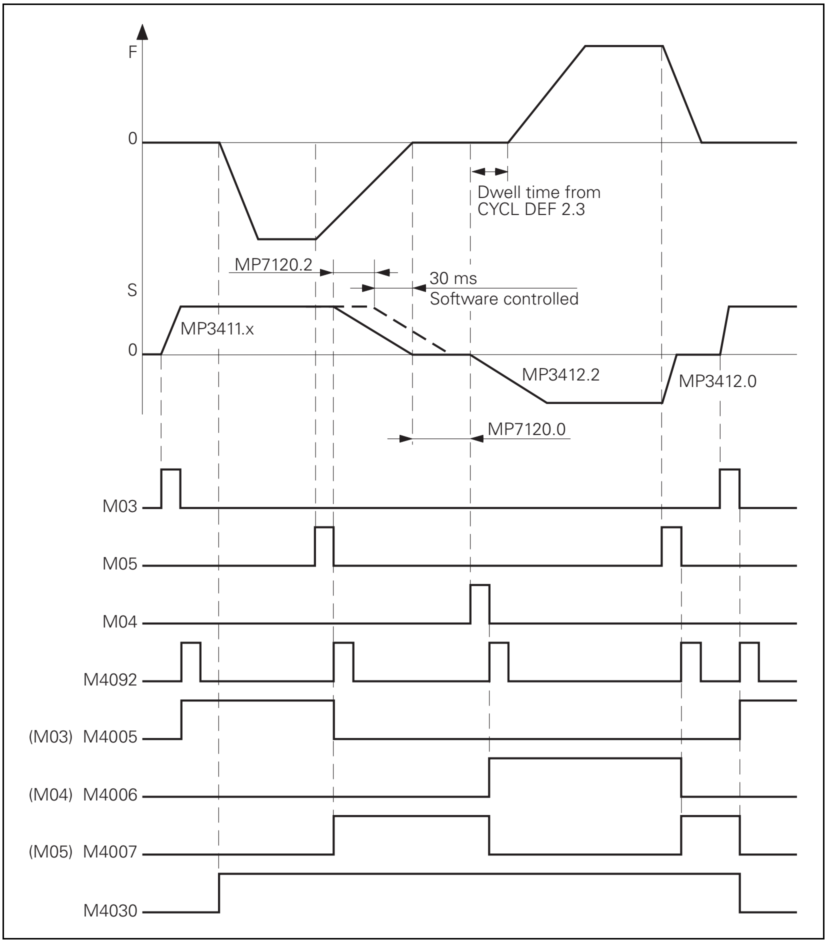
- Change the programmed dwell time in the cycle.

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 [%]

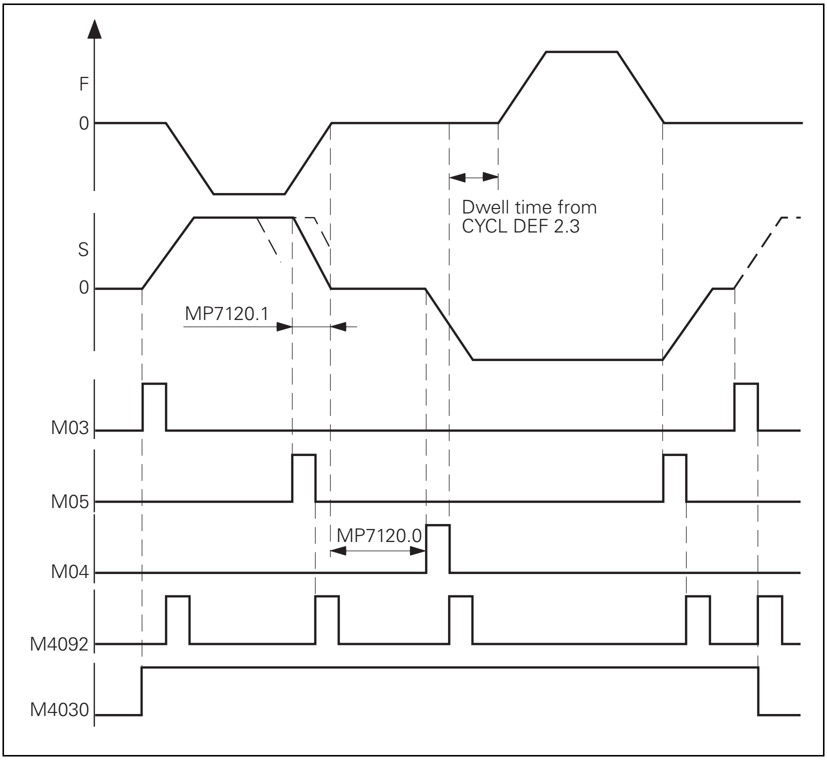
MP7110.1 **Maximum for feed rate override during tapping**
Input: 0 to 150 [%]

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]

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

The following diagram shows the time sequence of the cycle:



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.

MP7120.1	**Advanced switching time of the spindle during tapping with coded spindle-speed output**
Input:	0 to 65.535 [s]

6.12.10 Rigid Tapping

Cycle 17

- Define the rigid tapping process in the NC program with Cycle 17.
  While Cycle 17 is running, the TNC 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 – 109); (See "Spindle” on page 6 – 180).

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 IPC and 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:

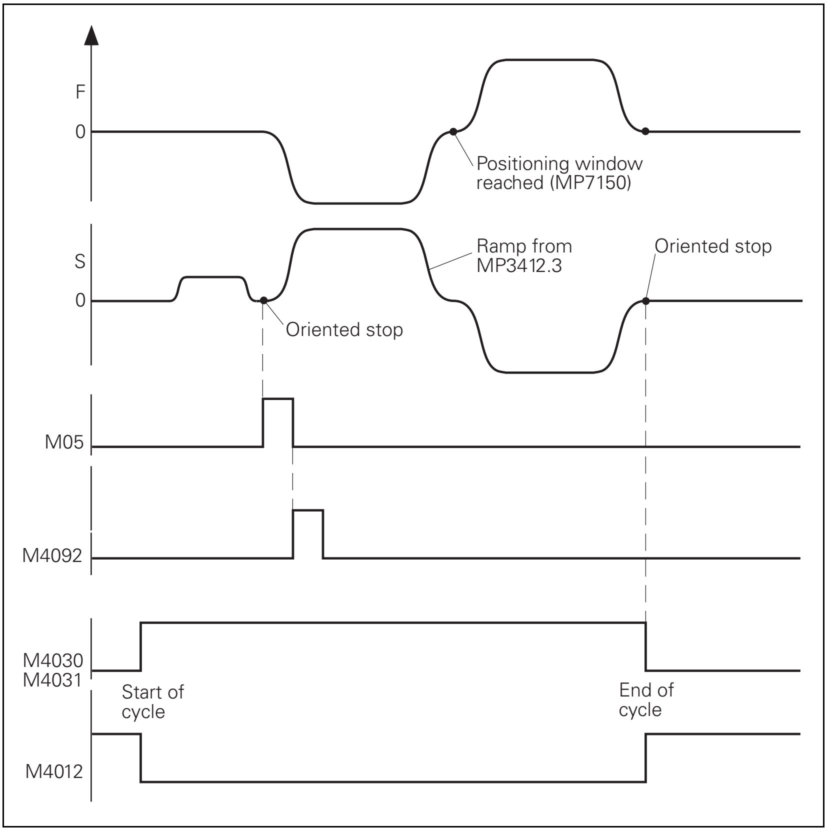
- In MP7160, set bit 3 = 1 to switch off IPC and acceleration feedforward control for Cycle 17.

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 "Oriented Spindle Stop” on page 6 – 201)

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.

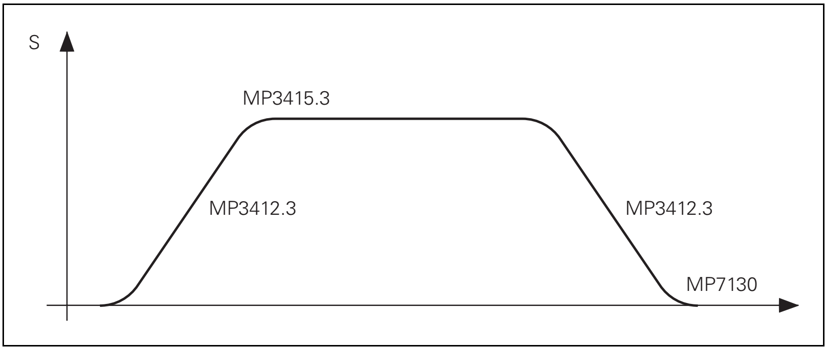


While Cycle 17 is running, the positioning window from MP7150 applies to the tool axis:

▶ 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 and 18**
Format: %xxxx
Input: Bit 0 – Oriented spindle stop with Cycle 17
0: Spindle orientation before execution of Cycle 17
1: No spindle orientation before execution of Cycle 17
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 – IPC and acceleration feedforward control
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.

6.12.11 Speed Controller (Only TNC 426 PB/M, TNC 430 PA/M)

A digital speed controller for the spindle is integrated in the TNC 426 PB/M and TNC 430 PA/M:

- With MP2501.x adjust the proportional factor, and with MP2511.x the integral factor of the speed controller of the first spindle.

With Module 9164 you can read the actual speed value of the motor. (See "The Control Loop" on page 6 – 109).

For the speed controller of the spindle, you can define the differential factor, the low-pass filter, PT_2 second-order time-delay element, and the band-rejection filter (See "Commissioning" on page 6 – 462).

- In MP2521.0–1, enter the differential factor.
- If required, enter the filter order in MP2561.
- If required, in MP2531.0–1 enter the PT_2 second-order time delay element.
- If required, enter in MP2541 and MP2551 the band-rejection filter damping and the center frequency.

MP2501.0-1 Proportional factor of the spindle speed controller for wye and delta connection
Input: 0 to 100 000 000.000 [As]

MP2511.0-1 Integral-action factor of the spindle speed controller for wye and delta connection
Input: 0 to 100 000 000 [A]

MP2521.0-1 Differential factor of the spindle speed controller for wye and delta connection
Input: 0 to 1.0000 [As^2]

MP2531.0-1 PT_2 second-order time delay element of the speed controller for the first spindle for wye and delta connection
Input: 0 to 1.0000 [s] 0 = 0.001 s

MP2541 Band-rejection filter damping
Input: 0.0 to 18.0 [dB]

MP2551 Band-rejection filter for center frequency
Input: 0.0 to 999.9 [Hz]

MP2561 Low-pass filter
Input: 0: No low-pass filter
1: 1st-order low-pass filter
2: 2nd-order low-pass filter

6.12.12 Current Controller (Only TNC 426 PB/M, TNC 430 PA/M)

The TNC has various current controllers, depending on the hardware version. There are logic units **with and without** spindle DSP:

- TNC 426 PB/M max. spindle speed 12 000 rpm, without spindle DSP
- TNC 426 PB/M max. spindle speed 30 000 rpm, with spindle DSP
- TNC 430 PA/M max. spindle speed 30 000 rpm, with spindle DSP

Current controller without DSP

▶ In MP2401, define the current gain for the spindle at standstill.

Some asynchronous motors run rough at relatively high speeds:

▶ In this case, enter in MP2402.x a current gain greater than that in MP2400.x. The current gain from MP2403 is reached at maximum speed. The current gain is interpolated linearly between standstill and maximum speed.

- If you enter the value zero in MP2403, then the current gain from MP2401 is effective for the entire speed range.

MP2401 **Gain for the spindle current controller at standstill**
Input: 0.00 to 9999.99 [V/A]
0: Controller disable

MP2403 **Gain for the spindle current controller at maximum speed**
Input: 0.00 to 9999.99 [V/A]
0: Value from MP2401

Current controller with DSP

▶ With MP2421.x adjust the proportional factor, and with MP2431.x the integral factor of the current controller of the first spindle. (See "Commissioning" on page 6 – 462).

MP2421.0-1 **Proportional factor of the spindle current controller for wye and delta connection**
Input: 0.00 to 9999.99 [VA]

MP2431.0-1 **Integral factor of the spindle current controller for wye and delta connection**
Input: 0.00 to 9999.99 [V/As]

6.12.13 Wye/Delta Connection (Only with Spindle DSP)

You can run the motor in either a wye (Y) or delta (D) connection. The switchover can be carried out during standstill or with a revolving spindle.

Delta connections enable you to run the motor at higher speeds than wye connections. Wye connections allow more power at low speeds. The motor specifications must be saved for wye and delta connections in the list of asynchronous motors. Machine parameters for current and speed controllers are available in two settings: Index 0 is for wye connection, Index 1 for delta connection. You can therefore switch to the optimum setting for the speed:

- Activate with Module 9163 the switchover between wye and delta connections.

As soon as Module 9163 is called, the NC disables the spindle drive and activates the motor specifications and machine parameters for the selected connection. You can check this with Module 9162.

After the connection has been changed externally by relay, activate with Module 9161 the drive release for the main spindle.

Module 9163 Wye/delta connection switchover

Call:

```
PS     B/W/D/K   <Axis>
                 15: Spindle
PS     B/W/D/K   <Type of connection>
                 0: Wye connection
                 1: Delta connection
CM     9163
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Incorrect axis, incorrect type of connection, or missing motor specifications

6.12.14 Operating a Second Spindle

With the TNC you can operate two spindles alternately, i.e., only one spindle can be active at a given time. The TNC provides digital and analog speed command signals for both spindles.

The following combinations of speed command signals are available for the two spindles:

Operation of the first spindle	Operation of the second spindle
Digital (TNC without integral spindle DSP)	Analog
Digital (TNC with integral spindle DSP)	Digital or analog
Analog (TNC with and w/o integral spindle DSP)	Analog

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" on page 6 – 15).

Note

If the speed encoder (with active reference mark monitoring, MP2221 bit 0) of a spindle with integral spindle DSP 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 – 180)
- Current and speed controller: For commissioning, use the machine parameters MP2020.x to MP2930.x. The index *x* depends on the PWM output in use; see also the following table.

PWM output of the second spindle	Machine parameters of the second spindle
X51	MP2020.0 to MP2930.0
X52	MP2020.1 to MP2930.1
X53	MP2020.2 to MP2930.2
X54	MP2020.3 to MP2930.3
X55	MP2020.4 to MP2930.4
X56	MP2020.5 to MP2930.5
X57	MP2020.6 to MP2930.6
X58	MP2020.7 to MP2930.7
X59	MP2020.8 to MP2930.8

Note

Axis-specific parameters must be set to zero. Machine parameters that are available for the first spindle may not be available for a second spindle.

MP4020 PLC compatibility

Format: %xxxxxxxx
Input: Bit 5 — Single- or double-spindle operation
0: Single-spindle operation
1: Double-spindle operation

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.

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

Error recognition:

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

MP13010 to MP13520 Machine parameter block for the second spindle

Input: Function and input range are identical to MP3010 to MP3520

6.12.15 C-Axis Operation

In C-axis operation, an axis and a spindle are driven alternately by the same motor.
The axis and spindle can be operated as an analog or digital axis or spindle. It is not important whether the digital spindle is controlled with an integral DSP.
The axis and spindle can each have their own position encoders. 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 in MP121.x for the spindle and in MP120.x for the axis the same speed command output.

Commissioning of the axis and the spindle:

- The current and speed controllers are commissioned only for the spindle. Note the differences in the machine parameters between spindles with and without integral spindle DSP. (See “Current Controller (Only TNC 426 PB/M, TNC 430 PA/M)” on page 6 – 213)
- Enter MP2xxx.x = 0 in the corresponding machine parameters of the axis.
- 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.

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 servo-controlled state.
- With Module 9161 bit 15, release the current and speed controllers.
- 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.

Note

Up to NC software 280 476-02, the bit for the spindle must always be transferred to Modules 9161 and 9162. As of 280 476-03, the bit for the axis **or** for the spindle can be given.

Module 9146 Saving and reestablishing actual position values

Module 9146 saves and later reestablishes the actual position values of axes. If the actual position values were saved, the last value displayed remains until they are reestablished.

Call:
PS B/W/D/K <Axes bit-encoded>
PS B/W/D/K <Mode>
0: Save actual position values
1: Reestablish actual position values
CM 9146

Error recognition:

Marker	Value	Meaning
M4203	0	Actual position values saved or reestablished
	1	Error code in W1022
W1022	1	Invalid mode
	2	Invalid axes
	24	Module was called in a spawn job or submit job

Module 9155 Axis switchover from closed loop to open loop

With Module 9155 you can switch an axis from the closed-loop to the open-loop state.

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

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing strobe or M4176 = 1
	24	Module was called in a spawn 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.

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

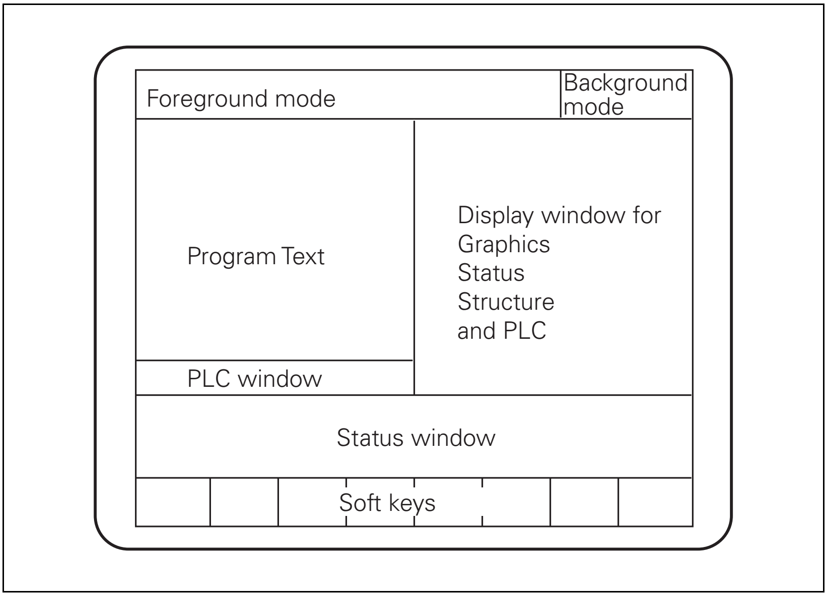
Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis number
	21	Missing strobe or M4176 = 1
	24	Module was called in a spawn job or submit job

6.13 Display and Operation

You can modify the display and operating modes of the TNC by editing the machine parameters.

The display screen is divided into separate windows. The user can select the operating functions through soft keys. (Also see the User's Manual)



6.13.1 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 OPERATION and ELECTRONIC HANDWHEEL modes you can define the **workpiece datum** with the “datum setting” function. NC programming blocks are entered with respect to the 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.

Note

M91 and M92 are active only in the block in which they are programmed.

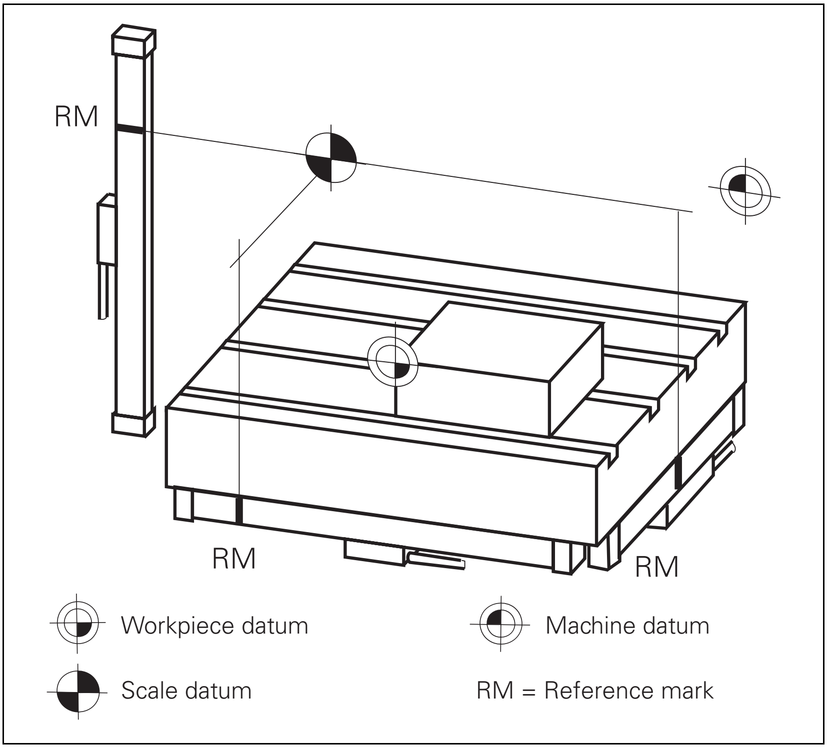
The datum can be set either only by the Datum Setting soft key or only by the soft key plus the axis keys:

- ▶ 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



Note

As of NC software 280 476-01 you can save the values for MP950.x with the actual-position-capture key.

MP950.0-8 **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

MP960.0-8 **Machine datum for axes 1 to 9**
Input: -99 999.9999 to +99 999.9999 [mm] or [°]
Values with respect to the scale reference point

MP7295 **Disabling "datum setting"**
Format: %xxxxxxxxx
Input: Bits 0 to 8 correspond to axes 1 to 9
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

6.13.2 Color Setting

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 64 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 3	0 to F	0 to 3	0 to F	0 to 3	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	Symbols
MP7358	**”Programming” soft-key display**
MP7358.0	Background
MP7358.1	Symbols
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
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	Unresolved 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	Channel 1
MP7365.2	Channel 2
MP7365.3	Channel 3
MP7365.4	Channel 4
MP7365.5	Selected channel
MP7365.6	Grid
MP7365.7	Cursor and text

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	Color 1
MP7367.2	Color 2
MP7367.3	Color 3
MP7367.4	Color 4
MP7367.5	Color 5
MP7367.6-14	Colors 6 to 14

MP7392	**Screen saver**
Input:	1 to 99 [min]
	0: No screen saver

The standard color setting is shown in the following list:

Machine parameter	Hex code up to 280 474-xx	Hex code as of 280 476-01
MP7350	$030200C	$0202020
MP7351	$03F3F0F	$0FF8888
MP7352.0 MP7352.1 MP7352.2	$0000000 $0342008 $03F3828	$0FBFBFB $0000000 $00000FF
MP7353.0 MP7353.1 MP7353.2	$0000000 $0342008 $03F3828	$0F0F0F0 $0000000 $00000FF
MP7354.0 MP7354.1 MP7354.2 MP7354.3	$0080400 $038240C $038341C $00C0800	$0FFFFFF $0000000 $00000FF $0F0F0F0
MP7355.0 MP7355.1 MP7355.2 MP7355.3	$0080400 $038240C $038341C $00C0800	$0FFFFFF $0000000 $00000FF $0FBFBFB
MP7356.0 MP7356.1 MP7356.2	$00C0800 $03F2C18 $03F280C	$0FBFBFB $00000FF $00000FF
MP7357.0 MP7357.1	$0000000 $03F3828	$0FBFBFB $0000000
MP7358.0 MP7358.1	$0000000 $03F3828	$0FBFBFB $0000000
MP7360.0 MP7360.1 MP7360.2 MP7360.3 MP7360.4	$0000000 $0203038 $00C1820 $03F3F3F $0102028	$0FBFBFB $000EEFF $00000FF $0FF0000 $00000DD
MP7361.0 MP7361.1 MP7361.2 MP7361.3 MP7361.4	$0000000 $0203038 $0203038 $03F3F3F $03F0000	$0FBFBFB $00000FA $00000FA $0FF0000 $0FF00FF
MP7362.0 MP7362.1 MP7362.2 MP7362.3	$0080400 $00C0800 $038240C $03F2C18	$0FBFBFB $0FFFFFF $00000FF $00000FF

Machine parameter	Hex code up to 280 474-xx	Hex code as of 280 476-01
MP7363.0 MP7363.1 MP7363.2 MP7363.3 MP7363.4	$0000000 $03F3F3F $0003F00 $0003F00 $03F0000	$0FBFBFB $00000FF $0FF00FF $000FB00 $0FF0000
MP7364.0 MP7364.1 MP7364.2 MP7364.3 MP7364.4 MP7364.5-6 MP7364.7 MP7364.8 MP7364.9	$0000000 $0000000 $0000000 $0000000 $0000000 $0000000 $038240C $038341C $0000000	$0AA0000 $0FF0000 $0202020 $0000000 $00000FF $0000000 $0AA0000 $000EEEE $0FBFBFB
MP7365.0 MP7365.1 MP7365.2 MP7365.3 MP7365.4 MP7365.5 MP7365.6 MP7365.7	$0000000 $0203038 $0003F00 $03F3F00 $03F003F $03F0000 $030200C $03F3F3F	$0FFFFFF $030200C $0003F00 $0FF00FF $00000FF $0FF0000 $0202020 $00000FF
MP7366.0 MP7366.1 MP7366.2 MP7366.3 MP7366.4 MP7366.5 MP7366.6 MP7366.7 MP7366.8 MP7366.9 MP7366.10 MP7366.11 MP7366.12 MP7366.13 MP7366.14	$0333333 $0281408 $0140A04 $02F2818 $0100C08 $02F2818 $02F2818 $02F2818 $02F2818 $02F2818 $02F2818 $02F2818 $02F2818 $02F2818 $02F2818	$0FBFBFB $0000000 $00000FF $0FF0000 $0FFFFFF $0FF0000 $0000000 $0080808 $0101010 $0181818 $0202020 $0282828 $0303030 $0383838 $03F3F3F
MP7367.0 MP7367.1 MP7367.2 MP7367.3 MP7367.4 MP7367.5 MP7367.6-14	$0333333 $0281408 $0140A04 $02F2818 $0100C08 $02F2818 $02F2818	$0FBFBFB $0FF0000 $000FF00 $00000FF $0F0F0F0 $0FFFFFF $0000000

6.13.3 Graphic Display

In the graphics window you can view the following graphics:

- Test graphics
- Parallel graphics
- Programming graphics
- Help illustration

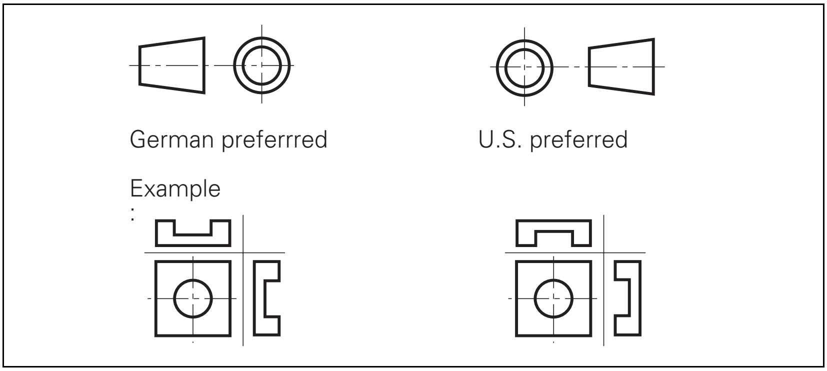
For the test graphics and parallel graphics you can choose one of three display modes:

- 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:

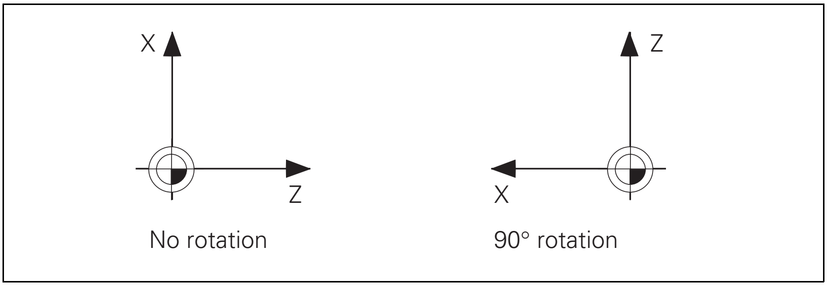
▶ Select the type of projection with MP7310, bit 0.



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 cursors

In the display in three planes you can display the position of the cursor:

▶ Switch this function on with MP7310, bit 3.

MP7310 **Graphic display mode**
Format: %xxxx
Input: Bit 0 – Projection in three planes:
0: German-preferred projection
1: US-preferred projection
Bit 1 – Rotation of 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

6.13.4 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
- Nominal 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.

MP7290.0-8	**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: 0.005 mm or 0.005° 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 determined:

▶ 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:	%xxx 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.

MP810.0-8 **Display mode for rotary axes and PLC auxiliary axes in axes 1 to 9**
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: %xxxxxxxxx
0: Software limit switch not active
1: Software limit switch active

MP7682 **Machine parameter with multiple function**
Input: %xxx
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. (TNC 426: 5 double words, TNC 430: 9 double words). 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 recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Invalid coordinate type, target address too large, or given target address is not a double-word address

Free rotation

Free rotation means that the rotary axis rotates as often as required (with a display range of 0 to 360°) without being affected by software limit switches. You can define the free rotation function through words (axis 4 and 5) or with Module 9223 (axis 1 to 9).

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 accounted for.

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 8]>
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 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 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 MANUAL OPERATION and ELECTRICAL HANDWHEEL operating modes**
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
Bit 4 – Non-functional
Bit 5 – Reserved
Bit 6 – Non-functional

		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 TNC 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:

▶ With MP7510 and following, define the center of axis rotation (See "Tilting Axes” on page 6 – 52). In the NC program, the miscellaneous function M116 is automatically cancelled with END PGM.

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/M04 and changing the polarity of the analog voltage for 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.

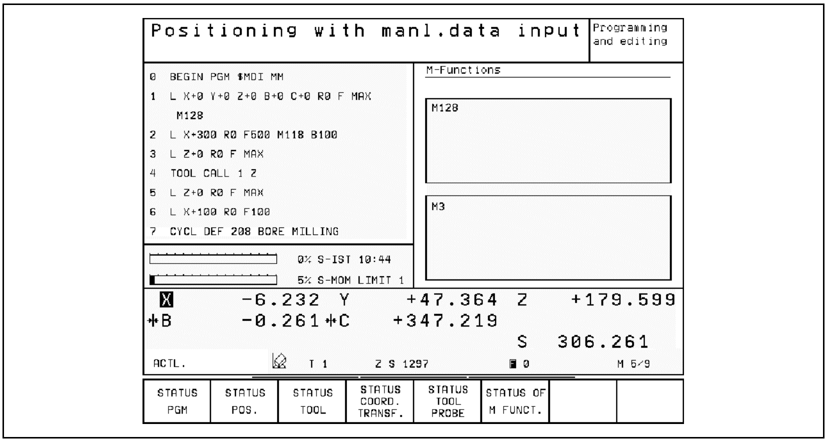
Other M functions are shown in the PLC window.

		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 can be displayed in their own status window:

▶ With Module 9088 you can display M functions in the status window or delete them.



Module 9088 Displaying the M functions

Call:

PS	B/W/D/K	<Number of the M function to be displayed>
PS	B/W/D/K	<Mode> –1: Delete all M functions in the status window 0: Delete M function 1: Display M function
CM	9088	

Error recognition:

Marker	Value	Meaning
M4203	0	M function displayed or deleted
	1	Error code in W1022
W1022	1	Invalid M-function number
	2	Invalid mode number

Control in operation

If the control is at work, e.g. executing a positioning movement or an M function, a symbol resembling an asterisk "*" is shown in the status window.

If a running NC program is interrupted with an external stop key, the control-in-operation symbol blinks in the status display window.

In the POSITIONING WITH MANUAL DATA INPUT, the PROGRAM RUN, SINGLE BLOCK and the 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: Erase 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 erased/displayed 1: Incorrect error code 2: Control-in-operation symbol is already 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 recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Incorrect transfer parameter

		Set	Reset
M4175	**Program interruption, control-in-operation symbol blinks**	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 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.

Input	Erase with PGM MGT	Erase with M02, M30, END PGM	Status display	Tool data	Q para-meters
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	Main operating mode of editor	0: PROGRAMMING AND EDITING 1: TEST RUN
1	Main operating mode of machine	0: REFERENCE MARK TRAVERSE 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 (as of 280 472-xx) 4: Pocket table selected (as of 280 472-xx)
4	Displayed screen window	Bit encoded Bits 0 to 7: Editing screen Bit 0=1: Editing screen is displayed Bit 1=1: Window mode 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/graphic window active Bit 6/7: Reserved Bits 8 to 15: Machine screen Bit 8=1: Machine screen is displayed Bit 9=1: Window 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/graphic window active Bit 14/15: Reserved

Transferred number		Return code
5	Selected file in ”Programming and editing” and ”Test run” modes	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	0 to 8: Axes 1 to 9
8	Selected axis for actual position capture in POSITIONING WITH MANUAL DATA INPUT	0 to 8: Axes 1 to 9
9	Handwheel axis	–1: None or more than one Axes 1 to 9
10	Handwheel axis, bit encoded	Axis 1 to 9
	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 file (ISO NC PGM)
17	Display format	0: MM 1: INCHES
18	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
22	Status of M128	0: M128 not active 1: M128 active

Transferred number		Return code
23	Handwheel superimposition with M118	
	Handwheel interpolation factor	
31	Axis 1	0 to 10
32	Axis 2	
33	Axis 3	
34	Axis 4	
35	Axis 5	
36	Axis 6	
37	Axis 7	
38	Axis 8	
39	Axis 9	
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 recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid
	20	Call was not in a submit or spawn job

6.13.5 NC Program

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

Canceling an NC Program

An NC macro can be called automatically if an NC program was cancelled 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 TNC 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 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

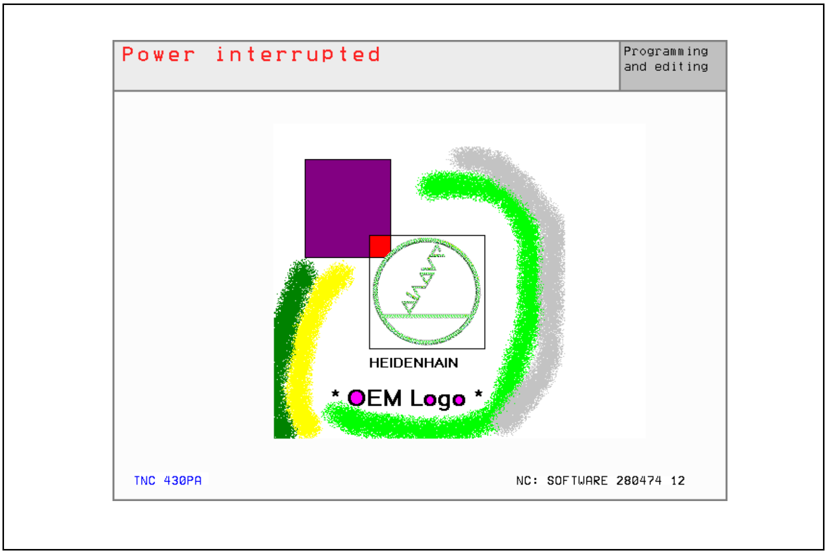
MP7683 **Executing pallet tables**

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.

6.13.6 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.



Note

Your logo must exist in 16-color bitmap format. The logo's width in pixels must be divisible by eight.

- Convert the logo with the HEIDENHAIN conversion tool "Bmp2Logo" from .BMP format to .VEC format. The tool also creates a *.SYS file.
- The two files must be given the names LOGO.SYS and LOGO.VEC.
- On the TNC, create a new directory named PLC:\LOGO.
- Move LOGO.VEC to the PLC:\LOGO directory.
- Move LOGO.SYS to the main directory PLC:\.

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 9189. If the control is shut down (either with Module 9189 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.

		Set	Reset
M4179	**Control is shut down**	NC	NC

Module 9189 Shutting down the control

The control is shut down with Module 9189. After shutdown, the PLC remains operable. It can therefore react to a signal to switch off the machine after conclusion of this module.
The information windows, which appear during shutdown via soft key, do not appear.

Call:
CM 9189

Error recognition:

Marker	Value	Meaning
M4203	0	Control was shut down
	1	Error code in W1022
W1022		Module was not called in a spawn job or submit job

Resetting the control

Module 9279 Control reset

Module 9279 carries out a control reset. This means the control is shut down and then restarted (the PLC cannot be run) **or** only the control is shut down (the PLC can still be run).
In either case, no message is shown on the monitor to say that the control is being shut down.

Call:
PS B/W/D/K <Mode>
0: Shut down the control
1: Control is shut down and restarted
CM 9279

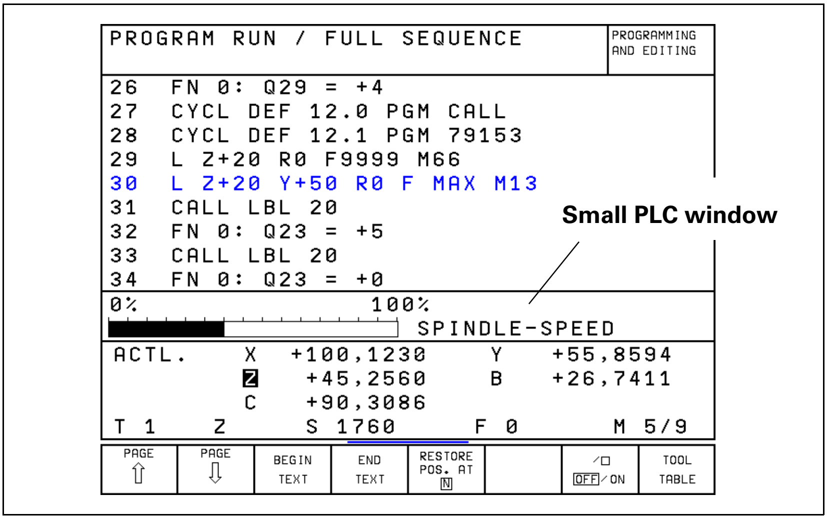
Error recognition:

Marker	Value	Meaning
M4203	0	Control reset is carried out
	1	Error code in W1022
W1022	2	Invalid mode
	20	Module was not called in a spawn job or submit job

6.13.7 Small PLC Window

The small PLC window is shown in the following operating modes:

- MANUAL
- 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.

- 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, 9083 must be called in a submit 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.

The background color is defined in MP7320.2 and MP7356.0.

Call:
CM 9080

Error recognition:

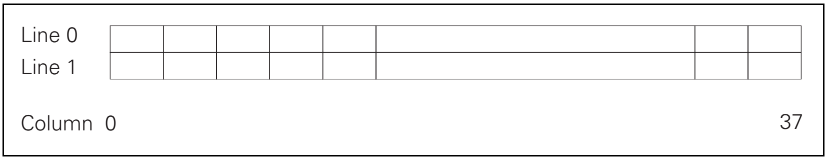
Marker	Value	Meaning
M4203	0	No error
	1	Call was not in a submit or spawn job

Module 9081 Interrogating the status of the small PLC window
With this module you can ascertain whether a small PLC window is being displayed.

Call:
CM 9081
PL B/W/D <Status of the small PLC window>
Bit 0 = 1: A small window is in the selected screen (background or foreground)
Bit 1 = 1: Small PLC window in the foreground

Module 9082 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 (see table). In the event of error, no string is shown.



Number	Machine parameter
1	MP7354.0
2	MP7356.0
3	MP7352.0
4	MP7353.0
5	MP7357.0
6	MP7352.1
7	MP7353.1
8	MP7350
9	MP7357.1
10	MP7354.1
11	MP7356.2
12	MP7356.1
13	MP7354.2
14	MP7352.2
15	MP7351

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.

Color two is the background color of the PLC window. It cannot be used as foreground color.

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>
                 0 to 7
CM    9082
```

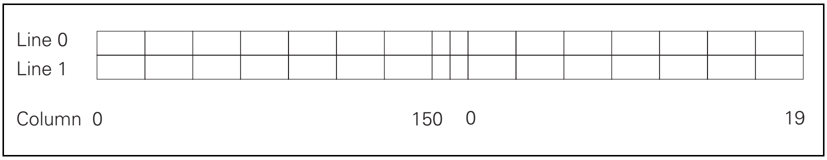
Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Call was not in a submit or spawn job or line less than zero or greater than 1 or 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 can no longer be displayed in the screen window.

Module 9083 Showing a moving-bar diagram in the small PLC window

The moving-bar diagram is shown in the specified line with the specified length and colors.

The diagram can also be limited to the left half of each line. In this case the ASCII text is limited to max. 19 characters in the right half.



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 of the currently selected graphic. (MP736x.1). Color two is the background color of the PLC window. It can be used, for example, for margin or scale graduation if they are not to be shown. The color must be selected with MP735x according to the table ((See "Color Setting" on page 6 – 225)).

Call:

```
PS      K/B/W/D   <Line number>
                  0 or 1
PS      K/B/W/D   <Color for bar>
                  0 to 15
PS      K/B/W/D   <Color for margin and scale graduation>
                  0 to 15
PS      K/B/W/D   <Current length of the bar>
                  0 to 150
PS      K/B/W/D   <Maximum length of the bar>
                  0 to 150
CM      9083
```

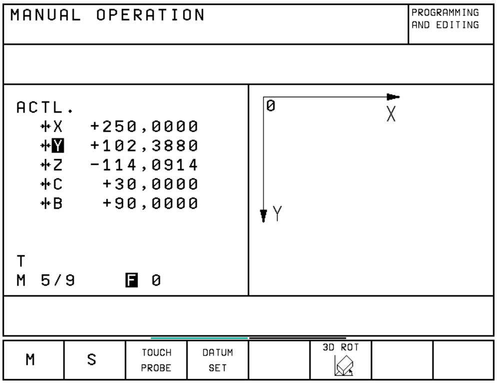
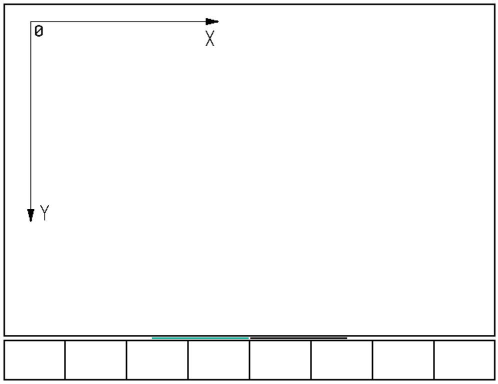
Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Call was not in a spawn or submit job or line less than 0 or greater than 1

6.13.8 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.

- Select the display mode with the screen management key or with Module 9202.

<table>
<tr><th colspan="2">Mode 1</th><th colspan="2">Mode 2</th></tr>
<tr><td colspan="2">
</td><td colspan="2">
</td></tr>
<tr><td>SMALL</td><td>17 lines, 39 columns
0 = y16/x8</td><td>SMALL</td><td>27 lines, 79 columns
0 = y16/x8</td></tr>
<tr><td>MEDIUM</td><td>11 lines, 19 columns
0 = y24/x8</td><td>MEDIUM</td><td>18 lines, 39 columns
0 = y24/x8</td></tr>
<tr><td>LARGE</td><td>5 lines, 9 columns
0 = y48/x8</td><td>LARGE</td><td>9 lines, 19 columns
0 = y48/x8</td></tr>
</table>

- Define the character size with the special command "charsize = " (See "Special commands:" on page 6 – 260). 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:

- In Module 9210, enter the name of the screen mask in order to activate the PLC window.

Format instructions

Format instructions are enclosed in quotation marks (“ ”). Variables are transferred as parameters.

Example:

"Position: %+8.3F",D3884/n=4/c

Always display the algebraic sign

Total number of digits, including decimal point and algebraic sign

3 decimal places

Display as a floating-point

Convert D3884 into a floating-point number and divide by 10^x

Update cyclically

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 TNC 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
`B0 to B4095`	PLC bytes, integer
`W0 to W4094`	PLC words, integer
`D0 to D4092`	PLC double words, integer 0/1
`M0 to M4999`	PLC markers, integer 0/1
`I0 to I383`	PLC inputs, integer 0/1
`O0 to O191`	PLC outputs, integer 0/1
`T0 to T303`	PLC timers, integer 0/1
`C0 to C143`	PLC counters, integer 0/1
`S0 to S15`	PLC strings, string [128]
`S#D0 to S#D999`	PLC dialogs, string
`S#E0 to S#E999`	PLC error texts, string
`TIME[0] to TIME[15]`	System time as in Module 9055, char
`AXISCHAR[0] to AXISCHAR[4]`	Code letters for NC axis, char
`MP up to..`	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 of the machine		
MACHINE.LIMIT1PL [5]	Double	Pos. limit switch group 1
MACHINE.LIMIT1MI [5]	Double	Neg. limit switch group 1
MACHINE.LIMIT2PL [5]	Double	Pos. limit switch group 2
MACHINE.LIMIT2MI [5]	Double	Neg. limit switch group 2
MACHINE.LIMIT3PL [5]	Double	Pos. limit switch group 3
MACHINE.LIMIT3MI [5]	Double	Neg. limit switch group 3
MACHINE.PRESET1 [5]	Double	Preset values 1
MACHINE.PRESET2 [5]	Double	Preset values 2
MACHINE.PRESET3 [5]	Double	Preset values 3
MACHINE.POSINC [5]	Double	
MACHINE.DRVOFFSET [5]	Double	
MACHINE.HANDW_FACTOR [5]	Double	Handwheel 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 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 codes 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 “TNC Character Set” on page 6 – 268)

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 Setting” on page 6 – 225).

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," "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 S0 to S15 or in an immediate string.
  If no path name is specified, the path for the language indicated in MP7230.3 (help files) is used.

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>
0 to 7: String memory to S15
–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:

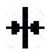
PS	B/W/D/K	<Number>
CM	9211	
PL	B/W/D	<Status information> –1: Error

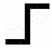
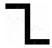
Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Invalid number of the status information

6.13.9 TNC Character Set

Small characters

Nr.	Character	Nr.	Character	Nr.	Character	Nr.	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

Nr.	Character	Nr.	Character	Nr.	Character	Nr.	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

Nr.	Character	Nr.	Character	Nr.	Character	Nr.	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

Nr.	Character	Nr.	Character	Nr.	Character	Nr.	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

Nr.	Character	Nr.	Character	Nr.	Character	Nr.	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		

6.13.10 PLC Soft Keys

In the following operating modes you can display your own soft keys through the PLC:

- MANUAL
- ELECTRONIC HANDWHEEL
- POSITIONING WITH MANUAL DATA INPUT
- PROGRAM RUN, FULL SEQUENCE
- PROGRAM RUN, SINGLE BLOCK

You can create the soft keys with PLCdesign.

When a PLC soft key is pressed the NC enters the soft-key number in W302. 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.

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 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 (as of 280 472-xx)
CM      9200
```

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Incorrect transfer parameters (e.g. KF address not in address range of the PLC code)
	2	Line nr. < 0 (not –1) in the constants field
	3	Invalid KF address
	24	Module was called in a spawn job or submit job
	25	More than 32 elements in the constants field

Display/delete PLC soft key

▶ Procedure for displaying/deleting a PLC soft key

Module 9201 Display/delete PLC soft key
If no PLCSOFTK.SYS file exists, or if the specified line does not exist, no soft key is generated. In an existing PLC soft-key level, the soft key is displayed/deleted at the specified position.

Can only be called from the sequential program.

Call:
PS B/W/D/K <Soft-key number (line no./delete)>
 0: Line no.
 –1: Delete soft key
PS B/W/D/K <Position no.>
 0 to 31
PS B/W/D/K <Soft-key mode>
 0: Soft key for displayed PLC window
 1: Soft key in current operating mode
 2: Append soft key to NC soft key
CM 9201

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Transfer parameter out of value range
	2	Line nr. < –1
	24	Module was called in a submit job

Select/deselect PLC soft keys and PLC windows

Module 9202 Select/deselect PLC soft keys and PLC windows
With this Module 9202 you activate the display with PLC windows or the PLC soft-key display. This module works like the screen management key.

Call:
PS B/W/D/K <Display mode>
 0: PLC soft key/window deselected
 1: Small PLC soft key/window deselected
 2: Large PLC soft key/window deselected
 3: Large PLC soft key/window selected while table editor is active
CM 9202

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Transfer parameter out of value range

6.13.11 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 an HLP file.

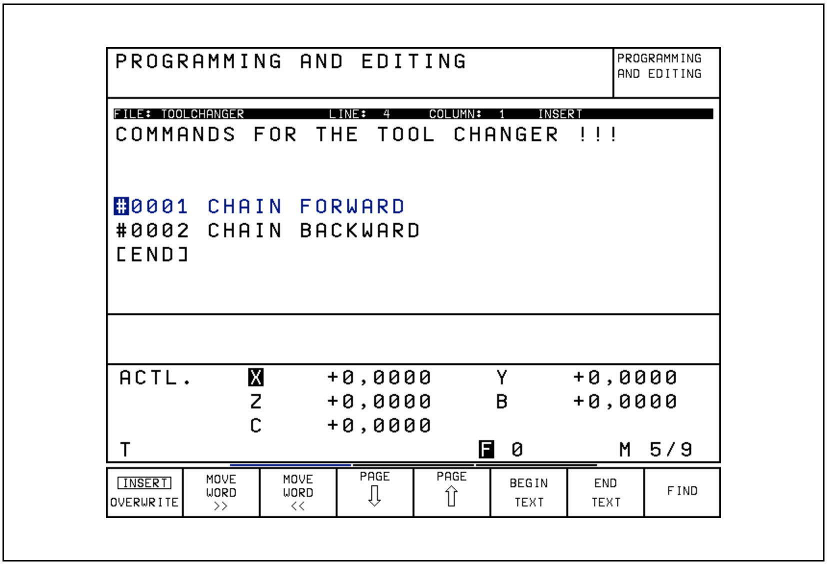
With machine commands:

- ▶ 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.

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.

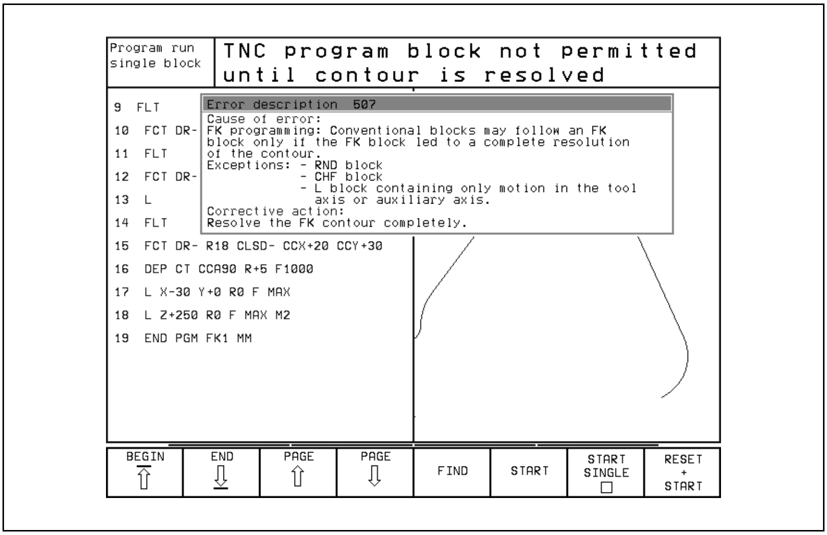
HELP files are saved externally with the identifier “J”.

		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”
  - Texts under the heading “Corrective Action”
- Define the names of the files in the system file OEM.SYS with the commands:
  - PLCERRREASON= cause of error
  - PLCERRFIX= corrective action
- 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 TNC 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 TNC.

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

6.13.12 PLC Pop-Up Window

As of NC software: 280 472-01

The PLC pop-up (i.e. superimposed) window is shown in the following operating modes:

- MANUAL
- 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 by using the cursor keys and the ENTER key, or the hot keys.

The module 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 TNC 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 (e.g. help window) are put into 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 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.

Module 9215 Activating a PLC pop-up window
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 – Positioning error 0: No hot keys 1: Hot keys (0 to 9 and A to Z) before menu items Bit 5 – Vertical / Horizontal 0: Vertical arrangement 1: Horizontal arrangement (bit 4 = 0)
CM 9215	
PL B/W/D	<Selected line> 0 to n: Line number from list –1: No selection made (END, NOENT) 2: For error see W1022

Error recognition:

Marker	Value	Meaning
M1022	0	Incorrect mode transferred
	3	Not a valid string for file name or heading
	6	Window cannot be displayed (internal error, e.g. problems with memory or operating system)
	20	Module was not called in a spawn job
	28	Another PLC pop-up window is open
	36	File with the list could not be found

6.13.13 M Functions

In the TNC 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	Feedrate F in mm per spindle revolution[c]	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
c. 280 474-xx: µm per spindle revolution

- In the PLC, evaluate the M functions which 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 recognition:

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 recognition:

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 the 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 6 – 452).

- 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 TNC 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: 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

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 also 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 be present. 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.
To synchronize the current machine status and the look-ahead calculation with an NC macro call, (See "NCMACRO.SYS" on page 7 – 44).

For behavior during a block scan, (See "Instructions in MGROUPS.SYS" on page 6 – 297).

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.	

6.13.14 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.

- 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 page 6 – 276).
  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
- **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.

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 no. .PET table>
0 to 999: Line number
–1: Blinking error message **EMERGENCY STOP PLC**
CM 9085

Error recognition:

Marker	Value	Meaning
M4203	0	Error message displayed or in queue
	1	Error code in W1022
W1022	1	Line number 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 no. .PET table>
0 to 999: Line number
–1: Erase all PLC error messages

CM 9086

Error recognition:

Marker	Value	Meaning
M4203	0	Error message displayed or in queue
	1	Error code in W1022
W1022	1	Line number not available
	8	Incorrect operating mode, error marker compatibility set

Module 9087 Status of PLC error message

Call:

PS B/W/D/K <Line no. .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 recognition:

Marker	Value	Meaning
M4203	0	Status information was read
	1	Error code in W1022
W1022	1	Invalid line number of status code

6.13.15 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 "Application of OEM Cycles” on page 9 – 5)). You can influence the function of many HEIDENHAIN standard cycles through machine parameters.

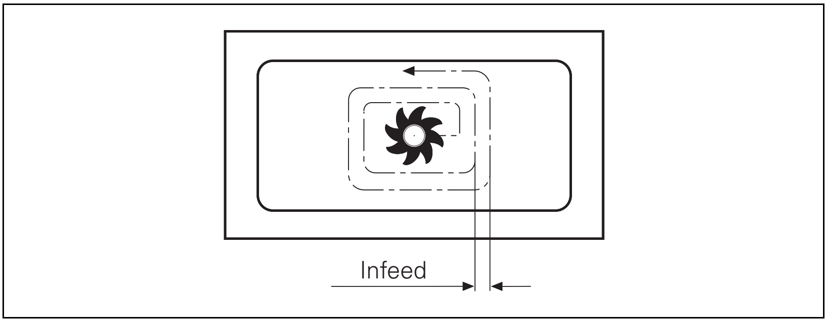
For the Tapping and Oriented spindle stop cycles:
(See "Spindle” on page 6 – 180).

For the Touch probe cycles: (See "Touch Probe” on page 6 – 346).

Pocket milling

Cycles 4 and 5:

▶ In MP7430, enter the overlap factor for roughing out a rectangular or circular pocket.



Infeed = (MP7430) · cutter radius

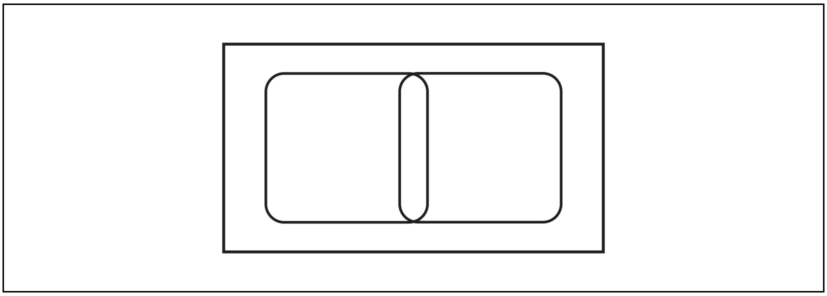
MP7430	**Overlap factor for pocket milling**
Input:	0.001 to 1.414

Cycles for milling pockets with combined contours

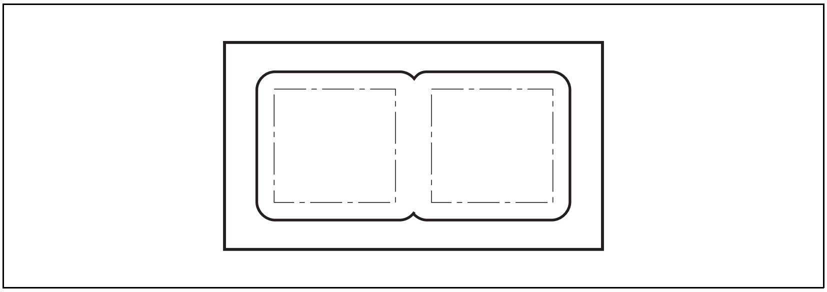
Cycles 6, 14, 15, 16:

▶ 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

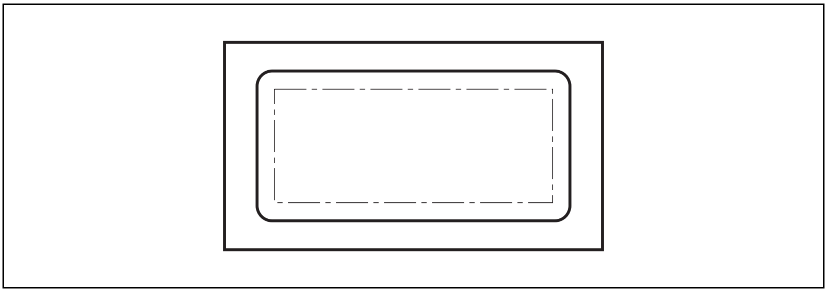


The programmed contours of two pockets intersect slightly.



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).

- With MP7510 and following, define the center of axis rotation.
(See "Tilting Axes" on page 6 – 52)
- 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

6.13.16 Returning to the Contour

With HEIDENHAIN contouring controls you can resume an interrupted 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 causes 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.
- Activate the calculation for the specific axes with MP7450.

With flexible tool-pocket coding in the central tool file (See "Tool Changer” on page 6 – 400), the change of pocket number in the tool file must be prevented during block scan if the TOOL CALL blocks are not collected:

- Set M4542.

The block scan can be interrupted by a programmed STOP or with M06, whereby you can have the programmed dwell time included:

- With MP7680, bits 3 and 4, select the parameters for the block scan.
- With MP7451.x, define the feed rate for returning to the contour.

If an NC program block is interrupted in **`Single block mode`** or by a STOP block and the positions of NC axes are changed, the NC program can be restarted at the changed positions. If in OEM.SYS **`STRICTREPOS = YES,`** the function for restoring the position is activated. (See “OEM.SYS” on page 7 – 41)

Note

The tool data cannot be correctly offset in the block scan if you change them in the PLC or update them with M4538.

As of NC software 280 476-01 you can save the values for MP9510.x with the actual-position-capture key.

MP951.0-8	**Simulating tool change position for TOOL-CALL during block scan for axes 1 to 9**
Input:	-99 999.9999 to +99 999.9999 [mm] or [°]
MP7450	**Offsetting the tool change position from MP951.x in block scan**
Format:	%xxxxxxxxx
Input:	Bits 0 to 8 correspond to axes 1 to 9: 0: Do not offset 1: Offset

MP7451.0-8	**Feed rate for returning to the contour for axes 1 to 9**
Input:	10 to 300 000 [mm/min]
MP7680	**Machine parameter with multiple function**
Format:	%xxxxxxxxxxxxxx
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 it 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

As of NC software 280 470-12, 280 472-13, 280 474-12, 280 476-01:
The error message **PLC function not permitted** appears if during **RESTORE MACHINE STATUS** the PLC shifted the datum or switched traverse ranges or spindles.

As of NC software 280 476-03:
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 the 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 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.

		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

In the following functions you must use REMAIN=OUTPUT:

- 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 6 – 400) and (See "Calling an NC macro with an M function” on page 6 – 287)).
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

6.13.17 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

6.13.18 Files

The TNC enables you to edit various file types. File types are designated with an extension after the file name. A file name can consist of up to eight characters (letters and numbers).

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

Block number increment for ISO programs

▶ Enter the block number increment in MP7220 for ISO programs.

MP7220 **Block number increment for ISO programs**
Input: 0 to 250

Selecting a file

If you are in the PROGRAM RUN, SINGLE BLOCK or PROGRAM RUN, FULL SEQUENCE mode, you can select a file through the PLC. W1018 returns the number of files opened by the PLC. W1020 returns the number of all open files. A maximum of 61 files can be open at the same time (total number of users and files opened by the PLC and the NC).

▶ With Module 9290, transfer the name of the file to be selected.

Module 9290 Selecting a file
In the PROGRAM RUN, SINGLE BLOCK or PROGRAM RUN, FULL SEQUENCE mode you can select a file.

Call:
PS B/W/D/K/S<String number or file>
CM 9290

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid string was transferred
	8	Control is not in the SINGLE BLOCK or FULL SEQUENCE 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

6.13.19 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 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 Table, Pocket Table” on page 6 – 400).

With FN17 and FN18 you can read and overwrite the values in the datum table (OEM cycles).

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)

6.13.20 Pallet Management (as of NC software 280 472-01)

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 6 – 310).

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 two prototypes into the directory PLC:\PROTO.
Both prototypes are offered when you create a new pallet table. If you do not want this to happen, delete a prototype from the PLC:\PROTO directory. Then the existing prototype is used automatically.

- Prototyp.P = standard prototype (PAL/PGM, NAME, DATUM, X, Y, Z)
- Proto_to.P = prototype for tool-oriented machining

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-STATE	Tool-oriented	Optional field: Machining status, BLANK, ENDED, INCOMPLETE
METHOD	Tool-oriented	Mandatory field: Type of machining TO = tool-oriented, WPO = workpiece-oriented, CTO = continued tool-oriented (for multiple entries)
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.

Name	Type of machining	Meaning
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. For pallet entries, the values are referenced to the machine datum (MP960.x). For NC programs the values are referenced to the pallet datum. Only the fields X, Y and Z are used in the standard format.
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 TNC enters a code. This code enables the TNC 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 field is used, an NC program can be run only if this field contains the entry MA (= pallet for the machine).
LOCK	Workpiece-oriented/tool-oriented	Optional field: If an entry is made in this column, the line of the entry 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

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.

- With MP7683 bit 4, specify whether the active pallet table should be editable with the EDIT PALLET soft key.

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 6 – 310)).

With Module 9035 you can interrogate the active line of the pallet file, and with Module 9090 or 9281 you select a certain line in the pallet table. Unlike Module 9090, with Module 9281, a datum shift or datum setting can be executed immediately.

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 “DATUM1.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

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 TNC is in another mode, the selection will be made when the control returns to the program run 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. The 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

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 mode. Datum shift and datum setting can be run immediately. If the TNC is in another mode, the selection will be made when the control returns to the program run 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. The 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. If this specific NC macro is not defined in NCMACRO.SYS, a HEIDENHAIN standard macro is run.

The HEIDENHAIN standard macro performs the following functions:

- Positioning to clearance height
- Execution of M146
- Tool change through TOOL CALL. The standard tool-change macro is called.

With FN18: 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 interrogate the current line or pallet name with FN18: 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 7 – 44).

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 full-sequence or single-block mode, a pallet table is selected, and no macro or NC program is running.

Call:
PS B/W/D/K <Pallet number>
PS B/W/D/K <Line number>
CM 9280

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	7	The file entered in the entry PALETT= does not exist
	8	Control is not in the SINGLE BLOCK or FULL SEQUENCE mode
	20	Module was not called in a spawn job or submit job
	28	An NC program or NC macro is running
	29	Selected file is invalid or does not exist
	30	There is no PALETT= entry in the NCMACRO.SYS file
	36	NCMACRO.SYS does not exist

NC macro at the end of an NC program

- In NCMACRO.SYS, use the entry **PALEPILOG=** to define the complete path and name of the NC macro that is to be called at the end of an NC program that was started from the pallet table.

To synchronize the current machine status and the look-ahead calculation with an NC macro call, (See “NCMACRO.SYS” on page 7 – 44).

NC software 280 470-xx

In the pallet tables you can assign NC programs and datum tables to specific workpieces.

You can define up to 255 assignments per table:

- In MP7226.0, define the size of the table.
- With Module 9090, select the assignment by entering the line number. With the next NC START the selected NC program is run with the associated datum table.

The desired pallet must have the status “M”:

- With PGM MGT, select the pallet table in the PROGRAM RUN, FULL SEQUENCE operating mode.

If no datum table is specified in the pallet table, the previous file is kept.

Files that you have disabled in MP7224.0 are ignored in the pallet table. If locked HEIDENHAIN programs or ISO programs are selected, the error message **NC PROGRAM NOT FOUND** appears.

MP7226.0 Size of the pallet table
Input: 0 to 255 [lines]

Module 9090 Assignment of a pallet table
Call only in a submit job or spawn job.

Call:

PS	B/W/D/K	<Line number in pallet table>
CM	9090	
PL	B/W/D	<Error condition> As of 280 470-01: 0: Files were selected 1: Call was not in a submit or spawn job 2: Call during running program 3: Pocket calculator not in default state 4: Pallet table not found 5: Line in pallet table does not exist 6: Incorrect type entered for NC program or point missing 7: NC program not found 8: NC program ambiguous 9: Datum table not found 11: Pallet entry missing in NCPATH.SYS 12: Incorrect file name extension As of 280 472-01: 0: Files were selected 1: Call was not in a submit or spawn job 2: Call during running program 3: No pallet table selected in main program 4: Line in pallet table does not exist

6.13.21 Freely Definable Tables

As of NC software: 280 472-01

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, FN27 and FN28 (see 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 7 – 3)
- In the PLC:\PROTO directory, create a table with the extension .TAB.

If you have not yet defined prototypes, you will be given a standard prototype. 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 includes 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 200 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 stand behind 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>
                 0 to 15
PS    B/W/D/K/S<String number, result>
                 0 to 15
CM    9245
```

Error recognition:

Marker	Value	Meaning
M4203	0	Field was read
	1	Error code in W1022
W1022	1	Line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	7	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 (extension .TAB, .P)
	30	Column name not found

Module 9255 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 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> 0 to 15
CM	9255	
PL	B/W/D	<Result>

Error recognition:

Marker	Value	Meaning
M4203	0	Field was read
	1	Error code in W1022
W1022		See Module 9245

Module 9246 Writing to a field in a table

Open the table with the file name extension .TAB or .P with Module 9240, and not in the “buffered” mode.

The field defined by the column name and line number is overwritten.

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> 0 to 15
PS	B/W/D/K/S	<String number, contents to be written> 0 to 15
CM	9246	

Error recognition:

Marker	Value	Meaning
M4203	0	Field was written to
	1	Error code in W1022
W1022	1	Line does not exist in table
	2	Incorrect “file handle” or table was opened in “buffered” mode
	3	Impermissible string numbers
	6	Table is write-protected
	7	Not a numerical field (Module 9256)
	11	The transferred value cannot be saved to the addressed field. Incorrect format.
	20	NCMACRO.SYS does not exist
	29	The opened file is not a table (extension .TAB, .P)
	30	Column name not found

Module 9256 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.

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> 0 to 15
PS	B/W/D/K	<Numerical value to be written>

CM 9256

Error recognition:

Marker	Value	Meaning
M4203	0	Field was written to
	1	Error code in W1022
W1022		See Module 9246

Module 9247 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> 0 to 7
CM	9247	
PL	B/W/D	<Line that fulfills the condition> –1: Error code in W1022

Error recognition:

Marker	Value	Meaning
W1022	1	Start line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	7	Module could not be read from the table
	20	Module was not called in a spawn job or submit job
	29	Incorrect file format
	30	Column name not found
	31	Syntax error in the transferred condition
	32	No data record found that fulfills the condition

Starting the PLC editor for tables

In the machining modes a table editor can be started:

▶ Specify the lines and column that are to be displayed.

You can provide the PLC editor only with tables with the file name extensions TAB or P. A temporary file with the name SYS:\TEMP\PLCTABED.TAB is saved.

With the Modules 9240, 9241, 9245 and 9247 you can check this temporary file before you place the edited data with Module 9251 into the original table.

▶ 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. The 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 in the original table directly, you cannot cancel the changes with Module 9251.

If you do not edit directly in the original table 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 interrogate 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 formula
CM	9250	

Error recognition:

Marker	Value	Meaning
M4203	0	Editor was opened
	1	Error code in W1022
W1022	1	First and last line do not define a meaningful range, or incorrect value for mode was transferred
	3	Impermissible string number
	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 recognition:

Marker	Value	Meaning
M4203	0	Editor was opened
	1	Error code in W1022
W1022	3	Incorrect value was transferred for mode
	6	Changes could not be saved in the original table
	20	Module was not called in a spawn job or submit job
	28	PLC editor had not been opened with Module 9250

Module 9252 Positioning the cursor in the PLC editor

With this module you place the entry field of the PLC editor on a defined line and column. The line is defined relative to the starting line of Module 9250. The designated column must be defined in Module 9250.

Call:
PS B/W/D/K/S<String with column name>
PS B/W/D/K <Line>
CM 9252

Error recognition:

Marker	Value	Meaning
M4203	0	Cursor was set
	1	Error code in W1022
W1022	1	Incorrect line number
	3	Incorrect string number
	20	Module was not called in a spawn job or submit job
	30	Incorrect column name
	35	PLC editor is not open (Module 9250)

6.13.22 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 recognition:

Marker	Value	Meaning
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 recognition:

Marker	Value	Meaning
M4203	0	File was closed
	1	Error code in W1022
W1022	2	Incorrect file handle
	20	Module was not called in a 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 result from Module 9242.
If the file was opened in the "record oriented" mode (tables), the cursor is positioned line by line.
If the file was opened in the "buffered" mode, the cursor is positioned character by character.
If you indicate a position before the beginning or after the end of the file, the cursor is positioned to the beginning or end of the file, respectively. The addressing of the new position is relative to the beginning or end of the file, or to the current position. You can interrogate the current position by transferring the position value zero relative to the current position.

Call only in a submit job or spawn job.

Call:

PS	D	<File handle> Number from Module 9240
PS	B/W/D/K	<Desired position>
PS	B/W/D/K	<Mode> 0: Position relative to the file beginning 1: Position relative to the current position 2: Position relative to the file end
CM	9242	
PL	B/W/D/K	<New position> –1: Error code in W1022

Error recognition:

Marker	Value	Meaning
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), maximum 126 characters.

Call:

PS	D	<File handle> Number from Module 9240
PS	B/W/D/K	<String number with result> 0 to 7
CM	9243	
PL	B/W/D	<Number of read bytes> >0: Line has been read 0: File end has been reached –1: Error code in W1022

Error recognition:

Marker	Value	Meaning
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 number of data specified in the transfer string is overwritten.

If file is opened in “record oriented” mode:

- Processing time is longer.
- The data are saved immediately 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> 0 to 7
CM	9244	
PL	B/W/D	<Number of written bytes (including LF)> –1: Error code in W1022

Error recognition:

Marker	Value	Meaning
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

6.13.23 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)

6.13.24 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 the machine parameter list
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 (option)
LOGBOOK	Read out the log
FAILTEST	Simulate an internal EMERGENCY STOP

The code of the entered code number is entered in the 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

6.13.25 Programming Station

With MP7210 you can set the control for use as a programming station without a machine.

In this setting, the PROGRAMMING AND EDITING and TEST RUN modes are operable.

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

6.13.26 Conversational Language

The TNC is delivered with all 13 NC-dialog human languages already loaded:

- In MP7230.0 select the conversational language in which you wish to work.

If the NC dialog messages for the selected language are not on the hard disk, the error message **LANGUAGE LOAD ERROR** appears. You can continue working in the default language English.

You can write your own dialog messages and save them in several languages:

- Save your dialog messages in permanently defined directories in the PLC partition.

These directories are:

PLC:\LANGUAGE\ CZECH\
DANISH\
DUTCH\
ENGLISH\
FINNISH\
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

6.13.27 Memory Test

- With MP7690 you specify the memory that is to be tested during switch-on. The message MEMORY TEST is displayed for the duration of the test.

