HEIDENHAIN

Technical Manual

TNC 426 CB/PB
TNC 430

NC-Software	280 470-12 280 471-12
	280 472-12 280 473-12
	280 474-12 280 475-12
	280 476-01 280 477-01

April 1999

208 896 21 · 7 · 4/99 · S · Printed in Germany · Subject to change without notice
(208 896 EC)

Foreword

This Technical Manual is intended for manufacturers and distributors of machine tools. It contains all the necessary information for the assembly, electrical installation, start-up, and PLC programming for the HEIDENHAIN contouring controls.

When hardware or software is improved in these HEIDENHAIN contouring controls you will receive a free delivery of updated information. Please arrange and insert this updated information in your manual without delay. This will ensure that your manual always remains at the current revision level.

You can use extracts from this manual for your machine documentation. An enlargement of the manual's format (17 cm x 24 cm) by a factor of 1.225 will produce pages in A4 format.

No documentation can be perfect. Like all living things it must grow and change. Among other things, it lives from your impulses and suggestions for improvement. Please help us by letting us know your ideas.

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

Contents Technical Manual TNC 426 B, TNC 430

1 Update Information No. 6

The following NC software has been released:

NC Software	Date of release		
280 472 05	11/97	Export-Version:	280 473 05

Improvements:

- On the TNC 426 PB without spindle DSP (from hardware version xxx xxx 4x) the maximum spindle speed was increased from 9000 rpm to 12 000 rpm.
- On the TNC 426 PB with spindle DSP and the TNC 430 PA (from hardware version xxx xxx 4x) the maximum spindle speed was increased from 15 000 rpm to 24 000 rpm.
- Module 9135 has been introduced:
  The infrared touch probe TS 630 can be switched on by the PLC. If the touch probe does not report readiness while M4056 is set, the feed rate enable is reset (previously: NC stop).
  Call:
  CM 9135

  M4203= 0: no error during module execution
  1: error during module execution
- MP3210.x extended:
  Input range (S analog voltage or motor revolutions) increased to 100.000
- D364 (nominal speed) and D368 (actual speed) have been added, since speeds greater than 32767 rpm cannot be represented in the words W320 (nominal speed) and W322 (actual speed).
- In the print masks of the touch probe cycles, it is now possible to distinguish between languages by means of code words. The text block defined in MP7230.0 is output.

Language code word	Dialog language
L_ENGLISH	0 = English
L_GERMAN	1 = German
L_CZECH	2 = Czech
L_FRENCH	3 = French
L_ITALIAN	4 = Italian
L_SPANISH	5 = Spanish
L_PORTUGUE	6 = Portuguese
L_SWEDISH	7 = Swedish
L_DANISH	8 = Danish
L_FINNISH	9 = Finnish
L_DUTCH	10 = Dutch
L_POLISH	11 = Polish
L_HUNGARIA	12 = Hungarian
L_ALL	Language-neutral texts

- Code number for LOGBOOK has been added:
  The data from the log can be transferred to an ASCII file. The user must specify a file name and the time at which the data transfer is to begin. The log contains the most recent keystrokes, control resets, error messages and the register status in case of blinking error messages.

- MP7471 has been added:
  Maximum speed of linear axes for compensating movements caused by the positioning of angular axes with M128.
- New machine parameters for new touch probe cycle (CALIBRATE TS):
  MP6180.0-2, MP6181.0-2 and MP6182.0-2: Approximate position of the ring gauge center (X, Y and Z in REF coordinates for three traverse ranges)
  Input: –99 999.9999 to +99 999.9999 [mm]
  MP6185: Distance below the upper edge of the ring gauge to be probed during calibration.
  Input: 0.001 to 99 999.9999 [mm]
- FN18:SYSREAD has been expanded:
  It is now possible to determine whether a datum table is selected in the current operating mode.
  ID505
  NR1
  0 = no datum table selected
  1 = datum table selected
- FN18: SYSREAD has been expanded:
  It is now possible to determine whether the addressed MP exists.
  ID1010
  NRxxxx MP number
  IDXxxxx MP index
  0 = MP does not exist
  1 = MP exists

NC software	**Date of release**		
280 472 06	12/97	Export version:	280 473 06

Improvements:

- New touch probe cycles. These cycles are defined like the fixed cycles via the TOUCH PROBE key and soft keys. All touch probe system functions are now described in a separate manual:
  User's Manual touch probe cycles – German 329203 10
  – English 329203 20
- Three new markers for workpiece measurement:

	Set	Reset
M4065: All dimensions of the workpiece are OK	NC	PLC
M4066: Workpiece needs rework	NC	PLC
M4067: Workpiece must be scrapped	NC	PLC

- Timers T96 to T143 have been added:
  The new timers can be started only through Module 9006. The timer is set immediately after module call and reset after expiration of the run time.
- FN17:SYSWRITE has been expanded:
  The touch probe monitoring can be switched on and off.
  ID990
  NR2 = numerical value
  Numerical value = 0 touch probe monitoring off
  Numerical value ≠ 1 touch probe monitoring on
- FN17:SYSWRITE has been expanded:
  The touch probe data of the manual probing cycles are transferred to the tool table.
  ID990
  NR3 = Qxxx or any numerical value

- FN17:SYSWRITE has been expanded:
  A point in the working plane (i.e., the plane perpendicular to the tool axis) of the workpiece coordinate system can be transformed into the corresponding plane of the machine coordinate system and vice versa, whereby the corresponding plane of the machine coordinate system is the plane whose normal vector has the designation of the tool axis.
  ID990
  NR4
  IDX 1 = Qxxx (Transformation of workpiece coordinate system to machine coordinate system)
  2 = Qxxx (Transformation of machine coordinate system to workpiece coordinate system)
  Qxxx Number of the first of four consecutive Q parameters
  1. Q parameter: Coordinate of the 1st axis of the point to be transformed
  2. Q parameter: Coordinate of the 2nd axis of the point to be transformed
  3. Q parameter: Coordinate of the 1st axis of the transformed point
  4. Q parameter: Coordinate of the 2nd axis of the transformed point

1 Update Information No. 5

1.1 NC Software

HEIDENHAIN has released a new NC software:

NC software	**Release**		
280 472 04	10/97	Export version:	280 473 04

Improvements:

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

1.2 Hardware

The maximum input frequency of the position encoder inputs X1 to X5 was reduced to 50 kHz for 1 V_{PP} signals. You will find the new ID numbers on page 3-10. Changeover date will be end of December 1997. We will still provide the old logic units with unchanged input frequency under the old ID numbers upon special request.

1.3 Documentation

Various changes were made to the Technical Manual. The list on the next 3 pages gives an insight into what changes were made, and where the information can be found.

1 Update Information No. 4

With Update Information No. 3 you received completely new pages for your TNC 426 B / TNC 430 Technical Manual. Since then the following changes have been made to the software.

1.1 NC Software 280 472

NC Software	Release		
280 472 01	6/97	Export version:	280 473 01

NC Software	Release		
280 472 02	7/97	Export version:	280 473 02

- MP6500 expanded:
  Bit 9 *reserved*
  Bit 10 probing routine (Bit 8 = 1)
  0 = The starting point is pre-positioned in all three principle axes.
  1 = The starting point is pre-positioned only in the tool axis and in the axis of the probing direction (MP6505).
  Bit 11 Checking the tool and adjusting the tool table
  0 = After "tool checking" the tool table is adjusted.
  1 = After "tool checking" the tool table is not adjusted.
- MP7500 expanded:
  Bit 3 Setting the datum in a tilted coordinate system
  0 = Datum setting is possible in tilted coordinate system.
  1 = During datum setting the current positions of the tilting axes are not offset.

NC Software	Release		
280 472 03	8/97	Export version:	280 473 03

- MP7680 expanded:
  Bit 10 see item 1.1.2 "Optimization of Tool-Radius-Compensated Outside Corners."
- The software also runs on the old hardware of the LE 426 CB/PB and LE 430 CA/PA, however with less feature content.

1.1.1 Nominal Position Value Filter

For optimum adjustment of the velocity and acceleration the nominal position values are filtered. This results in smoother (jerk-limited) traverse. The TNC calculates the filter parameters weighting and width (order) using the permissible axis-specific jerk and the tolerance. The filter is effective in all operating modes. For rigid tapping (Cycle 17) the nominal position value filter is automatically switched off.

With MP1095 you can select whether the TNC uses a single or double filter. The single filter causes a linear change in acceleration and therefore a step in the jerk.

With Cycle 32 the user can overwrite the tolerance defined in MP1096 for contour transitions. Cycle 32 was renamed to "fast contour milling" because the nominal position value filter is effective not only for 3-D contours.

MP1095	Nominal position value filter
Input:	0 = single filter
	1 = double filter
	Suggested input value = 0

MP1096	Tolerance
Input:	0 = no nominal position value filter
	0.001 to 3.000 [mm] = permissible tolerance at contour transitions

MP1097	Axis specific jerk for single filters (MP1096 = 0)
Input:	1 to 1 000 [m/s^3]
MP1097.0-8	Axis 1 to axis 9

MP1098	Axis specific jerk for double filters (MP1096 = 1)
Input:	1 to 1 000 [m/s^3]
	Suggested input value = 2 • MP1097.x
MP1098.0-8	Axis 1 to axis 9

MP1099	Minimum filtering order
Input:	0 to 20
MP1099.0	Minimum filtering order for single filters (MP1096 = 0)
	Suggested input value = 5
MP1099.1	Minimum filtering order for double filters (MP1096 = 1)
	Suggested input value = 3

Commissioning

- ➤ Determine minimum filtering order. Suggested input values: MP1099.0 = 5, MP1099.1 = 3
- ➤ Switch off the nominal position value filter (MP1096 = 0).
- ➤ Determine MP1090.x, MP1092, MP1510.x as described on page 4-318/4-319. Enter the optimum jerk values for each axis MP1097.x.
- ➤ In MP1098.x enter twice the value from MP1097.x.
- ➤ Define the tolerance in MP1096 (e.g. 0.02 mm)

1.1.2 Optimization of Tool-Radius-Compensated Outside Corners

With MP7680, bit 10 you set whether a circular arc or a spline should be inserted for the tool center path at tool-radius-compensated outside corners. Inserting a spline has the advantage of limiting the jerk at the corners and, when the nominal position value filter (MP1096 > 0) is active, of milling the corners more precisely.

MP7680	Machines parameters with multiple function
Input:	%xxxxxxxxxxx
Bit10	Tool-radius-compensated outside corners
	0 = Insert a circular arc
	1 = Insert a spline curve

Suggested input value = %1xxxxxxxxxx

1.1.3 New Backlash Compensation

A new type of backlash compensation is effective beginning with NC software 280 470 08 and 280 472 01. Unlike the backlash compensation possible with MP710, you can compensate the backlash in the entire controlled system with MP750 and MP752. This means that you can now also compensate play between the motion of the motor and the table with position measurement via linear encoders. This feature also compensates the reversal spikes resulting from circular traverse, and the machine parameters MP711 to MP716 are therefore no longer needed.

In MP750 you enter the backlash in mm. In MP752 you enter the time within which the compensated distance should be traversed.

Example:
MP750 = 0.03 mm, MP752 = 15 ms
For every reversal in axis direction, for 15 ms a nominal speed command signal is output corresponding to a feed rate of 120 mm/min (0.03 mm / 15 ms = 0.002 m/s = 120 mm/min).

MP750	Backlash
Input:	–1.0000 to +1.0000 [mm]
MP750.0-8	Axis 1 to axis 9

MP752	Compensation time for value from MP750.x
Input:	0 to 1000 [ms]
MP750.0-8	Axis 1 to axis 9

Commissioning

➤ Enter the following test program:

```
LBL 1
L X100 R0 F10
L X0
CALL LBL 1 REP 100/100
```

➤ With the internal oscilloscope, record ACTL.SPEED and V (ACT RPM)

➤ At the reversal point the actual feed rate lags behind the actual RPM with the time delay *t*.

➤ Input values: MP750 = t • ΔACTL.SPEED
MP752 = approx. 20 ms (optimum value determined empirically with this test)

1.1.4 Other Changes in the Technical Manual

- MP7460 (constant contouring speed at corners) has been replaced by MP1096 (tolerance), pages 4-77, 9-45.
- MP1091 (Jerk limiting for 3-D milling with Cycle 32) will not be introduced, pages 4-62, 4-78, 9-13.
- Module 9037 (read general axis information) was renamed to Module 9038, pages 4-7, 11-1.

1.2 NC Software 280 470

NC Software	**Release**		
280 470 08	5/97	Export version:	280 471 08

- Hungarian dialog added
- D760 (Offset for tilting axes, probe center offset) added
- MP750 and MP752 (backlash compensation) added
- MP3143 expanded:
  3 = same as input value 1, except that the second reference mark is evaluated first.

NC Software	**Release**		
280 470 09	6/97	Export version:	280 471 09

- MP6500 expanded:
  Bit 9 *reserved*
  Bit 10 probing routine (bit 8 = 1)
  0 = The starting point is pre-positioned in all three principle axes.
  1 = The starting point is pre-positioned only in the tool axis and the axis for the probing direction (MP6505).
  Bit 11 Checking the tool and adjusting the tool table
  0 = After "tool checking" the tool table is adjusted.
  1 = After "tool checking" the tool table is not adjusted.

NC Software	**Release**		
280 470 10	7/97	Export version:	280 471 10

1 Update Information No. 3

HEIDENHAIN released the new NC software 280 472 01 in June 1997. This software will only run on logic units with Id. Nr. xxx xxx 4x.

The new functions (listed below) and the changes in hardware are described in detail in the new Technical Manual (English version not yet available).

Conditions of delivery
As of June 1997 HEIDENHAIN will only supply the new hardware (Id. Nr. xxx xxx 4x). The NC software 280 480 runs on this new hardware as of version 06. For the present the new software (Id. Nr. 280 472 xx) will be supplied only where expressly ordered. As of September 1997 the old NC software will no longer be delivered.

NC software	**Release**		
280 472 01	6/97	Export version:	280 473 01

Improvements	**Technical Manual**	**User's Manual**
New function "3-D milling": Cycle 32 or G62 and MP1091.x	4–78	213
Automatic calculation of cutting data	4–255	92
TCPM (Tool Center Point Management): With M128 you can superimpose manual axis traverse during program run. Misalignments in the tilting axes are then automatically compensated.	–	151
Additional information with HELP key	4–154	64
Input menu for fixed input values can be selected with the GOTO key (e.g. baud rate)	–	326
New pallet management	4–174	65
Freely-definable tables	4–178	95
You can select between standard and enhanced file management with the MOD function PGM MGT	–	330
The positions of all nine axes are shown in the status display. The spindle position overwrites the ninth axis	–	5
Copying progress is shown in a superimposed window	–	35
The number of Q parameters was increased from 299 to 399	–	250
Q parameters also permissible in FK blocks	–	125
M110 also effective in contour pocket cycle	–	146
Cycle 204: Counterbore back	–	163
With MP7682 Bit 2 you can set whether rotary axes should always be positioned by the shorter path	4–124	148
A chamfer feed rate can be entered in the NC block "Chamfer" (CHF)	–	109

Improvements	**Technical Manual**	**User's Manual**
Cycle 19 "Machining plane" was expanded by the parameters feed rate and setup clearance (only when the tilting axes are positioned with Cycle 19. This is set in MP7500)	4–29	227
M114 can also be used with non-controlled or PLC axes	4–34	150
Hungarian as additional conversational language	4–189	344
All soft keys appear in the set conversational language	–	–
Language-dependent soft keys for OEM cycles	–	–
The soft keys for FK programming only appear once you press the FK key	–	126
Soft key F for feed rate in the manual operating modes	4–127	17
New soft key: INCREMENT OFF/ON	4–242	17
New soft key: HIDE TOOLS OFF/ON: the only tools shown in the tool table are those in the tool magazine	–	74
New soft keys for copying fields in the tool table	–	74
PLC soft keys can be added to NC soft-key rows	4–151	–
Ethernet: the name of a network printer can be given	6–40	–
The probe results of the manual probing function can be taken over immediately in the datum tables	–	293
MP6170, MP6171: Multiple measurement with measuring tolerance	4–207	305
A separate block of calibration data for TS and TT for every traverse range	4–206	–
With MP6500 Bit 4 you can set whether or not speed should be limited to 1000 rpm during tool measurement with TT	4–227	–
MP6500 Bit 9: Automatic determination of the basic rotation for cubical probe contacts	4–227	–
W760: Angular offset of the tilting axes for automatic adjustment of the touch probe center offset	4–207	–
The TS calibration data can be stored in the tool table using soft key	–	297
Cycles 31 to 33 (Tool measurement) were expanded by the input field "Q parameters for result"	–	87
With MP6500 Bit 5 and Bit 6 you can define how to proceed if tool breaks	4–227	–
FN17, FN18 ID990 NR1 Programmed probing	5–13	–
FN17: ID210 NR6 Tilting the working plane during program run active/inactive	5–13	–
FN18 ID350 Enhanced touch probe data	5–20	263

Improvements	**Technical Manual**	**User's Manual**
FN23: CDATA Calculating the circle center from 3 touch points	–	255
FN24: CDATA Calculating the circle center from 4 touch points	–	255
FN25: Setting the datum	4–117	–
ISO: Cycles with a number ≥ 200 can be programmed with graphic support (also OEM cycles)	–	–
ISO: Cycles G75 and G76 (Rectangular pocket) were expanded by the input field "corner radius"	–	–
ISO: Parameter H (max. permissible angle) can be entered after M112	–	–
ISO: G60 Running digitized data is new	–	–
MP2000 was removed. Digital axes can be defined in MP120	4–6	–
In the compensation value tables COM and CMA you can use soft keys to select the columns for the desired axes	4–17	–
Non-linear axis error compensation: Max. number of compensation points increased from 640 to 1280	4–16	–
A formula can be input in MP2020 (Distance covered in one motor revolution)	4–81	–
MP2541, MP2551: Frequency filter for spindle	4–114	–
The number of tools in the tool table was increased from 254 to 65 535	4–248	71
M4014: Reversing the count direction of the linear encoder for the spindle	4–97	–
Cooperative multi-tasking in the PLC (SPAWN command)	5–156	–
Module 9019: Check program storage	5–34	–
Oscilloscope recordings can be stored in a file.	4–305	–

If you are using OEM cycles, you need to create a new cycle structure with CycleDesign to be able to program the new HEIDENHAIN cycles 32 and 204. You will need the CycleDesign software version 1.21

New hardware

Since February 1997 HEIDENHAIN has been delivering a new hardware for the LE 426 B and LE 430.

The advantages of the new hardware:

- 3-row VGA connector for BC 120. With the new connecting cable Id. Nr. 312 878 .. there is no longer need for an adapter connector
- Internal working memory doubled (4 MB)
- More memory space available on larger hard disk (1.5 GB)
- **LE 426 PB and LE 430 PA:** Three current controllers. Maximum speed = value from Siemens data sheet

The NC software 280 470 runs on this new hardware as of version 06.

The full benefit of its new range of features can only be seen however if the new hardware is used together with the new NC software 280 472 as of version 01.

LE 426 PB with digital spindle to 15 000 rpm

As an option the LE 426 PB is supplied for digital spindles with up to 15 000 rpm. See pages 2–10 and 3–10 in the new Technical Manual.

2 Introduction

2.1 Integrated Current Control

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

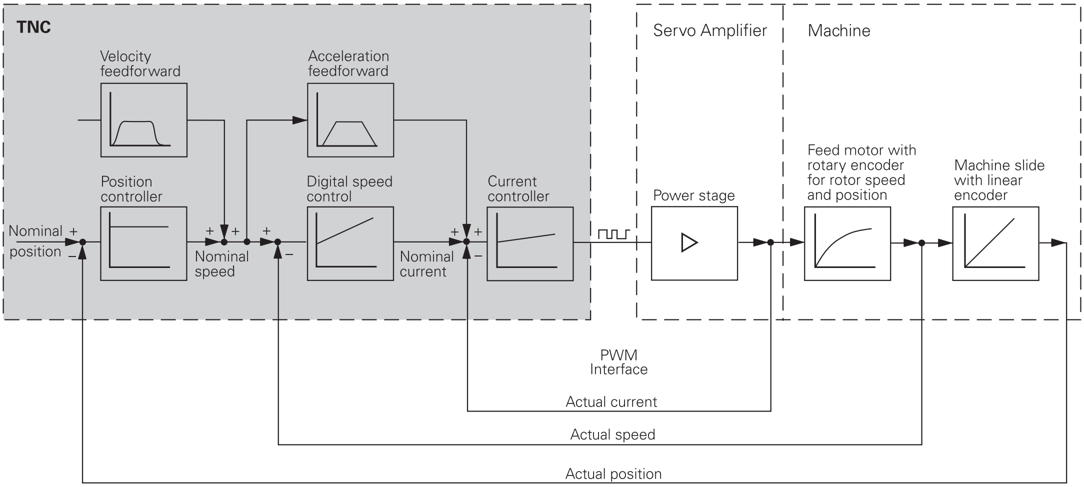
The **TNC 426 PB / TNC 430 PA** has integral drive controllers; these control the drive amplifier by means of Pulse Width Modulated (PWM) signals.
Integration of the drive controllers in the TNC 426 PB / 430 PA offers the following **advantages:**

- 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 control, speed control and current control are combined into one unit.
- For commissioning, optimization and diagnosis, the same functions are available both for the feed drive and the main spindle.

The **TNC 426 PB** can control machines with up to five axes and spindle speeds up to 9000 rpm, option: 15 000 rpm (spindle speeds are valid for motors with two pole pairs). Powerful microprocessor hardware and an integrated hard disk with 900 MB guarantee almost unlimited NC memory and a short processing time — even for long programs. These are ideal preconditions for tool and mold construction.

The **TNC 430 PA** supports six digitally controlled NC axes and three analog controlled secondary axes, as well as digitally controlled spindle speeds up to 15 000 rpm (for motors with two pole pairs).

Control concept of the TNC 4xx PA



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 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 Brief Description

Technical Data	TNC 426	TNC 430
Basic model with integral drive control	All position encoder and motor encoder inputs 1 V_{PP}	
	TNC 426 PB: 5 axes plus spindle (max. 9000 rpm, option: 15 000 rpm) with position encoder and motor encoder inputs.	**TNC 430 PA:** • 5 axes plus spindle (max. 15 000 rpm) with position encoder and motor encoder inputs • 6th axis with motor encoder input
Basic model with analog speed command interface	All position inputs 1 V_{PP}	
	TNC 426 CB: 5 axes plus spindle	**TNC 430 CA:** 8 axes plus spindle
Options	Position inputs for the axes: 11 µA_{PP}	–
	–	**TNC 430 CA:** Position input for 9th axis **TNC 430 PA:** • Position inputs for 3 additional axes with analog speed command interface • Position input for 6th axis
	• Digitizing with 3-D triggering touch probe • Digitizing with 3-D measuring touch probe • Ethernet interface	
Display	15-inch CRT color monitor	
Program memory	Hard disk with 900 MB	
Input precision 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
Circular arcs	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	Combination of circular and linear motion	
Spline	–	Cubical splines can be input
Block processing time	From the hard disk: 4 ms	

Machine Integration	TNC 426	TNC 430
Control		
Integral drive control	**TNC 426 PB**	**TNC 430 PA**
Analog speed command interface	**TNC 426 CB**	**TNC 430 CA**
Position loop resolution	$\frac{\text{Signal period}}{1024}$	
Cycle time: path interpolation	3 ms	
Cycle time: speed	**TNC 426 PB, TNC 430 PA:** 0.6 ms	
Feed rate	**TNC 426 PB, TNC 430 PA:** $\frac{18\,000}{\text{No. pole pairs}} \bullet \text{Screw pitch} \frac{1}{\text{min}}$ **TNC 426 CB, TNC 430 CA:** Up to 60 $\frac{\text{m}}{\text{min}}$ for encoders with 20 µm grating period Up to 300 $\frac{\text{m}}{\text{min}}$ for encoders with 100 µm grating period	
Speed	**TNC 426 PB (standard):** $\frac{18\,000}{\text{No. pole pairs}} \frac{1}{\text{min}}$ **TNC 426 PB (option), TNC 430 PA:** $\frac{30\,000}{\text{No. pole pairs}} \frac{1}{\text{min}}$ **TNC 426 CB, TNC 430 CA:** 100 000 $\frac{1}{\text{min}}$	
Error compensation	Linear / non-linear axis error, backlash, reversal spikes during circular movements, offset, thermal expansion, stiction, sliding friction	
Integrated PLC		
PLC memory	Hard disk	
Main memory (RAM)	128 KB (approx. 16 000 commands)	
PLC cycle time	21 ms	
PLC inputs 24 Vdc	56 (additional inputs as option: see PL 410 B)	
PLC outputs 24 Vdc	31 (additional outputs as option: see PL 410 B)	
Analog inputs ±10 V	3 (additional analog inputs as option: see PL 410 B)	
Analog outputs ±10 V	**TNC 426 PB:** 13 **TNC 430 CB:** 7 (with 5 NC axes + spindle)	**TNC 430 PA:** 13 **TNC 430 CA:** 3 (with 9 NC axes + spindle)
Inputs for thermistors	3 (additional inputs as option: see PL 410 B)	
Commissioning aids	• Oscilloscope • Trace function	
Data interfaces	• One each RS-232-C/ V.24 and RS-422/ V.11 up to 115 kbaud • Expanded data interface with LSV2 protocol for external operation of the TNC • Option: Ethernet interface, approx. 200 to 1000 kilobaud	

User Functions	**TNC 426**	**TNC 430**
Programming	HEIDENHAIN plain language and ISO	
Position data	Nominal positions for straight lines and circular arcs in Cartesian or polar coordinates, absolute or incremental dimensional data, display and entry in mm and inches	
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 and tool length • Radius-compensated contour look ahead for up to 99 blocks (M120)	
Cutting data tables	For the automatic calculation of speed and feed rate from various definable cutting/workpiece material combinations	
Constant contouring speed	• Relative to the tool's center point path • Relative to the tool's cutting edge (M109, M110, M111)	
3-D machining	• Insertion of rounding radius between two straight line segments (M112, M113, M124) • Feed rate reduction during plunging (M103) • 3-D tool compensation through surface normal vectors • Automatic compensation of machine geometry when working with tilted axes (M114, M115, M128, M130) • Changing the position of the swivel head with the electronic handwheel during program run. The position of the tool tip does not change. • Jolt reduction • Spline	
Machining with rotary tables	• Programming of cylindrical contours • Feed rate in mm/min (M116)	
FK free contour programming	FK free contour programming in HEIDENHAIN plain language with graphic support for workpiece drawings not dimensioned for NC	
Subprogramming	Program section repeats, subprograms, program calls	
Background programming	Creating programs while another program is being run, also with graphic support	
Fixed cycles	• Peck drilling, tapping with or without a floating tap holder, reaming, boring, hole patterns, slot milling, rectangular and circular pocket milling, multipass 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	
Coordinate transformation	• Datum shift, rotation, mirror image • Scaling factor (axis specific) • Tilting the working plane	

User Functions	TNC 426	TNC 430
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	
Teach in	Actual positions can be transferred directly into the NC program	
Test graphics	Graphic simulation before a program run: • Plan view • Projection in three planes • 3-D representation • 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 representation	
Machining time	• Calculation of machining time in the Test Run operating mode • Display of the current machining time in the program run modes	
Returning to the contour	• Mid-program startup in any block in the program, returning the tool to the calculated nominal position to continue machining • Program interruption, contour departure and reapproach	
Datum tables	Several datum tables, each with 254 datums	
Pallet tables	Several pallet tables with various different entries for selection of pallets, NC programs and datums	

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

Accessories	TNC 426	TNC 430
Electronic handwheels	One **HR 410** portable handwheel, or one **HR 130** panel-mounted handwheel, or up to 3 **HR 150** panel-mounted handwheels via HRA 110 handwheel adapter	
	Superimpose handwheel positioning during program run (M118)	
Touch probe systems	**TS 220** 3-D triggering touch probe with cable connection, or **TS 630** 3-D triggering touch probe with infrared transmission **TT 120** 3-D triggering touch probe for tool measurement and inspection	
Digitizing of 3-D surfaces	• Option: with the **TS 220** 3-D triggering touch probe and a software module for the TNC • Option: adapter kit for a 3-D measuring touch probe. • PC evaluation software for digitized data: **SUSA**	
PC evaluation software for digitized data	**SUSA** for PCs	
Interface card	**TNC 426 PB, TNC 430 PA:** For connection of power modules for the SIMODRIVE 611 inverter system	
Protective PCB for dc-link power supply	Protective PCB for integration in the SIEMENS input module to secure the dc-link power supply of the logic unit.	
PLC input/output unit	Up to four **PL 410 B**	
	Version 1: Additional 64 PLC inputs and 31 PLC outputs per input/output unit **Version 2:** Additional 64 PLC inputs and 31 PLC outputs, plus four ± 10 V analog inputs and 4 inputs for thermistors per input/output unit	

2.3 Software

The logic unit contains separate software for the NC section and the PLC section. The software is identified by an eight-figure number.

After switching on the control, the NC software, PLC software and software-options numbers are displayed on the screen. The software number can also be directly requested with the aid of the MOD function.

2.3.1 NC Software

NC software number
The first 6 figures of the NC software number identify the type of control, the last two identify the version of the software.

Software type
Due to restrictions on the export of the TNC, HEIDENHAIN can also deliver a special export version. This export version differs from the standard control through the installed software type.
HEIDENHAIN assigns a new software type whenever comprehensive new functions are introduced.

	Software type		**Linear interpolation**
TNC 426 PB, TNC 426 CB TNC 430 PA, TNC 426 CA	280 470	280 472	Up to 5 axes
TNC 426 PF, TNC 426 CF TNC 430 PE, TNC 426 CE	280 471	280 473	Up to 4 axes

2.3.2 Software Option

HEIDENHAIN offers “Digitizing with a Triggering Touch Probe” and “Digitizing with Triggering and Measuring Touch Probes” as software options (see chapter “Machine Integration”). Whenever a contouring control is ordered with one of these options, HEIDENHAIN installs an additional software module or PCB in the logic unit and assigns another variant to the part identification number (Id. Nr.) of the logic unit. The option number is displayed on the screen in addition to the NC and PLC software numbers.

Logic units already in the field can be retrofitted by the end user with the digitizing software module. Please contact HEIDENHAIN for more information.

Option	**Option No.**	**Id. Nr. of the component set**	**Id. Nr. of the software module**
Digitizing with triggering touch probe	1	286 405 01	246 051 01
Digitizing with measuring and triggering touch probes			
SP 2/1	11	311 647 51	–

2.3.3 PLC Software

The PLC software is stored on the hard disk of the TNC. HEIDENHAIN has developed a PLC commissioning program for the TNC. The source code is available from HEIDENHAIN. This program can be easily adapted to suit your machine with the **PLC programming software PLCdesign.**

2.3.4 Software Exchange

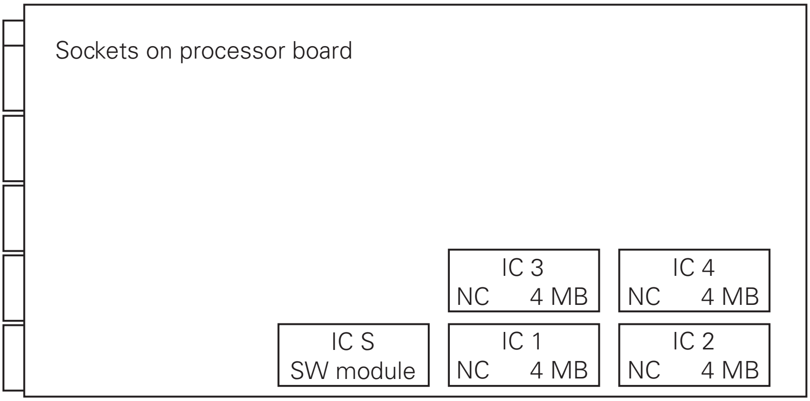
The NC software is located on EPROMs (the sockets are illustrated below). The dialog languages are stored on the hard disk. If there are no up-to-date dialog languages on the hard disk, load the English dialog language from the EPROMs (selectable with MP7230.x). If a software exchange becomes necessary, HEIDENHAIN will provide the EPROMs with the NC software and a floppy disk with all dialog languages.

Procedure for software exchange:
The software must be exchanged only by trained personnel.

⇒ Enter the code number 95148
⇒ Press the MOD key
⇒ Press the UPDATE DATA and CONVERT BIN=>ASC soft key:
All files on the hard disk will be converted from binary to ASCII format. The free space on the hard disk must be at least 50% larger than the largest file. If this is not the case you must save this file through the data interface. The extensions of the binary files and of the corresponding converted ASCII files are:

.H	⇔	.H%	.I	⇔	.I%	.T	⇔	.T%
.TCH	⇔	.TC%	.D	⇔	.D%	.P	⇔	.P%
.PNT	⇔	.PN%	.COM	⇔	.CO%	.CMA	⇔	.CM%

⇒ Exchange the EPROMs



Danger of electrical shock!
Switch off the main switch before opening the housing.

⇒ Edit or erase the machine parameters. You will find information on the machine parameters in the MPDOC.A file on the supplied floppy disk.
⇒ Press the END key to exit the machine parameter editor. The error message `LANGUAGE LOAD ERROR` appears.
⇒ On the PC, enter the SETUP command to load the NC dialogs, HEIDENHAIN cycles etc. from the provided floppy disk. The floppy disk also contains a detailed description in the README.TXT file.
⇒ Press the UPDATE DATA and CONVERT ASC=>BIN soft key:
All files on the hard disk are converted from ASCII into binary format.
⇒ Reload the files that you have backed up through the data interface.
⇒ Switch the TNC off and on to activate the new NC dialogs.

2.3.5 Data Backup

HEIDENHAIN provides a free program, TNCBACK.EXE, for backing up files in the TNC 426. We recommend to the manufacturer whenever he supplies a machine tool to also provide a floppy disk containing a copy of all **machine-specific data,** backed up with **TNCBACK.EXE.** The floppy disk must also contain a copy of TNCBACK.EXE.

Before exchanging his control unit, the customer can save the data from the TNC, especially the TNC:\ partition with its directories containing the part programs (see the user's instructions on the floppy disk).

2.4 Hardware

The eight-digit ID number of the logic unit consists of the 6-digit basic ID number followed by the 2-digit version number. The basic ID number designates significant hardware differences (e.g. type of logic unit encoder inputs). The version number identifies the following differences:

version xy:
x = Identifier for a hardware change
y = 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 Probes” option
8 = Export version without option
9 = Standard version without option

2.4.1 ID Numbers

		BC 120	BF 120
LE 426 CB			
5 position inputs 1 spindle position	**1 V_{PP} (350 kHz)** 1 V_{PP} (350 kHz)	Id. Nr. 312 001 ..	Id. Nr. 313 524 ..
5 position inputs 1 spindle position	**1 V_{PP} (50 kHz)** 1 V_{PP} (350 kHz)	Id. Nr. 326 415 ..	Id. Nr. 326 419 ..
5 position inputs 1 spindle position	**11 μA_{PP} (50 kHz)** 1 V_{PP} (350 kHz)	Id. Nr. 312 002 ..	Id. Nr. 313 525 ..
LE 426 PB			
5 position inputs 1 spindle position 6 motor encoder inputs	**1 V_{PP} (350 kHz)** 1 V_{PP} (350 kHz) 1 V_{PP}		
Spindle with up to 9000 rpm		Id. Nr. 312 000 ..	Id. Nr. 313 527 ..
Spindle with up to 15 000 rpm		Id. Nr. 315 475 ..	Id. Nr. 318 178 ..
5 position inputs 1 spindle position 6 motor encoder inputs	**1 V_{PP} (50 kHz)** 1 V_{PP} (350 kHz) 1 V_{PP}		
Spindle with up to 9000 rpm		Id. Nr. 326 414 ..	Id. Nr. 326 421 ..
Spindle with up to 15 000 rpm		Id. Nr. 326 416 ..	Id. Nr. 326 420 ..
5 position inputs 1 spindle position 6 motor encoder inputs	**11 μA_{PP} (50 kHz)** 1 V_{PP} (350 kHz) 1 V_{PP}		
Spindle with up to 9000 rpm		Id. Nr. 311 999 ..	Id. Nr. 313 526 ..
Spindle with up to 15 000 rpm		Id. Nr. 317 349 ..	Id. Nr. 318 177 ..
LE 430 CA			
8 position inputs 1 spindle position	1 V_{PP} **(350 kHz)** 1 V_{PP} (350 kHz)	Id. Nr. 311 050 ..	Id. Nr. 313 523 ..
5 position inputs 3 position inputs 1 spindle position	1 V_{PP} **(50 kHz)** 1 V_{PP} **(350 kHz)** 1 V_{PP} (350 kHz)	Id. Nr. 326 418 ..	Id. Nr. 326 424 ..
LE 430 PA			
5 position inputs 1 spindle position 7 motor encoder inputs	1 V_{PP} **(350 kHz)** 1 V_{PP} (350 kHz) 1 V_{PP}	Id. Nr. 311 049 ..	Id. Nr. 313 521 ..
5 position inputs 1 spindle position 7 motor encoder inputs	1 V_{PP} **(50 kHz)** 1 V_{PP} (350 kHz) 1 V_{PP}	Id. Nr. 326 417 ..	Id. Nr. 325 716 ..