MP7690 **MEMORY TEST during switch-on**

Format: %xxx
Input: Bit 0 – Test the RAM
Bit 1 – Test the EPROM
Bit 2 – Test the hard disk
0: MEMORY TEST during switch-on
1: No MEMORY TEST during switch-on

6.13.28 Arc End-Point Tolerance

The TNC 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]

6.13.29 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

6.13.30 Power Interrupted Message

After the machine supply voltage is switched off, the TNC displays the error message **POWER INTERRUPTED**:

▶ Press the CE key to acknowledge this message and activate the PLC.

With MP7212 you can suppress this message, e.g. for unattended operation.

MP7212 **Power interrupted message**
Input: 0: Acknowledge **Power Interrupted** message with the CE key.
1: **Power Interrupted** message does not appear

6.13.31 Operating Times

The TNC 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: The operating times are displayed in the MOD function "Machine Time":

- 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 TNCTIME, MACHINETIME and PROGTIME.

For the operating times PLCTIME0 to PLCTIME7:

- Start with Module 9190.
- Stop with Module 9191.

Except for TNCTIME, all operating times are saved during a hard-disk backup with the program TNCBACK (See "NC Software Exchange" on page 2 – 33).

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 **Displaying and resetting 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 4095
Dialog no. from the file **PLCDIALOG=** (OEM.SYS)

Module 9190 Starting the operating times
You start one or more operating times.

Call:
PS B/W/D/K <PLC operating time>
Bits 0 to 7 represent PLC operating times 1 to 8
CM 9190

Error recognition:

Marker	Value	Meaning
M4203	0	PLC operating time started
	1	Incorrect parameter

Module 9191 Stopping the operating times
You stop one or more operating times.

Call:
PS B/W/D/K <PLC operating time>
Bits 0 to 7 represent PLC operating times 1 to 8
CM 9191

Error recognition:

Marker	Value	Meaning
M4203	0	PLC operating time started
	1	Incorrect parameter

Module 9192 Reading the operating times
You read the current value of an operating time. The current value is transferred in seconds. If the value is greater than 2 147 483 648 (approx. 69 years), a negative number will be transferred.

Call:

PS	B/W/D/K	<Number of the operating time> –3: TNCTIME –2: MACHINETIME –1: PROGTIME 0 to 7: PLCTIME0 to PLCTIME7
CM	9192	
PL	B/W/D	<Current time [s]> –1: Error

Error recognition:

Marker	Value	Meaning
M4203	0	PLC operating time started
	1	Incorrect transfer value, or module was not called in a spawn job or submit job

Module 9193 Setting the operating times
You overwrite the current value of the operating time. The old value is lost irretrievably.

The time for "control on" (TNCTIME) 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: MACHINETIME –1: PROGTIME 0 to 7: PLCTIME0 to PLCTIME7
PS	B/W/D/K	<New time [s]>
CM	9193	

Error recognition:

Marker	Value	Meaning
M4203	0	Operating time was overwritten
	1	Incorrect transfer value, or module was not called in a spawn 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: TNCTIME –2: MACHINETIME –1: PROGTIME 0 to 7: PLCTIME0 to PLCTIME7
PS	B/W/D/K	<Alarm threshold [s]>
PS	B/W/D/K	<Number of the alarm markers>
CM	9194	

Error recognition:

Marker	Value	Meaning
M4203	0	Alarm function activated
	1	Incorrect transfer value, or module was not called in a spawn job or submit job

Module 9195 System time
The TNC operates with UNIX system time. The system time contains the number of seconds accumulated since 0:00 hours on January 1, 1970.

When the TNC is shipped it is calibrated for Universal Time (also known as Greenwich Mean Time).

In MP7235, enter the time difference between the local time and Universal Time so that the time of the program management matches the local time.

With Module 9195 you can read the current value of the 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.

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> 0 to 7
PS	B/W/D/K	<Format> 0: DD.MM.YYYY hh:mm:ss 1: D.MM.YYYY h:mm:ss 2: D.MM.YYYY h:mm 3: D.MM.YY h:mm 4: YYYY-MM-DD- hh:mm:ss 5: YYYY-MM-DD- hh:mm 6: YYYY-MM-DD h:mm 7: YY-MM-DD- h:mm 8: DD.MM.YYYY 9: D.MM.YYYY 10: D.MM.YY 11: YYYY-MM-DD 12: YY-MM-DD 13: hh:mm:ss 14: h:mm:ss 15: h:mm
CM	9055	

Error recognition:

Marker	Value	Meaning
M4203	0	String was generated
	1	Incorrect transfer value

MP7235 Time difference to Universal Time (Greenwich Mean Time)

Input: –23 to +23 [hours]
0: Universal Time (Greenwich Mean Time)
1: Central European Time (CET)
2: Central European daylight-saving time

6.13.32 Log

The log serves as a troubleshooting aid. There are 2 MB of memory available for this purpose.

All entries in the log are marked with the current date and time.

Entry		Description
RESET		Powering up the control
BERR		Blinking error message
BREG		Register contents with a blinking error message
ERR		Error message P: PLC error message with the line number in the PLC error text file N: NC error message with number
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 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
		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.24/V.11) Byte 2/3 xx xx Internal error class Byte 4...7 xx xx xx xx Internal error code
INFO WARNING ERROR	PLC <log identifier>		Entries through PLC Modules 9275 and 9276
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 begin recording. After that, an ASCII file is generated and opened with the log entries.
- The PC software 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          09:18:19 Mon Oct 09 2000
  TNC 426/430 M
Info:    MAIN   START          09:18:19 Mon Oct 09 2000
  NC SOFTWARE = 280476 10
Error:                         09:18:29 Mon Oct 09 2000
  N-1 Power interruption
Key: 0x01AE -> CE              09:41:37 Mon Oct 09 2000
Error:                         09:41:40 Mon Oct 09 2000
  P88 88 MPs being read
Key: 0x01F0 -> NC Start        09:41:46 Mon Oct 09 2000
Error:                         09:41:46 Mon Oct 09 2000
  P93 93 Feed rate override Poti = 0 !
Key: 0x01F0 -> NC Start        09:41:48 Mon Oct 09 2000
Stib: ON                       09:41:57 Mon Oct 09 2000
Error:                         09:41:57 Mon Oct 09 2000
  P93 93 Feed rate override Poti = 0 !
Info:    MAIN   PATH           09:41:57 Mon Oct 09 2000
  RUNTAB =
Stib: OFF                      09:41:57 Mon Oct 09 2000
Info:    MAIN   PGMEND         09:41:57 Mon Oct 09 2000
  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           09:41:57 Mon Oct 09 2000
  RUNBRKPGM = TNC:\STEFAN\NC\TEST.H
Error:                         09:41:58 Mon Oct 09 2000
  N56 Limit switch X+
Key: 0x01EC -> Screen Change   09:42:08 Mon Oct 09 2000
```

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 it 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 recognition:

Marker	Value	Meaning
M4203	0	Entry was written
	1	Error code in W1022
W1022	1	Invalid priority
	2	Invalid string number or invalid immediate string
	12	No string end identifier
	20	Module was not called in a spawn 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 recognition:

Marker	Value	Meaning
M4203	0	Entry was written
	1	Error code in W1022
W1022	1	Invalid priority
	2	Invalid identifier for operand name
	3	Invalid first operand address
	4	Sum of first operand address and number of operands invalid
	5	Address is not a word/double-word address
	12	No string end identifier
	20	Module was not called in a spawn job or submit job
	36	Entry in the log was shortened to 210 characters

6.14 Keystroke Simulation

HEIDENHAIN contouring controls have two control panels:

- TNC keyboard unit
- The machine operating panel from the machine tool builder

The control panels are connected with the logic unit at connections X45 and X46.

The key code of the TNC keyboard unit is evaluated directly by the NC.

PLC inputs and outputs for the machine control panel are available on connector X46. The PLC must evaluate these PLC inputs and outputs and set the appropriate markers.

6.14.1 TNC Keyboard Unit

The key code of the TNC keyboard unit is evaluated by the NC. The key code is displayed in W274 while a key is being pressed. (See "Codes for keystroke simulation” on page 6 – 342).

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 recognition:

Marker	Value	Meaning
M4203	0	NC key was simulated
	1	Error code in W1022
W1022	1	Transferred parameter > maximum value
	2	Transferred parameter invalid
	22	Keystroke queue overflow

Module 9181 Disable individual NC keys

With this module you can disable individual NC keys.

If you press a disabled key, marker M4577 is set.

Call:

PS	B/W/D/K	<Key code>
CM	9181	
PL	B/W/D	<Error status> 0: NC key disabled -1: For error see W1022

Error recognition:

Marker	Value	Meaning
M4203	0	NC key was disabled
	1	Error code in W1022
W1022	1	Transferred parameter > maximum value
	2	Transferred parameter invalid

Module 9182 Re-enabling individual NC keys

With this module you cancel the effect of Module 9181.

Call:
PS B/W/D/K <Key code>
CM 9182
PL B/W/D <Error status>
0: NC key enabled
-1: For error see W1022

Error recognition:

Marker	Value	Meaning
M4203	0	Disabling was cancelled
	1	Error code in W1022
W1022	1	Transferred parameter > maximum value
	2	Transferred parameter invalid

Module 9183 Disabling groups of NC keys

The key-group codes are:

- 0: All keys
- 1: ASCII
- 2: 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 recognition:

Marker	Value	Meaning
M4203	0	The group of NC keys was disabled
	1	Error code in W1022
W1022	2	Transferred parameter invalid

Module 9184 Re-enabling groups of NC keys

With this module you cancel the effect of Module 9183.

Call:
PS B/W/D/K <Key group code>
CM 9184
PL B/W/D <Error status>
0: Group of NC keys enabled
–1: Transferred value> maximum value

Error recognition:

Marker	Value	Meaning
M4203	0	Disabling was cancelled
	1	Error code in W1022
W1022	2	Transferred parameter invalid

Module 9186 Call a soft-key function

With this module you can call certain soft-key functions in the machine operating modes.

Do not call a new function until the previous one is completed. You can interrogate this condition with Module 9187.

For a soft-key function to be simulated it must be displayed either in the foreground or background operating mode. Otherwise 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 recognition:

Marker	Value	Meaning
M4203	0	Soft-key function was called
	1	Error code in W1022
W1022	1	Parameter out of value range
	28	Previous call not ended

Module 9187 Status of a soft-key function call

Immediately after Module 9186 is called, the status 1= (soft-key function not yet completed) is set — regardless of whether the function can be run in the current operating mode. Module 9186 cannot be called again until status 0 or 2 is set. The error status 2 is erased if Module 9186 is called or if power is switched on.

Call:
CM 9187
PL B/W/D <Status>
0: Soft-key function executed or none called
1: Soft-key function not yet executed
2: Error: Soft-key function cannot be executed 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
$3A	:	ASCII

Code	Key	Group
$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
$57	W	ASCII
$58	X	ASCII

Code	Key	Group
$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
$1C5	HANDWHEEL	Operating mode
$1C6	TEST	Operating mode

Code	Key	Group
$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	

6.14.2 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
M4562	**Memory function for axis direction keys (MP7680 bit 0 = 1)**	PLC	PLC
M4560	**NC stop (0: Stop)**	PLC	PLC
M4561	**Rapid traverse**	PLC	PLC
M4564	**NC start**	PLC	PLC
M4230	**NC start via LSV2**	NC	NC
M4231	**NC stop via LSV2**	NC	NC

6.15 Touch Probe

The following touch probes can be connected:

- Touch trigger probes
  - TS 120, TS 220: With cable connection for digitizing, tool setup and measuring during machining
  - TS 632: With infrared transmission for workpiece setup and measurement during machining
  - TT 130: For tool measurement
- Measuring touch probe

For connecting the touch probes: (See "Mounting and Electrical Installation" on page 3 – 5).

- With MP6010 and MP6200, specify which touch probes are connected.
- Make sure that the spindle is locked during the measuring process.

With FN18 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

MP6200 **Selection of triggering or measuring touch probe (only with "digitizing with measuring touch probe" option)**
Input: 0: Triggering touch probe (e.g. TS 220)
1: Measuring touch probe

Note

The TNC 426/430 always emits a start signal when beginning a touch probe cycle, meaning Module 9135 does 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

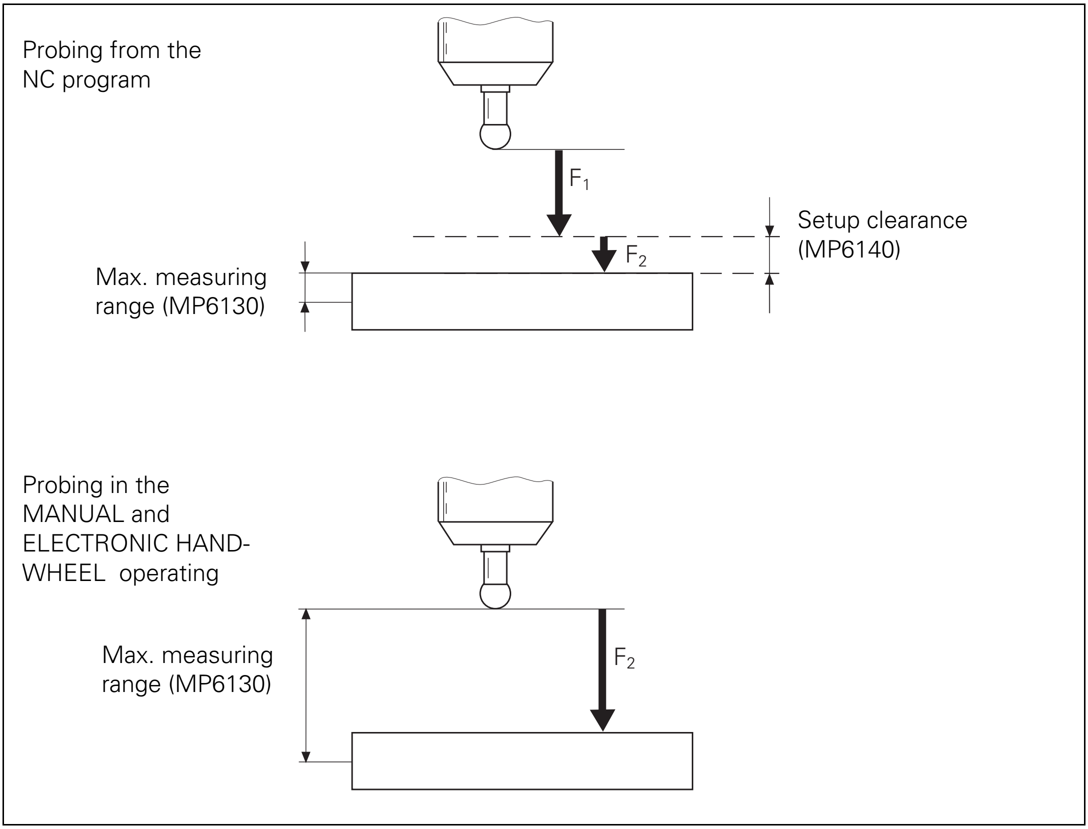
Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error in module run

6.15.1 Touch Probe Cycles

The probing cycles are available in the MANUAL and ELECTRONIC HANDWHEEL modes and in the NC program (see User's Manual, Touch Probe Cycles).

- 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
MP6361 for measuring touch probe

F2 = probing feed rate:
MP6120 for triggering touch probe
MP6360 for measuring touch probe

If the maximum measuring range (MP6130) is exceeded, the error message **TOUCH POINT INACCESSIBLE** appears.

In the MANUAL and ELECTRONIC HANDWHEEL modes, MP6140, MP6150, MP6361 have no meaning.

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 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 or MP6361. 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 operating modes MANUAL and ELECTRONIC HANDWHEEL, enter the tool number in the menu for touch probe calibration.

The TNC 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.

As of NC software 280 476-01, several blocks of touch probe data can be managed through the tool table. 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 1 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.

		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 ended 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

Orient when MP6163 < (current spindle angle – MP6162)

MP6120 **Probing feed rate (triggering touch probe)**
Input: 1 to 3000 [mm/min]

MP6360 **Probing feed rate (measuring touch probe)**
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 (triggering touch probe)**
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

MP6361	**Rapid traverse in probing cycle (measuring touch probe)**
Input:	10 to 10 000 [mm/min]

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
```

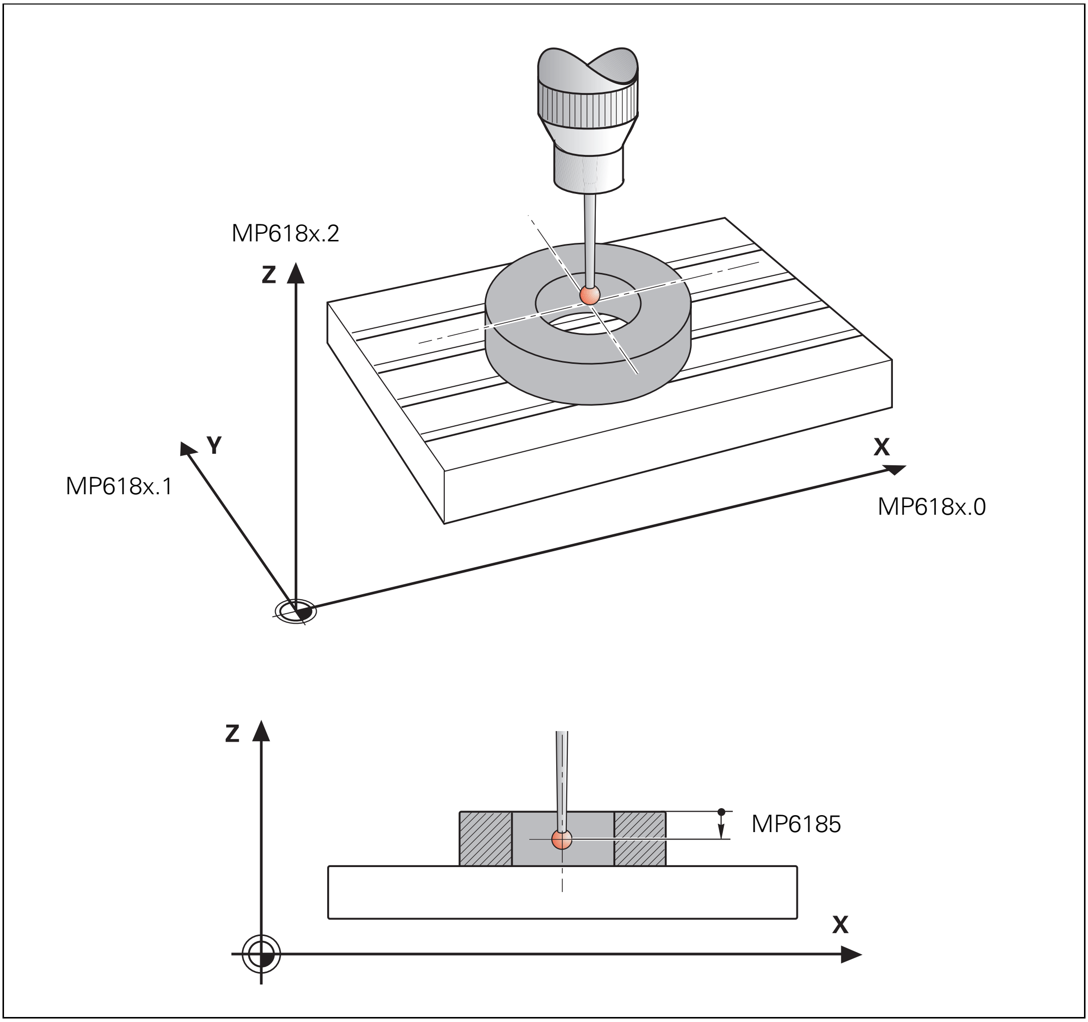
Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	2	Invalid axis number
	20	Module was not called in a spawn job or submit job

Calibration

To calibrate the touch probe from within the NC program:

- In MP618x.0 and MP618x.1.x, 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.
  If the spindle is oriented by the NC, you must reset M4012 (See "Oriented Spindle Stop" on page 6 – 201).
  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 modes, a triggering touch probe is rotated by pressing a soft key. A measuring touch probe is automatically rotated during calibration:

- With MP6321, select if the rotation should be automatic. M4017 is set for every spindle orientation.

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 TNC 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 TNC 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]

MP6321	**Measuring the center offset while calibrating the measuring touch probe**
Input:	0: Calibration with measurement of the center offset 1: Calibration without measuring the center offset

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]

6.15.2 Logging Measurements by 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 6 – 325).

File names of the individual print masks

- Calibration for length, triggering touch probe: TSLCAL.A
- Calibration for radius, triggering touch probe: TSRCAL.A
- Basic rotation: ROT_2PTS.A
- Point measuring: DAT_SURF.A
- Corner as datum: DAT_CORN.A
- Circle center as datum: DAT_CC.A
- Basic rotation over 2 holes: ROT_2HLS.A
- Datum over 4 holes: DAT_IS4H.A
- Circle center over 3 holes as datum: DAT_CC3H.A
- Calibration for length, measuring touch probe: TBLCAL.A
- Calibration of measuring touch probe: TMCAL.A

Format of the print masks

For the text lines of the print masks:

- Lines of text must be put into quotation marks.
- Each line must be concluded with a semicolon.
- Format instructions can be given from the programming language C.
- 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 of triggering touch probe
TS.LEN	Double	Calibrated probe length of triggering touch probe
TS.OFF1	Double	Calibrated center offset in reference axis of triggering touch probe
TS.OFF2	Double	Calibrated center offset in minor axis of triggering touch probe
TS.RINGRAD	Double	Calibration ring radius for triggering touch probe
TM.RAD	Double	Probe radius 1 of measuring touch probe
TM.RAD2	Double	Probe radius 2 of measuring touch probe
TM.LEN	Double	Calibrated probe length of measuring touch probe
TM.OFF1	Double	Calibrated center offset in ref. axis of meas. touch probe
TM.OFF2	Double	Calibrated center offset in minor axis of meas. touch probe
TM.CORSTAT[0]	Double	Calibrated stylus bending in X axis of meas. touch probe
TM.CORSTAT[1]	Double	Calibrated stylus bending in Y axis of meas. touch probe
TM.CORSTAT[2]	Double	Calibrated stylus bending in Z axis of meas. touch probe
TM.CORDYN[0]	Double	Force ratio in X/Z axis of measuring touch probe
TM.CORDYN[1]	Double	Force ratio in Y/Z axis of measuring touch probe

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	Main axis character

Datum at corner, circle, 4 holes, 3 holes on a circle		
Name	**Format type**	**Description**
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

```
"Calibration of measuring touch probe";
"___________________";
%02.2d-%02.2d-%4d:%02.2d:%02.2d "Time",DAY,MONTH,YEAR4,HOUR,MIN,SEC;
Probe axis:"%s",TA;
Probe radius 1: "%4.3lf" TM.RAD;
Probe radius 2: "%4.3lf",TM.RAD2;
Ring diameter: "%4.3lf",TM.RINGDIA;
Factors: X = "%4.4lf",TM.CORSTA[0];
Y = "%4.4lf",TM.CORSTA[1];
Z = "%4.4lf",TM.CORSTA[2];
Force ratio: FX/FZ = "%4.4lf",TM.CORDYN[0];
FY/FZ = "%4.4lf", TM.CORDYN[1]
```

6.15.3 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 ****************************";
```

6.15.4 Digitizing with the Touch Trigger Probe

For digitizing, HEIDENHAIN recommends the TS 220 touch trigger probe.

Touch probes with infrared transmission are not suitable for digitizing because the operating time is limited by the battery charge. Such systems with battery charge run in continuous operation for no more than eight hours.

Technical prerequisites

- Integrating the software module "Digitizing with TS." If the module is already installed, the following appears beneath the NC and PLC software numbers when you press the MOD key:
  %00000001 (see software option, Id. Nr. of the logic unit).
- Adapted TS 220 touch probe
- Optimization for operation with following error

The digitizing process is optimized by machine parameter:

▶ In MP6210, enter the probing rate during scanning of the model (number of oscillations in normal direction per second).

MP6210 depends on the dynamic behavior of the machine. The k_V factor affects the dynamic behavior (operation with following error): The higher the k_V factor, the greater the number of oscillations.

Together with the probe point interval PP.INT, MP6210 determines the maximum scanning feed rate of the "meander" and "contour line" probing cycles:

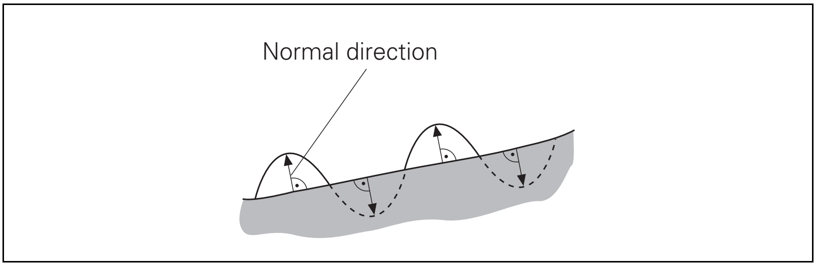
$$F_{probe}[\text{mm/min}] = \text{PP.INT [mm]} \cdot \text{Oscillations [1/s]} \cdot 60\ [\text{s/min}]$$

This results in the formula for the calculation of the value to be entered in MP6210. The optimized scanning feed rate depends on the feed rate in normal direction (MP6230).

$$\text{MP6210 [1/s]} = \frac{\text{optimized } F_{probe}[\text{mm/min}]}{\text{PP.INT [mm]} \cdot 60\ [\text{s/min}]}$$

▶ In MP6230, enter the feed rate in normal direction.

The feed rate in normal direction (MP6230) is the velocity with which the touch probe moves perpendicularly to the contour from the non-deflected to the deflected condition and vice versa.



The feed rate in normal direction determines the maximum oscillation amplitude and maximum scanning feed rate:

If MP6230 is too low, the machine's dynamic capabilities will not be fully exploited and the scanning feed rate will also be too low.

If MP6230 is too high, the oscillation amplitude is too large. In this case the stylus will visibly lift off from the contour and "tap" the surface of the workpiece, and the scanning feed rate will no longer increase.

▶ In MP6240, specify the maximum stylus deflection.

MP6240 depends on the length of the stylus being used.

MP6240 specifies the maximum travel by which the stylus retracts on inside corners. If the touch probe is not clear of the surface after the stylus backs off by the travel defined in MP6240, it will retract in the positive direction of the probe axis (e.g. Z). Digitizing continues as soon as the touch probe is clear. If the input value is too small, the touch probe may get caught up in an endless repetitive attempt to come free from an inside corner.

▶ With MP6260, select whether an M90 is appended to every NC block in the transmitted digitized data (See "Contouring Behavior" on page 6 – 152).

▶ With MP6270, specify the number of decimal places to which the coordinates are output.

MP6210	**Number of oscillations in normal direction per second**
Input:	0 to 65.535 [1/s]
MP6230	**Feed rate in normal direction**
Input:	0 to 1000 [mm/min]
MP6240	**Maximum deflection of the stylus**
Input:	0 to 10.000 [mm]
MP6260	**Output of M90 in NC blocks with digitized data**
Input:	0: No output of M90 1: Output of M90 in every NC block
MP6270	**Rounding of decimal places**
Input:	0: Output in 0.001-mm steps (1 µm) 1: Output in 0.01-mm steps (10 µm) 2: Output in 0.0001-mm steps (0.1 µm)

Scanning cycles

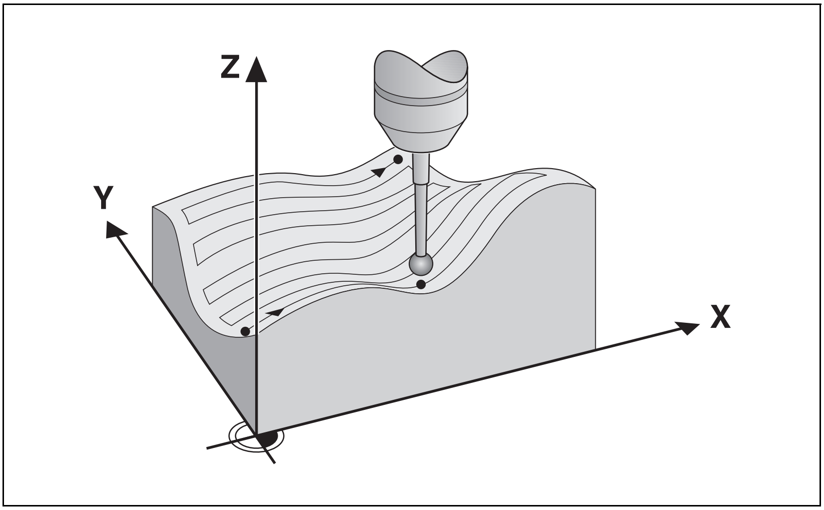
Its direct access to the position control loop of the TNC controller enables the touch probe to measure values very rapidly (3 to 5 values per second). This results in a scanning feed rate of 180 to 300 mm/min at a programmed probe point interval of 1 mm.

Three scanning cycles are available for digitizing:

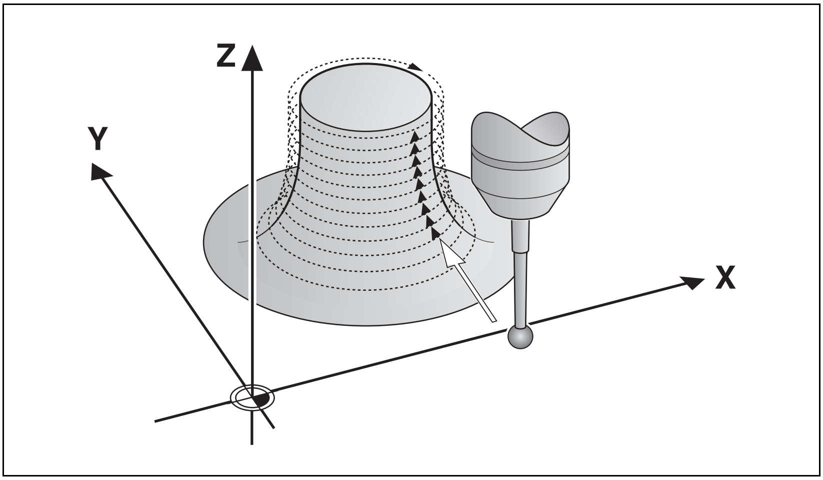
- Range
- Meander
- Contour lines

The "range" cycle defines the cuboid scanning range and the file in which the digitized data is stored. You can file the digitized data in the program memory of the control or on a PC.

The “meander” cycle digitizes a 3-D model in meanders (successive back-and-forth lines) in the predefined range.



The “contour lines” cycle digitizes a 3-D model level-by-level in contour lines within a predefined range. This level-by-level digitizing is used mainly for surfaces with steep edges.



Lubrication

During meander digitizing a very flat surface may cause little movement in the probe axis. This can result in a lack of lubrication in the probe axis:

▶ With MP6220 and MP6221, set the additional lubrication at the line end.

MP6220	**Traverse distance for lubrication of the touch probe axis at line end**
Input:	0.000 to 999.999 [mm]
MP6221	**Time after which the probe axis must be lubricated**
Input:	0 to 65 535 [mm]

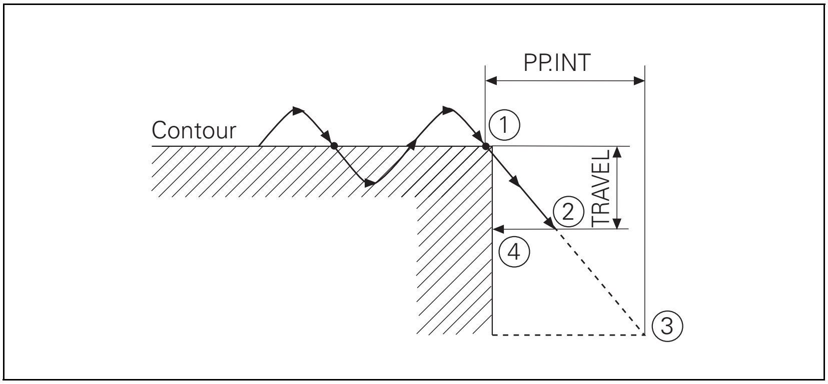
Scanning process at corners

The two parameters PP.INT (maximum probe point interval) and TRAVEL from the “meander” and “contour line” scanning cycles operate as limit values.

Depending on the values that are entered for these parameters (see the User’s Manual), either the travel of the probing stroke or the probe point interval is limited.

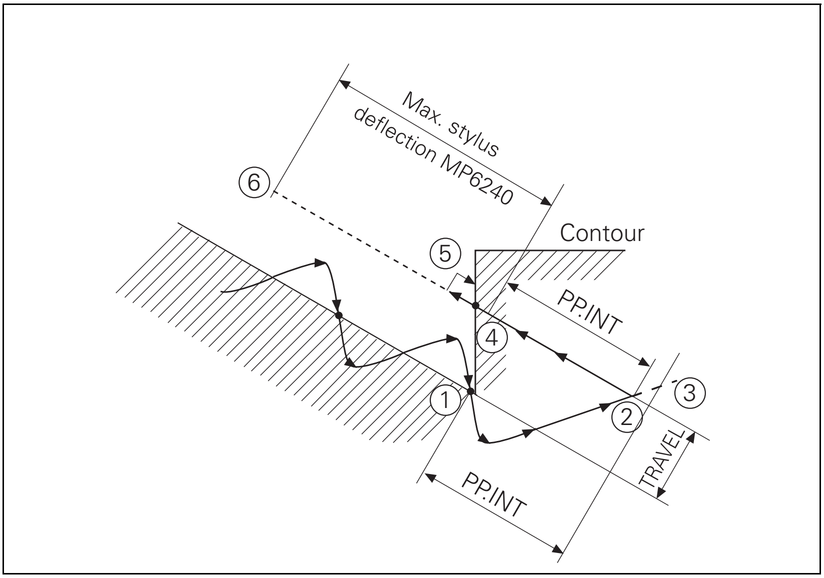
The travel determines the contour of the corners. The smaller the travel, the more exact the resolution of the corner. If too small a travel is defined, it may prevent the probe from getting clear at acute sharp inside corners (minimum travel 0.1 mm).

Outside corners



On outside corners, having probed the last point (1), the touch probe moves down the resultant straight line until it either makes contact again or reaches one of the two limits TRAVEL (2) or PP.INT (3). The illustration shows how the TRAVEL works as a limit and the touch probe returns to the contour (4) in the opposite direction. The new scanning direction is defined by the probed points (1) and (4).

Inside corners



On inside corners, having probed the last point (1), the touch probe continues to oscillate in the same scanning direction, but changes direction because it cannot get clear. It then moves down the resultant straight line until it either gets clear or reaches one of the limits TRAVEL (2) or PP.INT (3).

The touch probe moves in the reverse scanning direction to get clear again. If the programmed probe point interval PP.INT (4) is too small for the probe to clear, it travels in negative direction by up to the value of MP6240 (max. stylus deflection). As soon as the touch probe is clear it returns to the contour (5) in the reverse travel direction. The new scanning direction is defined by the probed points (1) and (5).

If, after it backs off by the travel defined in MP6240, the touch probe is not clear of the surface (6), it will retract in the positive direction of the probe axis (e.g. Z+). If the stylus is still deflected after it reaches the clearance height (See "Scanning cycles” on page 6 – 361), “Range” cycle), the scanning sequence is aborted and an error message is displayed.

Optimizing the scanning sequence

Preparation:

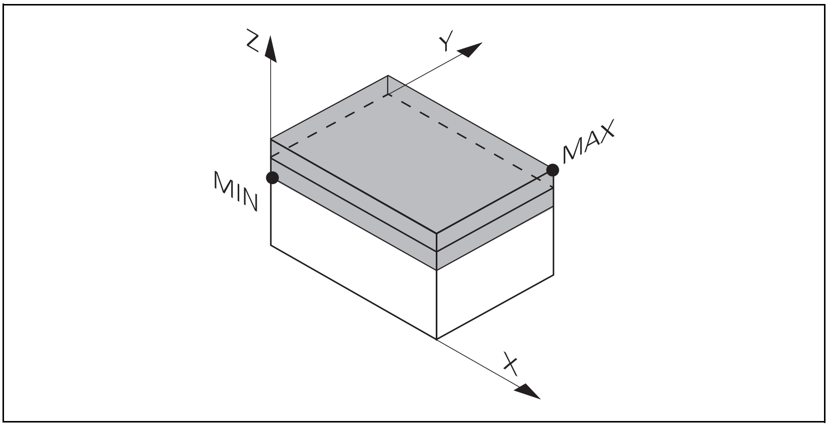
- Set up a flat workpiece with vertical side and smooth top in the machining plane (e.g. XY plane).
- Select the MANUAL or ELECTRONIC HANDWHEEL mode of operation.
- Probe the surface with the surface = “datum” function and enter +0 mm for the datum plane.
- For the following machine parameters select the default setting:
  - MP6210 = 5 [1/s] Oscillations in normal direction
  - MP6230 = 30 [mm/min] Feed rate in normal direction
  - MP6240 = 5 [mm] Maximum deflection of the stylus
- Enter an NC program with the scanning cycles “range” and “meander,” and specify the scanning direction X and the point spacing 1 mm.

Example:

```
0   BEGIN PGM OPTIDIGI MM
1   BLK FORM 0.1 Z X+0 Y+0 Z–10           ;REQUIRED FOR THE PARALLEL
2   BLK FORM 0.2 X+100 Y+100 Z+10         ;GRAPHICS OF THE TNC
3   TOOL DEF 1 L+0 R+4
4   TOOL CALL 1 Z S1000
5   TCH PROBE 5.0 RANGE                   ;DEFINITION OF THE
6   TCH PROBE 5.1 PGM NAME: DIGIDAT       ;DIGITIZED
7   TCH PROBE 5.2 Z X+= Y+= Z–10          ;RANGE INCLUDING THE
8   TCH PROBE 5.3 X+100 Y+100 Z+10        ;PROGRAM NAME FOR
9   TCH PROBE 5.4 HEIGHT: +25             ;THE SURFACE-DATA
                                          ; FILE AND FOR THE
                                          ; CLEARANCE HEIGHT
                                          ; (ABSOLUTE DIMENSION)
10 TCH PROBE 6.0 MEANDER                  ;"MEANDER-TYPE"
                                          ;SCANNING IN
11 TCH PROBE 6.1 DIRECTION: X             ;X DIRECTION PLUS
12 TCH PROBE 6.2 TRAVEL: 0.5 PP.INT:1 L.SPAC:1 ;THE POINT AND LINE
                                          ;SPACING AND THE TRAVEL
13 END PGM OPTIDIGI MM
```

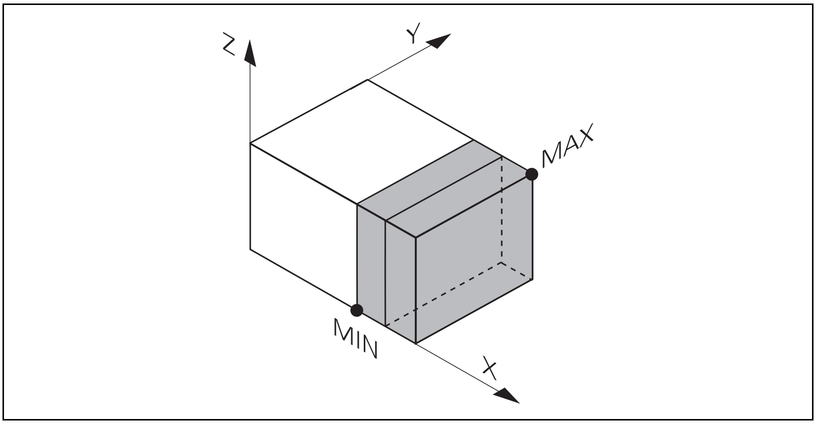
To optimize the X and Y axes:

- Select the range so that only the smooth top of the part is scanned.



To optimize the Z axis:

▶ Select the range so that the vertical side is scanned.



Procedure for optimizing the machine parameters

Procedure for optimizing the machine parameters:

▶ Record ACTL. SPEED with the internal oscilloscope.

▶ In the PROGRAM RUN, FULL SEQUENCE mode, select the OPTIDIGI program and press the external START key. The machine runs and the digitizing process begins.

▶ Increase the feed rate with the override potentiometer until the nominal X value has reached the maximum value and shows only minor voltage dips. Make a note of the maximum feed rate that is possible in the worst axis. (Axis should be just able to maintain a constant velocity.)

▶ Are the oscillations in normal direction already perceptible, i.e. does the ball tip lift off visibly from the workpiece surface?
- If not, then increase MP6230, e.g. by 10 mm/min. Return to the PROGRAM RUN, FULL SEQUENCE mode, select the OPTIDIGI program and press the machine START button...
- If so, then reduce MP6230 again, e.g. by 10 mm/min.

▶ Repeat the optimization process for the Y axis. Also, switch the NC to the scanning direction Y. The optimization process is oriented to the worst axis. The feed rate must not be increased any further with either the override potentiometer or the feed rate in normal direction (MP6230).

▶ Repeat the optimization process for the Z axis. Change the scanning range accordingly. The optimization process is oriented to the worst axis. The feed rate must not be increased any further with either the override potentiometer or the feed rate in normal direction (MP6230). Pay particular attention to the vertical side.

Calculation of possible oscillations in normal direction

$$\text{MP6210 [1/s]} = \frac{\text{optimized } F_{scan}\text{[mm/min]}}{\text{PP.INT [mm]} \cdot \text{60 [s/min]}}$$

▶ Enter the calculated value in MP6210.
This standardizes the feed rate override potentiometer to the “attained feed rate.”

6.15.5 Digitizing with the Measuring Trigger Probe

The measuring touch probe permits scanning speeds up to 3 m/min. The stylus deflection is measured in every direction directly by integral encoders and evaluated in the TNC.

Technical prerequisites

You need:

- An interfaced measuring touch probe
- A “digitizing with measuring touch probe” adapter kit

The machine must be prepared for the use of the measuring touch probe:

- Lock the spindle mechanically.
- Ensure that the spindle drive cannot be started while the probe is in use.

The “digitizing with measuring touch probe” adapter kit also includes the “digitizing with TS” function (triggering touch probe).

- With MP6200, select whether a measuring or triggering touch probe is used.

Warning

If you wish to use both the triggering and the measuring touch probes, you must make quite sure that the type of touch probe in the spindle at any given time is entered in MP6200.

Danger of breakage!

The counting direction of the encoder signals in the probe must match the counting direction of the encoder signals of the axes. (The axis-specific parameters are defined in MP210):

- Select the POSITIONING WITH MANUAL DATA INPUT mode of operation.
- Press the PNT soft key and position the machine by touching the stylus. The machine must move in the direction in which the stylus is deflected. If that does not happen: Change the counting direction in MP6320.
- With MP6322, assign the touch probe axes (the encoders in the probe) to the machine axes. For machines with swivel heads:
  Enter the respective mounting attitude of the touch probe in MP6322.

In a horizontal attitude the probe cannot make comparative probing measurements from opposite orientations. The measuring touch probe can be used only for digitizing in the horizontal attitude. It is not possible to align the workpiece in a horizontal attitude.

Warning

Be sure to enter the correct value in MP6322 for the attitude of the touch probe. Otherwise the calculation of the maximum deflection from MP6330 may by incorrect.

Danger of breakage!

If the stylus is deflected by a distance larger than that defined in MP6330, the blinking error message **`Stylus deflection exceeds max.:`** appears.

▶ In MP6310, select the mean deflection depth during digitizing.
- For standard parts: Input value = 1 mm
- On parts with steep sides that are scanned at high speed:
  Input value > 1 mm

With the deflection depth, the probing force of the touch probe can be optimized.

After starting the "meander" or "contour lines" cycle, the probe moves at the feed rate defined in MP6361 to the clearance height, and then in the working plane to the point above the starting point. It then moves at the reduced feed rate defined in MP6350 to the MIN point or to the first touch point:

▶ In MP6361, enter a feed rate for rapid traverse in the probing cycle.
▶ In MP6350, enter a feed rate for positioning to the MIN point and probing the contour.

▶ With MP6362, select whether the scanning feed rate is automatically reduced if the ball tip deviates from the path.

During scanning of contour lines, the probe sometimes ends the contour line at a point located near but not exactly at the starting point:

▶ In MP6390, define a target window within which the probe is considered to have returned to the starting point. The target window is a square.
Input value: Half the edge length of the square

MP6360 (probing feed rate) and MP6361 (rapid traverse in the probing cycle) are effective in the standard probe cycles.

MP6200	**Selection of triggering or measuring touch probe (only with "digitizing with measuring touch probe" option)**
Input:	0: Triggering touch probe (e.g. TS 220) 1: Measuring touch probe
MP6310	**Deflection depth of the stylus (measuring touch probe)**
Input:	0.1000 to 2.0000 [mm]
MP6320	**Counting direction of encoder output signals (measuring touch probe)**
Format:	%xxx
Input:	Bits 0 to 2 represent axes X to Z 0: Positive 1: Negative

MP6321 **Measuring the center offset while calibrating the measuring touch probe**

Input: 0: Calibration with measurement of the center offset
1: Calibration without measuring the center offset

MP6322.0-2 **Assignment of the touch probe axes to the machines axes X, Y and Z**

Input: 0: Touch probe axis X
1: Touch probe axis Y
2: Touch probe axis Z

MP6330 **Maximum deflection of the stylus (measuring touch probe)**

Input: 0.1 to 4.000 [mm]

MP6350 **Feed rate for positioning to the MIN point and approaching the contour (measuring touch probe)**

Input: 1 to 3000 [mm/min]

MP6360 **Probing feed rate (measuring touch probe)**

Input: 1 to 3000 [mm/min]

MP6361 **Rapid traverse in probing cycle (measuring touch probe)**

Input: 10 to 20 000 [mm/min]

MP6362 **Feed rate reduction, if the stylus of the measuring touch probe is deflected to the side**

Input: 0: Feed rate reduction not active
1: Feed rate reduction active

MP6370 **Radial acceleration when digitizing with measuring touch probe**

Input: 0.001 to 3.000 [m/s]
Recommended input value: 0.1

MP6390 **Target window for contour line**

Input: 0.1000 to 4.0000 [mm]

6.15.6 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 User's Manual).

Technical prerequisites

You need:

- TT 130
- Central tool file TOOL.T must be active (via machine parameter)

The TNC 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. (See "Oriented Spindle Stop" on page 6 – 201)
  - 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, define a safety zone around the probe contact of the TT 130.

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, 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 is enough 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, then set bit 9 = 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 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 TNC 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 = measuring tolerance · 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.

$$\text{measuring tolerance} = \frac{r}{5\ [\text{mm}]} \cdot \text{MP6510.}$$

r: Tool radius [mm]
MP6510.0: Max. permissible measuring error [mm]
v = measuring tolerance · n
n: Speed [rpm]

$$v = \frac{\text{MP6570} \cdot \text{MP6510}}{2 \cdot \pi \cdot 10^{-3}}$$

v: Probing feed rate [m/min]
MP6570: Maximum permissible surface speed of the tool edge [m/min]

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.

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
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

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]

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, 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, an error message is displayed.

MP6585 **Monitoring the position of the rotary and additional linear axes during the tool measurement cycles**
As of software version:280 476-01
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**
As of software version:280 476-01
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
MP6586.0–5 Axes A to W

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 breaking 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

6.16 Electronic Handwheel

The following handwheels can be connected with HEIDENHAIN contouring controls (See "Mounting and Electrical Installation" on page 3 – 5):

- One panel-mounted HR 130 handwheel, or
- Three HR 150 panel-mounted handwheels via the HRA 110 handwheel adapter
- 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 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	6 000
2	5	3 000
3	2.5	1 500
4	1.25	750
5	0.625	80
6	0.312	80
7	0.156	80
8	0.078	80
9	0.039	80
10	0.019	80

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 TNC keyboard 1: Through PLC Module 9036
MP7650	**Counting direction for handwheel**
As of software version:only before 280 474-07	
Input:	0: Negative counting direction 1: Positive counting direction
MP7650	**Handwheel counting direction for each axis**
Format:	%xxxxxxxxx
Input:	Bits 0 to 8 correspond to 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 recognition:

Marker	Value	Meaning
M4203	0	Status information was written
	1	Error code in W1022
W1022	1	Transferred value out of range
	2	Incorrect number of the status information
	6	Input disabled

6.16.1 HR 130 Panel-Mounted Handwheel

▶ Enter MP7640 = 2 (HR 130)

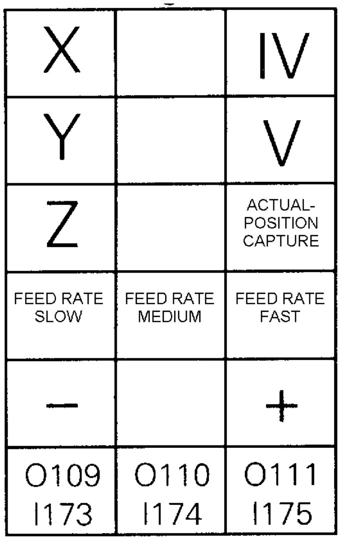
When the axis keys are pressed, the associated cursor and the handwheel symbol are displayed simultaneously.

6.16.2 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 the 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 the PLC: MP7645.0 = 1

O96 I160		O97 I161
O98 I162		O99 I163
O100 I164		O103 I167
O104 I168	O105 I169	O106 I170
I171		I172
O109 I173	O110 I174	O111 I175

All keys are evaluated by the PLC.

Module 9036 sets the handwheel axis and handwheel 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)

6.16.3 HR 150 Panel-Mounted Handwheels with HRA 110 Handwheel Adapter

▶ Enter MP7640 = 5 (HR 150 via HRA 110)

For selecting the interpolation factor you can use the switch S1 (see Mounting and Electrical Installation). 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 third handwheel if MP7645.2 = 1		
Input:	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

6.17 PLC Inputs/Outputs

The logic unit 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 4xxB (See "Mounting and Electrical Installation" on page 3 – 5).

	Logic unit				**PLC input/output unit**	
	X9	**X41**	**X42**	**X48**	**PL 410B**	**PL 405B**
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	13[a])	–	–	–	–	–
Control-is-ready output	–	2	–	–	1	1
Control-is-ready input	–	–	2	–	–	–

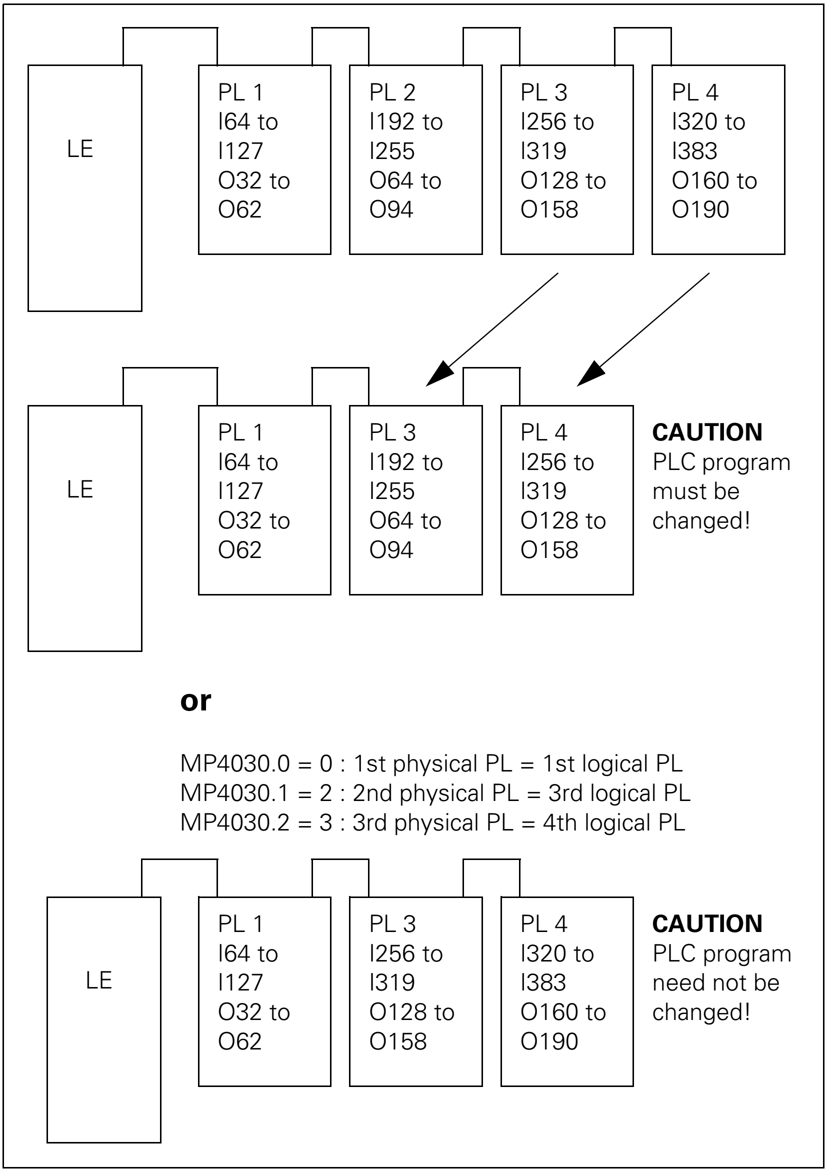
a. You need one analog output for each analog axis.

To interrogate and set the inputs and outputs of the PLC I/O unit you need PLC modules.

PL assignment

Up to four PL 4xxB PLC I/O units can be connected. The first PL is connected to the LE, 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 LE).

Example:



MP4030 Assignment of physical to logical PL

As of software version:280 476-01

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 on the PL 4xxB. 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 LE
                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 recognition:

Marker	Value	Meaning
M4203	0	Diagnostic information was read
	1	Error code in W1022
W1022	1	Invalid code
	2	Invalid PL
	24	Module was called in a spawn job or submit job

6.17.1 24 Vdc Switching Input/Outputs

The current conditions of the switching inputs and outputs are available for you in PLC addresses
(See "Mounting and Electrical Installation” on page 3 – 5).

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

As of NC software 280 476-03, before the PLC program is converted, the PLC outputs are reset.
As of NC software 280 476-03, the memories of the PLC outputs are also reset.
As of NC software 280 476-09, an attempt is made to reset the PLC outputs when a power fail occurs.

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. (See "Mounting and Electrical Installation” on page 3 – 5).
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 recognition:

Marker	Value	Meaning
M4203	0	Inputs were read
	1	Error code in W1022
W1022	2	Invalid PL no.
	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. (See "Mounting and Electrical Installation" on page 3 – 5).
The memory contents remain unchanged until you call this module or Module 9002. As of NC software 280 474-xx, the module recognizes whether a PLC input/output unit is 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..0.31>
PS	D/K	<Bits 0...31 = inputs 32...63>
CM	9008	

Error recognition:

Marker	Value	Meaning
M4203	0	Inputs were read
	1	Error code in W1022
W1022	2	Invalid PL number (as of 280 474-xx: PL not connected or invalid PL no.)
	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 (See "Mounting and Electrical Installation" on page 3 – 5). 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 recognition:

Marker	Value	Meaning
M4203	0	Outputs were set
	1	Error code in W1022
W1022	2	Invalid PL no.
	24	Module was called in a spawn job or submit job

Module 9009 Update certain 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 (See "Mounting and Electrical Installation" on page 3 – 5). 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. As of NC software 280 474-xx, the module recognizes whether a PLC input/output unit is connected.

The program can be called only in the cyclic PLC program.

Call:

```
PS    B/W/D/K   <Number of the PL>
                0: First PLC input/output unit
                1: Second PLC input/output unit
                2: Third PLC input/output unit
                3: Fourth PLC input/output unit
PS    D/K       <Bit 0...31 = outputs 0...31>
CM    9009
```

Error recognition:

Marker	Value	Meaning
M4203	0	Outputs were set
	1	Error code in W1022
W1022	2	Invalid PL number (as of 280 474-xx: PL not connected or invalid PL no.)
	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 recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Invalid transfer parameter

6.17.2 Analog Inputs

Socket X48 of the logic unit provides ±10 Vdc analog inputs and analog inputs for connecting Pt 100 thermistors. (See “Analog Input” on page 3 – 47)

The PLC input/output unit is available in a version with additional analog inputs. (See “Overview of Components” on page 2 – 5)

The temperatures measured by the Pt100 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

Do not call the module as long as Modules 9005 or 9009 are active through a submit job.

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 be called only 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° Nr. 64 to 69: Natural number with the unit 0.01 V or 0.1°

Error recognition:

Marker	Value	Meaning
M4203	0	Input was read
	1	Error code in W1022
W1022	2	Invalid PL number or invalid analog input number
	24	Module was called in a spawn job or submit job

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

6.17.3 Analog Outputs

You can drive analog outputs 1 to 13 at sockets X8 and X9 (See "Mounting and Electrical Installation” on page 3 – 5).

Note

Every analog axis or analog spindle needs an analog output. These outputs are no longer available to the PLC.

Module 9130 Output of an analog voltage
With this module you place an analog voltage on an analog output. The voltage is output with a slight delay after the end of the PLC scan.

Call the module only once for each output per PLC scan!

Format: 1 mV
Voltages greater than +10 V or less than –10 V are limited to the respective maximum value.

Call:
```
PS   B/W/D/K   <Number of the analog output>
               1 to 6: Analog outputs 1 to 6 (X8)
               7 to 13: Analog outputs 7 to 13 (X9)
PS   B/W/D/K   <Analog voltage in mV>
CM   9130
```

Error recognition:

Marker	Value	Meaning
M4203	0	Analog voltage was output
	1	Error code in W1022
W1022	1	Invalid analog output
	2	Disabled analog output

6.18 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 maker M4579 you can interrogate the current state.

With Module 9036 you can limit the jog increment.

You can ascertain the current increment jog 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

NC software 280 470-xx

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 the 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 the incremental jog positioning**	PLC	PLC
W1050	**Incremental jog positioning in positive direction** Bits 0 to 8 correspond to axes 1 to 9 0: Not active 1: Active	PLC	PLC

		Set	Reset
W1052	**Incremental jog positioning in negative direction** Bits 0 to 8 correspond to 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 the jog increment limitation and activation of the jog increment entered last –2: Cancellation of the jog increment limitation and activation of the minimum from jog increment entered last and 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 out of range 3: Input disabled

Error recognition:

Marker	Value	Meaning
M4203	0	Status information was written
	1	Error code in W1022
W1022	1	Transferred value out of range
	2	Incorrect number of the status information
	6	Input disabled

Module 9035 Reading status information

Call:

PS B/W/D/K <26>
CM 9035
PL B/W/D <Jog increment>

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	Error code in W1022
W1022	1	Status information invalid
	20	Call was not in a submit or spawn job

6.19 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 according to the grid spacing from MP430:

- 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.

MP420.0-8 **Hirth coupling for axes 1 to 9**
Input: 0: No Hirth coupling
1: Hirth coupling

MP430.0-8 **Prescribed increment for Hirth coupling**
Input: 0.0000 to 30.0000 [°]

6.20 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 geometry via MP7510 and following (See "Tilting Axes" on page 6 – 52) and 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., datum shift is active
New actual value display X = +70 (the old datum receives the value 20).

		Set	Reset
D528	**Datum shift for axis 1**	PLC	PLC
D532	**Datum shift for axis 2**	PLC	PLC
D536	**Datum shift for axis 3**	PLC	PLC
D540	**Datum shift for axis 4**	PLC	PLC
D544	**Datum shift for axis 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 <Axis [0 to 8]>
PS B/W/D/K <Shift [0.1 µm]>
CM 9230

Error recognition:

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

6.21 Tool Changer

You control the tool changer through PLC outputs.

If the tool changer is operated with controlled axes, then use the PLC axes (See "PLC Axes" on page 6 – 28). 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.

6.21.1 Tool Table, Pocket Table

You can edit the tool table in the PROGRAM RUN mode of operation:

- Ensure that the tool table and pocket table are neither locked nor protected with MP7224 (See "Files" on page 6 – 300)
- 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.

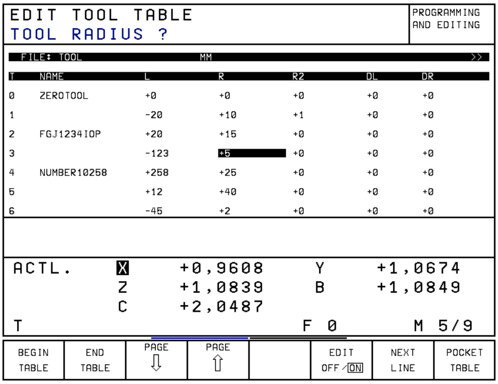
The current tool table is TOOL.T, the pocket table is TOOL_P.TCH. Both files are saved in the root directory TNC:\ .

In the PROGRAMMING AND EDITING mode of operation you can read the tool table in and out through the data interface:

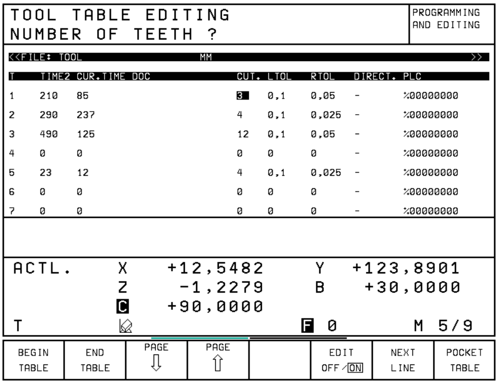
- Press the PGM MGT soft key. The pocket table is always transferred along with the tool table.

On the external storage medium the tool table has the identifier T, and the pocket table has the identifier R.

▶ In MP7266.x, specify the fields of the tool table that are to be displayed and the sequence in which they appear.



Left side of the tool table



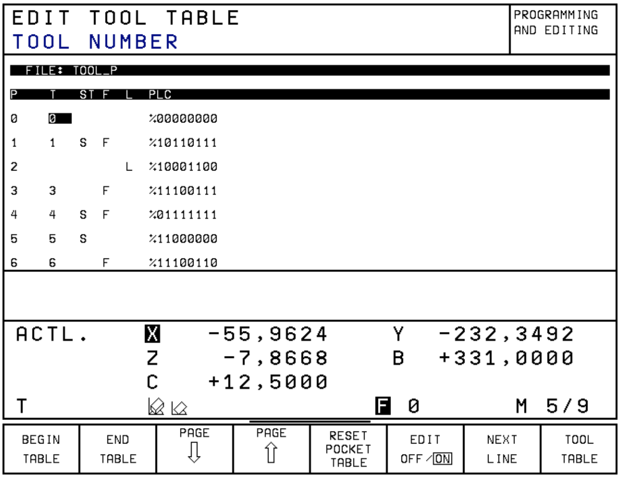
Right side of the tool table

Note

The complete width of the tool table cannot exceed 250 characters. Wider tables cannot be transmitted through the data interface.

For the width of the individual columns, see MP7266.x.

▶ In MP7267.x, specify the fields of the pocket table that are to be displayed and the sequence in which they appear.



Example pocket table

- Ensure that the tool table and pocket table are neither locked nor protected with MP7224.x (See "Files" on page 6 – 300)
- In MP7260, specify the number of the tools in the tool table.
  - If MP7260 = 0, no tool table is used (TOOL.T will 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 6 – 411).
  - 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.

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 **Hiding/showing the POCKET TABLE soft key**
Format: %x
Input: Bit 0 –
0: POCKET TABLE soft key is shown
1: POCKET TABLE soft key is hidden

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

MP	Meaning	Column name	Column width
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
MP7266.24[a]	Tool type (MILL=cutter/ DRILL=drill)	TYPE	5
MP7266.25[a]	Tool material	TMAT	16
MP7266.26[a]	Cutting data table	CDT	16
MP7266.27[b]	PLC value	PLC-VAL	11
MP7266.28[c]	Center misalignment in reference axis	CAL-OF1	11
MP7266.29[c]	Center misalignment in minor axis	CAL-OF2	11
MP7266.30[c]	Spindle angle during calibration	CAL-ANG	8

a. As of NC software 280 472-01
b. As of NC software 280 474-05
c. As of NC software 280 476-01

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)

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)

.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: Pocket locked (L); (0: No, 1: Yes)
4: PLC status (PLC)

PS B/W/D/K <Line number for beginning of search>
CM 9092
PL B/W/D <Line number (in case of error –1)>
PL B/W/D <Error number>
0: No error. Element was found.
1: Call was not in a submit or spawn job
2: File type does not exist
3: No file of the entered type was found with M status
4: Line number not in file
5: Incorrect element number
6: Element value not found

Error recognition:

Marker	Value	Meaning
M4203	0	No error
	1	For error, see above

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 recognition:

Marker	Value	Meaning
M4203	0	No error
	1	For error, see above

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 recognition:

Marker	Value	Meaning
M4203	0	No error
	1	For error, see above

Module 9096 Erasing 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 recognition:

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

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 recognition:

Marker	Value	Meaning
M4203	0	Pocket table locked/released
	1	Error code in W1022
W1022	2	Invalid parameter for locking/releasing the pocket table
	6	Pocket table was already locked/released
	20	Module was not called in a spawn 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 recognition:

Marker	Value	Meaning
M4203	0	Pocket has been exchanged
	1	Error code in W1022
W1022	2	Invalid parameter
	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., first tool magazine 1 with tool 1 to <MP7261.0>. 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 recognition:

Marker	Value	Meaning
M4203	0	Number of the entry was found
	1	Error code in W1022
W1022	1	Invalid tool magazine number
	2	Invalid pocket number
	20	Module was not called in a spawn job or submit job

Module 9302 Search for a free pocket in the tool magazine

Module 9302 searches for a free 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 free pocket>
–1: No vacant pocket available

Error recognition:

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 remain 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 recognition:

Marker	Value	Meaning
M4203	0	Pocket has been exchanged
	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
	21	Module was called during an NC program run
	30	No valid tool in the original pocket
	36	Error in file handling

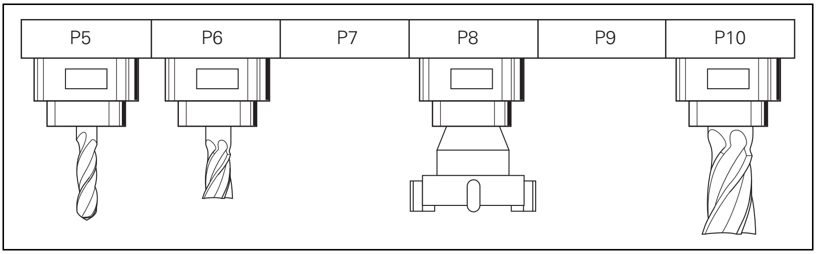
Special tools

In the pocket table:

- In the "ST" field, 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).
- Lock the pockets to be kept free with the "L" field.
- 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 "F" field (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.

TOOL-CALL key:

- 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 7 – 44).

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, DL) can be entered for each tool in the tool table. These delta values are taken into account by the TNC.

If the radius of the replacement tool differs from the radius of the original tool, you must define this in the “DR” field. 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, FULL SEQUENCE and PROGRAM RUN, SINGLE BLOCK modes if the following conditions are fulfilled:

- Spindle ON
- No F MAX
- F enable
- Control-in-operation symbol is on

After program interruption with "internal stop," M02, M30 or END PGM, the tool age counter is stopped.

The tool age counter does not run in the MANUAL OPERATION, ELECTRONIC HANDWHEEL, and POSITIONING WITH MDI operating modes.

The user can reset the current tool age by entering zero.

		Set	Reset
M4543	**Tool life 1 expired (TIME1 in the tool table)**	NC	NC/PLC
M4546	**Tool life 2 expired (TIME2 in the tool table)**	NC	NC/PLC

MP7680	**Machine parameter with multiple function**
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 recognition:

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

6.21.2 Automatic Calculation of Cutting Data

As of NC software: 280 472-01

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 TYP column define the type of tool:
  - DRILL = drilling tool
  - TAP = tapping tool
  - MILL = milling cutter
- Enter the following values in the table:
  - Tool radius R
  - Tool material TMAT
  - For cutter: Number of teeth CUT

The tool types are defined in the file PLC:\TTYP.TAB.

If you edit this file, you must enter the new name and path in the system file OEM.SYS using the command TTYP=.

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 TNC (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 into 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 into 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 the short description for the material (e.g. HSS).
- Enter additional information on the material in DOC column.

The tool material is defined by choice of the tool type in the tool table (See "Tool Table, Pocket Table” on page 6 – 400). You can edit the associated material table:

- Go into the tool table and press the SELECT WORKPIECE MATERIAL soft key.

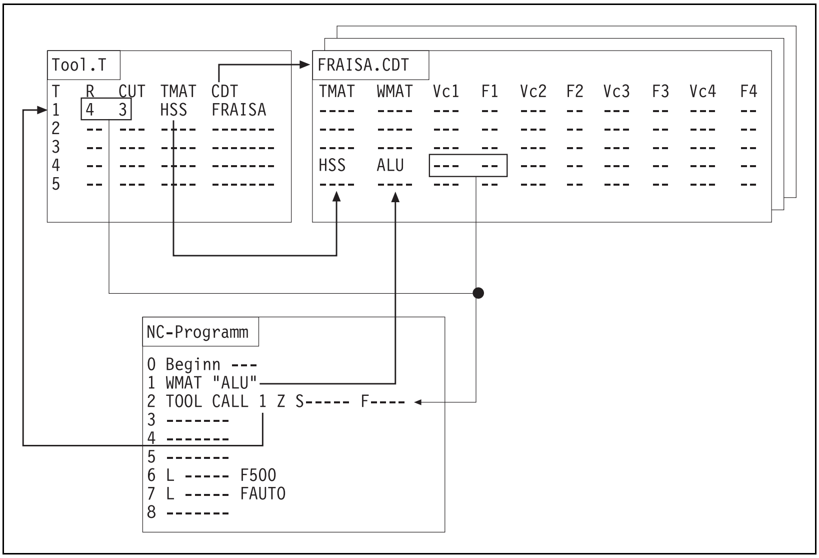
Calculation of cutting data

▶ 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.

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



6.21.3 Automatic Tool Recognition

Automatic tool identification is possible with the Balluff tool identification system (BIS).

Please contact HEIDENHAIN for further information.

6.21.4 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 command 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 7 – 44).
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 (See "PLC Programming” on page 7 – 3).

▶ In the called program, enter a TOOL CALL so that the tool data become active and a T strobe is transferred to the PLC.

With FN17 you can overwrite the software limit switch for the tool-change position. If you use FN18 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 (M112).

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 "PLC Programming" on page 7 – 3).

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. (See "Returning to the Contour" on page 6 – 294). If no NC program is specified in the NCMACRO.SYS file, the TOOL CALL is executed as before.

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 TCALL MM
1 M112 T4 ; INSERT ROUNDING TO POSITION CONTINUOUSLY
2 FN18: SYSREAD Q1 = ID60 NR1 IDX0 ; TOOL NUMBER
3 FN18: SYSREAD Q2 = ID60 NR2 IDX0 ; TOOL AXIS
4 FN18: SYSREAD Q3 = ID60 NR3 IDX0 ; SPEED
5 FN18: SYSREAD Q4 = ID60 NR4 IDX0 ; OVERSIZE IN TOOL LENGTH DL
6 FN18: SYSREAD Q5 = ID60 NR5 IDX0 ; OVERSIZE IN TOOL RADIUS DR
7 FN19: PLC=+Q / +0 ; INFO FOR PLC FOR PRE-POSITIONING THE MAGAZINE
8 LBL 5 ; CHECK WHETHER TOOL IS ALREADY IN THE SPINDLE
9 FN 18: SYSREAD Q18 = ID2000 NR60 IDX2301; READ BYTE 2301
10 FN 9: IF +Q18 EQU +0 GOTO LBL 5 ; BYTE2301=0: WAIT FOR PLC
11 FN 11: IF +Q18 GT +1 GOTO LBL 3 ; BYTE2301=2: TOOL IS ALREADY
12 ; IN THE SPINDLE
13 FN 18: SYSREAD Q10 = ID1000 NR4210 IDX0 ; CHANGE POSITION IN AXIS X
14 FN 18: SYSREAD Q11 = ID1000 NR4210 IDX2; CHANGE POSITION 1 IN AXIS Y
15 FN 18: SYSREAD Q12 = ID1000 NR4210 IDX5; CHANGE POSITION IN AXIS Z
16 FN 18: SYSREAD Q15 = ID1000 NR4210 IDX3; CHANGE POSITION IN AXIS Y
17 L X+Q10 Y+Q11 Z+Q12 R0 F MAX M91 ; MOVE TO TOOL CHANGE POSITION
18 LBL 4 ; BYTE2300=1: SPINDLE AND MAGAZINE IN POSITION ?
19 FN18: SYSREAD Q18 = ID2000 NR60 IDX2300
20 FN10: IF +Q18 NE +1 GOTO LBL 4
21 L Y+Q15 R0 F MAX M91 ; TOOL IN CHANGER
22 L Y+Q11 M71 ; CLAMP THE TOOL AND RETURN TO THE CHANGE POSITION
23 LBL 3
24 TOOL CALL Q1 Z SQ3 DL+Q4 DR+Q5 ; TOOL CALL WITH T STROBE
25 M113 ; CANCEL M112
26 END PGM TCALL MM
```

Variable and fixed pocket coding

You can work with either variable or fixed pocket coding.