Version	Change
xxx xxx 2x	First issue
xxx xxx 3x	More powerful inverter
xxx xxx 4x	4 MB RAM; 3-row VGA connection; 3-phase current controller

TE 401 B	
Suitable for BC 110 B (black)	Id. Nr. 250 517 05
TE 420	
Suitable for BC 120 / BF 120 (gray)	Id. Nr. 313 038 10
MB 420	
Suitable for BC 120 / BF 120 (gray)	Id. Nr. 293 757 12

BC 110 B (black)	
14-inch screen with color graphics (640 x 490 pixels)	Id. Nr. 260 520 01
2-row/3-row adapter for connection to LE with Id. Nr. xxx xxx 4x via connecting cable Id. Nr. 312 878 ..	Id. Nr. 313 434 01
2-row/3-row adapter for connection to LE with Id. Nr. xxx xxx 4x via connecting cable Id. Nr. 250 477 ..	Id. Nr. 324 862 01
BC 120 (gray)	
15-inch screen with color graphics (640 x 490 pixels)	Id. Nr. 313 037 01
3-row/2-row adapter for connection to LE with Id. Nr. xxx xxx 3x	Id. Nr. 313 434 02
BF 120	
TFT color flat-panel display	Id. Nr. 313 506 10

PL 410 B	
64 switching inputs 24 Vdc 31 switching outputs 24 Vdc	Id. Nr. 263 217 12
64 switching inputs 24 Vdc 31 switching outputs 24 Vdc 4 analog inputs ±10 V 4 inputs for thermistors	Id. Nr. 263 217 02

Options	
Interface card for connection of power modules for the SIMODRIVE 611 inverter system	Id. Nr. 291 070 01
Protective PCB for dc-link power supply	Id. Nr. 296 965 01
Additional position input for a 6th/9th axis (350 kHz)	Id. Nr. 311 537 51
Position inputs for 3 additional axes with nominal speed command interface (350 kHz)	Id. Nr. 294 130 51
Ethernet interface	Id. Nr. 293 890 51
Digitizing with triggering 3-D touch probe	Id. Nr. 286 405 01
Digitizing with measuring 3-D touch probe (SP 2/1)	Id. Nr. 311 647 51

Adapter for encoder signals	
TTL (HEIDENHAIN layout) / 1 V_{PP}	Id. Nr. 317 505 01
TTL (SIEMENS layout) / 1 V_{PP}	Id. Nr. 317 505 02
11 μA_{PP} / 1 V_{PP}	Id. Nr. 313 119 01

2.5 Release Dates

2.5.1 NC Software Versions 280 470 and 280 471

NC software	**Release date**		
280 470 01	05/96	Export version:	280 471 01

First release:

NC software	**Release date**		
280 470 02	06/96	Export version:	280 471 02

Improvements:

- M132 with TIME parameter
- Module 9035 Parameter 21: Control model
- M118, M120 also in ISO
- Cycle 27: Cylinder surface also in tilted working plane
- MP7680 Bit 9 was added
- MP2423, MP2425, MP2427, MP2433, MP2451, MP2451, MP7245, MP7250 were removed
- MP2402 was changed: Gain for axis current controller at maximum speed

NC software	**Release date**		
280 470 03	08/96	Export version:	280 471 03

Improvements:

- **`GROSS POSITIONING ERROR F`** is new

NC software	**Release date**		
280 470 04	09/96	Export version:	280 471 04

Improvements:

- MP6500 Bit 4 and Bit 5 were added
- FN18: Group numbers 350 and 500 are new
- FN17: Group number 500 is new
- A Q-parameter was added to the cycles for tool measurement (31 to 33), where the measurement result will be stored

NC software	**Release date**		
280 470 05	12/96	Export version:	280 471 05

Improvements:

- Rotary axes can be synchronized axes
- MP7682 Bit 1 was added
- In the compensation value tables .CMA and .COM the numbers of the axes will be entered as opposed to the names
- Threshold for `PLC: Time Out` was increased from 200% to 300%
- MP6500 Bit 5, Bit 6 and Bit 8 were added
- FN18: Group numbers 51 and 52 are new
- FN17: Group number 210 is new
- Input range for MP2500 and MP2501 increased to 1000
- The maximum number of points for all compensation value tables was increased to 1280
- Coded NC error messages are displayed in the conversational format

NC software	Release date		
280 470 06	2/97	Export version:	280 471 07

Improvements:

- Datums can be set using 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 version of the LE 426 PB with spindle speeds up to 15 000 rpm
- The NC software also runs on the new LE 426 B and LE 430 hardware with the Id. Nr. xxx xxx 4x

NC software	Release date		
280 470 07	03/97	Export version:	280 471 07

Improvements:

- MP2541 and MP2551 (frequency filter for spindle) are new. Input as with MP2540 and MP2551 for the axes.
- The compensation of reversal peaks during circular traverse was improved (MP711.x to MP716.x).

NC software	Release date		
280 470 08	5/97	Export version:	280 471 08

Improvements:

- Hungarian conversational language new
- D760 (offset for tilting axes, touch probe center offset) new
- MP750 and MP752 (compensation of reversal error) new
- MP3143 expanded:
  3 = as for input value 1, however the second reference mark is evaluated first

NC software	Release date		
280 470 09	6/97	Export version:	280 471 09

Improvements:

- MP6500 expanded: bit 10 and bit 11

NC software	Release date		
280 470 10	7/97	Export version:	280 471 10

2.5.2 NC software versions 280 472 and 280 473

NC software	**Release date**		
280 472 01	4/97	Export version:	280 473 01

Improvements:

- New function "Fast contour milling": Cycle 32 or G62 and MP1091.x
- Automatic calculation of cutting data
- TCPM (Tool Center Point Management): With M128 you can superimpose manual axis traverse during machine run. Any misalignments in tilted axes are then corrected automatically.
- Additional information with the HELP key
- Input menu for fixed input values can be selected with the GOTO key (e.g. baud rate)
- New pallet management
- Freely-definable tables
- NC blocks can be transferred in spline format
- More hard-disk memory space (1.5 gigabytes)
- You can select between standard and enhanced file management with the MOD function PGM MGT
- The positions of all nine axes are shown in the status display. The spindle position overwrites the ninth axis
- Copying progress is shown in a superimposed window
- The number of Q parameters was increased from 299 to 399
- Q parameters also permissible in FK blocks
- M110 also effective in contour pocket cycle
- Cycle 204: Back boring
- With MP7682 Bit 2 you can set whether rotary axes should always be positioned by the shorter path
- A chamfer feed rate can be entered in the NC block "Chamfer" (CHF)
- Cycle 19 "Machining plane" was expanded by the parameters feed rate and setup clearance (only when the tilting axes are positioned with Cycle 19. This is set in MP7500)
- M114 can also be used with non-controlled or PLC axes
- Hungarian as additional conversational language
- All soft keys appear in the set conversational language
- Language-dependent soft keys for OEM cycles
- The soft keys for FK programming only appear once you press the FK key
- Soft key F for feed rate in the manual operating modes
- New soft key: INCREMENT OFF/ON
- New soft key: HIDE TOOLS OFF/ON: the only tools shown in the tool table are those in the tool magazine
- New soft keys for copying fields in the tool table
- PLC soft keys can be added to NC soft-key rows
- Ethernet: the name of a network printer can be given
- The probe results of the manual probing function can be taken over immediately in the datum tables
- MP6170, MP6171: Multiple measurement with measuring tolerance
- A separate block of calibration data for TS and TT for every traverse range
- With MP6500 Bit 4 you can set whether or not speed should be limited to 1000 rpm during tool measurement with TT
- MP6500 Bit 9: Automatic determination of the basic rotation for the cubical probe contact
- W760: Angular misalignment of the tilting axes for automatic adjustment of the touch probe center misalignment
- The TS calibration data can be stored in the tool table via soft key

- Cycles 31 to 33 (Tool measurement) were expanded by the input field "Q parameters for result"
- With MP6500 Bit 5 and Bit 6 you can define how to proceed if tool breaks
- FN17, FN18 ID990 NR1 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 Enhanced touch probe data
- FN23: CDATA Calculating the circle center from 3 touch points
- FN24: CDATA Calculating the circle center from 4 touch points
- FN25: Setting the datum
- ISO: Cycles with a number ≥ 200 can be programmed with graphic support (also OEM cycles)
- ISO: Cycles G75 and G76 (Rectangular pocket) were expanded by the input field "corner radius"
- ISO: Parameter H (max. permissible angle) can be entered after M112
- ISO: G60 Running digitized data is new
- MP2000 was removed. Digital axes can be defined in MP120
- In the compensation value tables COM and CMA you can use soft keys to select the columns for the desired axes
- Non-linear axis error compensation: Max. number of compensation points increased from 640 to 1280
- A formula can be input in MP2020 (Distance covered in one motor revolution)
- MP2541, MP2551: Frequency filter for spindle
- The number of tools in the tool table was increased from 254 to 37 767
- M4019: Reversing the count direction of the linear encoder on the spindle
- Cooperative multi-tasking in the PLC (SPAWN command)
- Automatic tool recognition (BIS)
- String operand S#Axx new
- Module 9019: Checking program storage
- Module 9035: Expansion of parameters 3, 1000, 1001
- Module 9038: Reading axis information
- Module 9096: Deleting a line in the tool table
- Module 9112: Sending ASCII characters via RS-232
- Module 9113: Receiving ASCII characters via RS-232
- Module 9151: Selecting traverse range and axis designation
- Module 9200/9201: Expanded (PLC soft keys can be added to NC soft-key rows)
- Module 9215: Superimposing PLC window
- Module 9270: Reading from OEM.SYS
- Module 9271: Writing to OEM.SYS
- Automatic offset compensation of the encoder signals
- The oscilloscope recordings can be stored in a file.
- MP7365.5: Selected oscilloscope channel (input $00000FF)

NC software	**Release date**		
280 472 02	7/97	Export version:	280 473 02

Improvements:

- Cycle 32 changed to "Tolerance"
- M134 new
- System file TNC.SYS new
- MP6500 expanded: Bit 10 Probing routine, Bit 11 Checking tool 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	**Release date**		
280 472 03	8/97	Export version:	280 473 03

Improvements:

- Spline blocks also in tilted working plane
- MP7680, bit 10 new (spline at compensated outside corners)
- This software also runs on the old LE 426 CB/PB and LE 430 CA/PA hardware, however with reduced function range
- Cycle 19: Dialog box "Setup clearance" new

NC software	**Release date**		
280 472 04	10/97	Export version:	280 473 04

Improvements:

- Code words LSV2TIME0 to LSV2TIME2 new
- Module 9038 expanded by transfer parameter –1
- DR2 can now be defined in TOOL CALL block
- Spindle DSP limits maximum torque to 2.5 • rated torque

3 Mounting and Electrical Installation

3.1 Electrical Noise Immunity

Location for use
This device corresponds to Class A according to EN 55022 and is intended primarily for operation in industrially zoned areas.

Remember that the vulnerability of electronic equipment to noise increases with faster signal processing and higher sensitivity. Protect your equipment by observing the following rules and recommendations.

Noise voltages are mainly produced and transmitted by capacitive and inductive coupling. Electrical noise can be picked up by the inputs and outputs to the equipment, and the cabling.

Likely sources of interference are:

- Strong magnetic fields from transformers and electric motors
- Relays, contactors and solenoid valves
- High-frequency equipment, pulse equipment and stray magnetic fields from switch-mode power supplies
- Mains leads and leads to the above equipment

Electrical interference can be avoided by:

- A minimum distance between the logic unit (and its leads) and interfering equipment > 20 cm.
- A minimum distance between the logic unit (and its leads) and cables carrying interference signals > 10 cm. (Where signal cables and cables that carry interference signals are laid together in metallic ducting, adequate decoupling can be achieved by using a grounded separation shield.)
- Shielding according to IEC 742 EN 50 178.
- Potential compensating lines dia. ≥ 6 mm^2 (see Grounding Diagram).
- Use of original HEIDENHAIN cables, connectors and couplings.

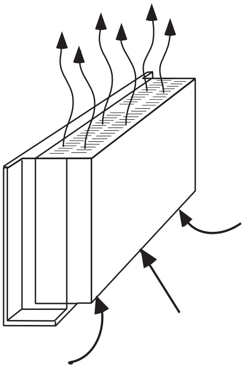
3.2 Heat Generation and Cooling

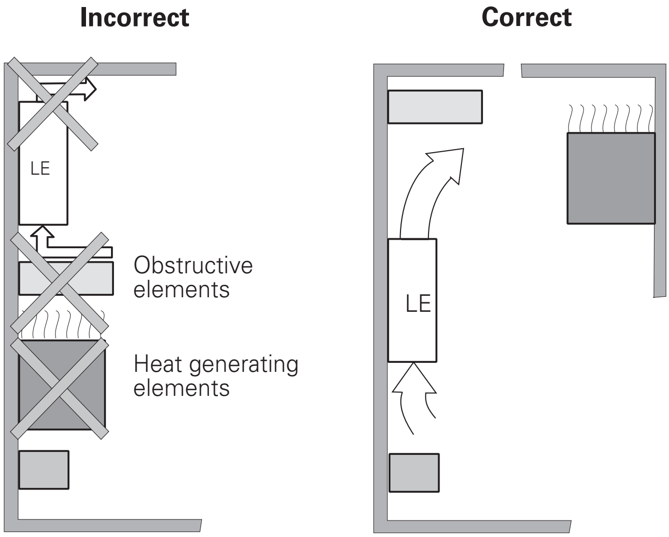
Please note that the reliability of electronic equipment is greatly reduced by continuous operation at high temperatures. Be sure to make the necessary arrangements to keep within the permissible ambient temperature range.

Permissible ambient temperature in operation: **0° C to 45° C**

The following means may be employed to ensure adequate heat removal:

- Provide sufficient space for air circulation.
- Build in a ventilator fan to circulate the air inside the control cabinet. The fan must reinforce the natural convection. It must be mounted so that the warm air is extracted from the logic unit and no pre-warmed air is blown into the unit. The warmed-up air should flow over surfaces that have good thermal conductivity to the external surroundings (e.g. sheet metal).
- For a closed steel housing without assisted cooling, the figure for heat conduction is 3 watt/m² of surface per °C air temperature difference between inside and outside.
- Use of a heat exchanger with separate internal and external circulation.
- Cooling by blowing external air through the control cabinet to replace the internal air. In this case the ventilator fan must be mounted so that the warm air is extracted from the control cabinet and only filtered air can be drawn in. HEIDENHAIN advises against this method of cooling, since the function and reliability of electronic assemblies are adversely affected by contaminated air (fine dust, vapors etc.). In addition to these disadvantages, a filter that is not adequately serviced leads to a loss in cooling efficiency. Regular servicing is therefore absolutely vital.





3.3 Humidity

Permissible humidity: < 75% in continuous operation,
< 95% for not more than 30 days p.a. (randomly distributed).
In tropical areas it is recommended that the TNC not be switched off, so that condensation is avoided on the circuit boards. The heat generation prevents condensation and has no further disadvantages.

3.4 Mechanical Vibration

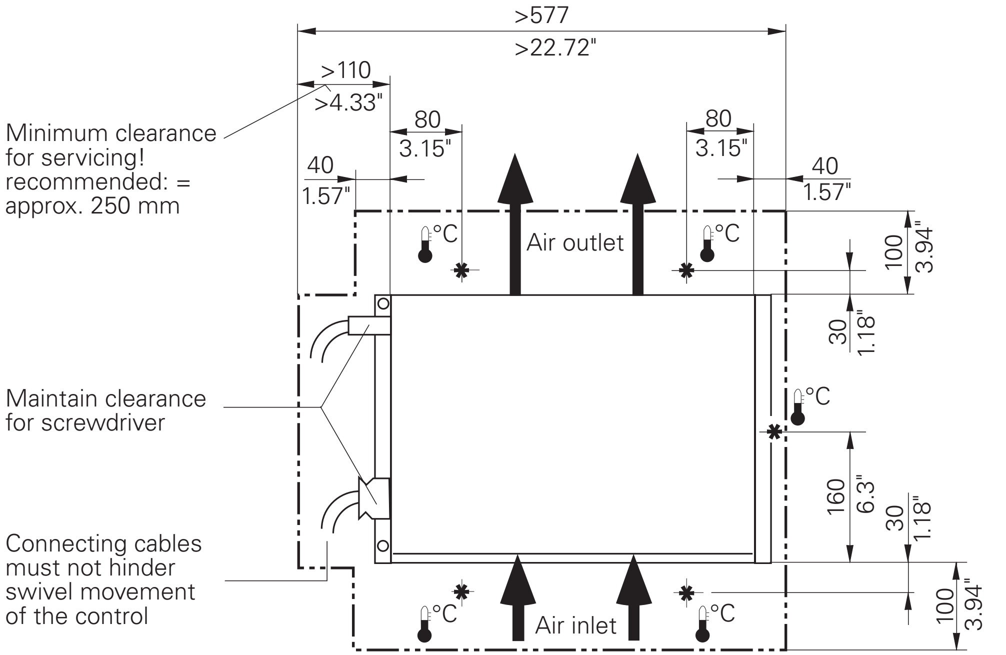
Permissible vibration: < 0.5 g

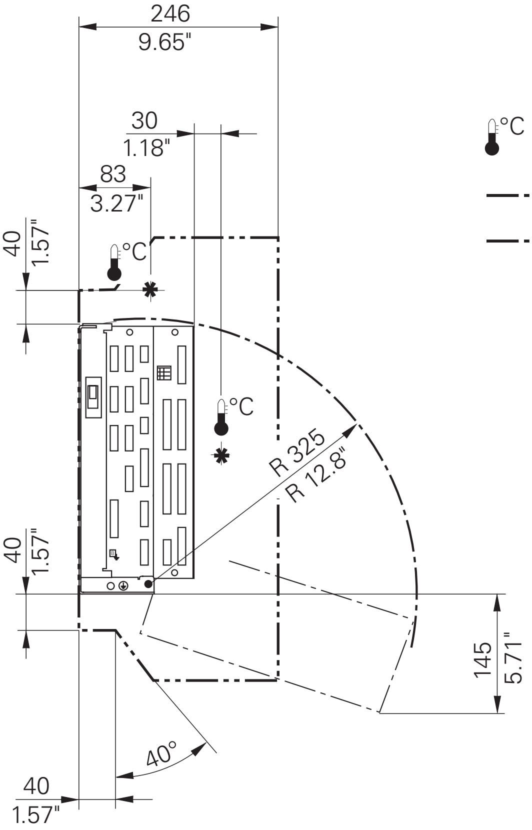
3.5 Mounting Position

Note the following fundamental points on mounting:

- mechanical accessibility,
- permissible environmental conditions,
- electrical noise immunity,
- the electrical regulations that are in force in your country.

LE 426 CB Logic Unit, TNC 430 CA


>577
>22.72"
>110
>4.33"
Minimum clearance for servicing! recommended: = approx. 250 mm
80
3.15"
80
3.15"
40
1.57"
40
1.57"
°C
Air outlet
°C
100
3.94"
30
1.18"
Maintain clearance for screwdriver
°C
160
6.3"
30
1.18"
Connecting cables must not hinder swivel movement of the control
°C
Air inlet
°C
100
3.94"



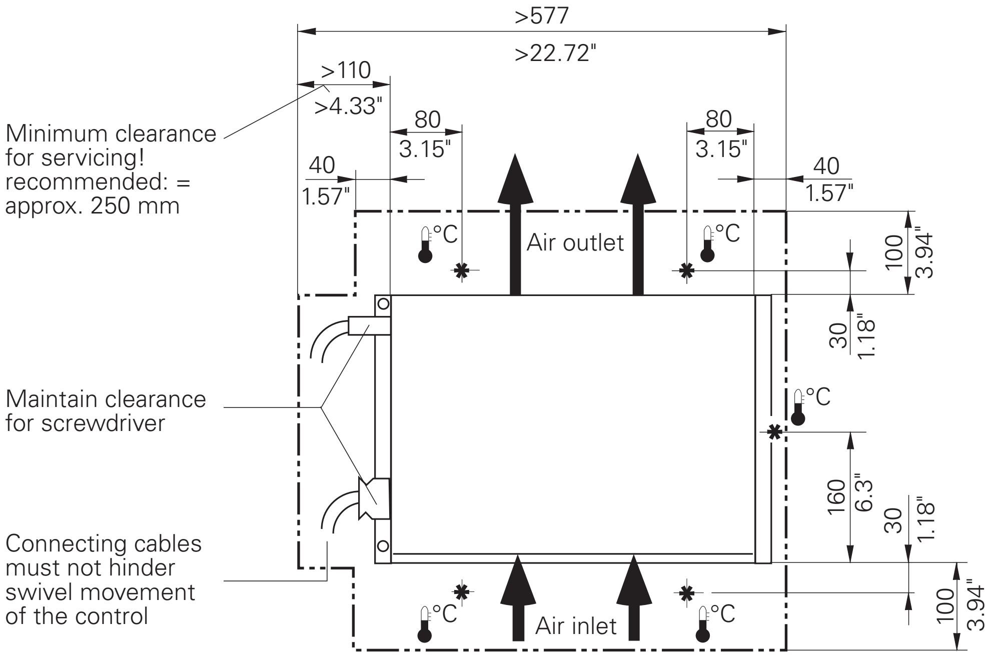
🌡°C ✱ Measuring point for ambient temperature

—··—··— Free space for air circulation

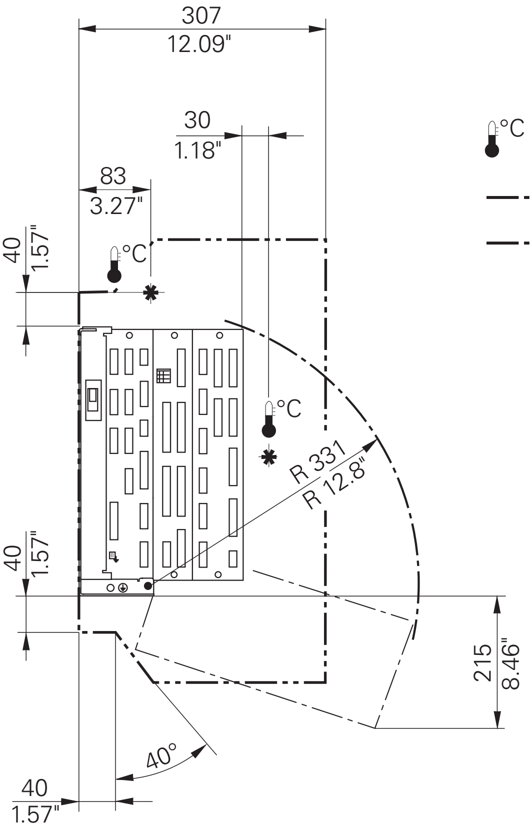
—·—·— Free space for servicing

Illustration of
max. swivel range.
The minimum angle of
swivel for exchange
of subassemblies should
be at least 90°.

LE 426 PB Logic Unit, TNC 430 PA



>577
>22.72"
>110
>4.33"
80
3.15"
80
3.15"
40
1.57"
40
1.57"
Minimum clearance for servicing! recommended: = approx. 250 mm
°C
°C
Air outlet
100
3.94"
30
1.18"
Maintain clearance for screwdriver
°C
160
6.3"
30
1.18"
Connecting cables must not hinder swivel movement of the control
°C
Air inlet
°C
100
3.94"



°C ✱ Measuring point for ambient temperature

—‑·· Free space for air circulation

—·— Free space for servicing

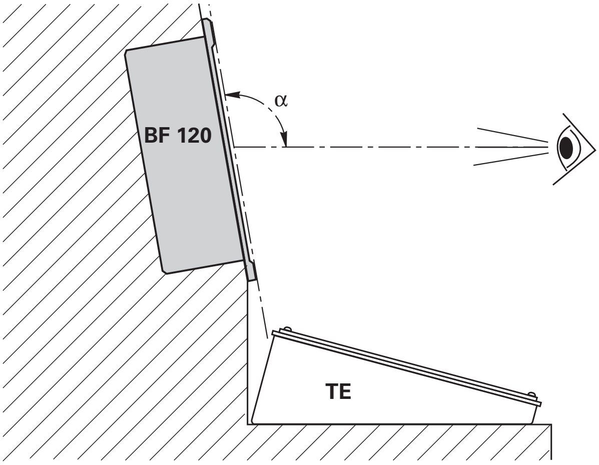
Illustration of max. swivel range. The minimum angle of swivel for exchange of subassemblies should be at least 90°.

Visual Display Unit

BC 120
When mounting the BC 120, remember that this unit is very sensitive to magnetic or electromagnetic pick-up. The picture can be disturbed by strong magnetic fields. For this reason, keep a minimum distance of 0.5 m between the VDU housing and the source of any disturbance (e.g. permanent magnets, motors, transformers etc.).

Free space for air circulation, see dimension drawing in Appendix

BF 120
The BF 120 flat-panel display must be viewed at the so-called "6 o'clock angle". Thus the display must be mounted as shown below.



150° > α > 90°

PLC Input/Output Unit
Up to four PL 410 B input/output units can be connected to the TNC.

TNC 426 CB, TNC 430 CA: One PL can be mounted on the logic unit. the others must be mounted in a separate switch cabinet.

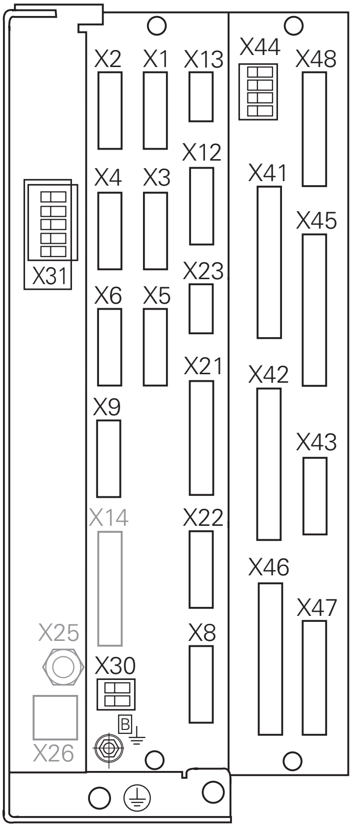
TNC 426 PB, TNC 430 PA: No PL can be mounted on the logic unit.

3.6 Degree of Protection

When mounted, the visual display unit and the keyboard unit provide class IP54 protection against dust and splashwater.

3.7 Connection Overview

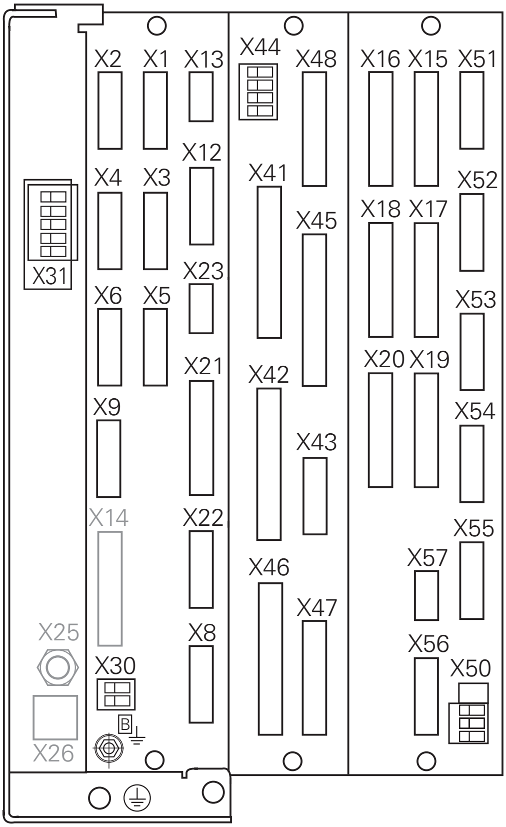
3.7.1 LE 426 CB



X1 to X5	Position encoder
X6	Encoder for spindle
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/V24 data interface
X22	RS-422/V11 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 visual display unit or
X49	Color flat panel display
X44	PLC power supply
X45	TNC keyboard
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
B	Signal ground
⏚	Protective ground (YE/GN)

⚠ **Danger to internal components!**
Do not engage or disengage any connections while the unit is under power.

3.7.2 LE 426 PB (Spindle with up to 9000 rpm)



X1
to
X5 Position encoder

X6 Encoder for spindle
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 Speed encoder

X21 RS-232-C/ V24 data interface
X22 RS-422/V11 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 visual display unit or
X49 Color flat panel display

X44 PLC power supply
X45 TNC keyboard
X46 Machine operating panel
X47 PLC expansion
X48 PLC analog input
X50 Input for drive motor enabling
X51
to
X56 Output to motor power stage

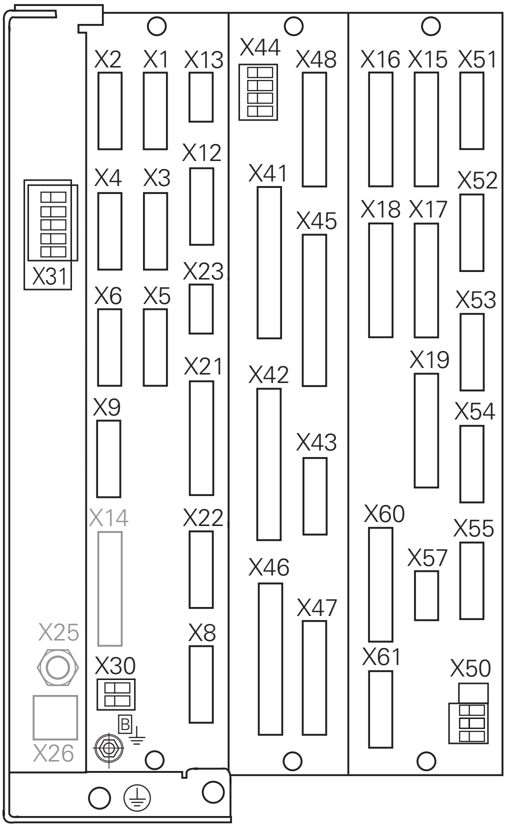
X57 *Reserved*

B Signal ground

⏚ Protective ground (YE/GN)

⚠ **Danger to internal components!**
Do not engage or disengage any connections while the unit is under power.

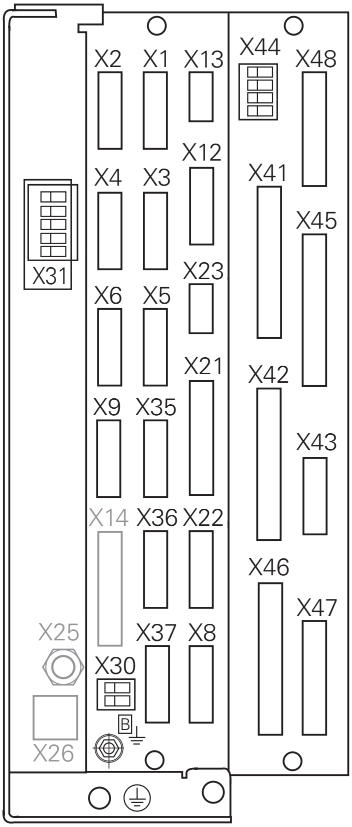
3.7.3 LE 426 PB (Spindle with up to 15 000 rpm)



X1	
to	
X5	Position encoder
X6	Encoder for spindle
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	Speed encoder
X21	RS-232-C/ V24 data interface
X22	RS-422/V11 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 visual display unit or
X49	Color flat panel display
X44	PLC power supply
X45	TNC keyboard
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
X50	Input for drive motor enabling
X51	
to	
X55	Output to motor power stage
X57	*Reserved*
X60	Encoder for spindle speed
X61	Output to motor power stage of the spindle
B	Signal ground
⏚	Protective ground (YE/GN)

⚠ **Danger to internal components!**
Do not engage or disengage any connections while the unit is under power.

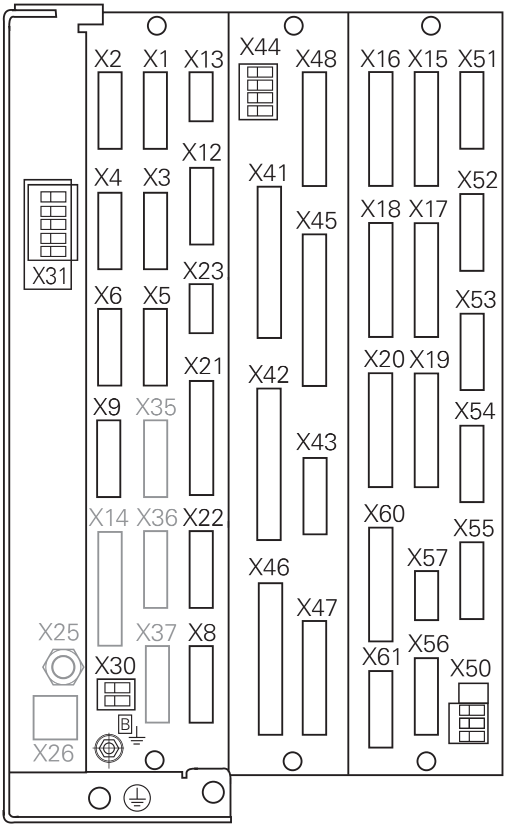
3.7.4 LE 430 CA



X1 to X5	Position encoder
X6	Encoder for spindle
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/V24 data interface
X22	RS-422/V11 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 visual display unit or
X49	Color flat panel display
X44	PLC power supply
X45	TNC keyboard
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
B	Signal ground
⏚	Protective ground (YE/GN)

⚠ **Danger to internal components!**
Do not engage or disengage any connections while the unit is under power.

3.7.5 LE 430 PA



X1 to X5	Position encoder
X6	Encoder for spindle
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	Speed encoder
X21	RS-232-C/ V24 data interface
X22	RS-422/V11 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 for 3 axes with analog speed command interface (Option)
X41	PLC output
X42	PLC input
X43	CRT visual display unit or
X49	Color flat panel display
X44	PLC power supply
X45	TNC keyboard
X46	Machine operating panel
X47	PLC expansion
X48	PLC analog input
X50	Input for drive motor enabling
X51 to X56	Output to motor power stage
X57	*Reserved*
X60	Encoder for spindle speed
X61	Output to spindle drive power stage
B	Signal ground
⏚	Protective ground (YE/GN)

⚠ **Danger to internal components!**
Do not engage or disengage any connections while the unit is under power.

3.8 Power Supply

3.8.1 NC Power Supply

LE 426 CB, LE 430 CA:
An NC power supply of 140 Vac to 450 Vac at terminals U_1 and U_2 is adequate for the LE 426 CB and the LE 430 CA. Power supply monitoring is switched off with Module 9167. To ensure compliance with the European standards for electromagnetic emission (EN 55022), the LE may be connected to the public power lines only through an isolating transformer or in conjunction with line filters. Compliance to these standards is one of the requirements for the use of the CE mark. If a line filter is already provided for the power stage, this power supply may also be used for the LE.

LE 426 PB, LE 430 PA:
For the LE 426 PB and the LE 430 PA, an NC power supply of 330 Vac to 450 Vac must be connected to U_1 and U_2. Here however this must be supplied via an isolating transformer (100 VA) with basic insulation according to IEC 742 EN 50 178.