- Specify with MP7480 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: It is advisable to work with the tool number. Set MP7480.x = 1 or 2.

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

		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.

- With every TOOL CALL or TOOL DEF block: MP7480.x = 2
- When changing a tool number: 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.

You define a fixed pocket with the "F" field. 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 programmed with pocket number if MP7480.0 = 3 or 4 and TOOL CALL**	NC	NC
M4523	**Tool programmed without pocket number is effective if 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

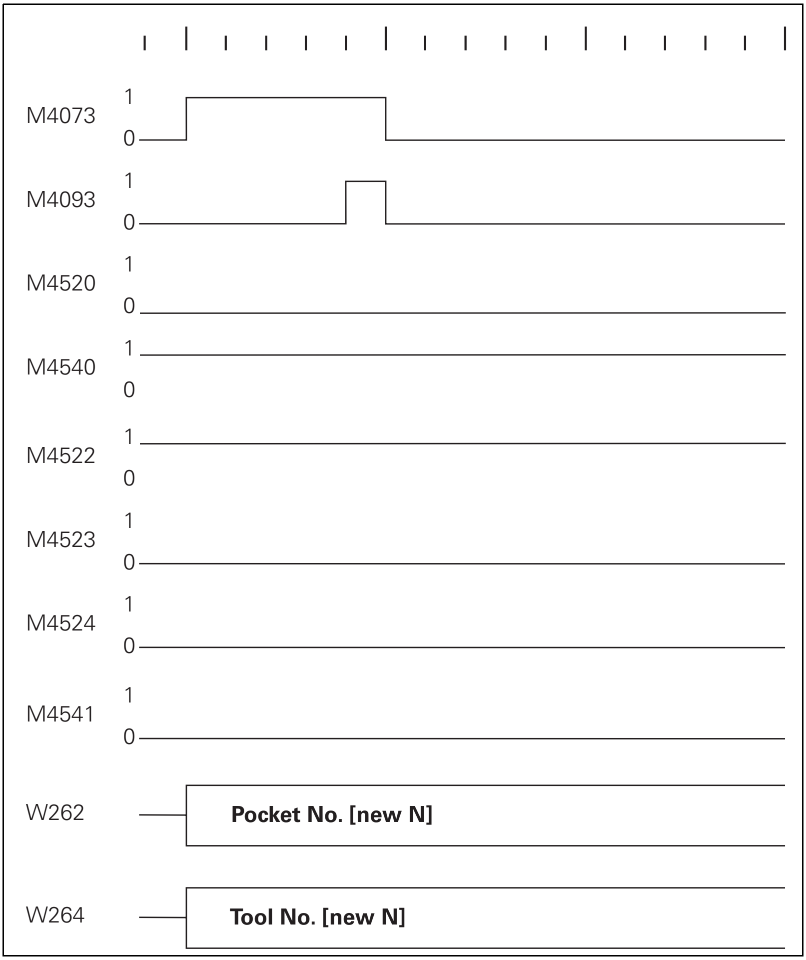
A variety of tool types can be called from the machining program. The abbreviations below are defined for the following examples:

- **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 recognize this with M4520 and M4540.
You must evaluate and acknowledge both pocket or tool numbers.

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:

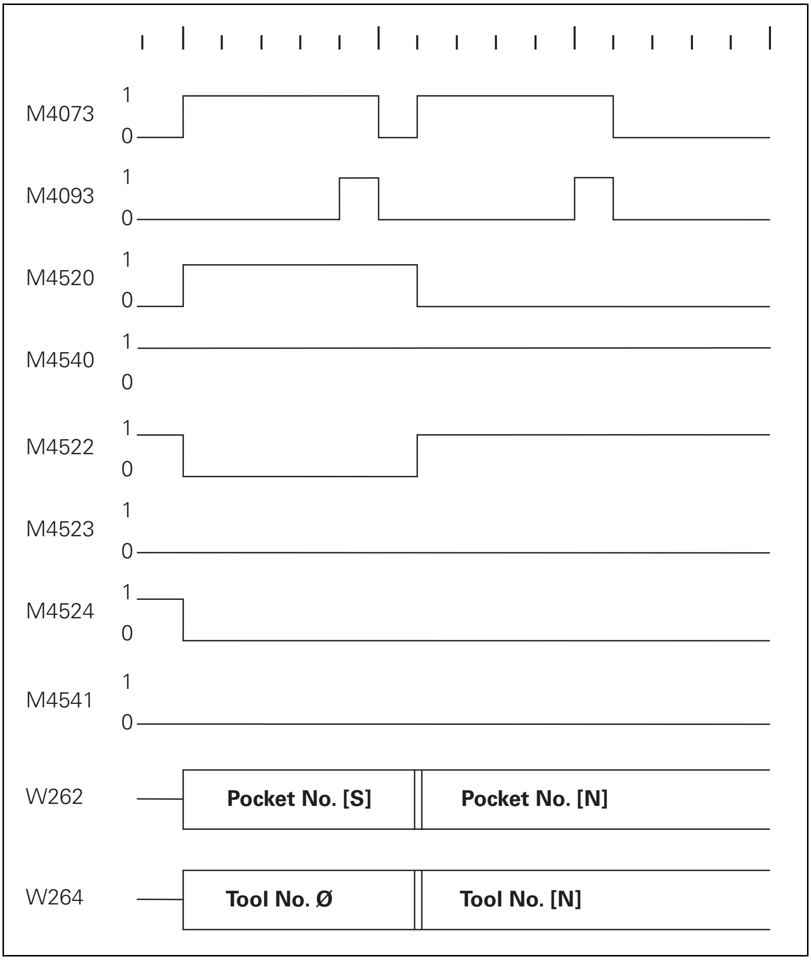
- With M4540 specify the sequence in which the pocket numbers are transferred, depending on whether single or double changing arm.

S → N, Single changing arm, M4540 = 0

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.

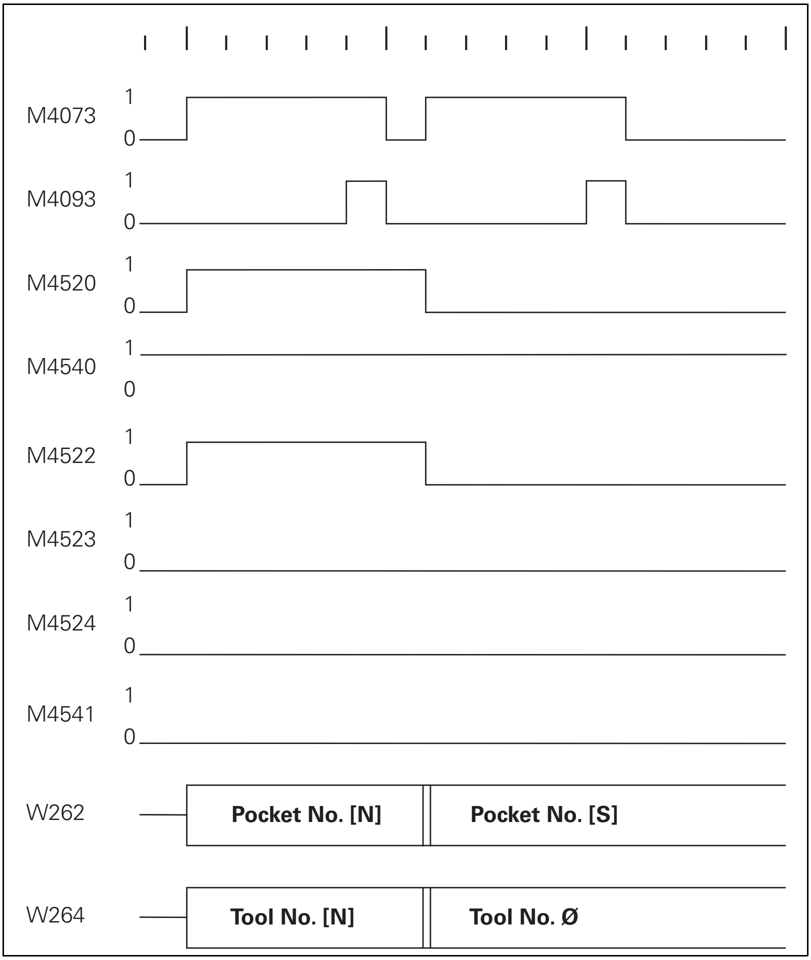


S → N, Double changing arm, M4540 = 1

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!

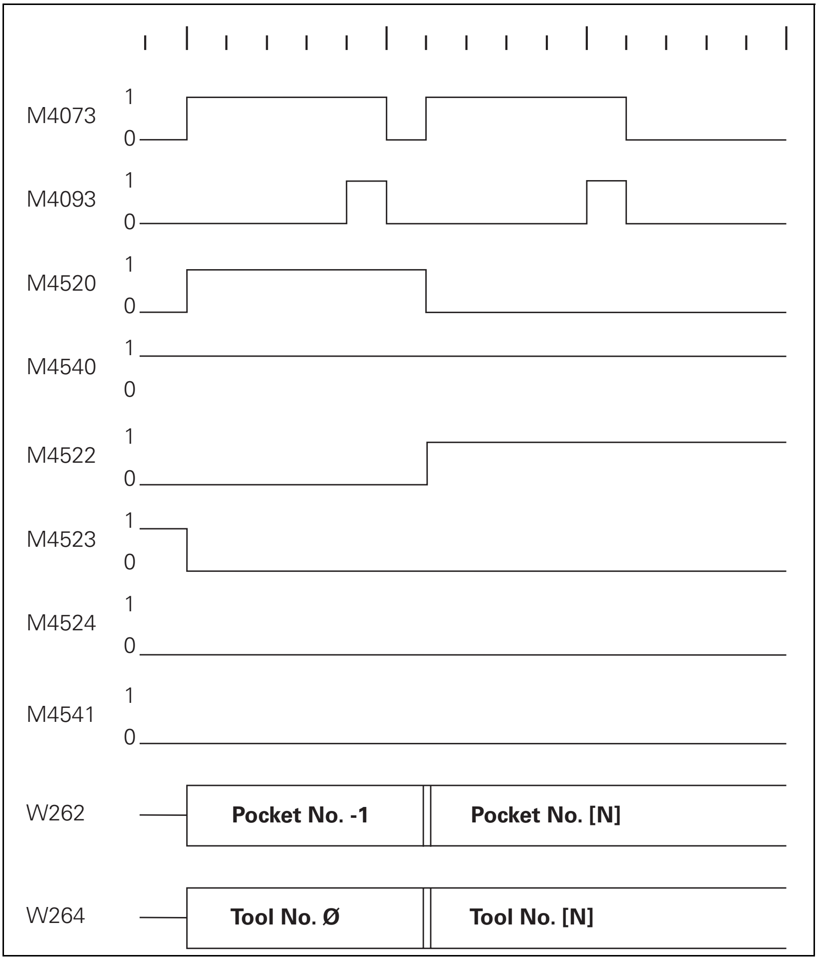


M → N: Normal tool follows a manual tool

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 marker M4540, the pocket number –1 and tool number zero are transferred first.

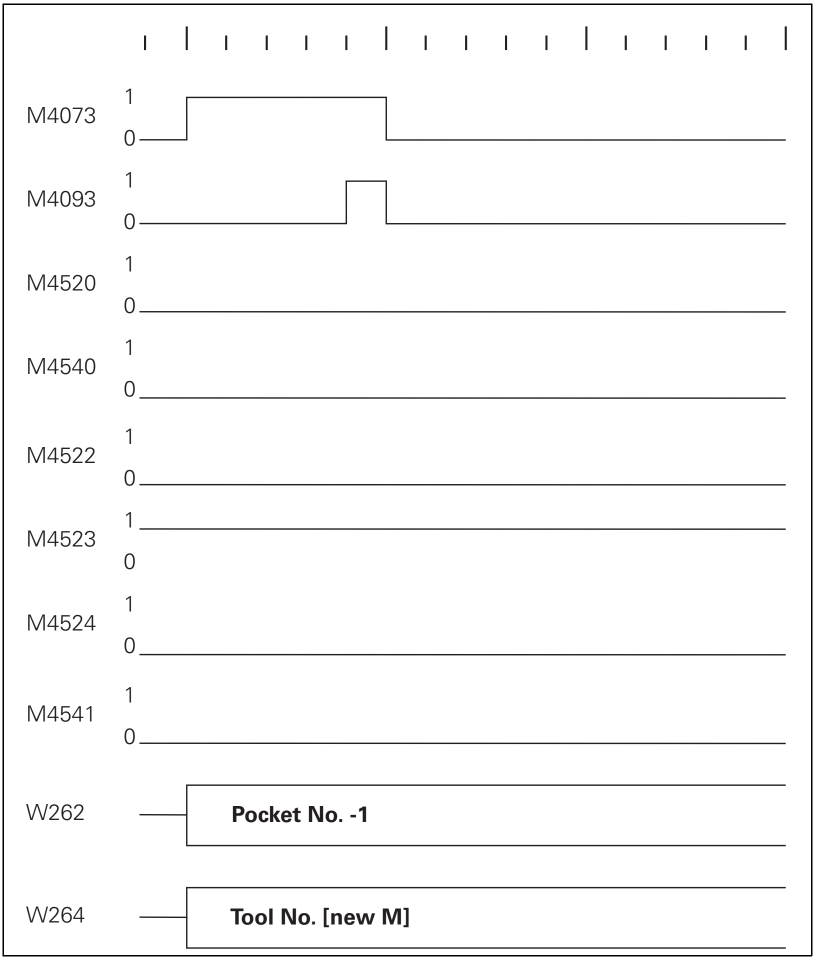
Zero means clear the spindle! Pocket number –1 means: no pocket in the tool magazine!

- 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!



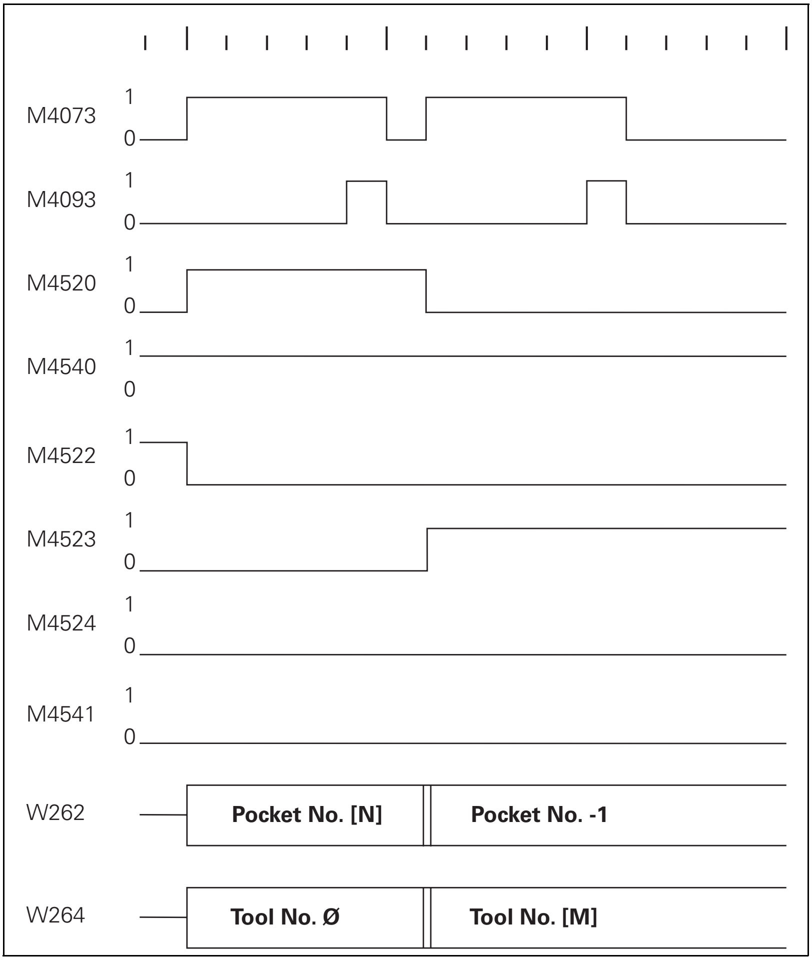
N → M: Manual tool follows a normal tool

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 marker M4540, 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!



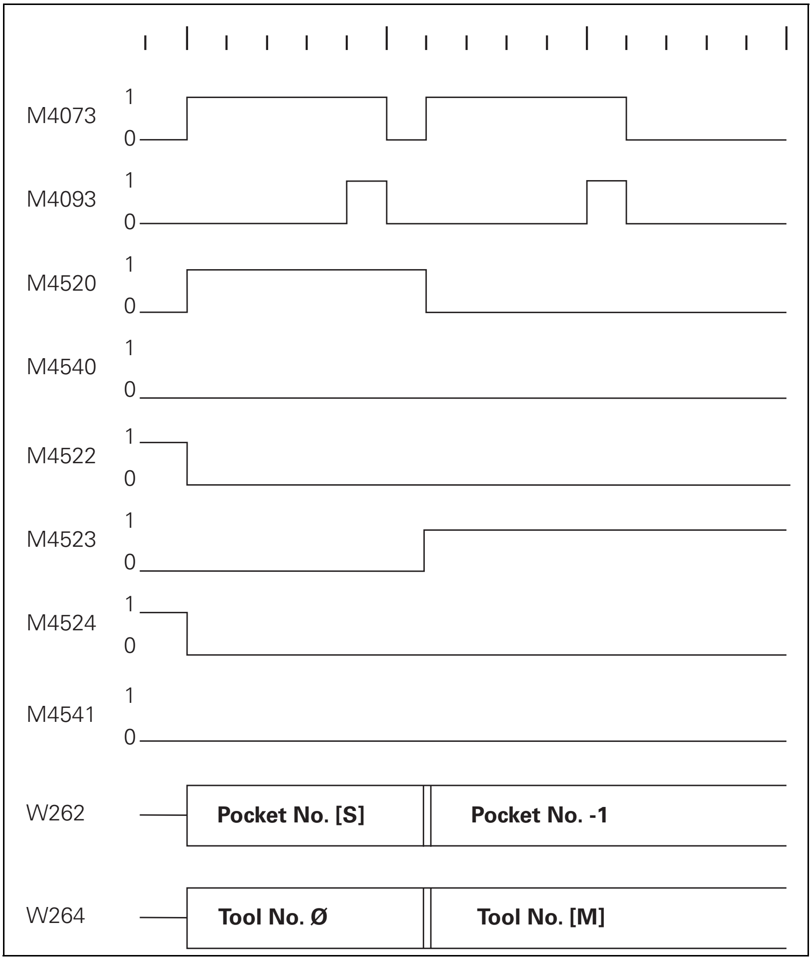
S → M: Manual tool follows a special tool

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 marker M4540, 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.

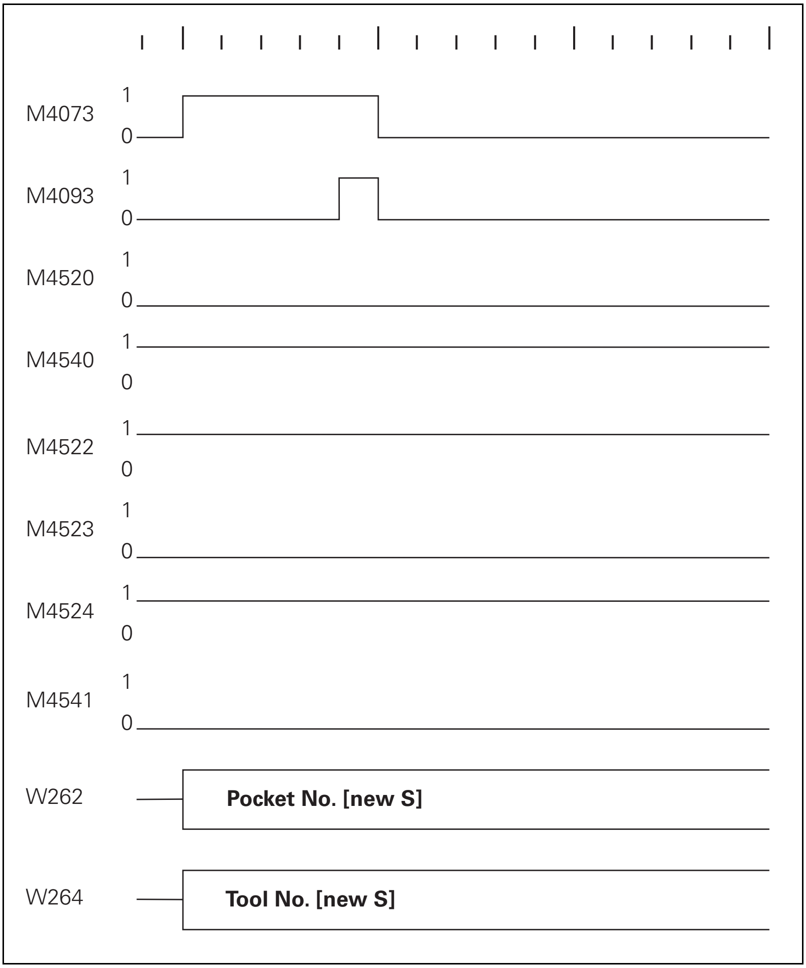
Pocket number –1 means: no pocket in the tool magazine!



S → S: Special tool follows a special tool

▶ With M4541 or the "F" field 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.

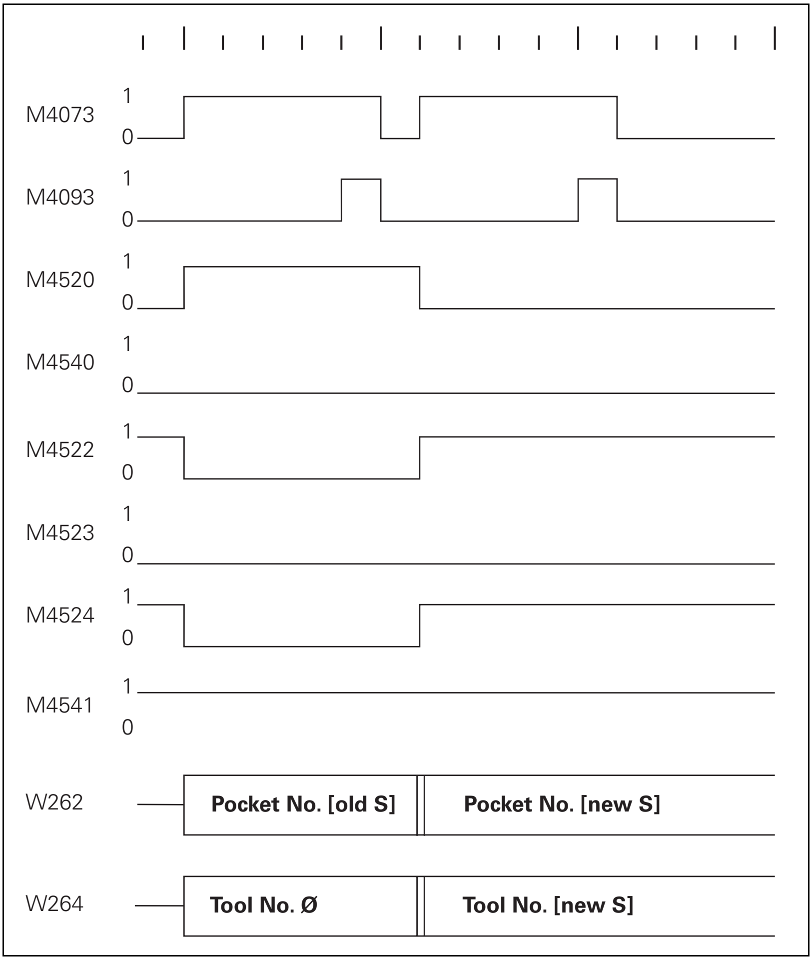


S → S, Single changing arm, M4540 = 0

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.

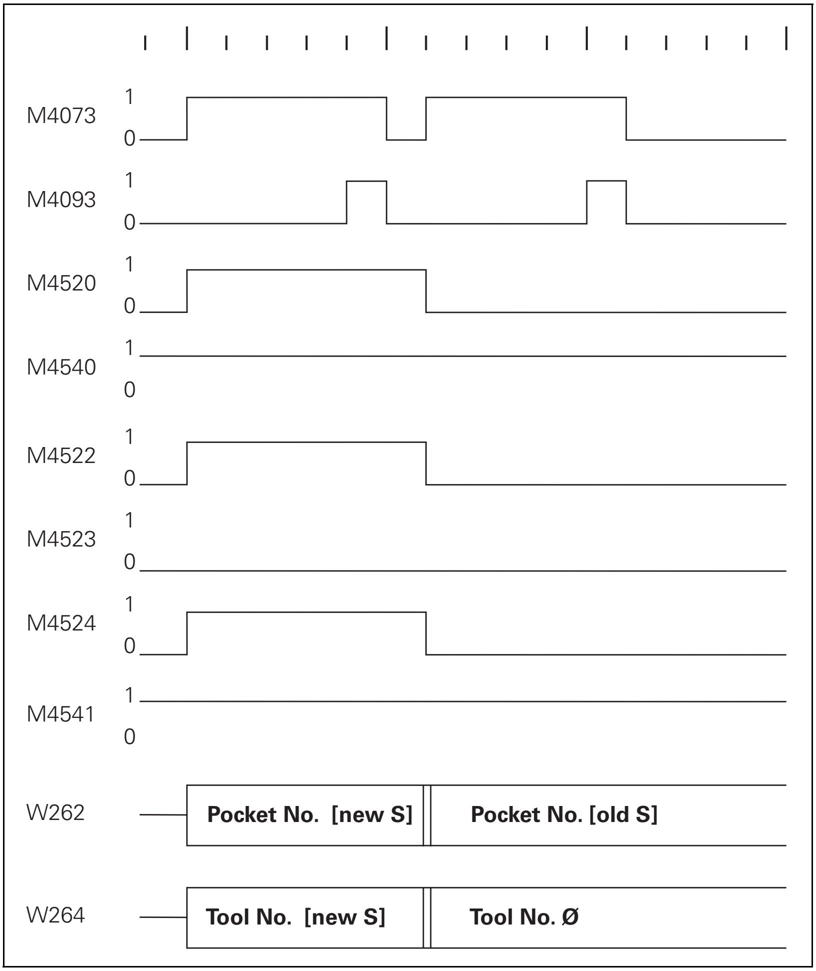


S → S, Double changing arm, M4540 = 1

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

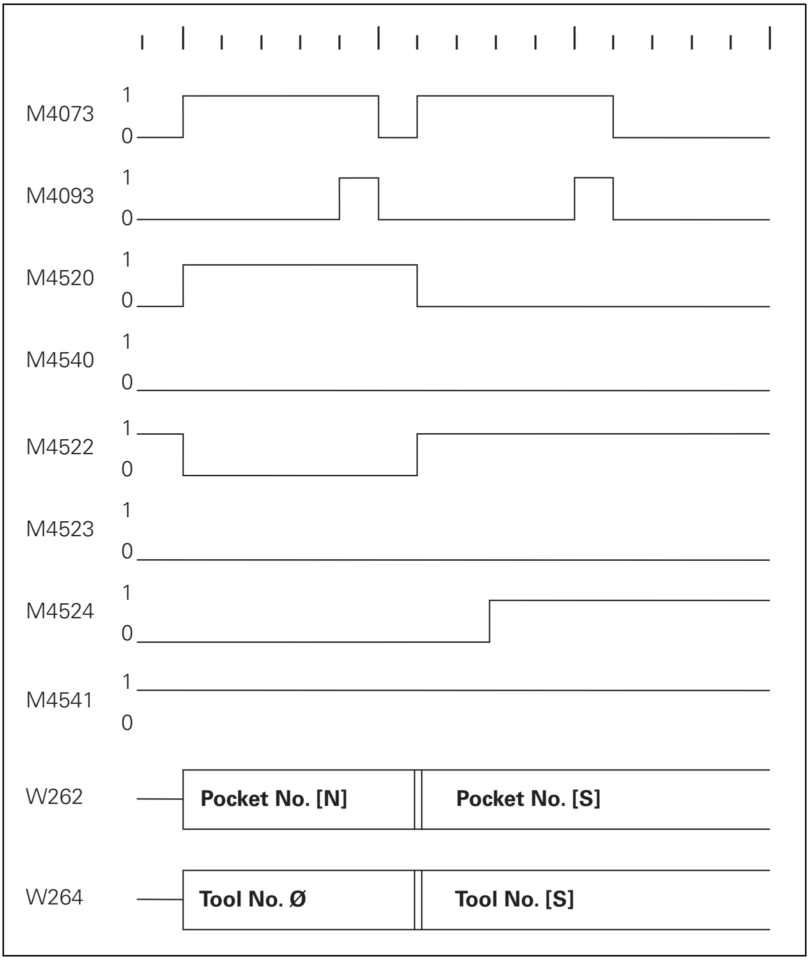
With this change sequence, two pocket numbers or two tool numbers must be transferred in succession. M4520 indicates that another TOOL CALL strobe (M4073) will follow. Regardless of M4541, there is a different sequence of pocket number transfer for single and double-arm changers (M4540).

N → S, Single changing arm, M4540 = 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.

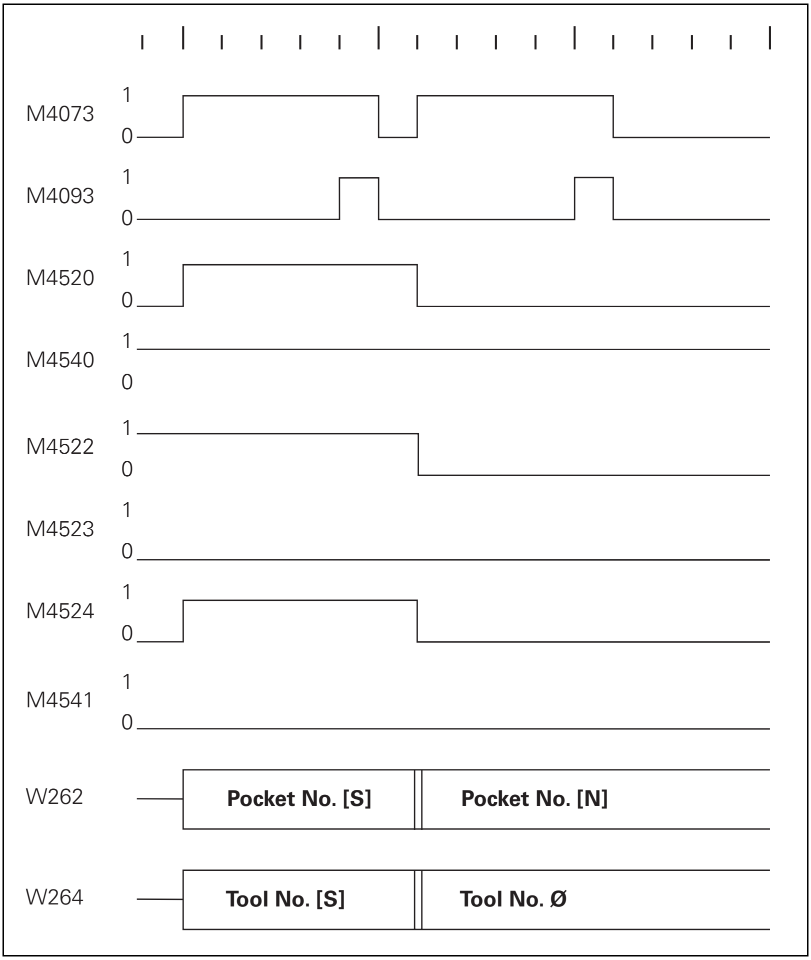


N → S, Double changing arm, M4540 = 1

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!

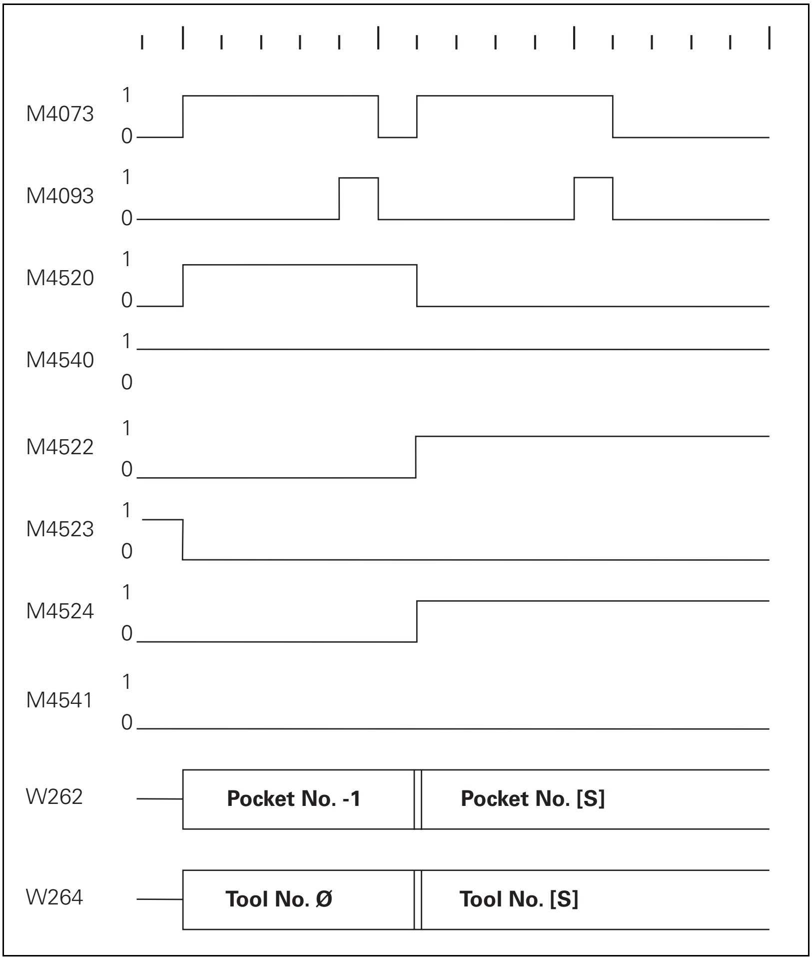


M → S: Special tool follows a manual tool

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 markers M4540 and M4541, 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.



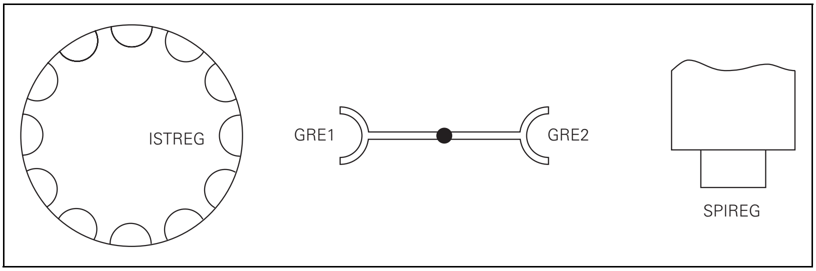
6.21.5 PLC Programming Example

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:

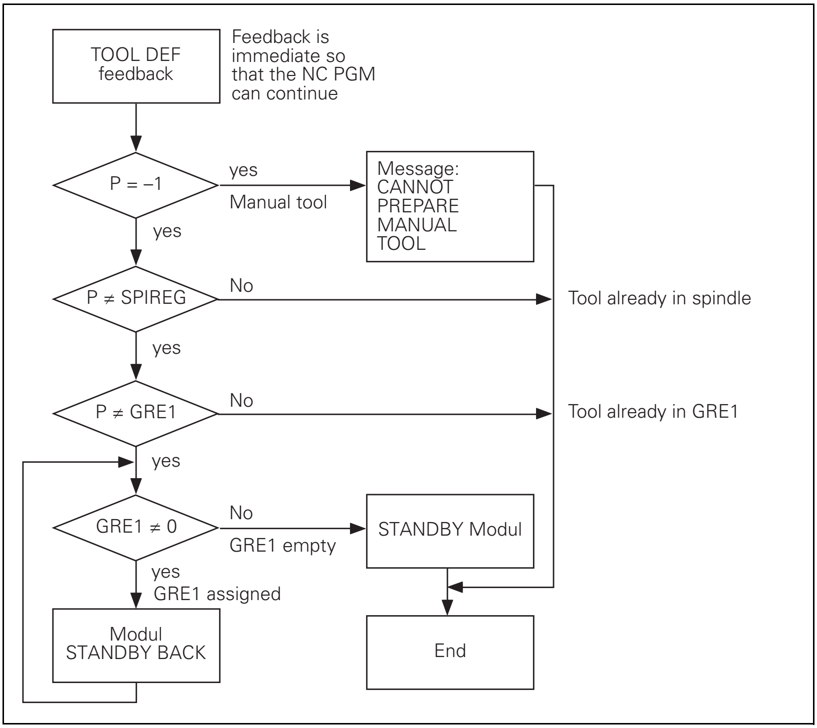
Machine parameter	Meaning
MP7260 = 90	Number of tools in the tool table
MP7261 = 12	Number of the pockets in the tool magazine
MP7480.0 = 4	Output of the pocket number and tool number for every TOOL CALL block
MP7480.1 = 4	Output of the pocket number and tool number for every TOOL DEF block

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	

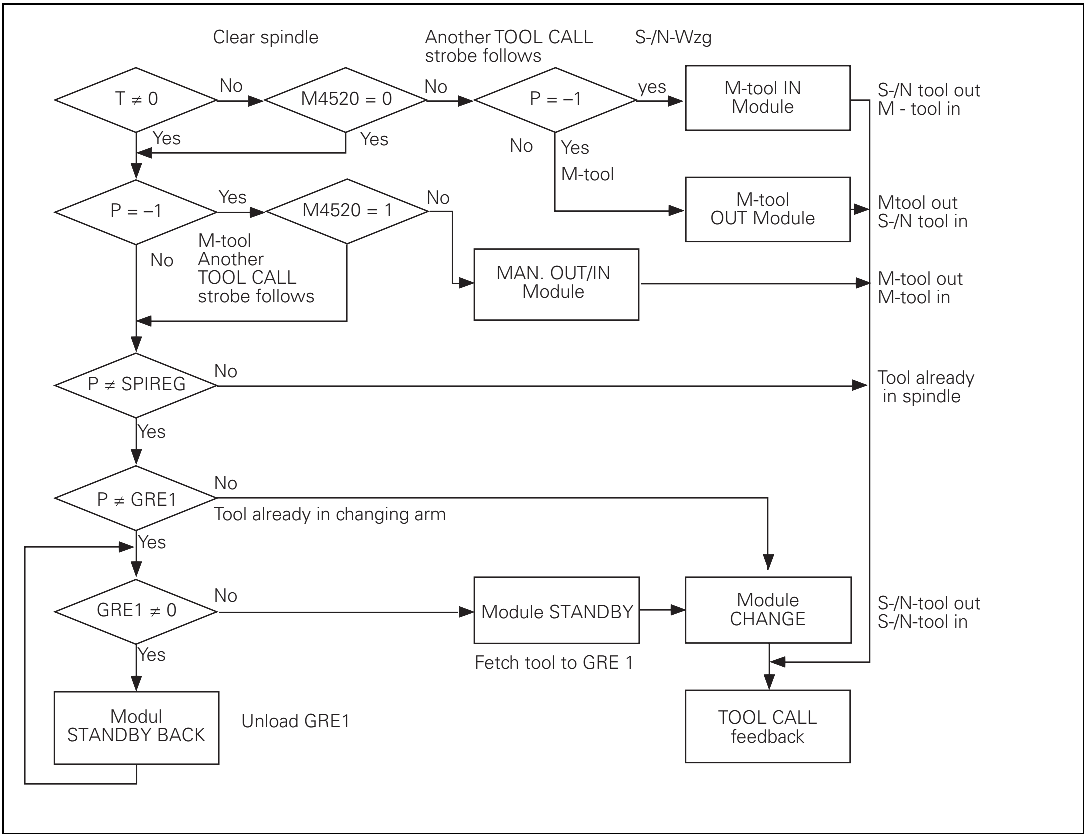
TOOL DEF program module

Search for tool and load in GRE1



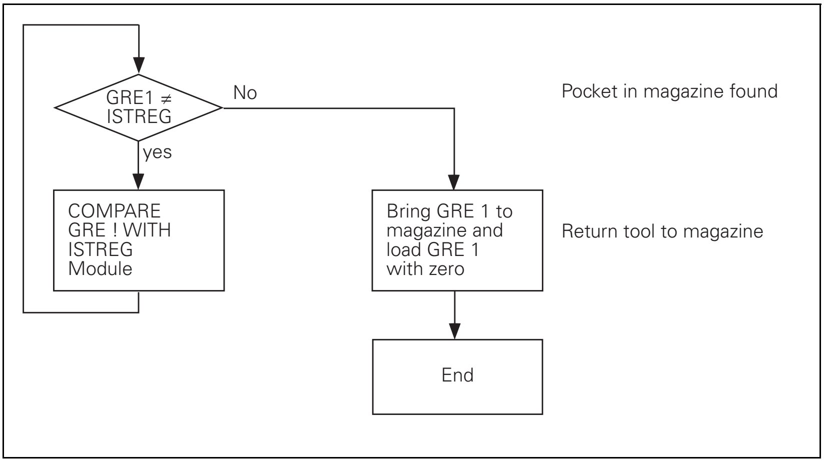
TOOL CALL program module

Automatic tool change, main program



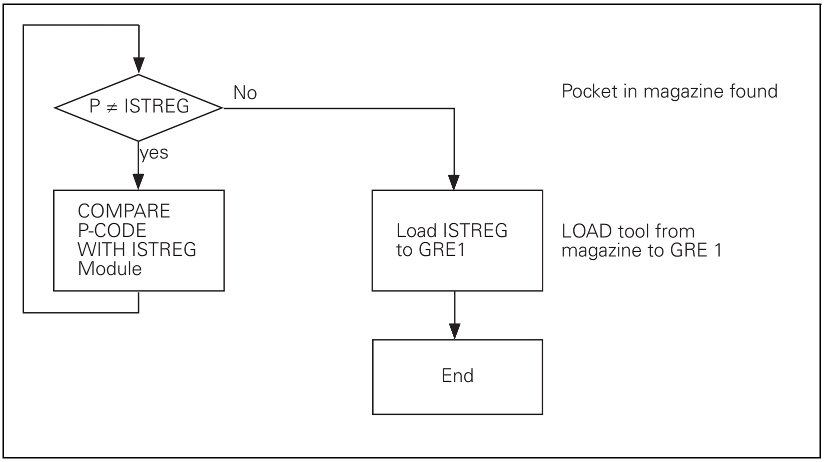
STANDBY program module

Search for tool and load in GRE1



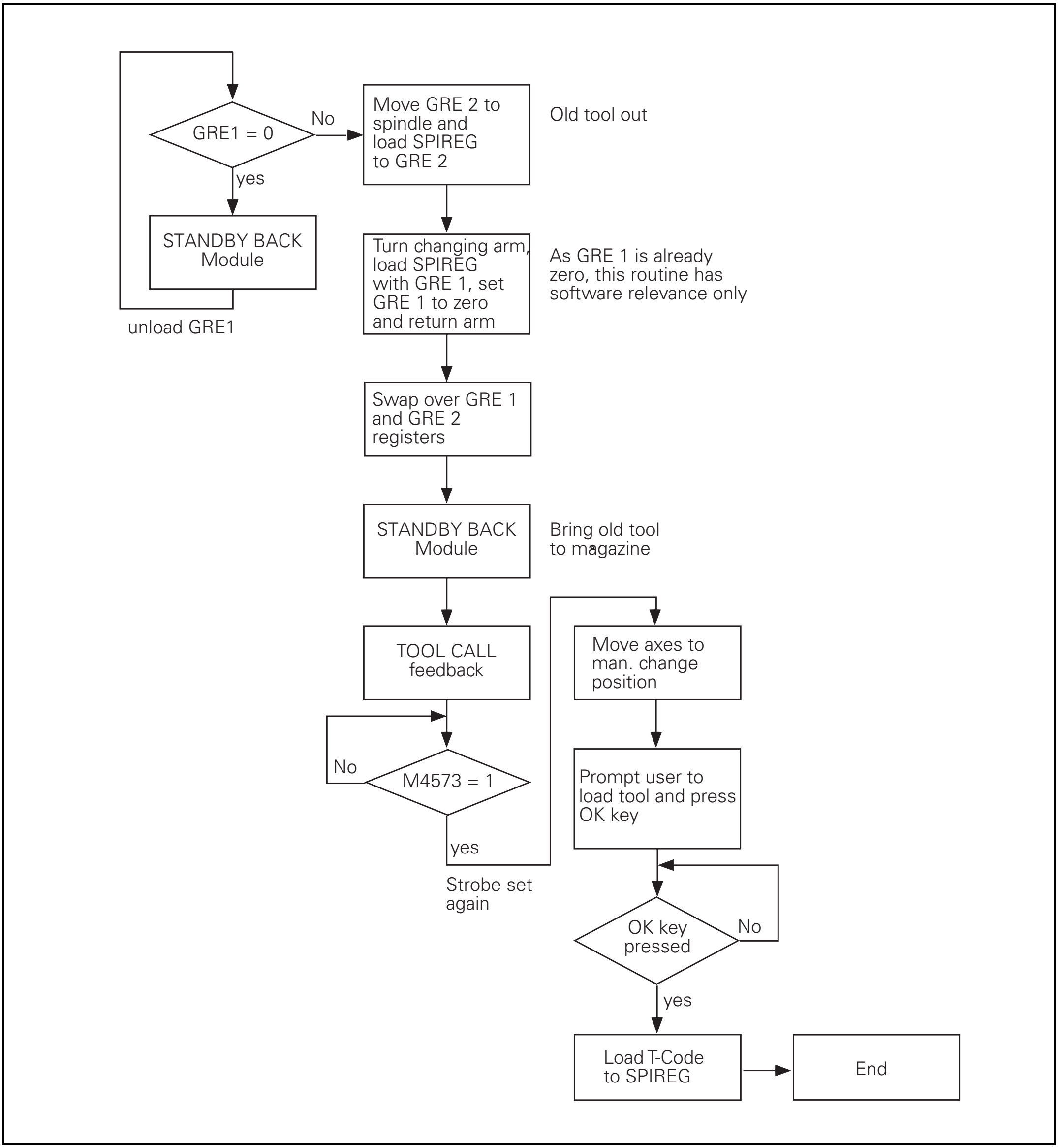
STANDBY BACK program module

Return tool from GRE1 to the tool magazine



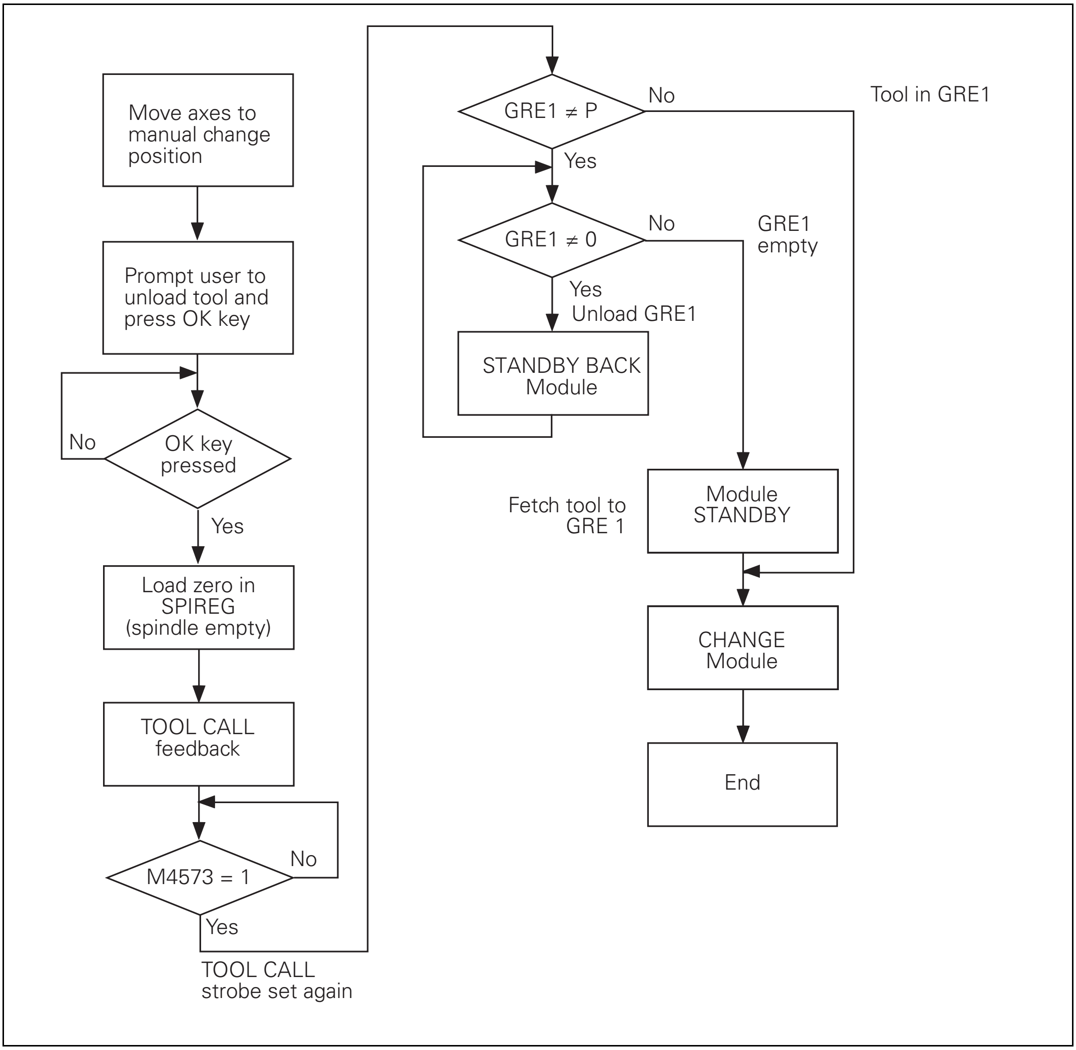
MANUAL TOOL IN program module

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).



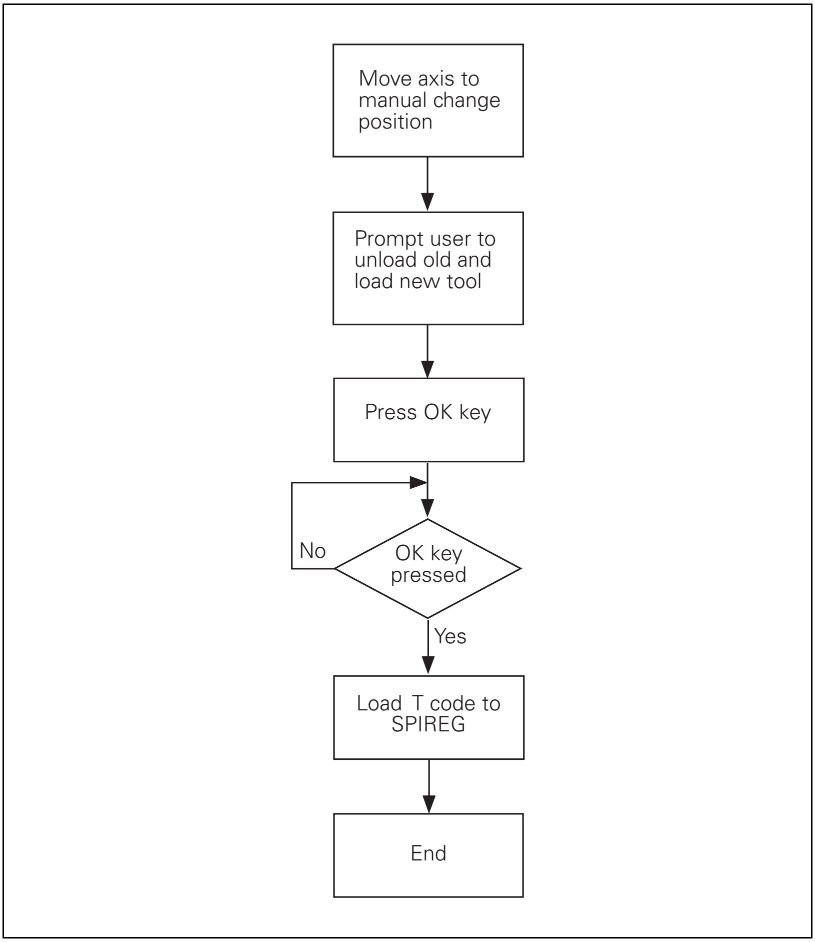
MANUAL TOOL OUT program module

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.



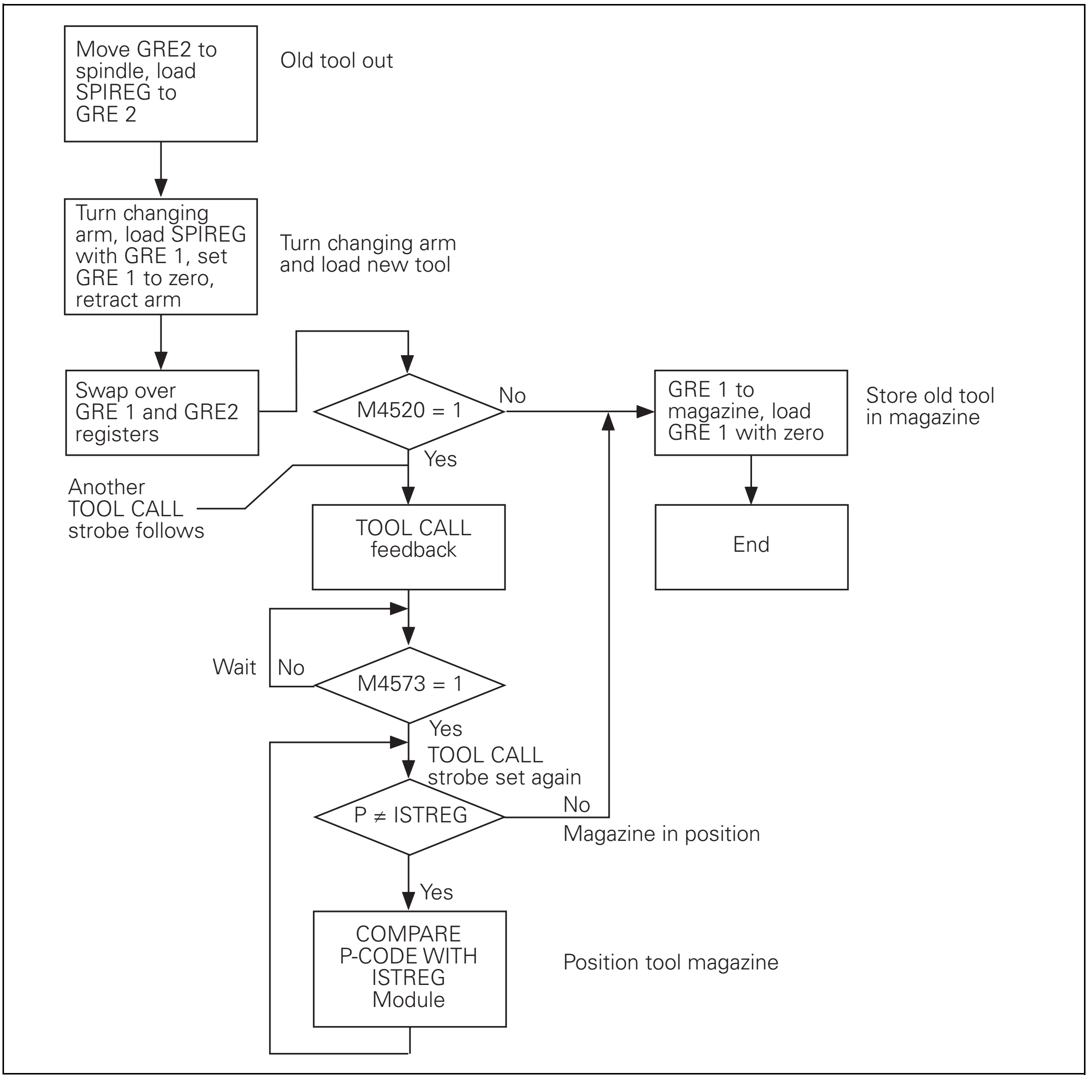
MANUAL TOOL OUT/IN program module

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.



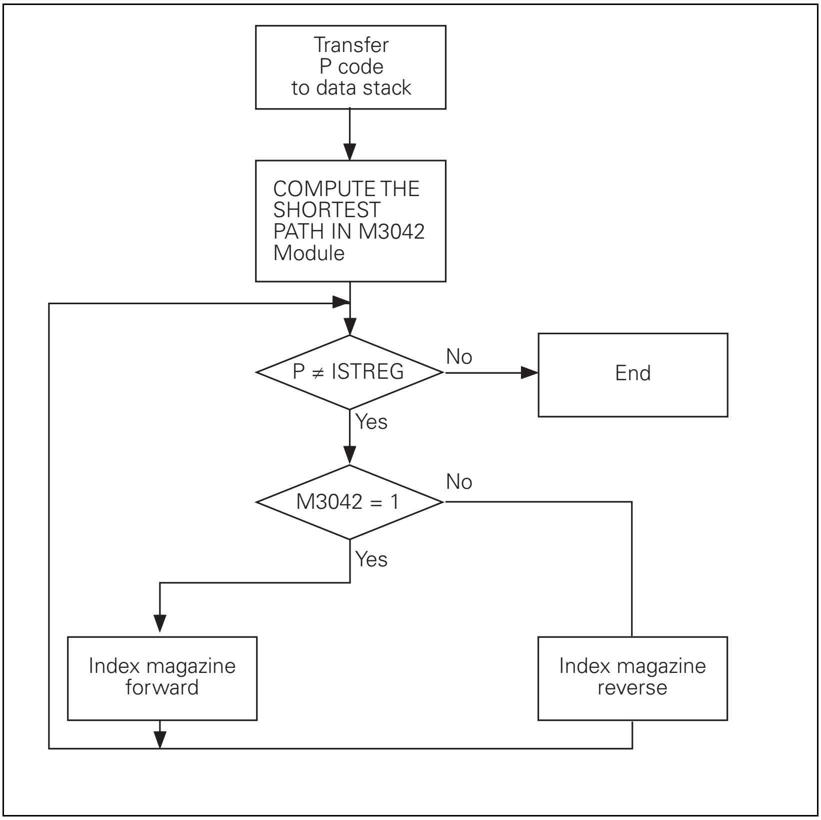
INSERT program module

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).



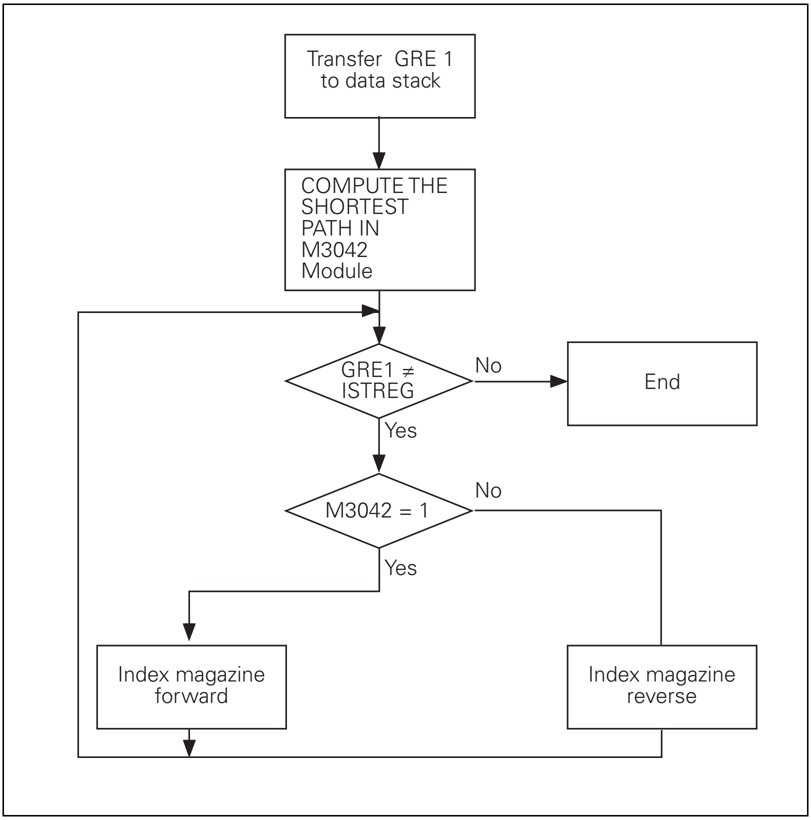
COMPARE P CODE WITH ISTREG

The tool magazine is positioned in the shortest direction to the desired pocket number.



COMPARE GRE1 WITH ISTREG program module

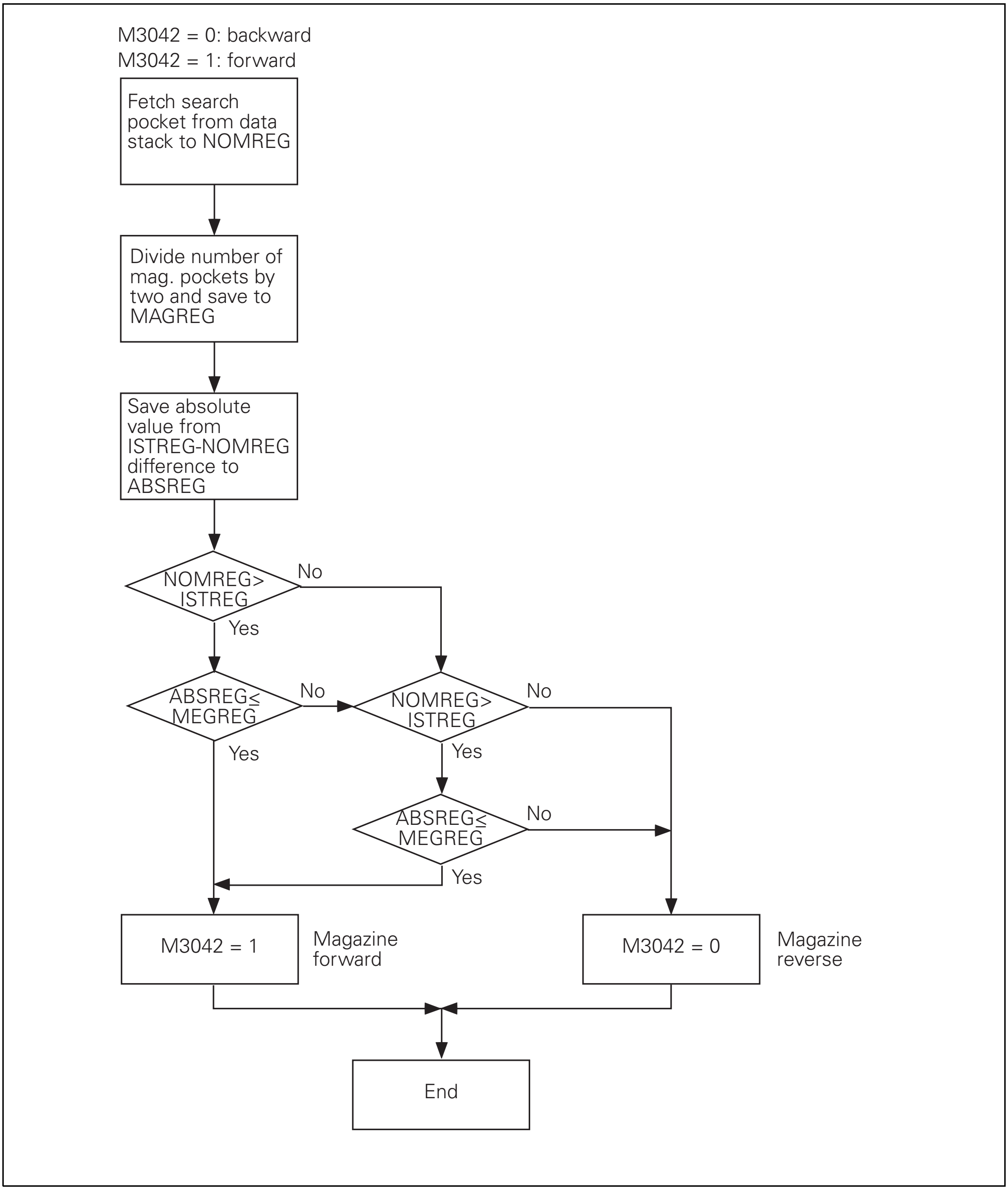
The tool magazine is positioned in the shortest direction to the pocket number that is located in GRE1.



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



6.22 Special Functions for Laser Cutting Machines

You can activate special functions to interface the TNC to laser cutting machines and water jet machines.

6.22.1 Analog Voltage Output

If you do not need the analog output S (X8, pin 8) 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 above-mentioned M functions will not be 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 TNC 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 TNC outputs the voltage as a function of the traversed distance. Starting from the active voltage, the TNC 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 TNC 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 TNC 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 TNC detects the beginning of a new curve from the input value zero.

Example:

Velocity		**Voltage**		**Curve**
MP3013.0	0	MP3014.0	0	1
MP3013.1	25	MP3014.1	0	
MP3013.2	500	MP3014.2	4.5	
MP3013.3	1000	MP3014.3	9.999	
MP3013.4	0	MP3014.4	0	2
MP3013.5	10 000	MP3014.5	9.999	
MP3013.6	0	MP3014.6	0	3
MP3013.7	50	MP3014.7	0.5	
MP3013.8	300	MP3014.8	1.5	
MP3013.9	5000	MP3014.9	9.999	
MP3013.10	0	MP3014.10	0	Not used
MP3013.11	0	MP3014.11	0	

MP3013.x **Characteristic curve kink points (velocity) for output of the analog voltage with M202**
Input: 10 to 300 000 [mm/min]

MP3014.x **Characteristic curve kink points (voltage) for output of the analog voltage with M202**
Input: 0.000 to 9.999 [V]

Voltage output varies with the time (time-dependent ramp): M203 V... TIME...

The TNC outputs the voltage as a function of the time. Starting from the active voltage, the TNC 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 TNC 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.

6.22.2 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

6.22.3 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

6.23 Integrated Oscilloscope

The TNC features an integrated oscilloscope.

With this oscilloscope you can record the following characteristics in up to four channels:

Actl. speed	Actual value of the axis feed rate [mm/min]. Calculated from position encoder
Noml. speed	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]
Actual pos	Actual position [mm]
Noml. pos	Nominal position [mm]
Lag	Following error of the position controller [µm]
Position: I1	Signal 1 of the position encoder
Position: I2	Signal 2 of the position encoder
SAVED	The signal last recorded is saved.
PLC	The PLC operands (B, W, D, I, O, T, C) are recorded. Enter the operands in the input field next to the PLC.
Acceleration	Nominal value of the acceleration [m/s^2]
Jerk	Nominal value of the jerk [m/s^3]
Pos. Diff.	Difference between position and speed encoder [mm]
Current Accel.	Current acceleration value [m/s^2]. Calculated from position encoder.
Current Jerk	Current jerk value [m/s^3]. Calculated from position encoder.

Analog axes:	
Volt.analog	Analog voltage = nominal velocity value [mV]

Digital axes:	
V (ACT RPM)	Shaft speed actual value [mm/min]; Calculated from rotary speed encoder and standardized with MP2020
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

The oscilloscope provides additional functions for commissioning the current controller. (See "Commissioning” on page 6 – 462).

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

Colors

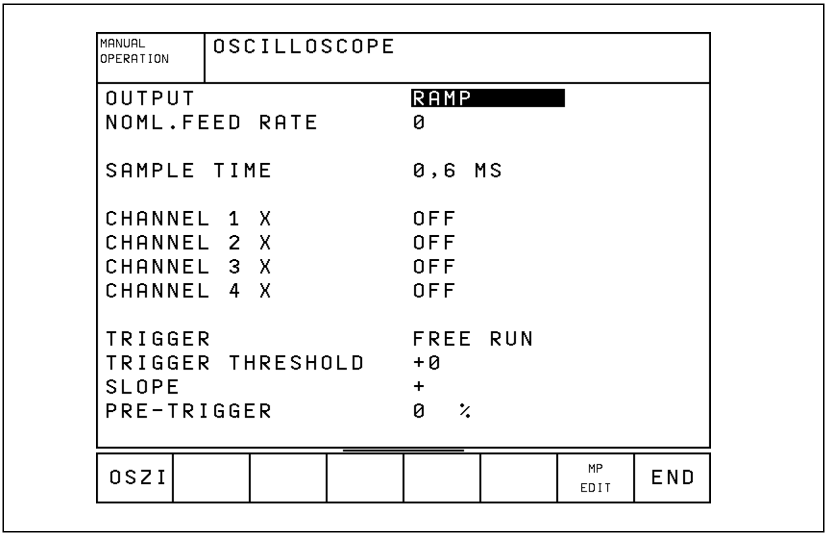
▶ In MP7365.x, define the colors for the oscilloscope.

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: 0.6 to 6 ms. 4096 samples are stored. The signals are therefore stored for a duration of 2.4576 to 24.576 seconds.

Channel 1 to channel 4:

▶ 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 4 The recording begins when the triggering threshold of the selected channel is exceeded.

Trigger threshold:

▶ Enter the trigger threshold in the following dimensions:

- Velocity [mm/min]
- Position [mm]
- Shaft speed [mm/min]
- Following error [µm]
- Analog voltage [mV]
- Current [A]
- Acceleration [m/s^2]
- Jerk [m/s^3]

Slope:

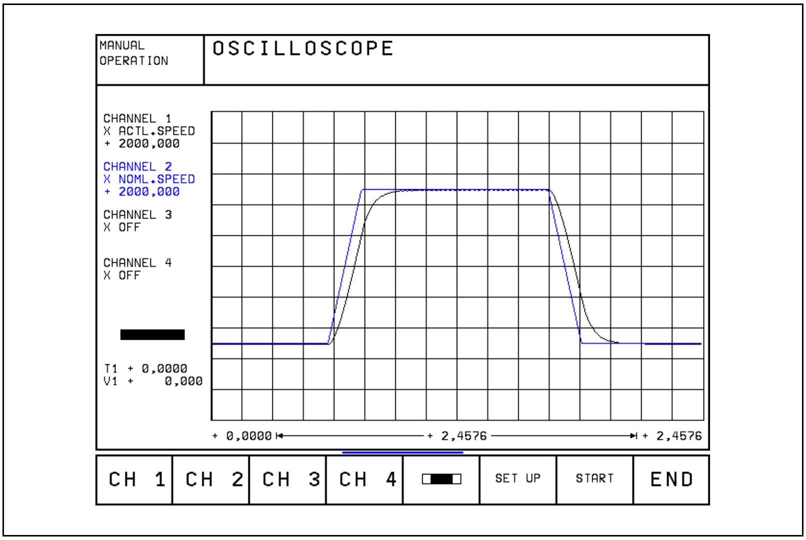
▶ Select whether the rising edge (positive slope) or falling edge (negative slope) of the signal acts as trigger.

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

▶ Enter a value.

Oscilloscope display:

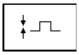
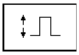
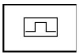
▶ Press the OSCI Soft key.

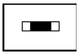
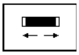
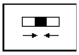


During recording, the selected signals are continuously displayed. After recording ends, the memory contents are displayed. For every channel, the type of signal and the resolution are also shown. The length of the recorded range, with respect to the entire memory content, is shown as a bar in the status field.