To guarantee a secure power supply for the drive controllers in the event of a power failure, the LE must be supplied with the DC link voltage of the power stage (385 Vdc to 660 Vdc) at the terminals $+U_Z$ and $-U_Z$. If the DC link voltage is guaranteed to be available after switching the main switch on, the power supply of 190 Vac to 440 Vac at terminals U_1 and U_2 can be omitted. In this case, a jumper must be connected between $+U_Z$ and U_1. The rectified voltage is monitored. A brief surge voltage (approx. 5 s) of up to 720 Vdc is permitted. If 720 Vdc is exceeded, the NC revokes the pulse release ($\overline{\text{Reset}}$) for the IGBT of the power stage. The motors run down out of loop and no feedback to the DC link is possible. If 385 Vdc is not reached (power fail), all drives are brought to a stop under control. This power supply monitoring is enabled and disabled with Module 9167. If the voltage falls below 155 Vdc, a control unit reset takes place; the dc link power supply disconnects at 135 Vdc.

X31 NC power supply

Terminals	Assignment	LE 426 PB, LE 430 PA	LE 426 CB, LE 430 CA
⏚	Protective ground (YE/GN)		
U_1	Phase 1	330 Vac to 450 Vac via isolating transformer 50 to 60 Hz	140 Vac to 450 Vac 50 to 60 Hz
U_2	Phase 2		
$-U_Z$	DC-link voltage –	385 Vdc to 660 Vdc [1]	–
$+U_Z$	DC-link voltage +		

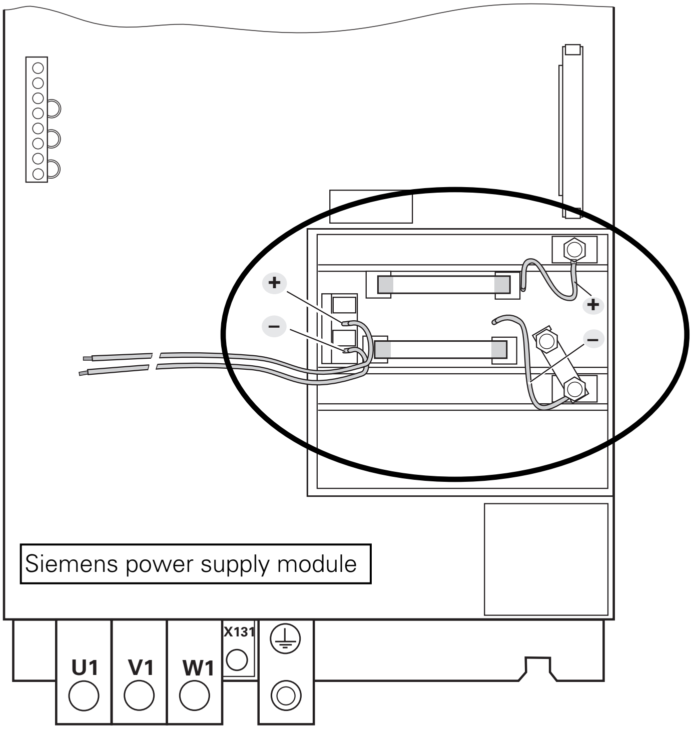
[1] other voltage ranges available upon request

Power consumption: approx. 55 W

⚠ **Danger of electrical shock!**
The dc-link power supply must be opened only by your HEIDENHAIN service agency.

The NC power supply is stored in the logic unit. You must provide fuse protection for the supply line. To protect the circuitry of the dc-link voltage, HEIDENHAIN offers a protective PCB for installation in SIEMENS power supply modules. This PCB is equipped with 4 A / 500 V fuses.

Id. Nr. 296 965 01



3.8.2 Buffer Battery Backup

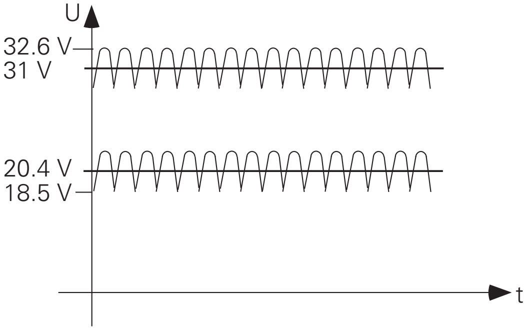
When the control is switched off, a buffer battery backup supplies the TNC with enough power to prevent data being lost from the RAM memory. If the message **EXCHANGE BUFFER BATTERY** appears on the TNC it is time for you to change the batteries. The batteries can be found in the logic unit, beside the power supply (round, black housing). The TNC also has an energy storage mechanism for supplying power to the TNC while the batteries are being changed (max. stored energy time: 24 hours).

Danger of electrical shock!
The machine tool and the TNC should be switched off while the battery is being changed!
The buffer batteries may only be changed by trained personnel!

Battery type: 3 round cells, leak-proof, IEC designation "LR6"

3.8.3 PLC Power Supply

The PLC of the LE and PL is powered by the 24 Vdc control voltage of the machine, which is generated in accordance with IEC 742 EN 50 178 (base insulation). Superimposed ac components, such as those caused by a three-phase bridge rectifier without smoothing, are permissible up to a ripple factor of 5% (see DIN 40110/10.75, Section 1.2).



The 0 V-line of the PLC-power supply must be grounded with an earth lead (∅ ≥ 6 mm^2) to the main frame ground of the machine.

Supply voltage	**Voltage range, mean dc voltage**	**Max. current consumption** (when half of the outputs are on simultaneously)	**Current consumption** (when half of the outputs are on simultaneously)
24 Vdc IEC 742 EN 50 178, base insulation	Lower limit 20.4 V ⎓ Upper limit 31 V ⎓ Voltage surges up to 36 V ⎓ for t < 100 ms are permissible.	**LE:** 2 A **PL 410 B:** 20 A	**LE:** 48 W **PL 410 B:** 480 W

X44 PLC power supply for the LE

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

Danger to internal components!
Use only original replacement fuses.

If the TNC 426 is used as a programming station, the PLC power supply must also be connected (connections no. 1 and 2).

Power supply for the PL 410 B

Terminal	Assignment	1st PL	2nd PL	3rd PL	4th PL
X9	0 V				
X10	+24 Vdc power for logic unit and for control-is-ready signal				
X11	+24 Vdc power supply for outputs	O32 to O39	O64 to O71	O95 to O102	O127 to O134
X12	+24 Vdc power supply for outputs	O40 to O47	O72 to O79	O130 to O110	O135 to O142
X13	+24 Vdc power supply for outputs	O48 to O55	O80 to O87	O111 to O118	O143 to O150
X14	+24 Vdc power supply for outputs	O56 to O62	O88 to O94	O114 to O125	O151 to O157

The PL 410 B input/output unit for the PLC is available as an option. It provides additional analog inputs and inputs for Pt 100 thermistors. The power supply for these analog inputs and thermistors must comply with IEC 742 EN 50 178, 5.88, "low-voltage electrical separation."

X23 Power supply of analog inputs on the PL 410 B

Terminal	Assignment
1	+ 24 Vdc (IEC 742 EN 50 178, 5.88, low-voltage electrical separation)
2	0 V

3.8.4 Power Supply for the Visual Display Units

BC 120

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

Connection to line voltage via Euro connector

Terminal	Assignment
L1	L1 (BK)
N	MP (BL)
⏚	Protective ground (YL/GY)

BF 120

X1 Power supply

Terminal	Assignment
1	+24Vdc operational voltage with basic insulation in accordance with IEC 742, EN 50 178
2	0 V

Power consumption: 15 W

3.9 Measuring Systems

HEIDENHAIN TNC contouring controls are designed for use with incremental linear and angular position feedback encoders as measuring systems.

It **does not matter** whether the encoders feature single or distance-coded reference marks, the TNC supports both types. However, HEIDENHAIN recommends using encoders with distance-coded reference marks because they significantly reduce the traverse distance required to establish the absolute position.

Maximum current consumption per encoder input:	200 mA
Maximum total current consumption for all encoder inputs:	
TNC 426 CB, TNC 430 CA:	1.2 A
TNC 426 PB, TNC 430 PA	2.4 A

Use only original HEIDENHAIN encoder cables, connectors and couplings. For maximum cable lengths see “Cable Overview.”

3.9.1 Encoders for Position

LE 426 PB	**Id. Nr.**	**311 999 ..**	**313 526 ..**	**317 349 ..**	**318 177 ..**
LE 426 CB	**Id. Nr.**	**312 002 ..**	**313 525 ..**		

X1, X2, X3 X4, X5 Encoder 1, 2, 3, 4, 5 (11 µA$_{PP}$)
Maximum input frequency: 50 kHz

Logic Unit		**Encoder cable**	
D-sub terminal (male) 15-pin	**Assignment**	**D-sub connector (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

The interface complies with the recommendations in IEC 742, EN 50 178 for separation from line power.

Encoder (1 V_{PP})

Connection	Id. Nr.	Max. input frequency
LE 426 PB		
X6	311 999 .., 312 000 .., 313 526 .., 313 527 .., 315 475 .., 317 349 .., 318 177 .., 318 178 .., 326 414 .., 326 416 .., 326 420 .., 326 421 ..	350 kHz
X1 to X5	312 000 .., 313 527 .., 315 475 .., 318 178 ..	350 kHz
	326 414 .., 326 416 .., 326 420 .., 326 421 ..	50 kHz
LE 426 CB		
X6	312 001 .., 312 002 .., 313 524 .., 313 525 .., 326 415 .., 326 419 ..	350 kHz
X1 to X5	312 001 .., 313 524 ..	350 kHz
	326 415 .., 326 419 ..	50 kHz
LE 430 PA		
X6	311 049 .., 313 521 .., 326 417 .., 325 716 ..	350 kHz
X1 to X5	311 049 .., 313 521 ..	350 kHz
	325 716 .., 326 417 ..	50 kHz
X35 to X38	311 049 .., 313 521 .., 326 417 .., 325 716 ..	350 kHz
LE 430 CA		
X6	311 050 .., 313 523 .., 326 418 .., 326 424 ..	350 kHz
X1 to X5	311 050 .., 313 523 ..	350 kHz
	326 418 .., 326 424 ..	50 kHz
X35 to X38	311 050 .., 313 523 .., 326 418 .., 326 424 ..	350 kHz

Logic Unit		Encoder cable	
D-sub terminal (male) 15-pin	**Assignment**	**D-sub connector (female) 15-pin**	
1	+ 5 V (U_P)	1	Brown/Green
2	0 V (U_N)	2	White/Green
3	A+	3	Brown
4	A–	4	Green
5	0 V	5	
6	B+	6	Gray
7	B–	7	Pink
8	0 V	8	
9	+ 5 V	9	Blue
10	R+	10	Red
11	0 V	11	White
12	R–	12	Black
13	0 V	13	
14	*Do not assign*	14	Violet
15	*Do not assign*	15	
Housing	External shield	Housing	External shield

☝ The interface complies with the recommendations in IEC 742, EN 50 178 for separation from line power.

3.9.2 Encoders for Speed (TNC 426 PB and TNC 430 PA Only)

X15 to X20, X60 Encoder for speed (1 V_{PP})

Maximum input frequency:	X15 to X20, X60	350 kHz
	X60 as of NC software 280 472 01	410 kHz

Logic unit		Adapter Cable Id. Nr. 289 440 ..		
D-sub terminal (male) 25-pin	**Assignment**	**D-sub connector (female) 25-pin**		**Connector (female) 17-pin**
1	(U_P) + 5 V or + 6.4V[1]	1	Brown/Green	10
2	0 V (U_N)	2	White/Green	7
3	A+	3	Green/Black	1
4	A–	4	Yellow/Black	2
5	0 V	5		
6	B+	6	Blue/Black	11
7	B–	7	Red/Black	12
8	0 V	8	Internal shield	17
9	*Do not assign*	9		
10	0 V	10		
11	*Do not assign*	11		
12	*Do not assign*	12		
13	Temperature +	13	Yellow	8
14	+ 5 V or not used[1]	14	Blue	16
15	Analog output (test)	15		
16	0 V	16	White	15
17	R+	17	Red	3
18	R–	18	Black	13
19	C+	19	Green	5
20	C–	20	Brown	6
21	D+	21	Gray	14
22	D–	22	Pink	4
23	+ 5 V (Test)	23		
24	0 V	24		
25	Temperature –	25	Violet	9
Housing	External shield	Housing	External shield	Housing

[1] The operating voltage is set by jumper on the PCB to match the voltage of the connected encoders.

The interface complies with the recommendations in IEC 742, EN 50 178 for separation from line power.

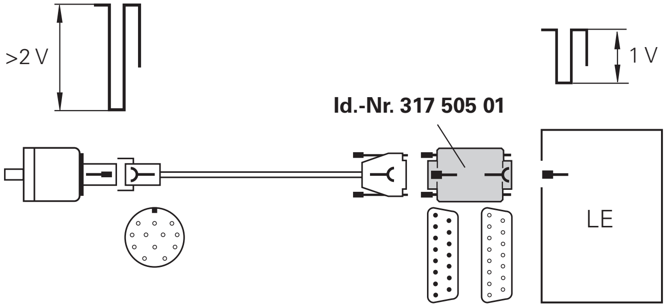
X15	Axis 1	X18	Axis 4	X60	**TNC 430 PA only:** Spindle
X16	Axis 2	X19	Axis 5		**TNC 426 PB (15 000):**
X17	Axis 3	X20	**TNC 430 PA:** Axis 6 **TNC 426 PB (9000):** Spindle		

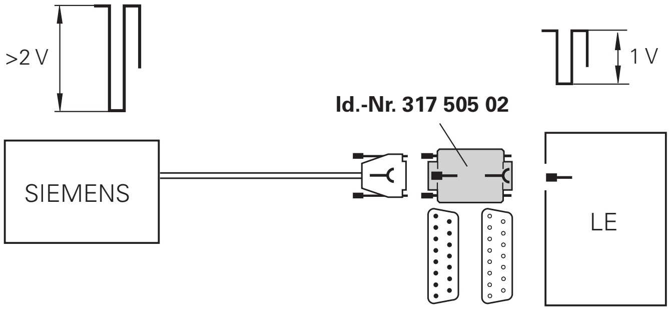
3.9.3 Adapter for Encoder Signals

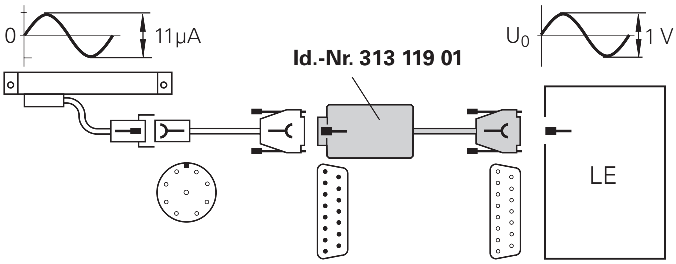
HEIDENHAIN offers a range of adapter connectors for adapting 11µA$_{PP}$ or TTL encoder signals to the 1V$_{PP}$ interface of the logic unit.

Please note that these adapters change the signal levels only, while the form of the signal remains the same.

Remember also that a square-wave signal has only 4-fold evaluation (cannot be interpolated). Neither can the TNC evaluate the fault detection signal of a square-wave encoder.







3.10 Motor Power Stage Connection

The SIMODRIVE 611 D is fitted with an expansion board ordered separately from HEIDENHAIN. These expansion boards are designed for two axes (inputs X1 and X2) and are connected to the **logic unit** via a connecting cable (see “Cable Overview”).

The expansion board must be connected according to the Basic Circuit Plan.

X51 to X56, X61 Output to motor power stage (only TNC 426 PB, TNC 430 PA)

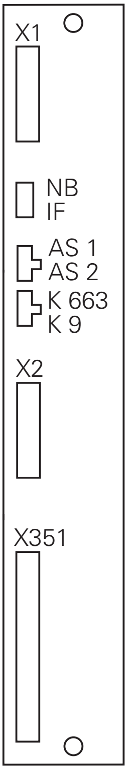
Logic unit		Connecting Cable Id. Nr. 289 208 ..			Expansion board Id. Nr. 324 952 ..	
D-sub terminal (female) 15-pin	**Assignment**	**D-sub connector (male) 15-pin**		**D-sub connector (female) 15-pin**	**X1, X2 D-sub terminal (female) 15-pin**	
1	*Do not assign*	1	Black	1	1	
2	PWM U_1	2	Blue	2	2	
3	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/Green	6	6	
7	$I_{\text{actual. 2}-}$	7	Gray/Pink	7	7	
8	$I_{\text{actual. 1}-}$	8	Black	8	8	
9	0V U_1	9	Red	9	9	
10	0V U_2	10	Pink	10	10	
11	0V U_3	11	Brown	11	11	
12	0V (analog)	12	Yellow	12	12	
13	$\overline{\text{Tempertr. warning}}$	13	Brown/Green	13	13	
14	$I_{\text{actual. 2}+}$	14	Red/Blue	14	14	
15	$I_{\text{actual. 1}+}$	15	Violet	15	15	
Housing	External shield	Housing	External shield	Housing	Housing	

The interface complies with the recommendations in IEC 742, EN 50 178 for separation from line power.

Logic level:	5 V
Analog signals I_{actual}:	± 7.5 V
Maximum PWM frequency:	5 kHz

X51	Axis 1	X54	Axis 4	X61	**Only TNC 430 PA:** Spindle
X52	Axis 2	X55	Axis 5		**TNC 426 PB (15 000):**
X53	Axis 3	X56	**TNC 430 PA:** Axis 6 **TNC 426 PB (9000):** Spindle		

Connection overview of the HEIDENHAIN expansion board



X1 and X2	Connection to LE 426 PA / LE 430 PA
X351	SIMODRIVE hardware bus
NB (Red)	Not ready Monitoring of U_z, temperature, power supply and pulse release
IF (Green)	Pulse release
AS1	Forced break contact 1
AS2	Forced break contact 2
K663	Safety relay for pulse release
K9	Power supply from SIMODRIVE hardware bus

3.11 Analog Inputs

The **logic unit** and the PLC input/output board have analog inputs (± 10 V) and inputs for Pt 100 thermistors. The PL 410 B is available with or without analog inputs. The analog inputs must be activated on the PL 410 B by a DIL switch.

	Analog inputs (±10 V)	**Inputs for Pt 100 thermistors**
Logic unit	3	3
PL 410 B	4	4

The current values of these inputs are interrogated with Module 9003.

Analog inputs:

Voltage range	– 10 V to + 10 V
Input resistance	> 250 kΩ
Resolution	100 mV
Internal value range	– 100 to + 100

Inputs for Pt 100 thermistors:

Constant current	5 mA
Temperature range	0° C to 100° C
Resolution	0.5° C
Internal value range	0 to 200

X48 Analog input (PLC) on the LE

D-sub terminal (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
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)
13, 20 to 25	*Do not assign*
Housing	External shield

The correct polarity of analog inputs is essential.

X15 to X18 Analog input on the PL 410 B

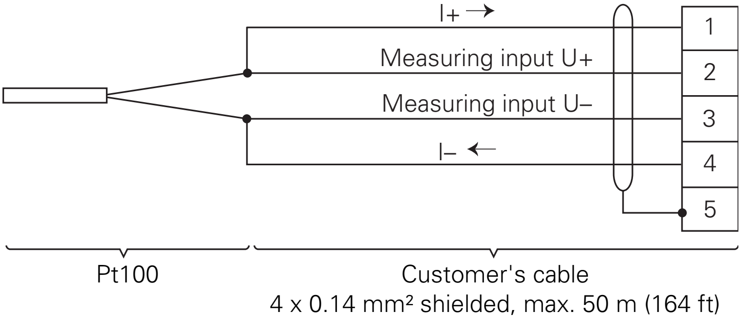
Terminal	Assignment
1	–10 V to +10 V
2	0 V (reference potential)
3	Shield

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

Terminal	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
Connecting cable, 2 x 0.14 mm^2 shielded, max. 50 m.

Connection to the inputs for Pt 100 thermistors
The connection to the Pt 100 thermistors must be arranged as a four-wire circuit.
e.g. PL 410 B X19:



3.12 Analog Outputs

Maximum loading of the analog outputs:	2 mA
Maximum capacitance:	2 nF

There are 13 analog outputs available:

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

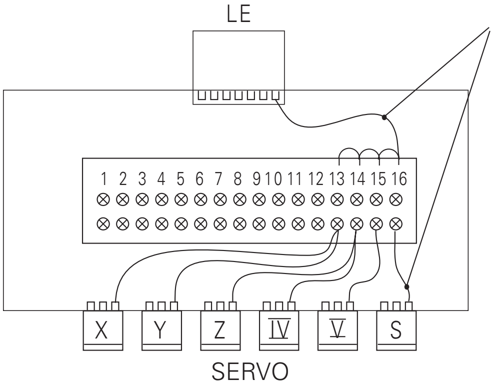
PLC analog outputs

The PLC analog outputs can be switched via Module 9130.

Nominal value output:

- For analog axes and analog spindle, use MP120.x and MP121 to assign the corresponding analog outputs on connection 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 it is necessary to branch to physically separate servo inputs, the connection must be made in a grounded terminal box. Suitable terminal boxes is available from HEIDENHAIN (Id. Nr. 251 249 01).
- The chassis of the terminal box must be electrically connected with the frame of the machine.
- The 0 V connection of the nominal-value-difference inputs must be connected with signal ground. Required cross section ≥ Ø 6 mm².
- Use only original HEIDENHAIN cables and connecting elements.

The following wiring plan is suggested for shielding the terminal box:



Insulated against housing.

Leads are provided with end sleeves.

Cable shielding is led onto 14 mm² insulated strands via crimp eyelets.

Connection terminal	Assignment	
1	Nominal value output: ±10 V	X axis
2	Nominal value output: 0 V	X axis
3	Nominal value output: ±10 V	Y axis
4	Nominal value output: 0 V	Y axis
5	Nominal value output: ±10 V	Z axis
6	Nominal value output: 0 V	Z axis
7	Nominal value output: ±10 V	Axis 4
8	Nominal value output: 0 V	Axis 4
9	Nominal value output: ±10 V	Axis 5
10	Nominal value output: 0 V	Axis 5
11	Nominal value output: ±10 V	Spindle
12	Nominal value output: 0 V	Spindle
13	Shield connection	
14	Shield connection	
15	Shield connection	
16	Shield connection	

X8 Analog output 1 to 6

Logic unit		Connecting Cable	
D-sub terminal (female) 15-pin	**Assignment**	**D-sub connector (male) 15-pin**	**Color**
1	Analog output 1: ±10 V	1	BN
2	*Do not assign*	2	BN/GN
3	Analog output 2: ±10 V	3	YL
4	Analog output 5: ±10 V	4	RD/BL
5	Analog output 3: ±10 V	5	PK
6	Analog output 5: 0 V	6	GY/PK
7	Analog output 4: ±10 V	7	RD
8	Analog output 6: ±10 V	8	VI
9	Analog output 1: 0 V	9	WH
10	*Do not assign*	10	WH/GY
11	Analog output 2: 0 V	11	GN
12	*Do not assign*	12	
13	Analog output 3: 0 V	13	GY
14	Analog output 4: 0 V	14	BL
15	Analog output 6: 0 V	15	BK
Housing	External shield	Housing	External shield

X9 Analog output 7 to 13

Logic unit		Connecting cable	
D-sub terminal (female) 15-pin	**Assignment**	**D-sub connector (male) 15-pin**	**Color**
1	Analog output 7: ±10 V	1	BN
2	Analog output 13: ±10 V	2	BN/GN
3	Analog output 8: ±10 V	3	YL
4	Analog output 12: ±10 V	4	RD/BL
5	Analog output 9: ±10 V	5	PK
6	Analog output 12: 0 V	6	GY/PK
7	Analog output 10: ±10 V	7	RD
8	Analog output 11; ±10 V	8	VI
9	Analog output 7: 0 V	9	WH
10	Analog output 13: 0 V	10	WH/GY
11	Analog output 8: 0 V	11	GN
12	*Do not assign*	12	
13	Analog output 9: 0 V	13	GY
14	Analog output 10: 0 V	14	BL
15	Analog output 11: 0 V	15	BK
Housing	External shield	Housing	External shield

For connecting cable see “Cable Overview.”

3.13 Touch Probes

The following touch probes can be connected:
The triggering touch probes

TS 220	With cable connection; for digitizing, workpiece setup and measurement during machining
or	
TS 630	With infrared transmission; for workpiece setup and measurement during machining
and	
TT 120	For tool measurement
and	
One measuring touch probe	

For information on touch probe connecting cables see “Cable overview.”

3.13.1 Triggering Touch Probe for Workpiece Measurement

X12 Triggering Touch Probe for workpiece measurement

Logic unit	
D-sub terminal (female) 15-pin	**Assignment**
1	0 V (internal shield)
3	Ready
4	Start
5	+15 V ± 10% (U_P)
6	+ 5 V ± 5% (U_P)
7	$\overline{\text{Battery warning}}$
8	0 V (U_N)
9	Trigger signal
10	$\overline{\text{Trigger signal}}$ [1]
2, 11 to 15	*Do not assign*
Housing	External shield

[1] Stylus at rest means logic level High.

The interface complies with the recommendations in IEC 742, EN 50 178 for separation from line power.

Adapter Cable Id. Nr. 274 543		
D-sub connector (male) 15-pin		**Coupling on mounting base 6-pin**
3	Pink	4
5	Gray	
6	Brown/Green	2
7	Gray	3
8	White/Green	1
9	Green	5
10	Yellow	6
Housing	External shield	Housing

TS120 Id. Nr. 265 348 .. TS220 Id. Nr. 293 488 ..	
Quick disconnect 6-pin	
4	Gray
2	Brown
3	Gray
1	White
5	Green
6	Yellow
Housing	External shield

Connecting Cable Id. Nr. 310 197 ..		
D-sub connector (male) 15-pin		**Connector (female) 7-pin**
1	White/Brown Internal shield	7
3	Gray	5
4	Yellow	3
5	Brown	2
7	Blue	6
8	White	1
10	Green	4
Housing	External shield	Housing

EA Id. Nr. 262 904 01	
Coupling on mounting base 7-pin	
7	Internal shield
5	Gray
3	
2	Brown
6	Blue
1	White
4	Green
Housing	External shield

TS 630
Id. Nr. 293 714 ..

3.13.2 Touch Trigger Probe for Tool Measurement

X13 Touch Trigger Probe for tool measurement

Logic unit	
D-sub terminal (female) 9-pin	**Assignment**
1	Ready
2	0 V (U_N)
4	+15 V ± 5% (U_P)
7	+5 V ± 5% (U_P)
8	Trigger signal
9	$\overline{\text{Trigger signal}}$ 1)
3, 5, 6	*Do not assign*
Housing	External shield

1) Stylus at rest means logic level High.

The interface complies with the recommendations in IEC 742, EN 50 178 for separation from line power.

Adapter Cable Id. Nr. 310 200 ..		
D-sub connector (male) 9-pin		**Coupling on mounting base (female) 6-pin**
1	Pink	6
2	White/Green	1
4	Gray	5
7	Brown/Green	2
8	Green	3
9	Yellow	4
Housing	External shield	Housing

TT 120 Id. Nr. 295 743 03	
Connector (male) 6-pin	
6	
1	White
5	
2	Brown
3	Green
4	Yellow
Housing	External shield

3.13.3 Measuring Touch Probe (Option)

In the standard version, the X14 socket is not present on the logic unit. With the "Digitizing with measuring touch probe" option you receive an adapter kit with the X14 socket. In the TNC 430 you can install either the adapter kit for the measuring touch probe at X14 or the adapter kit for an additional linear encoder input X38.

- Adapter kit for SP 2/1 Id. Nr. 311 647 51

X14 Measuring Touch Probe SP 2/1

Logic unit (adapter kit Id. Nr. 311 647 51)			Adapter Cable Id. Nr. 296 839			Connecting Cable A-1016-6640 Renishaw			Renishaw SP 2/1
D-sub terminal (female) 25-pin	Assignment		D-sub connector (male) 25-pin		Coupling on mounting base (female) 21-pin	Connector (male) 21-pin			
3	$\overline{\text{Ua2}}$	Axis X	3	Pink	7				
4	$\overline{\text{Ua1}}$		4	Yellow	5				
16	Ua2		16	Gray	6				
17	Ua1		17	Green	4				
7	$\overline{\text{Ua2}}$	Axis Y	7	Brown/Blue	11				
8	$\overline{\text{Ua1}}$		8	Red	9				
20	Ua2		20	White/Blue	10				
21	Ua1		21	Blue	8				
11	$\overline{\text{Ua2}}$	Axis Z	11	Violet	19				
12	$\overline{\text{Ua1}}$		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	External shield		Housing	External shield	Housing	Housing	External shield	Housing	Housing

The interface complies with the recommendations in IEC 742, EN 50 178 for separation from line power.

3.14 Data Interface

The TNC features three data interfaces:

- one RS-232-C/V.24 data interface and
- one V.11/RS-422 data interface
- one Ethernet interface

Devices can be connected to all three interfaces, and the user can select which one he wishes to use, see "Data Interface."

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

- Maximum cable length is 20 meters (66 ft).
- To connect a peripheral device you must install an adapter cable either in the switching cabinet or on the operating panel. See also the "Dimensions" section in the Appendix.
- For information on interface cables, see "Cable Overview."

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

Logic unit		**Connecting Cable Id. Nr. 239 760 ..**			**Adapter Block Id. Nr. 310 085 01**		**Connecting Cable Id. Nr. 274 545 01**		
D-sub terminal (female) 25-pin	**Assignment**	**D-sub connector (male) 25-pin**		**D-sub connector (female) 25-pin**	**D-sub terminal (male) 25-pin**	**D-sub terminal (female) 25-pin**	**D-sub connector (male) 25-pin**		**D-sub connector (female) 25-pin**
1	GND	1	WH/BN External 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	External shield	Housing	External shield	Housing	Housing	Housing	Housing	External shield	Housing

The interfaces complies with the recommendations in IEC 742 EN 50 178 for separation from line power.

3.14.2 RS-422/V.11 Data Interface

- Maximum cable length is 1000 meters (3280 ft).
- To connect a peripheral device you must install an adapter cable either in the switching cabinet or on the operating panel. See also the "Dimensions" section in the Appendix.
- For information on interface cables see “Cable Overview.”

Cable type LIYCY [7 (2 x 0.14 mm^2)] Cu must be used for connection to the peripheral device. HEIDENHAIN offers a 15-pin D-subminiature connector (Id. Nr. 315 650 03) for this cable.

X22 RS-422/V.11 Data interface

Logic unit		**Connecting Cable Id. Nr. 289 208 ..**			**Adapter Block Id. Nr. 311 086 01**	
D-sub terminal (female) 15-pin	**Assignment**	**D-sub connector (male) 15-pin**		**D-sub connector (female) 15-pin**	**D-sub terminal (male) 15-pin**	**D-sub terminal (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		Housing	Housing	Housing

The interfaces complies with the recommendations in IEC 742 EN 50 178 for separation from line power.

3.14.3 Ethernet Interface (Option)

Maximum transfer speed: 200 to 1000 kilobaud

X25 Ethernet Interface RJ45 connection

Maximum cable length:	unshielded:	100 m (329 ft)
	shielded:	400 m (1312 ft)

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*

The interfaces complies with the recommendations in IEC 742 EN 50 178 for separation from line power.

X26 Ethernet Interface BNC connection (coax cable)

Maximum cable length: 180 m

BNC connection (female)	Assignment
Inner conductor (core)	Data (RXI, TXO)
Shielding	GND

3.15 Handwheel Input

The following handwheels are compatible with HEIDENHAIN contouring controls:

- One HR 130 panel-mounted handwheel, or
- HR 150 panel-mounted handwheels via HRA 110 handwheel adapter, or
- One HR 410 portable handwheel

3.15.1 Connector Assignment on the LE

X23 Handwheel Input

D-sub terminal (female) 9-pin	Assignment
2	0 V
4	+12 V ± 0.6 V (Uv)
6	DTR
7	TxD
8	RxD
9	DSR
1, 3, 5	*Do not assign*
Housing	External shield

The interfaces complies with the recommendations in IEC 742 EN 50 178 for separation from line power.

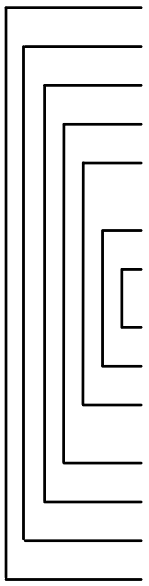
3.15.2 HR 410 Portable Handwheel

The HR 410 is a portable electronic handwheel with:

- Five axis-selection keys
- Two traverse direction keys
- Three keys with predefined traverse speeds (slow, medium, fast)
- Actual-position-capture key
- Three keys for machine functions to be determined by the machine tool builder
- Two permissive keys
- EMERGENCY STOP button
- Holding magnets

Dummy plug for EMERGENCY STOP circuit		Id. Nr. 271 958 03
Connecting cables:	Helix cable 3m	Id. Nr. 312 879 01
	Normal cable	Id. Nr. 296 467 ..
	Metal armor	Id. Nr. 296 687 ..

Extension Cable Id. Nr. 281 429..			**Adapter Cable Id. Nr. 296 466..**			**Connecting Cable Id. Nr. see above**			**HR 410 Id. Nr. 296 469 01**	
D-sub connector (male) 9-pin		**D-sub connector (female) 9-pin**	**D-sub connector (male) 9-pin**		**Coupling on mounting base (female) (5+7)-pin**	**Connector (male) (5+7)-pin**		**Connector (female) (5+7)-pin**	**Connector (male) (5+7)-pin**	
Housing	Shield	Housing	Housing	Shield	Housing	Housing	Shield	Housing	Housing	Shield
2	White	2	2	White	E	E	White	E	E	
4	Brown	4	4	Brown	D	D	Brown	D	D	
6	Yellow	6	6	Yellow	B	B	Yellow	B	B	
7	Gray	7	7	Gray	A	A	Gray	A	A	
8	Green	8	8	Green	C	C	Green	C	C	
					6	6	BK	6	6	
					7	7	RD/BL	7	7	
					5	5	RD	5	5	
					4	4	BL	4	4	
					2	2	WH/GN	2	2	
					3	3	BN/GN	3	3	
					1	1	GY/PK	1	1	

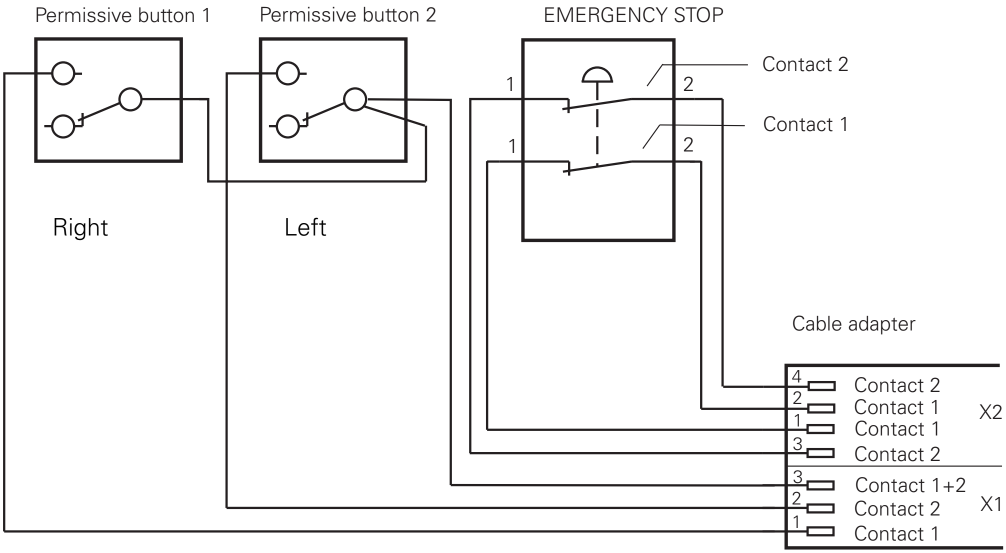


WH/BN	3	Contact 1 + 2	
WH/YL	2	Contact 2 (left)	Permissive button
WH/GN	1	Contact 1 (right)	

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 (maximum load 1.2 A).

Internal wiring of the contacts to the permissive buttons and the EMERGENCY STOP button of the HR 410:



The plug-in terminal strips are included in delivery with the adapter cable. If you have an immediate need for these terminal strips before the adapter cable, they can be ordered separately:

Plug-in terminal strip, 3-pin	Id. Nr. 266 364 06
Plug-in terminal strip, 4-pin	Id. Nr. 266 364 12

3.15.3 Panel-Mounted Handwheel HR 130

The HR 130 is the panel-mount version of the HR 410 without axis keys, rapid traverse keys, etc. It is connected to the logic unit directly or by extension cable.