- Move the cursor with the arrow keys. 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:		
CH 1	You select one of the four channels, and a new soft-key row with the following soft keys appears.	
	INVERT	Invert the signal.
	↓	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. With NO ENT you return to the resolution chosen originally.
	CURSOR 1/2	Switch to second cursor.
	END	Return to oscilloscope display.

Meaning of the soft keys:		
	Select the memory area to be displayed. A new soft-key row with the following soft keys appears:	
	←	Move the signal to the left
	→	Move the signal to the right
		Decrease the horizontal resolution.
		Increase the horizontal resolution.
	END	Return to oscilloscope display.
SET UP	Back to setup menu.	
START	Start recording. The recording is ended either with a trigger condition or with the STOP soft key.	
END	Exit the oscilloscope function.	

Saving the recording

With “Saved” you can store the last recorded signal.

With the SAVE SCREEN soft key you can save the recorded signals with all settings in a file on the hard disk. The file must have the extension .DTA. You can recall these data with the PC program PLCdesign.

6.24 Commissioning

6.24.1 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 – 173))
- 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 the instruction PLCMAIN= in the system file OEM.SYS refers to the current PLC program.

6.24.2 Digital Axes

Digital and analog axes are defined with MP120.

The TNC must be adjusted in sequence for the:

- Current controller
- Speed controller
- Position controller

The signals that you need are recorded with the integral oscilloscope.

NC software 280 470-xx

- Define digital and analog axes with MP2000.x.

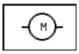
MP2000 **Type of drive**
Input: 0: Output of nominal speed command (analog axis)
1: Output of current pulse (digital axis)

MP2001 **Type of drive for spindle**
Input: 0: Output of nominal speed command (analog spindle)
1: Output of current pulses (digital spindle)

Motor and power module

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

- Call the respective menu with the corresponding soft key (see illustration).

Meaning of the soft keys:		
	Call a list of power modules.	
	SELECT AXIS	Select axis and confirm the marked power module with SELECT.
	PRESENT VALUE	Open the table of power modules and jump to the selected power module.
	APPEND AMPLIFIER	Add a power module.
	END	Return to the machine parameter editor.
	Call a list of motors.	
	SELECT AXIS	Select axis and confirm the marked motor with SELECT.
	PRESENT VALUE	Open the table of motors and jump to the selected motor.
	APPEND ASYNC	Add the motor.
	END	Return to the machine parameter editor.

After you have selected the motor and the power module, the models are automatically entered in MP2100.x, MP2101, MP2200.x and MP2201.

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 list of the motor models or power modules, the changed tables are filed in the PLC partition:

- PLC:\MP\MOTOR.MOT List of synchronous and asynchronous motors
- PLC:\MP\MOTOR.ASN List of asynchronous motors
- PLC:\MP\MOTOR.SN List of synchronous motors
- PLC:\MP\MOTOR.AMP List of power modules

These tables are then taken into account by the TNC. 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 the PLC partition does not contain a MOTOR.MOT table, the control searches the PLC partition for the MOTOR.ASN and MOTOR.SN lists. If these are not available either, the HEIDENHAIN standard table is used.

Entries in the list of power modules:

- Power module designation (NAME)
- Maximum current (I-MAX) in A
- Rated current (I-N) in A
- Current sensor voltage (U-IMAX) in V/A
- Rated current with DC (I-N-DC) in A
- Thermal time constant DC in s
- Transition frequency on T-DC (F-DC) in Hz
- Thermal time constant AC (T-AC) in s
- Transition frequency on T-AC (F-AC) in Hz
- Protection time of the IGBTs (T-IGBT) in µs
- Rated current at 3.33 kHz (I-N-AC-3333) in A
- Rated current at 4.0 kHz (I-N-AC-4000) in A
- Rated current at 5.0 kHz (I-N-AC-5000) in A
- Rated current at 6.66 kHz (I-N-AC-6666) in A
- Rated current at 8.0 kHz (I-N-AC-8000) in A
- Rated current at 10.0 kHz (I-N-AC-10000) in A

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)
  - 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

---

1. (See "Field orientation" on page 6 – 470)

- 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 DC (T-DC) in s
- Transition frequency on T-DC (F-DC) in Hz
- Thermal time constant AC (T-AC) in s
- Transition frequency on T-AC (F-AC) in Hz

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]
  Spindle with integral DSP: 5000 Hz
  Spindle without integral DSP: Value from MP2180.0-8
- 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.

MP2100.0-8 Type of power module for axes 1 to 9
Input: Name of the selected power module (entered by the TNC)

MP2101 Model of power module for the spindle
Input: Name of the selected power module (entered by the TNC)

MP2200.0-8 Motor model for axes 1 to 9
Input: Name of the selected motor (entered by the TNC)

MP2201 Motor model for the spindle
Input: Name of the selected motor (entered by the TNC)

Maximum revolutions per minute

	Maximum revolutions per minute
Axis drives TNC 426 PB/M TNC 430 PA	$\frac{24\,000}{\text{No. of pole pairs}}$ rpm
Spindle drives TNC 426 PB standard TNC 426 M / 12 000	$\frac{24\,000}{\text{No. of pole pairs}}$ rpm
Spindle drives TNC 426 PB option TNC 426 M / 30 000 TNC 430 PA/M	$\frac{60\,000}{\text{No. of pole pairs}}$ rpm

The maximum revolutions per minute in the motor data sheets are indicated for a definite dc-link voltage. If you work with a lower dc-link voltage, the given speed is not reached.

You can combat this effect and reach a higher speed on synchronous motors by entering a field-angle offset:

- In MP2340.x enter a speed from which the field angle is to be shifted. This increases the current starting from the threshold speed. The thermal limit curve is shifted.
- In MP2350.x enter the maximum angle of the shift.

MP2340.0-8 Speed starting from which the field angle begins to shift on synchronous motors for axes 1 to 9
Input: 0 to 100 000 rpm
0: No field angle offset

MP2350.0-8 Field-angle offset on synchronous motors for axes 1 to 9
Input: 0 to 60 [°]

Logic unit up to Id. Nr. xxx xxx–3x:

The HEIDENHAIN and SIEMENS current controllers differ in their characteristics. The maximum speed for synchronous motors attainable with the TNC lies 15% below the value given in the SIEMENS data sheets. Please take this into account when you choose motors. By entering a field-angle offset you can reach the maximum speed specified in the SIEMENS data sheet.

To do this, enter the following values:

- MP2340 = rated speed / 1.2
- MP2350 = 30°

Note

Remember that with these data, starting from the threshold speed the motors draw 16% more current than SIEMENS specifies. The thermal limit curve is shifted.

dc-link 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
UV 120, UV 140, UV 150, UR 2xx: 650 V

PWM frequency with HEIDENHAIN inverters

The HEIDENHAIN TNC 426 PB/M, TNC 430 PA/M controls and the HEIDENHAIN inverters work with a PWM frequency of 5000 Hz:

▶ In all MP2180.x and in MP2181, enter the PWM frequency 5000 Hz (or input value 0)

MP2180.0-8 **PWM frequency of the axes**
Input: 3000 to 7000 [Hz]
0 = 5000 Hz (for HEIDENHAIN inverters)
MP2181 **PWM frequency of the spindle**
Input: 3000 to 7000 [Hz]
0 = 5000 Hz (for HEIDENHAIN inverters)

Note

The values between 1 Hz and 2999 Hz and between 7001 Hz and 10 000 Hz cause the DSP error message **C013 PWM frequency incorrect** after the acknowledgement of the **POWER INTERRUPTION** message.

PWM frequency with INDRAMAT "POWER DRIVE" inverters

▶ In all MP2180.x and in MP2181, enter the PWM frequency 4000 Hz.

PWM frequency with SIEMENS "SIMODRIVE" inverters

The HEIDENHAIN TNC 426 PB/M and TNC 430 PA/M controls work 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 (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 all MP2180.x and in MP2181, enter the same PWM frequency.
or

▶ Reduce the factor for I^2t monitoring, or reduce the rated current I_N in the list or 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 without integral DSP 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 (MP2303) or the rated current I_N must be reduced.

Reduction of the reference value for the I^2t monitoring or the rated current I_N

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} = PWM frequency in kHz set in MP2180.0-8

This results in the reference value for I^2t monitoring:

$$X_B = 1 - \frac{X_R[\%]}{100}$$

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

$$X_R[\%] = 100 - \left(\frac{(100 - 40) \cdot (8\ \text{kHz} - 4\ \text{kHz})}{8\ \text{kHz} - 3.2\ \text{kHz}} + 40\right) = 10\ \%$$

$$X_B = 1 - \frac{10}{100} = 0.90$$

▶ 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.

$$MP230x = X_B$$

Note

A reduction of the rated current of the power module can cause a reduction of the rated torque and 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.

Field orientation

If a synchronous spindle is used and 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, the LE 426 M/30 000 rpm or the LE 430 M automatically determine the assignment between the encoder and the rotor magnets (field angle) during commissioning and save this information on the hard disk.

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.
- Press the I CONTROL soft key.
- Acknowledge the **`Power Interrupted`** message.
- Press the SELECT AXES soft key.
- Press the SPINDLE soft key.
- Press the FIELD ORIENT. soft key.
  The PLC must
  - switch the drive on/off.
  - release and lock the brakes

The spindle rotates with rated speed for the duration of 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.

- 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.

Temporary input values

▶ Enter the following temporary input values when you begin commissioning:

MP	Temporary input value	Meaning
MP20.0	%00000000	Monitoring the absolute position of the distance-coded reference marks
MP1030.x	0.01	Positioning window
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)
MP1340.x	0	No evaluation of reference marks
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
MP2020.x	?	Linear distance of one motor revolution (depends on the machine)
MP2221	%010	Current and speed controller 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)

MP	Temporary input value	Meaning
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
MP2800.x	0	Motion monitor for position and speed

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 instruction PLCPWM=.

It suffices to program an EM (end module).

The drive must be enabled externally and the TNC needs the “ready” signal.

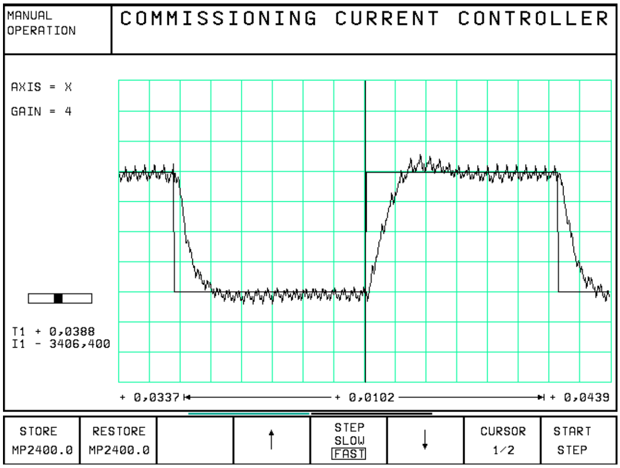
As soon as the PLC program defined with PLCPWM= is active, you can use Module 9168 to interrogate the commissioning status.

Module 9168 Interrogating the commissioning status

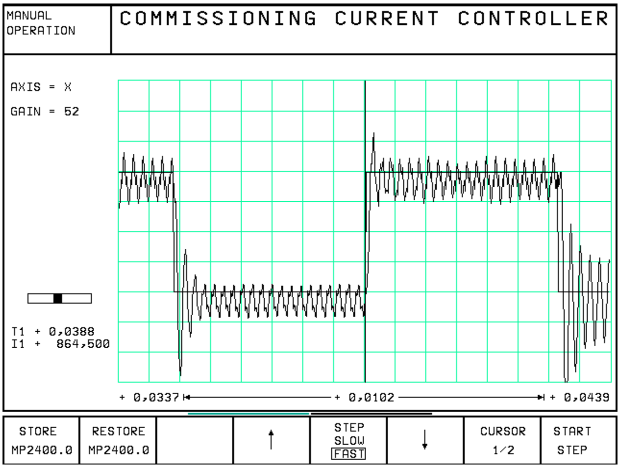
Call:
CM 9168
PL D <Status>
–1: Commissioning not active or as yet no axis is selected
Bits 0 to 5 correspond to selected axes 1 to 6
Bit 15 – Spindle selected
Bit 16 – Circuit type of the spindle
0: Wye connection
1: Delta connection

Adjusting the current controller:

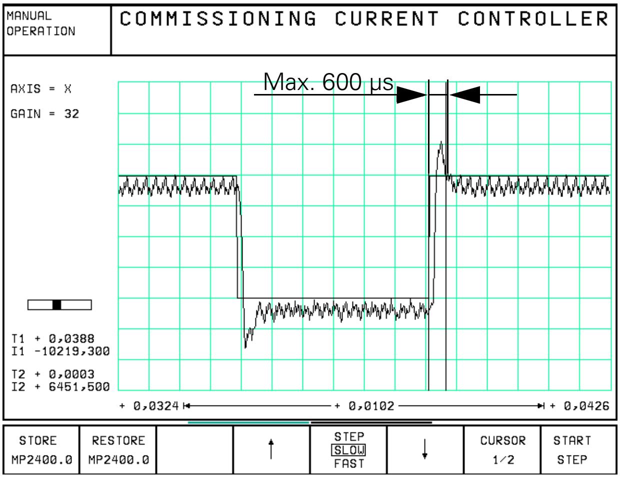
- Switch on the control.
- **Do not** acknowledge the message POWER INTERRUPTED. In the PROGRAMMING AND EDITING mode of operation, enter the code number **688379**.
- Press the I CONTROL soft key.
- In the MANUAL mode of operation, acknowledge the message POWER INTERRUPTED. The PLC program defined in the OEM.SYS file with the "PLCPWM=" command is compiled.
- Switch on the machine control voltage.
- In the OSCILLOSCOPE mode, press the SELECT AXIS soft key to select the axis to be optimized.
- Press the START STEP soft key. This sends a step function to the current controller and measures the step response. Height and length of the step function are calculated by the TNC automatically from the entered machines parameters.
- With the ↓ and ↑ soft keys, change the current gain until the step response shows only a slight overshoot. The settling time t_{out} should be ≤ 600 µs.



MP2400 too small



MP2400 too large



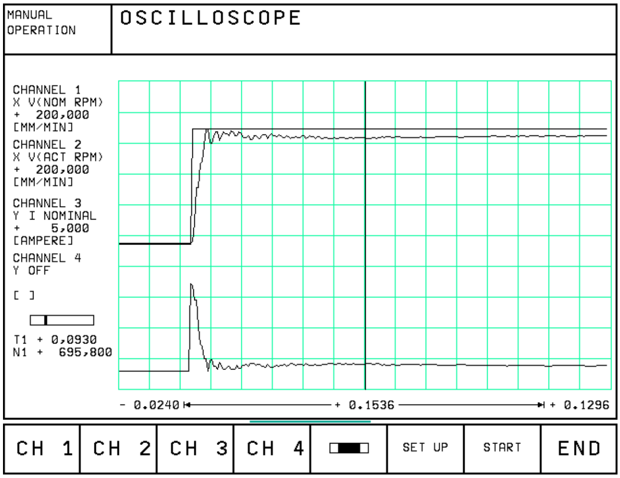
MP2400 optimum

- When the current gain is properly adjusted, press the STORE MP2400.x soft key to transfer the optimized value directly into the machine parameter.
- Press the END key to exit the I CONTROL mode again.

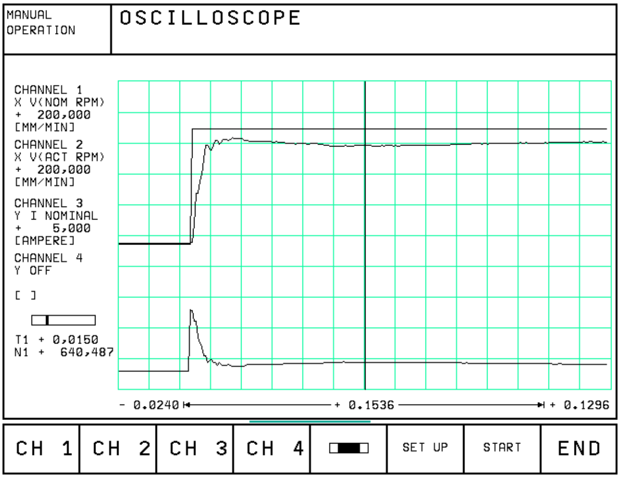
Speed controller

Adjusting the speed controller:

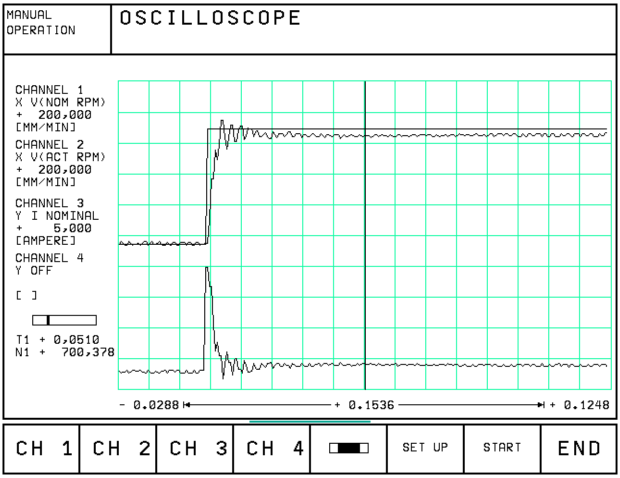
- ▶ Deselect “Pass over reference point” by setting MP1340 = 0.
- ▶ Ensure that the loaded PLC program fulfills the following conditions:
  - Position control loop is closed (W1038/W1040 = 0). If the position controller is not optimized, error messages appear if the position controller is closed.
  - Servo drive controller is enabled (Module 9161)
  - NC stop is inactive, M4560 = 1
  - Axis direction buttons active
  - Axes are 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 at the oscillation limit



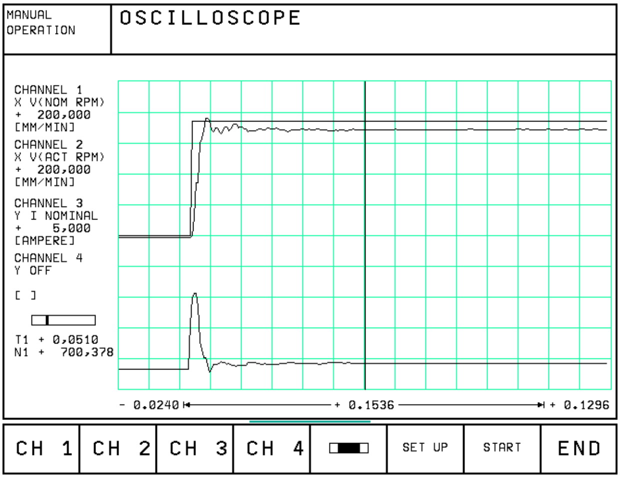
MP2500 too small



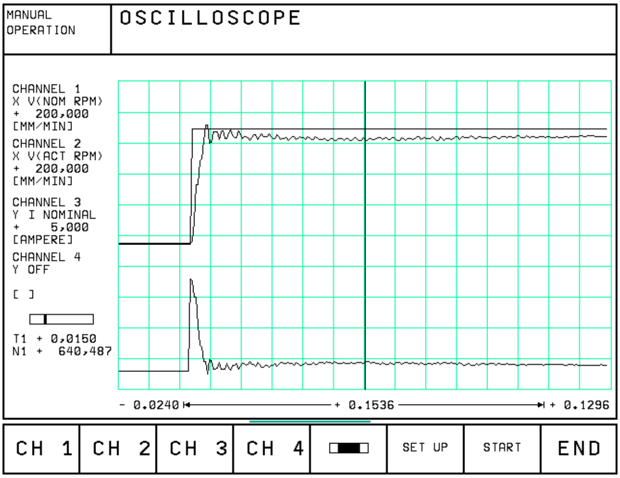
MP2500 too large

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

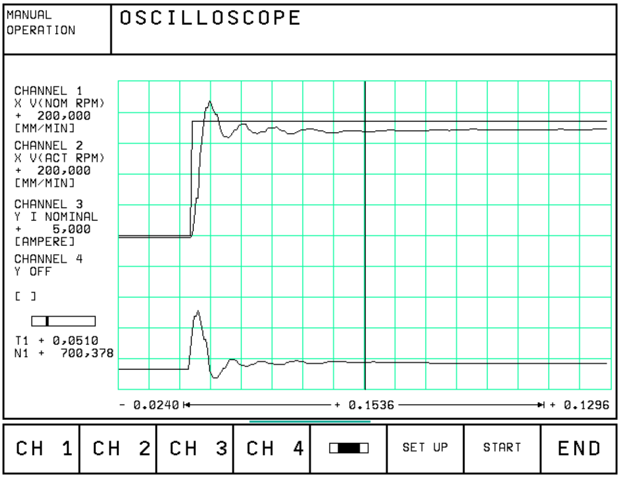
▶ Compensate high-frequency interference oscillations (> 400 Hz) with MP2530.x or MP2560.x.



MP2530 optimum



MP2530 too small



MP2530 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:

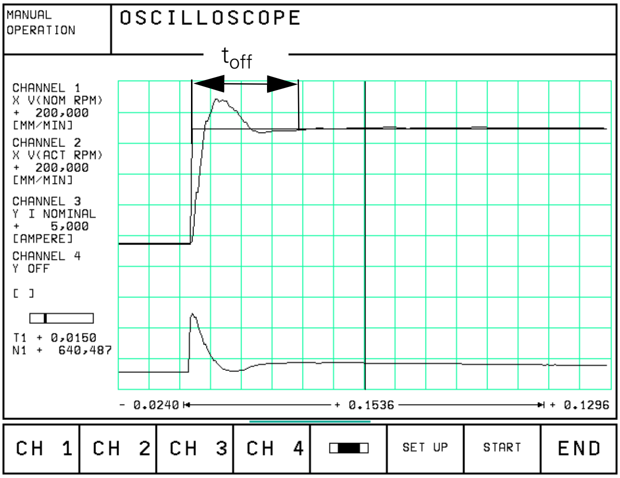
- Calculate the frequency of the oscillation and enter it in MP2550.x.
- Increase the band-rejection filter damping in MP2540.x until the interfering oscillation is minimized. Realistic input values: 3 to 9 [dB]

Note

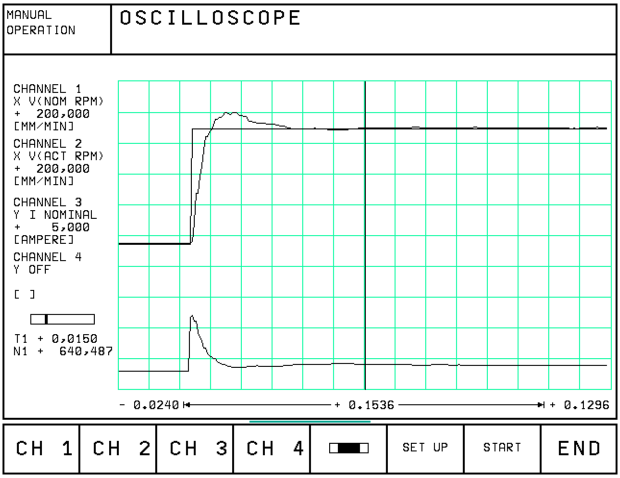
The compensation dampens the control loop. Try first to remove the mechanical causes of the disturbing oscillations.

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).

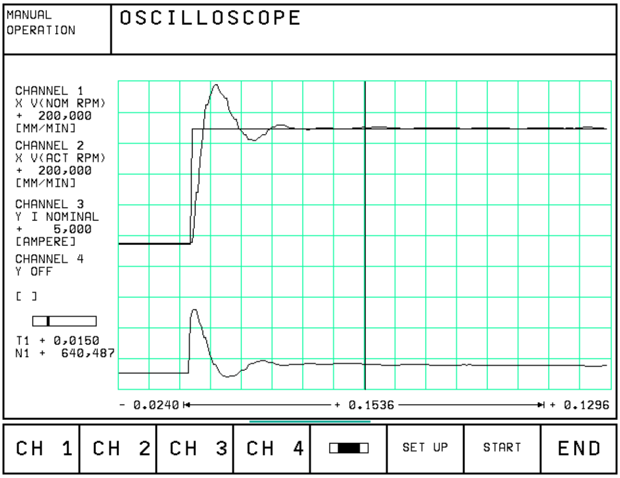
▶ Increase MP2510.x (I factor) until you see one overshoot followed by a slight undershoot and the settling time t_{off} is as small as possible (realistic value: 3 ms to 15 ms)



MP2510 optimum

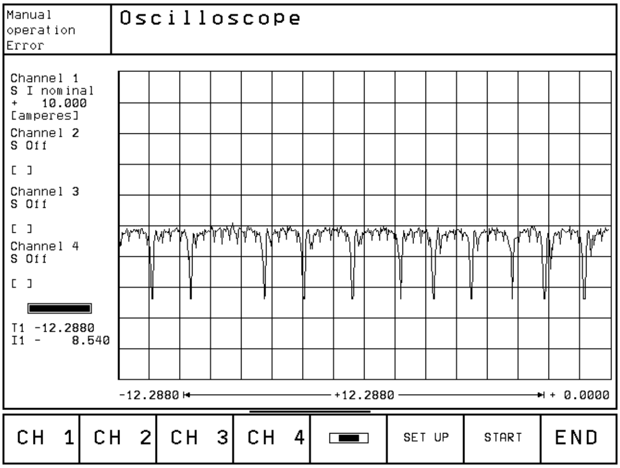


MP2510 too small



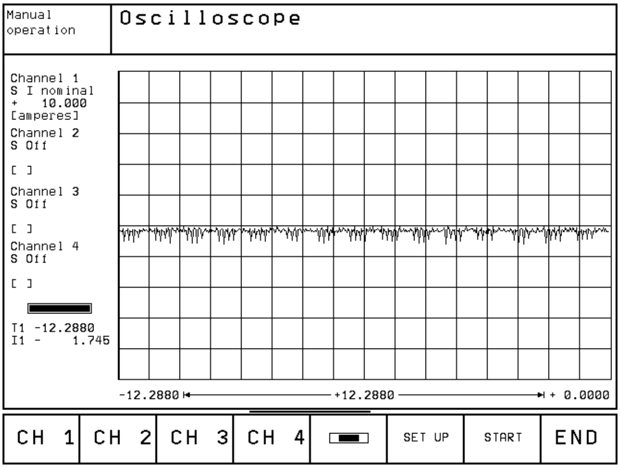
MP2510 too large

The motor might run rough at maximum speed. This can be seen from a continuous readjustment of the current controller (see I NOMINAL on the oscilloscope) and a fluctuation of the utilization display, and it is usually audible. This mostly happens with spindle (asynchronous) motors.



I NOMINAL: Readjustment of the speed

▶ Increase MP2402.x until the motor begins to run smoothly. (Empirical value: MP2402.x = approx. 2...3 * MP2400.x)



MP2403.x optimum

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.

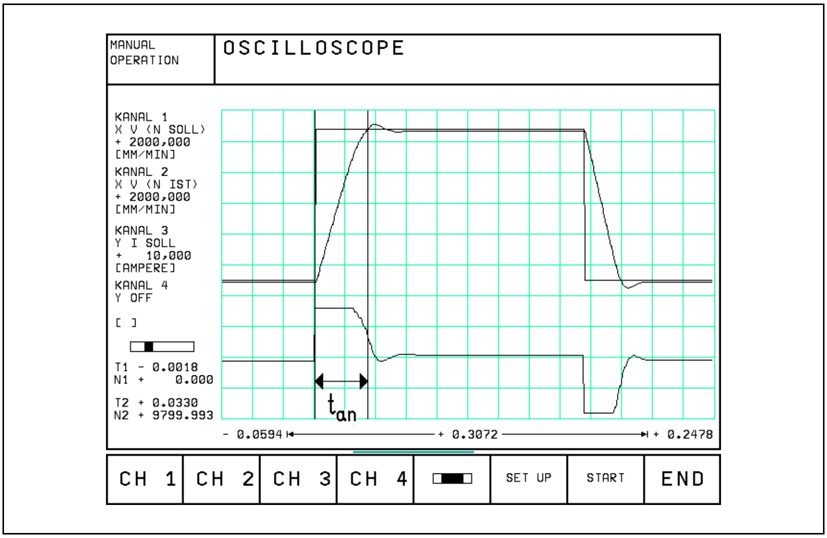
$$a = \frac{F_{max}}{t_{an} \cdot 66\,000}$$

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

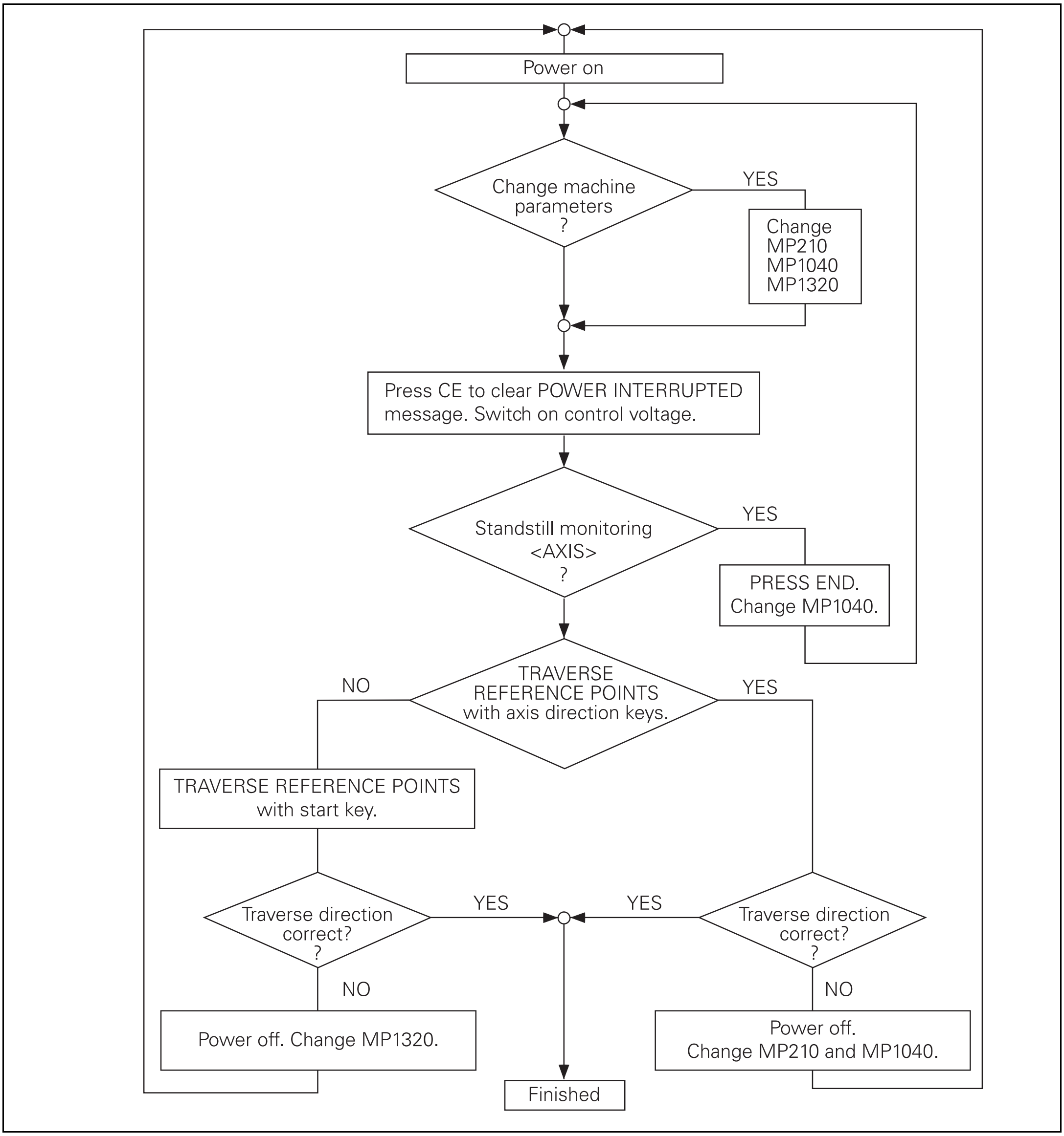
Adjusting the position control loop:

- 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).

1. Check the traversing direction (see flowchart):

- In MP1340.x, enter the sequence in which the reference points are to be traversed.



2. Set the traverse range:

You can enter up to three traverse ranges.
(See "Traverse Ranges” on page 6 – 22). 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.

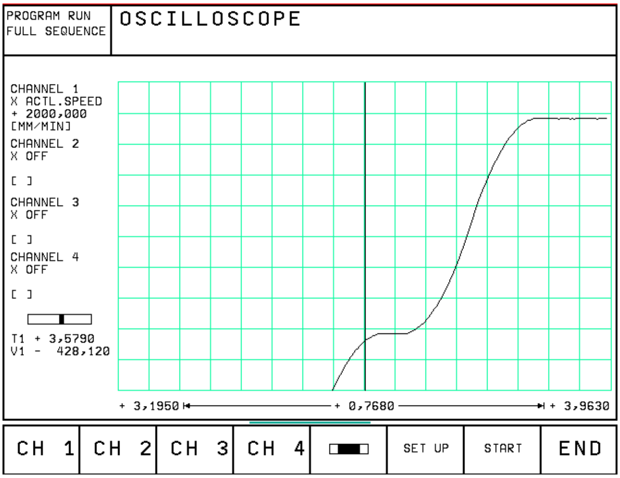
3. In MP1390 or MP1392, select the type of control:

For control with velocity feedforward:

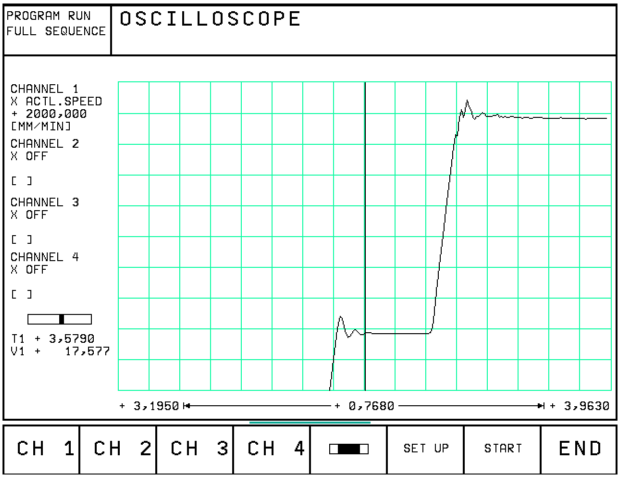
- Enter the temporary input values.

<table>
<tr><th>Machine parameter</th><th>Temporary input value</th></tr>
<tr><td>MP1391 and MP1392</td><td>Set to velocity feedforward control</td></tr>
<tr><td>MP1090.0</td><td rowspan="2">Enter a very small jerk, e.g. 1 (dependent on the machine)</td></tr>
<tr><td>MP1090.1</td></tr>
<tr><td>MP1092</td><td><greater than rapid traverse></td></tr>
<tr><td>MP1095</td><td>0</td></tr>
<tr><td>MP1096</td><td>0</td></tr>
<tr><td>MP1099.0</td><td>5</td></tr>
<tr><td>MP1099.1</td><td>3</td></tr>
</table>

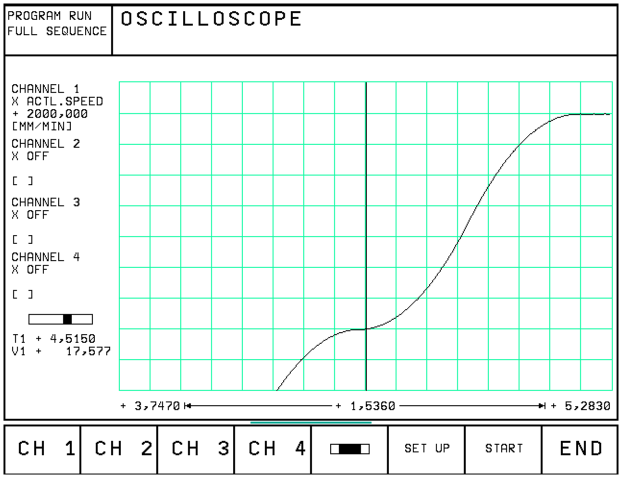
- Enter the following test program:
```
LBL 1
L X <maximum traverse>
R0 FMAX
LX0 FMAX
CALL LBL1 REP 100/100
```
- Display the actual speed (act. speed) with the integrated oscilloscope and, if necessary, also show the following error (lag).
- Start the test program with feed rate override = 100%.
- In MP1090.0 increase the jerk until the overshoot just disappears.



MP1090 optimum

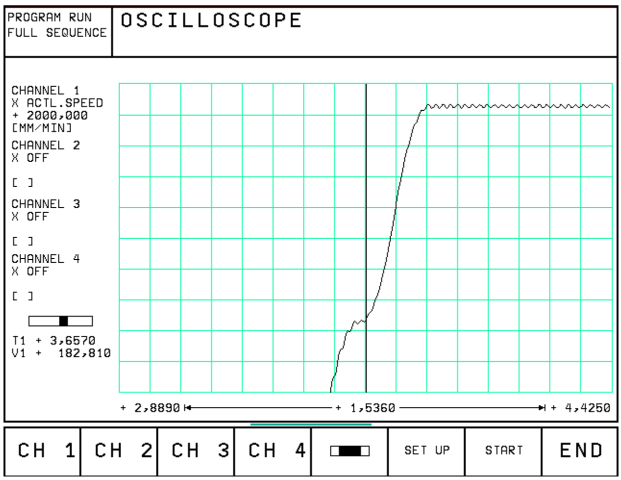


MP1090 too large

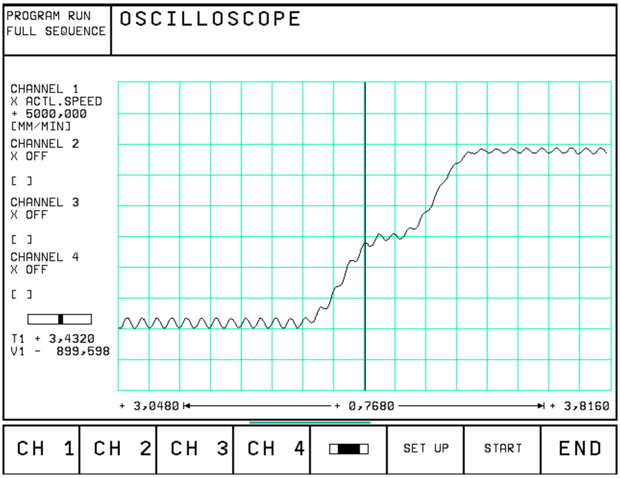


MP1090 too small

- Transfer the resulting jerk value from MP1090.0 to the axis specific parameters MP1097.x and MP1098.x.
- 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 TNC and activate them with M functions (See "The Control Loop" on page 6 – 109). MP1090.x applies to all axes. The worst axis determines the input value.

Procedure:

- Take 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.
- Select from the following tables 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

Single filter (MP1099.0)

Damping [dB]	Frequency to be damped [Hz]										
	10	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**	**55**	**60**
3	10	–	–	–	3	2	2	–	–	–	1
4	12	7	5	4	–	–	–	2	–	–	–
5	13	8	6	–	–	3	–	–	2	–	–
6	14	9	–	5	4	–	–	–	–	2	–
7	15	10	7	–	–	–	3	–	–	–	-
8	16	–	–	6	–	4	–	3	–	–	2
9	17	11	8	–	5	–	–	–	–	–	–
10	18	–	–	–	–	–	–	–	–	–	–
11	19	12	–	–	–	–	4	–	3	–	–
12	–	–	9	7	–	–	–	–	–	–	–

Double filter (MP1099.1)

Damping [dB]	Frequency to be damped [Hz]										
	10	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**	**55**	**60**
3	7	4	3	2	–	–	1	1	–	–	–
4	8	5	–	–	2	–	–	–	1	–	–
5	9	6	4	3	–	2	–	–	–	1	–
6	10	–	–	–	–	–	–	–	–	–	1
7	11	7	5	–	3	–	2	–	–	–	–
8	–	–	–	4	–	–	–	–	–	–	–
9	12	8	–	–	–	–	–	2	–	–	–
10	13	–	6	–	–	3	–	–	–	–	–
11	–	–	–	–	–	–	–	–	2	–	–
12	14	9	–	5	4	–	–	–	–	–	–

HSC filter (MP1094)

Damping [dB]	Frequency to be damped [Hz]										
	10	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**	**55**	**60**
3	12	19	24	29	34	39	44	49	54	59	64
4	10	17	22	27	32	37	42	47	52	57	62
5	6	15	21	26	31	36	41	46	51	56	61
6	1	14	20	25	30	35	39	45	50	55	60
7	–	13	18	23	28	33	38	43	48	54	59
8	–	11	17	23	28	33	38	43	48	53	58
9	–	10	16	22	27	32	37	42	47	52	57
10	–	9	16	21	26	31	36	41	46	51	56
11	–	7	15	20	25	30	35	40	45	50	55
12	–	6	14	19	24	29	34	39	44	49	54

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):

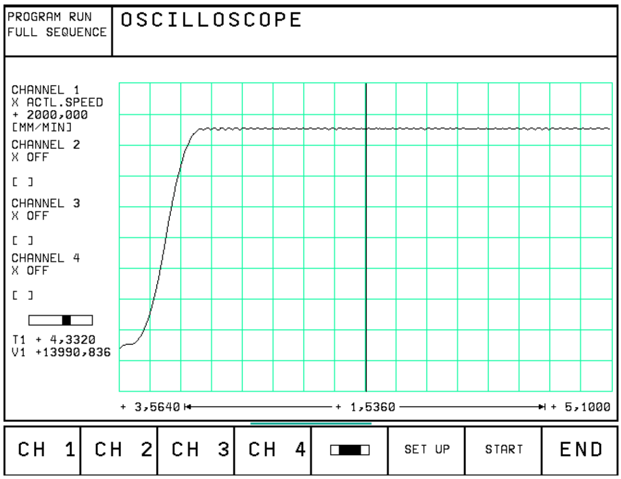
The adjusted maximum jerk works during operation with following error. MP1090 is not changed.

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 (actl. speed) 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 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 TNC (MP1815.x) and activate them with M functions (See "The Control Loop” on page 6 – 109).

Procedure for defining a characteristic curve kink point:

- Specify the k_V factor for rapid traverse (characteristic curve kink point):

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

MP1820.x = 1

Set to operation with following error.

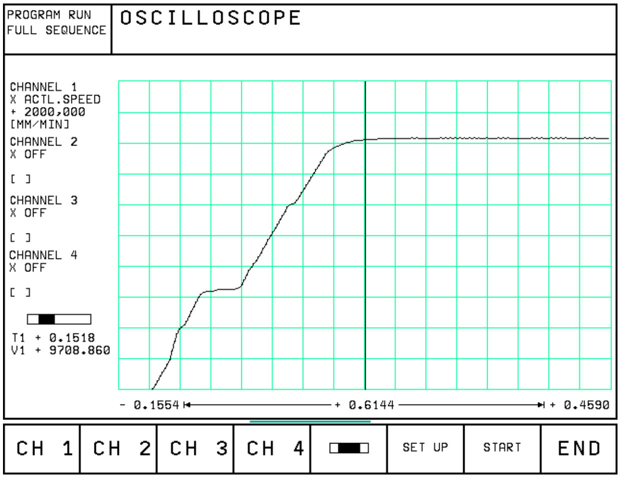
▶ Enter the following test program:

```
LBL2
L X<maximum traverse> R0 FMAX
L X0 R0 FMAX
CALL LBL1 REP 100/100
```

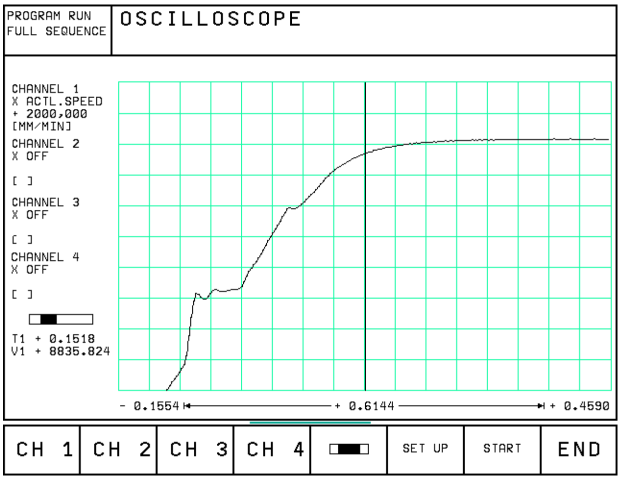
▶ Start the test program.

▶ Display the actual feed rate (actl. speed) with the internal oscilloscope:

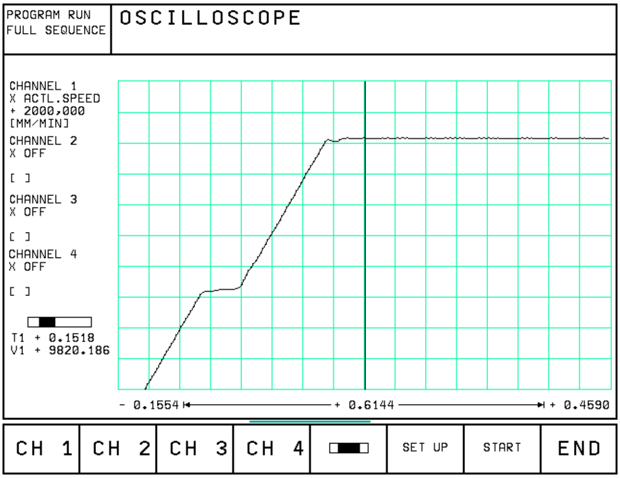
- If no oscillations are recognizable, no kink point is required.
- If oscillations are visible, you must reduce MP1820.x until the oscillations have disappeared.



MP1820 optimum



MP1820 too small



MP1820 too large

4. Switch on the nominal position value filter:

- In MP1096, enter a defined tolerance (e.g. 0.02 mm).

5. Activate monitoring functions:

> **Note**
>
> To ensure that the monitoring functions become effective at the right moment, you must enter meaningful values.

HEIDENHAIN recommends the following input values. You must change these values slightly to adapt them to the design of the machine.

MP	Temporary input value	Meaning
MP1030.x	0.01 mm	Positioning window
MP1110.x	2 · MP1030.x	Standstill monitoring
MP1140.x	0.03 [1000 rpm]	Movement monitoring
MP2800.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)

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 ACTL. SPEED and V (ACT RPM). At the reversal point the actual feed rate follows the actual shaft speed by the time delay *t*.

▶ Set the machine parameters:
- MP750 = $t \cdot \Delta$V ACTL (Keep in mind the units for *t* and ΔV ACTL)
- MP752 = approx. 20 ms (determined in test)

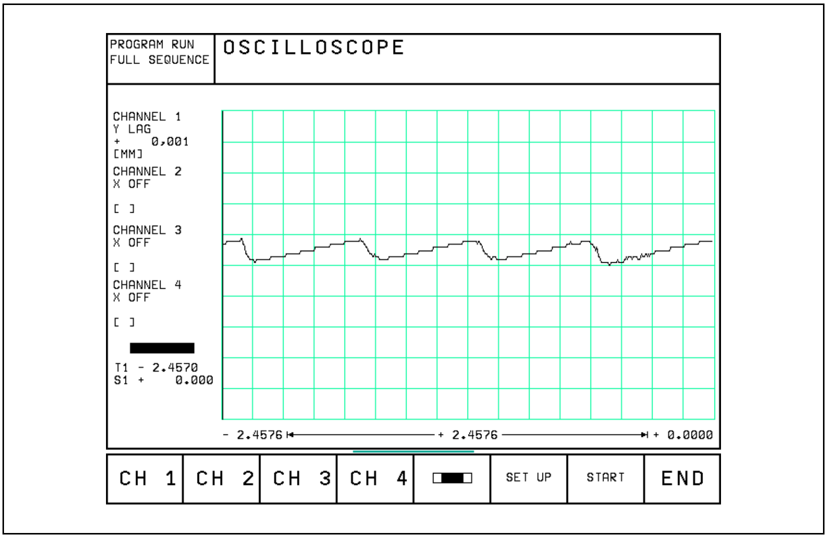
$$\Delta V\ ACTL = |V\ ACTL - V\ (ACTL\ RPM)|$$

7. Compensate the static friction:

- Enter the backlash, if any exists.
- Enter the following test program (static friction in the Y axis):

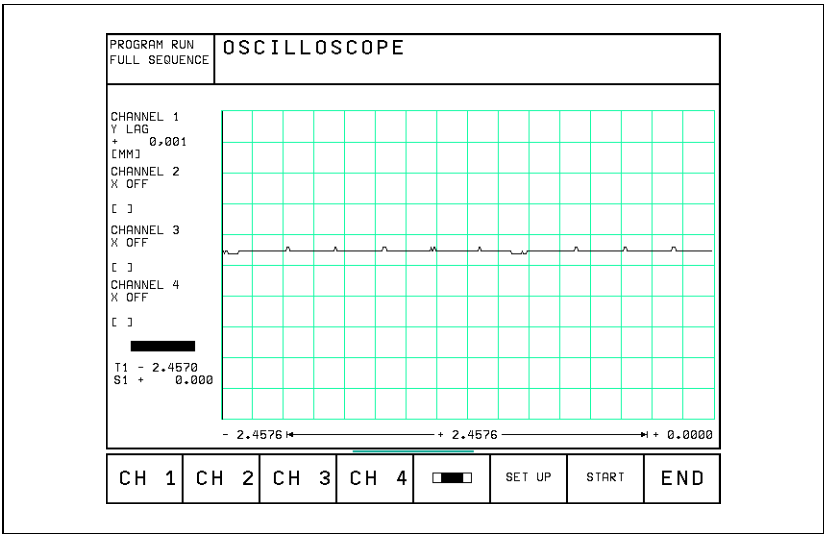
```
LBL 1
L X+400 IY+0.5 R0 F200
L X0 IY+0.5 R0
CALL LBL1 REP 100/20
```

- 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.

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.

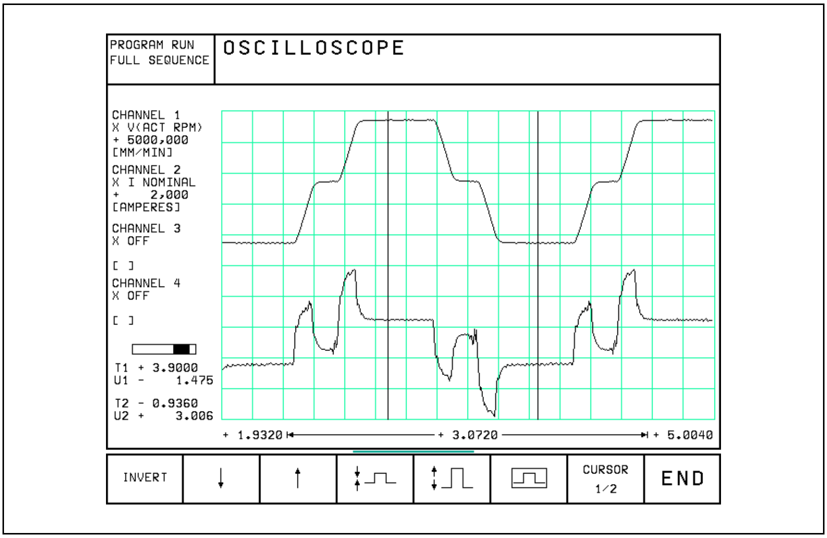
▶ Increase MP2512.x until the axis remains stationary.

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 (ACTUAL RPM) and the nominal current value (I NOMINAL).
- Start the program.
- With the feed rate override knob, adjust the motor speed to ± 10 rpm (MP2020.x).
- Determine the current (I NOMINAL) in both directions of rotation.



- Calculate MP2630.x:

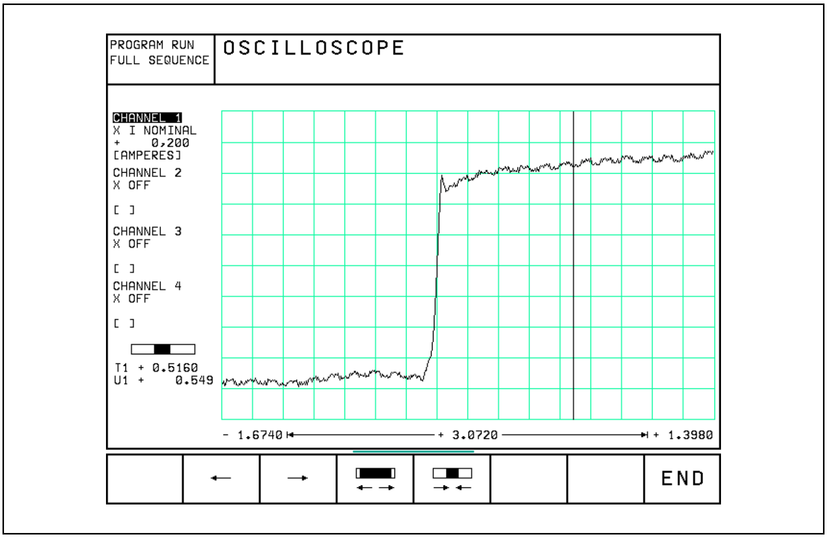
$$MP2630.x = \frac{I\ NOML_1 + I\ NOML_2}{2}$$