The HR 130 is available in various versions (standard cable length 1 meter):

• Small knob, axial cable outlet:	Id. Nr. 254 040 01
• Small knob, radial cable outlet:	Id. Nr. 254 040 02
• Large knob, axial cable outlet:	Id. Nr. 254 040 03
• Large knob, radial cable outlet:	Id. Nr. 254 040 04
• Ergonomic knob, radial cable outlet:	Id. Nr. 254 040 05

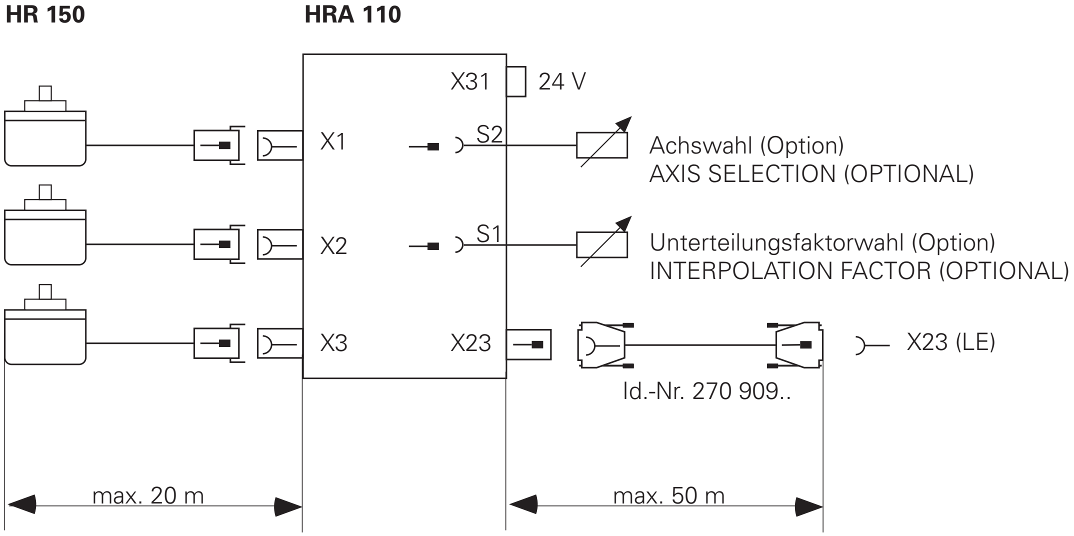
See also the "Dimensions" section in the Appendix.

Extension Cable Id. Nr. 281 429 ..		
D-sub connector (male) 9-pin		**D-sub connector (female) 9-pin**
Housing	Shield	Housing
2	White	2
4	Brown	4
6	Yellow	6
8	Green	8
7	Gray	7

HR 130 Id. Nr. 254 040 ..	
D-sub connector (male) 9-pin	
Housing	Shield
2	White
4	Brown
6	Yellow
8	Green

3.15.4 Handwheel Adapter HRA 110

You can connect two or three HR 150 panel-mount handwheels to the TNC using the HRA 110 handwheel adapter. The first two handwheels are assigned to axes 1 and 2. The third handwheel can be assigned to axes 1 to 5 either via a step switch (option) or with MP7645 (see also the chapter on "Machine Integration").



A further step switch (option) offers the possibility to select the interpolation factor for the handwheels. You have to evaluate the current position of the switch in the PLC, and then you can activate the corresponding interpolation factor with Module 9036.

X1, X2, X3 on the HRA 110: Handwheel Inputs for HR 150

HRA 110 Terminal (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	0V
9	Internal shield
Housing	External shield

X23 on the HRA 110: Connection to the Logic Unit

HRA 110	
D-sub connector (male) 9-pin	**Assignment**
1	RTS
2	0V
3	CTS
4	+12V + 0.6 V (Uv)
5	*Do not use*
6	DSR
7	RxD
8	TxD
9	DTR
Housing	External shield

X31 on the HRA 110: Power Supply

HRA 110	
Terminal	**Assignment**
1	+ 24 Vdc
2	0V

Power supply:	24 Vdc in accordance with IEC 742 EN 50 178 (basic insulation)
Max. current consumption:	200 mA

☞ The same power supply must not be used for the PLC and the HRA 110 at the same time, because this would bridge the metallic isolation of the PLC inputs/outputs.

3.16 Input: Spindle Reference Signal

With MP3143 you can set whether or not you will use the input X30 for the evaluation of the spindle reference signal, see section "Spindle." If you have mounted the HEIDENHAIN rotary encoder directly on the spindle, you must not switch this input.

X30 Spindle Reference Signal

Terminal	Assignment
1	+24 V
2	0V

3.17 Input: Drive Motor Enabling (Only TNC 426 PB, TNC 430 PA)

A power supply of 24 V is necessary for drive motor enabling.

Logic unit to Id. Nr. xxx xxx 3x
X50 Drive motor enabling

Terminal	Assignment
1	+ 24 Vdc
2	*Do not assign*
3	0 V

Logic unit from Id. Nr. xxx xxx 4x
X50 Drive motor enabling

Terminal	Assignment
1	+ 24 Vdc
2	*Do not assign*
3	0 V

3.18 Switching Inputs 24 Vdc (PLC)

Voltage range:	**Logic unit**	**PL 410 B**
“1” signal: U_i	13 V to 30.2 V	
“0” signal: U_i	– 20 V to 3.2 V	
Current ranges:		
“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

Address	**No. of inputs**	**Device**
I0 to I31	31 + control-is-ready signal	Logic unit X42 (PLC input)
I128 to I152	25	Logic unit X46 (machine operating panel)
I64 to I127	64	First PLC input/output board
I192 to I255	64	Second PLC input/output board
I256 to I319	64	Third PLC input/output board
I320 to I383	64	Fourth PLC input/output board

X42 PLC Inputs on the LE

Logic unit		Conn. Cable Id. Nr 244 005 .. / Id. Nr. 263 954 ..	
D-sub terminal (female) 37-pin	**Assignment**	**D-sub connector (male) 37-pin**	
1	I0	1	Gray/Red
2	I1	2	Brown/Black
3	I2	3	White/Black
4	I3 acknowledgment of control-is-ready signal	4	Green/Black
5	I4	5	Brown/Red
6	I5	6	White/Red
7	I6	7	White/Green
8	I7	8	Red/Blue
9	I8	9	Yellow/Red
10	I9	10	Gray/Pink
11	I10	11	Black
12	I11	12	Pink/Brown
13	I12	13	Yellow/Blue
14	I13	14	Green/Blue
15	I14	15	Yellow
16	I15	16	Red
17	I16	17	Gray
18	I17	18	Blue
19	I18	19	Pink
20	I19	20	White/Gray
21	I20	21	Yellow/Gray
22	I21	22	Green/Red
23	I22	23	White/Pink
24	I23	24	Gray/Green
25	I24	25	Yellow/Brown
26	I25	26	Gray/Brown
27	I26	27	Yellow/Black
28	I27	28	White/Yellow
29	I28	29	Gray/Blue
30	I29	30	Pink/Blue
31	I30	31	Pink/Red
32	I31	32	Brown/Blue
33	*Do not use*	33	Pink/Green
34	*Do not use*	34	Brown
35	0 V (PLC) Test output; *Do not use*	35	Yellow/Pink
36	0 V (PLC) Test output; *Do not use*	36	Violet
37	0 V (PLC) Test output; *Do not use*	37	White
Housing	External shield	Housing	External shield

PLC Inputs on the PL 410 B

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

3.19 Switching Outputs 24 Vdc (PLC)

Transistor outputs with current limitation

	Logic unit	PL 410 B
Min. output voltage for “1” signal	3 V below supply voltage	
Nominal operating current per output	0.1 A	1.2 A

- Permissible load: resistive load; inductive load only with quenching diode parallel to the inductance.
- No more than one output may be shorted on the logic unit at any time. Short circuit of **one** output does not cause an overload.
- No more than half the PLC outputs may be driven at the same time (simultaneity factor 0.5).

Address	No. of outputs	Device
O0 to O30	31	Logic unit X41 (PLC output)
O0 to O7		Logic unit X46 (machine operating panel)
O32 to O62	31	First PLC input/output board
O64 to O94	31	Second PLC input/output board
O128 to O158	31	Third PLC input/output board
O160 to O190	31	Fourth PLC input/output board

X41 PLC output on the LE

Logic unit		Conn. Cable Id.-Nr 244 005 .. / Id. Nr. 263 954 ..	
D-sub terminal (female) 37-pin	**Assignment**	**D-sub connector (male) 37-pin**	
Power 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
Power supply via X44, pin 2; can be switched off with EMERGENCY STOP			
17	O16	17	Gray
18	O17	18	Blue
19	O18	19	Pink
20	O19	20	White/Gray
21	O20	21	Yellow/Gray
22	O21	22	Green/Red
23	O22	23	White/Pink
24	O23	24	Gray/Green
Power supply via X44, pin 1; **cannot** be switched off with EMERGENCY STOP			
25	O24	25	Yellow/Brown
26	O25	26	Gray/Brown
27	O26	27	Yellow/Black
28	O27	28	White/Yellow
29	O28	29	Gray/Blue
30	O29	30	Pink/Blue
31	O30	31	Pink/Red

Logic unit		Conn. Cable Id.-Nr 244 005 .. / Id. Nr. 263 954 ..	
D-sub terminal (female) 37-pin	Assignment	D-sub connector (male) 37-pin	
32	Test output; *Do not use*	32	Brown/Blue
33	Test output; *Do not use*	33	Pink/Green
34	Control-is-ready	34	Brown
35	Test output; *Do not use*	35	Yellow/Pink
36	Test output; *Do not use*	36	Violet
37	Test output; *Do not use*	37	White
Housing	External shield	Housing	External shield

PLC outputs on the PL 410 B

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			

3.20 Connection of the PLC Expansion

Up to four PL 410 B can be connected to the TNC.

TNC 426 CB, TNC 430 CA: One PL can be mounted on the logic unit. Further PLs must be mounted separately in the control cabinet.

TNC 426 PB, TNC 430 PA: No PL may be mounted on the logic unit.

The PL 410 B is available with or without analog inputs. The analog inputs must be activated by means of a DIL switch on the PL.

PL 410 B	**Switching inputs 24 Vdc**	**Switching outputs 24 Vdc**	**Analog inputs (±10 V)**	**Inputs for Pt 100 thermistors**
Id. Nr. 263 371 12	64	31	–	–
Id. Nr. 263 371 02	64	31	4	4

X47 PLC expansion on the LE

Logic unit		Conn. Cable Id. Nr. 289 111 ..			1st PL 410 B	
D-sub terminal (male) 25-pin	Assignment	D-sub connector (female) 25-pin		D-sub connector (male) 25-pin	X1 D-sub terminal (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 use*	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	$\overline{\text{WRITE EXTERNAL}}$	8	White/Blue	8	8	$\overline{\text{WRITE EXTERNAL}}$
9	WRITE EXTERNAL	9	White/Red	9	9	WRITE EXTERNAL
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 use*	13		13	13	*Do not use*
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 use*	16	Gray/Blue	16	16	PCB identifier 2
17	*Do not use*	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	EMERGENCY STOP	20	Green/Red	20	20	EMERGENCY 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 on the PL 410 B

PL 410 B		Connecting Cable Id. Nr. 289 111 ..			PL 410 B	
X2 D-sub terminal (male) 25-pin	Assignment	D-sub connector (female) 25-pin		D-sub connector (male) 25-pin	X1 D-sub terminal (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 use*	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	$\overline{\text{WRITE EXTERNAL}}$	8	White/Blue	8	8	$\overline{\text{WRITE EXTERNAL}}$
9	WRITE EXTERNAL	9	White/Red	9	9	WRITE EXTERNAL
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 use*	13		13	13	*Do not use*
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	EMERGENCY STOP	20	Green/Red	20	20	EMERGENCY 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

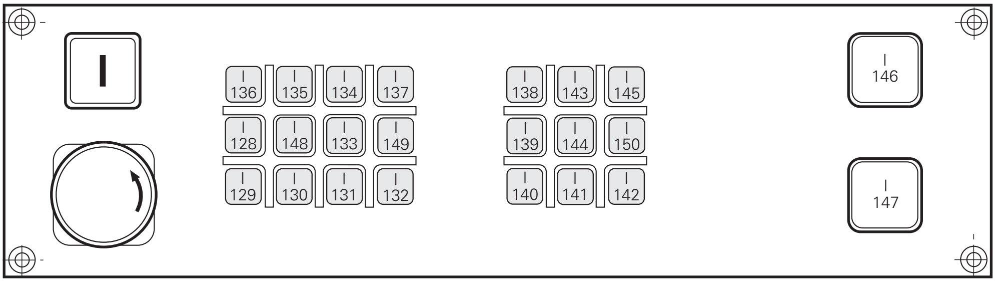
3.21 Machine Operating Panel

For machines with up to four axes, HEIDENHAIN offers the MB 420 machine operating panel. This is mounted below the TNC operating panel. The standard arrangement of keys may be seen in the dimension drawing.

The operating panel is supplied with four additional black keys, which can be substituted for unrequired axis keys. Other keys with different symbols are also available on request.

MB 420 Id. Nr. 293 757 12

Assignment of the PLC inputs to the keys of the MB 420:



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

Danger to internal components!
PLC inputs I128 to I152 must be switched only with the power supply from pins 36 and 37, since this power supply is internally protected for this purpose (PLC supply voltage from X44 connection 2).

X46 Machine Operating Panel

Logic unit		Connecting Cable Id. Nr. 263 954 ..			MB 420	
D-sub terminal (female) 37-pin	Assignment	D-sub connector (male) 37-pin		D-sub connector (female) 37-pin	D-sub terminal (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	
25	I152	25	Yellow/Brown	25	25	
26	O0	26	Gray/Brown	26	26	
27	O1	27	Yellow/Black	27	27	
28	O2	28	White/Yellow	28	28	
29	O3	29	Gray/Blue	29	29	
30	O4	30	Pink/Blue	30	30	
31	O5	31	Pink/Red	31	31	
32	O6	32	Brown/Blue	32	32	
33	O7	33	Pink/Green	33	33	
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	

3.22 TNC Keyboard

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

X1 on the TNC Keyboard for connecting the soft keys of the Visual Display Unit

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

X45 TNC keyboard

Logic Unit		Connecting Cable Id. Nr. 263 954 ..			TE 401
D-sub terminal (female) 37-pin	**Assignment**	**D-sub conn. (male) 37-pin**		**D-sub connector (female) 37-pin**	**X2 D-sub terminal (male) 37-pin**
1	RL0	1	Gray/Red	1	1
2	RL1	2	Brown/Black	2	2
3	RL2	3	White/Black	3	3
4	RL3	4	Green/Black	4	4
5	RL4	5	Brown/Red	5	5
6	RL5	6	White/Red	6	6
7	RL6	7	White/Green	7	7
8	RL7	8	Red/Blue	8	8
9	RL8	9	Yellow/Red	9	9
10	RL9	10	Gray/Pink	10	10
11	RL10	11	Black	11	11
12	RL11	12	Pink/Brown	12	12
13	RL12	13	Yellow/Blue	13	13
14	RL13	14	Green/Blue	14	14
15	RL14	15	Yellow	15	15
16	RL15	16	Red	16	16
17	RL16	17	Gray	17	17
18	RL17	18	Blue	18	18
19	RL18	19	Pink	19	19
20	SL0	20	White/Gray	20	20
21	SL1	21	Yellow/Gray	21	21
22	SL2	22	Green/Red	22	22
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	27	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

3.23 Visual Display Unit

The soft keys of the visual display unit are connected by flat cable with the TNC keyboard. This cable is included in delivery with the visual display unit.
At the moment we are supplying two types of screens:

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

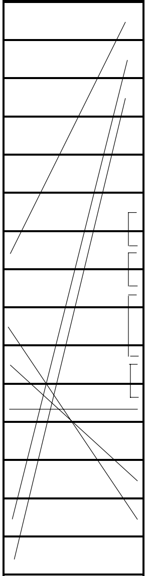
When ordering, make sure that you also order the corresponding logic unit for the screen. You will need the 3-row/2-row adapter, Id. Nr. 313 434 02, for connecting the BC 120 to LEs with Id. Nr. xxx xxx 3x. The predecessor of the BC 120, the BC 110 B (supplied in the past), could be connected directly to LEs with Id. Nr. xxx xxx 3x — no adapter was necessary. The BC 120 can be connected without an adapter to LEs with Id. Nr. xxx xxx 4x. If you want to connect these LEs with old BC 110 B screens, you will need a 3-row/2-row adapter (Id. Nr. 313 434 01).

X43 Visual Display Unit (BC 110 B)

Logic Unit		Connecting Cable Id. Nr. 250 477 ..			BC 110 B
D-sub terminal (female) 15-pin	**Assignment**	**D-sub connector (male) 15-pin**		**D-sub connector (female) 15-pin**	**X1 D-sub terminal (male) 15-pin**
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	GNC	8		8	8
9	VSYNC	9	Yellow	9	9
10	HSYNC	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

The interface complies with the recommendations in IEC 742 EN 50 178 for separation from line power.

X43 Visual Display Unit (BC 120)

Logic unit Id. Nr. xxx xxx 3x		Adapter 313 434 02	Extension cable Id. Nr. 312 878 ..			BC 120
D-sub connector (female) 15-pin 2-row	**Assignment**	**2-row / 3-row**	**D-sub connector (male) 15-pin 3-row**		**D-sub connector (female) 15-pin 3-row**	**D-sub connector (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 Green	7	7
8	GND		8	Coax S Blue	8	8
9	VSYNC		9		9	9
10	HSYNC		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

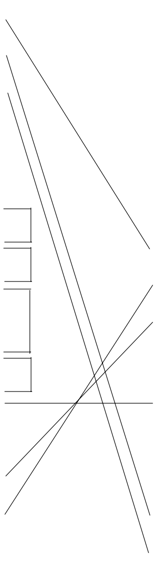
The interface complies with the recommendations in IEC 742 EN 50 178 for separation from line power.

X43 Visual Display Unit (BC 120)

Logic unit **Id.-Nr. xxx xxx 4x**		**Extension cable** **Id.-Nr. 312 878 ..**			**BC 120**
D-sub connector (female) 15-pin 3-row	**Assignment**	**D-sub connector (male) 15-pin 3-row**		**D-sub connector (female) 15-pin 3-row**	**D-sub connector (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 Green	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

The interface complies with the recommendations in IEC 742 EN 50 178 for separation from line power.

X43 Visual Display Unit (BC 110)

Logic unit Id.-Nr. xxx xxx 4x		Extension cable Id.-Nr. 312 878 ..			Adapter 313 434 01	BC 110 B
D-sub connector (female) 15-pin 3-row	Assignment	D-sub connector (male) 15-pin 3-row		D-sub connector (female) 15-pin 3-row	3-row / 2-row	D-sub connector (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 Green	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

The interface complies with the recommendations in IEC 742 EN 50 178 for separation from line power.

X49 Visual Display Unit (BF 120)

Logic unit		Extension Cable Id. Nr. 312 876 ..			Connecting Cable Id. Nr. 312 875 ..			BF 120
D-sub connector (female) 62-pin	Assignment	D-sub connector (male) 62-pin		D-sub connector (female) 62-pin	D-sub connector (male) 62-pin		D-sub connector (female) 62-pin	D-sub connector (male) 62-pin
1	0V	1		1	1		1	1
2	CLK. P	2		2	2		2	2
3	HSYNC	3		3	3		3	3
4	–BLANK	4		4	4		4	4
5	VSYNC	5		5	5		5	5
6	0V	6		6	6		6	6
7	R0	7		7	7		7	7
8	R1	8		8	8		8	8
9	R2	9		9	9		9	9
10	R3	10		10	10		10	10
11	0V	11		11	11		11	11
12	G0	12		12	12		12	12
13	G1	13		13	13		13	13
14	G2	14		14	14		14	14
15	G3	15		15	15		15	15
16	0V	16		16	16		16	16
17	B0	17		17	17		17	17
18	B1	18		18	18		18	18
19	B2	19		19	19		19	19
20	B3	20		20	20		20	20
21	0V	21		21	21		21	21
22	0V	22		22	22		22	22
23	–CLK. P	23		23	23		23	23
24	–HSYNC	24		24	24		24	24
25	BLANK	25		25	25		25	25
26	–VSYNC	26		26	26		26	26
27	0V	27		27	27		27	27
28	–R0	28		28	28		28	28
29	–R1	29		29	29		29	29
30	–R2	30		30	30		30	30
31	–R3	31		31	31		31	31
32	0V	32		32	32		32	32
33	–G0	33		33	33		33	33
34	–G1	34		34	34		34	34
35	–G2	35		35	35		35	35

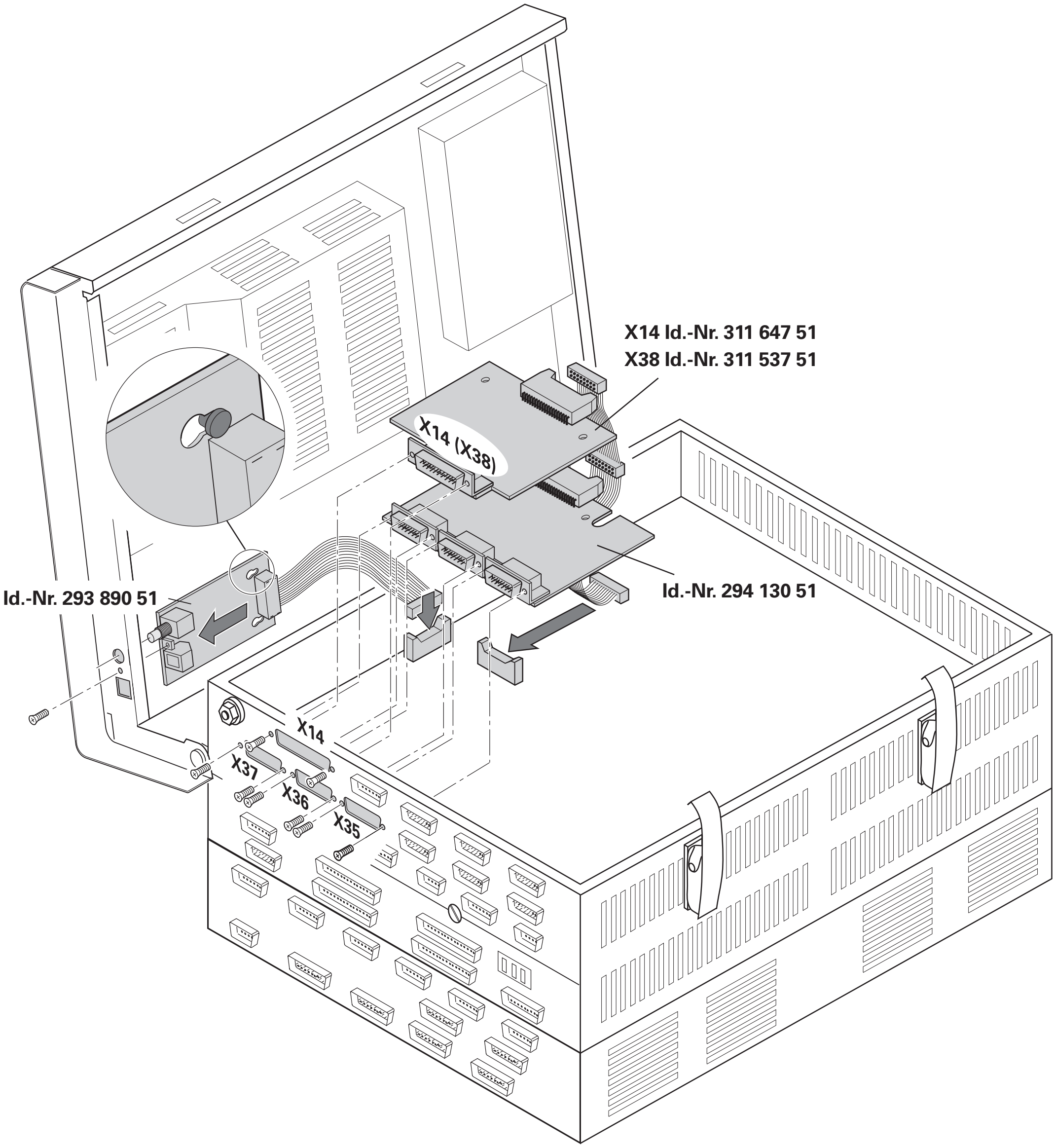
Logic unit		**Extension Cable Id. Nr. 312 876 ..**			**Connecting Cable Id. Nr. 312 875 ..**			**BF 120**
D-sub connector (female) 62-pin	**Assignment**	**D-sub connector (male) 62-pin**		**D-sub connector (female) 62-pin**	**D-sub connector (male) 62-pin**		**D-sub connector (female) 62-pin**	**D-sub connector (male) 62-pin**
36	–G3	36		36	36		36	36
37	0V	37		37	37		37	37
38	–B0	38		38	38		38	38
39	–B1	39		39	39		39	39
40	–B2	40		40	40		40	40
41	–B3	41		41	41		41	41
42	0V	42		42	42		42	42
43	–DISP.LOW	43		43	43		43	43
44	DISP.LOW	44		44	44		44	44
45	–DISP. ON	45		45	45		45	45
46	DISP.ON	46		46	46		46	46
47	C0	47		47	47		47	47
48	C1	48		48	48		48	48
49	C2	49		49	49		49	49
50	C3	50		50	50		50	50
51	C4	51		51	51		51	51
52	C5	52		52	52		52	52
53 to 62	*Do not assign*	53 to 62		53 to 62	53 to 62		53 to 62	53 to 62
Housing		Housing		Housing	Housing		Housing	Housing

The interface complies with the recommendations in IEC 742 EN 50 178 for separation from line power.

3.24 Mounting the Optional PCBs

The optional PCBs are delivered separately and you must incorporate them into your logic unit.

Mounting must be overseen by adequately trained personnel.



4 Machine Integration

4.1 Machine Axes

The number of machine axes that can be controlled depends on the hardware (see "Specifications"). With machine parameter MP10 you can define which axes on the machine are to be put into operation. If necessary you can use this parameter to deselect all functions for the axes (control, display, reference mark traversing, etc.).

MP10	Active axes		
Input:	%xxxxxxxxx		
	Bit	0 to 8	0 = not active
	Axis	1 to 9	1 = active

Screen display

With MP7291.x you can select the line on which the axis is to be displayed on the screen. You must set the axes to be displayed in MP100.x (capital letters for NC axes and small letters for PLC axes). You can define a new arrangement for each traverse range. The spindle is always displayed on the last line.

MP7291.0-3 Display on screen

Input: SXYZABCUVWxyzabcuvw- (capital letters for NC axes, small letters for PLC axes, hyphen for non-displayed axes)

Character	1 to 9 (1= right)
Line	1 to 9

Character 10 = spindle S (display always in line 9)

MP7291.0	Traverse range 1
MP7291.1	Traverse range 2
MP7291.2	Traverse range 3

4.1.1 Encoders

HEIDENHAIN contouring controls can be connected to incremental position measuring systems.

Signal period

The signal period is calculated by the TNC automatically. To make this possible, however, you must enter a distance (MP331.x) and the number of signal periods it covers (MP332.x).

$$\text{Signal period} = \frac{\text{MP331.x}}{\text{MP332.x}}$$

Encoder output signals with 1 V_{PP} and 7 to 16 μA_{PP} are interpolated 1024-fold.

MP331.0-8 Distance covered by the signal periods entered in MP332
Input: 0 to 99 999.9999 [mm or °]

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

Traverse direction

With machine parameters MP210 and MP1040 you define the traversed directions of the axes. The traverse directions for numerically controlled machine tools are described in ISO 841.

With MP 210 you define the counting direction of the measuring signals from the position encoder. The counting direction depends on the mounting arrangement of the encoder.

With MP1040 you define the polarity of the nominal value for the positive traverse direction. If MP1040 and MP210 do not match, the error message **`Movement monitoring error in <axis> C`** is displayed.

With W1030 the NC informs the PLC the direction in which the axes are being moved.

Digital axes: The counting direction of the measuring signals from the speed encoder is set in the motor table. If the error message **`Standstill monitoring error in <axis> D`** appears, you must change this value.

MP210 Counting direction of the measuring signals from the position encoder.
Input: %xxxxxxxxx
Bit 0 to 8 0 = positive
Axis 1 to 9 1 = negative

MP1040 **Analog axes:** Polarity of the nominal value voltage for the positive traverse direction
Digital axes: Polarity of the speed nominal value for the positive traverse direction
Input: %xxxxxxxxx
Bit 0 to 8 0 = positive
Axis 1 to 9 1 = negative

				Set	Reset
W1030	Current traversing direction			NC	NC
	Bit	0 to 8	0 = positive		
	Axis	1 to 9	1 = negative		

Encoder monitoring
HEIDENHAIN contouring controls monitor the quality of the signal transmission from the encoders.

Condition	Error message
Absolute position with distance-coded reference marks	**Measuring system <axis> defective**
Amplitude of encoder signals	**Encoder amplitude too small <axis>**
Edge separation of encoder signals	**Encoder <axis>: frequency too high**

Position encoder monitoring can be activated via machine parameter. The TNC automatically compensates offset of the encoder signals.

Digital axes:
The monitoring function for speed encoders is always active. If an error message refers to a speed encoder, an ` is displayed after the axis designation (e.g. **Encoder <axis>`: frequency too high**). The absolute position with distance-coded reference marks is not checked for speed encoders.

MP20 Checking the absolute position of distance-coded reference marks for the axes
Input: %xxxxxxxxx
Bit 0 to 8 0 = not active
Axis 1 to 9 1 = active

MP20.1 Monitoring the amplitude of the encoder signals for the axes
Input: %xxxxxxxxxx
Bit 0 to 8 0 = not active
Axis 1 to 9 1 = active

MP20.2 Checking the edge separation of the encoder signals for the axes
Input: %xxxxxxxxxx
Bit 0 to 8 0 = not active
Axis 1 to 9 1 = active

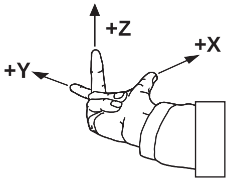
MP21.0 Checking the absolute position of distance-coded reference marks for the spindle
Input: %x 0 = not active
1 = active

MP21.1 Monitoring the amplitude of the encoder signals for the spindle
Input: %x 0 = not active
1 = active

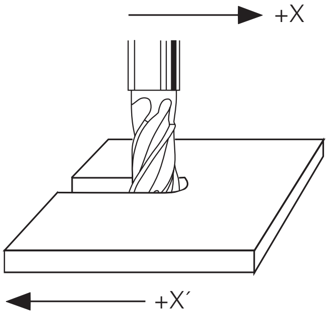
MP21.2 Checking the edge separation of the encoder signals for the spindle
Input: %x 0 = not active
1 = active

4.1.2 Axis Designation

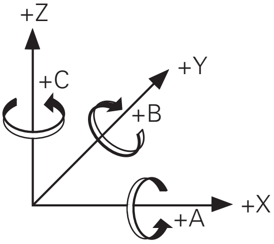
The coordinate axes and their directions of traverse are defined in the international standard ISO 841. An easy way to remember this system is to use the “right-hand rule.”



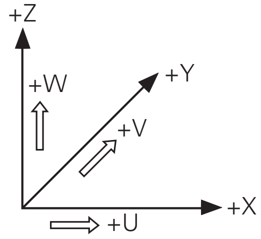
For the spindle axis direction, a movement of the tool toward the workpiece represents the negative direction. When writing an NC program, the programmer always enters the sign for traverse direction as if the tool is moving and the workpiece is stationary. If the machine table moves in a particular axis, then the direction of actual motion is opposite to the direction of axis traverse. Such positive relative axis movements are then designated with an index (+X', +Y' etc.).



An **axis of rotation** is designated by the letter A, B or C. The correlation of rotary axes with the primary axes and determination of the direction of rotation are standardized in ISO 841.



A **secondary linear axis** is designated by the letter U, V or W. The correlation of secondary axes with the primary axes and their directions of traverse are also standardized in ISO 841.



With MP410.x you can determine which axes are assigned to the keys IV and V on the keyboard and the HR 410. The other axes are assigned to soft keys or ASCII characters. The axis designations are set with MP100.x. Capitals are used for NC axes and small letters or a hyphen (-) for PLC axes. Each of the three traverse ranges can have different axis designations, see also "Screen display." Module 9151 is used to overwrite the designations set in MP100.x (see "Traverse Ranges").

MP410.3-4	Designation for the axis keys IV and V
Input:	A, B, C, U, V, W, a, b, c, u, v, w

MP410.3	Axis key IV
MP410.4	Axis key V

MP100.0-2	Designation for the axes		
Input:	XYZABCUVWxyzabcuvw-		(capitals for NC axes, small letters or hyphen for PLC axes)
	Character	1 to 9 (1 = right)	
	Line	1 to 9	

MP100.0	Traverse range 1
MP100.1	Traverse range 2
MP100.2	Traverse range 3

4.1.3 Assignment of Axes

With MP110.x you can assign the position encoder inputs to the individual axes, and with MP111 a position encoder can be assigned to the spindle.

Analog axes: With MP120.x the nominal speed command outputs can be assigned to the axes. If you do not assign a nominal speed output to an axis, this axis can only be displayed and not controlled. You can assign a nominal speed command output to the spindle with MP121.

Digital axes: You can define digital axes by entering D1 to D6 in MP120. To define a digital spindle enter S1 in MP121. If you enter zero in MP110.x or MP111, the speed encoder will be used for determining position.

MP110.0-8 Assignment of the position encoder inputs to the axes
Input: 0 = no position encoder input
1 = position encoder input X1
2 = position encoder input X2
3 = position encoder input X3
4 = position encoder input X4
5 = position encoder input X5
6 = position encoder input X6
35 = position encoder input X35
36 = position encoder input X36
37 = position encoder input X37
38 = position encoder input X38

MP111 Position encoder input for the spindle
Input: As MP 110

MP120.0-8 Assignment of the nominal speed command outputs to the axes
Input: 0 = non-controlled axis

A1 to A13 or 1 to 13 =	Analog axis with analog output 1 to 13 (analog outputs 1 to 6 on connection X8 analog outputs 7 to 13 on connection X9)
D1 to D6 =	Digital axis 1 to 6

NC software 280 470 ..: Only 0 to 13 and definition of digital axes in MP2000

MP121 Assignment of the nominal speed command outputs to the spindle
Input: 0 = non-controlled spindle

A1 to A13 or 1 to 13 =	Analog spindle with analog output 1 to 13 (analog outputs 1 to 6 on connection X8 analog outputs 7 to 13 on connection X9)
S1 =	Digital spindle

NC software 280 470 ..: Only 0 to 13 and definition of digital spindle in MP2001

4.1.4 Reading axis information

Reading general axis information

With Module 9038 you can request general axis status information. This saves you calling up individual machine parameters. The number of the axis and the number of the desired status information must be entered. If you enter -1 for the axis, you receive the status information for all of the axes in bit-coded form. In this case bit 0 to bit 8 correspond to axes 1 to 9, and bit 15 is the spindle. Please note that when you request the status information for a particular axis (transfer values 0 to 8 and 15), only bit 0 is changed.