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
```

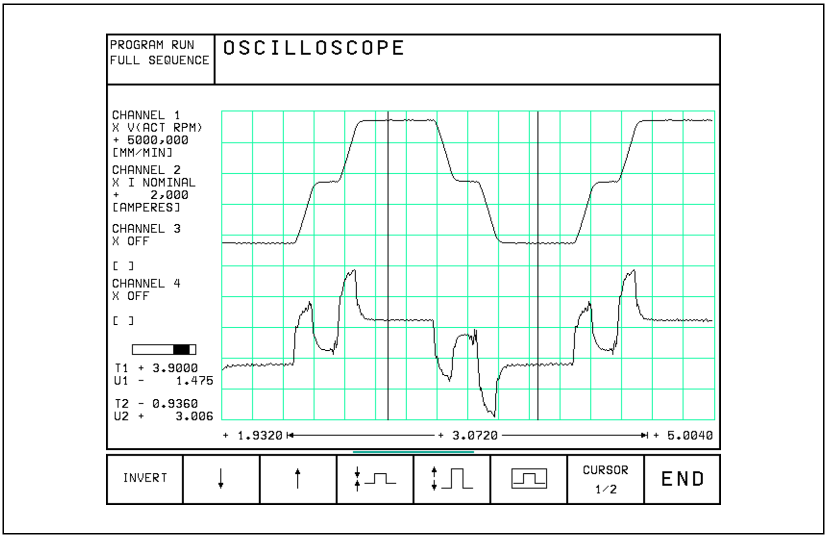
- Use the integrated oscilloscope to record the actual shaft speed (ACTUAL RPM) and the nominal current value (I NOMINAL).
- Start the test program.
- With the feed rate override knob, adjust the motor speed to 10 rpm (MP2020.x).
- Determine the current (I NOMINAL) in both directions of rotation.



- Calculate MP2610.x:

$$\text{MP2610.x} = \frac{\left|\text{I NOML}_1 - \text{I NOML}_2\right|}{2}$$

- 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:

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

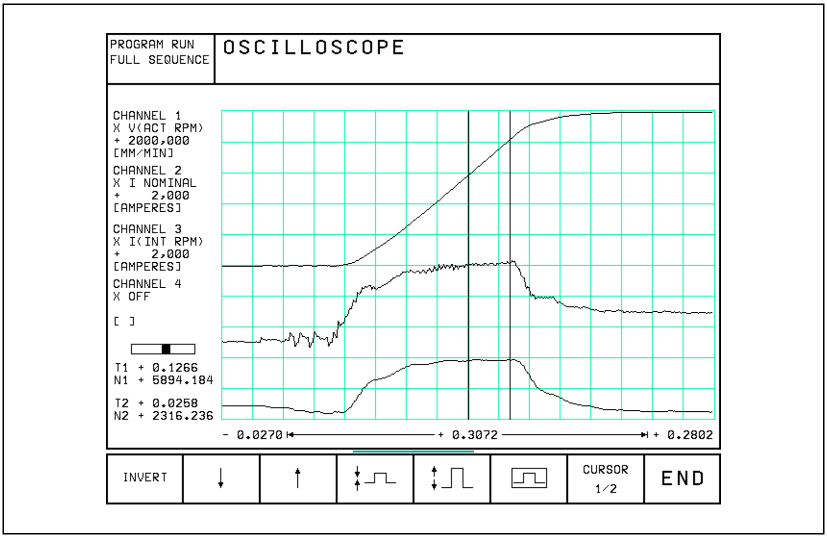
I_{nmax}: Current at rapid traverse

n_{max}: Shaft speed at rapid traverse

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 (ACTUAL 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:

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

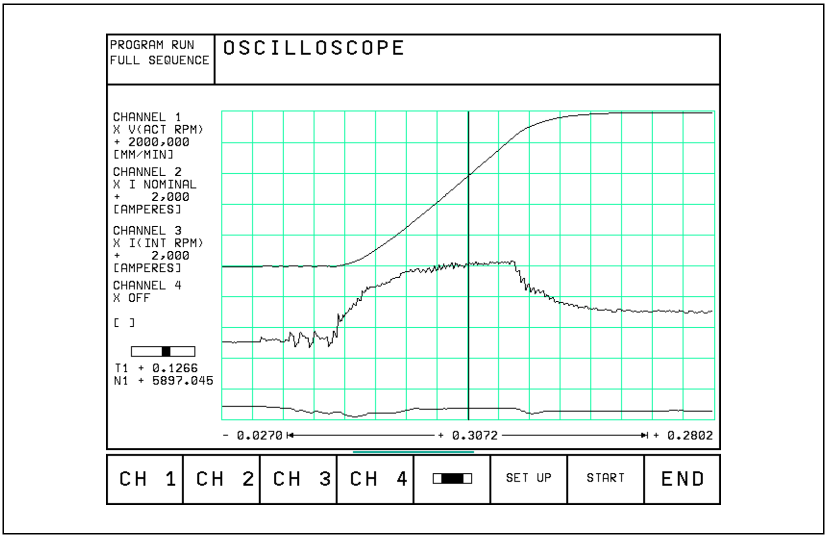
I (INT RPM) = Integral-action component of the nominal current value

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

ΔV (ACT RPM) = Change of actual rpm during *t*

MP2020.x = Traverse distance per motor revolution

- Repeat this measurement to check the input value of MP2600.x. I (INT RPM) must have approached zero.



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.

6.24.3 Analog Axes

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

Please note:

Note

For analog axes, you must adjust the servo amplifier before you optimize the position controller.

Procedure:

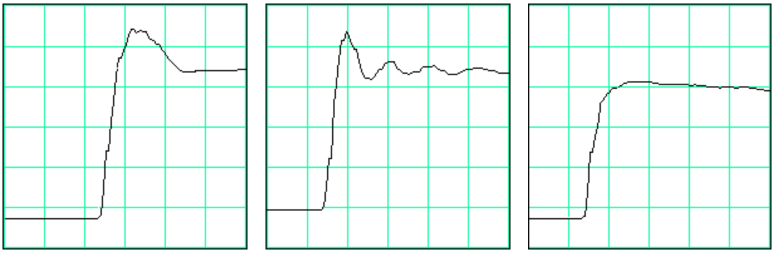
▶ Disconnect the nominal-value connection between the servo amplifier and the logic unit.
▶ Short-circuit the nominal value input on the servo amplifier.
The input must have a 0 V voltage.
▶ Activate control enabling at the servo amplifier.
▶ Connect the supply voltage to the servo amplifier.
▶ Perform 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 logic unit.

▶ 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 U 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

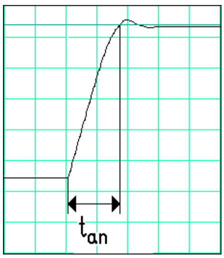
After adjusting the servo amplifier, you can determine from the step response the maximum possible acceleration:

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

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

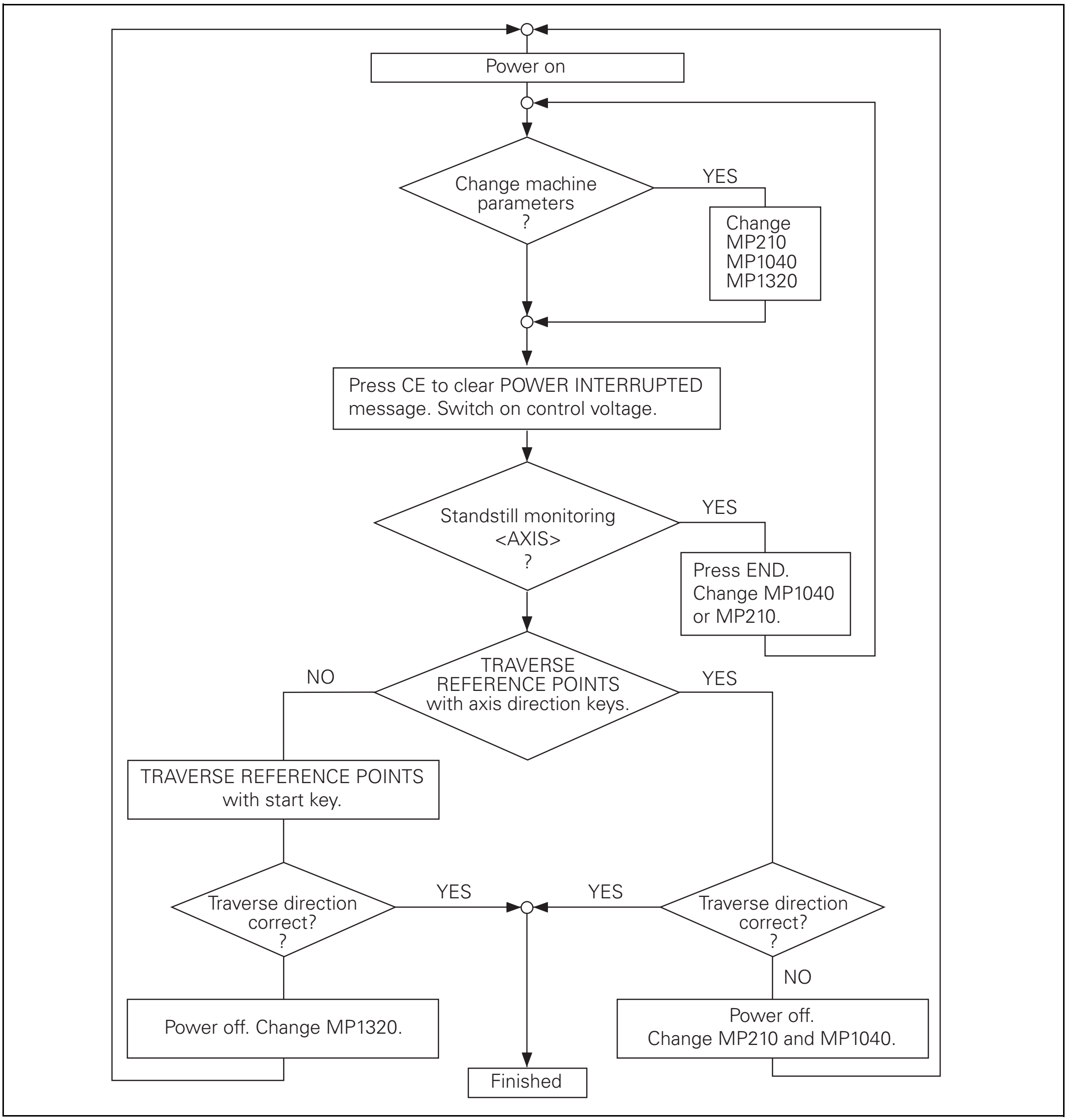
Please note:

> **Note**
>
> For analog axes, you must adjust the servo amplifier before you optimize the position controller.

Adjusting the position control loop:

- Activate a PLC program that is adapted to the machine.
- Ensure that the position control loop is closed (W1038/W1040) and all inputs/outputs are properly operated.
- To optimize the position control loop take the following steps:

1. Check the counting/traversing direction
(see flowchart)



2. Set the traverse range
Same procedure as for digital axes.

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.

4. Perform an offset adjustment
On the TNC: (See "The Control Loop" on page 6 – 109)

5. Activate monitoring functions:

- Enter the following temporary input values when you begin: (See "Digital Axes" on page 6 – 462).

6. Compensate the backlash
Same procedure as for digital axes.

7. Compensate the static (stick-slip) friction
Same procedure as for digital axes.

6.24.4 Digital Spindle for TNC 426 without Spindle DSP

Temporary input values

▶ Enter the following temporary input values when you begin:

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.

Adjust the current controller

Same procedure as for digital axes, with one exception:

▶ You must adjust MP2401 instead of MP2400.x.

Speed controller

Define the step function:

▶ In MP3411.x, enter the maximum acceleration and start the step by switching the spindle on.

▶ Set the following machine parameters:

- MP2501 = 2: Proportional factor of the speed controller
- MP2511 = 1: Integral factor of the speed controller
- MP2521 = 0: Differential factor of the speed controller
- MP2531 = 0: PT_2 element of the speed controller

▶ 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).

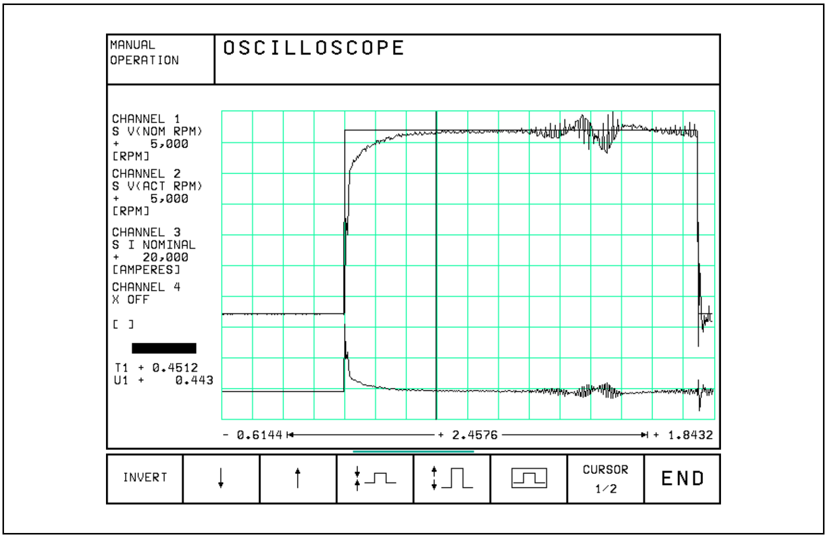
▶ Output a step by activating the spindle-on function (M03/M04).

▶ 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 (MP2501) until the system oscillates or no change is visible.

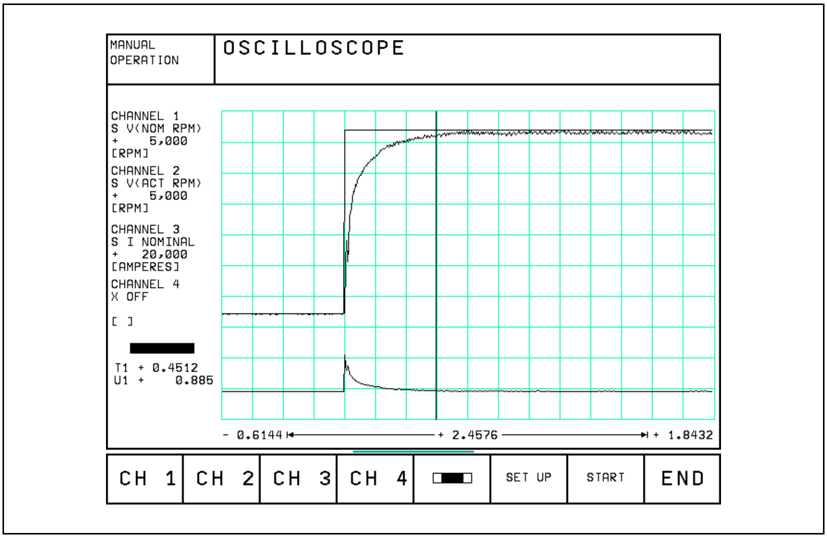
You can modify the machine parameters:

▶ In the setup menu, press the MP EDIT soft key.

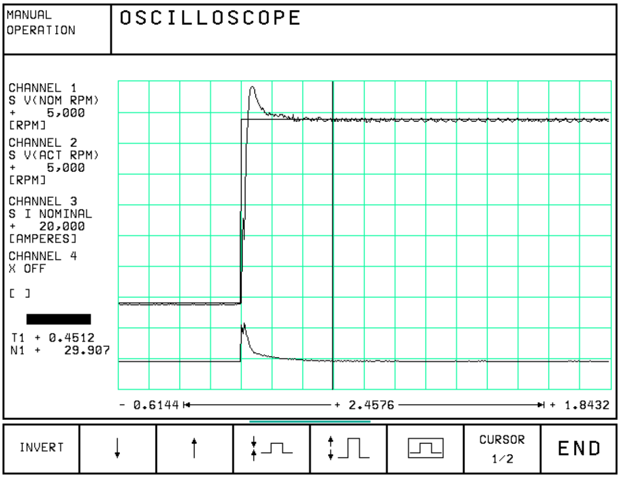


▶ Calculate MP2501:

MP2501 = MP2501 · 0.6

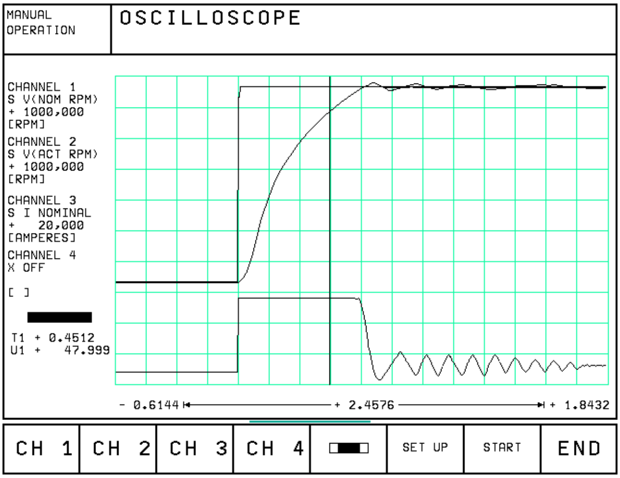


▶ Increase the I factor (MP2511) until you see one overshoot followed by a slight undershoot.

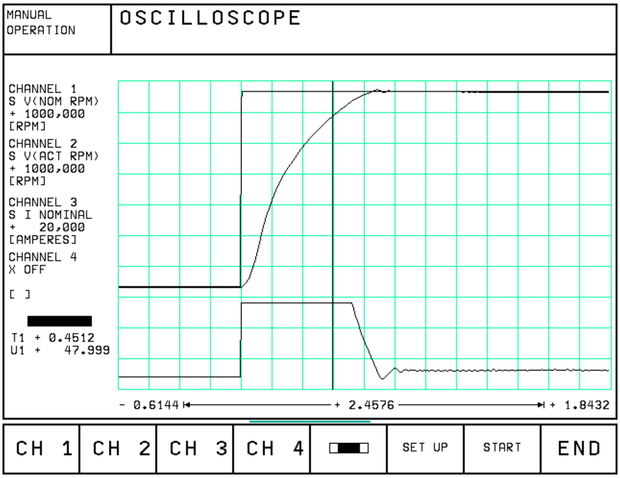


▶ Output the step with maximum shaft speed. I NOMINAL is within the limitation during acceleration. I NOMINAL must not oscillate after reaching the maximum speed. If I NOMINAL oscillates:

- Reduce MP2501 and MP2511 evenly until the overshoots are minimized.



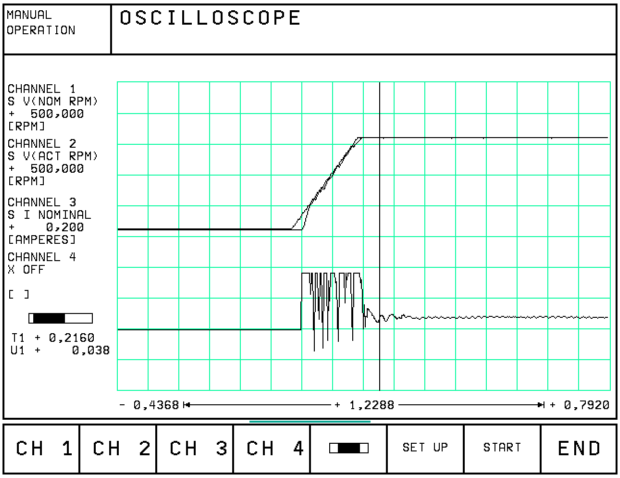
I NOMINAL oscillating



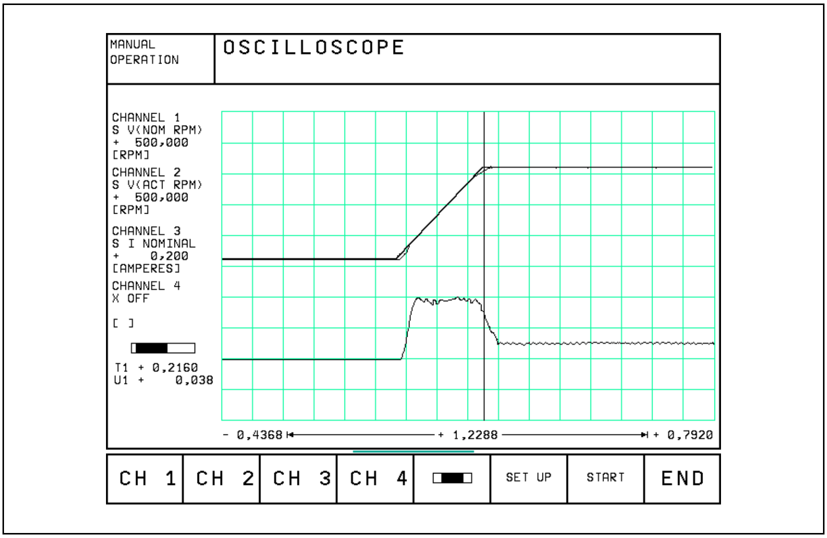
Only one overshoot

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 – 201).
  - 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 .

Higher current gain beginning with the rated speed

The counter EMF increases with increasing shaft speed. Therefore a higher current gain is needed at higher shaft speeds:

▶ With the integrated oscilloscope, record V (ACT RPM) and activate a shaft speed greater than the rated speed.

- If V (ACT RPM) “pumps” only at high shaft speeds, increase MP2403 until the spindle runs quietly.
- If V (ACT RPM) oscillates even at low shaft speeds, the problem lies with resonance oscillations: You can compensate these oscillations in the same way as for the axes, i.e., with the differential factor, low-pass filter or the PT_2 element.

6.24.5 Digital Spindle for TNC 430 / TNC 426 with Spindle DSP

Temporary input values

▶ Enter the following temporary input values when you begin:

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.

Wye/delta connection

During commissioning you can use soft keys to switch between wye (star) and delta connections. 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 wye and delta connection.
- If you do not use the delta connection, set the corresponding machine parameters to zero.

Current controller

Adjusting the current controller:

▶ Switch on the control.

▶ Do **not** acknowledge the “Power Interrupted” message. In the PROGRAMMING AND EDITING mode of operation, enter the code number **688379** with the MOD key. The integrated oscilloscope is started.

▶ Press the I CONTROL soft key.

▶ In the MANUAL mode of operation, acknowledge the “Power Interrupted” message. The PLC program defined in the OEM.SYS file with the “PLCPWM=” command is compiled.

▶ In the OSCILLOSCOPE mode, press the SELECT AXIS soft key to select the spindle.

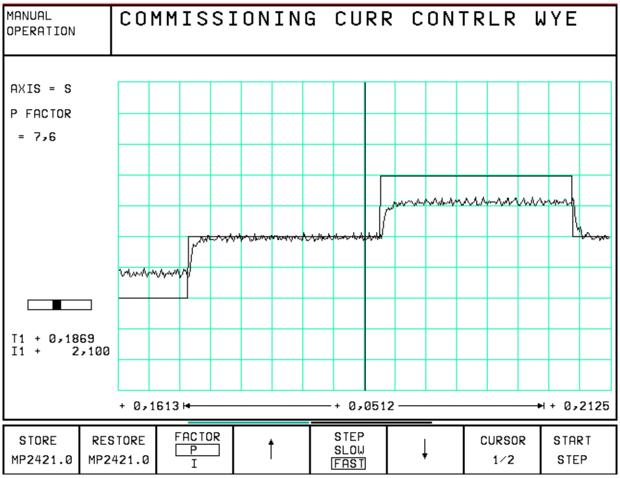
▶ With the STAR / DELTA soft key, select either the wye or delta connection.

▶ With the I FACTOR / P FACTOR soft key, select the I factor and set MP2431.x = 0.

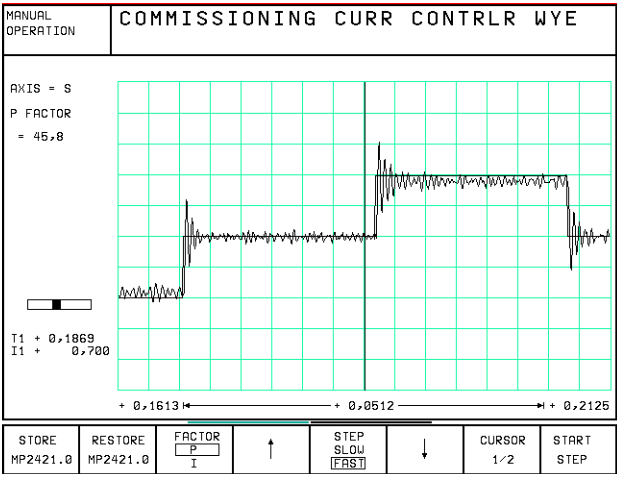
▶ With the I factor / P factor soft key, select the P factor.

▶ 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 TNC.

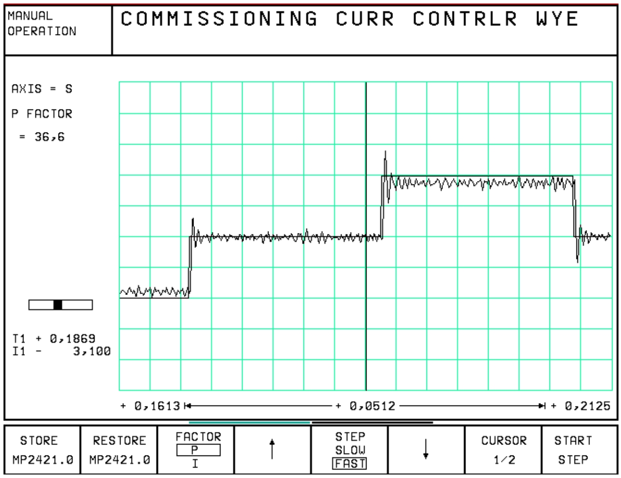
▶ With the ↑ soft key, increase the P factor (MP2421.x) to the oscillation limit.



MP2421 too small



MP2421 too large

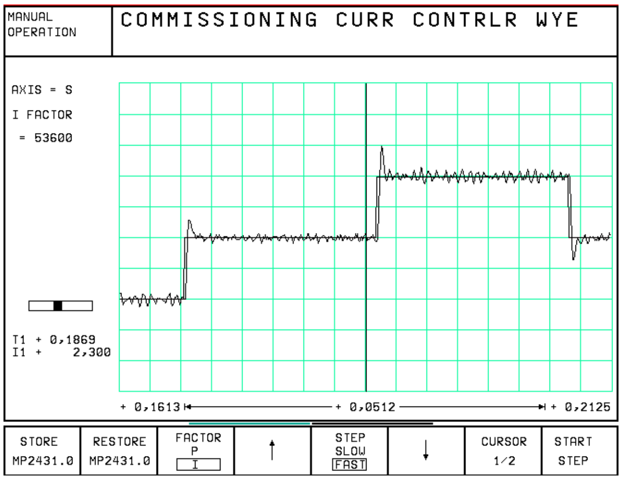


MP2421 at the oscillation limit

- Calculate MP2421.x:

MP2421.x = <determined value> · 0.6:

- Set this value and confirm it with the STORE MP2421.X soft key.
- With the I FACTOR / P FACTOR soft key, select the I factor.
- With the ↑ soft key, increase the I factor (MP2431.x) until you see one overshoot but no undershoot.



MP2431 optimum

- Confirm this value with the STORE MP2431.X soft key.
- Switch-off the machine to exit the I CONTROL mode.
- Press END.

Speed controller

Same procedure as for TNC 426 digital spindle.

Acceleration

Same procedure as for TNC 426 digital spindle.

Direction of rotation

Same procedure as for TNC 426 digital spindle.

Position controller

Same procedure as for TNC 426 digital spindle.

6.24.6 Analog Spindles

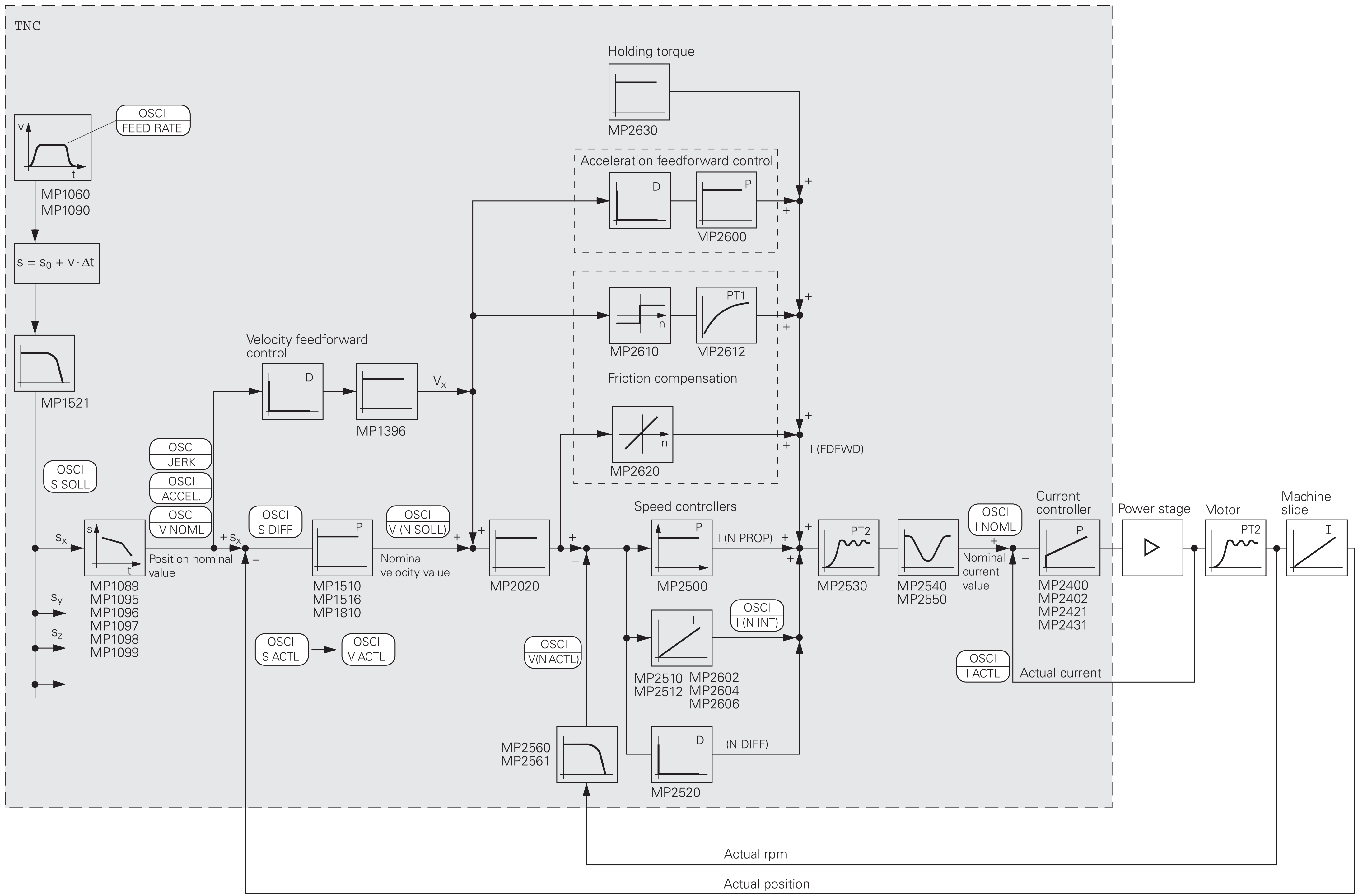
Adjusting the servo amplifier
Same procedure as for analog axes.

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.

6.25 Block Diagram TNC 426, TNC 430



7 PLC Programming

7 PLC Programming

7.1 PLC Functions

The integrated PLC of the TNC 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.

To prevent errors in the PLC program, the TNC operates a syntax monitor during program input. The TNC assists you with the COMPILE function, which checks the PLC program for logical errors, and the TRACE and TABLE functions, with which you can check the condition of the operands.

The process memory runs a compiled PLC program up to a certain size (as of 280 474-10: 512 KB; 280 472-xx: 256 KB; 280 470 470-xx: 128 KB). Every 21 ms—the PLC cycle time—the TNC begins a new PLC scan, i.e. every 21 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. For a PLC cycle time of 21 ms, a PLC scan must not take more than 10 ms.

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: 7)

7.1.1 Selecting the PLC Mode

In PLC mode you can:

- Create (and test) PLC programs
- Write PLC error messages
- Write dialog texts for OEM cycles
- Create help files
- Create compensation value lists for nonlinear axis error compensation

Select PLC Mode:

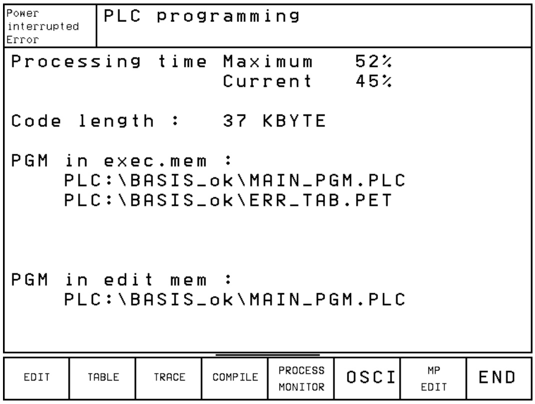
- Select the Programming and Editing mode of operation.
- Press the MOD key.
- Enter the code number 807 667 and confirm your entry with the ENT key, or if you already entered the code number, press the PLC EDIT soft key.

Exit PLC mode:

- Press the END hard key or soft key.

7.1.2 PLC Main Menu

After you have entered the code number (or pressed the PLC EDIT soft key) the TNC displays the PLC main menu:



`Processing time maximum:`
Maximal run time of the PLC program.

The processing time of the PLC (time for one PLC scan) is given as a percentage: 100 % is a run time of 3.5 ms.

The maximum run time of the sequential program must not exceed 300 % (=10.5 ms). If it is higher, the TNC outputs the blinking error message **`PLC: time out.`**

`Processing time current:`
The time taken for the latest PLC scan in %.

`Code length:`

Length of the translated sequential program in KB. Maximum value: 512 KB.

`PGM in exec.mem:`
Name of the last compiled PLC program (program in executive memory).

After switch-on, the TNC automatically compiles the program that was last selected as sequential program. The PLC program is not active until it has been compiled!

`PGM in edit mem:`
Name of the file in main 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	Load PLC program into main memory for editing.
TABLE	Check logical states (M/I/O/T/C), display contents (B/W/D). See “The Table Function” on page 7 – 11
TRACE	Display TRACE function or logic diagram. See “The TRACE Function” on page 7 – 7
COMPILE	Compile PLC program. See “COMPILE Function” on page 7 – 13
PROCESS MONITOR	Process monitor See “Process monitor” on page 7 – 165
OSCI	Activate the integrated oscilloscope. See “Integrated Oscilloscope” on page 6 – 457
MP EDIT	Display list of machine parameters.
END	Exit PLC mode.

7.1.3 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 TNC 426/TNC 430). If in PLC mode you press the PGM MGT key, the TNC displays the TNC partition as well as the PLC partitions at the upper left of the screen.

Differences from file management of NC part programs

The following are file types displayed by the TNC when you press the FILE TYPE soft key:

Soft key	Function
SHOW ALL	Show all files.
.PLC FILES	Show only PLC programs (*.PLC).
SHOW .A	Show only ASCII files (*.A).
.HLP FILES	Show only help files (*.HLP).
.SYS FILES	Show only system files (*.SYS).
.COM FILES	Show only compensation value tables (*.COM).
.CMA FILES	Show only tables with compensation value assignments (*.CMA).
.PET FILES	Show only tables with PLC error messages (*.PET).
END	Back to previous menu.

With the MORE FUNCTIONS soft key you select the following functions:

Soft key	Function
CONV. PLC CROSSREF	Generates a list of cross references. You cannot use a file name extension already used by the TNC (. PLC, .CMA etc.).
CONV. PLC LISTING	Generates a list of programs, numbered by line.
CONV. PLC LISTING+ CROSSREF	Generates a program list and cross reference list, numbered by line.

7.1.4 The TRACE Function

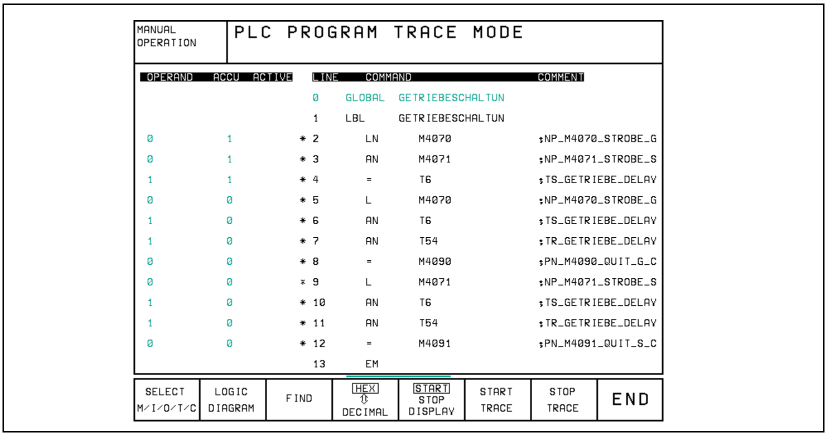
With the TRACE function you can:

- Control the logical states of markers, inputs, outputs, timers, and counters
- Check the content of bytes, words, and double words

You can select the trace function with the TRACE soft key from the PLC main menu. The TNC displays:

- The statement list (STL) of the selected PLC program
- The content of the operand and accumulator in hexadecimal or decimal code for each program line (selectable via soft key)

The TNC identifies every cyclically executed command with an asterisk: *****.
With the arrow keys or the GOTO function you can select the program section that you would like to see 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 M/I/O/T/C	Select **M**arkers/**I**nputs/**O**utputs/**T**imers/**C**ounters for a logic diagram.
LOGIC DIAGRAM	Show logic diagram.
FIND	Search for text in statement list (TRACE IN CODE).
HEX ⇕ DECIMAL	Show operand or accumulator contents in hexadecimal or decimal.
START STOP DISPLAY	Start and stop the dynamic display of the operand contents, accumulator contents, and logic diagram with STOP DISPLAY and START DISPLAY .
START TRACE	Start trace.
STOP TRACE	End trace.
END	Back to previous menu.

7.1.5 The Logic Diagram

Soft keys in 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	Show trace in code.
SAVE TRACE BUFFER	Save current logic diagram in an ASCII file (*.A)
RESTORE TRACE BUFFER	Display saved logic diagram
START STOP DISPLAY	Start and stop the dynamic display of the operand contents, accumulator contents, and logic diagram with STOP DISPLAY and START DISPLAY.
START TRACE	Start trace.
STOP TRACE	End trace.
END	Back 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 TNC 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 TNC asks per dialog for the individual positions in the table. To delete incorrect entries, simply press DEL.

You can enter a trigger condition for each operand. The TNC records 512 states both before and after a trigger event. The following are possible trigger conditions:

- **1:** Record if operand is logical 1 (trigger on positive edge).
- **0:** Record if operand is logical 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 TNC records the states of the operands continuously. The memory holds the most recent 1024 states at any given time.

To start recording:

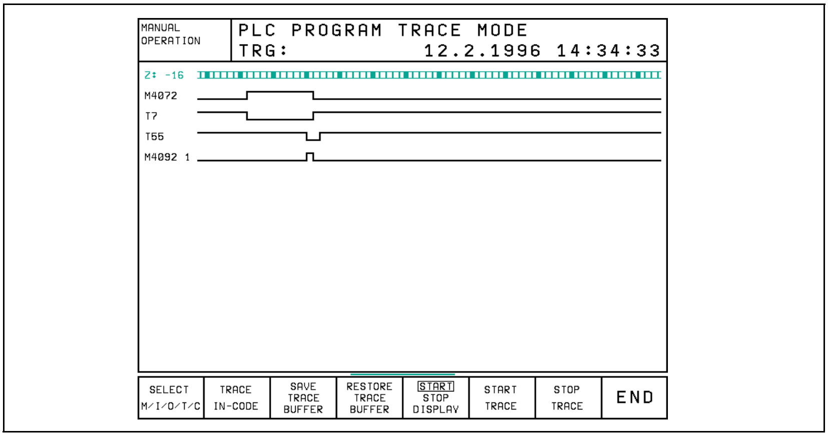
▶ Press the soft key START TRACE.

To stop recording:

▶ Press the STOP TRACE soft key, or the TNC terminates recording automatically as soon as the trigger event occurs.

The "PCTR" indicator blinks in the status window as long as the TNC 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 a logic diagram:



7.1.6 The Table Function

From the main menu, choose TABLE to select the table of markers, inputs, outputs, timers, counters, bytes, words, double words and strings in order to show their 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 in the table function (first soft-key row):

Soft key	Function
SET	Set the selected operands.
RESET	Reset the selected operands.
MARKER	Show list of markers.
INPUT	Show list of inputs.
OUTPUT	Show list of outputs.
COUNTER	Show list of counters.
TIMER	Show list of timers.
END	Back to previous menu.

Soft keys in the table function (second soft-key row):

Soft key	Function
BYTE	Show list of bytes.
WORD	Show list of words.
DOUBLE	Show list of double words.
HEX ⇕ DECIMAL	Show contents of operands in hexadecimal or decimal.
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.
END	Back to previous menu.

7.1.7 COMPILE Function

A newly written PLC program does not become effective until it is compiled and thereby transferred to the process memory. The name of the compiled program then appears in the main menu next to **PGM IN EXEC.MEM.**

To compile a PLC program:

- Press the COMPILE soft key. The TNC displays an overview of existing PLC programs.
- With the cursor keys, select the PLC program to be compiled.
- Press the SELECT soft key.

If the TNC displays error messages: See “Error Messages” on page 10 – 3

7.2 Operands

7.2.1 Overview of Operands

Operand	Symbol	Address range
Marker	M	**M0 to M4999** **M0 to M999** free, are deleted only after entry of the code number 531210, not by a reset. **M1000 to M3999** free, are deleted upon reset **M4000 to M4999** reserved for NC/PLC interface
Input	I	**I0 to I31 (LE); 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 (LE)** **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 C31** Counter contents: **C48 to C79** Counter pulse release: **C96 to C127**
Timers	T	Timer start: **T0 to T47** Timer is running: **T48 to T95 and T96 to T303**
Byte Word Double word	B W D	**B0 to B4095** (8 bits) **B0 to B127** free, are deleted only after entry of the code number 531210, not with a reset **B128 to B2047** reserved for NC/PLC interface **B2048 to B4095** free, are deleted by a reset.
Constant	K	**-2 147 483 647 to +2 147 483 647**
String	S	**S0 to S15**

7.2.2 Operand Addressing (Byte, Word, Double Word)

The memory for operands B (8 bits), W (16 bits), D (32 bits) is only 8 bits wide. Since the operands can be 8, 16 or 32 bits wide, an overlapping of the memory area will occur, which you must take into account when addressing memory:

Double word	Word	Byte	Memory	Word address	Double-word address
D0	W0	B0	8 bits	High byte	Highest byte
		B1	8 bits	Low byte	
	W2	B2	8 bits	High byte	
		B3	8 bits	Low byte	Lowest byte
D4	W4	B4	8 bits	High byte	
		B5	8 bits	Low byte	
• • •	• • •	• • •	• • •	• • •	• • •
D1020	W1020	B1020	8 bits	High byte	Highest byte
		B1021	8 bits	Low byte	
	W1022	B1022	8 bits	High byte	
		B1023	8 bits	Low byte	Lowest byte

During byte addressing every address, during word addressing every second address, and during double-word addressing every fourth address from 0 to 4092 is accessible. The address parameter indicates the high byte of the word address (W) and the highest byte of the double-word address.

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).

7.2.3 Timers

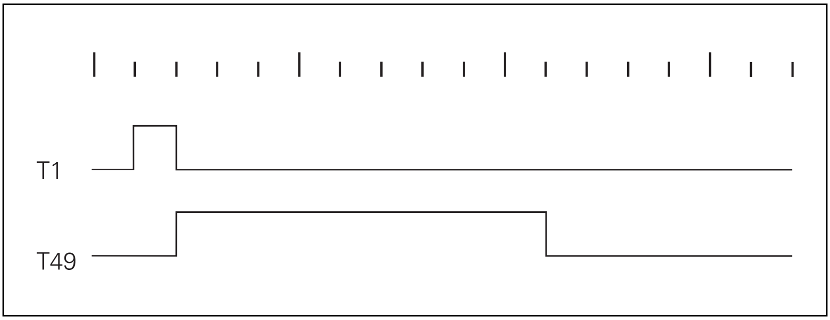
The PLC has over 256 timers, which you control through special markers with the symbol T. You must define the cycle time of timers T0 to T47 in machine parameter MP4110.x. One time unit (input value 1 in MP4110.x) corresponds to the PLC cycle time.

You can start the first 48 timers by setting one of the timers T0 to T47 for at most one PLC scan (otherwise the TNC restarts the timer with the negative edge for each additional scan). The TNC 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 T303 can be started only through Module 9006.

Example:

Start of timer 1
Cycle time in MP4110.1 = 9 (PLC cycles)



Start timer	Timer running	Machine parameter
T0	T48	MP4110.0
T1	T49	MP4110.1
T2	T50	MP4110.2
T3	T51	MP4110.3
T4	T52	MP4110.4
T5	T53	MP4110.5
T6	T54	MP4110.6
T7	T55	MP4110.7
T8	T56	MP4110.8
T9	T57	MP4110.9
T10	T58	MP4110.10
T11	T59	MP4110.11
T12	T60	MP4110.12
T13	T61	MP4110.13
T14	T62	MP4110.14
T15	T63	MP4110.15

Start timer	Timer running	Machine parameter
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 Timer preset value T0 to T47

Input: 0 to 65 535 [PLC cycle times]

Module 9006: Setting and starting PLC timers

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.
- One of the timers from T48 to T96 is set immediately after the module is called. T0 to T47 are not set.
- Unit of measure: milliseconds [ms]
- The TNC rounds the actual cycle time to whole-number PLC cycle times.
- Cancel cycle time: Reset timers T48 to T303.

Call:

PS	B/W/D/K	<Timer no.>
		Input value: 0 to 303
PS	B/W/D/K	<Cycle time>
		Input value: 0 to 1 000 000 [ms]
CM	9006	

Error recognition:

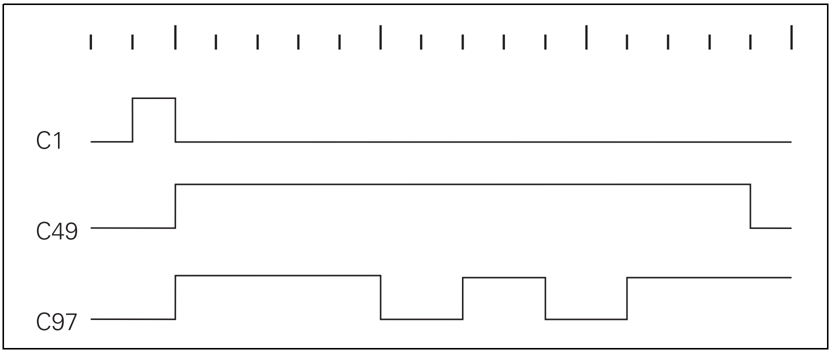
Marker	Value	Meaning
M4203	0	Timer started
	1	Error. See W1022.
W1022	1	Illegal timer number or excessive cycle time

7.2.4 Counters

The PLC has 32 counters, which you control through special markers with the symbol C. After you have set a marker from the C0 to C31 range, the TNC loads the counter with the value that is saved in machine parameter MP4120.x. The marker range C48 to C79 indicates whether the counter is finished. With markers C96 to C127 you can start and stop the counter.

Example:

Logic diagram for counter C1
Preset value in MP4120.1 = 10 (PLC cycles)



Set counter	Counter is running	Counter is starting	Machine parameter
C0	C48	C96	MP4120.0
C1	C49	C97	MP4120.1
C2	C50	C98	MP4120.2
C3	C51	C99	MP4120.3
C4	C52	C100	MP4120.4
C5	C53	C101	MP4120.5
C6	C54	C102	MP4120.6
C7	C55	C103	MP4120.7
C8	C56	C104	MP4120.8
C9	C57	C105	MP4120.9
C10	C58	C106	MP4120.10
C11	C59	C107	MP4120.11
C12	C60	C108	MP4120.12
C13	C61	C109	MP4120.13
C14	C62	C110	MP4120.14
C15	C63	C111	MP4120.15
C16	C64	C112	MP4120.16
C17	C65	C113	MP4120.17
C18	C66	C114	MP4120.18
C19	C67	C115	MP4120.19

Set counter	Counter is running	Counter is starting	Machine parameter
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

MP4120.0-31 Counter preset for C0 to C31
Input: 0 to 65 535 [PLC cycles]

7.2.5 Fast PLC Inputs

With MP4130 you can define PLC inputs that are not interrogated within the PLC cycle (21 ms), but rather in the control loop cycle (3 ms). 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 TNC 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 input**
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	**Condition for activating 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 LE can be defined as fast PLC inputs, and not the inputs on a PL 4xxB.

7.3 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.

7.3.1 Data Transfer NC Program → PLC

With the Q-parameter function **FN19** you can transfer two values from an NC program to the PLC. The TNC stores the transferred values as integer values of the form 1/10 000 in the double words D280 and D284. M4570 determines the unit of measure for the two values transferred. During transfer, the marker M4075 is set by the NC. Transfer is acknowledged by the PLC by setting the marker M4095.

		Set	Reset
M4075	**Transfer active with FN19**	NC	NC
M4095	**Acknowledgement of transfer with FN19**	PLC	PLC
M4570	**Unit of measure for transfer with FN19** 0: mm 1: inch	NC	NC
D280	**1st integer value from FN19**	NC	NC
D284	**2nd integer value from FN19**	NC	NC

7.3.2 Data Transfer PLC → NC Program (Q Parameter)

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 TNC transfers the values with the next M/S/T 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 Nr. 0-7 for numerical data transfer PLC → NC**	PLC	PLC

7.3.3 Data Transfer NC Program → NC (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 TNC 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 (0 = inactive, –1 = active)
		7	–	Tilt working plane in Manual (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	Negative software limit switches in axes 1 to 9
		3	1 to 9	Positive software limit switches in axes 1 to 9
		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
Measuring touch probe				
	350	30	–	Effective length
		31	–	Effective radius 1
		32	–	Effective radius 2
		33	–	Diameter of calibration ring
		34	1	Center offset (reference axis)
			2	Center offset (minor axis)
		35	1	Compensation factor in axis 1
			2	Compensation factor in axis 2
			3	Compensation factor in axis 3
		36	1	Power ratio in axis 1
			2	Power ratio in axis 2
			3	Power ratio in axis 3
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

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
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
Coordinate transformation				
		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

7.3.4 Data Transfer NC → NC Program (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
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
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)

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
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 (0 = inactive, –1 = active)
		7	–	Tilt working plane in Manual (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	Current datum shift of the axes 1 to 9
		3	1 to 9	Difference between reference point and datum point
		4	1 to 9	Current PLC datum shift of the axes 1 to 9
Traverse range				
	230	2	1 to 9	Negative software limit switches in axes 1 to 9
		3	1 to 9	Positive software limit switches in axes 1 to 9
Nominal position in the REF system				
	240	1	1 to 9	Axis 1 to 9
Current position in the active coordinate system				
	270	1	1 to 9	Axis 1 to 9
M128 active				
	280	1	–	–1 = M128 active, 0 = M128 not active
		2	–	Feed rate programmed with M128

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
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°
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

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
Measuring touch probe				
	350	30	–	Effective length
		31	–	Effective radius 1
		32	–	Effective radius 2
		33	–	Diameter of calibration ring
		34	1	Center offset (reference axis)
			2	Center offset (minor axis)
		35	1	Compensation factor in axis 1
			2	Compensation factor in axis 2
			3	Compensation factor in axis 3
		36	1	Power ratio in axis 1
			2	Power ratio in axis 2
			3	Power ratio in axis 3
Datum from touch probe cycle				
	360	1	1 to 9	Last datum of a manual touch probe cycle or last touch point from cycle 0 for the axes 1 to 9 without probe length compensation, but with probe radius compensation (workpiece coordinate system)
		2	1 to 9	Last datum of a manual touch probe cycle or last touch point from cycle 0 for the axes 1 to 9 without probe length or radius compensation (machine coordinate system)
		3	–	Measurement result of touch probe cycles 0 and 1 without probe radius and 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	Tool-oriented machining 0 = Safety height not programmed 1 = Safety height programmed
		6	1 to 9	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

Group name	Group number ID....	System data number NR....	System data index IDX....	System data item
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
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

7.3.5 Data Transfer of Machine Parameters → PLC

In the PLC there are 122 machine parameters reserved for data transfer to the PLC. The TNC 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, PLC positioning, 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 TNC 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 to M4315	**Value from MP4310.0**	NC	NC
M4316 to M4331	**Value from MP4310.1**	NC	NC
M4332 to M4347	**Value from MP4310.2**	NC	NC
M4348 to M4363	**Value from MP4310.3**	NC	NC
M4364 to M4379	**Value from MP4310.4**	NC	NC
M4380to M4395	**Value from MP4310.5**	NC	NC
M4396 to 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

7.3.6 Interrogate PLC Operands in the NC Program (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.

7.4 Hard-Disk Organization

7.4.1 Partitions

The hard disk of the TNC 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 in the system partition can impair proper function of the TNC!

Size of the partitions

As of NC software 280 476-06, hard disks with more than 3.25 GB are supported. In a hard disk with 3.25 GB the partitions are divided as follows:

Partition	Content	Size
TNC	User data	2 GB
PLC	OEM data	128 MB
SYS	System data	1.125 GB

If the hard disk has more than 3.25 GB, the SYS partition receives the rest, up to a maximum of 2 GB. If there is still space available, it is used for the PLC partition up to a maximum of 2 GB. The rest is used by the TNC partition.

Directory structure HEIDENHAIN recommends creating the following directory structure in the PLC partition:

PLC:\	System files *.SYS, (MP_NAME.MP only for default setting)
AXIS_COR	Compensation value tables *.CMA and *.COM
PLC_PGM	PLC programs *.PLC (main program and modules)
LANGUAGE	Directory for PLC dialogs and error messages (created automatically)
ENGLISH FRENCH GERMAN ITALIAN SPANISH	PLC dialogs and error messages *.A; Help files *.HLP
MP	Machine parameter files *.MP
NC_MACRO	NC macros
OEMCYC	Directory for OEM cycles (created automatically by CycleDesign)
DES ELE NC SK	
PLCSOFTK	Pictures for PLC soft keys

Note

A maximum of 512 entries can be stored in the root directory, otherwise an error message appears.

7.4.2 Description of the System Files (*.SYS)

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 TNC automatically enters this MP file in the OEM.SYS file!

Example entry: MPFILE = PLC:\MP\NC430V02.MP

PLCMAIN (mandatory entry): Path for the active PLC program. If you compile a PLC program, the TNC automatically enters it in the OEM.SYS file!

Example entry: PLCMAIN = PLC:\PLC_PGM\MAIN_430.PLC

PLCPWM: Path for PLC program for commissioning of digital axes

Example entry: PLCPWM = PLC:\IB_PGM\IB430.PLC

PLCERRTAB (mandatory entry for PLC error messages): Path for PLC error message table. If you compile a PLC program, the TNC automatically enters it in the OEM.SYS file!

Example entry: 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.

Example entry: PLCERROR = PLC_ERR.A

PLCDIALOG: Name for text file with PLC dialogs; the path for the text file is permanently defined.

Example entry: PLCDIALOG = DIALOG.A

PLCSOFTVERS (mandatory entry): TNC displays the PLC software version when the MOD key is pressed.

TABCMA: Path for compensation value tables for axis error compensation (See "Nonlinear Axis Error Compensation" on page 6 – 39).

Example entry: TABCMA = PLC:\AXIS_COR\CORRECT.CMA

MODEHELP: Path for help texts and machine commands

Input example: MODEHELP = PLC:\LANGUAGE\ENGLISH\OPTIMIZE.HLP

PLCPASSWORD: Code number for calling the PLC mode (instead of 807667)

Example entry: 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)

Example entry: 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.

Example input: MPLOCKFILE = PLC:\MP\280474.MPL

TTYP: Path and file name for list of the tool types

PLCERRFIX: Path for "corrective action" help text

PLCERRREASON: Path for "cause of error" help text

PLCEVENTS: Path event list (spawn command)

LSV2TIME0: Timeout for block reception (STX to ETX)

LSV2TIME1: Timeout for acknowledgment of ENQ or checksum

LSV2TIME2: Timeout during transmission of DLE 0, DLE 1 or NAK until reception of a valid character

KINEMATIC: Path for the tilting-axis geometry description assignment table

Example input: KINEMATIC = PLC:\KINELIST.TAB

REMOTE.LOCKSOFTKEYVISIBLE: Display **External access ON/OFF** soft key.

REMOTE.PLCPASSWORDNEEDED: Access to the PLC partition using the LSV2 protocol only with the password from **PLCPASSWORD =**

REMOTE.PLCPASSWORDFORCED: Machine backup, full backup and setup only with the password from **PLCPASSWORD =**

AXISNUMBER: Number of axes for which machine parameter indexes should be entered in the machine parameter file.

STRICTREPOS: The function for restoring the position is activated when an NC program block is interrupted during **Program Run, Single Block** or by a STOP and the positions of the NC axes are changed.

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 [0 to 7]>
CM 9270

Error recognition:

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 [0 to 3]>
CM 9271

Error recognition:

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

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 6 – 294)

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 6 – 294)

PLCSOFTK.SYS

Path for the file names of the PLC soft-key pictures. (See “PLC Soft Keys” on page 6 – 273)

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).

NCMACRO.SYS

Names of the NC macros
TC = <Name of the tool change macro>
PALLET = <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 finished 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.

As of NC software 280 476-03:
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 is 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 when an "External emergency stop" message is current.
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 recognition:

Marker	Value	Meaning
M4203	0	NC macro was executed
	1	Error code in W1022
W1022	2	NCMACRO.SYS does not exist, code word does not exist, or invalid string
	7	Macro cannot be executed
	8	External emergency stop is active
	20	Module was not called in a spawn job or submit job
	28	NC program or other macro is already running
	29	The file given under the code word is not an NC program (*.H or *.I)
	36	The file given under the code word does not exist

TNC.SYS

(In the TNC partition)
TMAT = <Path for list of tool materials>
WMAT = <Path for list of workpiece materials>
PCDT = <Path for cutting data tables>

7.5 Program Creation

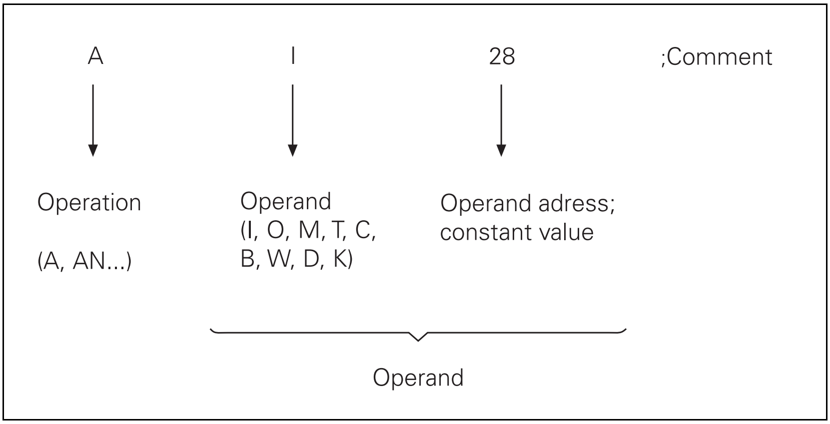
7.5.1 ASCII Editor

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.

7.5.2 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 TNC. 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). When you enter a command, the TNC immediately checks it for the correct syntax and, if necessary, displays an error message. See "Error Messages" on page 10 – 3.

7.5.3 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. (See "PLC Program Example" on page 7 – 50.) 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 bytes that lie on the processing stack of the PLC at the moment. If the processing stack is empty, the TNC 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>

7.5.4 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**

Name of the entry

Group name (NCPATH or OEM)

Identifier for logical name

Examples:

>NCPATH.NCEDIT:
The TNC transfers the complete name and path of the file that is currently selected in the editing mode.

>OEM.PLCMAIN:
The TNC 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. e.g., HUGO=TNC:\HUGO\TEST.H

7.5.5 PLC Compatibility with TNC 415 / TNC 425

With machine parameter MP4020 you can establish compatibility with the TNC 415 by making available the marker range and word range of the TNC 415.

MP4020	**PLC Compatibility with TNC 415 / TNC 425**
Format:	%xxxxx
Input:	Bit 0 = Convert **axis** words (W1024 and following) to markers Bit 1 = Convert **new** markers (4000 and following) to **old** markers (2000 and following) Bit 2 = Convert configuration bits from MP4310 into markers (M2192 to M2239 and M3200 to M3263) Bit 3 = Error markers are available Bit 4 = Remanent markers in the range M1000 to M1999

7.6 PLC Program Example

The following PLC program example was written on a PC with the PLC programming software **PLCdesign. PLCdesign** is supplied together with additional comprehensive PLC program examples in data form.

The PLC program is divided into various PLC modules, where each module performs a specific task. This will help you to quickly recognize the program structure and easily insert your own functions.

A file known as the **documentation file** describes the PLC programming example, which can be output by **PLCdesign** in addition to the individual PLC programs (see the User's Manual for the **PLCdesign** PLC programming software). In the right column of the **documentation file** you will find the source code of the individual modules. This was created by the programmer using symbolic operands and label names. The left column shows the statements lists (STL) as they are needed by the TNC. The compiler automatically generates these statement lists.

This juxtaposition of source code and statements list is very helpful for understanding the program. Abbreviations were defined for the symbolic label numbers and symbolic operands contained in the source code. These abbreviations clearly identify the functions and thereby make the program more understandable.

The following is an example for the definition of a general symbol name, in which individual concepts are separated by an underline "_":

MG_active_PWM_axis

Description of operand function by any text

Separator

Status: Enter if the operand type is not specified.
Input: L = locally effective (only in the module)
G = globally effective (in all modules)

Input for operand type: M, B, W, D, K

Special cases

Interface operands PLC-NC or NC-PLC, inputs and outputs, timers and counters, and positive and negative edge markers are always globally effective and are therefore not indicated as such.

Example

```
NP_M2008_X_InPos
I_release_tool
TS_5_clamp_unclmp
CS_RS_Err_ReStart
```

PLC program example

```

*+--------------------------------------------------+
*|  Main-Program for TNC 430                        |
*+--------------------------------------------------+

#plcpath PLC:\EXAMPLE\
#pragma symsort
#pragma nsc
#pragma dl 1
*#pragma nsw

*+--------------------------------------------------+
*|  Marker range definition                         |
*+--------------------------------------------------+

#define /MN 3200 3999
#define /MR 200   999
#define /BN 2048 4095
#define /BR 4     127

*+--------------------------------------------------+
*|  Gloal file definition                           |
*+--------------------------------------------------+

#define /g  Config.Def
```

```

              *+-------------------------------------------------+
              *| Configuartion file for this PLC program         |
              *+-------------------------------------------------+

              * Number of PL boards
K0            #define       First_PL                          0

              * Input Belegung definieren
K0            #define       IO_MB410                          K+0
K1            #define       IO_SPG                            K+1
K0            #define       IO_Belegung                       K &IO_MB410

              * Monitoring motor temperature axes 1...5
K255          #define       Motor_Temp_1                      255
K255          #define       Motor_Temp_2                      255
K255          #define       Motor_Temp_3                      255
K255          #define       Motor_Temp_4                      255
K255          #define       Motor_Temp_5                      255
K255          #define       Motor_Temp_S                      255


              #ifdef $TNC410M$ or $TNC410MA$ or $TNC410PA$
K0              #define       NC_Type_Digital                   0
              #endif

              #ifdef $TNC426M$ or $TNC426PB$ or $TNC430M$ or $TNC430PA$
                #define       NC_Type_Digital                   0
              #endif

              #ifdef $TNC410$ or $TNC410M$ or $TNC410M$ or $TNC410MA$ or $TNC410CA$ or $TNC410PA$
K4              #define       Max_NC_Axis                       4     * NC-axis without spindle
              #else
                #define       Max_NC_Axis                       5     * NC-axis without spindle
              #endif

```

```
#define /g  GLB_TCMB.Def
```

```
              *+--------------------------------------------------+
              *|  Global makrker Byte Word  DWord                 |
              *+--------------------------------------------------+

              #Type M
M3999            MG_one_marker
M3998            MG_zero_marker
M3997            MG_spindle_on_M03
M3996            MG_spindle_on_M04
M3995            MG_spindle_off_M05
M3994            MG_spi_Pos_M19_R_M0X
M3993            MG_T_I_N_supervision
M3992            MG_Spindle_RPM_Zero
M3991            MG_spindle_on_gear
M3990            MG_gear_change_activ

```

```
                     #Type
M992                 /r MG_closed_loop           M[8]
M992                 /c MG_1_clamp_mode_activ   M &MG_closed_loop +  0
M993                 /c MG_2_clamp_mode_activ   M &MG_closed_loop +  1
M994                 /c MG_3_clamp_mode_activ   M &MG_closed_loop +  2
M995                 /c MG_4_clamp_mode_activ   M &MG_closed_loop +  3
M996                 /c MG_5_clamp_mode_activ   M &MG_closed_loop +  4
M997                 /c MG_S_clamp_mode_activ   M &MG_closed_loop +  5

M3982                   MG_active_PWM_axis       M[8]
M3982                /c MG_active_PWM_axis_1     M &MG_active_PWM_axis + 0
M3983                /c MG_active_PWM_axis_2     M &MG_active_PWM_axis + 1
M3984                /c MG_active_PWM_axis_3     M &MG_active_PWM_axis + 2
M3985                /c MG_active_PWM_axis_4     M &MG_active_PWM_axis + 3
M3986                /c MG_active_PWM_axis_5     M &MG_active_PWM_axis + 4
M3987                /c MG_active_PWM_axis_S     M &MG_active_PWM_axis + 5

                     #Type
B4088                   BG_MPAxis.x_CA_PA         B[6]
B4088                /c BG_MPAxis.0_CA_PA_1       B &BG_MPAxis.x_CA_PA + 0
B4089                /c BG_MPAxis.1_CA_PA_2       B &BG_MPAxis.x_CA_PA + 1
B4090                /c BG_MPAxis.2_CA_PA_3       B &BG_MPAxis.x_CA_PA + 2
B4091                /c BG_MPAxis.3_CA_PA_4       B &BG_MPAxis.x_CA_PA + 3
B4092                /c BG_MPAxis.4_CA_PA_5       B &BG_MPAxis.x_CA_PA + 4
B4093                /c BG_MPSpin.0_CA_PA_S       B &BG_MPAxis.x_CA_PA + 5


                     #Type W
W4086                   WG_MP10_Active_Axis
W4084                   WG_servo_enable_internal_servo
W4082                   WG_Active_PWM_Axis

                     #Type
W4068                   WG_motor_temp        W[6]
W4068                /c WG_motor_temp_1      W &WG_motor_temp +  0
W4070                /c WG_motor_temp_2      W &WG_motor_temp +  2
W4072                /c WG_motor_temp_3      W &WG_motor_temp +  4
W4074                /c WG_motor_temp_4      W &WG_motor_temp +  6
W4076                /c WG_motor_temp_5      W &WG_motor_temp +  8
W4078                /c WG_motor_temp_S      W &WG_motor_temp + 10

                     *+---------------------------------------------------+
                     *|  Error-Marker                                     |
                     *+---------------------------------------------------+

                     #Type
M4800                   PN_error_mod_9167                              M4800
M4801                   PN_error_mod_9002                              M4801
M4802                   PN_error_mod_9005                              M4802
M4803                   PN_error_mod_9161                              M4803
M4804                   PN_error_Submit_Queue_Full                     M4804
M4805                   PN_error_not_used_M_function                   M4805
M4806                   PN_error_9171_Spi_Pos                          M4806
M4807                   PN_error_servo_activ                           M4807
M4808                   PN_error_Temp_powersupply                      M4808
M4809                   PN_error_I2T_caution                           M4809
M4810                   PN_error_modul_9xxx_Supervision                M4810
M4811                   PN_error_utilization_motor                     M4811
M4812                   PN_error_motor_temp                            M4812
M4813                   PN_error_I2T_limitation                        M4813
M4814                   MG_Function_On                                 M4814
K15                     KG_Error_Modul_9200                             K+15
K16                     KG_Error_Modul_9220                             K+16
K17                     KG_Error_Modul_9210                             K+17
K18                     KG_Error_Modul_9202                             K+18
M4815                   PN_Error_gear_switching                        M4815
M4816                   PN_Error_spindle_zero                          M4816

                     *+---------------------------------------------------+
                     *|  Timer Counter definition                         |
                     *+---------------------------------------------------+

                     #type    T
                       * This Timer must be in this sequenze
T0                       TS_1_clamping              0
T48                      TR_1_clamping              &TS_1_clamping              + 48
T1                       TS_2_clamping              1
T49                      TR_2_clamping              &TS_2_clamping              + 48
T2                       TS_3_clamping              2
T50                      TR_3_clamping              &TS_3_clamping              + 48
T3                       TS_4_clamping              3
T51                      TR_4_clamping              &TS_4_clamping              + 48
T4                       TS_5_clamping              4
T52                      TR_5_clamping              &TS_5_clamping              + 48
T7                       TS_M_func_delay            7
T55                      TR_M_func_delay            &TS_M_func_delay            + 48
T8                       TS_1_servo_supervison      8
T56                      TR_1_servo_supervison      &TS_1_servo_supervison      + 48
T9                       TS_2_servo_supervison      9
```

```
T57        TR_2_servo_supervison      &TS_2_servo_supervison      + 48
T10        TS_3_servo_supervison      10
T58        TR_3_servo_supervison      &TS_3_servo_supervison      + 48
T11        TS_4_servo_supervison      11
T59        TR_4_servo_supervison      &TS_4_servo_supervison      + 48
T12        TS_5_servo_supervison      12
T60        TR_5_servo_supervison      &TS_5_servo_supervison      + 48
T13        TS_6_servo_supervison      13
T61        TR_6_servo_supervison      &TS_6_servo_supervison      + 48
T14        TS_7_servo_supervison      14
T62        TR_7_servo_supervison      &TS_7_servo_supervison      + 48
T15        TS_8_servo_supervison      15
T63        TR_8_servo_supervison      &TS_8_servo_supervison      + 48

T20        TS_gear_timeout         20
T68        TR_gear_timout         &TS_gear_timeout        + 48
T21        TS_grear_toggel_all  21
T69        TR_grear_toggel_all  &TS_grear_toggel_all + 48
T22        TS_grear_toggel_right  22
T70        TR_grear_toggel_right  &TS_grear_toggel_right + 48


          #define /g  GLB_NC.Def
          #type

          *+--------------------------------------------------+
          *|  Spindle                                         |
          *+--------------------------------------------------+

M4000        NP_M4000_S_in_position                        M4000
M4001        NP_M4001_S_analog_not_in_ramp                 M4001
M4002        NP_M4002_S_analog_0_V                         M4002

M4005        PN_M4005_S_M03_analog_volt_status             M4005
M4006        PN_M4006_S_M04_analog_volt_status             M4006
M4007        PN_M4007_S_M05_0V_status                      M4007
M4012        PN_M4012_S_close_loop_open                    M4012

          *+--------------------------------------------------+
          *|  Strobe signal from NC to PLC                    |
          *+--------------------------------------------------+

M4070        NP_M4070_strobe_G_code                        M4070
M4071        NP_M4071_strobe_S_code                        M4071
M4072        NP_M4072_strobe_M_function                    M4072

          *+--------------------------------------------------+
          *|  Quit NC strobe signal                           |
          *+--------------------------------------------------+

M4090        PN_M4090_quit_G_code                          M4090
M4091        PN_M4091_quit_S_code                          M4091
M4092        PN_M4092_quit_M_function                      M4092

          *+--------------------------------------------------+
          *|  Strobe signal  from PLC to NC                   |
          *+--------------------------------------------------+

M4130        PN_M4130_Strobe_PLC_pos_spindle               M4130
M4134        PN_M4134_strobe_G_step_rpm                    M4134
M4010        PN_M4010_S_swing_right                        M4010
M4009        PN_M4009_S_swing_left                         M4009
          *+--------------------------------------------------+
          *|  NC modes and status                             |
          *+--------------------------------------------------+

M4150        NP_M4150_manuel_mode                          M4150
M4151        NP_M4151_electronic_handwhell                 M4151
M4155        NP_M4155_reference_mode                       M4155
M4156        NP_M4156_Softkey_Manual_Operation             M4156
M4172        NP_M4172_1_PLC_after_power_on                 M4172
M4173        NP_M4172_1_PLC_after_compile                  M4173
M4174        NP_M4172_1_PLC_after_MP_edit                  M4174

          *+--------------------------------------------------+
          *|  Arithim or modul error in PLC                   |
          *+--------------------------------------------------+

M4203        NP_M4203_error_Modul_9xxx                     M4203

          *+--------------------------------------------------+
          *|  Marker influenzend bei machine parameter        |
          *+--------------------------------------------------+

M4300        NP_M4300_PowerFailOn_MP4310.0_Bit_00          M4300

```

```
                       *+--------------------------------------------------+
                       *|  Extended button                                 |
                       *+--------------------------------------------------+

M4560                     PN_M4560_NC_STOP_0_activ                     M4560
M4561                     PN_M4561_rapide                              M4561
M4562                     PN_M4562_axis_button_latch                   M4562
M4563                     PN_M4563_feed_enable                         M4563
M4564                     PN_M4564_NC_start                            M4564

M4572                     NP_M4572_enable_jog_mode_Posit               M4572

                       *+--------------------------------------------------+
                       *|  List of words (Word bitcoded 54ZYX)             |
                       *+--------------------------------------------------+
                       *+--------------------------------------------------+
                       *|  Axis                                            |
                       *+--------------------------------------------------+

W1026                     NP_W1026_axis_in_position                    W1026
W1032                     NP_W1032_reference_necessary                 W1032
W1038                     PN_W1038_closed_loop_open_active             W1038
W1040                     PN_W1040_closed_loop_open                    W1040
W1042                     PN_W1042_supervision_inactiv                 W1042
W1044                     PN_W1044_actul_nominal_transfer              W1044
W1046                     PN_W1046_manuel_dircetion_plus               W1046
W1048                     PN_W1048_manuel_direction_minus              W1048
W1050                     PN_W1050_jog_mode_Posit_plus                 W1050
W1052                     PN_W1052_jog_mode_Posit_minus                W1052
W1054                     PN_W1054_reference_endswitch                 W1054
                       *+--------------------------------------------------+
                       *|  Data for strobe signal                          |
                       *+--------------------------------------------------+
D756                      PN_D756_S_nominal_rpm_PLC                     D756
W260                      NP_W260_M_code                                W260
W256                      NP_W256_G_code_spindle                        W256

W302                      NP_W302_Number_PLC_Soft_Key                   W302
W274                      NP_W274_Button_Key_Code                       W274


                       #if  IO_Belegung = &IO_MB410
                         #define /g  GLB_IOMB.Def
                       *
                       *+--------------------------------------------------+
                       *|  PLC-input declaration                           |
                       *+--------------------------------------------------+
                       *
                       #type
I0                        I_Ref_Endswitch_1_axis                        I0
I1                        I_Ref_Endswitch_2_axis                        I1
I2                        I_Ref_Endswitch_3_axis                        I2
I3                        I_not_emergency_stop                          I3
I4                        I_Ref_Endswitch_4_axis                        I4
I6                        I_Ref_Endswitch_5_axis                        I6
I3                     /c I_feed_enable                                 I3
I133                      I_1_axis_Plus                                 I133
I128                      I_1_axis_Minus                                I128
I134                      I_2_axis_Plus                                 I134
I129                      I_2_axis_Minus                                I129
I135                      I_3_axis_Plus                                 I135
I130                      I_3_axis_Minus                                I130
I136                      I_4_axis_Plus                                 I136
I131                      I_4_axis_Minus                                I131
M3981                     I_5_axis_Plus                                 M
M3980                     I_5_axis_Minus                                M
I146                      I_NC_Start                                    I146
I147                      I_NC_Stop                                     I147
I148                      I_rapid_button                                I148

I3                     /c I_servo_ready_1                               I3
I3                     /c I_servo_ready_2                               I3

M991                   /r I_gear_range_1                                M
M990                   /r I_gear_range_2                                M
                       *
```

```
                          *+--------------------------------------------------+
                          *| Define PLC outputs                               |
                          *+--------------------------------------------------+
                          *
                          #type
   O0                        O_1_axis_enable                              O0
   O1                        O_2_axis_enable                              O1
   O2                        O_3_axis_enable                              O2
   O3                        O_4_axis_enable                              O3
   O4                        O_5_axis_enable                              O4

   O7                        O_1_clamping                                 O7
   O8                        O_2_clamping                                 O8
   O9                        O_3_clamping                                 O9
   O10                       O_4_clamping                                 O10
   O11                       O_5_clamping                                 O11

   O15                       O_Spindle_servo_enable                       O15

   M991                   /c O_Gear_Range_1                     M &I_gear_range_1
   M990                   /c O_Gear_Range_2                     M &I_gear_range_2


                          #else
                            #define /g  GLB_IO.Def
                          #endif


                          *+---------------------------------------------------+
                          *|  list of include files                            |
                          *+---------------------------------------------------+

                          #ifdef $280470$
                            USES Initi470.Src
                          #else
                            USES Initi472.Src
                          #endif
                          EXTERN initialization

                          USES M_Funct.Src
                          EXTERN M_Function

                          USES Ref_Endl.Src
                          EXTERN reference_endswitch

                          USES DircBut.Src
                          EXTERN Manuel_button_funcktion

                          USES Axis.Src
                          EXTERN NC_Axis

                          USES Spindle.Src
                          EXTERN spindle_function

                          USES Gear.Src
                          EXTERN Gear_Changing

                          USES HelpDiag.Src
                          EXTERN Axis_Supervision

                          Uses Softkeys.Src
                          Extern PLC_Soft_keys

                          *+---------------------------------------------------+
                          *|  PLC-program                                      |
                          *+---------------------------------------------------+

                          #ifdef First_PL
  0 PS K0                   PS     K+0
  1 CM 9002                 CM     9002
  2 L M4203                 L      NP_M4203_error_Modul_9xxx
  3 S M4801                 S      PN_error_mod_9002
                          #endif

  5 L M4172                 L      NP_M4172_1_PLC_after_power_on
  6 O M4173                 O      NP_M4172_1_PLC_after_compile
  7 O M4174                 O      NP_M4172_1_PLC_after_MP_edit
  8 CMT INITIALIZATION      CMT    initialization

 10 CM M_FUNCTION           CM     M_Function
```

```
  12 L W1032              88   L      NP_W1032_reference_necessary
  13 <> K0                89   <>     K+0
  14 O M4155              90   O      NP_M4155_reference_mode
  15 CMT REFERENCE_ENDSWI 91   CMT    reference_endswitch
                          92
  17 CM MANUEL_BUTTON_FU  93   CM     Manuel_button_funcktion
  18 CM NC_AXIS           94   CM     NC_Axis
  19 CM SPINDLE_FUNCTION  95   CM     spindle_function
  20 CM GEAR_CHANGING     96   CM     Gear_Changing
  21 CM AXIS_SUPERVISION  97   CM     Axis_Supervision
  22 CM PLC_SOFT_KEYS     98   CM     PLC_Soft_keys
                          99
                         100 #ifdef First_PL
  24 PS K0               101   PS     K+0
  25 CM 9005             102   CM     9005
  26 L M4203             103   L      NP_M4203_error_Modul_9xxx
  27 S M4802             104   S      PN_error_mod_9005
                         105 #endif
  28 EM                  106 EM
________________________________ Local Labels ________________________________

9002                                                    : 9002
     76
9005                                                    : 9005
     102



                           1
                           2 *+---------------------------------------------------+
                           3 *|  Initialize PLC programm                           |
                           4 *+---------------------------------------------------+
                           5
                           6 GLOBAL initialization
                           7
     B4067                 8 #define /s    BL_MPs_read_identify                    B
     B4066                 9 #define /s    BL_Case                                 B
     W4080                10 #define /s    WL_Index_Reg                            W
                          11
     K0                   12 #define       KL_Off_Power_Fail                       K+0
     K1                   13 #define       KL_On_Power_Fail                        K+1
                          14
  29 LBL INITIALIZATION   15 LBL initialization
  30 LN M3999             16   LN     MG_one_marker
  31 S M3999              17   S      MG_one_marker
                          18
  33 L M3998              19   L      MG_zero_marker
  34 R M3998              20   R      MG_zero_marker
                          21
  36 L M3999              22   L      MG_one_marker
  37 S M4572              23   S      NP_M4572_enable_jog_mode_Posit
  38 R M992               24   R      MG_1_clamp_mode_activ
  39 R M993               25   R      MG_2_clamp_mode_activ
  40 R M994               26   R      MG_3_clamp_mode_activ
  41 S M995               27   S      MG_4_clamp_mode_activ
  42 S M996               28   S      MG_5_clamp_mode_activ
                          29
  44 L M4300              30   L      NP_M4300_PowerFailOn_MP4310.0_Bit_00
  45 IFT                  31   IFT
  46 PS K1                32     PS     KL_On_Power_Fail
  47 ELSE                 33   ELSE
  48 PS K0                34     PS     KL_Off_Power_Fail
  49 ENDI                 35   ENDI
  50 CM 9167              36   CM     9167
  51 PLW                  37   PLW
  52 <> K0                38   <>     K+0
  53 S M4800              39   S      PN_error_mod_9167
                          40
  55 L K255               41   L      Motor_Temp_1
  56 = W4068              42   =      WG_motor_temp_1
  57 L K255               43   L      Motor_Temp_2
  58 = W4070              44   =      WG_motor_temp_2
  59 L K255               45   L      Motor_Temp_3
  60 = W4072              46   =      WG_motor_temp_3
  61 L K255               47   L      Motor_Temp_4
  62 = W4074              48   =      WG_motor_temp_4
  63 L K255               49   L      Motor_Temp_5
  64 = W4076              50   =      WG_motor_temp_5
  65 L K255               51   L      Motor_Temp_S
  66 = W4078              52   =      WG_motor_temp_S
                          53
  68 RPLY B4067           54   RPLY   BL_MPs_read_identify
  69 <> K0                55   <>     K+0
  70 EMT                  56   EMT
  71 SUBM MPS_READ_SUBMIT 57   SUBM   MPs_read_Submit
  72 = B4067              58   =      BL_MPs_read_identify
  73 == K0                59   ==     K+0
  74 S M4804              60   S      PN_error_Submit_Queue_Full
  75 EM                   61 EM
                          62
```

```
 77 LBL MPS_READ_SUBMIT    63 LBL MPs_read_Submit
 78 L K0                   64   L      K+0
 79 = W4080                65   =      WL_Index_Reg
 80 =X                     66   =X
 81 REPEAT                 67   REPEAT
 82 PS KF MP_READ_TABLE[   68     PS     KF MP_Read_Table[X]
 83 INCX                   69     INCX
 84 PS KF MP_READ_TABLE[   70     PS     KF MP_Read_Table[X]
 85 CM 9032                71     CM     9032
 86 INCX                   72     INCX
 87 L KF MP_READ_TABLE[X   73     L      KF MP_Read_Table[X]
 88 = B4066                74     =      BL_Case
 89 INCX                   75     INCX
 90 L KF MP_READ_TABLE[X   76     L      KF MP_Read_Table[X]
 91 =X                     77     =X
 92 CASE B4066             78     CASE   BL_Case
 93 CM PL_BYTE_INDEX       79       CM     PL_Byte_Index
 94 CM PL_WORD_INDEX       80       CM     PL_Word_Index
 95 CM PL_DWORD_INDEX      81       CM     PL_DWord_Index
 96 ENDC                   82     ENDC
 97 L W4080                83     L      WL_Index_Reg
 98 + K4                   84     +      K+4
 99 = W4080                85     =      WL_Index_Reg
100 =X                     86     =X
101 L KF MP_READ_TABLE[X   87     L      KF MP_Read_Table[X]
102 < K0                   88     <      K+0
103 UNTILT                 89   UNTILT
                           90
105 L K0                   91   L      K+0
106 =X                     92   =X
107 REPEAT                 93   REPEAT
108 PS KF AXISNUMBER[X]    94     PS     KF AxisNumber[X]
109 PS K2 ; AXIS UNDER C   95     PS     K+2                        ; Axis under control (0=no, 1=yes)
110 CM 9038                96     CM     9038
111 PS KF AXISNUMBER[X]    97     PS     KF AxisNumber[X]
112 PS K8 ; AXIS DIGITAL   98     PS     K+8                        ; Axis digital control
113 CM 9038                99     CM     9038
                          100
115 PLW                   101     PLW
116 A[                    102     A[
117 PLW                   103       PLW
118 ]                     104     ]
119 <> K0                 105     <>     K+0
120 S M3982[X]            106     S      MG_active_PWM_axis[X]
121 INCX                  107     INCX
122 LX                    108     LX
123 > K5                  109     >      K+5
124 UNTILT                110   UNTILT
                          111
126 ;-----Spindle Bit fr   112   ;-----Spindle Bit from Bit 5 into Bit 15 copieren
127 LB M3982              113   LB     MG_active_PWM_axis
128 = W4082               114   =      WG_Active_PWM_Axis
129 L M3987               115   L      MG_active_PWM_axis_S
130 IFT                   116   IFT
131 L W4082               117     L      WG_Active_PWM_Axis
132 A K$1F                118     A      K$001F
133 O K$8000              119     O      K$8000
134 = W4082               120     =      WG_Active_PWM_Axis
135 ENDI                  121   ENDI
136 EM                    122 EM
                          123
138 LBL PL_BYTE_INDEX     124 LBL PL_Byte_Index
139 PL B0[X]              125   PL     B0[X]
140 EM                    126 EM
                          127
142 LBL PL_WORD_INDEX     128 LBL PL_Word_Index
143 LX                    129   LX
144 / K2 ;TYPE CASTING F  130   /      K+2                        ;Type casting from Word to Index:=Byte adress
145 =X                    131   =X
146 PL W0[X]              132   PL     W0[X]
147 EM                    133 EM
                          134
149 LBL PL_DWORD_INDEX    135 LBL PL_DWord_Index
150 LX                    136   LX
151 / K4 ;TYPE CASTING F  137   /      K+4                        ;Type casting from DWord Index:=Byte adress
152 =X                    138   =X
153 PL D0[X]              139   PL     D0[X]
154 EM                    140 EM
                          141
    K0                    142 #define       KL_Byte_Type                          K$00
    K1                    143 #define       KL_Word_Type                          K$01
    K2                    144 #define       KL_DWord_Type                         K$02
                          145
    K0                    146 #define       KL_Index_0                            K$00
    K1                    147 #define       KL_Index_1                            K$01
    K2                    148 #define       KL_Index_2                            K$02
    K3                    149 #define       KL_Index_3                            K$03
    K4                    150 #define       KL_Index_4                            K$04
    K5                    151 #define       KL_Index_5                            K$05
                          152
```

```
 158 KFIELD MP_READ_TABLE  153 KFIELD MP_Read_Table
 159  K10                  154   K+10
 160  K$0                  155   KL_Index_0
 161  K$1                  156   KL_Word_Type
 162  K4086                157   K&WG_MP10_Active_Axis
                           158
 164  K-1                  159   K-1
 165 ENDK                  160 ENDK
                           161
 167 KFIELD AXISNUMBER     162 KFIELD AxisNumber
 168  K0 ;AXIS 1           163   K+0                                  ;Axis 1
 169  K1 ;AXIS 2           164   K+1                                  ;Axis 2
 170  K2 ;ACIS 3           165   K+2                                  ;Acis 3
 171  K3 ;AXIS 4           166   K+3                                  ;Axis 4
 172  K4 ;AXIS 5           167   K+4                                  ;Axis 5
 173  K15 ;AXIS S          168   K+15                                 ;Axis S
 174 ENDK                  169 ENDK

________________________________ Local Symbols _______________________________

KL_BYTE_TYPE                                            :  142     K0
KL_DWORD_TYPE                                           :  144     K2
KL_INDEX_0                                              :  146     K0
       KF:160
KL_INDEX_1                                              :  147     K1
KL_INDEX_2                                              :  148     K2
KL_INDEX_3                                              :  149     K3
KL_INDEX_4                                              :  150     K4
KL_INDEX_5                                              :  151     K5
KL_OFF_POWER_FAIL                                       :   12     K0
       PS:48
KL_ON_POWER_FAIL                                        :   13     K1
       PS:46
KL_WORD_TYPE                                            :  143     K1
       KF:161
________________________________Static Symbols _______________________________

BL_CASE                                                 :    9     B4066
        =:88  CASE:92
BL_MPS_READ_IDENTIFY                                    :    8     B4067
     RPLY:68     =:72
WL_INDEX_REG                                            :   10     W4080
        =:79     L:97     =:99
________________________________ Local Labels ________________________________

9032                                                    : 9032
     71
9038                                                    : 9038
     96   99
9167                                                    : 9167
     36
AXISNUMBER                                              :  162
     94   97
MPS_READ_SUBMIT                                         :   63
     57
MP_READ_TABLE                                           :  153
     68   70   73   76   87
PL_BYTE_INDEX                                           :  124
     79
PL_DWORD_INDEX                                          :  135
     81
PL_WORD_INDEX                                           :  128
     80
```

PLC program example