Status information	Meaning
0	0 = axis (spindle) not active (MP10 or MP3010 or no encoder) 1 = axis (spindle) active
1	Dependent on current traverse range: 0 = NC axis or not active 1 = PLC axis
2	0 = non-controlled axis (spindle), display only or not active 1 = controlled axis (spindle)
3	Maximum motor temperature [°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, spindle = 15
                                      bit-coded = -1
PS   B/W/D/K   <Status information>   [see table for numbers]
CM   9038
PL   B/W/D     <Information>          [see table]
```

```
Error recognition: M4203 = 0:    Information was read
                   M4203 = 1:    Error code in W1022
                   W1022 = 1:    Status information not available on this TNC
                   W1022 = 2:    Axis not available
```

Current Tool Axis

In HEIDENHAIN dialog programming, enter the tool axis (X, Y, Z, U, V, W) in the NC block TOOL CALL. In ISO programming, you can define the tool axis with G17 to G20. The tool axis for G20 is defined in MP410.3. In the PLC you can investigate the current tool axis via markers.

		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

4.1.5 Traverse Ranges

Three different traverse ranges can be defined by machine parameters (e.g. for reciprocal machining). With MP100.x you can set an axis designation for each traverse range, and with MP7291.x you can define which axes will be displayed for each traverse range.

The traverse ranges are defined by so-called software limit switches. The input values for the software limit switches are referenced to the machine datum (MP960.x). The software limit switches for rotary axes are effective only if the value 0 is entered in MP810. The MOD function LIMIT SWITCH can be used to enter an additional limitation for each traverse range.

If one of the software limit switches is reached, the error message **`Limit switch <axis>`** appears and the NC reports to the PLC in words W1034 and W1036 that a software limit switch has been traversed.

In MP7490 you select whether one or three traverse ranges can be defined with the MOD function. With MP7490 you can also select whether one datum applies for all traverse ranges or each traverse range can have a separate datum.

Selecting traverse range with M4135

With markers M4574 and M4575 you select the current traverse range, and you activate it with the strobe marker M4135. This strobe-marker is reset by the NC after the change-over has been carried out. In the "Manual" and "Electronic handwheel" modes of operation you can always switch ranges; in all other modes you can switch only together with an M/S/T/Q strobe.

M4574	M4575	Traverse range / Datum
0	0	Range 1
1	0	Range 2
0	1	Range 3

Selecting traverse range and axis designation with Module 9151

With Module 9151 you can select one of the three traverse ranges and the new axis designations you assign it. Module 9151 is used to overwrite the axis designations set in MP 100.x.
The same conditions hold as when selecting the traverse range with M4135. M4135 is set when the module is called and is reset by the NC once the selection has been made. The same syntax is used for inputting the new axis designations as for the input in MP100.x.

Call:

```
PS   B/W/D/K/S  <String with axis designation>
                –1 = use axis designation from MP100.x
PS   B/W/D/K    <Traverse range>
                0 to 2 =   traverse range
                –1 =       do not change traverse range
CM   9151
```

Error recognition: M4203 = 0: Traverse range or axis designation changed
M4203 = 1: Error code in W1022
W1022 = 2: Value for the traverse range was not valid
W1022 = 3: Neither a string or –1 was transferred for the axis designation.
W1022 = 21: The module was called with the program already started or without a M/S/T/Q strobe being called

Special case: Overwriting the software limit switch with FN17:SYSWRITE

With FN17.SYSWRITE ID230 you can overwrite the current software limit switches. You can use this function for example in conjunction with an automatic tool change, see Chapters "Tool Changer" and "PLC Programming."

The software limit switches set with FN17 are only active until the end of the program or until the next GOTO function. Then the original limit switches become active once again.

MP910.0-8 Positive software limit switch for traverse range 1; default setting after power-on;
Activation via PLC M4575 = 0, M4574 = 0
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
(Input values are referenced to the machine datum)

MP911.0-8 Positive software limit switch for traverse range 2
Activation via PLC M4575 = 0, M4574 = 1
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
(Input values referenced to the machine datum)

MP912.0-8 Positive software limit switch for traverse range 3
Activation via PLC: M4575 = 1, M4574 = 0
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
(Input values referenced to the machine datum)

MP920.0-8 Negative software limit switch for traverse range 1; Default setting after power-on;
Activation via PLC M4575 = 0, M4574 = 0
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
(Input values referenced to the machine datum)

MP921.0-8 Negative software limit switch for traverse range 2
Activation via PLC M4575 = 0, M4574 = 1
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
(Input values referenced to the machine datum)

MP922.0-8 Negative software limit switch for traverse range 3
Activation via PLC: M4575 = 1, M4574 = 0
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
(Input values referenced to the machine datum)

MP7490 Functions for traverse ranges
Input: %xxxx
Bit 0 Traverse ranges displayed via MOD
0 = one traverse range
1 = three traverse ranges
Bit 1 Number of references points
0 = one reference point for each traverse range
1 = one reference point for all traverse ranges

		Set	Reset
M4574	Selection of the traverse range	PLC	PLC
M4575	Selection of the traverse range	PLC	PLC
M4135	Activation of the selected traverse range	PLC	PLC
W1034	Positive software limit switch was traversed Bit 0 to 8 Axis 1 to 9	NC	NC
W1036	Negative software limit switch was traversed Bit 0 to 8 Axis 1 to 9	NC	NC

4.1.6 Lubrication Pulse

You can use the PLC to control the lubrication of the guideway according to the distance traveled on each axis. In the machine parameter MP4060.X you register the distance after which lubrication should be performed for each axis. When the accumulated traverse is reached in a particular axis, the NC reports this to the PLC with word W1056. After the axis is lubricated, the accumulated distance must be reset in the PLC with W1058.

MP4060.0-8 Path-dependent lubrication
Input: 0 to 99 999.9999 [mm]

		Set	Reset
W1056	Lubrication pulse. Value from MP4060.x exceeded Bit 0 to 8 Axis 1 to 9	NC	NC
W1058	Reset the accumulated distance Bit 0 to 8 Axis 1 to 9	PLC	PLC

4.2 PLC Axes

The controlled axes can also be assigned to the PLC individually. With MP100.x you can define which axes are to be controlled by the NC and which ones will be controlled by the PLC:

- Axis designation in capitals = NC axis
- Axis designation in small letters or hyphen = PLC axis
- A hyphen is used when the PLC axis is not to be displayed.

PLC axes are always controlled in servo lag mode. All other axis-specific machine parameters have the same function as for NC axes. You can start several PLC axes simultaneously. However, these axes are not interpolated with each other.

In MP810.x you can enter a modulo value for the counting mode of rotary axes (see the section "Display and Operation"). If you have entered a modulo value in MP810.x, the PLC axis will always be moved by the shortest path to the target position, even if this means crossing the zero point. You program the positioning of PLC axes in the PLC program using the following modules:

- Module 9120: Start PLC axis
- Module 9121: Stop PLC axis
- Module 9122: Status of PLC axis
- Module 9123: Traverse the reference marks for PLC axis
- Module 9124: Feed rate override for PLC axis

MP100.0-2 Designation of the axes
Input: XYZABCUVWxyzabcuvw- (capital letters for NC axes, small letters or hyphen for PLC axes)
Byte 0 to 8
Axis 1 to 9

MP100.0	Traverse range 1
MP100.1	Traverse range 2
MP100.2	Traverse range 3

Start PLC axis (Module 9120)
With Module 9120 you start the positioning of a PLC axis, transferring the following parameters:

- The axis to be positioned
- Target position
- Feed rate
- Absolute or incremental position

The axis is positioned fully independently of any other processes in the control. There is no path interpolation with other axes.

Constraints:

- You must activate the axis using MP10 and identify it as a PLC axis using MP100.x.
- The system does not check for limit switch overshoot.
- The axis must be stationary. You must abort any positioning movement beforehand with Module 9121.
- There is no feed-rate override. You can influence the feed rate with Module 9124.
- If the reference marks were not yet traversed, the positioning movement starts from the counter value as it existed upon switch-on.

```
Call:
PS   B/W/D/K    <Axis>                [0 to 8]
PS   B/W/D/K    <Target position>     [0.0001mm]
PS   B/W/D/K    <Feed rate>           [mm/min]
PS   B/W/D/K    <Mode>
                                   Bit 0: Type of target position data
                                   =0:    Absolute, referenced to the machine datum
                                   =1:    Incremental
CM   9120
PL   B/W/D      <Error code>       0:     No error. Positioning is started
                                   1:     Axis does not exist
                                   2:     Axis is not defined as a PLC axis
                                   3:     The axis is already being positioned
                                   4:     Absolute position is outside of the modulo range
```

Stop the PLC axis (Module 9121)

With Module 9121 you can abort anywhere a PLC axis positioning command that has already started.

```
Call:
PS   B/W/D/K    <Axis>             [0 to 8]
CM   9121
PL   B/W/D      <Error code>       0:     Positioning command is aborted
                                   1:     Axis does not exist
                                   2:     Axis is not defined as a PLC axis
                                   3:     Axis was already stationary
```

Status of PLC axis (Module 9122)

With Module 9122 you can interrogate the status of the PLC positioning axis movement. Changes in status due to commands to the controller of the PLC axes (Modules 9120, 9121, 9123) are not recognized until the next PLC scan.

```
Call:
PS   B/W/D/K    <Axis>      [0 to 8]
CM   9122
PL   B/W/D      <Status>    Bit 0:  Type
                                   =0:    NC axis
                                   =1:    PLC axis
                            Bit 1:  Reference mark
                                   =0:    Reference mark has not yet been traversed
                                   =1:    Reference mark has been traversed
                            Bit 2:  Positioning
                                   =0:    Not active
                                   =1:    Active
                            Bit 3:  Direction of traverse
                                   =0:    Positive
                                   =1:    Negative
                            Bit 4:  Positioning error
                                   =0:    No error
                                   =1:    Positioning error
```

Traverse the reference marks for PLC axes (Module 9123)

You can use the same procedures to traverse the reference marks for PLC axes as for NC axes (see section "Reference Marks"). If you cannot use any of the known procedures, simply use this module to program your own procedure for traversing the reference marks. The module starts positioning a PLC axis in a defined direction and continues until a reference mark is found. Due to the deceleration path, the axis will stop at a position a small distance off the reference mark.

```
Call:
PS   B/W/D/K    <Axis>          [0 to 8]
PS   B/W/D/K    <Feed rate>     [mm/min]
PS   B/W/D/K    <Mode>
                                Bit 0: Traverse direction
                                =0:   positive
                                =1:   negative
CM   9123
PL   B/W/D      <Error code>    0:    No error. Positioning has started
                                1:    Axis does not exist
                                2:    Axis is not defined as a PLC axis
                                3:    Axis is already being positioned
```

Feed rate override for PLC axis (Module 9124)

With Module 9124 you can set the feed rate override for a PLC axis. The override value can lie between 0% and 100.00% (resolution 0.01%) and must be transmitted as an natural number (0 .. 10 000). The last transmitted override value is accounted for at the beginning of movement. After a reset or interruption of the PLC program the override value of all PLC axes is set to 100.00%. The feed rate override can also be changed while a PLC axis is being positioned.

```
Call:
PS   B/W/D/K    <Axis>          [0 to 8]
PS   B/W/D/K    <Override>      (Format 0.01%)
CM   9124
PL   B/W/D      <Error code>    0:    No error: feed rate override was set.
                                1:    Axis does not exist
                                2:    Axis is not defined as a PLC axis
                                3:    Override value is incorrect
```

4.3 Axis Error Compensation

HEIDENHAIN contouring controls can compensate for error resulting from mechanical imperfections in the machine.

The following axis-error compensation is possible:

- Backlash compensation
- compensation of reversal errors in circular motion
- Compensation of static friction
- Linear axis-error compensation
- Nonlinear axis-error compensation
- Compensation of thermal expansion

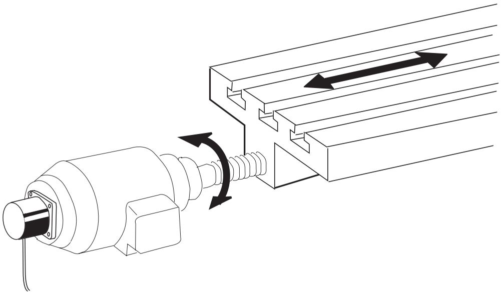
You can activate either linear or nonlinear axis-error compensation. The other types of error compensation can always be activated in parallel.

4.3.1 Backlash Compensation

Cause outside the controlled system

If linear traverse is measured by ballscrew and rotary encoder, a small amount of play between the movement of the table and that of the rotary encoder can result during reversals in traverse direction.

Depending on the design, the movement of the rotary encoder may be advanced or retarded relative to the table. The machine tool trade describes this as positive or negative backlash. To compensate this backlash, the TNC adds or subtracts the value from MP710.x to the encoder signals with every direction change.



Positive backlash:	rotary encoder advanced relative to the table (traverse movement of the table is too short).
Negative backlash:	rotary encoder retarded relative to the table (traverse movement of the table is too long).

MP710.0-8 Backlash compensation
Input: – 1.0000 to + 1.0000 [mm] or [°]

Cause within the controlled system
As of NC software 280 470 08 and 280 472 01.
With MP750 and MP752 the TNC can compensate backlash within the controlled loop. This means it is also possible to compensate backlash between motor and table movement in direct distance measurement with linear encoders. In this process, the reversal peaks in circular paths are also compensated, with the result that MP711 to MP716 are no longer necessary.

Enter the backlash in MP750 in mm. In MP752 enter the time in which the path to be compensated will be traversed.
Example:
MP750 = 0.03 mm, MP752 = 15 ms
At every direction change a nominal speed value is output for 15 ms, which corresponds to a feed rate of 120 mm/min (0.03 mm / 15 ms = 0.002 m/s = 120 mm/min).

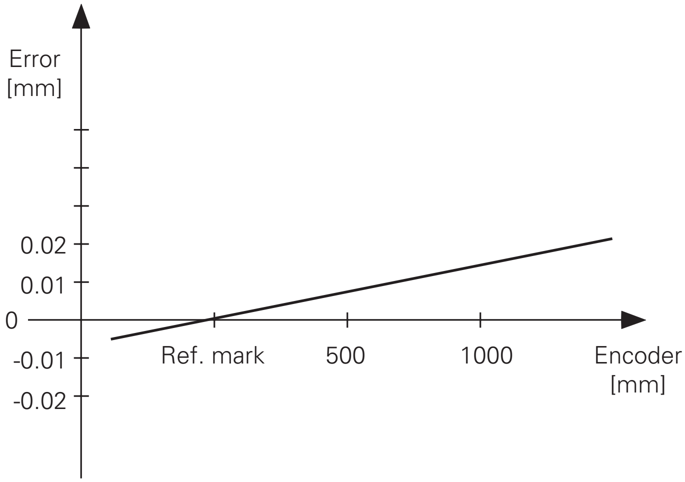
MP750.0-8	Backlash
Input:	–1.0000 to +1.0000 [mm]

MP752.0-8	Compensation time for value from MP750.x
Input:	0 to 1000 [ms]

4.3.2 Linear Axis-Error Compensation

One linear axis-error can be compensated per axis. You enter the axis error, with the correct sign, in MP720. The error is positive if the table travel is too long, and negative if the travel is too short.

With MP730 you set the axis error compensation to linear or nonlinear. Linear axis error compensation is not active for rotary axes.



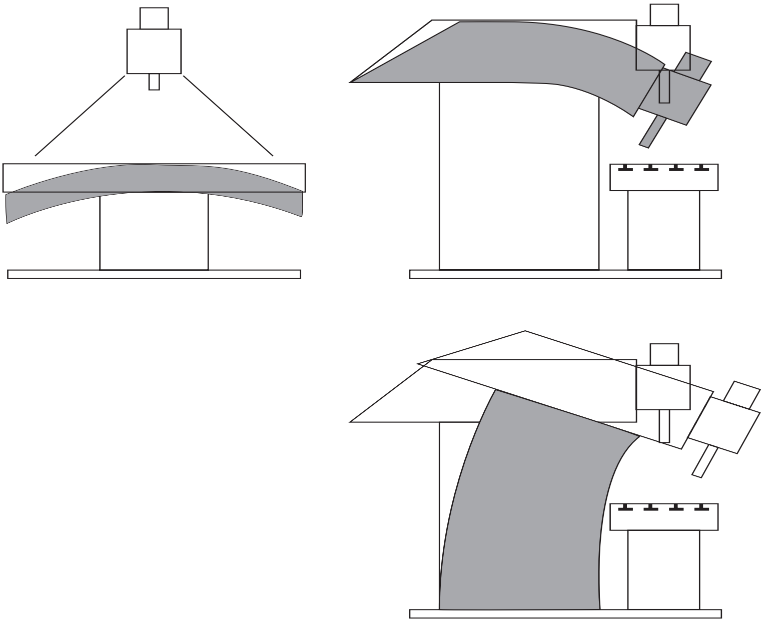
MP720.0-8	Linear axis-error compensation
Input:	– 1.000 to + 1.000 [mm/m]

MP730 Selection of linear or nonlinear axis-error compensation
Input: %xxxxxxxxx

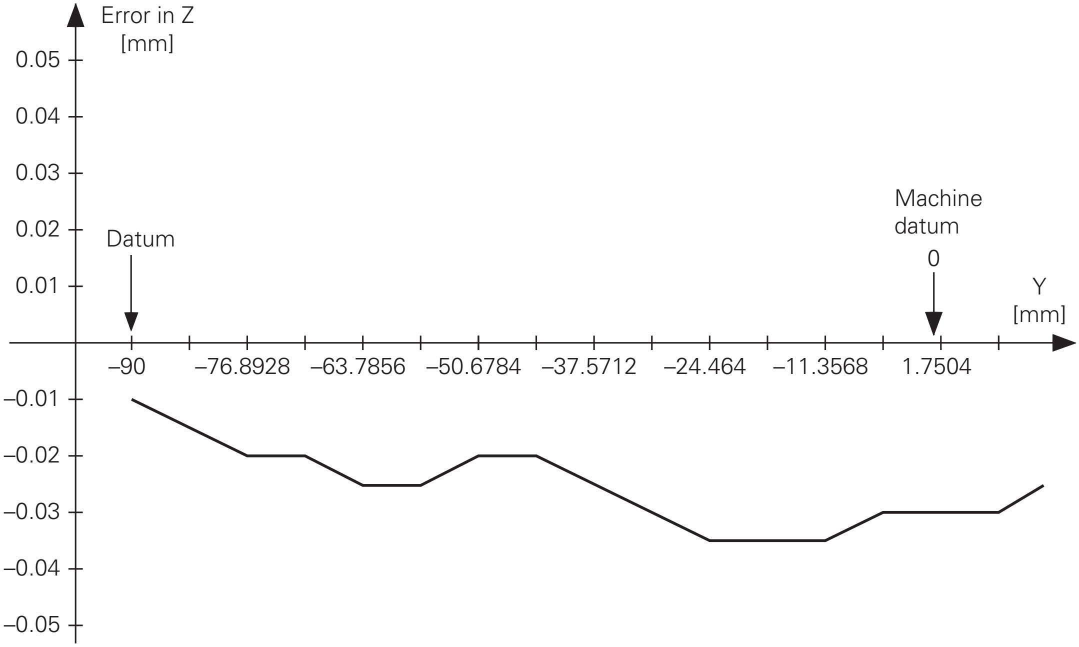
Bit	0 to 8	0 = linear axis error compensation
Axis	1 to 9	1 = nonlinear axis error compensation

4.3.3 Nonlinear Axis Error Compensation

Depending on the design of the machine or external factors (e.g. temperature) a nonlinear axis-error can occur.



Such an axis-error is usually determined by a comparator measuring instrument (e.g. the VM 101 from HEIDENHAIN).
Error trace for sag as a function of Y (Z = F(Y)):

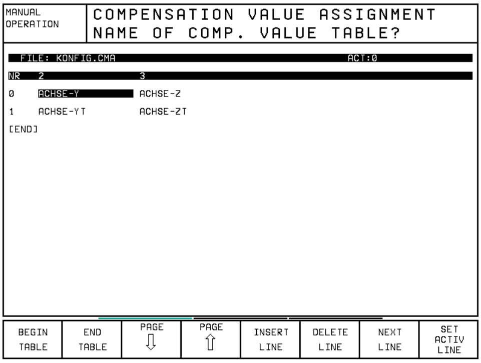


The TNC can compensate the error in ballscrew pitch and sag at the same time.

Inputting the error trace into the TNC:

- In MP730 activate nonlinear axis-error compensation
- Enter the code number 807667.
- Create a configuration file type .CMA using the program manager
- Using soft keys activate columns for the axes to be included in the compensation value table
- Enter the compensation value table:
  You can assign different compensation value tables to each axis (e.g. temperature-dependence).
  The name of the table must be input in various lines.
  Select the current line via soft key or with Module 9095. You can interrogate the current active line with Module 9035.

  Example: Axis 2 = Y
  Axis 3 = Z
  Compensation value table at 20°C = AXIS-Y.COM and AXIS-Z.COM
  Compensation value table at 35°C = AXIS-YT.COM and AXIS-ZT.COM



```
MANUAL       COMPENSATION VALUE ASSIGNMENT
OPERATION    NAME OF COMP. VALUE TABLE?

  FILE: KONFIG.CMA                                  ACT:0
NR   2                  3
0    ACHSE-Y            ACHSE-Z
1    ACHSE-YT           ACHSE-ZT
[END]
```

- Enter the complete name of the configuration file with path in the OEM.SYS file using the command `TABCMA =`
- Create a compensation value table type .COM using the program manager
- Using soft keys enter columns for the dependencies of the axis for which you have created the compensation table
- Press HEAD LINE soft key
- Enter the datum for the compensation values. Input value = distance from machine datum (MP960.x)
- Enter the distance of the compensation points. Input value = exponent to the base 2 (e.g. input 16 = 2^{16} = 6.5536 mm). Maximum input value = 23
- Press the END key to exit the header
- With the soft key APPEND N LINES enter the number of compensation points.
  A maximum of 256 compensation points can be entered per column; a maximum of 10 columns in all active compensation value tables; in total not more than 1280 compensation points

➤ Enter the compensation values. You need only enter the kink points on the error curve. The TNC automatically performs a linear interpolation between the kink points.

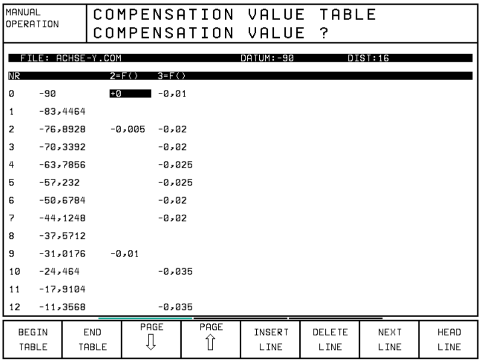
Example: Axis 2 = Y
Axis 3 = Z
Ballscrew pitch error in Z and Y (Z = F(Z) and Y = F(Y))
Sag in Z as dependency of Y.
Traverse range: Z = 800 mm, Y = 500 mm.
Datum for the compensation values: Z = –90 mm, Y = –200 mm.
Desired distance of compensation points: 7 mm

Distance of the compensation points: possible power of two = 2^{16} = 6.5536 mm

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

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

In the compensation value table for axis Y enter the ballscrew pitch error in the column 2 = F( ) and the sag in the column 3 = F( )



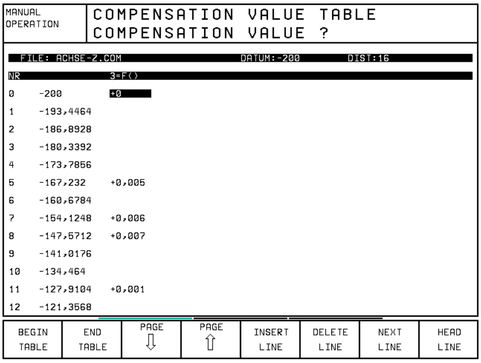
MANUAL OPERATION

COMPENSATION VALUE TABLE
COMPENSATION VALUE ?

FILE: ACHSE-Y.COM DATUM:-90 DIST:16

NR		2=F()	3=F()
0	-90	+0	-0,01
1	-83,4464		
2	-76,8928	-0,005	-0,02
3	-70,3392		-0,02
4	-63,7856		-0,025
5	-57,232		-0,025
6	-50,6784		-0,02
7	-44,1248		-0,02
8	-37,5712		
9	-31,0176	-0,01	
10	-24,464		-0,035
11	-17,9104		
12	-11,3568		-0,035

In the compensation value table for axis Z enter the ballscrew pitch error in the column 3 = F( )



MANUAL OPERATION

COMPENSATION VALUE TABLE
COMPENSATION VALUE ?

FILE: ACHSE-Z.COM DATUM:-200 DIST:16

NR		3=F()
0	-200	+0
1	-193,4464	
2	-186,8928	
3	-180,3392	
4	-173,7856	
5	-167,232	+0,005
6	-160,6784	
7	-154,1248	+0,006
8	-147,5712	+0,007
9	-141,0176	
10	-134,464	
11	-127,9104	+0,001
12	-121,3568	

Input and output of the compensation value tables via the data interface

In the PLC Programming mode you can upload and download the .CMA and .COM files via the data interface. The .CMA file is assigned the extension .S and the .COM file the extension .V.

A rotary axis is a special case

With an axis of rotation, the system only recognizes corrections of entries from 0° to 360° (with reference to the machine datum). The datum for nonlinear compensation must be located in the range +0° to +360°. If you want to compensate a full circle, the datum must be identical with the machine datum.

MP730 Selection of linear or nonlinear axis error compensation

Input:	%xxxxxxxxx		
	Bit	0 to 8	0 = linear axis error compensation
	Axis	1 to 9	1 = nonlinear axis error compensation

Call only in the submit job:

```
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 (.COM) does not exist
                3:      Compensation value table (.COM) >256 entries
                4:      Total permissible number of compensation points exceeded
                5:      Too many compensation value tables (.COM)
                6:      .CMA file does not exist
                7:      Call did not come from a submit job
                8:      Call during running program without strobe
                10:     .CMA file is protected
```

Call:

```
PS   B/W/D/K    <19>
CM   9035
PL   B/W/D      <Active line number>
                ≥0:     Active line number
                –1:     No .CMA file active
```

4.3.4 Temperature Compensation

Exact measurements of machine thermal behavior (center of expansion in the axes, amount of expansion as a function of temperature) are necessary to compensate the effects of thermal expansion.

The temperature measured through the Pt 100 thermistors is filed in PLC words (see section “Analog Inputs”). The thermal expansion is largely proportional to the temperature value: you can directly determine the amount of expansion by multiplying the temperature value by a certain factor. The value to be compensated can be given with Module 9231. As soon a you transfer a value with Module 9231, the “lag tracking” becomes active. “Lag tracking” means that the actual machine position changes by a certain value per PLC cycle until it has been completely compensated. This does not change the value in the actual position display. The increment of change per PLC cycle must be defined in MP4070.

You don't have to use Module 9231 for axes 1 to 5. In this case the value to be compensated can be entered directly into W576 to W584.

MP4070	Compensation per PLC cycle for lag-tracking error compensation
Input:	0.0001 to 0.005 [mm]

		Set	Reset
W576 to W584	Lag-tracking error compensation (compensation speed from MP4070) Input: +32 767 to –32 768 [1/10 µm]	PLC	PLC

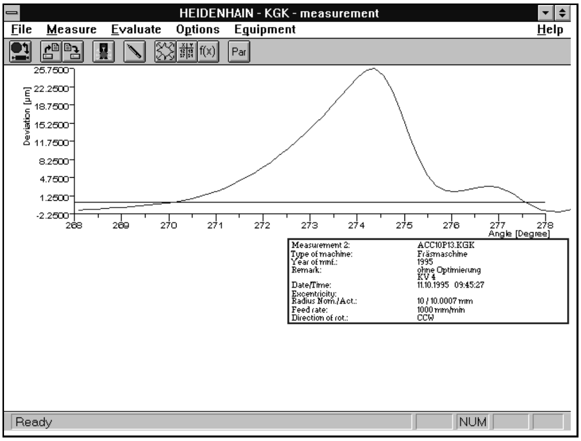
W576	to	W584
Axis 1	to	Axis 5

Call:

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

4.3.5 Compensation for Reversal Errors in Circular Motion

Static friction in the axis bearings can lead to reversal peaks at the quadrant transitions during circular movements. Commercially available devices such as the HEIDENHAIN KGM grid encoder and the HEIDENHAIN ACCOM evaluation software can diagnose these errors and determine the size and duration of their peaks.



Measurement 2:	ACC10P13.KGK
Type of machine:	Fräsmaschine
Year of mnf.:	1995
Remark:	ohne Optimierung KV 4
Date/Time:	11.10.1995 09:45:27
Excentricity:	
Radius Nom./Act.:	10 / 10.0007 mm
Feed rate:	1000 mm/min
Direction of rot.:	CCW

Digital axes
For digital axes you must enter the friction compensation within the range of the speed controller (MP2610.x to MP2620.x). You cannot use MP711.x to MP716.x.

Analog axes
If you have compensated the backlash using MP750.x, there should be no more occurrence of reversal peaks. If reversal peaks can however still be measured, you can try to compensate them with MP711.x to MP716.x. In MP711.x you enter the height of the reversal peaks, in MP712.x the share of the reversal peaks to be compensated per control-loop cycle time.

Calculate the input value for MP712.x as follows:

- Duration of the reversal peak:

$$t_{RP}\ [\text{s}] = \frac{\text{Peak width } [°] \times 2\pi \times \text{Radius [mm]} \times 60}{360[°] \times \text{Feed rate [mm/min]}}$$

  The peak width [°] can be seen in the diagram;
  The feed rate [mm/min] is the programmed contouring feed rate.

- Compensation per control loop cycle time (3 ms)

$$\text{Compensation [mm]} = \frac{\text{Reversal peaks [µm]} \times \text{Control-loop cycle time [s]}}{0.5 \times t_{RP}\ [\text{s}]} \cdot 10^{-3}$$

  The compensation value is entered in MP712.x.

If the calculated values have no effect, this indicates that the machine dynamics are too weak. With the M function M105 you can switch to a second set of K_v factors (see section “Control loop” in this chapter). This activates a second set of machine parameters for compensating the reversal peaks: MP715.x (like MP711.x) and MP716.x (like MP712.x). By selecting a higher K_v factor you can selectively increase the contouring accuracy. The M function M106 reactivates the original set of K_v factors. You must enable the M functions M105/M106 with MP7440, Bit 3.

MP711.0-8 **Analog axes:** Height of the reversal peaks during circular traverse
Input: – 1.0000 to + 1.0000 [mm]

Digital axes: No meaning (see MP2610.x to MP2620.x)
Input: 0

MP712.0-8 **Analog axes:** Compensation value per control-loop cycle time
Input: 0.000000 to 99.999999 [mm]

Digital axes: No meaning (see MP2610.x to MP2620.x)
Input: 0

MP715.0-8 **Analog axes:** Height of the reversal peaks during circular traverse with M105
Input: – 1.0000 to + 1.0000 [mm]

Digital axes: No meaning (see MP2610.x to MP2620.x)
Input: 0

MP716.0-8 **Analog axes:** Compensation value per control-loop cycle time with M105
Input: 0.000000 to 99.999999 [mm]

Digital axes: No meaning (see MP2610.x to MP2620.x)
Input: 0

MP7440 Output of M functions
Input: %xxxxx
Bit 3 Switching k_V factors with M105/M106
0 = function not effective
1 = function effective

4.3.6 Compensation of Static Friction

The compensation of static friction (stiction) is effective only during operation with velocity feedforward control. If you wish the stiction compensation to be effective also in the manual operating modes you must activate the velocity feedforward control with MP1391.x for each axis for manual operation.

Guideways with a relatively high level of stiction can cause servo lag at low feed-rates, even when using velocity feedforward. You can detect a servo lag by using, for example, the TNC's integral oscilloscope; and the TNC can also compensate the error. Enter an axis-specific factor to compensate the stiction in machine parameter MP1511 (recommended value: 5000 to 10 000). A higher nominal speed value based on this factor is then output while traversing.

$$F_{add} = \frac{\Delta s_a}{t_C} \times k_v \times MP1511$$

F_{add} = additional feed rate $[\frac{m}{min}]$

Δs_a = lag difference after one control loop cycle [mm]

t_C = control loop cycle time [µs]

k_v = position loop gain $[\frac{m/min}{mm}]$

MP1511.x = factor for stiction compensation [µs]

This increase in nominal value is limited by MP1512. If you set this limit too high, the machine will oscillate at standstill (recommended value: < 50).

$$MP1512.x = \frac{\Delta s_{alimit} * 256}{GP}$$

MP1512.x = limit to the amount of stiction compensation [counting steps]

$\Delta\ s_{alimit}$ = limit value Δs_a [µm]

GP = grating period of the encoder

Compensation may only be active at low feed rates because the increased nominal value would cause oscillation at high speed. The feed-rate limit for stiction compensation is defined in MP1513.x.

MP1511.0-8 Factor for stiction compensation
Input: 0 to 16 777 215

MP1512.0-8 Limit to amount of stiction compensation
Input: 0 to 16 777 215 [counting steps]

MP1513.0-8 Feed rate limit for stiction compensation
Input: 0 to 300 000 $[\frac{mm}{min}]$

MP1391 Velocity feedforward control for Manual and Handwheel operating modes
Input: %xxxxx

Bit	0 to 8	0 = operation with lag
Axis	1 to 9	1 = operation with velocity feedforward

With digital axes only: Limiting the integral factor
In machines with a lot of stiction, a high integral-action component can result if there is a position deviation at standstill. This can lead to a jump in position when the axis "takes off." In such cases the integral-action component of the speed controller can be limited with MP2512.x.

MP2512.0-5 Limiting the integral-action component of the speed controller
Input: 0.000 to 30.000 [s]
Realistic values: 0.1 to 2.0

4.3.7 Compensation of Sliding Friction

Only for digital axes
Sliding friction is compensated within the range of the motor speed controller. With the oscilloscope integrated in the TNC you can find the nominal current value (I NOML) at very low speed (approx. 10 rpm). This current is entered in MP2610.x.

At every change in direction this value is fed forward to the speed controller in order to compensate friction at low speeds (speed-independent friction compensation). You also measure the nominal current value (I NOML) at the rated speed (MP2210). This current is entered in MP2620.x. Depending on the nominal speed value, a corresponding current is fed forward to the speed controller (speed-dependent friction compensation).

When the traverse direction is reversed at high feed rates overcompensation may occur. In the circular interpolation test such overcompensation appears in the form of reversal peaks that jut inward. With MP2612.x you can enter a delay in compensation to prevent overcompensation.

MP2610.0-5 **Digital axes:** Friction compensation at low motor speed
Input: 0 to 30.0000 [A]
0 = no sliding friction
Analog axes: Non-functional
Input: 0

MP2612.0-5 **Digital axes:** Delay of friction compensation
Input: 0.0000 to 1.0000 [s]
Typical value: 0.015 s
Analog axes: Non-functional
Input: 0

MP2620.0-5 **Digital axes:** Friction compensation at rated motor speed
Input: 0 to 30.0000 [A]
0 = no sliding friction
Analog axes: Non-functional
Input: 0

4.4 PLC Positioning Commands

The TNC lets you position its axes through the PLC. For a description of how the PLC positions the spindle, see section "Spindle" in this chapter.

When to position through the PLC

- In the Manual and Handwheel operating modes, positioning is possible only when the axes are not already being positioned.
- In all other operating modes, positioning is possible only in conjunction with an M/S/T/Q strobe or if no program has been started.

If one axis is being positioned by the NC, you can move another axis only if you have defined it as a PLC axis (a separate chapter on this).

How to program a PLC positioning command

- With Modules 9221 and 9222
- To ensure compatibility (for axis 1 to axis 5), with M4120 to M4124, D528 to D544 and W560 to W568

PLC Positioning with Modules 9221 and 9222

With Module 9221 you can position a NC axis by transferring the following parameters:

- Axis to be positioned
- Target position
- Feed rate
- Absolute or incremental position
- Software limit switch status (active or not active)

A simultaneous PLC positioning movement of several axes is interpolated. If you start an additional axis while already positioning another, the first movement is aborted, and then all the programmed axes (e.g. X, Y and Z) are positioned together.

There is no tool compensation. The tool path compensation must be ended before a PLC positioning command. PLC positioning is not shown in the test graphics.

After the module call the corresponding markers M4120 to M4128 are set. You can abort the PLC positioning command by resetting this marker. If you wish to change a parameter (e.g. feed rate) of a positioning command in progress, you must first abort it with M4120 to M4128, change the parameter, and call Module 9221 again.

The NC aborts the PLC positioning command when:

- An NC STOP occurs in the Manual or Handwheel mode of operation.
- An NC STOP and internal stop occur in the automatic modes of operation.
- An EMERGENCY STOP occurs.
- An error message occurs that results in a STOP.

Start PLC positioning command :

```
PS   B/W/D/K     <Axis>               [0 to 8]
PS   B/W/D/K     <Target position>    [0.0001mm]
PS   B/W/D/K     <Feed rate>          [mm/min]
PS   B/W/D/K     <Mode>
                 Bit 0:  Target position type
                         =0:  Absolute, referenced to the machine datum
                         =1:  Incremental
                 Bit 1:  Software limit switch
                         =0:  Not active
                         =1:  Active
CM   9221
PL   B/W/D       <Error code>    0:   No error. Positioning command  is started.
                                 1:   A non-controlled axis or auxiliary axis was transferred.
                                 2:   Illegal feed rate parameters.
                                 3:   Axis has not traversed the reference mark.
                                 4:   No M/S/T/Q strobe and a program is in progress.
```

Interrogate status:

```
PS   B/W/D/K     <Axis>          [0 to 8]

CM   9222
PL   B/W/D       <Status>        0:   No PLC positioning command started yet.
                                 1:   Target position has been reached
                                 2:   PLC positioning command started.
                                 3:   Aborted. Target position not reached.
                                 4:   Target position outside of the traverse range.
                                 5:   Positioning not possible (e.g. because of “free rotation”).
```

PLC positioning with M4120 to M4124, D528 to D544 and W560 to W568 (only for axis 1 to axis 5)

Unlike PLC positioning with Module 9221, here the parameters are transferred in words.

- The target position is in doublewords D528 to D544.
- The feed rate is in words W560 to W568

To start the PLC positioning command, set markers M4120 to M4124. Software limit switches are not recognized. Otherwise the restraints are the same as for Module 9221.

			Set	Reset
D528 to D544		Target for PLC positioning [0.0001 mm]	PLC	PLC
D528	to	D544		
Axis 1	to	Axis 5		
W560 to W568		Feed rate for PLC positioning [mm/min]	PLC	PLC
W560	to	W568		
Axis 1	to	Axis 5		

		Set	Reset
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

4.5 Tilting Axes

Swiveling milling heads and tilting tables are often used on milling machines to machine workpieces from several sides. You can create such part programs either with a CAD system or right at the TNC by using its Tilt Working Plane function.

4.5.1 “Tilt Working Plane” Feature

You can position tilting axes either manually or under NC control.

In Cycle 19, Tilt Working Plane, you define the position of the working plane (e.g. A-45, B-45). With MP7500 bit 1 you define whether the entered value should correspond to the position of the individual tilting axes or the position of the working plane. The position of the working plane can be entered only for certain tilting-axis combinations. If you enter the position of the working plane, the TNC offsets the positions of the tilting axes. The resulting coordinates are stored in the following Q parameters:
Q120 = coordinate of the A axis
Q121 = coordinate of the B axis
Q122 = coordinate of the C axis

After activation of Cycle 19, the TNC undertakes a coordinate transformation. The other axes remain parallel to the tool axis, and the X-Y plane is perpendicular to the directional vector of the tool axis. With MP7500, bit 2, you define whether the tilting axes should be positioned when Cycle 19 is called. If you select automatic positioning in bit 2, you may enter a feed rate and a setup clearance in Cycle 19. With Cycle 19 the TNC then moves automatically to setup clearance and interpolates the main axes and the tilting axes so that the tool tip is once again at the same position in the tilted coordinate system.

The values displayed in the status window refer to the tilted coordinate system. In this way you can transform the working plane to any desired position, then enter the part program in the X-Y plane as usual. The NC automatically interpolates the affected axis during program run. You can use all the path functions, cycles, and datum setting and probing functions in the transformed working plane. For controlled axes the current coordinates of the tilting axes are accounted for during datum setting. The reference in this case is the machine datum. Therefore in the 0° position the set datum for the tilting axes must coincide with the coordinates referenced to the machine datum. A misalignment can be compensated with MP960.x. In MP7682 bit 1 you can set whether the nominal or the actual values should be used for calculating the preset during "datum setting". For non-controlled axes you must enter the current positions of the tilting axes using the 3D ROT soft key.

Workpieces on rotary tables are often aligned with the turning of the table. After this alignment the set datum no longer coincides with the machine zero position. If MP7500 bit 3 is set, the TNC assumes that the workpiece is aligned at "datum setting" and the Cycle Tilt Working Plane uses the new position after the table has been rotated as datum.

When combining coordinate transformation cycles you must consider the sequence of activation and deactivation. The tool radius offset in the working plane and the tool length offset parallel to the tool axis are active. With the 3D ROT soft key you can activate the tilted working plane separately for the Manual and Program Run modes.

You must enter the mechanical offset between the tilting axes in the machine parameters. The tilting head must be in its home position (all tilting axes in the 0° position) when the shifts are entered. For tilting heads the initial position is the tool datum of the machine (usually the spindle head). Enter the shift or rotation to the tilting axis closest to the machine, then do the same for the next axis, and continue this in sequence until you reach the end of the tilting-axis chain. If a rotation has been entered, this must be removed in an additional transformation.

For a tilting table, you do not begin the description of the machine geometry at the tool datum, but rather at the center of rotation of the first tilting axis (seen from the workpiece). First, enter the center of rotation in absolute coordinates, referenced to the machine datum. Then enter, in sequence, the shifts and rotations until you reach the end of the tilting-axis chain. For machining with the tilting tables the coordinate system remains parallel to the machine coordinate system.

With MP7500 you enable the Tilt Working Plane function. The description of machine geometry in MP7510 to 7530 is also used for other functions (e.g. Cylinder Surface).

- If Cycle 19 “Tilt Working Plane” is active it is not possible to position with M91 or M92.
- The “Basic Rotation” function can be performed only when the tilting axes are at their 0° positions.
- PLC positioning commands are always executed paraxially to the machine coordinate system. Cycle 19 therefore has no influence on PLC positioning commands.
- The datum shift via PLC also works together with the Tilt Working Plane function
- Entering the angle for the angle in space (MP7500 Bit 1 = 1) is possible only for the following combination of tilting axes:
  - 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
  - Tilt and turn table: axis sequence C variable / A or B variable
  - Swivel head and rotary table: axis sequence A or B variable / C variable
  - Rotary table 45°: axis sequence C variable / A fixed / B variable / A fixed

  As of NC software 280 472 01:
  - Double swivel head 90°: axis sequence A variable / B variable
- The axis designation for tilting axes is limited to A, B, C. Each designation can only occur once.

MP7500 “Tilting the Working Plane” function
Input: %xxx
Bit 0 Turn on “Tilt Working Plane“ function 0 = off
1 = on
Bit 1 0: Angle corresponds to the position of the tilting axes of the head / table
1: Angle corresponds to the position of the working plane (the TNC calculates the position of the tilt 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: <Set datum> = <machine datum> also valid for rotary tables
1: With rotary tables, the Cycle "Tilt Working Plane" is referenced to the new position from "Datum setting" after the table has been turned for the workpiece to be aligned

MP7510 Transformed axis
Input: %xxxxxx
0 = end of the transformation sequence

Bit	5	4	3	2	1	0
Axis	C	B	A	Z	Y	X

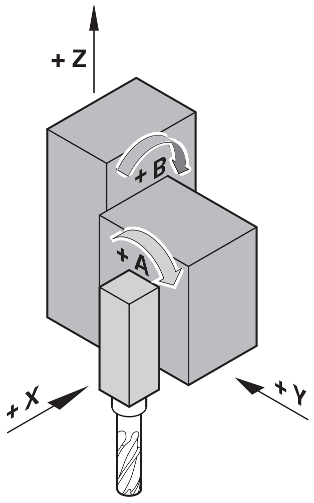
MP7510.0 1st transformation
to to
MP7510.14 15th transformation

MP7520 Supplementary identifier for transformation
Input: %xx
Bit 0 Swivel axis 0 = swivel head
1 = tilting table
Bit 1 Type of dimension in MP7530 0 = incremental dimension
1 = absolute dimension referenced to the machine datum
MP7520.0 1st transformation
to to
MP7520.14 15th transformation

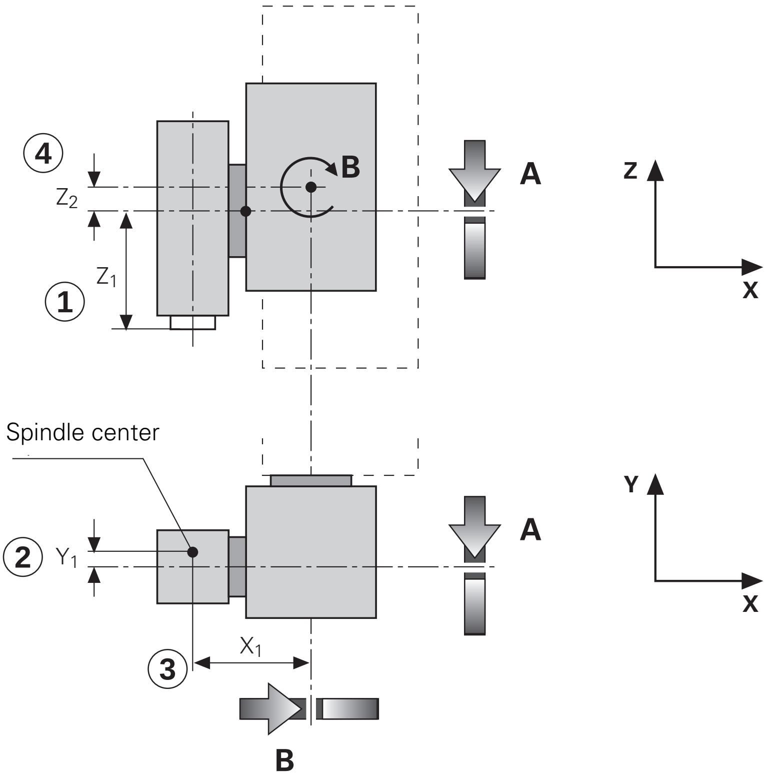
MP7530 Type of dimension for transformation
Input: – 99 999.9999 to + 99 999.9999
0 = free swivel axis
MP7530.0 1st transformation
to to
MP7530.14 15th Transformation

MP7682 Machine parameter with multiple functions
Input: %xxx
Bit 1 Reference value for calculating the preset in "datum setting"
0 = actual value is used
1 = nominal value is used

Example 1: Double swivel head, rectangular

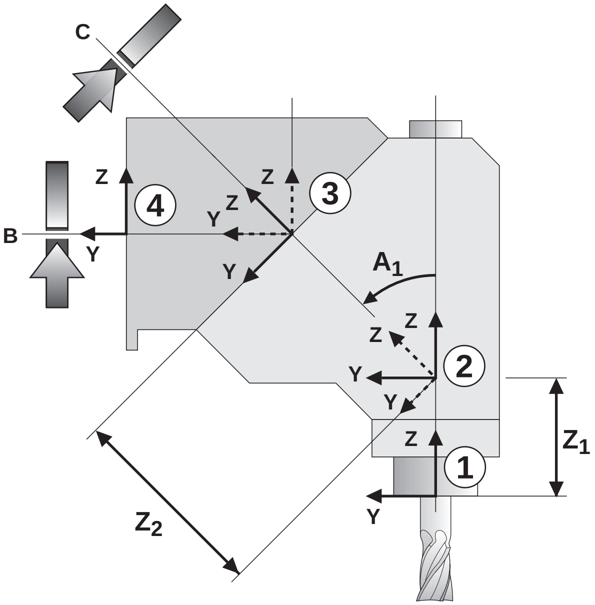


Z_1 =	200.4 mm
Z_2 =	3.1 mm
X_1 =	201.5 mm
Y_1 =	1.9 mm



```
MP 7500   : %111      ;activate functions

MP 7510.0 : %000100   ;shift in axis Z (Z1)
MP 7510.1 : %000010   ;shift in axis Y (Y1)
MP 7510.2 : %001000   ;shift in axis A
MP 7510.3 : %000001   ;shift in axis X (X1)
MP 7510.4 : %000100   ;shift in axis Z (Z2)
MP 7510.5 : %010000   ;free tilting axis B
MP 7510.6 : %000000   ;end of the transformation sequence

MP 7520.0 : %00       ;tilting head, incremental dimension
MP 7520.1 : %00       ;tilting head, incremental dimension
MP 7520.2 : %00       ;tilting head, incremental dimension
MP 7520.3 : %00       ;tilting head, incremental dimension
MP 7520.4 : %00       ;tilting head, incremental dimension
MP 7520.5 : %00       ;tilting head, incremental dimension

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)
```

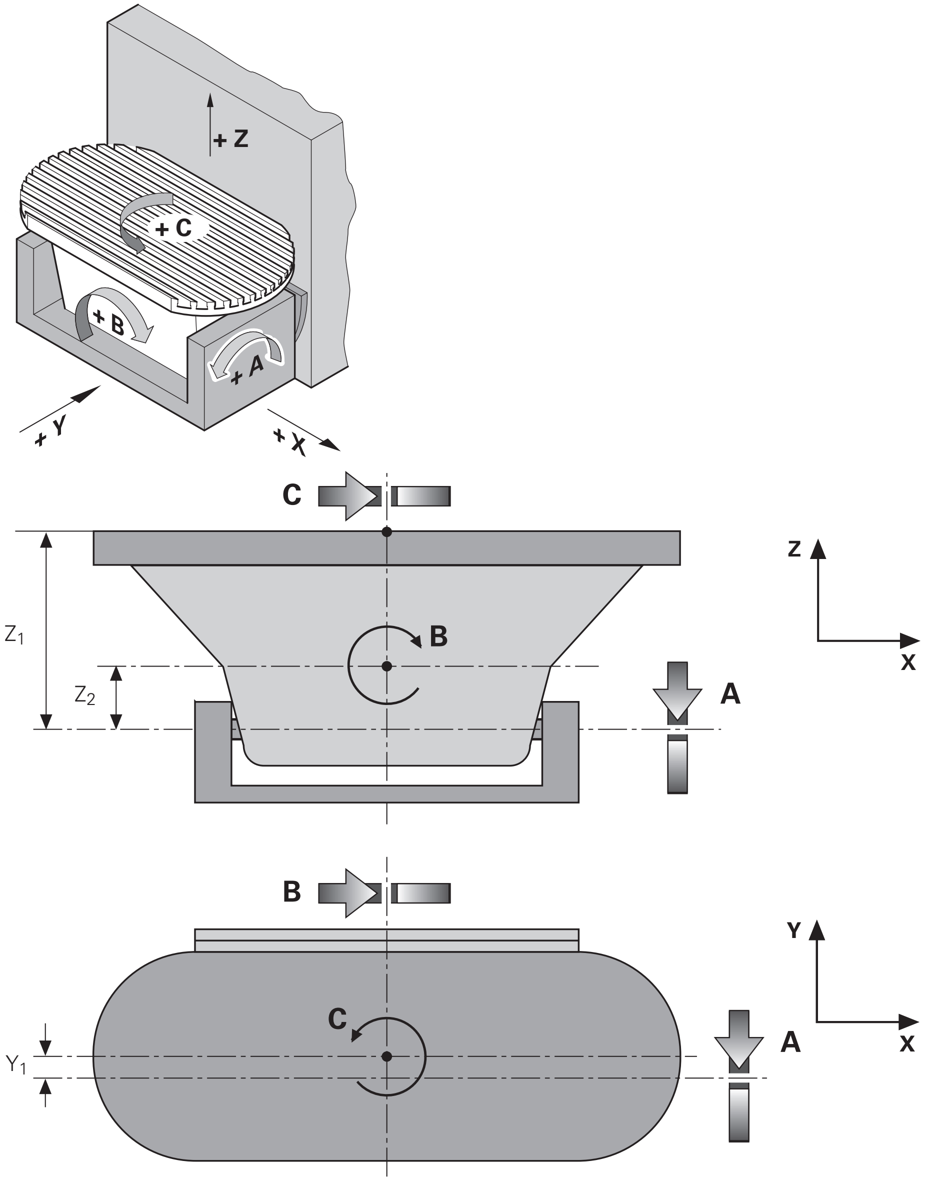
Example 2: Double swivel head 45°



Z1 = 150.5 mm
Z2 = 251.5 mm
A1 = 45°

```
MP 7500   : %111      ;activate functions

MP 7510.0 : %000100   ;shift in axis Z (Z1)
MP 7510.1 : %001000   ;rotate coordinate system around axis A (A1)
MP 7510.2 : %000100   ;shift in axis Z (Z2)
MP 7510.3 : %100000   ;free tilting axis C
MP 7510.4 : %001000   ;rotate coordinate system around axis A (A1)
MP 7510.5 : %010000   ;free tilting axis B
MP 7510.6 : %000000   ;end of the transformation sequence

MP 7520.0 : %00       ;tilting head, incremental dimension
MP 7520.1 : %00       ;tilting head, incremental dimension
MP 7520.2 : %00       ;tilting head, incremental dimension
MP 7520.3 : %00       ;tilting head, incremental dimension
MP 7520.4 : %00       ;tilting head, incremental dimension
MP 7520.5 : %00       ;tilting head, incremental dimension

MP 7530.0 : +150.5    ;dimension Z1
MP 7530.1 : -45       ;dimension A1
MP 7530.2 : +251.5    ;dimension Z2
MP 7530.3 : +0        ;variable dimension (free tilting axis C)
MP 7530.4 : +45       ;dimension A1
MP 7530.5 : +0        ;variable dimension (free tilting axis B)
```

Example 3: Universal table (tilting, pitching, rotating)



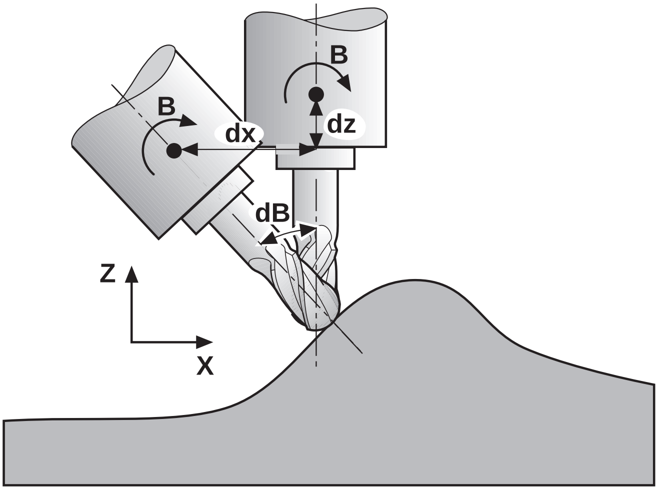
Y_1 =	2.7 mm
Z_1 =	331.3 mm
Z_2 =	125.9 mm

Coordinates (referenced to the machine datum) of the center of rotation of the rotary table C with all swivel axes in home position:

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

```
MP 7500   : %111      ;activate functions

MP 7510.0 : %000001   ;X coordinate of center of rotation in axis C
MP 7510.1 : %000010   ;Y coordinate of center of rotation in axis C
MP 7510.2 : %000100   ;Z coordinate of center of rotation in axis C
MP 7510.3 : %100000   ;free tilting axis C
MP 7510.4 : %000010   ;shift in axis Y (Y1)
MP 7510.5 : %000100   ;shift in axis Z (Z1)
MP 7510.6 : %001000   ;free tilting axis A
MP 7510.7 : %000100   ;shift in axis Z (Z2)
MP 7510.8 : %010000   ;free tilting axis B
MP 7510.9 : %000000   ;end of transformation sequence

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)
```

4.5.2 Automatic Compensation of Offset for Tilting Axes



Tilting axes and linear axes can be interpolated at the same time. With M114 or M128 the TNC compensates the tool offset resulting from tilting axis positioning due to the axes' geometrical design. This is a 3-D length compensation. The radius compensation must be calculated by the CAD system or by the postprocessor. A programmed radius compensation (RL or RR) results in an error message.

Since the TNC accounts for the machine geometry values from MP7510 to MP7530, the postprocessor does not need to account for them. The tool point is always on the programmed nominal coordinates.

Unlike the Tilting Working Plane function, the coordinate system is not tilted.

If the TNC performs the tool length compensation, the programmed feed rate refers to the tool point, otherwise it refers to the tool datum.

M114

- Kinematic deviations resulting from the superposition of translational and rotatory movements are dependent on the length of the linear interpolation.
- For machines with rotary tables note that the rotating of the table causes a rotation in the workpiece coordinate system relative to the machine coordinate system. The TNC does not take this into consideration.
- M114 can also be used with non-controlled axes or PLC tilting axes. The current tilting angle and the tilting axis are entered in the NC block after M114.

M128

- The superposition of translational and rotatory movements does not cause kinematic deviations.
- For machines with rotary tables, the TNC takes into consideration the rotation in the workpiece coordinate system caused by the rotating of the table.
- M128 is not deactivated by a change in operating mode, which means that the tilting axes can be traversed with machine-geometric compensation using the axis-direction keys in the Manual operating mode and using the handwheel in the Handwheel operating mode.
- With M118, handwheel positioning of the tilting axes can be superimposed during program run, and the TNC automatically carries out an offset compensation in the main axes.

4.5.3 Cylinder Surface

Cycle 27 "cylinder surface" enables you to machine a cylindrical surface by entering the surface curvature separately and programming the contour in two axes as if it were on a plane surface (see the TNC User's Manual). For Cycle 27 to function properly, the center of rotation must have been defined in MP7510 to MP7530 (see example 3).

If you wish to use a PLC datum shift together with the Cylinder Surface cycle, ensure that the same datum position is used for the description of the machine geometry in MP7510 to MP7530 as for the datum shift (home position of the swivel head).

4.6 Synchronized Axes

With the HEIDENHAIN TNC, you can couple two controlled axes in such a way that they can only be moved simultaneously. This facility is required, for example, for gantry axes and tandem tables, and can be activated both for operation with servo lag and in the feedforward control mode.

In the following description the main axis and tracking axis are referred to as master and slave, respectively.

You activate the Synchronize function by defining in MP850 the master axis that the specified slave axis must track.

Example: Axis 4 is the slave of 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

Of the nine controlled axes, two pairs can be synchronized.

MP850.0-8 Synchronized axes
Input: 0 = main axis
1 = axis tracked to axis 1
2 = axis tracked to axis 2
3 = axis tracked to axis 3
4 = axis tracked to axis 4
5 = axis tracked to axis 5
6 = axis tracked to axis 6
7 = axis tracked to axis 7
8 = axis tracked to axis 8
9 = axis tracked to axis 9

4.6.1 Synchronization Control

The TNC monitors the synchronization of the master and slave axes. If the positions of the master and slave axes differ by a distance greater than that entered in MP855.x, the error message `Excessive servo lag in <axis> # A` appears indicating the slave axis. The positional difference is shown in the LAG display for the slave axis. Synchronization monitoring is entered in MP855.x of the slave axis. Machine parameter MP860.x defines the datum for synchronization control.

MP860.x = 0: Datum at position upon switch-on
When the machine is switched on it is assumed that the master and slave axes are synchronized with one another. Their position upon switch-on is taken as the synchronization reference. In this mode, passing over the reference mark is necessary only for the master axis, and only if the defined references are to be reproduced. Synchrony monitoring begins immediately upon switch-on.

MP860.x = 1: Datum at reference marks (machine datum)
Both axes are positioned to the same reference value when they have passed over their respective reference marks. The default setting can be modified with MP960.x (machine datum).

In this mode, any offset between the two axes is compensated in the slave axis at the speed from MP1330.x after both reference marks are traversed, and synchronization does not activate until compensation is completed. The manner in which the reference marks are traversed (MP1350.x) must be set the same for both axes. The master axis must be defined before the slave axis in the sequence for approaching the reference marks (MP1340.x).

Where rotary encoders are used for linear measurement (MP1350.x = 2), only one limit switch should be used for both axes because the marker "Reference end position for the slave axis" is only for evaluation of the reference mark, and not for evaluation of the traverse direction, when controlling the sequence. The direction of traverse is defined by the value of the reference end position marker for the master axis.

Traversing the reference mark is completed when one reference mark has been evaluated for both axes.

MP855.0-8 Synchronization monitoring
Input: 0 to 100.0000 [mm]
0 = Monitoring not active

MP860.0-8 Datum for synchronization control
Input: 0 = datum at position upon switch-on
1 = datum at reference marks (machine datum)

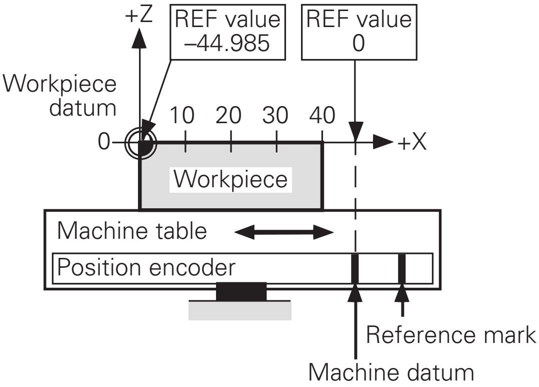
4.6.2 Conventions

- The slave axis cannot be moved independently.
- The nominal value displayed for the slave axis indicates 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 (locking, feed-rate enable, etc.).
- The bits for direction of traverse (W1030) and axis in motion (W1028) are not set for the slave axis.
- An axis cannot be master and slave at the same time.
- You must enter axis error compensation (both linear and nonlinear) separately for both axes.
- The values for rapid traverse, acceleration, software limit switches, feed-rate for passing over reference marks, and manual feed-rate are confirmed from the input values of the master axis.
- In lag mode, the k_v factor for master and slave axis should be the same.
- Both axes must be under either analog or digital control.
- *Up to NC software 280 470 04: master and slave must be linear axes.*

4.7 Reference Marks

By setting a datum point, you assign a distinct positional value (coordinate) to each axis position for the machining of the workpiece. Since the actual-position value is established incrementally by the encoder, this relationship between axis positions and positional values must be re-established after every power interruption.

HEIDENHAIN linear encoders are therefore equipped with one or more reference marks. On passing a reference mark, a signal is generated that identifies that position as a reference point. By passing over the reference marks after a power interruption, you re-establish the assignment of positional values to axis positions as it was most recently defined by datum setting. Passing over the reference marks also restores all machine-based references.



Since it is often inconvenient to re-establish the reference points by traversing large distances after switching on, HEIDENHAIN recommends the use of encoders with distance-coded reference marks. With distance-coded reference marks, the absolute position is available after crossing two reference marks.

The scale graduation consists of the line grating track and a reference mark track parallel to it. The distances between any two consecutive reference marks are defined differently, so that the absolute position of the machine slide can be determined from this distance.

Machine datum

The reference mark identifies a point on the position encoder. The reference points of all axes define the scale reference point. The distance from the scale reference point to the machine datum is defined in MP960.x. All REF-based display values and positioning movements are referenced to the machine datum.

4.7.1 Traversing the Reference Marks

The reference marks for the axes must be passed after the control is switched on. This can be achieved by

- pressing the external START key. The axis sequence is determined by machine parameter MP1340.X (automatic traversing of the reference marks),
- pressing the external axis direction keys. The sequence is determined by the operator.

Only after passing over the reference mark

- can the software limits be activated,
- can the most recently set datum point be reproduced,
- is PLC positioning and positioning with the miscellaneous functions M91 and M92 possible,
- is the counter value set to zero for non-controlled axes.

For distance-coded measuring systems the machine datum (MP960.x) is referenced to the scale reference point. In linear encoders the scale reference point is the first reference mark after the start of the measuring length; in angle encoders the scale reference point is marked.

The direction of traverse and the velocity on passing the reference marks is defined by machine parameters (MP1320.x, MP1330.x). The functional sequence for passing the reference marks can be fixed specifically for the axes by machine parameters (MP1350.x).

The operating condition "PASS OVER REFERENCE MARKS" is sent to the PLC by the NC (W272). If the operating mode is changed before all reference marks have been passed, the PASS OVER REFERENCE soft key will appear. The markers M2136 to M2140 inform the PLC of the axes for which the reference marks have not yet been crossed.

In order to avoid exceeding the traverse range when passing over the reference marks, a trip dog (reference end-position) is necessary in each axis. This trip dog must be fixed at the end of the traverse range by the manufacturer of the machine. The signals from the trip dogs are connected to available PLC inputs. In the PLC program these PLC inputs are combined with W1054 (Reference end-position).

Repeated Traversing of the Reference Marks (Module 9220)
With this module you move an axis or controlled spindle over the reference mark. You can also reevaluate the reference mark for a particular axis.

NC axis:

- The module can be called only in the operating modes “Traverse Reference Points,” “Manual Operation,” and “Electronic Handwheel.”
- The functional sequence (MP1350.x) and the velocity for leaving the reference end position (MP1331.x) is always defined by the machine parameter values.
- The velocity and traversing direction for traversing the reference mark is either taken from MP1330.x and MP1320.x or entered in the module. Only in specific exceptional cases should the traversing direction be defined in the module. This is because the module does not regard the reference end positions and therefore the traverse limits may be exceeded.

- If an axis already has its reference but is started for referencing again, the corresponding bit is set in W1032 and the reference mark is reevaluated. The same constraints apply here as when traversing the reference mark for the first time.
- An axis cannot be started for referencing until all other axes are in position.

Controlled spindle:

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

Call:

```
PS   B/W/D/K    <Axis / Spindle>
                0 to 8 for axis 1 to 9; 15 for spindle
PS   B/W/D/K    <Feed rate / Speed>
                0:      Feed rate from MP1330.x
                >0:     Feed rate in mm/min or 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:      No error: Ref positioning is commanded
                1:      Axis or spindle does not exist
                2:      Illegal values for feed rate / direction were transferred
                3:      Wrong operating mode (only NC axes)
                4:      Ref positioning not possible because previous reference run was not yet
                        started
                5:      Axis is already being positioned or spindle in motion
                6:      Another axis is being positioned (only NC axes)
```

MP960.0-8 Machine datum
Input: –99 999.9999 to +99 999,9999 [mm] or [°]
Values referenced to the scale reference point

MP1320 Direction for traversing the reference marks
Input: %xxxxx

Bit	0 to 8	0 = positive
Axis	1 to 9	1 = negative

MP1330.0-8 Velocity when traversing the reference marks
Input: 80 to 300 000 [mm/min]

MP1331.0-8 Velocity when leaving the reference end position (only for rotary encoders MP1350=2)
Input: 80 to 500 [mm/min]

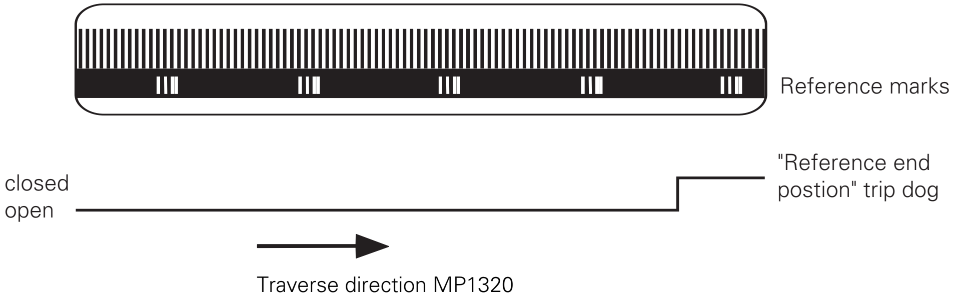
MP1340.0-8 Axis sequence when traversing the reference marks
Input: 0 = no evaluation of the reference marks
1 = axis X
2 = axis Y
3 = axis Z
4 = axis 4
5 = axis 5

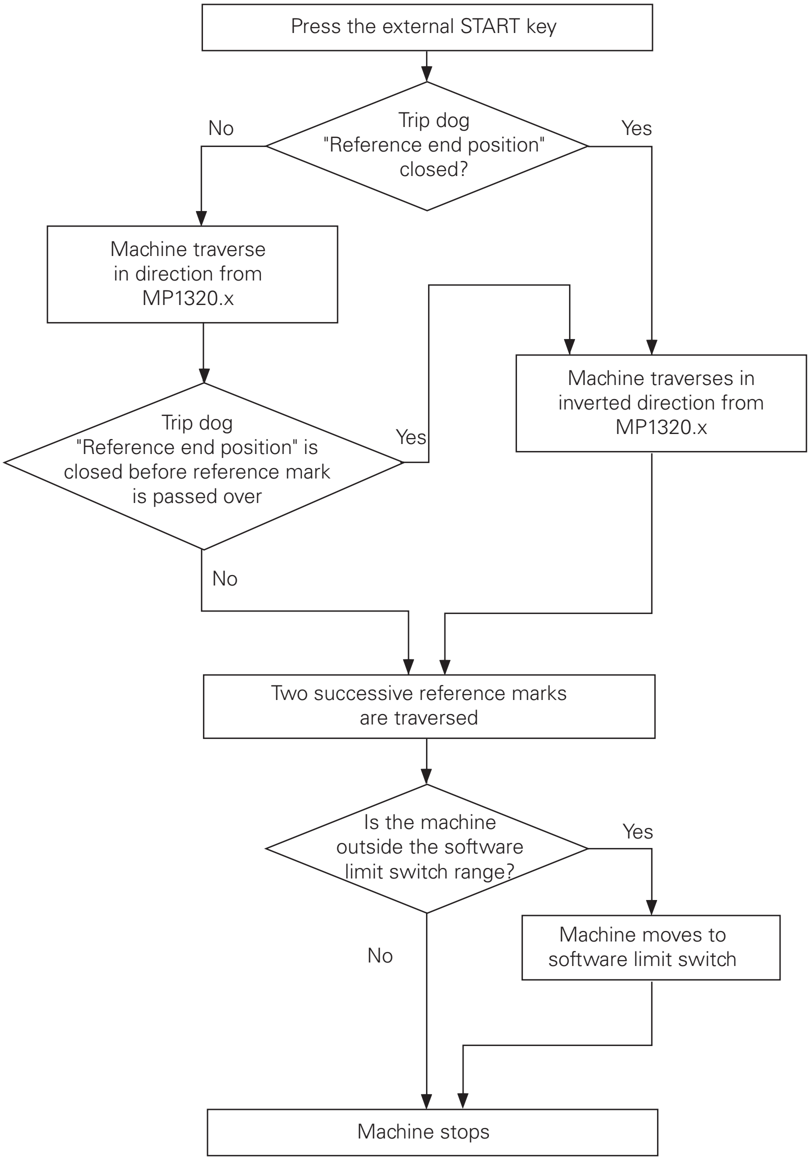
MP1350.0-8 Functional sequence when traversing the reference marks
Input: *0 = linear encoder with distance-coded reference marks (old routine)*
1 = linear encoder with one reference mark
2 = special sequence (linear measurement via ROD)
3 = linear encoder with distance-coded reference marks (new routine)

		Set	Reset
W272	Mode of operation 1 = manual 2 = electronic handwheel 3 = positioning with Manual Data Input 4 = program run, single block 5 = program run, full sequence 7 = traversing the reference points	NC	NC
W1032	Reference points not yet traversed Bit 0 to 8 Axis 1 to 9	NC	NC
W1054	Reference end position Bit 0 to 8 Axis 1 to 9	PLC	PLC

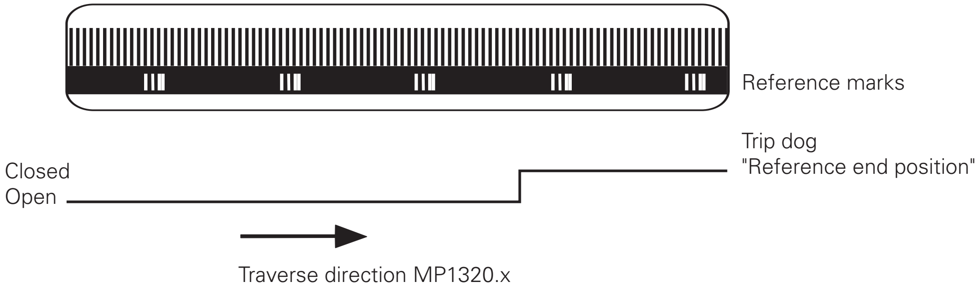
Linear encoder with distance-coded reference marks

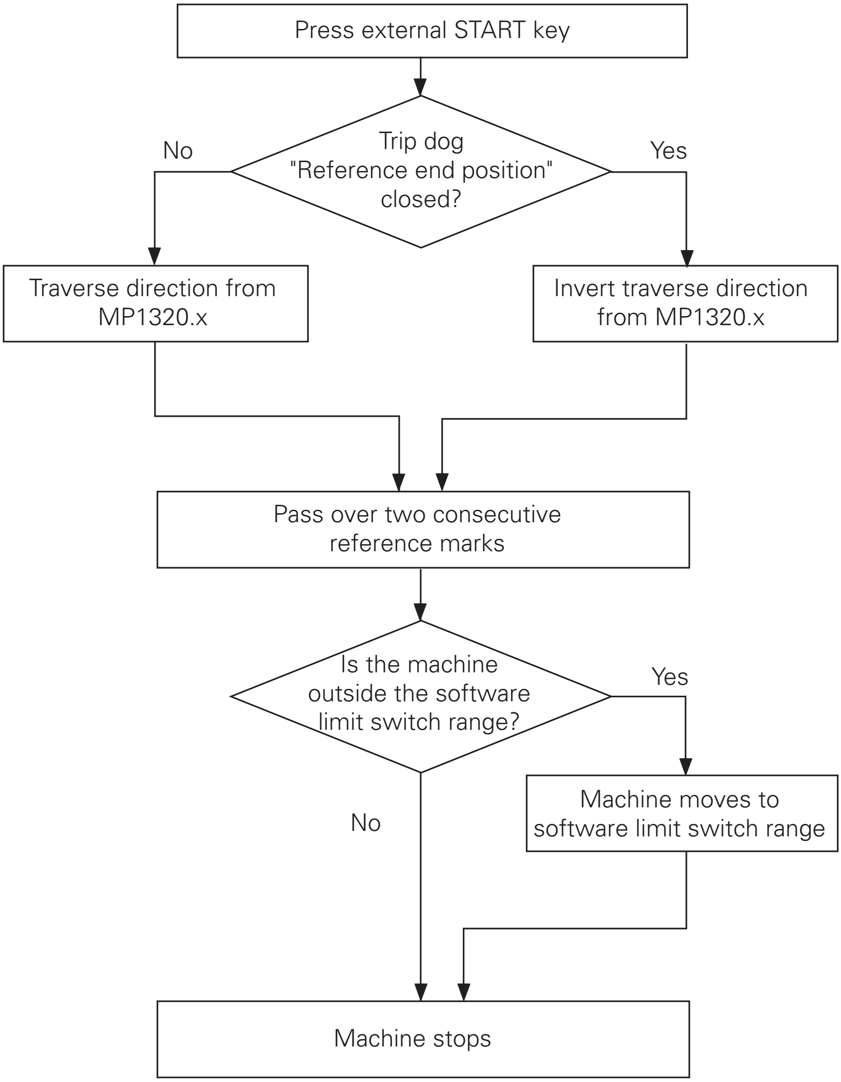
Machine parameter MP1350.x=3





Machine parameter MP1350.x=0 (This parameter is kept only to ensure compatibility. It should not be used for new installations.)

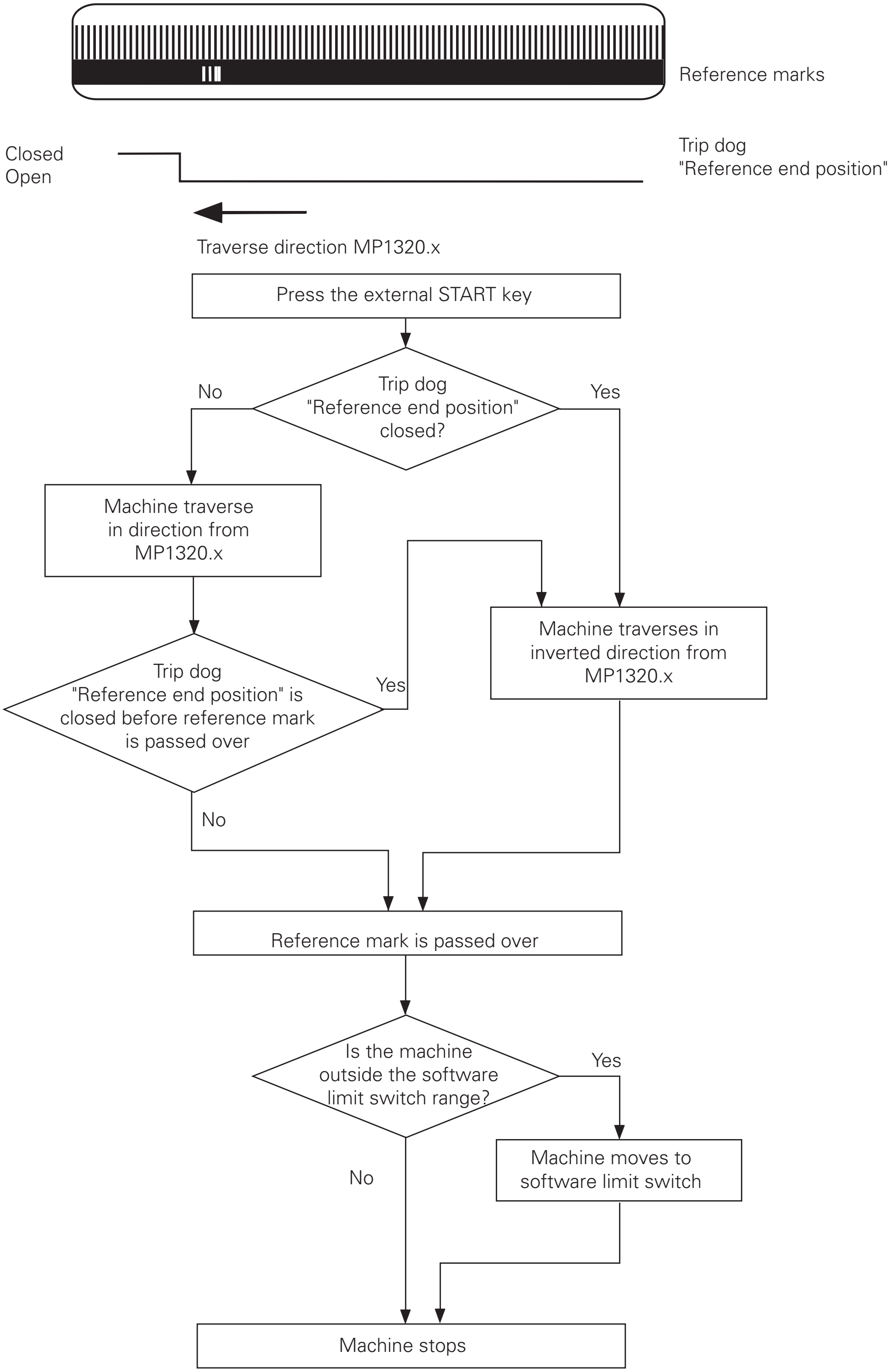




If during automatic referencing the trip dog is not closed until it is in the reference end position range, the contouring control will ignore the signal. It is therefore necessary that there be at least two reference marks in the range of the reference end position.

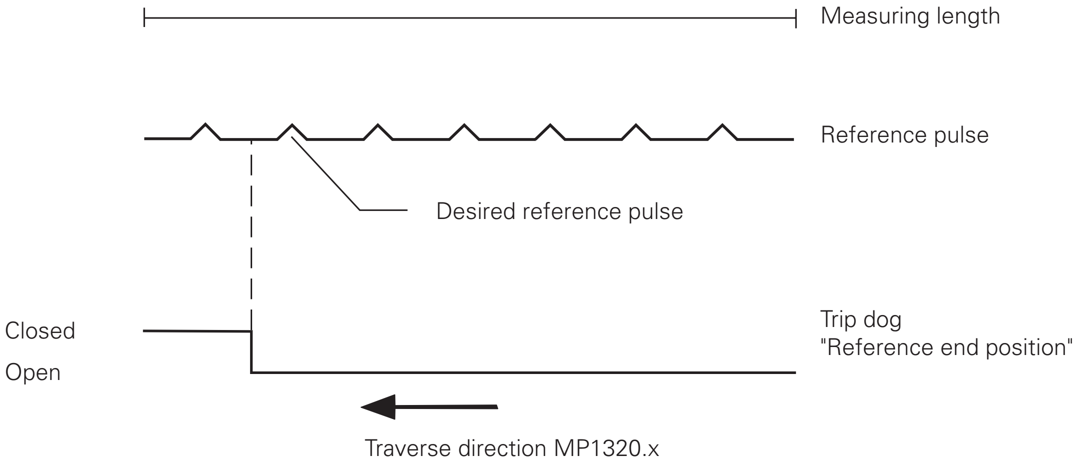
Linear encoder with one reference mark

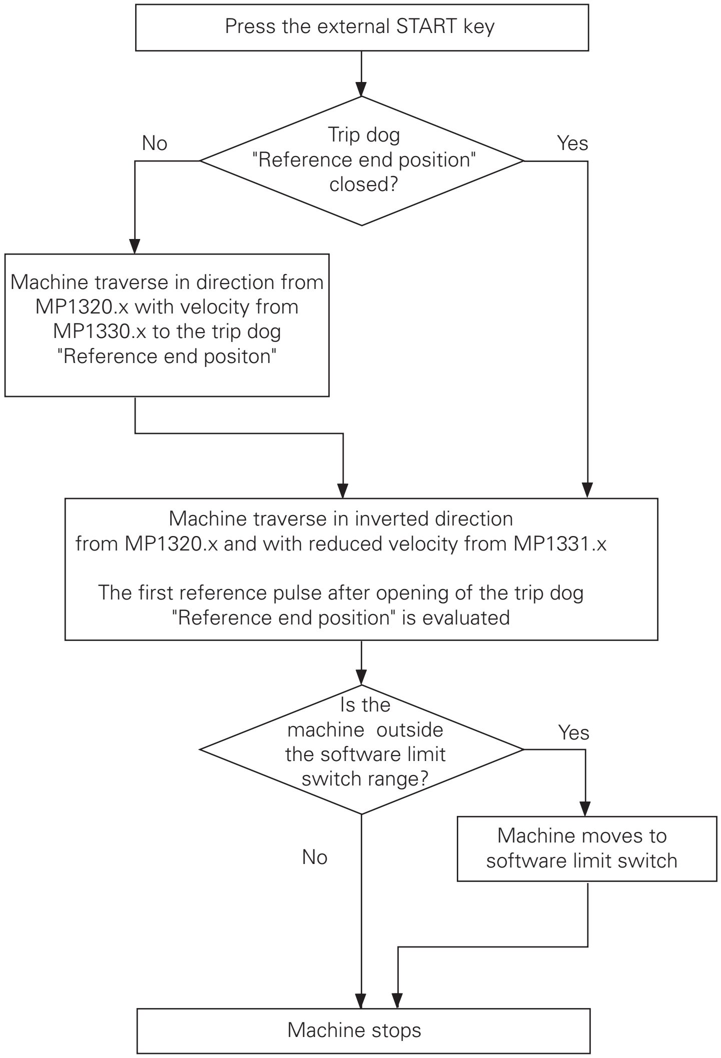
Machine parameter MP1350.x=1


Reference marks
Closed
Open
Trip dog
"Reference end position"
Traverse direction MP1320.x
Press the external START key
Trip dog "Reference end position" closed?
No
Yes
Machine traverse in direction from MP1320.x
Machine traverses in inverted direction from MP1320.x
Trip dog "Reference end position" is closed before reference mark is passed over
Yes
No
Reference mark is passed over
Is the machine outside the software limit switch range?
Yes
No
Machine moves to software limit switch
Machine stops

Linear measurement via rotary encoder
Machine parameter MP1350.x = 2

For linear measurement using a rotary encoder, a reference pulse is produced on each revolution of the encoder. You must ensure that during referencing the same reference pulse is always evaluated. This can also be achieved using the trip dog for the reference end position.




Press the external START key
Trip dog "Reference end position" closed?
No
Yes
Machine traverse in direction from MP1320.x with velocity from MP1330.x to the trip dog "Reference end positon"
Machine traverse in inverted direction from MP1320.x and with reduced velocity from MP1331.x
The first reference pulse after opening of the trip dog "Reference end position" is evaluated
Is the machine outside the software limit switch range?
Yes
No
Machine moves to software limit switch
Machine stops

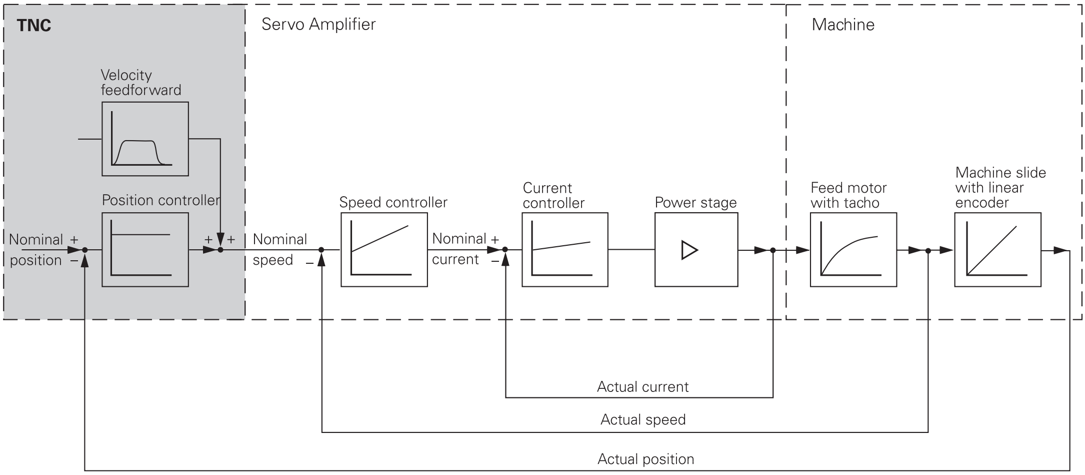
4.8 The Control Loop

Machine tools normally function according to the principle of cascade control. Here the position control loop is prior to the speed and current control loops. Cascade control offers the following benefits:

- Good overview of the structures of the individual control loops.
- Disturbances can be directly and therefore quickly compensated by the subsequent controllers, thereby relieving the prior control.
- The respective outer control loop (see illustration below) protects the inner control loop by limiting its reference input value.
- Individual commissioning of each control loop, starting with the innermost loop.

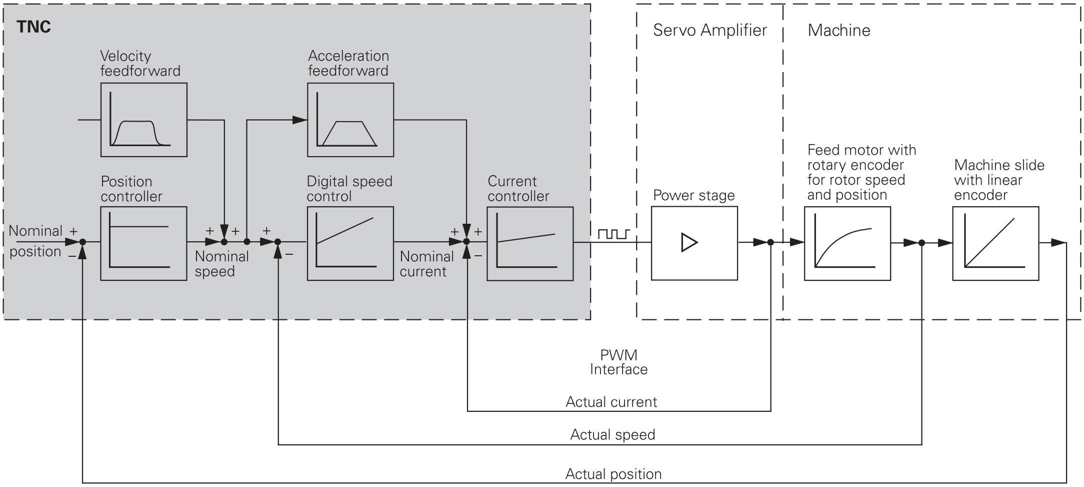
TNC 426 CB, TNC 430 CA:

- The control loop is integrated in the TNC.
- The speed and current controllers are located in the servo amplifier.
- The nominal speed command signal is transferred through an analog ±10 V interface from the TNC to the servo amplifier.



TNC 426 PB, TNC 430 PA:

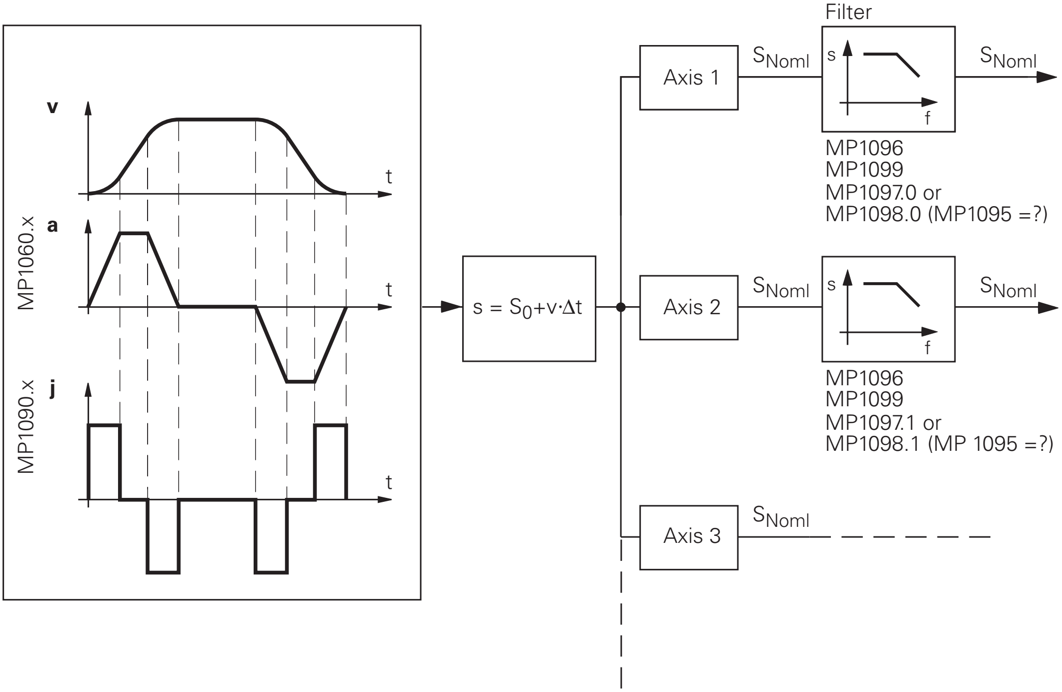
- The position, speed, and current controllers are located in the TNC.
- Only the power stage remains in the servo amplifier.
- The power stage is controlled by the TNC by means of Pulse-Width Modulated (PWM) signals.



The individual control loops and their machine parameters are described in the following pages. A block diagram for the control loop of a digital axis is provided in the Appendix.

In the TNC 426 CB, TNC 430 CA the machine parameters for the speed and current controllers are not accessible. You must adjust these controllers in accordance with the documentation for your servo amplifier.

4.8.1 Interpolator



The interpolator calculates a velocity every 3 ms from the feed rate in the NC program. The value is also dependent on the acceleration curve and the end position. If several axes are traversed at the same time, the smallest acceleration value is effective.
In two machine parameters you adjust the velocity feedforward value to the dynamics of the machine.

- With MP1060.x you determine the acceleration (= slope of the velocity curve with feedforward).
- With MP1090.x you limit the jerk during acceleration and deceleration. The jerk is the derivative of the acceleration. The greater the entered value, the more the system will tend to oscillate.

In order for the desired acceleration to be reached, the following must be taken into consideration:

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

$$\text{<MP1090>} \geq \frac{(\text{<MP1060>})^2 \bullet 60\,000}{\text{<MP1010>}}$$

MP1090 = jerk limiting [m/s^3)
MP1060 = acceleration [m/s^2)
MP1010 = rapid traverse [mm/min]

At high feed rates (e.g. rapid traverse), a higher rate-of-change (jerk) is permissible than at lower feed rates. Thus you can set the jerk for low feed rates in MP1090.0 and for high feed rates in MP1090.1. In MP1092 you can set the feed rate threshold from which MP1090.1 becomes effective.

MP1090 is not axis-specific. Thus when entering the values you must consider the weakest axis.

Every 3 ms a nominal position value is derived from the calculated velocity. For linear interpolation, the following formula applies

$s = s_o + v \bullet \Delta t$

	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.

Nominal position value filter

For optimal adjusting of speed and acceleration, the nominal position values are filtered. This results in smoother (jerk-limited) traverse. The TNC calculates the filter parameters weighting and width (order) from the permissible axis-specific jerk value (MP1097.x or MP1098.x) and the tolerance (MP1096).

Set the minimum filtering order in MP1099. You can use Cycle 32 "Tolerance" to overwrite the tolerance value at contour transitions defined in MP1096. With MP1095 you can select whether the TNC uses a single or double filter. The single filter results in a linear change in acceleration and therefore a step in the jerk. The double filter gives a curved-form change in acceleration and jerk.

The filter is effective in all operating modes. For rigid tapping (Cycle 17) the nominal position value filter is automatically switched off..

MP1060.0-8 Acceleration
Input: 0.001 to 5.0 [m/s^2]

MP1090 Jerk limiting
Input: 1 to 1000 [m/s^3]
MP1090.0 Jerk limiting with machining feed rate
MP1090.1 Jerk limiting as of feed rate in MP1092

MP1092 Threshold as of which MP1090.1 effective
Input: 10 to 300 000 [mm/min]

MP1095 Nominal position value filter
Input: 0 = single filter
1 = double filter
Suggested input value = 0

MP1096 Tolerance
Input: 0 = no nominal position value filter
0.001 to 3.000 [mm] = permissible tolerance at contour transitions

MP1097.0-8 Axis-specific jerk for single filter (MP1096 = 0)
Input: 1 to 1 000 [m/s^3]

MP1098.0-8 Axis-specific jerk for double filter (MP1096 = 1)
Input: 1 to 1 000 [m/s^3]
Suggested input value = 2 • MP1097.x

MP1099 Minimum filtering order

Input: 0 to 20

MP1099.0 Minimum filtering order for single filter (MP1095 = 0)
Suggested input value = 5

MP1099.1 Minimum filtering order for double filter (MP1095 = 1)
Suggested input value = 3

4.8.2 Position Controllers

With MP1390 and MP1391 you can choose between two types of control:

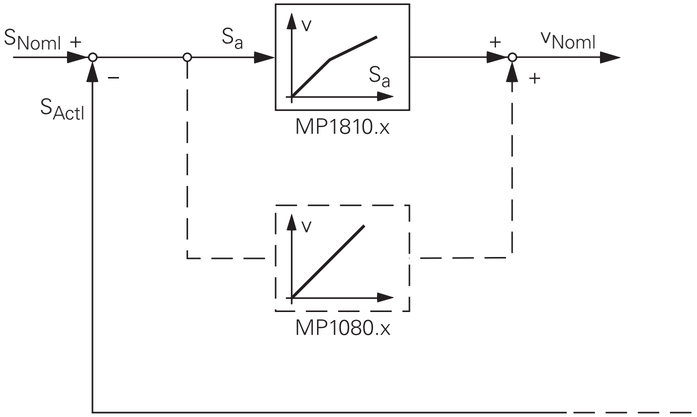
- Control with lag
- Velocity feedforward control

Always adjust the machine for both types of control.

MP1390	Velocity feedforward control in the "Positioning with MDI," "Program run, single block" and "Program run, full sequence" operating modes		
Input:	0 = velocity feedforward control		
	1 = control with lag		

MP1391	Velocity feedforward control in the "Manual" and "Handwheel" operating modes		
Input:	%xxxxx		
	Bit	0 to 8	0 = control with lag
	Axis	1 to 9	1 = velocity feedforward control

Control with lag
The servo lag (also known as following error) is a gap that remains between the nominal position commanded by the NC and the actual position. Control with servo lag is illustrated in the following simplified block diagram.



The axis-dependent nominal position value is compared with the actual position value and the servo lag s_a is calculated..

$s_a = s_{Noml} - s_{Actl}$

s_a = lag
s_{Noml} = nominal position value
s_{Actl} = actual position value

The lag is multiplied by the k_v factor MP1810.x and passed on as a nominal velocity value.

$v = k_v \cdot s_a$ v = nominal velocity value

Analog axes: If the axes are stopped, the integral factor MP1080.x becomes effective as well. It causes an offset adjustment.

Digital axes: There is no offset. MP1080.x is without function.

The k_v factor (position loop gain) determines the amplification of the control loop. The optimal k_v factor must be determined by trial and error. If you choose a very high k_v factor, the servo lag is very small. However, this can lead to oscillations when moving into a new position. If the k_v factor is too small, the new position will be reached too slowly. For axes that are interpolated with each other the k_v factor must be equal to prevent contour deviations.

With MP1815.x you define a second set of k_v factors that the operator can activate with the M function M105. This allows the operator to selectively choose higher contour accuracy at critical contours. M105 also influences the compensation of reversal peaks during circular movement.
M106 returns the control to the original set of k_v factors.
You must enable the M functions M105/M106 with MP7440, bit 3.

The following formula shows the relationship among k_v factor, feed rate and servo lag:

$k_v = \frac{v_e}{s_a}$

k_v = position loop gain $[\frac{m/min}{mm}]$

v_e = rapid traverse $[\frac{m}{min}]$

s_a = servo lag [mm]

or

$s_a = \frac{v_e}{k_v}$

MP1810.0-8 k_v factor for control with servo lag
Input: 0.100 to 20.000 $[\frac{m/min}{mm}]$

MP1815.0-8 k_v factor for control with servo lag effective after M105
Input: 0.100 to 20.000 $[\frac{m/min}{mm}]$

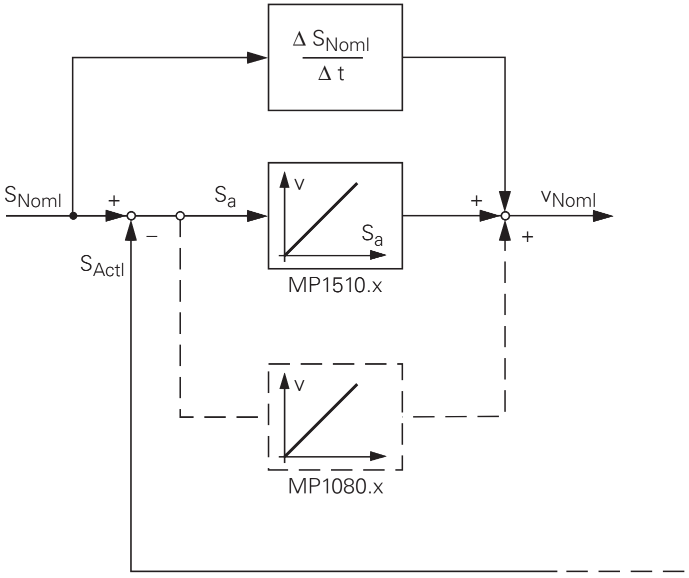
MP7440 Output of M functions
Input: %xxxxx
Bit 3 Switching of k_v factors with M105/M106
0 = function active
1 = function not active

Control with velocity feedforward

Control with velocity feedforward means that the nominal velocity value is adjusted for the machine (controlling portion of the nominal velocity). This is added to the velocity value that is calculated from the servo lag (servo-controlled portion) to form the final nominal velocity value.

The servo lag is very small with this method. Operation with velocity feedforward has the advantage that contours can be followed very accurately even at a high speed. Normally, work will be carried out using velocity feedforward. MP1390 switches velocity feedforward on in the "Positioning with manual data input," "Program run, single block" and "Program run, full sequence" modes. MP1391 has the same effect for the "Manual" and "Handwheel" modes.

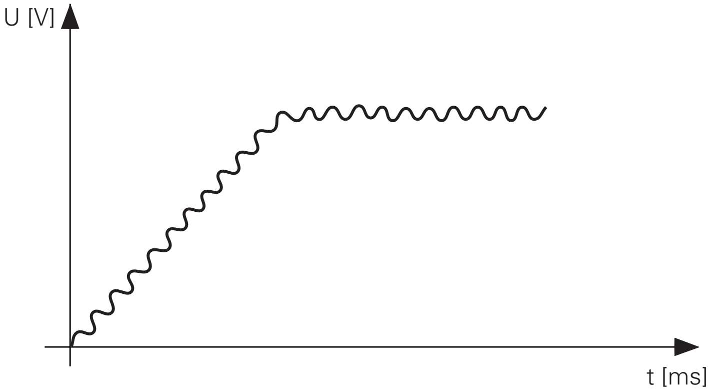
The following block diagram gives a simplified illustration of operation with velocity feedforward.



Analog axes: If the axes are stopped, the integral factor MP1080.x becomes effective as well. It causes an offset adjustment.

Digital axes: There is no offset. MP1080.x is without function.

You influence the fine control of the feedforward-controlled speed nominal value with the k_v factor for velocity feedforward control MP1510.x.



If the k_V factor is too large, the system will oscillate about the feedforward-controlled speed nominal value. Unlike control with servo lag, here you enter the optimum k_v factor for each axis, even for interpolating axes. With MP1515.x you define a second set of k_v factors that you activate with the M function M105. By selecting a higher k_v factor you can selectively increase the contour accuracy. M105 also influences the compensation of reversal peaks in circular movement. M function M106 reselects the original set of k_v factors. You must enable the M functions M105/M106 with MP7440 bit 3.

MP1510.0-8 k_v factor for velocity feedforward

Input: 0.100 to 20.000 $[\frac{m/min}{mm}]$

MP1515.0-8 k_v factor for velocity feedforward control (M105)

Input: 0.100 to 20.000 $[\frac{m/min}{mm}]$

MP7440 Output of M functions
Input: %xxxxx
Bit 3 k_v factor switch-over with M105/M106
0 = function not active
1 = function active

Rapid traverse

In MP1010.x you define the rapid traverse for each axis of the machine. Through the PLC you can reduce this value for special cases by entering the desired reduced value in D596. If the value in D596 is greater than MP1010.x, the value in MP1010.x will remain in effect. After the control is switched on, or after an interruption in the PLC scan, the doubleword D596 is preset with the value 300 000, so that MP1010.x goes into effect.

In MP1020.x you enter the special feed rate for the manual operating mode. In general it is significantly lower than the rapid traverse. The programmed feed rate and the current contouring feed rate are shown in D360 and D388 in mm/min. The maximum possible feed rate always depends on the encoder being used. For digital axes it also depends on the number of pole pairs in the drive and on the pitch of the ballscrew.

v_{max} [mm/min]= P [µm] • f_i [kHz] • 60

v_{max} =	maximum traversing speed
P =	signal period of the encoder
f_i =	encoder input frequency

Digital axes:

$$v_{max}\ [\text{mm/min}] = \frac{18\,000}{\text{number of pole pairs}}\ [\text{1/min}] \bullet \text{screw pitch [mm]}$$

Analog axes: You must adjust the rapid traverse (maximum traversing speed) with the desired analog voltage (for example 9 V) at the servo amplifier. Enter the desired analog voltage for rapid traverse in MP1050.x.

MP1010.0-8 Rapid traverse
Input: 10 to 300 000 [mm/min]

MP1020.0-8 Manual feed rate
Input: 10 to 300 000 [mm/min]

MP1050.0-8 **Analog axes:** Analog voltage for rapid traverse
Input: 1.000 to 9.000 [V]
digital axes: Non-functional
Input: 1

		Set	Reset
D596	Max. feed rate from PLC [mm/min]	NC/PLC	PLC
D360	Programmed feed rate [mm/min]	NC	NC
D388	Current contouring feed rate [mm/min]	NC	NC

Position loop resolution

The encoder signals are interpolated 1024-fold. Therefore:

$$\text{Position loop resolution [µm]} = \frac{\text{Signal period [mm]}}{1024}$$

Analog axes: The TNC must be able to output voltage steps to indicated positional deviation. The 10 V analog voltage is subdivided 65 536-fold with a 16-bit D/A converter. This results in a smallest voltage step of 0.15 mV. As described above, rapid traverse is accompanied by a certain servo lag s_a. Rapid traverse is attained at a certain voltage (MP1005.x). From this a certain voltage ΔU can be calculated for the position error (servo lag).

$$\Delta U = \frac{\text{MP1050 [mV]}}{s_a \text{ [µm]}}$$

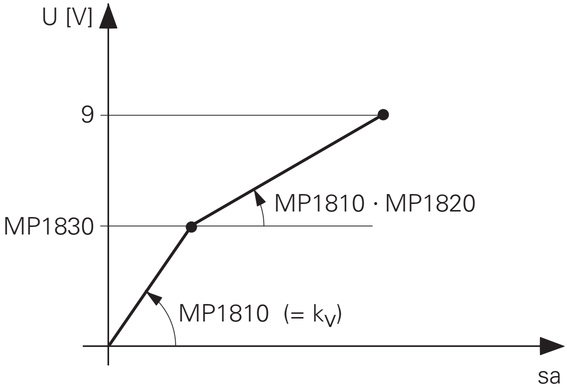
If ΔU is divided by the smallest possible output voltage step (0.15 mV), the result is the number of voltage steps output for the positional deviation.

Characteristic kink (only for control with servo lag):

For machines with high rapid-traverse rates you usually cannot raise the k_V factor enough to give an optimal loop characteristic over the entire range of speeds (stop, machining feed rates, rapid traverse). In such cases you can introduce a kink, which provides the following advantages:

- for lower feed rates, a high k_v factor, i.e. a larger voltage step per µm of positional deviation,
- for higher feed rates (above the machining rates) a lower k_v factor.

Enter the desired position of the kink in machine parameter MP1830. In the higher range the k_V factor is multiplied by the factor from MP1820.x.



The kink point must lie above the range of machining feeds! Under these conditions, the servo lag can be calculated as follows:

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

MP1820.0-8 Multiplication factor for the k_V factor
Input: 0.001 to 1.000

MP1830.0-8 Characteristic kink
Input: 0.000 to 100.000 [%]

Opening the position control loop

After setting M4581 the control loops of all axes and the spindle are opened and an NC stop is carried out. This permits you, for example, to open the position control loops and at the same time switch off the motors. If marker M4580 is set, an external EMERGENCY STOP (X42, pin 4 "control-is-ready") not reported to the NC. Rather, the same function is carried out as with marker M4581. For an open position control loop the axis release is reset in W1024.

				Set	Reset
M4580	Suppress EMERGENCY STOP, open all position control loops, NC stop			PLC	PLC
M4581	Open all position control loops, NC stop			PLC	PLC
W1024	Axis release			NC	NC
	Bit	0 to 8	0 = position control loop not closed		
	Axis	1 to 9	1 = position control loop closed		

Clamping axes

If you wish to lock one axis after running an NC block, you must:

- wait for the axis to come into position (W1026)
- clamp the axis
- open the position control loop (W1040)
- switch off the motor.

Normally a waiting period is required between "axis clamping" and "opening the position control loop." With W1038 you can prepare "opening the position control loop" after the positioning window has been reached. If the corresponding bit is set in W1038 for one axis, the next NC block will not be run until the target position has been reached and the position control loop has been opened with W1040.

With an open position control loop the axis enabling is reset in W1024. You can, therefore, link switching off the motors with W1024.

If a clamped axis is to be repositioned, the NC will reset the axis-in-position message in W1026. After this message has been reset you must:

- switch on the motors
- unlock the axis
- close the position control loop.

The motors are switched on and off through a PLC output for analog axes, and with Module 9161 for digital axes.

				Set	Reset
W1038	Preparing to open the position control loop			PLC	PLC
	Bit	0 to 8	0 = not active		
	Axis	1 to 9	1 = active		
W1040	Open the position control loop for each axis			PLC	PLC
	Bit	0 to 8	0 = do not open the control loop		
	Axis	1 to 9	1 = open the control loop		

Feed-rate enable

Before the axes can move, you must set the "feed-rate enable" through the PLC. As long as the feed-rate is not enabled, the nominal velocity output value remains at zero. In the status display the letter F is highlighted as long as the feed rate has not been enabled for any of the axes.

With M4563 you enable the feed rate for all axes. With W1060 you can set the feed-rate enable for specific axes if M4563 is reset.

				Set	Reset
M4563	Feed rate enable for all axes			PLC	PLC
W1060	Axis-specific feed-rate enable			PLC	PLC
	Bit	0 to 8	0 = no feed rate enable		
	Axis	1 to 9	1 = feed rate enable		

Actual position capture

In the Manual and Electronic Handwheel modes you can transfer the current position value to the nominal position value with W1044. You use this function if, for example, the axis was moved while the position control loop was open.

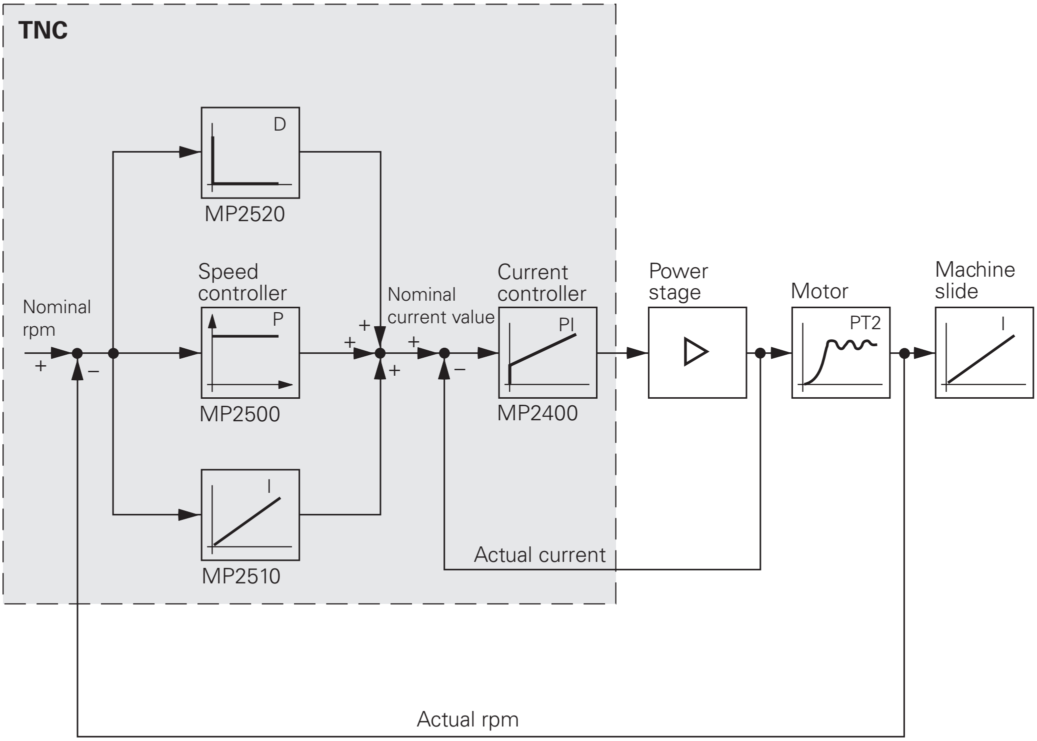
				Set	Reset
W1044	Actual position capture			PLC	PLC
	Bit	0 to 8	0 = no actual position capture		
	Axis	1 to 9	1 = actual position capture		

4.8.3 Motor Speed Controller (Only TNC 426 PB, TNC 430 PA)

Digital speed controllers are integrated in the TNC 426 PB, TNC 430 PA:

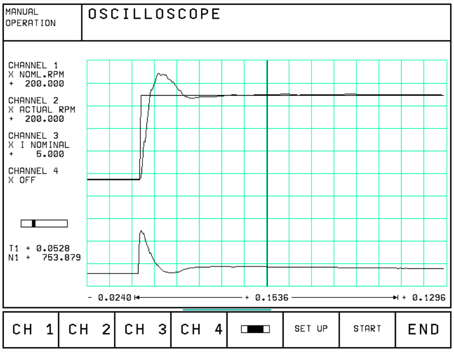
TNC 426 PB:	Digital speed controllers for 5 axes and 1 spindle
TNC 430 PA:	Digital speed controllers for 6 axes and 1 spindle

The actual rotational speed values are measured with HEIDENHAIN rotary encoders directly at the motors. The speed controller evaluates the difference between the nominal and actual speed value. It outputs the nominal current value. The position controller provides the rotational speed nominal value.



With MP2500.x (proportional component) and MP2510.x (integral component) you adjust the step response of the speed controller so that only one overshoot is visible and the settling time is as small as possible. 3 ms to 15 ms are realistic results for the settling time.

For more information on commissioning the speed controller, see section "Commissioning."