```
                             1
                             2 *+--------------------------------------------------+
                             3 *|  M-function                                      |
                             4 *+--------------------------------------------------+
                             5
                             6 GLOBAL M_Function
                             7
 175 LBL M_FUNCTION          8 LBL M_Function
 176 L W260                  9   L      NP_W260_M_code
 177 < K30                  10   <      K+30
 178 IFT                    11   IFT
 179 L W260                 12     L      NP_W260_M_code
 180 =X                     13     =X
 181 L KF M_FUNK_TAB[X]     14     L      KF M_Funk_Tab[X]
 182 =X                     15     =X
 183 L M4072                16     L      NP_M4072_strobe_M_function
 184 = M0[X]                17     =      M0[X]
 185 ENDI                   18   ENDI
 186 EM                     19 EM
                            20
 188 KFIELD M_FUNK_TAB      21 KFIELD M_Funk_Tab
 189  K3995 ; 0             22   K &MG_spindle_off_M05                ; 0
 190  K4805 ; 1             23   K &PN_error_not_used_M_function      ; 1
 191  K3995 ; 2             24   K &MG_spindle_off_M05                ; 2
 192  K3997 ; 3             25   K &MG_spindle_on_M03                 ; 3
 193  K3996 ; 4             26   K &MG_spindle_on_M04                 ; 4
 194  K3995 ; 5             27   K &MG_spindle_off_M05                ; 5
 195  K4805 ; 6             28   K &PN_error_not_used_M_function      ; 6
 196  K4805 ; 7             29   K &PN_error_not_used_M_function      ; 7
 197  K4805 ; 8             30   K &PN_error_not_used_M_function      ; 8
 198  K4805 ; 9             31   K &PN_error_not_used_M_function      ; 9
 199  K4805 ;10             32   K &PN_error_not_used_M_function      ;10
 200  K4805 ; 1             33   K &PN_error_not_used_M_function      ; 1
 201  K4805 ; 2             34   K &PN_error_not_used_M_function      ; 2
 202  K3997 ; 3             35   K &MG_spindle_on_M03                 ; 3
 203  K3996 ; 4             36   K &MG_spindle_on_M04                 ; 4
 204  K4805 ; 5             37   K &PN_error_not_used_M_function      ; 5
 205  K4805 ; 6             38   K &PN_error_not_used_M_function      ; 6
 206  K4805 ; 7             39   K &PN_error_not_used_M_function      ; 7
 207  K4805 ; 8             40   K &PN_error_not_used_M_function      ; 8
 208  K3994 ; 9             41   K &MG_spi_Pos_M19_R_M0X              ; 9
 209  K4805 ;20             42   K &PN_error_not_used_M_function      ;20
 210  K4805 ; 1             43   K &PN_error_not_used_M_function      ; 1
 211  K4805 ; 2             44   K &PN_error_not_used_M_function      ; 2
 212  K4805 ; 3             45   K &PN_error_not_used_M_function      ; 3
 213  K4805 ; 4             46   K &PN_error_not_used_M_function      ; 4
 214  K4805 ; 5             47   K &PN_error_not_used_M_function      ; 5
 215  K4805 ; 6             48   K &PN_error_not_used_M_function      ; 6
 216  K4805 ; 7             49   K &PN_error_not_used_M_function      ; 7
 217  K4805 ; 8             50   K &PN_error_not_used_M_function      ; 8
 218  K4805 ; 9             51   K &PN_error_not_used_M_function      ; 9
 219  K3995 ;30             52   K &MG_spindle_off_M05                ;30
 220 ENDK                   53 ENDK
________________________________ Local Labels ________________________________

M_FUNK_TAB                                              :   21
     14



                             1
                             2 *+--------------------------------------------------+
                             3 *|  Reference endswitch                             |
                             4 *+--------------------------------------------------+
                             5
                             6 GLOBAL reference_endswitch
                             7
     M3200                   8 #define Inputs          M[16]
     M3200                   9 #define /c Input_Bit0      M &Inputs + 0
     M3201                  10 #define /c Input_Bit1      M &Inputs + 1
     M3202                  11 #define /c Input_Bit2      M &Inputs + 2
     M3203                  12 #define /c Input_Bit3      M &Inputs + 3
     M3204                  13 #define /c Input_Bit4      M &Inputs + 4
                            14
 221 LBL REFERENCE_ENDSWI   15 LBL reference_endswitch
 222 L I0                   16   L      I_Ref_Endswitch_1_axis
 223 = M3200                17   =      Input_Bit0
 224 L I1                   18   L      I_Ref_Endswitch_2_axis
 225 = M3201                19   =      Input_Bit1
 226 L I2                   20   L      I_Ref_Endswitch_3_axis
 227 = M3202                21   =      Input_Bit2
 228 L I4                   22   L      I_Ref_Endswitch_4_axis
 229 = M3203                23   =      Input_Bit3
 230 L I6                   24   L      I_Ref_Endswitch_5_axis
 231 = M3204                25   =      Input_Bit4
                            26
 233 LB M3200               27   LB     Inputs
 234 = W1054                28   =      PN_W1054_reference_endswitch
 235 EM                     29 EM
                            30
```

PLC program example

```
________________________________ Local Symbols ________________________________

INPUTS                                                  :    8     M3200
       LB:233
INPUT_BIT0                                              :    9     M3200
        =:223
INPUT_BIT1                                              :   10     M3201
        =:225
INPUT_BIT2                                              :   11     M3202
        =:227
INPUT_BIT3                                              :   12     M3203
        =:229
INPUT_BIT4                                              :   13     M3204
        =:231



                              *+----------------------------------------------------+
                              *|  Direction button                                  |
                              *|  jog mode                                          |
                              *|  NC-Start                                          |
                              *|  NC-Stop                                           |
                              *|  rapid-button                                      |
                              *+----------------------------------------------------+

                              GLOBAL Manuel_button_funcktion

     M3200                    #define ML_XYZ45_Plus              M[8]
     M3200                    #define /c ML_1_Plus                  M &ML_XYZ45_Plus + 0
     M3201                    #define /c ML_2_Plus                  M &ML_XYZ45_Plus + 1
     M3202                    #define /c ML_3_Plus                  M &ML_XYZ45_Plus + 2
     M3203                    #define /c ML_4_Plus                  M &ML_XYZ45_Plus + 3
     M3204                    #define /c ML_5_Plus                  M &ML_XYZ45_Plus + 4

     M3208                    #define ML_XYZ45_Minus             M[8]
     M3208                    #define /c ML_1_Minus                 M &ML_XYZ45_Minus + 0
     M3209                    #define /c ML_2_Minus                 M &ML_XYZ45_Minus + 1
     M3210                    #define /c ML_3_Minus                 M &ML_XYZ45_Minus + 2
     M3211                    #define /c ML_4_Minus                 M &ML_XYZ45_Minus + 3
     M3212                    #define /c ML_5_Minus                 M &ML_XYZ45_Minus + 4

 237 LBL MANUEL_BUTTON_FU     LBL Manuel_button_funcktion
 238 L I147                     L      I_NC_Stop
 239 AN M3993                   AN     MG_T_I_N_supervision
 240 = M4560                    =      PN_M4560_NC_STOP_0_activ

 242 L I148                     L      I_rapid_button
 243 = M4561                    =      PN_M4561_rapide

 245 L I3                       L      I_feed_enable
 246 = M4563                    =      PN_M4563_feed_enable

 248 L I146                     L      I_NC_Start
 249 = M4564                    =      PN_M4564_NC_start
 250 = M4562                    =      PN_M4562_axis_button_latch

 252 L M4150                    L      NP_M4150_manuel_mode
 253 O M4151                    O      NP_M4151_electronic_handwhell
 254 O M4155                    O      NP_M4155_reference_mode
 255 O M4156                    O      NP_M4156_Softkey_Manual_Operation
 256 CMT INPUT_KEYBOARD         CMT    Input_keyboard
 257 CMT JOG_DIRECTION_BU       CMT    Jog_Direction_Button
 258 EM                       EM

 260 LBL INPUT_KEYBOARD       LBL Input_keyboard
 261 L I133                     L      I_1_axis_Plus
 262 = M3200                    =      ML_1_Plus
 263 L I134                     L      I_2_axis_Plus
 264 = M3201                    =      ML_2_Plus
 265 L I135                     L      I_3_axis_Plus
 266 = M3202                    =      ML_3_Plus
 267 L I136                     L      I_4_axis_Plus
 268 = M3203                    =      ML_4_Plus
 269 L M3981                    L      I_5_axis_Plus
 270 = M3204                    =      ML_5_Plus

 272 L I128                     L      I_1_axis_Minus
 273 = M3208                    =      ML_1_Minus
 274 L I129                     L      I_2_axis_Minus
 275 = M3209                    =      ML_2_Minus
 276 L I130                     L      I_3_axis_Minus
 277 = M3210                    =      ML_3_Minus
 278 L I131                     L      I_4_axis_Minus
 279 = M3211                    =      ML_4_Minus
 280 L M3980                    L      I_5_axis_Minus
 281 = M3212                    =      ML_5_Minus
 282 EM                       EM
```

```
 284 LBL JOG_DIRECTION_BU   73 LBL Jog_Direction_Button
 285 L M4572                74   L      NP_M4572_enable_jog_mode_Posit
 286 A M4151                75   A      NP_M4151_electronic_handwhell
 287 IFT                    76   IFT
 288 LB M3200               77     LB     ML_XYZ45_Plus
 289 = W1050                78     =      PN_W1050_jog_mode_Posit_plus
 290 LB M3208               79     LB     ML_XYZ45_Minus
 291 = W1052                80     =      PN_W1052_jog_mode_Posit_minus
 292 ELSE                   81   ELSE
 293 LB M3200               82     LB     ML_XYZ45_Plus
 294 = W1046                83     =      PN_W1046_manuel_dircetion_plus
 295 LB M3208               84     LB     ML_XYZ45_Minus
 296 = W1048                85     =      PN_W1048_manuel_direction_minus
 297 ENDI                   86   ENDI
 298 EM                     87 EM

________________________________ Local Symbols _______________________________

ML_1_MINUS                                           :   20     M3208
        =:273
ML_1_PLUS                                            :   13     M3200
        =:262
ML_2_MINUS                                           :   21     M3209
        =:275
ML_2_PLUS                                            :   14     M3201
        =:264
ML_3_MINUS                                           :   22     M3210
        =:277
ML_3_PLUS                                            :   15     M3202
        =:266
ML_4_MINUS                                           :   23     M3211
        =:279
ML_4_PLUS                                            :   16     M3203
        =:268
ML_5_MINUS                                           :   24     M3212
        =:281
ML_5_PLUS                                            :   17     M3204
        =:270
ML_XYZ45_MINUS                                       :   19     M3208
       LB:290   LB:295
ML_XYZ45_PLUS                                        :   12     M3200
       LB:288   LB:293
________________________________ Local Labels ________________________________

INPUT_KEYBOARD                                       :   49
     45
JOG_DIRECTION_BU                                     :   73
     46


                             1
                             2 * +--------------------------------------------------+
                             3 * | Axis control 5,4,3,2,1,                          |
                             4 * +--------------------------------------------------+
                             5
                             6 GLOBAL NC_Axis
                             7
     W4062                   8 #define /s  WL_current_rpm_control       W
     W4060                   9 #define /s  WL_old_current_rpm_control   W
                            10
     B4052                  11 #define /s  BL_Axis_Step                 B[5]
     W2048                  12 #define     WL_Axis_Mask                 W
                            13
     M3975                  14 #define /s  ML_servo_enable_axis         M[5]
     M3975                  15 #define /c  ML_1_servo_enable_axis       M &ML_servo_enable_axis + 0
     M3976                  16 #define /c  ML_2_servo_enable_axis       M &ML_servo_enable_axis + 1
     M3977                  17 #define /c  ML_3_servo_enable_axis       M &ML_servo_enable_axis + 2
     M3978                  18 #define /c  ML_4_servo_enable_axis       M &ML_servo_enable_axis + 3
     M3979                  19 #define /c  ML_5_servo_enable_axis       M &ML_servo_enable_axis + 4
                            20
     M3970                  21 #define /s  ML_clamping_Achsen           M[5]
     M3970                  22 #define /c  ML_clamping_1_axis           M &ML_clamping_Achsen + 0
     M3971                  23 #define /c  ML_clamping_2_axis           M &ML_clamping_Achsen + 1
     M3972                  24 #define /c  ML_clamping_3_axis           M &ML_clamping_Achsen + 2
     M3973                  25 #define /c  ML_clamping_4_axis           M &ML_clamping_Achsen + 3
     M3974                  26 #define /c  ML_clamping_5_axis           M &ML_clamping_Achsen + 4
                            27
```

PLC program example

```
299 LBL NC_AXIS            28 LBL NC_Axis
300 L K1                   29   L      K+1
301 = W2048                30   =      WL_Axis_Mask
302 L K0                   31   L      K+0
303 =X                     32   =X
304 REPEAT                 33   REPEAT
305 LN I3                  34     LN     I_not_emergency_stop
306 ON I3                  35     ON     I_servo_ready_1
307 ON I3                  36     ON     I_servo_ready_2
308 IFT                    37     IFT
309 L K0                   38       L      K+0
310 = B4052[X]             39       =      BL_Axis_Step[X]
311 ENDI                   40     ENDI
312 CASE B4052[X]          41     CASE   BL_Axis_Step[X]
313 CM INITIAL_AXIS        42       CM     Initial_Axis
314 CM WAITING_POS_STAR    43       CM     Waiting_Pos_Start
315 CM ON_CURRENTRPML_C    44       CM     On_currentRPMl_control
316 CM CLAMPING_OPEN       45       CM     Clamping_open
317 CM CLOSE_LOOP_CLOSE    46       CM     close_loop_close
318 CM POSITIONING         47       CM     positioning
319 CM CLOSE_LOOP_OPEN     48       CM     close_loop_open
320 CM CLAMPING_CLOSE      49       CM     clamping_close
321 CM OFF_CURRENTRPML_    50       CM     off_currentRPMl_control
322 CM STEP_CHAIN_END      51       CM     Step_chain_end
323 ENDC                   52     ENDC
324 L W2048                53     L      WL_Axis_Mask
325 << K1                  54     <<     K+1
326 = W2048                55     =      WL_Axis_Mask
327 INCX                   56     INCX
328 LX                     57     LX
329 >= K4                  58     >=     Max_NC_Axis
330 UNTILT                 59   UNTILT
                           60
332 CM WRITE_OUTPUTS       61   CM     Write_Outputs
                           62
                           63 #ifdef NC_Type_Digital
334 L O15                  64   L      O_Spindle_servo_enable
335 IFT                    65   IFT
336 L W4062                66     L      WL_current_rpm_control
337 BS K15                 67     BS     K+15
338 = W4062                68     =      WL_current_rpm_control
339 ELSE                   69   ELSE
340 L W4062                70     L      WL_current_rpm_control
341 BC K15                 71     BC     K+15
342 = W4062                72     =      WL_current_rpm_control
343 ENDI                   73   ENDI
                           74
345 L W4062                75   L      WL_current_rpm_control
346 <> W4060               76   <>     WL_old_current_rpm_control
347 IFT                    77   IFT
348 L W4062                78     L      WL_current_rpm_control
349 = W4060                79     =      WL_old_current_rpm_control
350 A W4082                80     A      WG_Active_PWM_Axis
351 = W4084                81     =      WG_servo_enable_internal_servo
352 PSW                    82     PSW
353 CM 9161                83     CM     9161
354 L M4203                84     L      NP_M4203_error_Modul_9xxx
355 S M4803                85     S      PN_error_mod_9161
356 ENDI                   86   ENDI
                           87 #endif
357 EM                     88 EM
                           89
359 LBL INITIAL_AXIS       90 LBL Initial_Axis
360 L W1038                91   L      PN_W1038_closed_loop_open_active
361 O W2048                92   O      WL_Axis_Mask
362 = W1038                93   =      PN_W1038_closed_loop_open_active
                           94
364 L W1040                95   L      PN_W1040_closed_loop_open
365 O W2048                96   O      WL_Axis_Mask
366 = W1040                97   =      PN_W1040_closed_loop_open
                           98
368 L W1042                99   L      PN_W1042_supervision_inactiv
369 O W2048               100   O      WL_Axis_Mask
370 = W1042               101   =      PN_W1042_supervision_inactiv
                          102
372 L W1044               103   L      PN_W1044_actul_nominal_transfer
373 O W2048               104   O      WL_Axis_Mask
374 = W1044               105   =      PN_W1044_actul_nominal_transfer
                          106
376 L M3999               107   L      MG_one_marker
377 R M3975[X]            108   R      ML_servo_enable_axis[X]
378 R M3970[X]            109   R      ML_clamping_Achsen[X]
                          110
380 L W4062               111   L      WL_current_rpm_control
381 AN W2048              112   AN     WL_Axis_Mask
382 = W4062               113   =      WL_current_rpm_control
                          114
```

PLC program example

```
L K0                    L      K+0
O W1026                 O      NP_W1026_axis_in_position
A W2048                 A      WL_Axis_Mask
<> K0                   <>     K+0
IFT                     IFT
INC B4052[X]              INC    BL_Axis_Step[X]
ENDI                    ENDI
EM                    EM

LBL WAITING_POS_STAR  LBL Waiting_Pos_Start
L K0                    L      K+0
O W1026                 O      NP_W1026_axis_in_position
A W2048                 A      WL_Axis_Mask
== K0                   ==     K+0
IFT                     IFT
INC B4052[X]              INC    BL_Axis_Step[X]
ENDI                    ENDI
EM                    EM

LBL ON_CURRENTRPML_C  LBL On_currentRPMl_control
L W4062                 L      WL_current_rpm_control
O W2048                 O      WL_Axis_Mask
= W4062                 =      WL_current_rpm_control

LN M3975[X]             LN     ML_servo_enable_axis[X]
S M3975[X]              S      ML_servo_enable_axis[X]

INC B4052[X]            INC    BL_Axis_Step[X]
EM                    EM

LBL CLAMPING_OPEN     LBL Clamping_open
LN M3970[X]             LN     ML_clamping_Achsen[X]
S M3970[X]              S      ML_clamping_Achsen[X]
= T0[X]                 =      TS_1_clamping[X]

LN T0[X]                LN     TS_1_clamping[X]
AN T48[X]               AN     TR_1_clamping[X]
IFT                     IFT
INC B4052[X]              INC    BL_Axis_Step[X]
ENDI                    ENDI
EM                    EM

LBL CLOSE_LOOP_CLOSE  LBL close_loop_close
LN M992[X]              LN     MG_1_clamp_mode_activ[X]
IFT                     IFT
L W1038                   L      PN_W1038_closed_loop_open_active
AN W2048                  AN     WL_Axis_Mask
= W1038                   =      PN_W1038_closed_loop_open_active
ENDI                    ENDI

L W1040                 L      PN_W1040_closed_loop_open
AN W2048                AN     WL_Axis_Mask
= W1040                 =      PN_W1040_closed_loop_open

L W1044                 L      PN_W1044_actul_nominal_transfer
AN W2048                AN     WL_Axis_Mask
= W1044                 =      PN_W1044_actul_nominal_transfer

L W1042                 L      PN_W1042_supervision_inactiv
AN W2048                AN     WL_Axis_Mask
= W1042                 =      PN_W1042_supervision_inactiv

INC B4052[X]            INC    BL_Axis_Step[X]
EM                    EM

LBL POSITIONING       LBL positioning
L K0                    L      K+0
O W1026                 O      NP_W1026_axis_in_position
A W2048                 A      WL_Axis_Mask
<> K0                   <>     K+0
A[                      A[
L M992[X]                 L      MG_1_clamp_mode_activ[X]
ON I3                     ON     I_not_emergency_stop
ON I3                     ON     I_servo_ready_1
ON I3                     ON     I_servo_ready_2
O M3993                   O      MG_T_I_N_supervision
]                       ]
IFT                     IFT
INC B4052[X]              INC    BL_Axis_Step[X]
ENDI                    ENDI
EM                    EM

LBL CLAMPING_CLOSE    LBL clamping_close
L M3970[X]              L      ML_clamping_Achsen[X]
R M3970[X]              R      ML_clamping_Achsen[X]
= T0[X]                 =      TS_1_clamping[X]
```

```
 471 LN T0[X]              202   LN      TS_1_clamping[X]
 472 AN T48[X]             203   AN      TR_1_clamping[X]
 473 IFT                   204   IFT
 474 INC B4052[X]          205     INC    BL_Axis_Step[X]
 475 ENDI                  206   ENDI
 476 EM                    207 EM
                           208
 478 LBL CLOSE_LOOP_OPEN   209 LBL close_loop_open
 479 L W1040               210   L       PN_W1040_closed_loop_open
 480 O W2048               211   O       WL_Axis_Mask
 481 = W1040               212   =       PN_W1040_closed_loop_open
                           213
 483 L W1038               214   L       PN_W1038_closed_loop_open_active
 484 O W2048               215   O       WL_Axis_Mask
 485 = W1038               216   =       PN_W1038_closed_loop_open_active
                           217
 487 INC B4052[X]          218   INC     BL_Axis_Step[X]
 488 EM                    219 EM
                           220
 490 LBL OFF_CURRENTRPML_  221 LBL off_currentRPMl_control
 491 L W4062               222   L       WL_current_rpm_control
 492 AN W2048              223   AN      WL_Axis_Mask
 493 = W4062               224   =       WL_current_rpm_control
                           225
 495 L M3975[X]            226   L       ML_servo_enable_axis[X]
 496 R M3975[X]            227   R       ML_servo_enable_axis[X]
                           228
 498 INC B4052[X]          229   INC     BL_Axis_Step[X]
 499 EM                    230 EM
                           231
 501 LBL STEP_CHAIN_END    232 LBL Step_chain_end
 502 L K1                  233   L       K+1
 503 = B4052[X]            234   =       BL_Axis_Step[X]
 504 EM                    235 EM
                           236
 506 LBL WRITE_OUTPUTS     237 LBL Write_Outputs
 507 L M3975               238   L       ML_1_servo_enable_axis
 508 = O0                  239   =       O_1_axis_enable
 509 L M3976               240   L       ML_2_servo_enable_axis
 510 = O1                  241   =       O_2_axis_enable
 511 L M3977               242   L       ML_3_servo_enable_axis
 512 = O2                  243   =       O_3_axis_enable
 513 L M3978               244   L       ML_4_servo_enable_axis
 514 = O3                  245   =       O_4_axis_enable
 515 L M3979               246   L       ML_5_servo_enable_axis
 516 = O4                  247   =       O_5_axis_enable
                           248
 518 L M3970               249   L       ML_clamping_1_axis
 519 = O7                  250   =       O_1_clamping
 520 L M3971               251   L       ML_clamping_2_axis
 521 = O8                  252   =       O_2_clamping
 522 L M3972               253   L       ML_clamping_3_axis
 523 = O9                  254   =       O_3_clamping
 524 L M3973               255   L       ML_clamping_4_axis
 525 = O10                 256   =       O_4_clamping
 526 L M3974               257   L       ML_clamping_5_axis
 527 = O11                 258   =       O_5_clamping
 528 EM                    259 EM

________________________________ Local Symbols _______________________________

ML_1_SERVO_ENABLE_AXIS                                  :   15     M3975
        L:507
ML_2_SERVO_ENABLE_AXIS                                  :   16     M3976
        L:509
ML_3_SERVO_ENABLE_AXIS                                  :   17     M3977
        L:511
ML_4_SERVO_ENABLE_AXIS                                  :   18     M3978
        L:513
ML_5_SERVO_ENABLE_AXIS                                  :   19     M3979
        L:515
ML_CLAMPING_1_AXIS                                      :   22     M3970
        L:518
ML_CLAMPING_2_AXIS                                      :   23     M3971
        L:520
ML_CLAMPING_3_AXIS                                      :   24     M3972
        L:522
ML_CLAMPING_4_AXIS                                      :   25     M3973
        L:524
ML_CLAMPING_5_AXIS                                      :   26     M3974
        L:526
WL_AXIS_MASK                                            :   12     W2048
        =:301    L:324    =:326    O:361    O:365    O:369    O:373
       AN:381    A:386    A:396    O:405   AN:430   AN:435   AN:439
       AN:443    A:452    O:480    O:484   AN:492
```

```
_______________________________Static Symbols ______________________________

BL_AXIS_STEP                                            :   11     B4052
        =:310 CASE:312 INCW:389 INCW:399 INCW:411 INCW:422 INCW:446
     INCW:462 INCW:474 INCW:487 INCW:498    =:503
ML_CLAMPING_ACHSEN                                      :   21     M3970
        R:378   LN:415    S:416    L:467    R:468
ML_SERVO_ENABLE_AXIS                                    :   14     M3975
        R:377   LN:408    S:409    L:495    R:496
WL_CURRENT_RPM_CONTROL                                  :    8     W4062
        L:336    =:338    L:340    =:342    L:345    L:348    L:380
        =:382    L:404    =:406    L:491    =:493
WL_OLD_CURRENT_RPM_CONTR                                :    9     W4060
       <>:346    =:349
_______________________________ Local Labels _______________________________

9161                                                    : 9161
     83
CLAMPING_CLOSE                                          :  197
     49
CLAMPING_OPEN                                           :  145
     45
CLOSE_LOOP_CLOSE                                        :  157
     46
CLOSE_LOOP_OPEN                                         :  209
     48
INITIAL_AXIS                                            :   90
     42
OFF_CURRENTRPML_                                        :  221
     50
ON_CURRENTRPML_C                                        :  134
     44
POSITIONING                                             :  180
     47
STEP_CHAIN_END                                          :  232
     51
WAITING_POS_STAR                                        :  124
     43
WRITE_OUTPUTS                                           :  237
     61


                             1
                             2 *+---------------------------------------------------+
                             3 *|  Spindle function                                  |
                             4 *+---------------------------------------------------+
                             5
                             6 GLOBAL spindle_function
                             7
     M3969                   8 #define /s ML_spi_pos_start           M
     M3968                   9 #define /s ML_servo_activ_poweron     M
                            10
 529 LBL SPINDLE_FUNCTION   11 LBL spindle_function
 530 L M3994                12   L      MG_spi_Pos_M19_R_M0X
 531 AN M3969               13   AN     ML_spi_pos_start
 532 CMT M19_START_SPI_PO   14   CMT    M19_start_spi_pos
 533 S M3969                15   S      ML_spi_pos_start
                            16
 535 LN M3994               17   LN     MG_spi_Pos_M19_R_M0X
 536 A M4000                18   A      NP_M4000_S_in_position
 537 R M3969                19   R      ML_spi_pos_start
                            20
 539 L M3997                21   L      MG_spindle_on_M03
 540 S M4005                22   S      PN_M4005_S_M03_analog_volt_status
 541 R M4006                23   R      PN_M4006_S_M04_analog_volt_status
                            24
 543 L M3996                25   L      MG_spindle_on_M04
 544 R M4005                26   R      PN_M4005_S_M03_analog_volt_status
 545 S M4006                27   S      PN_M4006_S_M04_analog_volt_status
                            28
 547 L M3994                29   L      MG_spi_Pos_M19_R_M0X
 548 O M3995                30   O      MG_spindle_off_M05
 549 O M3993                31   O      MG_T_I_N_supervision
 550 ON I3                  32   ON     I_not_emergency_stop
 551 R M4005                33   R      PN_M4005_S_M03_analog_volt_status
 552 R M4006                34   R      PN_M4006_S_M04_analog_volt_status
                            35
 554 LN M4005               36   LN     PN_M4005_S_M03_analog_volt_status
 555 AN M4006               37   AN     PN_M4006_S_M04_analog_volt_status
 556 = M4007                38   =      PN_M4007_S_M05_0V_status
                            39
 558 L M4012                40   L      PN_M4012_S_close_loop_open
 559 R M4012                41   R      PN_M4012_S_close_loop_open
                            42
 561 L M3995                43   L      MG_spindle_off_M05
 562 O M3993                44   O      MG_T_I_N_supervision
 563 O M4005                45   O      PN_M4005_S_M03_analog_volt_status
 564 O M4006                46   O      PN_M4006_S_M04_analog_volt_status
 565 S M4012                47   S      PN_M4012_S_close_loop_open
                            48
```

```
 567 L M4002              49   L      NP_M4002_S_analog_0_V
 568 S M3968              50   S      ML_servo_activ_poweron
 569 L M4005              51   L      PN_M4005_S_M03_analog_volt_status
 570 O M4006              52   O      PN_M4006_S_M04_analog_volt_status
 571 ON M4002             53   ON     NP_M4002_S_analog_0_V
 572 O M4130              54   O      PN_M4130_Strobe_PLC_pos_spindle
 573 A M3968              55   A      ML_servo_activ_poweron
 574 S O15                56   S      O_Spindle_servo_enable
                          57
 576 L M3995              58   L      MG_spindle_off_M05
 577 O M3993              59   O      MG_T_I_N_supervision
 578 ON I3                60   ON     I_not_emergency_stop
 579 ON I3                61   ON     I_servo_ready_1
 580 ON I3                62   ON     I_servo_ready_2
 581 R O15                63   R      O_Spindle_servo_enable
                          64
 583 LN M4072             65   LN     NP_M4072_strobe_M_function
 584 = T7                 66   =      TS_M_func_delay
                          67
 586 L M4072              68   L      NP_M4072_strobe_M_function
 587 A M4001              69   A      NP_M4001_S_analog_not_in_ramp
 588 AN M4130             70   AN     PN_M4130_Strobe_PLC_pos_spindle
 589 AN T7                71   AN     TS_M_func_delay
 590 AN T55               72   AN     TR_M_func_delay
 591 AN M4805             73   AN     PN_error_not_used_M_function
 592 = M4092              74   =      PN_M4092_quit_M_function
 593 EM                   75 EM
                          76
     K0                   77 #define KL_angle_spindle_pos            K+0
     K100000              78 #define KL_RPM_spindle_pos              K+100000
     K0                   79 #define KL_direction_spindle_pos        K+0
                          80
 596 LBL M19_START_SPI_PO   81 LBL M19_start_spi_pos
 597 PS K0                82   PS     KL_angle_spindle_pos
 598 PS K100000           83   PS     KL_RPM_spindle_pos
 599 PS K0                84   PS     KL_direction_spindle_pos
 600 CM 9171              85   CM     9171
 601 L M4203              86   L      NP_M4203_error_Modul_9xxx
 602 S M4806              87   S      PN_error_9171_Spi_Pos
 603 EM                   88 EM

________________________________ Local Symbols _______________________________

KL_ANGLE_SPINDLE_POS                                   :   77     K0
       PS:597
KL_DIRECTION_SPINDLE_POS                               :   79     K0
       PS:599
KL_RPM_SPINDLE_POS                                     :   78     K100000
       PS:598
________________________________Static Symbols _______________________________

ML_SERVO_ACTIV_POWERON                                 :    9     M3968
        S:568    A:573
ML_SPI_POS_START                                       :    8     M3969
       AN:531    S:533    R:537
________________________________ Local Labels ________________________________

9171                                                   : 9171
     85
M19_START_SPI_PO                                       :   81
     14
```

```
                                *+--------------------------------------------------+
                                *| gear change                                      |
                                *+--------------------------------------------------+

    B4065                       #define /s    BL_step_gear                         B
    B127                        #define /s /r BL_G_code                            B

    D120                        #define /s /r DL_N_programmed                      D

    M3200                       #define       ML_Gear_swicth_done                  M

                                GLOBAL Gear_Changing

LBL GEAR_CHANGING               LBL Gear_Changing
L M4172                           L      NP_M4172_1_PLC_after_power_on
O M4173                           O      NP_M4172_1_PLC_after_compile
S M4134                           S      PN_M4134_strobe_G_step_rpm
IFT                               IFT
L B127                              L      BL_G_code
= W256                              =      NP_W256_G_code_spindle
L D120                              L      DL_N_programmed
= D756                              =      PN_D756_S_nominal_rpm_PLC
ENDI                              ENDI

LN M4070                          LN     NP_M4070_strobe_G_code
R M4090                           R      PN_M4090_quit_G_code

L B4065                           L      BL_step_gear
== K0                             ==     K+0
= T20                             =      TS_gear_timeout

CASE B4065                        CASE   BL_step_gear
CM ACTIVATION ;00                   CM     Activation                 ;00
CM SPINDLE_ZERO ;01                 CM     spindle_zero               ;01
CM GEAR_RANGE_SWITC                 CM     gear_range_switch          ;02
CM QUIT ;03                         CM     quit                       ;03
CM END ;04                          CM     end                        ;04
ENDC                              ENDC

PLL                               PLL
IFT                               IFT
INC B4065                           INC    BL_step_gear
ENDI                              ENDI

LN M991                           LN     I_gear_range_1
XO M990                           XO     I_gear_range_2
AN M4070                          AN     NP_M4070_strobe_G_code
AN M4134                          AN     PN_M4134_strobe_G_step_rpm
AN M3990                          AN     MG_gear_change_activ
O[                                O[
AN T20                              AN     TS_gear_timeout
AN T68                              AN     TR_gear_timout
]                                 ]
= M4815                           =      PN_Error_gear_switching

L M4815                           L      PN_Error_gear_switching
AN M4070                          AN     NP_M4070_strobe_G_code
AN M3990                          AN     MG_gear_change_activ
ON I3                             ON     I_not_emergency_stop
CMT RESET                         CMT    reset
EM                              EM

LBL ACTIVATION ;00              LBL Activation                        ;00
L M4134                           L      PN_M4134_strobe_G_step_rpm
O M4070                           O      NP_M4070_strobe_G_code
PSL                               PSL
S M3990                           S      MG_gear_change_activ
EM                              EM

LBL SPINDLE_ZERO ;01            LBL spindle_zero                      ;01
LN T68                            LN     TR_gear_timout
S M4816                           S      PN_Error_spindle_zero

PS M4002                          PS     NP_M4002_S_analog_0_V
EM                              EM

LBL GEAR_RANGE_SWITC            LBL gear_range_switch                 ;02
LN T69                            LN     TR_grear_toggel_all
= T21                             =      TS_grear_toggel_all
= T22                             =      TS_grear_toggel_right

L T70                             L      TR_grear_toggel_right
= M4010                           =      PN_M4010_S_swing_right
=N M4009                          =N     PN_M4009_S_swing_left

CASE W256                         CASE   NP_W256_G_code_spindle
CM GEAR_RANGE_1 ;+00                CM     gear_range_1               ;+00
CM GEAR_RANGFE_2 ;+0                CM     gear_rangfe_2              ;+01
ENDC                              ENDC
EM                              EM
```

```
                           91
 682 LBL GEAR_RANGE_1 ;+0   92 LBL gear_range_1                      ;+00
 683 LN M991                93   LN     I_gear_range_1
 684 O M990                 94   O      I_gear_range_2
 685 S M991                 95   S      O_Gear_Range_1
 686 R M990                 96   R      O_Gear_Range_2
 687 =N M3200               97   =N     ML_Gear_swicth_done
                            98
 689 L M3200                99   L      ML_Gear_swicth_done
 690 PSL                   100   PSL
 691 EM                    101 EM
                           102
 693 LBL GEAR_RANGFE_2 ;+  103 LBL gear_rangfe_2                     ;+01
 694 L M991                104   L      I_gear_range_1
 695 ON M990               105   ON     I_gear_range_2
 696 R M991                106   R      O_Gear_Range_1
 697 S M990                107   S      O_Gear_Range_2
 698 =N M3200              108   =N     ML_Gear_swicth_done
                           109
 700 L M3200               110   L      ML_Gear_swicth_done
 701 PSL                   111   PSL
 702 EM                    112 EM
                           113
 704 LBL QUIT ;03          114 LBL quit                              ;03
 705 L M4070               115   L      NP_M4070_strobe_G_code
 706 S M4090               116   S      PN_M4090_quit_G_code
                           117
 708 LN M4070              118   LN     NP_M4070_strobe_G_code
 709 S M3991               119   S      MG_spindle_on_gear
 710 PSL                   120   PSL
 711 EMF                   121   EMF
                           122
 713 L W256                123   L      NP_W256_G_code_spindle
 714 = B127                124   =      BL_G_code
                           125
 716 L D756                126   L      PN_D756_S_nominal_rpm_PLC
 717 = D120                127   =      DL_N_programmed
 718 EM                    128 EM
                           129
 720 LBL END ;04           130 LBL end                               ;04
 721 PS M3998              131   PS     MG_zero_marker
 722 LBL RESET             132 LBL reset
 723 L M3999               133   L      MG_one_marker
 724 R M3990               134   R      MG_gear_change_activ
 725 R M3991               135   R      MG_spindle_on_gear
 726 R M4010               136   R      PN_M4010_S_swing_right
 727 R M4009               137   R      PN_M4009_S_swing_left
                           138
 729 L K0                  139   L      K+0
 730 = B4065               140   =      BL_step_gear
 731 EM                    141 EM

________________________________ Local Symbols ______________________________

ML_GEAR_SWICTH_DONE                                       :   10     M3200
     WHIL:687    L:689 WHIL:698    L:700
________________________________Static Symbols ______________________________

BL_G_CODE                                                 :    6     B127
        L:609    =:714
BL_STEP_GEAR                                              :    5     B4065
        L:618 CASE:622 INCW:632    =:730
DL_N_PROGRAMMED                                           :    8     D120
        L:611    =:717
________________________________ Local Labels ________________________________

ACTIVATION                                                :   63
     33
END                                                       :  130
     37
GEAR_RANGE_1                                              :   92
     87
GEAR_RANGE_SWITC                                          :   77
     35
GEAR_RANGFE_2                                             :  103
     88
QUIT                                                      :  114
     36
RESET                                                     :  132
     60
SPINDLE_ZERO                                              :   70
     34
```

PLC program example

```

                          GLOBAL Axis_Supervision

                          *+--------------------------------------------------+
                          *| Supervison Servo drive TNC 430                   |
                          *+--------------------------------------------------+

LBL AXIS_SUPERVISION      LBL Axis_Supervision
CM 9160                     CM     9160
L M4203                     L      NP_M4203_error_Modul_9xxx
S M4810                     S      PN_error_modul_9xxx_Supervision
PLW                         PLW
<> K0                       <>     K+0
S M4808                     S      PN_error_Temp_powersupply
PLW                         PLW
PSW                         PSW
A K$FFFF                    A      K$0000FFFF
<> K0                       <>     K+0
S M4809                     S      PN_error_I2T_caution
PLW                         PLW
A K$FFFF0000                A      K$FFFF0000
<> K0                       <>     K+0
S M4813                     S      PN_error_I2T_limitation

CM 9162                     CM     9162
L M4203                     L      NP_M4203_error_Modul_9xxx
S M4810                     S      PN_error_modul_9xxx_Supervision
LN W4084                    LN     WG_servo_enable_internal_servo
B= T8                       B=     TS_1_servo_supervison
L W4084                     L      WG_servo_enable_internal_servo
AN[                         AN[
PLW                           PLW
]                           ]
AN[                         AN[
LB T56                        LB     TR_1_servo_supervison
]                           ]
<> K0                       <>     K+0
S M4807                     S      PN_error_servo_activ

L K0                        L      K+0
=X                          =X
REPEAT                      REPEAT
L M3982                       L      MG_active_PWM_axis
IFT                           IFT
PS KF AXIS_BIT_CODE[            PS     KF Axis_Bit_Code[X]
CM 9165                         CM     9165
PLW                             PLW
>= W4068[X]                     >=     WG_motor_temp[X]
S M4812                         S      PN_error_motor_temp

L M4203                         L      NP_M4203_error_Modul_9xxx
S M4810                         S      PN_error_modul_9xxx_Supervision

PS KF AXIS_BIT_CODE[            PS     KF Axis_Bit_Code[X]
CM 9166                         CM     9166
PLW                             PLW
> K100                          >      K+100
S M4811                         S      PN_error_utilization_motor

L M4203                         L      NP_M4203_error_Modul_9xxx
S M4810                         S      PN_error_modul_9xxx_Supervision
ENDI                          ENDI
INCX                          INCX
LX                            LX
> K5                          >      K+5
UNTILT                      UNTILT

L M4808                     L      PN_error_Temp_powersupply
O M4809                     O      PN_error_I2T_caution
O M4812                     O      PN_error_motor_temp
O M4811                     O      PN_error_utilization_motor
O M4807                     O      PN_error_servo_activ
= M3993                     =      MG_T_I_N_supervision
EM                        EM

KFIELD AXIS_BIT_CODE      KFIELD Axis_Bit_Code
 K0                         K+00
 K1                         K+01
 K2                         K+02
 K3                         K+03
 K4                         K+04
 K15                        K+15
ENDK                      ENDK
```

```
________________________________ Local Labels ________________________________

9160                                             : 9160
     9
9162                                             : 9162
     25
9165                                             : 9165
     46
9166                                             : 9166
     55
AXIS_BIT_CODE                                    :   76
     45   54


                               1
                               2 *+---------------------------------------------------+
                               3 *|  PLC Softkeys                                     |
                               4 *+---------------------------------------------------+
                               5
     K1                        6 #define       KL_soft_key_Off                        K+1
     K2                        7 #define       KL_soft_key_On                         K+2
     K3                        8 #define       KL_Mask_1_On                           K+3
     K4                        9 #define       KL_Mask_2_On                           K+4
     K5                       10 #define       KL_Mask_Off                            K+5
     K0                       11 #define       KL_Empty                               K+0
     K83                      12 #define       KL_ASCII_Key                           K $53 *S ASCII-Key
                              13
     M3967                    14 #define /s    ML_NC_soft_key_On                      M
                              15
                              16 #define /i    Mac_lib.Def


                               1 *+---------------------------------------------------+
                               2 *| Macro library                                     |
                               3 *+---------------------------------------------------+
                               4 *| Macros for error messages                         |
                               5 *|                                                   |
                               6 *+---------------------------------------------------+
                               7
                               8
                               9 #define /m    M_Error_display(No)   \
                              10   IFT                               \
                              11     PS     (No)                     \
                              12     CM     9085                     \
                              13   ENDI
                              14
                              15 #define /m    M_Modul_Error_display(No)  \
                              16   L      NP_M4203_error_Modul_9xxx      \
                              17   IFT                                   \
                              18     PS     (No)                         \
                              19     CM     9085                         \
                              20   ENDI
                              21



                              17
                              18 GLOBAL PLC_Soft_keys
                              19
 808 LBL PLC_SOFT_KEYS        20 LBL PLC_Soft_keys
 809 L W302                   21   L      NP_W302_Number_PLC_Soft_Key
 810 >= K0                    22   >=     K+0
 811 IFT                      23   IFT
 812 CASE W302                24     CASE   NP_W302_Number_PLC_Soft_Key
 813 CM EMPTY_SK              25       CM     Empty_SK
 814 CM OFF_SK                26       CM     Off_SK
 815 CM ON_SK                 27       CM     On_SK
 816 CM MASKE1_SK             28       CM     Maske1_SK
 817 CM MASKE2_SK             29       CM     Maske2_SK
 818 CM DELMASKE_SK           30       CM     DelMaske_SK
 819 ENDC                     31     ENDC
 820 L K-1                    32     L      K-1
 821 = W302                   33     =      NP_W302_Number_PLC_Soft_Key
 822 ENDI                     34   ENDI
                              35
 824 L W274                   36   L      NP_W274_Button_Key_Code
 825 == K$53                  37   ==     KL_ASCII_Key
 826 IFT                      38   IFT
 827 L K-1 ;IMPULS            39     L      K-1                         ;Impuls
 828 = W274                   40     =      NP_W274_Button_Key_Code
                              41
 830 LN M3967                 42     LN     ML_NC_soft_key_On
 831 = M3967                  43     =      ML_NC_soft_key_On
 832 CMT SOFT_KEY_DISPLAY     44     CMT    Soft_key_displayOn
 833 CMF SOFT_KEY_DIPSLAY     45     CMF    Soft_key_dipslayOff
 834 ENDI                     46   ENDI
 835 EM                       47 EM
                              48
 837 LBL EMPTY_SK             49 LBL Empty_SK
 838 EM                       50 EM
```

```
                           51
840 LBL OFF_SK             52 LBL Off_SK
841 L M4814                53   L      MG_Function_On
842 R M4814                54   R      MG_Function_On
843 EM                     55 EM
                           56
845 LBL ON_SK              57 LBL On_SK
846 LN M4814               58   LN     MG_Function_On
847 S M4814                59   S      MG_Function_On
848 EM                     60 EM
                           61
850 LBL SOFT_KEY_DISPLAY   62 LBL Soft_key_displayOn
851 PS KF SOFT_KEY_ROW     63   PS     KF Soft_key_row
852 PS K0                  64   PS     K+0
853 PS K1                  65   PS     K+1
854 CM 9200                66   CM     9200
    ------ Macro ----->    67   M_Modul_Error_display(KG_Error_Modul_9200)
                           67
856 L M4203                67 L      NP_M4203_error_Modul_9xxx
857 IFT                    67 IFT
858 PS K15                 67 PS     KG_ERROR_MODUL_9200
859 CM 9085                67 CM     9085
860 ENDI                   67 ENDI
861 EM                     68 EM
                           69
863 LBL SOFT_KEY_DIPSLAY   70 LBL Soft_key_dipslayOff
864 PS K-1                 71   PS     K-1
865 PS K0                  72   PS     K+0
866 PS K1                  73   PS     K+1
867 CM 9200                74   CM     9200
    ------ Macro ----->    75   M_Modul_Error_display(KG_Error_Modul_9200)
                           75
869 L M4203                75 L      NP_M4203_error_Modul_9xxx
870 IFT                    75 IFT
871 PS K15                 75 PS     KG_ERROR_MODUL_9200
872 CM 9085                75 CM     9085
873 ENDI                   75 ENDI
874 EM                     76 EM
                           77
876 KFIELD SOFT_KEY_ROW    78 KFIELD Soft_key_row
877  K0                    79   KL_Empty
878  K1                    80   KL_soft_key_Off
879  K2                    81   KL_soft_key_On
880  K0                    82   KL_Empty
881  K3                    83   KL_Mask_1_On
882  K4                    84   KL_Mask_2_On
883  K5                    85   KL_Mask_Off
884  K0                    86   KL_Empty
885 ENDK                   87 ENDK
                           88
887 LBL MASKE1_SK          89 LBL Maske1_SK
888 PS K1 ; BIG PLC WIND   90   PS     K+1   ; Big PLC Window open
889 CM 9202                91   CM     9202
    ------ Macro ----->    92   M_Modul_Error_display(KG_Error_Modul_9202)
                           92
891 L M4203                92 L      NP_M4203_error_Modul_9xxx
892 IFT                    92 IFT
893 PS K18                 92 PS     KG_ERROR_MODUL_9202
894 CM 9085                92 CM     9085
895 ENDI                   92 ENDI
                           93
897 L S"MASKE1.A"          94   L      S"Maske1.A"
898 = S0                   95   =      S0
                           96
900 PS K0 ;NR. STRING BU   97   PS     K+0                          ;Nr. String buffer / clear>
901 CM 9210                98   CM     9210
902 PLW ;<STATUS/ERROR>    99   PLW                                 ;<Status/Error>
903 >= K1 ; FEHLER        100   >=     K+1                          ; error
    ------ Macro ----->   101   M_Error_display(KG_Error_Modul_9210)
                          101
905 IFT                   101 IFT
906 PS K17                101 PS     KG_ERROR_MODUL_9210
907 CM 9085               101 CM     9085
908 ENDI                  101 ENDI
909 EM                    102 EM
                          103
911 LBL MASKE2_SK         104 LBL Maske2_SK
912 PS K1 ; BIG PLC WIND  105   PS     K+1   ; Big PLC Window open
913 CM 9202               106   CM     9202
```

```
    ------ Macro ----->   107   M_Modul_Error_display(KG_Error_Modul_9202)
                          107
 915 L M4203              107 L      NP_M4203_error_Modul_9xxx
 916 IFT                  107 IFT
 917 PS K18               107 PS     KG_ERROR_MODUL_9202
 918 CM 9085              107 CM     9085
 919 ENDI                 107 ENDI
                          108
 921 L S"MASKE2.A"        109   L      S"Maske2.A"
 922 = S0                 110   =      S0
                          111
 924 PS K0 ;NR. STRING BU  112   PS     K+0                          ;Nr. String buffer / clear>
 925 CM 9210              113   CM     9210
 926 PLW ;<STATUS/ERROR>   114   PLW                                 ;<Status/Error>
 927 >= K1 ; FEHLER       115   >=     K+1                          ; error
    ------ Macro ----->   116   M_Error_display(KG_Error_Modul_9210)
                          116
 929 IFT                  116 IFT
 930 PS K17               116 PS     KG_ERROR_MODUL_9210
 931 CM 9085              116 CM     9085
 932 ENDI                 116 ENDI
 933 EM                   117 EM
                          118
 935 LBL DELMASKE_SK      119 LBL DelMaske_SK
 936 PS K0 ; BIG PLC WIND  120   PS     K+0   ; Big PLC Window clear
 937 CM 9202              121   CM     9202
    ------ Macro ----->   122   M_Modul_Error_display(KG_Error_Modul_9202)
                          122
 939 L M4203              122 L      NP_M4203_error_Modul_9xxx
 940 IFT                  122 IFT
 941 PS K18               122 PS     KG_ERROR_MODUL_9202
 942 CM 9085              122 CM     9085
 943 ENDI                 122 ENDI
                          123
 945 PS K-1 ;NR. STRING B  124   PS     K-1                          ;Nr. String buffer / clear>
 946 CM 9210              125   CM     9210
 947 PLW ;<STATUS/ERROR>   126   PLW                                 ;<Status/Error>
 948 >= K1 ; FEHLER       127   >=     K+1                          ; error
    ------ Macro ----->   128   M_Error_display(KG_Error_Modul_9210)
                          128
 950 IFT                  128 IFT
 951 PS K17               128 PS     KG_ERROR_MODUL_9210
 952 CM 9085              128 CM     9085
 953 ENDI                 128 ENDI
 954 EM                   129 EM

________________________________ Local Symbols _______________________________

KL_ASCII_KEY                                           :   12     K83
       ==:825
KL_EMPTY                                               :   11     K0
       KF:877   KF:880   KF:884
KL_MASK_1_ON                                           :    8     K3
       KF:881
KL_MASK_2_ON                                           :    9     K4
       KF:882
KL_MASK_OFF                                            :   10     K5
       KF:883
KL_SOFT_KEY_OFF                                        :    6     K1
       KF:878
KL_SOFT_KEY_ON                                         :    7     K2
       KF:879
________________________________Static Symbols _______________________________

ML_NC_SOFT_KEY_ON                                      :   14     M3967
       LN:830    =:831
```

```
________________________________ Local Labels ________________________________

9085                                                       : 9085
     67   75   92   101  107  116  122  128
9200                                                       : 9200
     66   74
9202                                                       : 9202
     91   106  121
9210                                                       : 9210
     98   113  125
DELMASKE_SK                                                :  119
     30
EMPTY_SK                                                   :   49
     25
MASKE1_SK                                                  :   89
     28
MASKE2_SK                                                  :  104
     29
OFF_SK                                                     :   52
     26
ON_SK                                                      :   57
     27
SOFT_KEY_DIPSLAY                                           :   70
     45
SOFT_KEY_DISPLAY                                           :   62
     44
SOFT_KEY_ROW                                               :   78
     63
PLCCOMP V4.00                                      Tue Mar 28 08:12:13 2000

________________________________ Global Labels ________________________________

AXIS_SUPERVISION                          HELPDIAG.SRC  :    8
     MAIN_PGM.SRC  97
________________________________ Global Labels ________________________________

GEAR_CHANGING                             GEAR.SRC      :   14
     MAIN_PGM.SRC  96
________________________________ Global Labels ________________________________

INITIALIZATION                            INITI472.SRC  :   15
     MAIN_PGM.SRC  84
________________________________ Global Labels ________________________________

MANUEL_BUTTON_FU                          DIRCBUT.SRC   :   26
     MAIN_PGM.SRC  93
________________________________ Global Labels ________________________________

M_FUNCTION                                M_FUNCT.SRC   :    8
     MAIN_PGM.SRC  86
________________________________ Global Labels ________________________________

NC_AXIS                                   AXIS.SRC      :   28
     MAIN_PGM.SRC  94
________________________________ Global Labels ________________________________

PLC_SOFT_KEYS                             SOFTKEYS.SRC  :   20
     MAIN_PGM.SRC  98
________________________________ Global Labels ________________________________

REFERENCE_ENDSWI                          REF_ENDL.SRC  :   15
     MAIN_PGM.SRC  91
________________________________ Global Labels ________________________________

SPINDLE_FUNCTION                          SPINDLE.SRC   :   11
     MAIN_PGM.SRC  95
__M0                                      M_FUNCT.SRC   :   17     M0
     M_FUNCT.SRC    =:184

I_GEAR_RANGE_2                            GLB_IOMB.DEF  :   32     M990
     GEAR.SRC      XO:636    O:684   ON:695

O_GEAR_RANGE_2                            GLB_IOMB.DEF  :   54     M990
     GEAR.SRC       R:686    S:697

I_GEAR_RANGE_1                            GLB_IOMB.DEF  :   31     M991
     GEAR.SRC      LN:635   LN:683    L:694

O_GEAR_RANGE_1                            GLB_IOMB.DEF  :   53     M991
     GEAR.SRC       S:685    R:696

MG_1_CLAMP_MODE_ACTIV                     GLB_TCMB.DEF  :   19     M992
     INITI472.SRC   R:38
     AXIS.SRC      LN:427    L:455
```

PLC program example

```
MG_CLOSED_LOOP                          GLB_TCMB.DEF  :   18     M992
     - not used -
MG_2_CLAMP_MODE_ACTIV                   GLB_TCMB.DEF  :   20     M993
     INITI472.SRC   R:39

MG_3_CLAMP_MODE_ACTIV                   GLB_TCMB.DEF  :   21     M994
     INITI472.SRC   R:40

MG_4_CLAMP_MODE_ACTIV                   GLB_TCMB.DEF  :   22     M995
     INITI472.SRC   S:41

MG_5_CLAMP_MODE_ACTIV                   GLB_TCMB.DEF  :   23     M996
     INITI472.SRC   S:42

MG_S_CLAMP_MODE_ACTIV                   GLB_TCMB.DEF  :   24     M997
     - not used -
I_5_AXIS_MINUS                          GLB_IOMB.DEF  :   23     M3980
     DIRCBUT.SRC    L:280

I_5_AXIS_PLUS                           GLB_IOMB.DEF  :   22     M3981
     DIRCBUT.SRC    L:269

MG_ACTIVE_PWM_AXIS                      GLB_TCMB.DEF  :   26     M3982
     INITI472.SRC   S:120   LB:127
     HELPDIAG.SRC   L:767

MG_ACTIVE_PWM_AXIS_1                    GLB_TCMB.DEF  :   27     M3982
     - not used -
MG_ACTIVE_PWM_AXIS_2                    GLB_TCMB.DEF  :   28     M3983
     - not used -
MG_ACTIVE_PWM_AXIS_3                    GLB_TCMB.DEF  :   29     M3984
     - not used -
MG_ACTIVE_PWM_AXIS_4                    GLB_TCMB.DEF  :   30     M3985
     - not used -
MG_ACTIVE_PWM_AXIS_5                    GLB_TCMB.DEF  :   31     M3986
     - not used -
MG_ACTIVE_PWM_AXIS_S                    GLB_TCMB.DEF  :   32     M3987
     INITI472.SRC   L:129

MG_GEAR_CHANGE_ACTIV                    GLB_TCMB.DEF  :   15     M3990
     GEAR.SRC      AN:639   AN:648    S:657    R:724

MG_SPINDLE_ON_GEAR                      GLB_TCMB.DEF  :   14     M3991
     GEAR.SRC       S:709    R:725

MG_SPINDLE_RPM_ZERO                     GLB_TCMB.DEF  :   13     M3992
     - not used -
MG_T_I_N_SUPERVISION                    GLB_TCMB.DEF  :   12     M3993
     DIRCBUT.SRC   AN:239
     AXIS.SRC       O:459
     SPINDLE.SRC    O:549    O:562    O:577
     HELPDIAG.SRC   =:797

MG_SPI_POS_M19_R_M0X                    GLB_TCMB.DEF  :   11     M3994
     SPINDLE.SRC    L:530   LN:535    L:547

MG_SPINDLE_OFF_M05                      GLB_TCMB.DEF  :   10     M3995
     SPINDLE.SRC    O:548    L:561    L:576

MG_SPINDLE_ON_M04                       GLB_TCMB.DEF  :    9     M3996
     SPINDLE.SRC    L:543

MG_SPINDLE_ON_M03                       GLB_TCMB.DEF  :    8     M3997
     SPINDLE.SRC    L:539

MG_ZERO_MARKER                          GLB_TCMB.DEF  :    7     M3998
     INITI472.SRC   L:33     R:34
     GEAR.SRC      PS:721

MG_ONE_MARKER                           GLB_TCMB.DEF  :    6     M3999
     INITI472.SRC  LN:30     S:31     L:36
     AXIS.SRC       L:376
     GEAR.SRC       L:723

NP_M4000_S_IN_POSITION                  GLB_NC.DEF    :    7     M4000
     SPINDLE.SRC    A:536

NP_M4001_S_ANALOG_NOT_IN                GLB_NC.DEF    :    8     M4001
     SPINDLE.SRC    A:587

NP_M4002_S_ANALOG_0_V                   GLB_NC.DEF    :    9     M4002
     SPINDLE.SRC    L:567   ON:571
     GEAR.SRC      PS:664

PN_M4005_S_M03_ANALOG_VO                GLB_NC.DEF    :   11     M4005
     SPINDLE.SRC    S:540    R:544    R:551   LN:554    O:563
                    L:569
```

```
PN_M4006_S_M04_ANALOG_VO                GLB_NC.DEF    :   12     M4006
     SPINDLE.SRC    R:541    S:545    R:552   AN:555    O:564
                    O:570

PN_M4007_S_M05_0V_STATUS                GLB_NC.DEF    :   13     M4007
     SPINDLE.SRC    =:556

PN_M4009_S_SWING_LEFT                   GLB_NC.DEF    :   39     M4009
     GEAR.SRC    WHIL:674    R:727

PN_M4010_S_SWING_RIGHT                  GLB_NC.DEF    :   38     M4010
     GEAR.SRC       =:673    R:726

PN_M4012_S_CLOSE_LOOP_OP                GLB_NC.DEF    :   14     M4012
     SPINDLE.SRC    L:558    R:559    S:565

NP_M4070_STROBE_G_CODE                  GLB_NC.DEF    :   20     M4070
     GEAR.SRC      LN:615   AN:637   AN:647    O:655    L:705
                   LN:708

NP_M4071_STROBE_S_CODE                  GLB_NC.DEF    :   21     M4071
     - not used -
NP_M4072_STROBE_M_FUNCTI                GLB_NC.DEF    :   22     M4072
     M_FUNCT.SRC    L:183
     SPINDLE.SRC   LN:583    L:586

PN_M4090_QUIT_G_CODE                    GLB_NC.DEF    :   28     M4090
     GEAR.SRC       R:616    S:706

PN_M4091_QUIT_S_CODE                    GLB_NC.DEF    :   29     M4091
     - not used -
PN_M4092_QUIT_M_FUNCTION                GLB_NC.DEF    :   30     M4092
     SPINDLE.SRC    =:592

PN_M4130_STROBE_PLC_POS_                GLB_NC.DEF    :   36     M4130
     SPINDLE.SRC    O:572   AN:588

PN_M4134_STROBE_G_STEP_R                GLB_NC.DEF    :   37     M4134
     GEAR.SRC       S:607   AN:638    L:654

NP_M4150_MANUEL_MODE                    GLB_NC.DEF    :   44     M4150
     DIRCBUT.SRC    L:252

NP_M4151_ELECTRONIC_HAND                GLB_NC.DEF    :   45     M4151
     DIRCBUT.SRC    O:253    A:286

NP_M4155_REFERENCE_MODE                 GLB_NC.DEF    :   46     M4155
     MAIN_PGM.SRC   O:14
     DIRCBUT.SRC    O:254

NP_M4156_SOFTKEY_MANUAL_                GLB_NC.DEF    :   47     M4156
     DIRCBUT.SRC    O:255

NP_M4172_1_PLC_AFTER_POW                GLB_NC.DEF    :   48     M4172
     MAIN_PGM.SRC   L:5
     GEAR.SRC       L:605

NP_M4172_1_PLC_AFTER_COM                GLB_NC.DEF    :   49     M4173
     MAIN_PGM.SRC   O:6
     GEAR.SRC       O:606

NP_M4172_1_PLC_AFTER_MP_                GLB_NC.DEF    :   50     M4174
     MAIN_PGM.SRC   O:7

NP_M4203_ERROR_MODUL_9XX                GLB_NC.DEF    :   56     M4203
     MAIN_PGM.SRC   L:2      L:26
     AXIS.SRC       L:354
     SPINDLE.SRC    L:601
     HELPDIAG.SRC   L:734    L:750    L:775    L:784
     SOFTKEYS.SRC   L:856    L:869    L:891    L:915    L:939

NP_M4300_POWERFAILON_MP4                GLB_NC.DEF    :   62     M4300
     INITI472.SRC   L:44

PN_M4560_NC_STOP_0_ACTIV                GLB_NC.DEF    :   68     M4560
     DIRCBUT.SRC    =:240

PN_M4561_RAPIDE                         GLB_NC.DEF    :   69     M4561
     DIRCBUT.SRC    =:243

PN_M4562_AXIS_BUTTON_LAT                GLB_NC.DEF    :   70     M4562
     DIRCBUT.SRC    =:250

PN_M4563_FEED_ENABLE                    GLB_NC.DEF    :   71     M4563
     DIRCBUT.SRC    =:246

PN_M4564_NC_START                       GLB_NC.DEF    :   72     M4564
     DIRCBUT.SRC    =:249
```

```
NP_M4572_ENABLE_JOG_MODE                GLB_NC.DEF    :   74     M4572
     INITI472.SRC   S:37
     DIRCBUT.SRC    L:285

PN_ERROR_MOD_9167                       GLB_TCMB.DEF  :   63     M4800
     INITI472.SRC   S:53

PN_ERROR_MOD_9002                       GLB_TCMB.DEF  :   64     M4801
     MAIN_PGM.SRC   S:3

PN_ERROR_MOD_9005                       GLB_TCMB.DEF  :   65     M4802
     MAIN_PGM.SRC   S:27

PN_ERROR_MOD_9161                       GLB_TCMB.DEF  :   66     M4803
     AXIS.SRC       S:355

PN_ERROR_SUBMIT_QUEUE_FU                GLB_TCMB.DEF  :   67     M4804
     INITI472.SRC   S:74

PN_ERROR_NOT_USED_M_FUNC                GLB_TCMB.DEF  :   68     M4805
     SPINDLE.SRC   AN:591

PN_ERROR_9171_SPI_POS                   GLB_TCMB.DEF  :   69     M4806
     SPINDLE.SRC    S:602

PN_ERROR_SERVO_ACTIV                    GLB_TCMB.DEF  :   70     M4807
     HELPDIAG.SRC   S:762    O:796

PN_ERROR_TEMP_POWERSUPPL                GLB_TCMB.DEF  :   71     M4808
     HELPDIAG.SRC   S:738    L:792

PN_ERROR_I2T_CAUTION                    GLB_TCMB.DEF  :   72     M4809
     HELPDIAG.SRC   S:743    O:793

PN_ERROR_MODUL_9XXX_SUPE                GLB_TCMB.DEF  :   73     M4810
     HELPDIAG.SRC   S:735    S:751    S:776    S:785

PN_ERROR_UTILIZATION_MOT                GLB_TCMB.DEF  :   74     M4811
     HELPDIAG.SRC   S:782    O:795

PN_ERROR_MOTOR_TEMP                     GLB_TCMB.DEF  :   75     M4812
     HELPDIAG.SRC   S:773    O:794

PN_ERROR_I2T_LIMITATION                 GLB_TCMB.DEF  :   76     M4813
     HELPDIAG.SRC   S:747

MG_FUNCTION_ON                          GLB_TCMB.DEF  :   77     M4814
     SOFTKEYS.SRC   L:841    R:842   LN:846    S:847

PN_ERROR_GEAR_SWITCHING                 GLB_TCMB.DEF  :   82     M4815
     GEAR.SRC       =:644    L:646

PN_ERROR_SPINDLE_ZERO                   GLB_TCMB.DEF  :   83     M4816
     GEAR.SRC       S:662

__B0                                    INITI472.SRC  :  125     B0
     INITI472.SRC  PL:139

BG_MPAXIS.X_CA_PA                       GLB_TCMB.DEF  :   35     B4088
     - not used -
BG_MPAXIS.0_CA_PA_1                     GLB_TCMB.DEF  :   36     B4088
     - not used -
BG_MPAXIS.1_CA_PA_2                     GLB_TCMB.DEF  :   37     B4089
     - not used -
BG_MPAXIS.2_CA_PA_3                     GLB_TCMB.DEF  :   38     B4090
     - not used -
BG_MPAXIS.3_CA_PA_4                     GLB_TCMB.DEF  :   39     B4091
     - not used -
BG_MPAXIS.4_CA_PA_5                     GLB_TCMB.DEF  :   40     B4092
     - not used -
BG_MPSPIN.0_CA_PA_S                     GLB_TCMB.DEF  :   41     B4093
     - not used -
__W0                                    INITI472.SRC  :  132     W0
     INITI472.SRC  PL:146

NP_W256_G_CODE_SPINDLE                  GLB_NC.DEF    :   99     W256
     GEAR.SRC       =:610 CASE:676    L:713

NP_W260_M_CODE                          GLB_NC.DEF    :   98     W260
     M_FUNCT.SRC    L:176    L:179

NP_W274_BUTTON_KEY_CODE                 GLB_NC.DEF    :  102     W274
     SOFTKEYS.SRC   L:824    =:828

NP_W302_NUMBER_PLC_SOFT_                GLB_NC.DEF    :  101     W302
     SOFTKEYS.SRC   L:809 CASE:812    =:821

NP_W1026_AXIS_IN_POSITIO                GLB_NC.DEF    :   83     W1026
     AXIS.SRC       O:385    O:395    O:451
```

```
NP_W1032_REFERENCE_NECES                GLB_NC.DEF    :   84     W1032
     MAIN_PGM.SRC   L:12

PN_W1038_CLOSED_LOOP_OPE                GLB_NC.DEF    :   85     W1038
     AXIS.SRC       L:360    =:362    L:429    =:431    L:483
                    =:485

PN_W1040_CLOSED_LOOP_OPE                GLB_NC.DEF    :   86     W1040
     AXIS.SRC       L:364    =:366    L:434    =:436    L:479
                    =:481

PN_W1042_SUPERVISION_INA                GLB_NC.DEF    :   87     W1042
     AXIS.SRC       L:368    =:370    L:442    =:444

PN_W1044_ACTUL_NOMINAL_T                GLB_NC.DEF    :   88     W1044
     AXIS.SRC       L:372    =:374    L:438    =:440

PN_W1046_MANUEL_DIRCETIO                GLB_NC.DEF    :   89     W1046
     DIRCBUT.SRC    =:294

PN_W1048_MANUEL_DIRECTIO                GLB_NC.DEF    :   90     W1048
     DIRCBUT.SRC    =:296

PN_W1050_JOG_MODE_POSIT_                GLB_NC.DEF    :   91     W1050
     DIRCBUT.SRC    =:289

PN_W1052_JOG_MODE_POSIT_                GLB_NC.DEF    :   92     W1052
     DIRCBUT.SRC    =:291

PN_W1054_REFERENCE_ENDSW                GLB_NC.DEF    :   93     W1054
     REF_ENDL.SRC   =:234

WG_MOTOR_TEMP                           GLB_TCMB.DEF  :   50     W4068
     HELPDIAG.SRC  >=:772

WG_MOTOR_TEMP_1                         GLB_TCMB.DEF  :   51     W4068
     INITI472.SRC   =:56

WG_MOTOR_TEMP_2                         GLB_TCMB.DEF  :   52     W4070
     INITI472.SRC   =:58

WG_MOTOR_TEMP_3                         GLB_TCMB.DEF  :   53     W4072
     INITI472.SRC   =:60

WG_MOTOR_TEMP_4                         GLB_TCMB.DEF  :   54     W4074
     INITI472.SRC   =:62

WG_MOTOR_TEMP_5                         GLB_TCMB.DEF  :   55     W4076
     INITI472.SRC   =:64

WG_MOTOR_TEMP_S                         GLB_TCMB.DEF  :   56     W4078
     INITI472.SRC   =:66

WG_ACTIVE_PWM_AXIS                      GLB_TCMB.DEF  :   47     W4082
     INITI472.SRC   =:128    L:131    =:134
     AXIS.SRC       A:350

WG_SERVO_ENABLE_INTERNAL                GLB_TCMB.DEF  :   46     W4084
     AXIS.SRC       =:351
     HELPDIAG.SRC  LN:752    L:754

WG_MP10_ACTIVE_AXIS                     GLB_TCMB.DEF  :   45     W4086
     - not used -
__D0                                    INITI472.SRC  :  139     D0
     INITI472.SRC  PL:153

PN_D756_S_NOMINAL_RPM_PL                GLB_NC.DEF    :   97     D756
     GEAR.SRC       =:612    L:716

I_REF_ENDSWITCH_1_AXIS                  GLB_IOMB.DEF  :    7     I0
     REF_ENDL.SRC   L:222

I_REF_ENDSWITCH_2_AXIS                  GLB_IOMB.DEF  :    8     I1
     REF_ENDL.SRC   L:224

I_REF_ENDSWITCH_3_AXIS                  GLB_IOMB.DEF  :    9     I2
     REF_ENDL.SRC   L:226

I_SERVO_READY_1                         GLB_IOMB.DEF  :   28     I3
     AXIS.SRC      ON:306   ON:457
     SPINDLE.SRC   ON:579

I_SERVO_READY_2                         GLB_IOMB.DEF  :   29     I3
     AXIS.SRC      ON:307   ON:458
     SPINDLE.SRC   ON:580

I_FEED_ENABLE                           GLB_IOMB.DEF  :   13     I3
     DIRCBUT.SRC    L:245
```

```
I_NOT_EMERGENCY_STOP                     GLB_IOMB.DEF  :   10     I3
     AXIS.SRC      LN:305   ON:456
     SPINDLE.SRC   ON:550   ON:578
     GEAR.SRC      ON:649

I_REF_ENDSWITCH_4_AXIS                   GLB_IOMB.DEF  :   11     I4
     REF_ENDL.SRC   L:228

I_REF_ENDSWITCH_5_AXIS                   GLB_IOMB.DEF  :   12     I6
     REF_ENDL.SRC   L:230

I_1_AXIS_MINUS                           GLB_IOMB.DEF  :   15     I128
     DIRCBUT.SRC    L:272

I_2_AXIS_MINUS                           GLB_IOMB.DEF  :   17     I129
     DIRCBUT.SRC    L:274

I_3_AXIS_MINUS                           GLB_IOMB.DEF  :   19     I130
     DIRCBUT.SRC    L:276

I_4_AXIS_MINUS                           GLB_IOMB.DEF  :   21     I131
     DIRCBUT.SRC    L:278

I_1_AXIS_PLUS                            GLB_IOMB.DEF  :   14     I133
     DIRCBUT.SRC    L:261

I_2_AXIS_PLUS                            GLB_IOMB.DEF  :   16     I134
     DIRCBUT.SRC    L:263

I_3_AXIS_PLUS                            GLB_IOMB.DEF  :   18     I135
     DIRCBUT.SRC    L:265

I_4_AXIS_PLUS                            GLB_IOMB.DEF  :   20     I136
     DIRCBUT.SRC    L:267

I_NC_START                               GLB_IOMB.DEF  :   24     I146
     DIRCBUT.SRC    L:248

I_NC_STOP                                GLB_IOMB.DEF  :   25     I147
     DIRCBUT.SRC    L:238

I_RAPID_BUTTON                           GLB_IOMB.DEF  :   26     I148
     DIRCBUT.SRC    L:242

O_1_AXIS_ENABLE                          GLB_IOMB.DEF  :   39     O0
     AXIS.SRC       =:508

O_2_AXIS_ENABLE                          GLB_IOMB.DEF  :   40     O1
     AXIS.SRC       =:510

O_3_AXIS_ENABLE                          GLB_IOMB.DEF  :   41     O2
     AXIS.SRC       =:512

O_4_AXIS_ENABLE                          GLB_IOMB.DEF  :   42     O3
     AXIS.SRC       =:514

O_5_AXIS_ENABLE                          GLB_IOMB.DEF  :   43     O4
     AXIS.SRC       =:516

O_1_CLAMPING                             GLB_IOMB.DEF  :   45     O7
     AXIS.SRC       =:519

O_2_CLAMPING                             GLB_IOMB.DEF  :   46     O8
     AXIS.SRC       =:521

O_3_CLAMPING                             GLB_IOMB.DEF  :   47     O9
     AXIS.SRC       =:523

O_4_CLAMPING                             GLB_IOMB.DEF  :   48     O10
     AXIS.SRC       =:525

O_5_CLAMPING                             GLB_IOMB.DEF  :   49     O11
     AXIS.SRC       =:527

O_SPINDLE_SERVO_ENABLE                   GLB_IOMB.DEF  :   51     O15
     AXIS.SRC       L:334
     SPINDLE.SRC    S:574    R:581

TS_1_CLAMPING                            GLB_TCMB.DEF  :   91     T0
     AXIS.SRC       =:417   LN:419    =:469   LN:471

TS_2_CLAMPING                            GLB_TCMB.DEF  :   93     T1
     - not used -
TS_3_CLAMPING                            GLB_TCMB.DEF  :   95     T2
     - not used -
TS_4_CLAMPING                            GLB_TCMB.DEF  :   97     T3
     - not used -
TS_5_CLAMPING                            GLB_TCMB.DEF  :   99     T4
     - not used -
TS_M_FUNC_DELAY                          GLB_TCMB.DEF  :  101     T7
     SPINDLE.SRC    =:584   AN:589
```

```
TS_1_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  103     T8
     HELPDIAG.SRC  B=:753

TS_2_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  105     T9
     - not used -
TS_3_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  107     T10
     - not used -
TS_4_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  109     T11
     - not used -
TS_5_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  111     T12
     - not used -
TS_6_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  113     T13
     - not used -
TS_7_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  115     T14
     - not used -
TS_8_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  117     T15
     - not used -
TS_GEAR_TIMEOUT                          GLB_TCMB.DEF  :  120     T20
     GEAR.SRC       =:620   AN:641

TS_GREAR_TOGGEL_ALL                      GLB_TCMB.DEF  :  122     T21
     GEAR.SRC       =:669

TS_GREAR_TOGGEL_RIGHT                    GLB_TCMB.DEF  :  124     T22
     GEAR.SRC       =:670

TR_1_CLAMPING                            GLB_TCMB.DEF  :   92     T48
     AXIS.SRC      AN:420   AN:472

TR_2_CLAMPING                            GLB_TCMB.DEF  :   94     T49
     - not used -
TR_3_CLAMPING                            GLB_TCMB.DEF  :   96     T50
     - not used -
TR_4_CLAMPING                            GLB_TCMB.DEF  :   98     T51
     - not used -
TR_5_CLAMPING                            GLB_TCMB.DEF  :  100     T52
     - not used -
TR_M_FUNC_DELAY                          GLB_TCMB.DEF  :  102     T55
     SPINDLE.SRC   AN:590

TR_1_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  104     T56
     HELPDIAG.SRC  LB:759

TR_2_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  106     T57
     - not used -
TR_3_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  108     T58
     - not used -
TR_4_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  110     T59
     - not used -
TR_5_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  112     T60
     - not used -
TR_6_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  114     T61
     - not used -
TR_7_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  116     T62
     - not used -
TR_8_SERVO_SUPERVISON                    GLB_TCMB.DEF  :  118     T63
     - not used -
TR_GEAR_TIMOUT                           GLB_TCMB.DEF  :  121     T68
     GEAR.SRC      AN:642   LN:661

TR_GREAR_TOGGEL_ALL                      GLB_TCMB.DEF  :  123     T69
     GEAR.SRC      LN:668

TR_GREAR_TOGGEL_RIGHT                    GLB_TCMB.DEF  :  125     T70
     GEAR.SRC       L:672

NC_TYPE_DIGITAL                          CONFIG.DEF    :   24     K0
     - not used -
FIRST_PL                                 CONFIG.DEF    :    7     K0
     - not used -
IO_MB410                                 CONFIG.DEF    :   10     K0
     - not used -
IO_BELEGUNG                              CONFIG.DEF    :   12     K0
     - not used -
IO_SPG                                   CONFIG.DEF    :   11     K1
     - not used -
MAX_NC_AXIS                              CONFIG.DEF    :   32     K4
     AXIS.SRC      >=:329

KG_ERROR_MODUL_9200                      GLB_TCMB.DEF  :   78     K15
     SOFTKEYS.SRC  PS:858   PS:871

KG_ERROR_MODUL_9220                      GLB_TCMB.DEF  :   79     K16
     - not used -
KG_ERROR_MODUL_9210                      GLB_TCMB.DEF  :   80     K17
     SOFTKEYS.SRC  PS:906   PS:930   PS:951

KG_ERROR_MODUL_9202                      GLB_TCMB.DEF  :   81     K18
     SOFTKEYS.SRC  PS:893   PS:917   PS:941
```

```
MOTOR_TEMP_3                          CONFIG.DEF    :   17     K255
     INITI472.SRC   L:59

MOTOR_TEMP_5                          CONFIG.DEF    :   19     K255
     INITI472.SRC   L:63

MOTOR_TEMP_S                          CONFIG.DEF    :   20     K255
     INITI472.SRC   L:65

MOTOR_TEMP_2                          CONFIG.DEF    :   16     K255
     INITI472.SRC   L:57

MOTOR_TEMP_4                          CONFIG.DEF    :   18     K255
     INITI472.SRC   L:61

MOTOR_TEMP_1                          CONFIG.DEF    :   15     K255
     INITI472.SRC   L:55


________________________________ Project Info ________________________________


Used Files:
File                                                         Class   Date
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\MAIN_PGM.SRC               Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\CONFIG.DEF                 Define  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\GLB_TCMB.DEF               Define  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\GLB_NC.DEF                 Define  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\GLB_IOMB.DEF               Define  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\INITI472.SRC               Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\M_FUNCT.SRC                Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\REF_ENDL.SRC               Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\DIRCBUT.SRC                Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\AXIS.SRC                   Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\SPINDLE.SRC                Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\GEAR.SRC                   Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\HELPDIAG.SRC               Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\SOFTKEYS.SRC               Module  29.10.98
C:\HEIDEN~1\PLC_GB\JHPGM\410_430D\MAC_LIB.DEF                Include 29.10.98

Compiler Memory Assignment:
                    Range      Max.Local/in File      Global/Static used
Marker (remanent) :  200- 999      -                   990- 999
Marker (nonrem. ) : 3200-3999  3200-3215 REF_ENDL.SRC 3967-3999
Byte   (remanent) :    4-   0      -                   120- 127
Byte   (nonrem. ) : 2048-   0  2048-2051 AXIS.SRC     4052-4095

Vacant Memory Fragments:
Marker (remanent) :    0     Marker (nonremanent) :    0
Byte (remanent)   :    3     Byte (nonremanent)   :    1
Word (remanent)   :    0     Word (nonremanent)   :    0
Double (remanent) :    0     Double (nonremanent) :    0
```

7.7 Commands

7.7.1 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
Increment		
	INC	Increment operand
	INCW	Increment word accumulator
	INCX	Increment index register

Group of functions	Syntax	Function
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

7.7.2 Execution Times

Commands

Specific execution times are assigned to the PLC commands.
If you use the index register, these execution times increase by 0.05 to 0.2 µs per indexed command.

Commands with operands	**Execution times**		
	I/O/M/C/T	**B/W/D/K**	**String**
L, LN, L–, =, =N, =–	Up to 0.5 µs	Up to 0.5 µs	1.0 to 15 µs
A, AN, O, ON, XO, XON	Up to 0.5 µs	Up to 0.5 µs	–
S, SN, R, RN	Up to 0.5 µs	–	–
OVWR	–	–	1.0 to 15 µs
+	–	Up to 0.5 µs	1.0 to 15 µs
–, x	–	Up to 0.5 µs	–
/, MOD	–	0.1 to 1.5 µs	–
==, <, >, <=, >=, <>	–	Up to 0.5 µs	1.0 to 15 µs
< <, >>	–	Up to 0.5 µs	–
BT, BS, BR	–	Up to 0.5 µs	–
LB, LW	0.1 to 1.5 µs	–	–
LD	1.0 to 15 µs	–	–
=B, =W	1.0 to 15 µs	–	–
=D	1.0 to 15 µs	–	–
PL, PS	0.1 to 1.5 µs	0.1 to 1.5 µs	–
CASE	–	0.1 to 1.5 µs	–
SUBM, RPLY, CAN	1.0 to 15 µs	–	–
INC, DECN	–	Up to 0.5 µs	–

Commands without operands	Execution times
PLL, PLW, PSL, PSW, PSX, PLX	Up to 0.5 µs
A[..., XON[..........] for these commands	Up to 0.5 µs
+[..., -[..., X[..........] for these commands	Up to 0.5 µs
/[..., MOD[..........] for these commands	Up to 0.5 s 0.1 to 1.5 [µs]
==[..., <>[..........] for these commands	Up to 0.5 µs
LBL	0 µs
JP, JPT, JPF	Up to 0.5 µs
CM, CMT, CMF in their own source module	0.1 to 1.5 µs
CM, CMT, CMF onto a global label	1.0 to 15 µs
EM, EMT, EMF	Up to 0.5 µs
IFc, ELSE, UNTILc, WHILEc	Up to 0.5 µs
ENDI, REPEAT, ENDW	0 µs
CM with Case (see previous table)	0 µs
ENDC	0 µs
SUBM, RPLY, CAN	1.0 to 15 µs
LX, =X	Up to 0.5 µs
INCW, DECW, INCX, DECX	Up to 0.5 µs

Execution times for modules that run as submit jobs

These modules communicate with other parts of the control software. This results in inevitable waiting and response times so that execution times cannot be specified here. If required, determine the execution times by trial.

Some factors that influence response times are:

- CPU load from processing an NC program run
- Load on the file system, for example from copying
- Load on the CPU and the video system, e.g., through PLC windows

Execution times for modules that run in the cyclic program

Unless otherwise indicated, the execution time of such modules lies between 10 µs and 100 µs. Unnecessary calls should therefore be avoided (e.g. cyclic setting of the pulse release, of a soft-key row, etc.).

For some modules the execution times are longer:

Module no.	Function	Time
9002	Reading 64 inputs from a PL I/O unit	450 µs
9005	Writing to 32 outputs from a PL I/O unit	280 µs
9004	Generating edge markers (e.g. 100 pieces)	150 µs
9003	Reading an analog input from a PL I/O unit	150 µs

7.7.3 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	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 1 x x x x x x x	1
Gate the content of the logic accumulator and input I5 with AND.	A I5	x ..... x x x x x x x x 0 x x x x x x x	0
Assign the gating result to output O2.	= O2	x ..... x x x x x x x x 0 x x x x x x x	0

Word processing with the LOAD command

Syntax: L (LOAD)

Operands: B, W, D, K

Action:

Load the value of the addressed operand, or of a constant, into the word accumulator. If necessary, the accumulator is supplemented with the correct algebraic sign. In contrast to logical operations, you must always begin a sequence of word gating operations with an L command. You cannot replace the L command with a logical gating instruction.

Example:

Gate a constant and byte B5 with AND, and assign the result to byte B8.
Initial state:

Constant	54	= 36 (hex)
Byte	B5	= 2A (hex)
Output	B8	= ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	7 0
Load the constant into the word accumulator.	**L K+54**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	
Gate the contents of word accumulator and byte B5 with AND.	**A B5**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0	0 0 1 0 1 0 1 0
Assign the gating result to byte B8.	**= B8**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0	0 0 1 0 0 0 1 0

7.7.4 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	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the inverted operand content into the logic accumulator.	LN I4	x ..... x x x x x x x x 1 x x x x x x x	0
Gate the content of the logic accumulator and input I5 with AND.	A I5	x ..... x x x x x x x x 1 x x x x x x x	1
Assign the gating result to output O2.	= O2	x ..... x x x x x x x x 1 x x x x x x x	1

Word processing with the LOAD NOT command

Syntax: LN (LOAD NOT)

Operands: B, W, D, K

Action:

Load the complement of the addressed operand, or of a constant, into the word accumulator. If necessary, the accumulator is supplemented with the correct algebraic sign. In contrast to logical operations, you must always begin a sequence of word gating operations with an L command. You cannot replace the L command with a logical gating instruction.

Example:

Gate the complement of byte B6 and byte B5 with AND, and assign the result to byte B8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = B6 (hex)
Byte B8 = ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	7 0
Invert byte 6, and load into the word accumulator.	**LN B6**	0 ..... 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1	1 0 1 1 0 1 1 0
Gate the contents of word accumulator and byte B5 with AND.	**A B5**	0 ..... 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	0 0 1 0 1 0 1 0
Assign the gating result to byte B8.	**= B8**	0 ..... 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	0 0 0 0 1 0 0 0

7.7.5 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 TNC supplements 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	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	7 0
Load byte B5 into the word accumulator, invert the algebraic sign.	**L– B5**	1 ..... 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 (– 1 5)	0 0 0 0 1 1 1 1 (+1 5)
Add the contents of the word accumulator and byte B6.	**+ B6**	0 ..... 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 (+ 5)	0 0 0 1 0 1 0 0 (+2 0)
Assign the gating result to byte B8.	**= B8**	0 ..... 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 (+ 5)	0 0 0 0 0 1 0 1 (+5)

To aid understanding of this example, the contents of the accumulator and operands are shown as decimal values in parentheses.

7.7.6 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 TNC saves the entered operand address in the accumulator as LSB, the entered address +1 as LSB +1 etc. The last (8th) operand is now the MSB! If necessary, the TNC supplements the accumulator with the correct algebraic sign.

Example:

A pure-binary coded value is read through inputs I3 to I10 and saved in byte B8 in order to process it later.
Initial state:

Input	I3	= 1	Input	I7	= 0
Input	I4	= 1	Input	I8	= 1
Input	I5	= 1	Input	I9	= 1
Input	I6	= 0	Input	I10	= 0

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	I10.......... 3
Load inputs I3 to I10 into the accumulator (bit 0 to bit 7).	**LB I3**	0 .... 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1	0 1 1 0 0 1 1 1
		31..15 .... 7 0	7 0
Assign accumulator contents to byte 8.	**= B8**	0 .... 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1	0 1 1 0 0 1 1 1

7.7.7 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 TNC saves the entered operand address in the accumulator as LSB, the entered address +1 as LSB +1 etc. The last (16th) operand is now the MSB! If necessary, the TNC supplements the accumulator with the correct algebraic sign.

Example:

See example command LB. Use command LW in the same way as LB. However, the TNC processes 16 operands.

7.7.8 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 TNC saves the entered operand address in the accumulator as LSB, the entered address +1 as LSB +1 etc. The last (32nd) operand is now the MSB! If necessary, the TNC supplements the accumulator with the correct algebraic sign.

Example:

See example command LB. Use command LD in the same way as LB. However, the TNC processes 32 operands.

7.7.9 ASSIGN (=)

Logic processing with the ASSIGN command

Syntax: = (STORE)

Operands: M, I, O, T, C

Action:

Assign the content of the logic accumulator to the addressed operand. Use the = command only at the end of a sequence of logical gating operations in order to transfer a gating result to a logic operand. This command can be used several times in succession (see example).

Example:

Gate the inputs I4 and I5 with AND, and assign the result to outputs O2 and O5.
Initial state:
Input I4 = 1
Input I5 = 0
Output O2 = ?
Output O5 = ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 1 x x x x x x x	1
Gate the content of the logic accumulator and input I5 with AND.	A I5	x ..... x x x x x x x x 0 x x x x x x x	0
Assign the gating result to output O2.	= O2	x ..... x x x x x x x x 0 x x x x x x x	0
Assign the gating result to output O5.	= O5	x ..... x x x x x x x x 0 x x x x x x x	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	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	7 0
Load the constant into the word accumulator.	**L K+54**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	
Assign the contents of the word accumulator to byte B8.	**= B8**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	00110110
Gate the contents of word accumulator and byte B5 with AND.	**A B5**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0	00101010
Assign the gating result to byte B8.	**= B8**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0	00100010
Assign the gating result to byte B10.	**= B10**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0	00100010

7.7.10 ASSIGN BYTE (B=)

Syntax: B= (STORE BYTE)

Operands: M, I, O, T, C

Action:

Assign 8 markers from the word accumulator to inputs, outputs, timer or counters with ascending numbering. Every bit occupies an operand. The TNC assigns the LSB in the accumulator to the operand address specified in the command, the specified address +1 as LSB +1 etc. The last (8th) operand is assigned the MSB.

Example:

See example of command D=. Use command B= in the same way as D=. However, the TNC processes 8 operands.

7.7.11 ASSIGN WORD (W=)

Syntax: W= (STORE WORD)

Operands: M, I, O, T, C

Action:

Assign 16 markers from the word accumulator to inputs, outputs, timer or counters with ascending numbering. Every bit occupies an operand. The TNC 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:

See example of command D=. Use command B= in the same way as D=. However, the TNC processes 16 operands.

7.7.12 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, timer or counters with ascending numbering. Every bit occupies an operand. The TNC 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:

Transfer a certain bit pattern, located in word W8, to the output addresses O5 to O20.
Initial state:
Word W8 = 36FF (hex)

Function	STL	Accumulator contents (dec or [bit])	Operand contents [bit]
		31...15 .... 7 0	15 .... 8 7 ... 0
Load content of word W8 into the word accumulator.	**L W8**	0 ..... 0 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1	0 0110110 11111111
		31...15 .... 7 0	O20.....O13 O12 ..... O5
Assign accumulator content to outputs O5 to O20.	**W= O5**	0 ..... 0 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1	0 0110110 11111111

7.7.13 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 example 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 example command ASSIGN (=).

7.7.14 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 example command ASSIGN (=).

7.7.15 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	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 1 x x x x x x x	1
Gate the content of the logic accumulator and input I5 with OR	O I5	x ..... x x x x x x x x 1 x x x x x x x	0
Since the result of the operation is 1, set output O2.	S O2	x ..... x x x x x x x x 1 x x x x x x x	1
Since the result of the operation is 1, set marker M500.	S M500	x ..... x x x x x x x x 1 x x x x x x x	1

7.7.16 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	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 1 x x x x x x x	1
Gate the content of the logic accumulator and input I5 with OR	O I5	x ..... x x x x x x x x 1 x x x x x x x	0
Since the result of the operation is 1, reset output O2.	R O2	x ..... x x x x x x x x 1 x x x x x x x	0
Since the result of the operation is 1, reset marker M500.	R M500	x ..... x x x x x x x x 1 x x x x x x x	0

7.7.17 SET NOT (SN)

Syntax: SN (SET NOT)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 0, then set the addressed operand to 1, otherwise do not change it. Use the SN command at the end of a sequence of logical gating operations in order to influence an operand independently from the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 0, then set output O2 and marker M500.
Initial state:

Input	I4	= 0
Input	I5	= 0
Output	O2	= ?
Marker	M500	= ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 0 x x x x x x x	0
Gate the content of the logic accumulator and input I5 with OR	O I5	x ..... x x x x x x x x 0 x x x x x x x	0
Since the result of the operation is 0, set output O2.	SN O2	x ..... x x x x x x x x 0 x x x x x x x	1
Since the result of the operation is 0, set marker M500.	SN M500	x ..... x x x x x x x x 0 x x x x x x x	1

7.7.18 RESET NOT (RN)

Syntax: RN (RESET NOT)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 0, then set the addressed operand to 0, otherwise do not change it. Use the RN command at the end of a sequence of logical gating operations in order to influence an operand independently from the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 0, then reset output O2 and marker M500.
Initial state:

Input	I4	= 0
Input	I5	= 0
Output	O2	= ?
Marker	M500	= ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 0 x x x x x x x	0
Gate the content of the logic accumulator and input I5 with OR	O I5	x ..... x x x x x x x x 0 x x x x x x x	0
Since the result of the operation is 0, reset output O2.	RN O2	x ..... x x x x x x x x 0 x x x x x x x	0
Since the result of the operation is 0, reset marker M500.	RN M500	x ..... x x x x x x x x 0 x x x x x x x	0

7.7.19 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 TNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with AND, and assign the result to output O2.
Initial state:

Input	I4	= 1
Input	I5	= 0
Output	O2	= ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 1 x x x x x x x	1
Gate the content of the logic accumulator and input I5 with AND.	A I5	x ..... x x x x x x x x 0 x x x x x x x	0
Assign the gating result to output O2.	= O2	x ..... x x x x x x x x 0 x x x x x x x	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 TNC saves the result of the operation in the word accumulator.

Example:

Gate the content of byte B5 and byte 6 with AND, and assign the result to byte B8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = 36 (hex)
Byte B8 = ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	7 0
Load byte B6 into the word accumulator.	**L B6**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	00110110
Gate the contents of word accumulator and byte B5 with AND.	**A B5**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0	00101010
Assign the gating result to byte B8.	**= B8**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0	00100010

7.7.20 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 TNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with AND NOT, and assign the result to output O2.
Initial state:

Input	I4	= 1
Input	I5	= 1
Output	O2	= ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 1 x x x x x x x	1
Gate the content of logic accumulator and input I5 with AND NOT.	AN I5	x ..... x x x x x x x x 0 x x x x x x x	1
Assign the gating result to output O2.	= O2	x ..... x x x x x x x x 0 x x x x x x x	0

Word processing with the AND NOT command

Syntax: AN (AND NOT)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with AND NOT. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The TNC saves the result of the operation in the word accumulator.

Example:

Gate the content of words W4 and W6 with AND NOT, and assign the result to word W8.
Initial state:
Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Function	**STL**	**Accumulator contents [bit]**	**Operand contents [bit]**
		31..15 .... 7 0	1 5.......... 8 7............ 0
Load W6 into the word accumulator.	**L W6**	0 ..... 0 0 1 1 1 1 0 0 0 0 1 1 0 1 1 0	0 0111100 00110110
Gate the content of word accumulator and word W4 with AND NOT.	**AN W4**	0 ..... 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0	0 0110110 10101010
Assign the gating result to word W8.	**= W8**	0 ..... 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0	0 0001000 00010100

7.7.21 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 TNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with OR, and assign the result to output O2.
Initial state:

Input	I4	= 0
Input	I5	= 1
Output	O2	= ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		3 ..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 0 x x x x x x x	0
Gate the content of the logic accumulator and input I5 with OR	O I5	x ..... x x x x x x x x 1 x x x x x x x	1
Assign the gating result to output O2.	= O2	x ..... x x x x x x x x 1 x x x x x x x	1

Word processing with the OR command

Syntax: O (OR)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with OR. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The TNC saves the result of the operation in the word accumulator.

Example:

Gate the content of byte B5 and byte 6 with OR, and assign the result to word W8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = 36 (hex)
Word W8 = ?

Function	**STL**	**Accumulator contents [bit]**	**Operand contents [bit]**
		31..15 .... 7 0	1 5......... 8 7........... 0
Load byte B6 into the word accumulator.	**L B6**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	00110110
Gate the contents of the word accumulator and byte B5 with OR.	**O B5**	0 ..... 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 0	00101010
Assign the gating result to word W8.	**= W8**	0 ..... 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0	0 0000000 00111110

7.7.22 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 TNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with OR NOT, and assign the result to output O2.
Initial state:
Input I4 = 0
Input I5 = 0
Output O2 = ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 0 x x x x x x x	0
Gate the content of logic accumulator and input I5 with OR NOT.	ON I5	x ..... x x x x x x x x 1 x x x x x x x	0
Assign the gating result to output O2.	= O2	x ..... x x x x x x x x 1 x x x x x x x	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 sizes of operand (B = 8 bit; W = 16 bit; D = K = 32 bit), 8, 16 or 32 bits will be 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 TNC saves the result of the operation in the word accumulator.

Example:

Gate the content of words W4 and W6 with OR NOT, and assign the result to word W8.
Initial state:
Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	1 5.......... 8 7............ 0
Load W6 into the word accumulator.	**L W6**	0 ..... 0 0 1 1 1 1 0 0 0 0 1 1 0 1 1 0	0 0111100 00110110
Gate the content of word accumulator and word W4 with OR NOT.	**ON W4**	1 ......1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1	0 0110110 10101010
Assign the gating result to word W8.	**= W8**	1 ......1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1	1 1111101 01110111

7.7.23 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 TNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with EXCLUSIVE OR, and assign the result to output O2.
Initial state:
Input I4 = 1
Input I5 = 1
Output O2 = ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I4	x ..... x x x x x x x x 1 x x x x x x x	1
Gate the content of logic accumulator and input I5 with EXCLUSIVE OR.	XO I5	x ..... x x x x x x x x 0 x x x x x x x	1
Assign the gating result to output O2.	= O2	x ..... x x x x x x x x 0 x x x x x x x	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 sizes of operand (B = 8 bit; W = 16 bit; D = K = 32 bit), 8, 16 or 32 bits will be 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 TNC 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	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	1 5......... 8 7........... 0
Load byte B6 into the word accumulator.	**L B6**	0 ..... 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	00110110
Gate the contents of the word accumulator and byte B5 with EXCLUSIVE OR.	**XO B5**	0 ..... 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0	00101010
Assign the gating result to word W8.	**= W8**	0 ..... 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0	00011100

7.7.24 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 TNC saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and marker M500 with EXCLUSIVE OR NOT, and assign the result to output O2.
Initial state:
Input I4 = 0
Marker M500 = 0
Output O2 = ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L M500	x ..... x x x x x x x x 0 x x x x x x x	0
Gate the content of logic accumulator and input I4 with EXCLUSIVE OR NOT.	XON I4	x ..... x x x x x x x x 1 x x x x x x x	0
Assign the gating result to output O2.	= O2	x ..... x x x x x x x x 1 x x x x x x x	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 sizes of operand (B = 8 bit; W = 16 bit; D = K = 32 bit), 8, 16 or 32 bits will be 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 TNC saves the result of the operation in the word accumulator.

Example:

Gate the content of words W4 and W6 with EXCLUSIVE OR NOT, and assign the result to word W8.
Initial state:
Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Function	**STL**	**Accumulator contents [bit]**	**Operand contents [bit]**
		31..15 .... 7 0	1 5......... 8 7........... 0
Load W6 into the word accumulator.	**L W6**	0 ..... 0 0 1 1 1 1 0 0 0 0 1 1 0 1 1 0	0 0111100 00110110
Gate the contents of word accumulator and word W4 with EXCLUSIVE OR NOT.	**XON W4**	1 ......1 1 1 1 0 1 0 1 0 1 1 0 0 0 1 1	0 0110110 10101010
Assign the gating result to word W8.	**= W8**	1 ......1 1 1 1 0 1 0 1 0 1 1 0 0 0 1 1	1 1110101 01100011

7.7.25 ADDITION (+)

Syntax: + (PLUS)

Operands: B, W, D, K

Action:

The TNC 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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	STL	Accumulator content (dec)	Operand content (dec)
		x xxx xxx xxx	
Load the constant into the word accumulator.	**L K10000**	10 000	
Add the content of the word accumulator and word W6.	**+ W6**	10 200	200
Assign the result to double word D8.	**= D8**	10 200	10 200

7.7.26 SUBTRACTION (–)

Syntax: – (MINUS)

Operands: B, W, D, K

Action:

The TNC 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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	STL	Accumulator content (dec)	Operand content (dec)
		x xxx xxx xxx	
Load the constant into the word accumulator.	**L K10000**	10 000	
Subtract word W6 from the content of the word accumulator.	**– W6**	9 800	200
Assign the result to double word D8.	**= D8**	9 800	9 800

7.7.27 MULTIPLICATION (X)

Syntax: x (MULTIPLY)

Operands: B, W, D, K

Action:

The TNC 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 TNC cannot execute the multiplication correctly, it then sets marker M4200; otherwise it resets it.

	Logic	Byte/Word	Double word	Constant
Processing time [s]	–	3.5 to 4.3	3.2 to 3.8	3.0 to 3.8
Number of bytes	–	14	10	14

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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	STL	Accumulator content (dec)	Operand content (dec)
		x x x x x x x x x x	
Load the constant into the word accumulator.	**L K100**	1 0 0	
Multiply the content of the word accumulator with word W6.	**x W6**	2 0 0 0	2 0
Assign the result to double word D8.	**= D8**	2 0 0 0	2 000

		Set	Reset
M4200	**Overflow during multiplication**	NC	PLC

7.7.28 DIVISION (/)

Syntax: / (DIVIDE)

Operands: B, W, D, K

Action:

The TNC 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 TNC 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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	STL	Accumulator content (dec)	Operand content (dec)
		x xxx xxx xxx	
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

7.7.29 REMAINDER (MOD)

Syntax: MOD (MODULO)

Operands: B, W, D, K

Action:

The TNC 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 TNC 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	= ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	STL	Accumulator content (dec)	Operand content (dec)
		x xxx xxx xxx	
Load W6 into the word accumulator.	**L W6**	5 0	5 0
Divide the content of the word accumulator by a constant, then save the integral REMAINDER in the word accumulator.	**MOD K15**	5	1 5
Assign the REMAINDER to double word D8.	**= D8**	5	5

		Set	Reset
M4202	**Incorrectly executed modulo**	NC	PLC

7.7.30 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.

7.7.31 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.

7.7.32 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 TNC 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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	STL	Accumulator contents (dec or [bit])	Operand content (dec or [bits])
		x x x x x x x x x x	
Load the constant into the word accumulator.	**L K16000**	1 6 0 0 0	
		31..15 .... 7 0	
Gate the content of the word accumulator with the operand content D8; if not equal, set the logic accumulator to 0.	== D8	x ..... x x x x x x x x x 0 x x x x x x x	1 5 0 0 0
Assign the result to marker M500.	= M500	x ..... x x x x x x x x x 0 x x x x x x x	0

7.7.33 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 TNC 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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	**STL**	**Accumulator contents (dec or [bit])**	**Operand content (dec or [bits])**
		x xxx xxx xxx	
Load the constant into the word accumulator.	**L K16000**	16 000	
		31..15 .... 7 0	
Check whether word accumulator < operand; if not, set logic accumulator to 0.	< D8	x ..... x x x x x x x x 0 x x x x x x x	15 000
Assign the result to marker M500.	= M500	x ..... x x x x x x x x 0 x x x x x x x	0

7.7.34 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 TNC 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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	**STL**	**Accumulator contents (dec or [bit])**	**Operand content (dec or [bits])**
		x xxx xxx xxx	
Load the constant into the word accumulator.	**L K16000**	16 000	
		31..15 .... 7 0	
Check whether word accumulator > operand; if so, set logic accumulator to 1.	> D8	x ..... x x x x x x x x 1 x x x x x x x	15 000
Assign the result to marker M500.	= M500	x ..... x x x x x x x x 1 x x x x x x x	1

7.7.35 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 TNC 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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	**STL**	**Accumulator contents (dec or [bit])**	**Operand content (dec or [bits])**
		x xxx xxx xxx	
Load the constant into the word accumulator.	**L K16000**	16 000	
		31..15 .... 7 0	
Check whether word accumulator <= operand; if not, set logic accumulator to 0.	<= D8	x ..... x x x x x x x x 0 x x x x x x x	15 000
Assign the result to marker M500.	= M500	x ..... x x x x x x x x 0 x x x x x x x	0

7.7.36 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 TNC 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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	**STL**	**Accumulator contents (dec or [bit])**	**Operand content (dec or [bits])**
		x xxx xxx xxx	
Load the constant into the word accumulator.	**L K16000**	16 000	
		31..15 .... 7 0	
Check whether word accumulator >= operand; if so, set logic accumulator to 1.	>= D8	x ..... x x x x x x x x 1 x x x x x x x	15 000
Assign the result to marker M500.	= M500	x ..... x x x x x x x x 1 x x x x x x x	1

7.7.37 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 TNC 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 = ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	**STL**	**Accumulator contents (dec or [bit])**	**Operand content (dec or [bits])**
		x xxx xxx xxx	
Load the constant into the word accumulator.	**L K16000**	16 000	
		31..15 .... 7 0	
Check whether word accumulator <> operand; if so, set logic accumulator to 1.	<> D8	x ..... x x x x x x x x 1 x x x x x x x	15 000
Assign the result to marker M500.	= M500	x ..... x x x x x x x x 1 x x x x x x x	1

7.7.38 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 TNC puts the content of the logic accumulator onto the program stack. When you address a word accumulator, the TNC saves the content of the word accumulator. With the closing-parenthesis command, the TNC 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 TNC assigns the result of the gating operation to the corresponding accumulator. Maximum nesting depth: 16 parentheses.

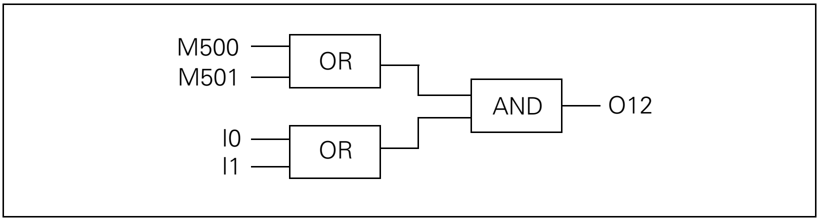
Please note:

The sequence of function is the same for word processing, however, the TNC writes the complete word accumulator onto the program stack.

Example:

Example for the commands AND [ ], AND NOT [ ], OR [ ], OR NOT [ ], EXCLUSIVE OR [ ], EXCLUSIVE OR NOT [ ]:

Use parentheses to develop a statement list in accordance with the following logic circuit diagram:



Initial state:

Marker	M500	= 0
Input	I0	= 0
Marker	M501	= 1
Input	I1	= 1
Output	O12	= ?

Function	STL	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load marker M500 into the logic accumulator.	L M500	x ..... x x x x x x x x **0** x x x x x x x	0
Gate logic accumulator with marker M501.	O M501	x ..... x x x x x x x x **1** x x x x x x x	1
Opening parenthesis: Buffer the accumulator content onto the program stack.	A[	x ..... x x x x x x x x 1 x x x x x x x	
Load the state of input I0 into the logic accumulator.	L I0	x ..... x x x x x x x x 0 x x x x x x x	0
Gate the logic accumulator with the state of input I1.	O I1	x ..... x x x x x x x x 1 x x x x x x x	1
Closing parenthesis: Gate the accumulator content with the program stack (A[, O[...).	]	x ..... x x x x x x x x 1 x x x x x x x	
Assign the result of the total operation to output O12.	= O12	x ..... x x x x x x x x 1 x x x x x x x	1

7.7.39 AND NOT [ ] (AN[ ])

Syntax: AN[ ] (AND NOT [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

7.7.40 OR [ ] (O[ ])

Syntax: O[ ] (OR [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

7.7.41 OR NOT [ ] (ON[ ])

Syntax: ON[ ] (OR NOT [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

7.7.42 EXCLUSIVE OR [ ] (XO[ ])

Syntax: XO[ ] (EXCL: OR [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

7.7.43 EXCLUSIVE OR NOT [ ] (XON[ ])

Syntax: XON[ ] (EXCL: OR NOT [ ])

Operands: None

Action:

See example A[ ] (AND [ ])

7.7.44 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 TNC saves the result in the accumulator again. Maximum nesting depth: 16 parentheses. If an error occurs during calculation, the TNC sets the marker M4201.

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

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	Accumulator content (dec)	Operand content (dec)
		x xxx xxx xxx	
Load the double word D12 into the word accumulator.	L D12	15 000	15 000
Opening parenthesis: Buffer the accumulator content onto the program stack.	+[	15 000	
Load the constant K 1000 into the word accumulator.	L K1000	1 000	
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 (+[, –[.....).	]	15 010	
Assign the result of the total operation to double word D100.	= D100	15 010	15 010

		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

7.7.45 SUBTRACTION [ ] (–[ ])

Syntax: –[ ] (MINUS –[ ])

Operands: None

Action:

See example for ADDITION [ ]

7.7.46 MULTIPLICATION [ ] (X[ ])

Syntax: x[ ] (MULTIPLY [ ])

Operands: None

Action:

See example for ADDITION [ ]

7.7.47 DIVISION [ ] (/[ ])

Syntax: /[ ] (DIVIDE [ ])

Operands: None

Action:

See example for ADDITION [ ]

7.7.48 REMAINDER [ ] (MOD[ ])

Syntax: MOD[ ] (MODULO [ ])

Operands: None

Action:

See example for ADDITION [ ]

7.7.49 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 TNC 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 TNC 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	= ?

To improve clarity, the contents of the accumulator and operand are shown in decimal notation. The ten-digit accumulator results from the greatest possible accumulator content (2 147 483 647).

Function	STL	Accumulator contents (dec or [bit])	Operand content (dec or [bits])
		x xxx xxx xxx	
Load the double word D12 into the word accumulator.	**L D12**	15 000	15000
Opening parenthesis: Buffer the accumulator content onto the program stack.	**== [**	15 000	
Load the constant into the word accumulator.	**L K1000**	1 000	
Multiply the content of the word accumulator with double word W36.	**x D36**	10 000	10
		31..15 .... 7 0	
Closing parenthesis: Gate the accumulator content with the program stack (==[, >=[...); if condition not fulfilled, set logic accumulator to 0.	**]**	x ..... x x x x x x x x 0 x x x x x x x	
Assign the result to output O15.	**= O15**	x ..... x x x x x x x x 0 x x x x x x x	0

7.7.50 LESS THAN [ ] (<[ ])

Syntax: <[ ] (LESS THAN [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

7.7.51 GREATER THAN [ ] (>[ ])

Syntax: >[ ] (GREATER THAN [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

7.7.52 LESS THAN OR EQUAL TO [ ] (<=[ ])

Syntax: <=[ ] (LESS EQUAL [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

7.7.53 GREATER THAN OR EQUAL TO [ ] (>=[ ])

Syntax: >=[ ] (GREATER EQUAL [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

7.7.54 NOT EQUAL [ ] (<>[ ])

Syntax: <>[ ] (NOT EQUAL [ ])

Operands: None

Action:

See example for EQUAL TO [ ]

7.7.55 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 TNC 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 = ?

The accumulator content is shown here in binary notation; the operand content in hexadecimal notation.

Function	STL	Accumulator contents [bit]	Operand content (hex)
		x xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx	
Load the double word D8 into the word accumulator.	**L D8**	0 0000000 00000000 00111110 10000000	0 0 00 3E 80
Shift the content of the word accumulator to the left by the number of bits that are specified in the operand.	**<<K+1**	0 0000000 00000000 01111101 00000000	
	<<K+1	0 0000000 00000000 11111010 00000000	
	<<K+1	0 0000000 00000001 11110100 00000000	
	<<K+1	0 0000000 00000011 11101000 00000000	
Assign the result to double word D12.	**= D12**	0 0000000 00000011 11101000 00000000	0 0 03 E8 00

Instead of using the << K+1 command four times, simply use the << K+4 command.

7.7.56 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 TNC shifts to the right out of the accumulator are then lost. The TNC 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 = ?

The accumulator content is shown here in binary notation; the operand content in hexadecimal notation.

Function	STL	Accumulator contents [bit]	Operand content (hex)
		x xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx	
Load the double word D8 into the word accumulator.	**L D8**	0 0000000 00000000 00111110 10000000	0 0 00 3E 80
Shift the content of the word accumulator to the right by the number of bits that are specified in the operand.	**>> K+1**	0 0000000 00000000 00011111 01000000	
	>> K+1	0 0000000 00000000 00001111 10100000	
	>> K+1	0 0000000 00000000 00000111 11010000	
	>> K+1	0 0000000 00000000 00000011 11101000	
Assign the result to double word D12.	**= D12**	0 0000000 00000000 00000011 11101000	0 0 00 03 E8

Instead of using the >> K+1 command four times, simply use the >> K+4 command.

7.7.57 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 TNC 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 = ?

Accumulator and operand contents are shown here in hexadecimal notation.

Function	STL	Accumulator content (hex)	Operand content (hex)
		x x x x x x x x	
Load the double word D8 into the word accumulator.	**L D8**	0 0 0 0 3 E 8 0	00 00 3E 80
Set the bit specified in the operand to 1.	**BS K+0**	0 0 0 0 3 E 8 1	
Assign the result to double word D12.	**= D12**	0 0 0 0 3 E 8 1	00 00 3E 81

7.7.58 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 TNC 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 = ?

Accumulator and operand contents are shown here in hexadecimal notation.

Function	STL	Accumulator content (hex)	Operand content (hex)
		x x x x x x x x	
Load the double word D8 into the word accumulator.	**L D8**	0 0 0 0 3 E 8 1	00 00 3E 81
Set the bit specified in the operand to 0.	**BC K+0**	0 0 0 0 3 E 8 0	
Assign the result to double word D12.	**= D12**	0 0 0 0 3 E 8 0	00 00 3E 80

7.7.59 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 TNC checks the state of a bit in the word accumulator and then sets the logic accumulator. If the interrogated bit = 1, the TNC 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 TNC 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 = ?

The word accumulator and operand contents are shown here in hexadecimal notation, the logic accumulator in binary notation.

Function	STL	Accumulator contents (hex or [bit])	Operand content (hex or [Bit])
		x x x x x x x x	
Load the double word D8 into the word accumulator.	**L D8**	0 0 0 0 3 E 8 1	00 00 3E 81
Check the state of the bit specified in the operand.	**BT K+0**	0 0 0 0 3 E 8 1	
		31..15 .... 7 0	
Assign the result to output O12.	**= O12**	x ..... x x x x x x x x 1 x x x x x x x	1

7.7.60 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. To do this, the TNC loads the addressed operand onto the data stack. Because the data stack has a width of 16 bits, you must write to it with a minimum width of one word. The TNC 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 TNC outputs an error message.

Memory assignment in the data stack [bit]
31..15 .... 7 0
x ..... x 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 TNC copies the addressed operand value into the current address of the data stack. During the word processing, the TNC 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 TNC 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.

7.7.61 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 TNC copies bit 7 of the data stack's current address into the addressed operand. If the stack is empty, the TNC 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 the word processing, the TNC copies with the PL command two words of the current data stack address into the addressed memory area. If the stack is empty, the TNC displays an error message.

Example:

See PSW command.

7.7.62 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 TNC copies the logic accumulator onto the data stack. Because the data stack has a width of 16 bits, you must write to it with a minimum width of one word. The TNC 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 TNC outputs an error message.

Memory assignment in the data stack [bit]
15 .... 7 0
x x x x x x x x L x x x x x x x

Example:

See PSW command.

7.7.63 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 TNC 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 TNC 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 calling the module: 1A 44 3E 18

Function	STL	Accumulator content (hex)	Data stack (hex)
		x x x x x x x x	
Buffer the word accumulator in the data stack.	**PSW**	1 A 4 4 3 E 1 8	1A 44 3E 18
Call subroutine 15.	**CM 15**		
Restore data stack into word accumulator.	**PLW**	1 A 4 4 3 E 1 8	1A 44 3E 18

7.7.64 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 TNC copies bit 7 of the data stack's current address into the logic accumulator. If the stack is empty, the TNC displays an error message.

Example:

See PSW command.

7.7.65 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 the word processing, the TNC copies with the PLW command two words of the current data stack address into the word accumulator. If the stack is empty, the TNC displays an error message.

Example:

See PSW command.

7.7.66 UNCONDITIONAL JUMP (JP)

Syntax: JP (JUMP)

Operands: Label (LBL)

Action:

After a JP command, the TNC jumps to the label that you have entered and resumes the program from there. JP interrupts a logic sequence.

Example:

See JPT command.

7.7.67 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 TNC resumes the program at the label that you have entered. If the logic accumulator = 0, the TNC 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	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	L I5	x ..... x x x x x x x x 1 x x x x x x x	1
If logic accumulator =1, jump to LBL 10.	**JPT 10**	x ..... x x x x x x x x 1 x x x x x x x	
Skip the function.	**L I3**		
Skip the function.	**O M500**		
Skip the function.	**= 020**		
Label	**LBL 10**		
Resume the program run.	**L M100 . . . .**	x ..... x x x x x x x x 0 x x x x x x x	0

7.7.68 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 TNC resumes the program at the label that you have entered. If the logic accumulator = 1, the TNC does not jump. JPF interrupts a logic sequence.

Example:

See JPT command.

7.7.69 CALL MODULE (CM)

Syntax: CM (CALL MODULE)

Operands: Label (LBL)

Action:

After a CM command, the TNC 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.

7.7.70 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 TNC calls the module that begins at the label that you have entered. If the logic accumulator = 0, the TNC does not call the module. CMT interrupts a logic sequence.

Example:

See command CMF.

7.7.71 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 TNC calls the module that begins at the label that you have entered. If the logic accumulator = 1, the TNC 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	Accumulator contents [bit]	Operand contents [bit]
		31..15 .... 7 0	
Load the operand content into the logic accumulator.	**L I5**	x ..... x x x x x x x x 0 x x x x x x x	0
If logic accumulator =0, jump to LBL 10.	**CMF 10**	x ..... x x x x x x x x 0 x x x x x x x	
Resume main program after module execution.	**L M100**	x ..... x x x x x x x x 1 x x x x x x x	1
	⋮		
End of the main program.	**EM**		
Label: Beginning of module.	**LBL 10**		
Statement in the module.	**L I3**	x ..... x x x x x x x x 0 x x x x x x x	0
Statement in the module.	**O M500**	x ..... x x x x x x x x 1 x x x x x x x	1
Statement in the module.	**= O20**	x ..... x x x x x x x x 0 x x x x x x x	0
End of module, resume the main program with the command L M100.	**EM**		

7.7.72 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 TNC then resumes the program with the instruction that follows the module call. The TNC interprets the command EM as program end. The TNC can reach the subsequent program instructions only through a jump instruction.

7.7.73 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.

7.7.74 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.

7.7.75 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 TNC evaluates only the first 16 characters.

For importing global labels, see EXTERN instruction.

7.8 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. However, the TNC uses only the lower 16 bits for indexed addressing.

You can use the X register anywhere in the program. The TNC does not check whether the current content is valid. Exception: During indexed write accessing the TNC checks whether the amount of available address space is exceeded.

Example: = B100[X]

If the permitted addressable storage is exceeded, the TNC issues a blinking error message: `PLC: index range incorrect.` Acknowledge the error message by pressing the END key. After restarting the TNC 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 TNC 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 operand	Content of index register =
BCX operand	Content of index register =
BSX operand	Content of index register =
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 of the n-th string	Substring from X-th character

The types “S”, K, and K$ cannot be indexed.

Note

If you address S#Dn[X] or S#En[X], the TNC 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)	

7.9 Commands for String Processing

String processing enables you to use the PLC program to generate and manipulate any texts. Use Module 9082 to display these texts in the PLC window of the screen, and delete them with Module 9080. The TNC features one string accumulator and eight string memories, 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 1 O N	

String accumulator and string memory are volatile, which means that they are erased by the TNC when power is switched off. The new operand “S” has been introduced 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 – S15).
- Address part of a string: Use the address Sn^X (see INDEX Register). The TNC 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 37 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 4095). Enter the string #Exx or #Dxx in the argument <arg> of the string command. The TNC then saves a 5-byte-long string <SUB> E0xx or <SUB> D0xx (<SUB> = ASCII <SUB>) in the accumulator. Instead of this string, the TNC 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. The TNC then 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 TNC does not check the remaining characters. During a comparison, the TNC 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 TNC compares the position in the error-message file or dialog file (0 to 4095), but not the actual text as with an immediate string.

The processing times depend on the length of the strings. The processing times given in Chapter 7.7.2 are maximum values. For immediate strings, you must add the length “n” of the string to the length of the command; if this length is an odd number, you must add the next larger even-numbered length.

7.9.1 LOAD STRING (L)

Syntax: L (LOAD)

Operands: S <arg>

Action:

Load the string accumulator. The string that the TNC is to load is selected through the argument <arg> after the operand designation. See also "Operand declaration."

Example:

See command OVWR.

7.9.2 ADD STRING (+)

Syntax: + (PLUS)

Operands: S <arg>

Action:

Attach another string to a string in the string accumulator. The string that the TNC 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.

7.9.3 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 TNC is to copy the string is selected through the argument <arg> after the operand designation. Permissible arguments: 0 to 7 (String memory S0 to S15). See also "Operand declaration."

Example:

See command OVWR.

7.9.4 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 TNC 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 TNC is to copy the string is selected through the argument <arg> after the operand designation. Permissible arguments: 0 to 7: (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		= **HYDRAULICS**
String memory	S0	= **OIL**
String memory	S1	= **COOLANT MISSING**

String memory (characters)	
	1 ...128
S0	O I L
S1	C O O L A N T M I S S I N G
...	

Function	STL	String accumulator (characters)
		1 ...128
Load the string into the string accumulator.	**L S "HYDRAULICS"**	H Y D R A U L I C S
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
...	

7.9.5 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 TNC sets the logic accumulator to 1. If they are not equal, the TNC sets the logic accumulator is set to 0.

Example:

See command <>.

7.9.6 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 TNC 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 <>.

7.9.7 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 TNC 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 <>.

7.9.8 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 TNC 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 <>.

7.9.9 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 TNC 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 <>.

7.9.10 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 TNC 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. [bit]**
		1 ..128
Load the string into the string accumulator.	**L S "SPINDLE 1"**	S P I N D L E 1
Gate the content of string memory S0 with content of string accumulator (=, <, >, >=, ...)	**<> S0**	S P I N D L E 2
		31.. 15 .... 7 0
If the condition is fulfilled, set logic accumulator to 1 and call the module.	**CMT 50**	x ..... x x x x x x x x 1 x x x x x x x

7.9.11 Modules for String Processing

Module 9070 Copy a number from a string

The TNC searches a selectable string memory (S0 to S15) for a numerical value. When the numerical value is first found, the TNC copies it as a string into another selectable string memory. The TNC 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 TNC recognizes unsigned and signed numbers, with and without decimal places. Both the period and comma are permitted as decimal point. The TNC returns the position (in characters) of the first character after the found number in the string memory to be searched.

Call:

PS	K/B/W/D	<Address of the string memory to be searched>
PS	K/B/W/D	<Address of the string memory for the found number>
CM	9070	
PL	B/W/D	<Offset end of numerical string in the searched string memory>

Error recognition:

Marker	Value	Meaning
M4203	0	Number 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 [bit]
	1 .... 10...128	
S0	X - P O S . : 1 2 3	
S1	1 2 3	10
...		

Module 9071 Find the string length

The TNC 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 recognition:

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 (0..7), string memory was searched but no string end was found

7.10 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 processor utilization, the TNC provides for a submit program a certain percentage of its computing power—at least 5%. 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 are 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 TNC 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 TNC 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 TNC 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. The processing times given in Chapter 7.7.2 are maximum values.

7.10.1 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 TNC writes the assigned number in the word accumulator. If programs are already entered in the submit queue, the TNC does not run the addressed program until the programs before it are finished. You can make an entry in the queue only in a PLC program. It is not possible to make a SUBM command in a submit program.

If there is no room in the queue, or if you program the SUBM command in a submit program (nesting), the TNC assigns the value "0" to the word accumulator.

Example:

See command CAN.

7.10.2 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 TNC 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.

7.10.3 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 TNC 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 accumulator.	**L I10**
If logic accumulator =0, jump to LBL 100.	**JPF 100**
Interrogate the status of the submit program and load it into the word accumulator.	**RPLY B128**
If the word accumulator is not equal to 0, i.e., the submit program has already been transferred to the NC for processing, set the logic accumulator to 1.	**<> K+0**
If logic accumulator =1, jump to LBL 100.	**JPT 100**
Call submit Program 300.	**SUBM 300**
Save the identifier of the submit program in byte 128.	**= B128**
Label	**LBL 100**
Load the state of input I11 into the logic accumulator.	**L I11**
If logic accumulator =0, jump to LBL 110 (skip the program cancellation).	**JPF 110**
Cancel the submit program.	**CAN B128**
Label	**LBL 110**
	⋮
End of 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.

7.11 Cooperative Multitasking

As of NC software: 280 472-01

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 TNC 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 stage they are at.

7.11.1 STARTING A PARALLEL PROCESS (SPAWN)

Syntax: SPAWN <label>

Operands: D

Action:

In the specified double word, the TNC returns the identifier, See "Submit Programs" on page 7 – 159. If no process could be started, the TNC returns the value –1. You can call the spawn only in a submit job or in another spawn process (maximum of eight parallel processes are permitted). If a process ends with EM, the TNC removes it from the memory to provide space.

7.11.2 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 TNC 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 Parameter)" on page 7 – 22 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 TNC will check for = 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 TNC 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 TNC 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 TNC then deletes all entries in the list.

The TNC 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 TNC issues a blinking error message if:

- A non-existing 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 TNC displays all parallel processes, including the process for the submit queue. In a time interval of 1 second, the TNC shows:

- The current process status (runable, running, waiting for event, waiting for a time period).
- How often the process changed contexts during the last time interval.
- The CPU time consumed by the process. The TNC also shows the distribution of CPU time in a bar chart.

Module 9260 Receiving 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 TNC returns all set events without deleting them. Otherwise, in a call with a waiting period, the TNC returns all the requested events and deletes them. For a call without a waiting period, the TNC returns and deletes the events only if the condition is met.

If the events are OR-gated, the TNC 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 TNC cancels access to interfaces and the network!
- Bit 17: Reserved, do not use
- Bit 18: Reserved, do not use
- Bit 19: QUIT, acknowledgment of a request. Use this bit only in the immediate context of a request.
- Bit 20 to bit 31: Reserved, do not use

Call:

```
PS     B/W/D/K   <Wait>
                 0 = do not wait
                 –1 = wait
PS     B/W/D/K   <AND/OR>
                 0 = OR-gated, otherwise AND-gated
PS     B/W/D/K   <Event mask>
                 0 = available events
CM     9260
PS     B/W/D/K   <Events>
                 Read events
```

Error recognition:

Marker	Value	Meaning
M4203	0	Event has been read
	1	Error code in W1022
W1022	2	Incorrect transfer value for <Wait> parameter
	20	Module was not called in a spawn job or submit job

Module 9261 Sending events

With this module you can send events to a spawn or submit job and then interrogate them with Module 9260. You can call the module in the cyclic program section, in submit jobs and in spawn processes. The TNC 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 TNC always assigns the events that you send to the submit queue to the job that is running at the time of arrival. If they are not read by this job, they remain for the next one. If you wish the receiver process to start immediately, after Module 9261 you must also call Module 9262 to enable a change of context.

Event bits 16 to 31 are reserved for the operating system (see Module 9260).

Call:

PS	D/K	<Identifier> Identifier from the spawn command of the receiver K$80000000 = submit queue
PS	B/W/D/K	<Events> Events to be triggered, bit encoded
CM	9261	

Error recognition:

Marker	Value	Meaning
M4203	0	Event has been sent
	1	Error code in W1022
W1022	30	Incorrect identifier

Module 9262 Context change between spawn processes

You can call Module 9262 only in a submit job or spawn job. The module switches the context to another PLC process or submit queue if such a process exists and is not waiting for an event or for the expiration of a dwell time. Markers 4200 to 4202 and 4204 have undefined changes after the module call.

Call:
CM 9262

Error recognition:

Marker	Value	Meaning
M4203	0	Context was changed
	1	Error code in W1022
W1022	30	Module was not called in a spawn 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 TNC changes the context to one of these processes. Markers 4200 to 4202 and 4204 have undefined changes after the module call. The waiting period is interpreted as an unsigned number, so that negative values result in very long waiting periods.

Call:
PS B/W/D/K <Waiting period in ms>
CM 9263

Error recognition:

Marker	Value	Meaning
M4203	0	Process waited
	1	Error code in W1022
W1022	30	Module was not called in a spawn job or submit job

7.12 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 TNC transfers the value of the constant defined by X in the constants field <Name>.
- L KF <Name> [X], with X = –1:
  The TNC transfers the length of the constants field <Name>.
- L KF <Name>: The TNC 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 TNC 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>.

7.13 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.

7.13.1 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**

7.13.2 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 TNC 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**

7.13.3 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 TNC 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 TNC generates two internal labels. The condition can also be produced in a manner 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**

7.13.4 CASE BRANCH

Indexed module call (CASE)

Syntax: CASE (CASE OF)

Operands: B/W

Action:

Select a certain subprogram from a list of module calls (CM). These CM commands must immediately follow the CASE statement 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 indexed module call (ENDC)

Syntax: ENDC (ENDCASE)

Operands: None

Action:

Use the ENDC command in connection with the CASE command. It must directly follow 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**

7.14 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 very beginning of your PLC program – i.e., before the first PLC command (see PLC programming example). 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. Each file can contain up to 1000 labels. You can link up to 60 files to one program. Each file can generate up to 64 KB of code. The permissible total length of the code is limited to 128 KB. 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. In the EPROM you must enter the option /M behind the main program in the linker file for binary output.

7.14.1 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 SPINDEL.PLC**
Link the file for tool change.	**USES TCHANGE.PLC**
Program code	⋮

Function	STL
File for spindle control	**SPINDEL.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	⋮

7.14.2 GLOBAL INSTRUCTION (GLOBAL)

Syntax: GLOBAL <label, declaration beyond the file boundary>

Operands: None

Action:

Up to 1000 local labels are permitted in every 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. Altogether, 1000 labels may be as defined as global in all modules.

7.14.3 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. Every external label reduces the number of local labels that remain available.

7.15 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 page 5 – 3)

If the TNC runs a module unsuccessfully, it sets marker 4203. You then can evaluate this marker to display an error message.

7.15.1 Markers, Bytes, Words, and Double Words

Module 9000/9001Copy 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 TNC 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 TNC overwrites the overlapping part of the source block before the copying process.

Call:
PS B/W/D/K <Number of the 1st marker in source block>
PS B/W/D/K <Number of the 1st marker in target block>
PS B/W/D/K <Length of block in markers>
CM 9000

PS B/W/D/K <Number of the 1st word in source block>
PS B/W/D/K <Number of the 1st word in target block>
PS B/W/D/K <Length of block in markers>
CM 9001

Error recognition:

Marker	Value	Meaning
M4203	0	Markers, bytes, words, or double words were copied
	1	Error code in W1022
W1022	1	Operand address invalid
	2	Address too high or block too long
	4	Programmed source or destination block too long

Module 9010/9011/9012 Read in the word range

From the specified location in the word memory the TNC reads a byte, word or double word and returns it as an output quantity to the stack. Indexed reading is possible by specifying a variable as designation of the memory location.

Call:

```
PS    B/W/D/K   <Address of the byte to be read>
CM    9010      ; READ BYTE
PL    B         <Target address for byte that was read>

PS    B/W/D/K   <Address of the word to be read>
CM    9011      ; READ WORD
PL    B         <Target address for word that was read>

PS    B/W/D/K   <Address of the double word to be read>
CM    9012      ;READ DOUBLE WORD
PL    B         <Target address for double word that was read>
```

Error recognition:

Marker	Value	Meaning
M4203	0	Byte was read
	1	Error code in W1022
W1022	3	Invalid address was programmed
	5	Module 9011: Specified address is not a word address Module 9012: Specified address is not a double word address

Example of Module 9010

Initial state:
Byte B10 = 35 (address)
Byte B35 = 80 (byte to be read)
Byte B100 = ?

Function	STL	Accumulator content (dec)	Data stack (dec)
		x x x x x x x x	
Save the address (B10) of the byte to be read from the word accumulator to the data stack.	**PS B10**	3 5	35
Read byte B35 and save to the data stack.	**CM 9010**		80
Save data stack in byte B100.	**PL B100**	8 0	80

Module 9020/9021/9022 Write in the word range

The TNC writes the given byte, word or double word to the defined location in the word memory. Indexed writing is possible by specifying a variable as designation of the memory location.

Call:

```
PS    B/W/D/K   <Address of the byte to be written>
PS    B/W/D/K   <Byte to be written>
CM    9020      ; WRITE BYTE TO ADDRESS

PS    B/W/D/K   <Address of the word to be written>
PS    B/W/D/K   <Word to be written>
CM    9021      ; WRITE WORD TO ADDRESS

PS    B/W/D/K   <Address of the double word to be written>
PS    B/W/D/K   <Double word to be written>
CM    9022      ; WRITE DOUBLE WORD TO ADDRESS
```

Error recognition:

Marker	Value	Meaning
M4203	0	Byte was written
	1	Error code in W1022
W1022	3	Invalid address was programmed
	5	Module 9021: Specified address is not a word address Module 9022: Specified address is not a double word address

Example of Module 9020

Initial state:
Byte B10 = 35 (address)
Byte B100 = 120 (byte to be written)
Byte B35 = ?

Function	STL	Accumulator content (dec)	Data stack (dec)
		x x x x x x x x	
Save the address (B10) of the byte to be written from the word accumulator to the data stack.	**PS B10**	3 5	35
Save byte B100 from the word accumulator in the data stack.	**PS B100**	1 2 0	120
Write data stack in byte B35.	**CM 9020**	1 2 0	

7.15.2 Number Conversion

Module 9050 Conversion of binary numbers → ASCII

Module 9050 converts a binary numerical value consisting of a mantissa and exponent to the base of 10 into a decimal number and saves it as a string in the specified addressed. The exponent refers to the least significant place of the number. The TNC detects a negative number when the mantissa corresponds to a negative number in the notation as a two's complement. The TNC 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 TNC 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 the base of 10 of the value>
PS	B/W/D/K	<String address in which the TNC saves the ASCII-coded decimal number>
CM	9050	

Error recognition:

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 TNC converts the absolute value of the number without putting a sign before the string. For algebraically signed notation, the TNC 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 TNC 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 TNC saves the ASCII-coded
                decimal number>
CM    9051
```

Error recognition:

Marker	Value	Meaning
M4203	0	Number was converted
	1	Error code in W1022
W1022	2	Invalid string address, invalid display mode or invalid number of places before or after the decimal point

Module 9052 Conversion 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 S0 to S15. If the number has no algebraic sign, the TNC 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 TNC 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 recognition:

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 TNC 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 2 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 TNC saves the hexadecimal numbers>
PS	B/W/D/K	<Number of data bytes>
CM	9053	

Error recognition:

Marker	Value	Meaning
M4203	0	Number was converted
	1	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 TNC 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 TNC saves the binary values>
CM	9054	

Error recognition:

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. [bit]
		1 ..128
Push string address S0 onto the data stack	**PS K+0**	6 3
		7 0
Push word address B0 onto the data stack	**PS B0**	x x x x x x x x
Conversion of the two ASCII characters 6 and 3 into the binary number 99	**CM 9054**	0 1 1 0 0 0 1 1 6 3

8 Data Interfaces

8 Data Interfaces

8.1 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

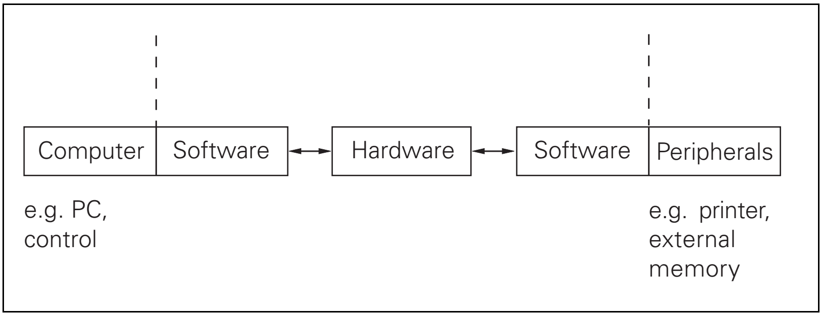
A data interface makes it possible for the CPU and its peripheral devices to communicate.

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
- Connection layout
- Electrical characteristics

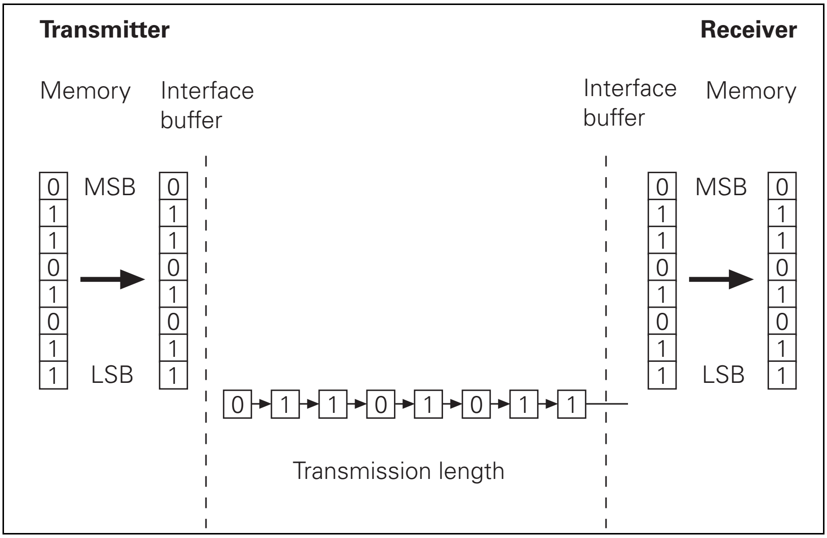
The “software” is the operating software, which includes, for example, the drivers for the output modules.

8.1.1 Principles of Data Transfer

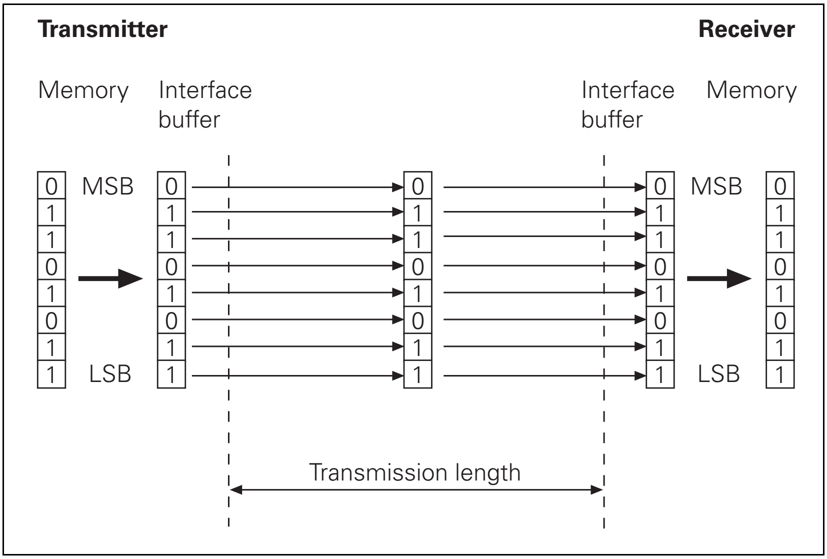
The term "data" is used to describe all of the information that the computer is capable of collecting and processing.

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 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:

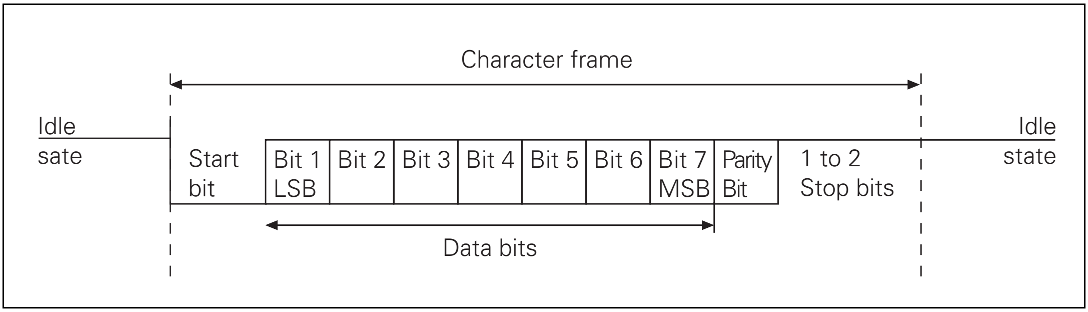
1 1 1 1 0 1 0 = 122 dec = 7A hex

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

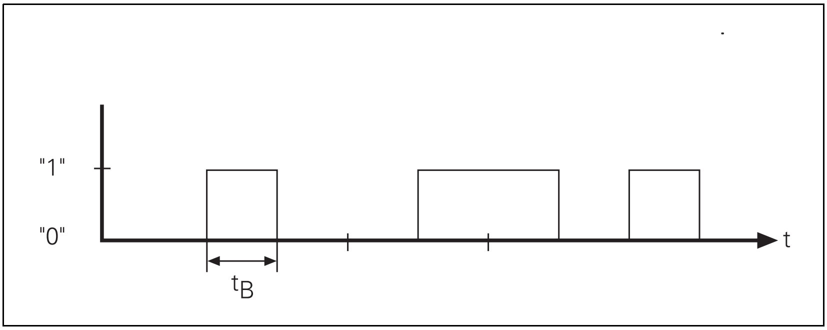
Data transfer rate

The data transfer rate is given in bits per second.

Common transfer rates are:

110, 150, 300, 600, 1200, 2400, 4800, 9600, 19 200, 38 400, 57 600 and 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. The number of characters transmitted per second can be calculated from the transfer rate and the transmission format:

$$t_B = \frac{1}{19200 \text{ (Bit/s)}} = 52.083\ \mu s$$

$$\text{Characters per second} = \frac{\text{Baud-Rate (Bit/s)}}{\text{Number of bits per character}}$$

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.

$$\text{Characters per second} = \frac{300 \text{ (Bit/s)}}{1 + 7 + 2} = 30$$

8.1.2 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 (switch 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

8.2 TNC Data Interfaces

8.2.1 General Information

The TNC 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

8.2.2 RS-232-C/V.24 Interface

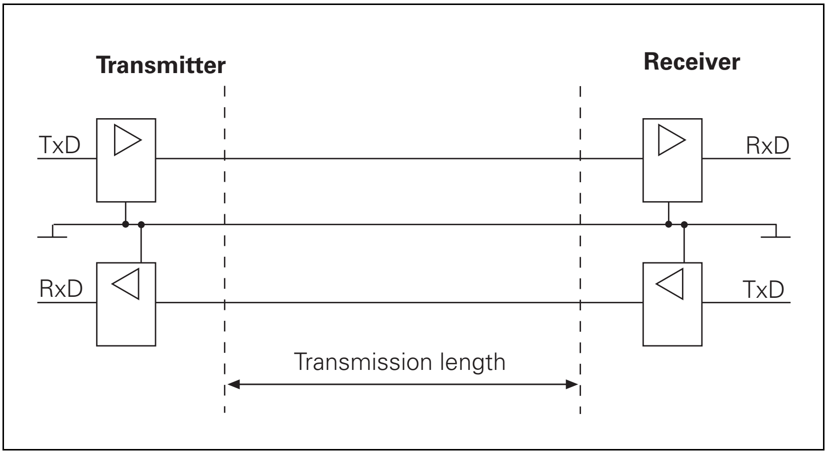
RS-232-C/V.24 is the designation for a serial interface. Data transfer is executed asynchronously, with a start bit before each character and one or two stop bits after each character.

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.

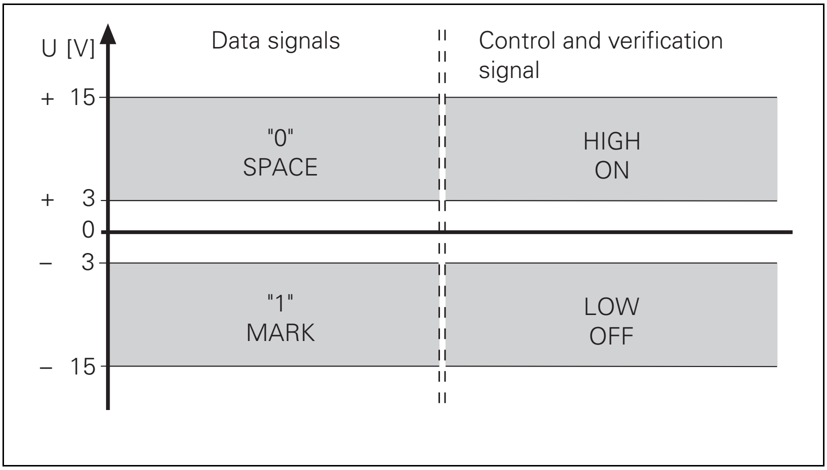
Physical connections:



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 (pin 8) is not used by the TNC. The TNC delivers no signal from this pin.
  - DTR (Data Terminal Ready):
    TNC 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. TNC 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-Volt lines for all signals

Pin layouts The pin layout of the TNC logic unit is different from that of its adapter block.

8.2.3 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 TNC can transmit data at up to 115 200 bps.

Transmission distance: over 1 kilometer

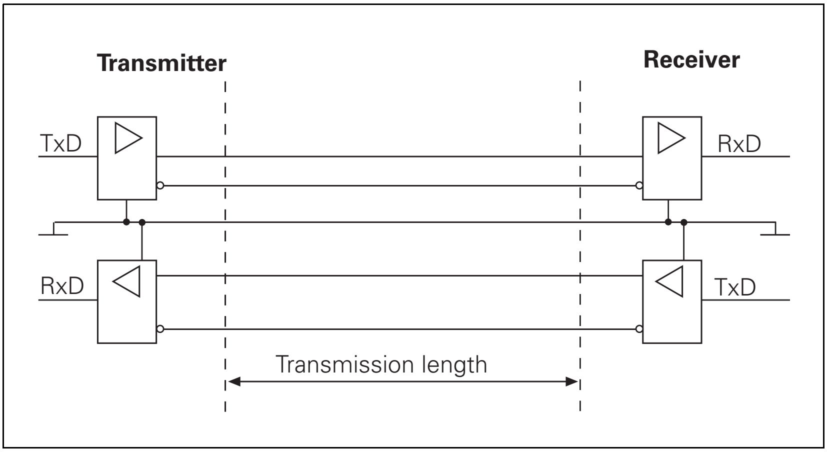
Hardware

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

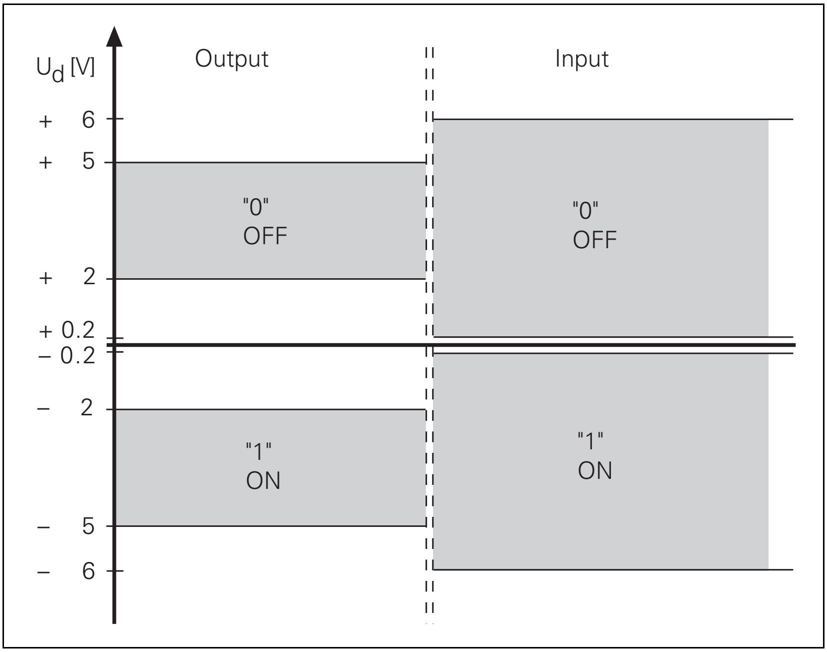
The signals are both transmitted and received as differential voltage.

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, RTS	CTS, CTS
	DSR, DSR	DTR, 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 logic unit and adapter block have the same pin layout.

8.3 Configuration of Interfaces

8.3.1 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 TNC 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.

8.3.2 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.

▶ Press the RS-232-C RS 422 SETUP soft key.

▶ 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 as of 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 TNCs

▶ Set both TNCs 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).

8.3.3 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)

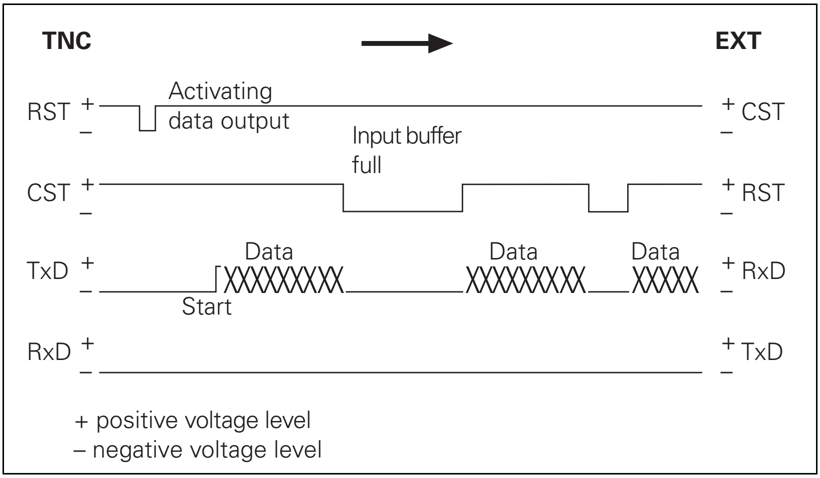
With bit 1 you can ensure that the BCC is not interpreted as a control character.

On the TNC, 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 therefore cannot be interpreted as a control character.

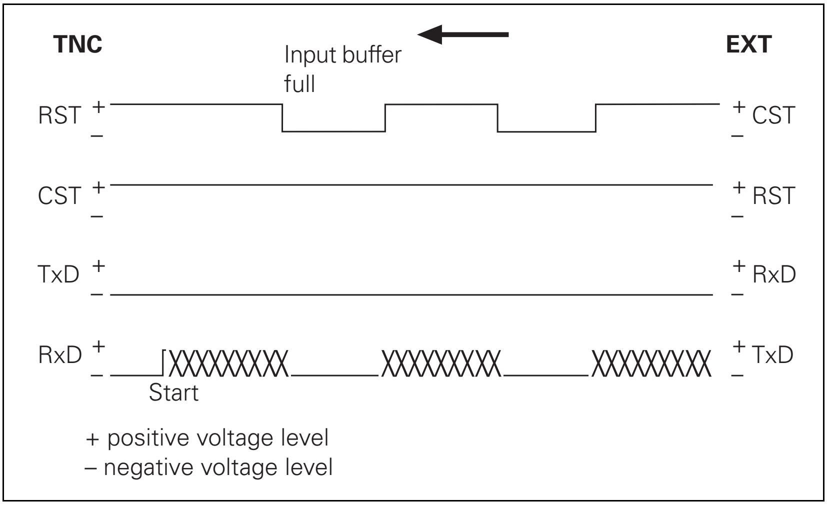
Hardware handshaking

With bit 2 you can determine whether the TNC stops transfer from an external device by sending an RTS signal.

- Data output from TNC to EXT
  When the receiving buffer is full, the external device resets the RTS signal. The TNC detects that the peripheral unit receiving buffer is full at its CTS input:



- Data input from EXT to TNC
  When the receiving buffer is full, the TNC 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 TNC and peripheral device:

- DTR: Interrogated by peripheral; it is logical one if TNC is ready for service.
- DSR: Interrogated by TNC.
  - 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 TNC 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 TNC reacts both to hardware and software handshakes, regardless of the setting in MP5020.x.

If no transmission stop is set in MP5020.x, the TNC stops the peripheral unit with the software handshake.

If a transmission stop by RTS and by DC3 is active simultaneously, the TNC 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 TNC:

- 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.
- Press the RS-232-C RS422 SETUP soft key.
- Choose the EXT1 operating mode.
- Set the baud rate for EXT1 to 9600 bps.

8.4 Data Transmission Protocols

8.4.1 Selection of Transmission Protocols

The operating modes are assigned the following transmission protocols:

Operating modes	Transmission protocol
FE1 and FE2	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 TNC. The protocol runs in the background of the TNC.

The following applies to all data transmission protocols except LSV2:

If an incoming file is already stored in the TNC, 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 TNC displays the error message **`Protected file!`**:

- Press the ADDITIONAL FUNCTION UNPROTECT soft key to cancel write-protection and continue the transmission.

If a file has been read out and the data transfer menu has been terminated with the END key, the TNC outputs the characters <EXT> and <EOT>.

If a transmission is terminated with the END key, the error message "Program incomplete" is issued.

8.4.2 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 TNC 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 TNC transmits the character <DC3> to the transmitter in order to terminate transmission.

Example: Protocol for conversational NC program

<NUL><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

TNC to peripheral	Peripheral to TNC
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 TNC outputs all of the program lines in order.

The peripheral unit can:

- Stop transmission with <DC3>
- Resume transmission with <DC1>

TNC to peripheral	Peripheral to TNC
<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 TNC.

The TNC can:

- Stop transmission with <DC3>
- Resume transmission with <DC1>

TNC to peripheral	Peripheral to TNC
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 TNC are not identical, the TNC 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 TNC remains static without an error message:

- In this case, terminate transfer with the END key.

8.4.3 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
- FE2 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).

TNC to peripheral	**Peripheral to TNC**
<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 TNC

FE1 mode is set.

If an error occurs at a peripheral device, the following block must be sent to the TNC:

<SOH><Error text><ETB>BCC

Peripheral to TNC	TNC to peripheral
<SOH>“Error”<ETB>BCC	<ACK><EOT>

The received error message is displayed on the TNC. To continue

- Press the CE key

Request external directory

FE1 mode is set.

This protocol is not available in FE2 and 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 TNC expects the following input to this request:

xxxxxx<Code letters><Sectors><Name><P[1)]><CR><LF>
[1)] P = Protected (optional)

The first four lines 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 TNC 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 TNC sends an <EOT>.

Output selected file

TNC to peripheral	Peripheral to TNC
<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 TNC sends a header with the file name.

TNC to peripheral	Peripheral to TNC
<SOH><K>Name<E><ETB>BCC <DC1>	<ACK> <STX>“1st line”<ETB>BCC<DC1>
<ACK>	...
...	<STX>“last line”<ETB>BCC <DC1>
<ACK>	<ETX>
<EOT>	

8.4.4 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 TNC error messages and keystrokes for service purposes. The last 1000 events are stored in the TNC.

HEIDENHAIN offers two LSV2 software packages:

- **TNCremo** — Software for TNC remote control. Can be run on an AT compatible PC with MS-DOS. All the above functions are available.
- **LSV2 TOOL BOX** — Software tools in C programming language for creating the transfer telegrams. The tool box comprises:
  - A library
  - Executable files for telegrams
  - Source codes
  - INCLUDE files for LSV2
  - MAKE files

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.

8.5 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.

8.6 Data Transfer by PLC

8.6.1 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 logic units 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 FE2, 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: 19 200 bps 9: 38 400 bps 10: 57 600 bps 11: 115 200 bps

8.6.2 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 recognition:

Marker	Value	Meaning
M4203	0	Interface was assigned
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect transfer parameter
	13	No connection
	14	Interface 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 recognition:

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 recognition:

Marker	Value	Meaning
M4203	0	Status read
	1	Error code in W1022
W1022	1	Incorrect interface

Module 9103 Transmit string through data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9103 transmits a string from one of the 16 string memories through one of the two interfaces. Links to the PLC error file and PLC dialog file are deleted. See "Error Messages" on page 3– 3.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	K/B/W/D	<Number of source string in the string buffer>
CM	9103	

Error recognition:

Marker	Value	Meaning
M4203	0	String 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 one of the 16 string memories 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 (0 to 15)>
CM	9104	

Error recognition:

Marker	Value	Meaning
M4203	0	String was received
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect string number
	12	String too long
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9105 Transmit binary data through data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9105 transmits a block of binary values from the word memory of the PLC to one of the two interfaces. The transfer is in the form of ASCII-coded hexadecimal values. Every byte in the source block makes two ASCII characters at the interface.

Can only be called in a submit job or spawn job!

Call:

```
PS    B/W/D/K   <Interface>
                0: RS232
                1: RS422
PS    K/B/W/D   <Number of the first byte in the binary block (0 to 1023)>
PS    K/B/W/D   <Length of the binary block (0 to 63)>
CM    9105
```

Error recognition:

Marker	Value	Meaning
M4203	0	Data was transmitted
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect byte number or block too long
	4	Block outside value range
	13	Interface not ready or no connection
	14	Interface not assigned
	15	Transmit buffer not empty
	20	Module was not called in a 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 (0 to 1023)>
CM	9106	
PL	B/W/D	<Length of binary block in bytes> –1: Incorrect module call

Error recognition:

Marker	Value	Meaning
M4203	0	Data was received
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect byte number or block too long
	4	Block outside value range
	11	Odd number of characters or illegal character
	12	String too long
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9107 Read from receiving buffer

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9107 reads two ASCII characters from the 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 recognition:

Marker	Value	Meaning
M4203	0	Receiving buffer was read
	1	Error code in W1022
W1022	1	Incorrect interface or incorrect byte number
	11	Illegal character
	12	String too long or offset too large
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9110 Transmit a message via LSV2

Module 9110 transmits a message (binary data or string) to a host computer connected by LSV2 protocol.

The message is transmitted to the host by the LSV2 command "M_PC<msg.l>".

Call:

PS	B/W/D/K	<Data type> 0: Binary data double word 1: String
PS	B/W/D/K	<Source address> With binary: Number of the double word (0 to 1020) With string: Number of the string
CM	9110	
PL	B/W/D	<Error code> 0: Message is being transmitted 1: No connection to host 2: Transmit buffer full 3: Incorrect data type (not 0 or 1) 4: Incorrect source address

Error recognition:

Marker	Value	Meaning
M4203	0	Message was transmitted
	1	Error code in W1022
W1022	2	Incorrect data type
	4	No double word address, or incorrect string number
	11	String too long
	13	No connection
	15	Transmit buffer not empty
	16	Receiving buffer empty

Module 9111 Receive a message via LSV2

Module 9111 reads a message (double word or string) that has been received from a host computer connected by LSV2 protocol.

The message must be transmitted from the host by the LSV2 command “M_PC<msg.l>”.

Call:

PS	B/W/D/K	<Data type> 0: Binary data double word 1: String
PS	B/W/D/K	<Target address> With binary: Number of the double word (0 to 1020) With string: Number of the string
CM	9111	
PL	B/W/D	<Error code> 0: Message was read 1: No connection to host 2: No message of this type in receiving buffer 3: Incorrect data type (not 0 or 1) 4: Incorrect target address

Error recognition:

Marker	Value	Meaning
M4203	0	Message 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                <ASCII code [0 to 255]>
CM      9112
```

Error recognition:

Marker	Value	Meaning
M4203	0	Character was transmitted
	1	Error code in W1022
W1022	1	Incorrect interface
	13	Interface not ready or no connection
	14	Interface not assigned
	15	Transmit buffer not empty
	20	Module was not called in a spawn job or submit job

Module 9113 Receive ASCII characters via data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9113 reads a single ASCII character from the receiving buffer of a serial interface and resets the receiving buffer.

If there is more than one character in the receiving buffer, the first is 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 recognition:

Marker	Value	Meaning
M4203	0	Character was received
	1	Error code in W1022
W1022	1	Incorrect interface
	12	String too long
	13	Interface not ready or no connection
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job
	37	Receiver queue full

8.7 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 TNC. They need not be programmed.

8.8 Ethernet Interface (Option)

HEIDENHAIN offers an Ethernet interface as an option on the TNC. Please contact HEIDENHAIN for further information.

With an Ethernet interface you can connect your TNC to your facility's local area network and all its PCs and workstations.

The data transfer rate is dependent on the amount of traffic at the time on the net.

Typical values:

- NC program up to 300 Kbps
- ASCII program up to 1 Mbps

8.8.1 Hardware

The integrated Ethernet expansion card provides you with a 10BaseT (twisted pair) port.

The port is metallically isolated from the control electronics.

For information on the pin layout: See “Mounting and Electrical Installation” on page 3– 5.

X25: Ethernet interface RJ45 port (10BaseT)

Maximum cable length:
Unshielded 100 m
Shielded 400 m

Network topology: Star configuration

This means a central node establishes the connection to the other participants.

8.8.2 Software

The TNC requires an NFS server (Network File System) as the remote station. The NFS server must work according to the TCP/IP protocol principle. The remote station must be an NFS server.

OSI 7-layer model		TNC
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.

TNC 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 TNC for networking	
	ADDRESS	Internet address of the TNC: Enter as four decimal numbers separated by points (dotted-decimal notation). Your network supervisor can give you an internet address.
	MASK	Subnet mask: Enter as four decimal numbers separated by points (dotted-decimal notation).
	ROUTER	Internet 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.
	PROT	Protocol (RFC/IEEE): You can choose one of two transmission protocols for the Ethernet interface: RFC 894 or IEEE 802.2/802.3.
	HW	Connection (10 BaseT/ 10Base2): Here you define which of the two ports you wish to use for your network.

Soft key	Option	Meaning
	As of NC software 280 472-01:	
	HOST	Host name: This is the name used by the TNC 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 TNC will use the so-called null authentication. If you work with null authentication, the entries under UID, GID, DCM and FCM will be ignored.
DEFINE MOUNT	Definition of the devices in the network that can be addressed from the TNC. For each device you define a separate line in the table.	
	ADDRESS	Internet address of server: Enter as four decimal numbers separated by points (dotted-decimal notation). Your network supervisor can give you the internet address.
	RS	Datagram size for input [byte]: An entry of zero means that the optimum transfer size as indicated by the NFS server is used. Do not enter any other input values unless you have encountered throughput problems. Input range: 512 to 4096 bytes
	WS	Datagram size for output [byte]: An entry of zero means that the optimum transfer size as indicated by the NFS server is used. Do not enter any other input values unless you have encountered throughput problems. Input range: 512 to 4096 bytes
	TIMEOUT	Timeout [ms]: A Remote Procedure Call that is not answered by the NFS server is repeated after expiration of the time defined here. The standard value is 0 = 700. Do not enter a higher value unless the datagrams are led through several routers.
	HM	Hard mount (yes = 1 / no = 0): With a hard mount, the Remote Procedure Call is repeated until an answer is received from the NFS server. This has the advantage that after a server crash you can continue normal operation as soon as the server is up again. Use a soft mount if the NFS server is not always available.
	DEVICE NAME	TNC device name: The device name entered here is displayed at the TNC in the program management for the mounted network.

Soft key	Option	Meaning
	PATH	Directory: Enter the complete directory (note the proper capitalization) of the NFS server that you wish to mount.
	UID	User ID: Enter the user identification for accessing the files in the network. The entry must be a decimal number.
	GID	Group ID: Enter the group identification for accessing the files in the network. The entry must be a decimal number.
	DCM	Directory Create Mode: Here you assign access rights to directories on the NFS server. The entry is binary, with 3 places each for owner, group and the other users. Sequence: %<Owner><Group><Rest>;<Read><Write> <Search>
	FCM	File Create Mode: Here you assign access rights to files on the NFS server. The entry is binary, with 3 places each for owner, group and the other users. Sequence: %<Owner><Group><Rest>;<Read><Write> <Execute>
	AM	Auto mount (yes = 1/ no = 0): Here you define whether during power-on the TNC automatically mounts the network. If you do not mount automatically, you can mount at any time afterward by using the NET soft key in the program manager.
	Only NC software 280 470-xx:	
	DOMAIN	Domain name: This is the name used by the TNC in the network. If you use a domain-name server, you must enter the Fully Qualified Domain Name here. If you leave this entry blank, the TNC will use the so-called null authentication. The entries under UID, GID, DCM and FMC will be ignored.

Soft key	Option	Meaning
DEFINE PRINT	Define the names and addresses of the network printers. You can print directly from the TNC on the printers defined here.	
	ADDRESS	Internet address of printer: Enter as four decimal numbers separated by points (dotted-decimal notation). Your network supervisor can give you the internet address.
	DEVICE NAME	TNC device name: The device name entered here is displayed on the TNC after the print soft key has been activated.
	PRINTER NAME	Printer name: Name of the printer for the printer server.
SHOW ERROR	Any errors occurring during network operation are displayed.	
PING	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 package was not sent back within a certain period of time CAN NOT ROUTE: TNC could not send data package to the receiver	

Settings on the NFS server “CimconNFS for HEIDENHAIN”

Along with the Ethernet card you receive a CD ROM with the NFS server software “CimcoNFS for HEIDENHAIN”.

Sample settings for the TNC:

Soft key	Option	Input
DEFINE NET	ADDRESS	160.1.180.21
	MASK	255.255.0.0
	ROUTER	
	PROT	RFC
	HW	10BaseT
	HOST	
DEFINE MOUNT	ADDRESS	160.1.113.5
	RS	8192
	WS	8192
	TIMEOUT	0
	HM	1
	DEVICE NAME	PC
	PATH	
	UID	0
	GID	0
	DCM	%111111111
	FCM	%111111111
	AM	1

8.9 Protection Against Data Tampering

Due to the possibility of networking the TNC 426/430 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
EXTERNER ZUGRIFF AUS/EIN	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

Up until NC software 280 476-09 it is always possible to enter the PLC partition via the LSV2 protocol using the code number 807667, even if **PLCPASSWORD =** was used to define a new codenumber in OEM.SYS.

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.

Up until NC software 280 476-18, codenumber 807667 is used during machine backup, full backup and setup 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.

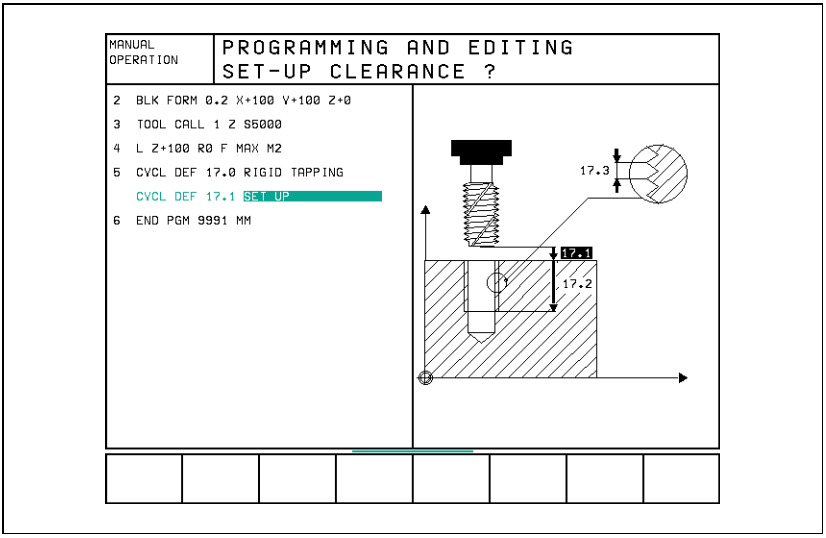
9 Original Equipment Manufacturer's (OEM) Cycles

9 Original Equipment Manufacturer's (OEM) Cycles

9.1 HEIDENHAIN Standard Cycles

Many common machining tasks requiring several routine steps are stored ready-programmed in the TNC as standard cycles. Coordinate transformations and several special functions are also available as cycles. The cycles are divided in groups and are called by soft key.

The TNC graphically illustrates the type of information required in the input parameters.



There are:

- cycles that go into effect immediately upon definition, and
- cycles that must be called with CYCL CALL after they have been defined.

The User's Manual for the TNC provides a comprehensive description of the HEIDENHAIN standard cycles.

9.2 CycleDesign

CycleDesign software enables you to:

- Add your own cycles and help graphics to the standard cycles, including the associated soft keys
- Change the soft-key structure
- Remove cycles
- Reorganize cycle groups
- Develop your own cycles

With CycleDesign you can access all files of the HEIDENHAIN standard cycles.

The program includes a soft-key editor for creating your own soft keys.

You can draw the help graphics to illustrate input parameters in any graphics program that can save files in DXF format, e.g., AutoSketch (not included).

The colors in which the TNC displays the graphics are defined in MP7364.x "Color Setting" on page 7–225.

9.3 Application of OEM Cycles

An OEM cycle is a HEIDENHAIN conversational program with variables called Q parameters. You write and test the program on the TNC. You define the numbers of the transfer parameters in the OEM cycle and in CycleDesign. The Q parameters Q200 through Q399 are reserved as transfer parameters for the OEM cycles. Up to NC software 280 474-xx, a maximum of 20 transfer parameters are allowed per cycle. Starting with 280 476-01, 32 are allowed.

Note

The Q parameters Q200 to Q336 are used by the HEIDENHAIN standard cycles. Always use the same parameter number for identical parameter functions!

You will find detailed information about programming with Q parameters in the User's Manual.

Illegal functions

The following functions are **not** permitted in OEM cycles:

- M functions M02, M30, M06 with program stop
- Programmed STOP block
- Program calls with PGM CALL
- Definition of Cycle 14 “Contour Geometry”

Cycle 14 “Contour Geometry” must be defined in the main program.

Q parameters with special meanings

Q parameters Q100 to Q199 are reserved for special functions of the TNC. Some of them have a specific meaning, such as tool radius, tool axes, etc.

You will find a description of these Q parameters in the User’s Manual.

Global and local Q parameters

The values of the global Q parameters can be entered or edited both in the calling part program as well as in the called OEM cycle. During value transfer from the calling program to the called program, the value of the global parameter is not changed.

Editing a local parameter in an OEM cycle does not affect the part program. Local Q parameters keep their values only in the current program. When an OEM cycle is called, all values of the local Q parameters are buffer stored and retain their old values after returning from the OEM cycle.

- Q200 to Q299 are global Q parameters.
- Q0 to Q59 are local Q parameters.
- Q60 to Q99 are either local or global. Use MP7251 to define the effect.

MP7251	**Defining Q parameters Q60 to Q99**
Input:	0 = Q60 to Q99 local
	1 to 40 = Q(100 – <input value>) to Q99 global

Parameters Q0 to Q59 are used in the HEIDENHAIN standard cycles for mathematical operations. We recommend this for your OEM cycles as well.

Optional parameters

CycleDesign lets you define optional parameters. Use a parameter as optional parameter if you expand an existing OEM cycle by one parameter.

The advantage is that if existing programs are loaded in which the old cycle (without the new parameters) is programmed, the new optional parameter is automatically inserted and preassigned with the default value.

If you were not to define the new parameter as optional, the cycle block could not be interpreted, and an ERROR block would be inserted.

Warning

The default value for the optional parameter must allow the TNC to run the old OEM cycle without error!

FN functions with special meanings

The following FN functions enable you to solve special tasks such as outputting error messages or transferring data from the NC to the PLC.

- FN14: Output of error messages and dialog prompts
- FN15: Transfer of error messages, dialogs and Q parameter values to a file or date interface
- FN17: Overwrite system data
- FN18: Read system data
- FN19: Assigning two numerical values or Q-parameter values from an OEM cycle to the PLC: “PLC Programming” on page 7–3
- FN20: Wait for condition to occur
- FN25: Overwrite the datum

Nesting of OEM cycles

When nesting OEM cycles, you must distinguish between DEF-active and CALL-active OEM cycles.

DEF-active cycles

- Are effective immediately upon definition
- Can be called by another OEM cycle

CALL-active cycles

- Must be specially activated with CYCL CALL
- Cannot be called by another OEM cycle

Note

When nesting OEM cycles, ensure that no Q parameter is assigned more than one meaning!

Managing more than one cycle tree

With CycleDesign you can build a cycle structure with up to 9 cycle trees. If you keep your HEIDENHAIN cycles only in the “HEIDENHAIN cycle tree” project type, in the event of an NC software update you need only update this project type, and not your OEM cycles.

CycleDesign manages the project types independently of each other. The control then links them into the soft-key rows.

Loading the OEM-specific cycle structure

- On your PC, use CycleDesign to transfer your customized OEM cycles to the new soft keys and help graphics.
- Transfer the new cycle structure with CycleDesign to the TNC.

The system file PLC:\CYCLE.SYS is opened on the TNC. The directories and files of the OEM cycles, soft keys and help graphics are defined in the system file.

Note

You can also store OEM cycles on the hard disk of the TNC in coded form to protect them from unauthorized alteration. If the TNC does not find the file PLC:\CYCLE.SYS, the HEIDENHAIN standard cycle structure goes into effect.

Save the PLC partition with TNCBACK.EXE and provide the floppy disk with your machine.

Example: HEIDENHAIN Standard Cycle 201 REAMING

The new HEIDENHAIN standard cycles (beginning with Cycle 200) are written like OEM cycles.

HEIDENHAIN conversational program	Comment
`0 BEGIN PGM 201 MM P1 FN 17: SYSWRITE ID212 = +3`	Automatic compensation in the tool axis
`2 FN 9: IF +Q110 EQU +0 GOTO LBL 199`	
`3 FN 9: IF +Q110 EQU +1 GOTO LBL 199`	Inquiry whether spindle on with M3 or M4
`4 FN 14: ERROR = 1000`	Error message, spindle
`5 LBL 199`	
`6 FN 10: IF +Q109 NE –1 GOTO LBL198`	Inquiry whether tool is active
`7 FN 14: ERROR = 1001`	Error message, tool axis is missing
`8 LBL 198`	
`9 FN 12: IF +Q201 LE +0 GOTO LBL 197`	Inquiry whether machining direction is negative
`10 FN 0: Q30 = -1`	If not, set the constant to negative
`11 FN 9: IF +0 EQU +0 GOTO LBL 194`	
`12 LBL 197`	
`13 FN 0: Q30 = +1`	Otherwise, set the constant to positive
`14 LBL 194`	
`15 FN 9: IF +Q97 EQU +1 GOTO LBL 193`	Inquiry whether signs were already negated
`16 CALL LBL 4`	
`17 LBL 193`	
`18 FN 1: Q19 = +Q203 + +Q200`	Calculate Z end position
`19 FN 9: IF +Q204 EQU +0 GOTO LBL 2`	Inquiry whether 2nd safety clearance was entered
`20 FN 1: Q19 = +Q203 + +Q204`	If so, set a new Z end position
`21 LBL 2`	
`22 FN 1: Q20 = +Q203 + +Q200`	Calculate Z preparatory position
`23 FN 1: Q24 = +Q203 + +Q201`	Calculate total hole depth
`24 FN 0: Q25 = +Q208`	Transfer the feed rate for retraction
`25 FN 10: IF +Q208 NE +0 GOTO LBL 4`	Inquiry whether feed rate for retraction was entered
`26 FN 0: Q25 = +Q206`	No: Feed rate for retraction = feed rate for reaming
`27 LBL 4`	Negate sign-critical parameters
`28 FN 9: IF +Q97 EQU +1 GOTO LBL 192`	If sign was already reversed in the DEF cycle, skip negation

HEIDENHAIN conversational program	Comment
29 FN 3: Q200 = +Q200 * +Q30	
30 FN 3: Q204 = +Q204 * +Q30	
31 LBL 0	
32 LBL 192	
33 L Z+Q20 R0 F MAX	Approach the Z preparatory position
34 L Z+Q24 R0 FQ206	Traverse to total hole depth
35 CYCL DEF 9.0 DWELL TIME	
36 CYCL DEF 9.1 DWELL Q211	If desired, dwell at bottom
37 L Z+Q20 R0 FQ25	Retract to setup clearance
38 L Z+Q19 R0 F MAX	If desired, retract to 2nd setup clearance
39 END PGM 201 MM P	

9.4 Compatibility with Earlier OEM Cycles

The following possibilities are provided to enable you to continue using earlier OEM cycles with program numbers 999 999 68.H to 999 999 99.H on the TNC.

Earlier OEM cycle:

- Use CycleDesign to include the cycle in the cycle structure of the TNC. Like a standard cycle, the cycle can be defined and called via soft key under dialog guidance.
- You can have the cycle run only in existing NC programs. The cycle cannot be used by the TNC user in the new NC programs, but existing NC programs containing the cycle can be run.

You can access earlier OEM cycles, even without CycleDesign.

Earlier OEM cycle:

- Call with **CYCL DEF 12** (program call) and **CYCL CALL.** The cycle cannot be programmed by soft key. It is defined with **CYCL DEF 12** and then called with **CYCL CALL** or **M99.**
- Call with **PGM CALL.** The cycle cannot be programmed by soft key. It is called with **PGM CALL.**

You will find detailed information about the use of earlier OEM cycles in the User's Manual for CycleDesign.

10 Error Messages

10 Error Messages

10.1 DSP Error Messages NC Software 280 470-xx, 280 472-xx, 280 474-xx

10.1.1 Non-Axis-Specific DSP Error Messages with Error Code

Error message	**Cause of error, corrective action**	**As of NC SW**
DSP ERROR FF01	Undefined error, no traceable cause (data processing error)	280 470-01, 280 472-01, to 280 474-03
DSP ERROR FF02	Host command not recognized, not valid (data processing error)	
DSP ERROR FF03	Host and DSP watchdogs disagree (data processing error)	
DSP ERROR FF04	Undefined interrupt (data processing error)	
DSP ERROR FF05	Unknown hardware ID (hardware and software incompatible)	
DSP ERROR FF06	No V_NOML value received from host (data processing error)	
DSP ERROR FF07	AC fail (power module error)	
DSP ERROR FF09	Stack overflow (data processing error)	
DSP ERROR FF0A	Triangular signal pulse-width modulation (hardware problem or wrong value in MP2180)	
DSP ERROR FF0B	Error upon memory request (data processing error)	
DSP ERROR FF0C	No velocity control interrupt (data processing error)	
DSP ERROR FF0D	Checksum error, code (data processing error)	
DSP ERROR FF0E	Timeout of the speed interrupt (data processing error)	
DSP ERROR FF0F	Error when initializing a software timer (data processing error)	
DSP ERROR FF10	Error during LSV2 transmission (data processing error)	
DSP ERROR FF11	Drive start without previous synchronization (data processing error)	
DSP ERROR C001	Undefined error, no traceable cause (data processing error)	280 474-04
DSP ERROR C002	Host command not recognized, not valid (data processing error)	
DSP ERROR C003	Host and DSP watchdogs disagree (data processing error)	
DSP ERROR C004	Undefined interrupt (data processing error)	
DSP ERROR C005	Unknown hardware ID (hardware and software incompatible)	
DSP ERROR C006	No V_NOML value received from host (data processing error)	
DSP ERROR C007	AC fail (power module error)	
DSP ERROR C009	Stack overflow (data processing error)	

Error message	Cause of error, corrective action	As of NC SW
DSP ERROR C00A	Triangular signal pulse-width modulation (hardware problem or wrong value in MP2180)	280 474-04
DSP ERROR C00B	Error upon memory request (data processing error)	
DSP ERROR C00C	No velocity control interrupt (data processing error)	
DSP ERROR C00D	Checksum error, code (data processing error)	
DSP ERROR C00E	Timeout of the speed interrupt (data processing error)	
DSP ERROR C00F	Error when initializing a software timer (data processing error)	
DSP ERROR C010	Error during LSV2 transmission (data processing error)	
DSP ERROR C011	Drive start without previous synchronization (data processing error)	
DSP ERROR C012	No TL and sync-source initializing (data processing error)	

In the following error messages, in addition to the error code, a number is output with the following meanings:

- y = 0: DSP axes of axes 1 to 6
- y = 1: Spindle DSP
- y = 2: DSP axes of axes 7 to 9

Error message	Cause of error, corrective action	As of NC SW
DSP ERROR 1000 y	Timeout during a command (data processing error)	280 470-01, 280 472-01, 280 474-01
DSP ERROR 1001 y	Incorrect acknowledgment of a command (data processing error)	
DSP ERROR 1002 y	Command is sent before the previous command is acknowledged (data processing error)	
DSP ERROR 1003 y	Synchronization error between DSP and NC (data processing error)	
DSP ERROR 1004 y	Incorrect message DSP/NC (data processing error)	
DSP ERROR 1005 y	Too many commands NC/DSP (data processing error)	
DSP ERROR 1100 y	Error during the checksum calculation (data processing error)	
DSP ERROR 1101 y	Timeout during word transfer command, loading DSP code (data processing error)	
DSP ERROR 1102 y	Timeout during the checksum calculation (data processing error)	
DSP ERROR 1103 y	Timeout during GO command (data processing error)	
DSP ERROR 1104 y	File not found (data processing error)	

10.1.2 Axis-Specific DSP Error Messages with Error Code

With the following error messages, in addition to the error code, the number of the axis or spindle is output:

- y = 0 to 8: Axes 1 to 9
- y = 9 or F: Spindle

Error message	Cause of error, corrective action	As of NC SW
DSP ERROR F010 y	Motor type unknown, MP2200 (error in motor table or MP2200)	280 470-01, 280 472-01, to 280 474-03
DSP ERROR F020 y	Reserved	
DSP ERROR F030 y	Reserved	
DSP ERROR F040 y	Number of pole pairs too large (error in motor table or MP2230)	
DSP ERROR F050 y	ASM: field-defining current (error in motor table)	
DSP ERROR F060 y	Grating period of velocity encoder (error in motor table)	
DSP ERROR F070 y	ASM: Rotor time constant (error in motor table)	
DSP ERROR F080 y	Break/rated speed (error in motor table)	
DSP ERROR F090 y	Unknown drive type, MP2000	
DSP ERROR F0A0 y	Reserved	
DSP ERROR F0B0 y	Reserved	
DSP ERROR F0C0 y	Reserved	
DSP ERROR F0D0 y	Current sensor voltage (error in power module table)	
DSP ERROR F0E0 y	Peak current in power module (error in power module table)	
DSP ERROR F0F0 y	Current controller proportional factor too high	
DSP ERROR F100 y	Current controller integral factor too high	
DSP ERROR F110 y	Motor temperature	
DSP ERROR F120 y	Reserved	
DSP ERROR F130 y	Oscilloscope parameter is incorrect. Only for test purposes (data processing error)	
DSP ERROR F140 y	Rated current of power module (error in power module)	
DSP ERROR F150 y	Rated current of motor (error in motor table)	
DSP ERROR F160 y	Peak current of motor (error in motor table)	
DSP ERROR F170 y	Max. motor speed (error in motor table)	
DSP ERROR F180 y	SM: wrong angle compensation values, MP2340, MP2350	
DSP ERROR F190 y	Power module dc-link voltage incorrect	
DSP ERROR F1A0 y	Invalid rotational speed input selected	
DSP ERROR F1B0 y	Invalid PWM output	
DSP ERROR F1C0 y	Band-pass parameter incorrect, MP2540, MP2550	
DSP ERROR F200 y	Contamination of encoder Zn track, amplitude too small	
DSP ERROR F210 y	Contamination of encoder Z1 track, amplitude too small	
DSP ERROR F220 y	Reserved	
DSP ERROR F230 y	Motor temperature too high	
DSP ERROR F240 y	Unknown counter chip type at rotational speed input (hardware problem)	

Error message	Cause of error, corrective action	As of NC SW
DSP ERROR F250 y	Power module switches off during operation (external operation error)	280 470-01, 280 472-01, to 280 474-03
DSP ERROR F260 y	Reserved	
DSP ERROR F270 y	Excessive angular deviation during alignment, Zn/Z1 tracks do not match (encoder error)	
DSP ERROR F280 y	Motor is uncontrollable, at I_{max} no expected rotary movement (drive error)	
DSP ERROR F290 y	Error in 3-D probe/evaluation. No latching with L1 input (G19/G26) (hardware problem)	
DSP ERROR F2A0 y	Incorrect ref. position found (hardware problem)	
DSP ERROR F2B0 y	Standstill detection (drive error)	
DSP ERROR F2C0 y	Actual current of motor exceeds limit (drive error)	
DSP ERROR F2D0 y	Status error in PWM chip (hardware problem)	
DSP ERROR F2E0 y	Incorrect rated voltage of motor (error in motor table)	
DSP ERROR C110 y	Motor type unknown, MP2200 (error in motor table or MP2200)	280 474-04
DSP ERROR C120 y	Reserved	
DSP ERROR C130 y	Reserved	
DSP ERROR C140 y	Number of pole pairs too large (error in motor table or MP2230)	
DSP ERROR C150 y	ASM: field-defining current (error in motor table)	
DSP ERROR C160 y	Grating period of velocity encoder (error in motor table)	
DSP ERROR C170 y	ASM: Rotor time constant (error in motor table)	
DSP ERROR C180 y	Break/rated speed (error in motor table)	
DSP ERROR C190 y	Unknown drive type, MP2000	
DSP ERROR C1A0 y	Reserved	
DSP ERROR C1B0 y	Reserved	
DSP ERROR C1C0 y	Reserved	
DSP ERROR C1D0 y	Current sensor voltage (error in power module table)	
DSP ERROR C1E0 y	Peak current in power module (error in power module table)	
DSP ERROR C1F0 y	Current controller proportional factor too high	
DSP ERROR C200 y	Current controller integral factor too high	
DSP ERROR C210 y	Motor temperature	
DSP ERROR C220 y	Reserved	
DSP ERROR C230 y	Oscilloscope parameter is incorrect. Only for test purposes (data processing error)	
DSP ERROR C240 y	Rated current of power module (error in power module)	
DSP ERROR C250 y	Rated current of motor (error in motor table)	
DSP ERROR C260 y	Peak current of motor (error in motor table)	

Error message	Cause of error, corrective action	As of NC SW
DSP ERROR C270 y	Max. motor speed (error in motor table)	280 474-04
DSP ERROR C280 y	SM: wrong angle compensation values, MP2340, MP2350	
DSP ERROR C290 y	Power module dc-link voltage incorrect	
DSP ERROR C2A0 y	Invalid rotational speed input selected	
DSP ERROR C2B0 y	Invalid PWM output	
DSP ERROR C2C0 y	Band-pass parameter incorrect, MP2540, MP2550	
DSP ERROR C300 y	Contamination of encoder Zn track, amplitude too small	
DSP ERROR C310 y	Contamination of encoder Z1 track, amplitude too small	
DSP ERROR C320 y	Reserved	
DSP ERROR C330 y	Motor temperature too high	
DSP ERROR C340 y	Unknown counter chip type at rotational speed input (hardware problem)	
DSP ERROR C350 y	Power module switches off during operation (external operation error)	
DSP ERROR C360 y	Reserved	
DSP ERROR C370 y	Excessive angular deviation during alignment, Zn/Z1 tracks do not match (encoder error)	
DSP ERROR C380 y	Motor is uncontrollable, at I_{max} no expected rotary movement (drive error)	
DSP ERROR C390 y	Error in 3-D probe/evaluation. No latching with L1 input (G19/G26) (hardware problem)	
DSP ERROR C3A0 y	Incorrect ref. position found (hardware problem)	
DSP ERROR C3B0 y	Standstill detection (drive error)	
DSP ERROR C3C0 y	Actual current of motor exceeds limit (drive error)	
DSP ERROR C3D0 y	Status error in PWM chip (hardware problem)	
DSP ERROR C3E0 y	Incorrect rated voltage of motor (error in motor table)	

10.1.3 DSP Error Messages with Text

Error message	Cause of error, corrective action
Power module in axis <axis> too weak	Power module for the displayed axis is too weak.
<axis> motor enc. line count too high	Line count of the motor encoder for the displayed axis too high.
Motor <axis>: Xh; X2; f-n; R2 incorrect	One datum is incorrect in the following motor data for the displayed axis: Xh (magnetizing reactance), X2 (rotor leakage reactance), f-n (rated frequency), R2 (Rotor resistance cold)
Motor <axis>: n-n; f-n incorrect	One datum is incorrect in the following motor data for the displayed axis: n-n (rated speed), f-n (rated frequency)
Power stage<axis>: U-Imax incorrect	Voltage of the current sensor (U-Imax) of the power module for the displayed axis is incorrect.
Power stage<axis>: I-max incorrect	Peak current (I-max) of the power module for the displayed axis is incorrect.
Motor <axis>: t-max incorrect	Max. temperature of the motor for the displayed axis is incorrect.
Motor <axis>: I-n incorrect	Rated current of the motor for the displayed axis is incorrect.
Motor <axis>: I-max incorrect	Peak current of the motor for the displayed axis is incorrect.
Motor <axis>: n-max incorrect	Max. speed of the motor for the displayed axis is incorrect.
Axis <axis>: MP2340/MP2350 incorrect	MP2340/MP2350 (field-angle offset) for the displayed axis is incorrect.
Axis <axis>: MP2190 incorrect	MP2190 (dc-link voltage) for the displayed axis is incorrect.
Axis <axis>: MP120/MP121 incorrect	MP120/MP121 (Assignment of the speed command signal outputs) for the displayed axis is incorrect.
Axis <axis>: MP2540/MP2550 incorrect	MP2540/MP2550 (damping of band-rejection filter) for the displayed axis is incorrect.
Motor encoder <axis> zn ampl. too small	Zn amplitude of the motor encoder for the displayed axis is too low.
of motor encoder <axis> zl ampl. too small	Z1 amplitude of the motor encoder for the displayed axis is too low.
Motor encoder <axis> temperature too low	Temperature of the motor encoder for the displayed axis is too low.
<axis> motor encoder defective	Motor encoder for the displayed axis is defective.
Motor <axis>: speed not equal to Imax	The current speed of the motor does not match the expected speed at Imax. Direction of rotation may be incorrect.
Motor encoder <axis> frequency too high	The maximum permissible input frequency at the motor encoder was exceeded.
Motor <axis>: is not turning	The motor of the displayed axis is not rotating.
Power module <axis> not ready	The readiness signal for the power module of the displayed axis was switched off during operation.
Axis <axis>: MP112/MP113 incorrect	MP112/MP113 (assignment of speed encoder inputs) for the displayed axis is incorrect.

10.2 DSP Error Messages as of NC Software 280 476-01

10.2.1 Non-Axis-Specific Error Messages without Control Reset

Error message	Cause	Corrective action	As of NC SW
8010 LSV2 transmission error	■ Interrupted LSV2 connection. ■ Internal software error.	■ Check the LSV2 connection. ■ Inform your service agency. • Check software version.	280 476-01
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).	280 476-09
8041 Iz of UV 1xx too high	■ DC-link current of UV 1xx too high.	■ Continue working with lower power (reduce the feed rate).	280 476-09
8042 Leakage current of UV 1xx too high	■ Isolation problem (e.g. defective motor).	■ Inform your service agency. • Check the motor. • Check the wiring.	280 476-09
8080 Uz of UV 1xx too high	■ DC-link voltage of the power supply unit too high.	■ Inform your service agency. • Check the machine parameters (deceleration of spindle). • If required, check the braking resistor. • Replace the power supply unit.	280 476-09
8082 MCU command unknown	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
8086 Probing already active	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	280 476-01
8092 Pos. contr. cyc. time error	■ MCU is outputting incorrect cycle time for CCU position controller. ■ A hardware error has occurred.	■ Inform your service agency. • Check machine parameter MP7600.x. • Exchange drive control board.	280 476-01
9800 MCU command unknown	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	280 476-04
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.	280 476-09
A001 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.	280 476-01

Error message	Cause	Corrective action	As of NC SW
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.	280 476-04
B800 Safe inputs <Input> not equal	■ Wiring error X65, X66 (,X67). ■ Safety module defective.	■ Inform your service agency. • Check the wiring X65, X66 (,X67). • Exchange the safety module.	280 476-09

10.2.2 Axis-Specific Error Messages without Control Reset

<Axis>: 1 to 9 = axes 1 to 9
S1 = spindle 1
S2 = spindle 2

Error message	Cause	Corrective action	As of NC SW
8100 Warning motor temperature from <axis>	■ If the motor temperature continues to increase, the unit will be switched off. • Motor overload. • Machine parameters incorrect.	■ Reduce the motor load. ■ Inform your service agency. • Reduce machine parameters.	280 476-01
8110 Warning I2t monitoring of <axis>	■ If the motor current continues to increase, the unit will be switched off. • Motor or power module overload. • Machine parameter MP230x.x incorrect.	■ Reduce motor or power module load. ■ Inform your service agency. • Check machine parameter MP230x.x.	280 476-09
8120 Heat-sink temperature UM 1xx <axis>	■ Temperature of UM1xx power modules' heat sinks 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).	280 476-09
8400 No drive-on command for <axis>	■ Speed controller waiting for drive-on command; PLC is not sending a drive-on command.	■ Check the PLC program. ■ Inform your service agency. • Check software version.	280 476-09

Error message	Cause	Corrective action	As of NC SW
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.	280 476-01
8B10 Traverse direction <axis> incorrect	■ DIR entry in motor table is incorrect. ■ Check MP1040. ■ Incorrect motor power connection.	■ Inform your service agency. • Check the DIR entry in the motor table. • Check MP1040. • Check the motor power connection.	280 476-09
8140 Error <axis> field orientation	■ No field orientation possible. ■ Incorrect relation between electrical field and mechanical motor motion. ■ Incorrect motor encoder signal. ■ Incorrect motor connection. ■ Mechanical brakes not released.	■ Inform your service agency. • Check entry in MP331 and MP332. • Check entry in MP2020. • For linear motors: In motor table, check distance per electrical motor revolution. • Check motor encoder connection. • Check motor connection. • Release brakes during orientation.	280 476-08
8B30 Motor temperature <axis> too high	■ Measured motor temperature 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 (2 kW at 25 °C).	280 476-01

Error message	Cause	Corrective action	As of NC SW
8B40 Power supply unit <axis> not ready	■ Inverter is not ready for operation. ■ No pulse release for the power module. ■ U_z too high. ■ 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 control and cabling of the pulse release. ■ Check U_z. ■ Check the emergency stop circuit. ■ If the power supply is not regenerative: Is the braking resistor connected? ■ If the power supply is regenerative: Is the 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.	280 476-01
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 control 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.	280 476-01
8B60 Axis module <axis> IGBT error	■ 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.	280 476-01
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 the motor.	280 476-01

Error message	Cause	Corrective action	As of NC SW
8BC0 Motor current <axis> too high	■ Incorrect current controller parameters. ■ Incorrect parameters in the motor table. ■ Power module defective. ■ Motor cable defective. ■ Motor defective. ■ 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.	280 476-01
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 feedrate, 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.	280 476-09
A110 Safe speed SRG exceeded <axis>	■ The rotational speed limit SBH was exceeded while the protective door was open and the key switch was turned to "automatic."	■ Inform your service agency.	280 476-01
AC00 Mot. enc. amp. too high <axis>	■ Noise on motor encoder signal. ■ Short circuit in motor encoder cable. ■ Signal amplitude of motor encoder is too high.	■ Inform your service agency. • Check connection of motor encoder (ground connection). • Check the motor encoder.	280 476-04
AC10 <Axis> amplitude too small	■ Interruption in motor encoder cable. ■ Motor encoder signal amplitude missing.	■ Inform your service agency. • Check connection of motor encoder. • Check the motor encoder.	280 476-06
AC20 <Axis> frequency too high	■ Noise on motor encoder signal.	■ Inform your service agency. • Check connection of motor encoder (ground connection). • Check the motor encoder.	280 476-06
E130 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.	280 476-01

10.2.3 Non-Axis-Specific Error Messages with Control Reset

Error message	Cause	Corrective action	As of NC SW
C001 Undefined error	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
C002 MCU command invalid	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
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.	280 476-01
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.	280 476-01
C005 Unknown hardware identifier	■ Software does not fit the hardware. ■ Hardware defective.	■ Inform your service agency. • Check software version. • Exchange drive control board.	280 476-01
C007 DC-link voltage too low	■ Line power interrupted. ■ Inverter defective.	■ Check your line power supply. ■ Inform your service agency. • Check the inverter.	280 476-01
C009 Stack overflow	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
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.	280 476-01
C00B Too little main memory	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
C00D Program checksum error	■ Internal software or hardware error.	■ Inform your service agency. • Check software version. • Exchange drive control board.	280 476-01
C00E Controller software timeout	■ Internal software or hardware error.	■ Inform your service agency. • Check software version. • Exchange drive control board.	280 476-01
C00F Error in software timer	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
C011 Softw. synchronization err.	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
C012 Pos. control err. Cycle time	■ MCU is outputting erroneous cycle time for CCU position controller. ■ Hardware error.	■ Inform your service agency. • Check machine parameter MP7600.x. • Exchange drive control board.	280 476-01

Error message	Cause	Corrective action	As of NC SW
C013 PWM frequency error	■ Entered PWM frequency in MP2180 lies outside the permissible input range.	■ Inform your service agency. • Check MP2180.	280 476-01
D000 DP RAM area overlap	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	280 476-01
D100 Software error	■ An internal software error has occurred.	■ Inform your service agency. • Check software version.	280 476-07
E001 Status NR1/NR2 not equal	■ NR2 input incorrectly connected. ■ Internal software error.	■ Inform your service agency. • Check the wiring. • Check software version.	280 476-01
E002 Status NE1/NE2 not equal	■ NE2 input incorrectly connected. ■ Internal software error.	■ Inform your service agency. • Check the wiring. • Check software version.	280 476-01
E003 PLC module 9169 illegal	■ PLC Module 9169 in safety-oriented software (illegal). ■ Internal software error.	■ Inform your service agency. • Check the PLC program. • Check software version.	280 476-01
E006 Wrong RDY status of spindle	■ Cabling to inverter defective. ■ Spindle not connected (spindle release relay). ■ Inverter defective.	■ Inform your service agency. • Check the inverter and cabling.	280 476-01
E007 Wrong RDY status of axes	■ Cabling to inverter defective. ■ No axis connected (axis release relay). ■ Inverter defective.	■ Inform your service agency. • Check the inverter and cabling.	280 476-01
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.	280 476-01
E009 Incorrect gear range	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
E00A Safe machine parameter erroneous	■ CRC checksum does not fit the entered safe MPs.	■ Inform your service agency. • Check the safe machine parameters.	280 476-01
E00B Cutout channels test error	■ Machine key depressed (ZT.HR, ZT.MB, MT signal).	■ Inform your service agency. • Check the wiring X65, X66 (, X67). • Check the machine keys.	280 476-01
E00C Error in MP transfer	■ MP3210 or MP3510 incorrect. ■ Software error MCU.	■ Inform your service agency. • Check MP3210 and MP3510. • Check software version.	280 476-01

Error message	Cause	Corrective action	As of NC SW
E00D Error in MP3510 transfer	■ MP3510 incorrect. ■ Software error MCU.	■ Inform your service agency. • Check MP3510. • Check software version.	280 476-06
E00E Error in MP2020 transfer	■ MP2020 incorrect. ■ Software error MCU.	■ Inform your service agency. • Check MP2020. • Check software version.	280 476-08

10.2.4 Axis-Specific Error Messages with Control Reset

<Axis>: 1 to 9 = axes 1 to 9
S1 = spindle 1
S2 = spindle 2

Error message	Cause	Corrective action	As of NC SW
C110 Unknown motor type <axis>	■ Error in MP file or in motor table. ■ Internal software error.	■ Inform your service agency. • Check MP file and motor table. • Check software version.	280 476-01
C140 Pole pair no. too large <axis>	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	280 476-01
C150 Field current error <axis>	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	280 476-01
C160 Grating per. of motor encoder <axis>	■ Measured grating period does not agree with entry in the motor table.	■ Inform your service agency. ■ Check the motor table. ■ Check the motor.	280 476-01
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.	280 476-01
C180 Rated speed error <axis>	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	280 476-01
C1D0 Current sensor voltage <axis>	■ Incorrect entry in power module table.	■ Inform your service agency. • Check the power module table.	280 476-01
C1E0 Imax of power module <axis>	■ Incorrect entry in power module table.	■ Inform your service agency. • Check the power module table.	280 476-01
C210 Tmax of motor table <axis>	■ Incorrect temperature entry in motor table.	■ Inform your service agency. • Check the motor table.	280 476-01
C230 Oscilloscope error <axis>	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
C240 Irated of power module <axis>	■ Incorrect entry in power module table.	■ Inform your service agency. • Check the power module table.	280 476-01
C250 Irated of motor <axis> incorrect	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	280 476-01
C260 Imax of motor <axis> incorrect	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	280 476-01
C270 Nmax of motor <axis> incorrect	■ Incorrect entry in motor table.	■ Inform your service agency. • Check the motor table.	280 476-01

Error message	Cause	Corrective action	As of NC SW
C280 Field angle <axis> incorrect	■ Incorrect entry in MP2340 or MP2350.	■ Inform your service agency. • Check entry in MP2340/ MP2350.	280 476-01
C290 Uz <axis> incorrect	■ Incorrect entry in MP2190 (dc-link voltage Uz).	■ Inform your service agency. • Check the entry in MP2190.	280 476-01
C2A0 Encoder input <axis>	■ Incorrect entry in MP112 or MP113 (speed encoder). ■ An internal software error has occurred.	■ Inform your service agency. • Check the entry in MP112/ MP113. • Check software version.	280 476-01
C2B0 PWM output <axis>	■ Incorrect entry in MP120 or MP121 (nominal speed output). ■ An internal software error has occurred.	■ Inform your service agency. • Check the entry in MP120/ MP121. • Check software version.	280 476-17
C2C0 Band filter parameter <axis>	■ Incorrect entry in MP2540, MP2541, MP2550 or MP2551. ■ An internal software error has occurred.	■ Inform your service agency. • Check the entry in MP2540, MP2541, MP2550 and MP2551. • Check software version.	280 476-01
C2D0 Motor encoder line count <axis>	■ Motor encoder line count was changed during operation.	■ Restart control with the END key.	280 476-08
C2E0 Motor pole-pair number <axis>	■ Motor pole-pair number was changed during operation.	■ Restart control with the END key.	280 476-08
C2F0 DIR in motor table <axis>	■ DIR in motor table was changed during operation.	■ Restart control with the END key.	280 476-08
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.	280 476-01
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.	280 476-01

Error message	**Cause**	**Corrective action**	**As of NC SW**
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]).	280 476-01
C340 Unknown counter range <axis>	■ Hardware defective. ■ Incorrect software version.	■ Inform your service agency. • Check software version. • Exchange drive control board.	280 476-01
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 control 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.	280 476-14
C370 Angular deviation of motor encoder <axis>	■ Motor encoder defective. ■ Motor encoder cable defective. ■ Drive control board defective.	■ Inform your service agency. • Check motor encoder and leads. • Exchange drive control board.	280 476-01
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 defective. ■ Incorrect motor table entry (direction of rotation). ■ Motor defective.	■ Check motor cabling. ■ Inform your service agency. • Check motor and motor encoder cable. • Check motor table entry.	280 476-14
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.	280 476-01

Error message	**Cause**	**Corrective action**	**As of NC SW**
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.	280 476-17
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). ■ Assignment of PWM outputs incorrectly entered in MP120. ■ Assignment of encoder inputs incorrectly entered in MP112. ■ Motor power cable switched. ■ Motor encoder cable switched. ■ Incorrect motor connection.	■ Inform your service agency. • Check the inverter. • Check motor and cabling. • Check machine parameters.	280 476-14
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	280 476-14
C3D0 PWM component defect <axis>	■ An internal hardware error has occurred.	■ Inform your service agency. • Exchange drive control board.	280 476-01
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.	280 476-01

Error message	Cause	Corrective action	As of NC SW
E120 Safe function call error	■ Internal software error.	■ Inform your service agency. • Check software version.	280 476-01
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.	280 476-01
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.	280 476-01
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.	280 476-01
E130 Position error too large <axis>	■ MP650 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.	280 476-01

10.3 Comparison of Old and New Error Messages

As of NC software 280 470-05 / 280 472-01

All NC error messages are displayed in plain language. As of NC software 280 472-01 or 280 470-05, when an error message appears you can press the HELP key to access more detailed information on the error message.

Old error message (coded)	New error message (conversational)
Gross positioning error <axis> A	Excessive following error in <axis>
Gross positioning error <axis> B	Nominal speed value too high <axis>
Gross positioning error <axis> C	Movement monitoring error in <axis> A
Gross positioning error <axis> D	Standstill monitoring error in <axis>
Gross positioning error <axis> E	Excessive offset in <axis>
Gross positioning error <axis> F	Movement monitoring error in <axis> B
Gross positioning error <axis> G	Analog output already assigned <axis>
Encoder <axis> defective A	Encoder amplitude too low <axis>
Encoder <axis> defective B	Encoder <axis> frequency too high
Encoder <axis> defective C	Encoder <axis> defective
Encoder <axis'> defective A	Encoder <axis'>: amplitude too low
Positioning error	Excessive following error in <axis>
Error in PLC program 1Q	PLC: M4005, M4006, M4007 incorrect
Error in PLC program 1R	PLC: More than one strobe active
Error in PLC program 00	PLC: Invalid command
Error in PLC program 02	PLC: Invalid operand type
Error in PLC program 03	PLC: Operand not found
Error in PLC program 04	PLC: Operand incorrect
Error in PLC program 05	PLC: Error in text after command
Error in PLC program 06	PLC: Line too long
Error in PLC program 07	PLC: Label not defined
Error in PLC program 08	PLC: End of block not found
Error in PLC program 09	PLC: Program too long
Error in PLC program 10	PLC: Assignment in parentheses
Error in PLC program 11	PLC: Too many parentheses
Error in PLC program 12	PLC: Jump incorrectly programmed
Error in PLC program 13	PLC: Closing parenthesis w/o opening
Error in PLC program 14	PLC: Label incorrectly programmed
Error in PLC program 15	PLC: Label incorrectly programmed
Error in PLC program 16	PLC: Jump incorrectly programmed
Error in PLC program 17	PLC: Parentheses not closed
Error in PLC program 18	PLC: Label defined twice
Error in PLC program 19	PLC: Word assignment missing
Error in PLC program 20	PLC: Logic assignment missing
Error in PLC program 21	PLC: Word accumulator not loaded

Old error message (coded)	New error message (conversational)
Error in PLC program 22	PLC: Logic accumulator not loaded
Error in PLC program 23	PLC: Opening parenth. incorrect
Error in PLC program 24	PLC: Incorrect type in parenth.
Error in PLC program 25	PLC: Jump incorrectly programmed
Error in PLC program 26	PLC: ENDC/ENDK without beginning
Error in PLC program 27	PLC: Error in CASE/KFIELD
Error in PLC program 28	PLC: Too many entries in CASE
Error in PLC program 29	PLC: CASE/KFIELD is empty
Error in PLC program 30	PLC: String accumulator not loaded
Error in PLC program 31	PLC: String within parentheses
Error in PLC program 32	PLC: String assignment missing
Error in PLC program 33	PLC: Global/external incorrect
Error in PLC program 34	PLC: Too many modules
Error in PLC program 35	PLC: File not found
Error in PLC program 36	PLC: File too long
Error in PLC program 37	PLC: Too many local labels
Error in PLC program 38	PLC: Too many global labels
Error in PLC program 39	PLC: External label not defined
Error in PLC program 40	PLC: External label in CASE
Error in PLC program 41	PLC: External label in JP
Error in PLC program 42	PLC: Global label defined twice
Error in PLC program 43	PLC: Incorrect program structure
Error in PLC program 44	PLC: Structure open at file end
Error in PLC program 45	PLC: Global in the main file
Error in PLC program 50	PLC: Excessive nesting
Error in PLC program 51	PLC: Stack underflow
Error in PLC program 52	PLC: Stack overflow
Error in PLC program 53	PLC: Timeout
Error in PLC program 54	PLC: CASE out of range
Error in PLC program 55	PLC: Subprogram not defined
Error in PLC program 56	PLC: Index range incorrect
Error in PLC program 57	PLC: Error table missing
Error in PLC program 58	PLC: Error in module call
Error in PLC program 90	PLC: Error table not .PET
Error in PLC program 91	PLC: Error table not found
Error in PLC program 92	PLC: Error table format incorrect

Up to NC software 280 470-04

Some of the NC error messages are displayed in code.

In the PLC editor or during compilation of the PLC program, the errors are displayed with the message **`Input error X`**. During compilation of the program after switch-on or when a run-time error occurs, the blinking error message **`Error in PLC program X`** is displayed.

Classification of errors:

Each error message indicates the time at which the error was recognized:

- E Error detected during editing. The line is not formatted.
- S Error detected during syntax check in the PLC editor (soft-key compile).
- (S) Error may have already been detected during the syntax check, otherwise during compiler run.
- C Error is detected during compiler run either after switching on the control or in the PLC programming operating mode.
- R Error detected during run time of PLC program.

Error code	Explanation
0 ESC	The line that has been read cannot be interpreted as a PLC command.
2 ESC	Invalid operand type: An unknown operand type was entered. The command cannot be applied to the entered operand type.
3 ESC	Operand not found. A type was entered for the operand, but no value.
4 ESC	Operand outside the permissible range. An operand number was specified that lies outside the value range for this operand.
5 ESC	No limiter after command. The PLC command is followed by further characters that cannot be interpreted.
6 ESC	No line ending found. The line is longer than 128 characters.
7 SC	Label not defined. A reference was made to a label that has not been defined with LBL, KFIELD or EXTERN.
8 SC	No block end found. At the end of the program file there are PLC commands that are not concluded by an EM or JP command. The danger therefore exists that an undefined program area is executed at run time.
9 SC	Program too long (RAM overflow). The complete length of the program code to be generated exceeds the storage space available in the control.
10 SC	Assignment in one parenthesis. An attempt was made to assign the result of a gated operand, although not all opening parentheses were closed.
11 SC	Excessive nesting. An attempt was made to nest more than 16 parenthetical expressions in each other.
12 SC	Jump in a sequence of gating operations. An unconditional jump was programmed although the assignment chain begun beforehand had not yet been assigned.

Error code	Explanation
13 SC	Closing parenthesis w/o opening. You programmed a closing parenthesis command without the associated opening parenthesis command.
14 SC	Label incorrectly programmed. A label was set within a parenthetical calculation. This is illegal because closing parenthesis commands cannot be executed without the associated opening parenthesis commands.
15 SC	Label within a sequence of gating operations. A label was programmed in a connective operation that was already started. This is illegal because the first command behind the label would then have to be interpreted, depending on the program, once as a logical connection and once as a load command.
16 SC	Jump within parentheses. A jump statement was programmed within parentheses. This is not possible because, due to the internal implementation, opening parentheses must be closed again. This could not happen in the event of a jump.
17 SC	Parentheses opened at block end. An EM instruction was programmed after an opening parenthesis. The parenthesis must be closed again.
18 SC	Label defined twice. A label name that was imported with EXTERN from another module was used again with a LBL or KFIELD instruction. A name reserved for internal modules (9000–9255) was used with an LBL, KFIELD or EXTERN instruction.
19 SC	Word assignment missing. A word logic operation was conducted. However, the result was not assigned to an operand, but to a new logic operation.
20 SC	Logic assignment missing. A logic operation was conducted. However, the result was not assigned to an operand, but to a new logic operation.
21 SC	Word accumulator not loaded. A command was programmed that logically connects, assigns or manipulates the loaded word accumulator, although the word accumulator was not previously loaded.
22 SC	Logic accumulator not loaded. A command was programmed that logically connects, assigns or manipulates the loaded logic accumulator, although the logic accumulator was not previously loaded.
23 SC	Accumulators not loaded for opening parenthesis. You programmed an opening parenthesis command without first beginning a logic or a word sequence.
24 SC	Incorrect type in parentheses. Depending on the logic operation formed before a parenthesis and the parenthesis command used, it is expected that the sequence in parentheses supplies a result of the same type (word/logic). If the types differ, the logic operation requested in the open-parenthesis command cannot be formed.

Error code	Explanation
25 SC	Conditional jump with invalid logic accumulator. You programmed a conditional jump (CMT/CMF/JPT/JPF/EMT/EMF) without first starting a logic operations sequence in the logic accumulator.
26 SC	ENDC/ENDK outside of a CASE/KFIELD statement. You programmed an ENDC command without a previous CASE statement. You programmed an ENDK command without a previous KFIELD label.
27 SC	Incorrect command within a CASE table/KFIELD. You programmed a command other than CM after a CASE statement and before the associated ENDC statement. You programmed a command other than K after a KFIELD statement and before the associated ENDK label.
28 SC	Too many table entries in CASE. You programmed a CASE branch with more than 128 entries.
29 SC	Empty CASE statement/KFIELD. You programmed a CASE statement followed immediately by an ENDC statement. You programmed a KFIELD statement followed immediately by an ENDK label.
30 SC	String accumulator not loaded. A command was programmed that logically connects, assigns or manipulates the loaded string accumulator, although the string accumulator was not previously loaded.
31 SC	String statement within parentheses. You programmed a string statement within parentheses. String operands cannot be nested with parentheses.
32 SC	Sting assignment missing. You started a new logic operations sequence without first assigning the logic operation formed in the string accumulator.
33 SC	GLOBAL/EXTERN not at beginning of file. You wrote the GLOBAL or EXTERN commands behind other program code in the file. These commands must always appear before the program code.
34 (S)C	Too many modules. You attempted to link more than 64 files into one program using the USES instruction.
35 (S)C	File not found. A file linked with the USES command cannot be found, or you attempted to link a *.PLC-type file when MP4010 = 0 (EPROM).
36 SC	File too long. The program code of a single file would be larger than 64 KB and therefore cannot be compiled. Split the file into several smaller files and link them with the USES command.
37 SC	Too many local labels. You assigned more than 1000 labels in a file. All LBL, KFIELD and EXTERN statements are added together along with the (hidden) labels created through structured commands. Split the file into several smaller files and link them with the USES command.
38 SC	Too many global labels. Over 1000 global labels were defined from all participating files.

Error code	Explanation
39 SC	External label not defined. A label declared with EXTERN has not been defined with GLOBAL in any of the associated modules.
40 SC	External label in CASE statement. A label declared with EXTERN has been inserted in the CM list of a CASE statement. Define a local module, which in the simplest case calls only the global module via CM.
41 SC	External label in CASE statement. You attempted to jump to a label defined with EXTERN using a JP/JPF/JPT statement.
42 (S)C	Global label defined twice. You defined the same label more than once with GLOBAL in the same or in several files.
43 SC	Incorrectly structured statement. You programmed an ELSE/ENDI/ENDW/UNTIL statement without a previous IF/ELSE/WHILE/REPEAT statement. Differently structured statements have been interlinked instead of nested within each other. The structures must always be closed in the order opposite to that in which they are opened!
44 SC	Structure open at file end. A structured command has been opened and not closed again prior to the end of the file.
45 SC	Global statement in the main file. You defined a module from the main file as GLOBAL. Only modules from files that are linked with the USES statement can be made accessible for other files through the GLOBAL statement.
50 R	Excessive nesting. You attempted to nest more than 32 module calls. You programmed a recursive module call that exceeds the limit of 32 levels.
51 R	Stack underflow. You attempted to retrieve data from the stack although it had not yet been written there.
52 R	Stack overflow. You attempted to write more than 128 bytes of data to the stack. Word operands (B/W/D/K) occupy 4 bytes each. Logic operands (M/I/O/T/C) occupy 2 bytes.
53 R	Timeout. The processing of the cyclically executed program section took longer than 10.5 milliseconds. Check the program substructure for very compute-intensive sections that you can start as submit jobs. The displayed processing time might increase during RS-232-C data transfer and in handwheel mode. In case of doubt, select handwheel mode and simultaneously start data transfer with RS-232-C (if possible 115 000 bps), then check "MAXIMUM PROCESSING TIME" in the PLC programming environment. 100% corresponds to 3.5 ms. At this utilization rate the block processing time is maintained. Values should not exceed 150% (Safety reserve for unfavorable operating conditions!).
54 R	CASE out of range. The operand for the CASE statement contains a value that cannot be interpreted as an offset in the CM table (< 0 or > table length –1).

Error code	Explanation
55 R	Subprogram not defined. At present this error cannot occur.
56 R	Indexed access outside the permissible range. The address for writing access to data types B/W/D/M/I/O/T/C is, through the inclusion of the index register, in an invalid region for these operand types. During access to a constant field, the index register contains a value that is not possible for this field (< 0 or > field length –1). Due to the inclusion of the index register, the address of a string leads to an illegal value. Due to the inclusion of the index register, the number of a dialog (S#Dn[X]) or an error message (S#En[X]) leads to an illegal value (< 0 or >999). During the addressing of a component string (Sn^X) the value range for the index register (0..127) was exceeded.
57 R	PLC error table missing. A PLC error module 9085/9086 was called although no error table was compiled, or there were no entries in the table. A PLC error module 9085/9086 was called or an error marker was set, although the error table was edited or deleted after compilation.
58 R	Error in module call. While PLC Module 9031 was overwriting an MP, an illegal value appeared during conversion of the MP.
90 C	PLC error table. The PLC error table selected in OEM.SYS is not a PET file.
91 C	PLC error table. The error table selected in OEM.SYS was not found (incorrect file name or path).
92 C	PLC error table. The error table selected in OEM.SYS file does not have an up-to-date binary format (e.g., after a software exchange).

10.4 TNC 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 TNC 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 TNC, 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.

10.5 Error Codes of the HEIDENHAIN Peripheral Devices

The following error messages can be displayed on the TNC:

Error code	Meaning
ERR: 001	Incorrect command code
ERR: 002	Illegal program name
ERR: 003	Data transfer error
ERR: 004	Program incomplete
ERR: 010	Program not on floppy disk
ERR: 011	Program is protected against deletion
ERR: 012	Program is now being written to
ERR: 013	Program table of contents is full
ERR: 014	Floppy disk is full
ERR: 100	Floppy disk not initialized
ERR: 101	Sector number too large
ERR: 102	Drive not ready
ERR: 103	Floppy disk is write-protected
ERR: 104	Data on floppy disk is faulty
ERR: 105	Sectors cannot be found
ERR: 106	Checksum error
ERR: 107	Disk controller faulty
ERR: 108	DMA error

10.6 Error Messages of the File System

The following error messages can be displayed on the TNC:

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 Subject Index

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

T

U

V

W

X