```
MANUAL
OPERATION    OSCILLOSCOPE

CHANNEL 1
X NOML.RPM
+  200.000
CHANNEL 2
X ACTUAL RPM
+  200.000
CHANNEL 3
X I NOMINAL
+    5.000
CHANNEL 4
X OFF

T1 + 0.0528
N1 +  753.879

- 0.0240 |<---------- + 0.1536 ---------->| + 0.1296
```

The step response shown in the above illustration is strongly idealized. In practice, disturbance oscillations are superposed on the step response. You can reduce these disturbance oscillations with the differential factor and the PT2 element on the load side of the speed controller.

With Module 9164 you can read the actual speed value of the motors. The resolution of the actual speed value is dependent on the line count of the rotary encoder used:

$$\text{Resolution} = \frac{1}{\text{line count} \bullet 1024} \bullet 100\,000\ [1/\text{min}]$$

Call:

```
PS   B/W/D/K    <Axis>      0 to 14, 15 = axis 1 to 5, spindle
CM   9164
PL   B/W/D      <Actual speed value in the format 0.001 [1/min]>
```

MP2500.0-5 Proportional factor of the speed controller
Input: 0 to 100.000 [As]

MP2510.0-5 Integral factor of the speed controller
Input: 0 to 100 000 [A]

Differential factor

Normally the speed controller does not need a differential factor. The differential factor can reduce low-frequency oscillations. However, it increases the tendency to oscillate in higher frequency ranges. You must therefore make very sure that the system is sufficiently stable.

The use of the differential factor is not recommended for machines with belt drive, since the aging and temperature influence is too great.

Approximation formula for estimating the differential factor:

$$\text{MP2520.x} \approx \frac{T \bullet \text{MP2500.x}}{8}$$

MP2520.x = differential factor of the speed controller [As²]

MP2500.x = proportional factor of the speed controller [As]
T = oscillation period of the lowest interfering frequency [s]

MP2520.0-5 Differential factor of the speed controller
Input: 0 to 1.0000 [As²]

PT2 element of the speed controller

If the controlled system is insufficiently dampened (e.g. direct coupling of the motors or roller bearings) it will not be possible to attain a sufficiently short settling time when adjusting the step response of the speed controller. The step response will oscillate even at a low proportional factor.

With MP2530.x you can dampen these high-frequency disturbance oscillations. The value in MP2530.x should not be unnecessarily high since otherwise the k_v factor of the position controller is restricted. Realistic values for MP2530.x are 0.0003 to 0.0020.

MP2530.0-5 PT2 element of the speed controller (second-order time-delay element)
Input: 0 to 1.0000 [s]

Frequency filter

On critical axes oscillations can arise in a frequency range that can neither be compensated with the differential factor or the PT2 element. In this case the frequency filter can be of help. You can determine the basic frequency of the disturbance oscillation with the oscilloscope of the TNC, and enter this in MP2550.x. You then increase MP2540.x enough to minimize the disturbance oscillation. You should not increase the dampening to an unnecessarily high level, as this limits the dynamic performance of the control loop. Realistic input values would be three to nine dB.

MP2540.0-5 Frequency filter damping
Input: 0.0 to 18.0 [dB]

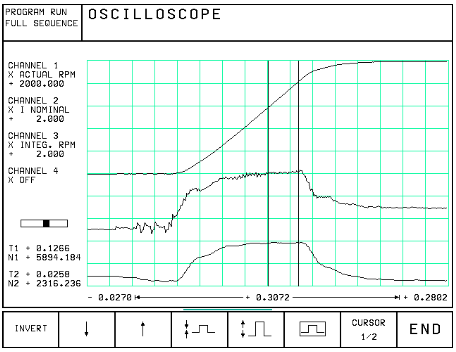
MP2550.0-5 Frequency filter for mean frequency
Input: 0.0 to 999.9 [Hz]

Acceleration feedforward

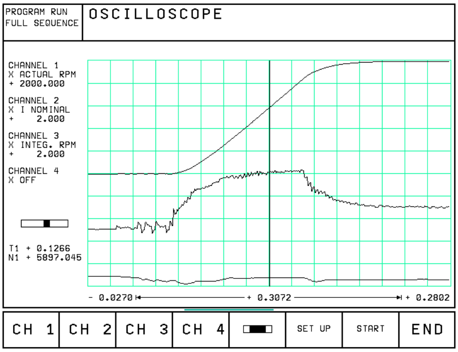
Acceleration feedforward is active only in velocity feedforward control (MP1390 = 0). Every change in velocity results in brief spikes in the servo lag. With acceleration feedforward, which functions parallel to the speed controller, you can minimize these spikes.

Before you adjust the acceleration feedforward, you must have adjusted the friction compensation with machine parameters MP2610.x to MP2620.x.

The input value for MP2600.x for acceleration feedforward is then calculated from the integral-action component of the nominal current value I (INT RPM). For this purpose, I (INT RPM) is recorded with the TNC's integral oscilloscope during a positioning movement. For better illustration, the actual speed value V (ACT RPM) and the nominal current value (I NOMINAL) are also recorded.



MP2600.x = 0



MP2600.x = optimal

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

I (INT RPM) =	Integral-action component of the nominal current value
t =	Acceleration time (ramp)
V (ACT RPM) =	Actual speed value after acceleration
MP2020.x =	Path of traverse per motor shaft revolution

MP2600.0-5 Acceleration feedforward
Input: 0 to 3.000 [A / (U/s²)]

Limiting the integral factor

In machines with a lot of stiction, a high integral-action component can result if there is a position deviation at standstill. This can lead to a jump in position when the axis "takes off". In such cases the integral-action component of the speed controller can be limited with MP2512.x.

MP2512.0-5 Limiting the integral-action component of the speed controller
Input: 0.000 to 30.000 [s]
Realistic values: 0.1 to 2.0

Holding current

The electrical current that is necessary to keep a vertical axis stationary is the holding current. This current is automatically preset by the TNC through the integral-action component of the nominal current value (N INTEG.)
In the event of an EMERGENCY STOP during axis movement, however, this integral action component is switched off immediately. If this occurs on your machine and the axis sinks, you must enter the required holding current in MP2630.x.

To find the proper value for the holding current, proceed as follows:
To exclude the effects of static friction, measure the current at low speed in both directions (e.g. ±10 rpm). The value for the holding current is calculated from the mean of the two measured current values:

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

MP2630.0-5 Holding current
Input: –30.000 to +30.000 [A]

Enabling the drive control loop (Module 9161)

With Module 9161 you must switch on the integrated drive controller (speed and current controller). You can use, for example, the "axis enabling" word W1024 as criteria for enabling the drive control loop. Also for enabling the drive control loop, terminal 1 on connection X50 must be supplied by 24 Vdc. The drive control loop cannot be switched on if there is no ready signal or if there is no voltage supply to connection X50. You can switch the drive controller on and off for the specific axes with module 9161. All of the drive controllers are automatically switched off if you remove the voltage supply to connection X50, terminal 1. They are only then switched on again when the voltage supply is reconnected and the PLC module 9161 activated. In order that the removal of voltage to X50 be recognized in the PLC, the drive control loop enabling signal must also be connected to a PLC input (see the Basic Circuit Diagram in the Appendix). Another contact assembly should be used for this, so that contact problems can be identified.

If you switch off the controller motor, the axis is decelerated and after coming to a halt the speed and current control loops are opened. Then the power stage is switched off by the $\overline{\text{Reset}}$ signal. With Module 9162 you can inquire whether the drive control loop was really switched on.

Call only from sequential program:

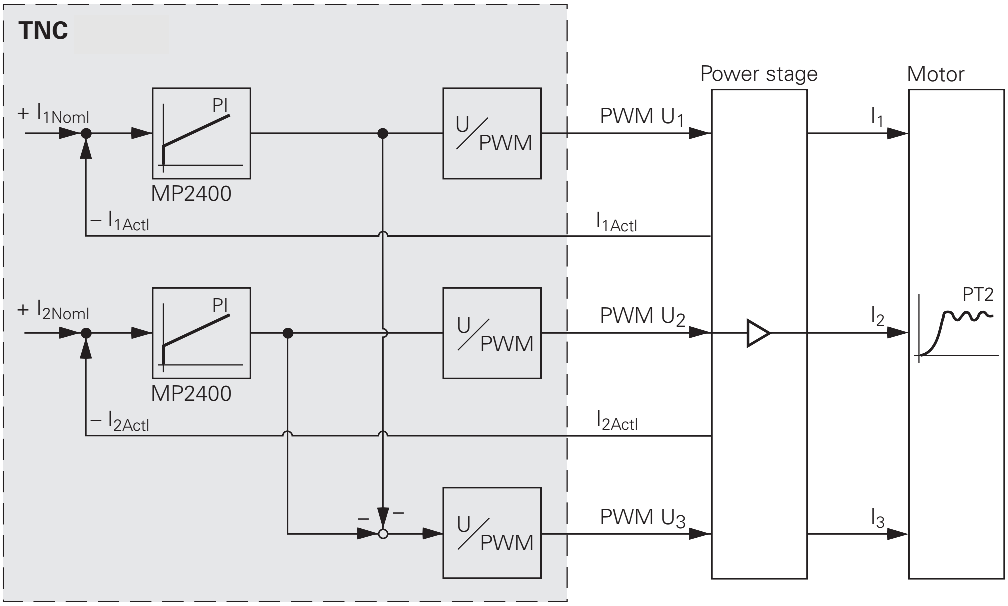
```
PS    W/D/K        <enabled axes>
                   Bit:    15              0              0 = do not enable the motor controller
                   Axis:   Sxxxxxx987654321               1 = enable the motor controller
CM       9161
```

4.8.4 Current Controller (Only TNC 426 PB, 430 PA)

Analog current controllers are integrated in the TNC 426 PB, TNC 430 PA:

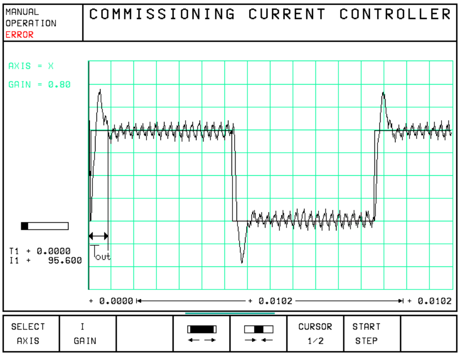
TNC 426 PB: Analog current controllers for 5 axes and 1 spindle
TNC 430 PA: Analog current controllers for 6 axes and 1 spindle

The phase currents I_1 and I_2 are controlled and the phase current I_3 is determined from I_1 and I_2 (the sum of all phase currents = 0). The actual current values for I_1 and I_2 are determined by the motor power stage and provided as voltage signals for X51 to X56 and X61. The phase current signals I_1, I_2 and I_3 are transferred as PWM signals to the power stage of the motor.



You design the current controller according to the optimum result, with the position and speed controller switched off. The step response shows a slight overshoot with a short rise time and settling time.

The settling time t_{out} should be less than the cycle time of the speed controller (600 µs). In MP2400.x you enter the current gain at standstill. For information on putting the current controller into service, see "Commissioning."


MANUAL
OPERATION
ERROR

COMMISSIONING CURRENT CONTROLLER

AXIS = X
GAIN = 0.80

T1 + 0.0000
I1 + 95.600

It can happen with certain asynchronous motors that the motor will not run as smoothly at high speeds. If this is the case enter a higher current gain in MP2402.x than in MP2400.x. At maximum speed the current gain in MP2402.x will be reached. It is interpolated linearly between standstill and maximum speed. If you enter the value zero in MP2402.x, the current gain from MP2400.x will be effective over the entire speed range.

MP2400.0-5 Gain for current controller at standstill
Input: 0.00 to 9 999.99 [V/A]
0 = controller disable

MP2402.0-5 Gain for current controller at maximum speed
Input: 0.00 to 9 999.99 [V/A]
0 = value from MP2400.x

4.9 Offset Adjustment

Digital axes:
If an offset exists at the output of the current controller, this is automatically compensated.

Analog axes:
If an offset exists at the output of the position controller, there are several possible methods of compensating it. The maximum permissible offset-voltage in the control is 100 mV. If this voltage is reached or exceeded, then the error message **`Excessive offset in <axis>`** will appear.

The automatic cyclical offset-adjustment and the adjustment via integral factor must not be active simultaneously!

4.9.1 Offset Adjustment by Code Number

With the code number 75 368 you activate an offset adjustment. After entering the code number the control shows the offset values for the axes X, Y, Z, 4, 5 in the dialog line. The values indicate the voltage in 0.15 mV units. Thus a display of 10 means 10 x 0.15 mV = 1.5 mV. The displayed offset value consists of the offset values from the motor controller and in the NC control.

On pressing the ENT key or the CONTINUE soft key the offset values are automatically compensated. The control puts out an appropriate compensating voltage. To switch off the automatic offset adjustment, enter the code number and press the NO ENT key or the QUIT soft key. If you have entered the code number but do not want any changes, press the END soft key.

The offset values are stored in the control and are non-volatile. If the control is exchanged, the code number must be entered to reactivate the offset adjustment.

4.9.2 Automatic Cyclic Offset Adjustment

With MP1220 you program a time interval after which an offset adjustment will be performed cyclically. An automatic adjustment will be carried out if the programmed time has elapsed and the following conditions are fulfilled:

- All axes are stopped
- The spindle is not switched on
- The axes are not clamped

For each adjustment cycle there will be a 1 mV correction if the offset voltage is larger than 1 mV. If the offset voltage is smaller than 1 mV, then compensation steps of 0.15 mV will be used.

MP1220 **Analog axes:** Automatic cyclic offset adjustment
Input: 0 to 65 536 [s]
0 = no automatic adjustment

Digital axes: Non-functional
Input: 0

4.9.3 Offset Adjustment with Integral Factor

With the integral factor MP1080.x you carry out an automatic offset adjustment. It is only effective in the stop condition (see block diagram of control loop). Depending on the size of the factor the offset voltage will be reduced quickly or slowly. Even a small amount of play in the drives can lead to instability in the control loop. In this case an integral factor of zero is entered.

MP1080.0-8 **Analog axes:** Integral factor
Input: 0 to 65 535

Digital axes: Non-functional
Input: 0

4.10 Contour Behavior

4.10.1 Radial Acceleration

In addition to the normal acceleration of the axes in MP1060.x you also define the radial acceleration in MP1070. MP1070 limits the feed rate at circular movements according to the following formula:

$v = \sqrt{r\,[m] \bullet \mathrm{MP1070}\,[m/s^2]}$ v= feed rate for circular movements [m/s]
r= radius [m] (path of the tool center)

It is recommended that a value should be entered which is between the half of and the same as that in MP1060 (Acceleration). If the programmed feed is lower than that above, then the programmed feed will be used. MP1070 is effective for operation with lag and with velocity feedforward.

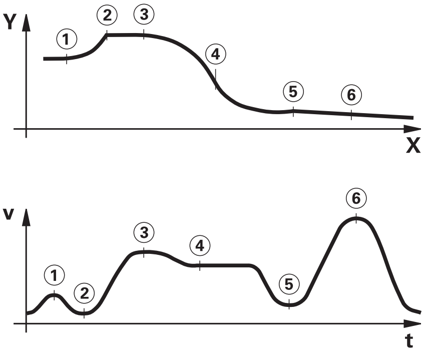
MP1070 Radial acceleration
Input: 0.001 to 5.000 [m/s²]

4.10.2 Contour Speed in Corners

NC software 280 472 ..

To ensure that the tolerance defined in MP1096 is not exceeded, the TNC reduces the speed accordingly in corners. This is facilitated by the TNC's ability to calculate up to 128 blocks ahead (look ahead). Speed can be reduced at sharp corners, tangential line/circle transitions as well as tangential circle/circle transitions. The tolerance value defined in MP1096 can be overwritten by the user with Cycle 32 "Tolerance" if required.

With its jerk limiting function and its nominal position value filter, the TNC is well suited to the high-speed machining of 3-D contours comprising short linear segments. In order that cutter-compensated outside corners remain exact, a spline must be inserted into the cutter midpoint path instead of a transitional arc. This can be set with MP7680, bit10. A spline also has the advantage of additionally limiting jerk.

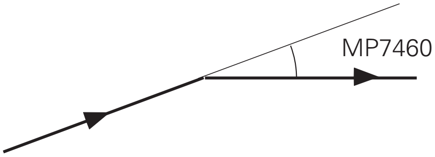


MP1096 Tolerance
Input: 0 = no nominal position value filter
0.001 to 3.000 [mm] = permissible tolerance at contour transitions

MP7680 Machine parameter with multiple function
Input: %xxxxxxxxxxx
Bit 10 Cutter compensated outside corners
0 = inserting an arc
1 = inserting a spline curve
Suggested input value = %1xxxxxxxxxx

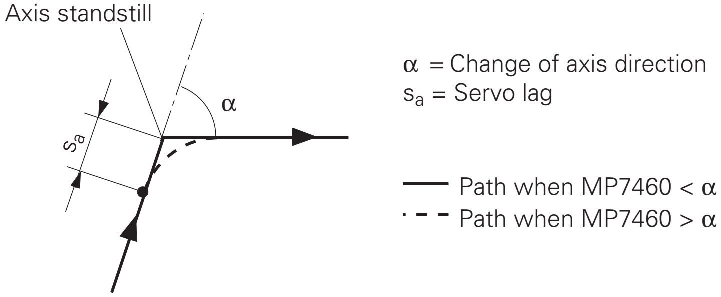
NC software 280 470 ..

With machine parameter MP7460 you define the angle which can still be traversed at constant speed. The permissible size of the angle depends on the machine's drives. MP7460 is effective for external corners without radius compensation and internal corners with radius compensation. It is effective for control with servo lag and with velocity feedforward.



Realistic values are 5° to 15°.

The contour is as shown below:



MP7460	*Constant feed rate in corners*
Input:	*0.0001 to 179.9999 [°]*

Constant feed rate in corners with M90
The miscellaneous function M90 is used in lag mode to enable constant surface speed to be achieved in corners without radius compensation. This results in a rounding of the corners that varies with the feed rate. See your User's Manual for a description of this effect.

Defined curves independent of feed rate are inserted at these corners with the M functions M112 and M124, see User's Manual. The feed rate is not decreased by the insertion of these curves, but twice the amount of NC blocks are generated. You can select with MP7680 bit 7 whether the curves are always to be inserted or only when the acceleration from MP1060.x or MP1070 is exceeded at the corners.

With the M function M132 you can reduce the jerk in axis-specific changes in acceleration along the contour. M132 can be switched off again with M133.

With MP7680, bit 8 you can select whether a curve or a cubic spline should be inserted between the straight lines during the M112 function. The cubic spline also causes a decrease in jerk, but requires more calculating time than a curve. If you have selected the cubic spline with bit 8, you can then set with bit 9 whether the jerk should remain constant along the spline or not. If the jerk remains constant, the speed will be adapted accordingly.

MP7680	*Machine parameter with multiple function*	
	Input: %xxxxxxxxx	
Bit 7	*Inserting the curve or spline defined in M112*	
	0 =	*curves defined in M112 are always inserted*
	1 =	*curves defined in M112 are only inserted if the acceleration from MP1060.x or MP1070 has been exceeded*
Bit 8	*M112: Inserting a curve or cubic spline*	
	0 =	*a curve is inserted with M112*
	1 =	*instead of the curve, a cubic spline is inserted with M112*
Bit 9	*M112: Jerk constant along the spline (bit 8 = 1)*	
	0 =	*jerk not constant*
	1 =	*jerk constant*

4.11 Monitoring Functions

The NC monitors the axis positions and the dynamic behavior of the machine. If the fixed values are exceeded, it displays an error message and stops the machine. You can switch off the following types of monitoring for the individual axes with W1042:

- Position monitoring
- Standstill monitoring
- Movement monitoring
- Nominal speed monitoring

				Set	Reset
W1042	Deactivate monitoring functions			PLC	PLC
	Bit	0 to 8	0 = monitoring functions active		
	Axis	1 to 9	1 = monitoring functions inactive		

Under certain circumstances the reaction time of the PLC for switching off the monitoring function (21 ms) may be too slow. In such cases you must use a fast PLC input. Fast PLC inputs are interrogated in the control loop cycle time of 3 ms. In MP4130.0 you enter the number of the PLC input that should be polled more quickly. In MP4131.0 you define the criterion for activating the input. In W522, bit 0 you must enable the function for switching off the monitoring functions with the fast PLC input that was defined in MP4130.0. Once you have accomplished this, as soon as the input is set the monitoring functions are switched off. Also, the axes are automatically stopped, the drives switched off, and an automatic nominal and actual value transfer is carried out. This automatic nominal and actual value transfer is only carried out if the servo lag is greater than MP1030 (positioning window). The monitoring functions are not activated again until the fast PLC input is reset or the function is disabled with W522, bit 0.

It is not possible to operate the machine safely with the monitoring functions switched off, since the TNC does not recognize noncontrolled movements of the axes.

MP4130.0 Number of the fast PLC input for switching off the monitoring functions
Input: 0 to 255 [No. of the PLC input]

MP4131.0 Criterion for activating a 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	PLC	PLC
Bit 0	Fast PLC input defined in MP4130.0 for switching off the monitoring functions		

4.11.1 Position Monitoring

The position of the axes is monitored by TNC as long as the control loop is active. You define two ranges in machine parameters for position monitoring. If the first limit is exceeded the error message **`Excessive servo lag in <axis>`** appears and the machine stops. You can erase this error with the CE key. If the second limit is exceeded the blinking error message **`Excessive servo lag in <axis>`** appears and the control-is-ready output is reset. You cannot erase this message. To correct the error you must switch off the control.

Since the input values depend on the highest possible servo lag for the position monitor there are different input ranges for control with servo lag and control with velocity feedforward. You must adjust the input values to the dynamics of the machine.

If blocked axes cause the erasable error message **`Excessive servo lag in <axis>`**, a nominal speed value may freeze, since the machine axes can no longer be moved. In MP1150 you define the time after which the nominal speed value is erased. After this time an automatic actual and nominal value transfer is carried out. If the error message is erased with the CE key before this time elapses, an automatic actual and nominal value transfer is carried out and the nominal speed value is erased.

MP1710.0-8 Position monitoring for control with servo lag (erasable)
Input: 0.0000 to 300.0000 [mm]
Recommended: 1.2 • servo lag

MP1720.0-8 Position monitoring for control with servo lag (EMERGENCY STOP)
Input: 0.0000 to 300.0000 [mm]
Recommended: 1.4 • servo lag

MP1410.0-8 Position monitoring for velocity feedforward control (erasable)
Input: 0.0010 to 30.0000 [mm]
Recommended: 0.5 mm

MP1420.0-8 Position monitoring for operation with velocity feedforward control (EMERGENCY STOP)
Input: 0.0010 to 30.0000 [mm]
Recommended: 2 mm

MP1150 Delay time for erasing the nominal velocity value after an erasable error message **`Excessive servo lag in <axis>`**
Input: 0 to 65.535 [s]
Recommended: 0

4.11.2 Nominal Speed Monitoring

For the axes, monitoring the nominal speed is only possible in operation with velocity feedforward control. For the spindle, nominal speed monitoring is possible in operation with servo lag insofar as the position control loop is closed (orientation). If the nominal speed value calculated by the position controller is greater than the maximum possible value, the flashing error message **`Nominal speed value too high <axis>`** is displayed and the output "control is ready for operation" is reset.
Analog axes: Maximum nominal speed value = 10 V
Analog spindle: Maximum nominal speed value = 20 V
Digital axes and spindle: Maximum nominal speed value = maximum motor speed value

4.11.3 Movement Monitoring

Movement monitoring is possible both in velocity feedforward control and in control with servo lag. During movement monitoring, the actual path traveled is compared with a nominal path calculated by the NC at short intervals (several servo cycles). If during this period the actual path traveled differs from the calculated path, the flashing error message **`Movement monitoring error in <axis> A`** appears.

Analog axes: An existing offset at a standstill may cause a potential at the analog output without any resulting positioning movement. In MP1140.x you must therefore enter a threshold from which the movement monitoring should go into effect.

Digital axes: Since there is no offset, always enter the minimum value in MP1140.x. On digital axes, the calculated position from the pulses of the position encoder (MP331.x, MP332.x) is compared with that calculated from shaft speed encoder. If this difference is greater than the input value from MP2800.x, the error message **`Movement monitoring error in <axis> B`** appears.

If you enter the maximum values in MP1140.x and MP2800.x, you effectively deactivate movement monitoring. It is not possible to operate the machine safely without movement monitoring.

MP1140.0-8 Threshold from which movement monitoring is effective
Input: **Analog axes:** 0.030 to 10.000 [V]
Digital axes: 0.030 to 10.000 [1000/min]

MP2020.0-5 Path for a motor revolution
Input: **Digital axes:** 0 to 100.000 [mm] (or as formula, for input format see Chapter 9 Machine Parameters)
Analog axes: No function

MP2800.0-5 Movement monitoring for position and rotational speed
Input: **Digital axes:** 0 to 99 999.9999 [mm]
0 = no monitoring
Analog axes: No function

4.11.4 Standstill Monitoring

The standstill monitoring operates both with velocity feedforward control and with servo lag. The monitoring begins when the axes have reached the positioning window. As soon as position deviation is greater than the value stored in MP1110, the flashing error message **`Standstill monitoring error in <axis>`** is displayed. The message also appears if, on running in to a position, an overshoot occurs that is larger than the value in MP1110.x, or if the axis moves in the opposite direction on beginning a positioning movement.

MP1110.0-8 Standstill monitoring
Input: 0.0010 to 30.0000 [mm]

4.11.5 Positioning Window

The positioning window defines the limits within which the control considers that a position has been reached. After reaching the position the control starts the execution of the next block. You define the size of the positioning window in machine parameter MP1030.x.

MP1030.0-8 Positioning windows
Input: 0.0001 to 2.0000 [mm]

Axes in position
If the axes have reached the positioning window after a movement, the corresponding bits are set in W1026. This also applies for the condition after the control voltage has been switched on. As soon as you start a positioning movement, the NC resets the bits. This also applies for traversing the reference marks. In the "Electronic handwheel" mode, the axis-in-position bit is reset for the active handwheel axis. On contours that can be machined with constant surface speed, W1026 is not set.

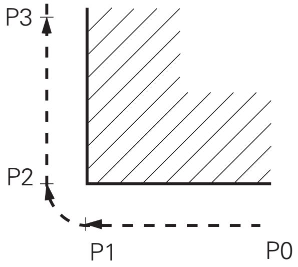
				Set	Reset
W1026	Axes in position			NC	NC
	Bit	0 to 8	0 = axis not in positioning window		
	Axis	1 to 9	1 = axis in positioning window		

Axes in motion

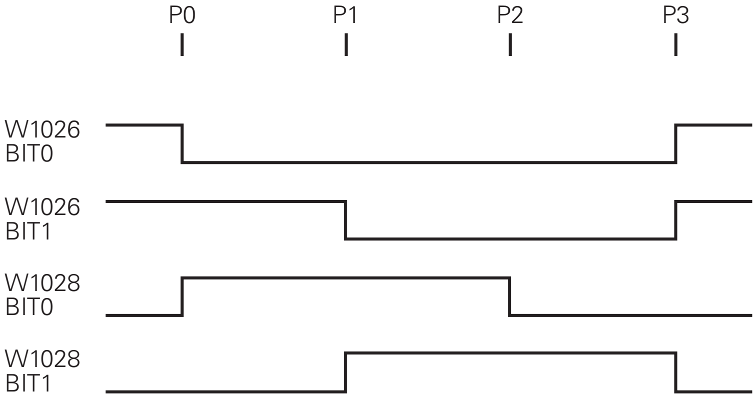
The NC sets the corresponding bits in W1028 during the axis movement.

				Set	Reset
W1028	Axes in motion			NC	NC
	Bit	0 to 8	0 = axis not in motion		
	Axis	1 to 9	1 = axis in motion		

Example of "axis in position" and "axis in motion":



```
.
.
.
L X80  Y20  RL
L X0        RL
L X80       RL
.
.
.
```



4.11.6 NC Supply Voltage Monitoring

The NC supply voltage must lie within the defined range (see chapter "Mounting and Electrical Installation"). Monitoring NC operating voltage is however only worthwhile with digital axes. A brief voltage surge (approx. 5 s) of up to 720 Vdc is permitted. If the voltage exceeds 720 Vdc, the NC revokes the pulse release ($\overline{\text{Reset}}$) for the IGBTs (insulated gate bipolar transistors) of the power stage. The motors come to stop out of loop and no feedback to the dc link is possible.

If the supply voltage falls below 385 Vdc (power fail) all drives will come to a controlled stop. The PLC outputs are switched off and the control displays the error message **`Power fail`**. If the voltage falls below 155 Vdc, the control is reset. At 135 Vdc the dc-link power supply is switched off.

You can switch the monitoring for supply voltage greater than 385 Vdc on and off with Module 9167. After the control is switched on this type of monitoring is off during the first PLC scan and is then automatically switched on if you have not called module 9167.

Call:

PS	B/W/D/K	<Error code>		
		0:	Switch the monitoring for supply voltage >385 Vdc off	
		1:	Switch it on	
CM	9167			
PL	B/W/D	<Error code>		
		0:	Command was executed	
		–1:	Transferred parameter is invalid	

4.11.7 Temperature Monitoring

Internal temperature of the logic unit
The internal temperature of the logic unit is constantly monitored. At approx. 70° C the flashing error message **`TNC operating temp. exceeded`** .

Motor temperature (digital axes only)
To measure the motor temperature, a KTY 84 thermistor must be connected to X15 to X20 and X60 at pins 13 and 25. The temperature value is sampled at least once every second. The maximum permissible motor temperature is taken from the motor table. If the entered temperature is exceeded, the flashing error message **`Motor temperature too high <Axis>`** appears and the drives are automatically switched off.

With Module 9165 you can interrogate the current motor temperature and, if desired, take appropriate measures before the maximum motor temperature has been exceeded.

Call:

PS	B/W/D/K	<Axis>	0 to 5, 15 = axis 1 to 6, spindle
CM	9165		
PL	B/W/D	<Temperature>	Range: 0 to 255 °C

Temperature of heat sink for the power stage (digital axes only)

Pin 13 of X51 to X56 and X61 provides the $\overline{\text{Temperature warning}}$ signal. If this signal is reset, it means that the permissible temperature of the heat sink for the power stage was exceeded. The manufacturer of the power stage determines when this signal is reset.

Normally the drives are brought to a stop immediately after a temperature warning in order to prevent the power stage from being destroyed. The power stage manufacturer can give you more detailed information. This signal is not evaluated in the NC. Therefore you must interrogate the temperature warning in the PLC with Module 9160 and provide for the appropriate reactions.

4.11.8 I^{2}t Monitoring (Digital Axes Only)

The instantaneous motor current is limited to the smaller value of the power-stage peak current and of the motor peak current. Once the model of power stage and motor have been set in the TNC, it knows these values.

An I^{2}t monitoring is also carried out. For I^{2}t monitoring, the squares of the actual current values are integrated. The duration of integration is 10 s for feed motors and 150 s for spindle motors. The lower value of the power-stage rated current and the "reference value for I^{2}t monitoring of the motor" (MP2302.x, MP2303.x) is used as the I^{2}t limit value. If you enter zero, the I^{2}t monitoring for the motor is switched off. The input value is a factor of the motor rated current (1 = motor rated current).

The I^{2}t warning occurs if the integral calculated from the mean current exceeds the I^{2}t limit, and is canceled when if the integral falls back below the I^{2}t limit. An I^{2}t limitation occurs if the mean current value calculated from the integral exceeds the I^{2}t limit value by a factor of 1.1, and is canceled again when it falls back below 1.1 times the I^{2}t limit value.

There is no I^{2}t limitation on the spindle drive.

MP2302.0-5 Reference value for I^{2}t monitoring of the feed motors
Input: 0 to 1000.000 [• motor rated current]
0 = I^{2}t monitoring for motor switched off
1 = rated current of motor as reference value

MP2303 Reference value for I^{2}t monitoring of the spindle motor
Input: 0 to 1000.000 [• motor rated current]
0 = I^{2}t monitoring for spindle switched off
1 = rated current of motor as reference value

With Module 9160 you request the current status of the temperature monitoring and the I^2t monitoring.

Call:

```
CM  9160
PL  D          <Temperature monitoring>
               Bit:  31               16 15           0
               Axis:                     Sxxxxxxxxxx54321

PL  D          <$I^2t$ monitoring>
               Bit:  31               16 15           0
               Axis  xxxxxxxxxxx54321 Sxxxxxxxxxx54321
                     $I^2t$ limitation     $I^2t$ warning
```

4.11.9 Monitoring: Power Stage, DC-Link Voltage (Digital Axes Only)

Pin 6 on X51 to X56 and X61 provides the Ready signal. The following conditions can reset this signal:

- Missing connection from K9 to K663 on the HEIDENHAIN Interface board.
- Missing voltage from unit bus (FR+)
- Error in the power stage (+5 V or U_Z missing, or U_Z too large)

As soon as the Ready signal is reset, the servo drive controllers are switched off. This means that an error message is output through the position control loop, usually in the form of **`Movement monitoring error in <axis> A`**.

After the servo drive controller is switched off, the $\overline{\text{Reset}}$ signal switches off the PWM signal release. After switching on the servo drive controller with Module 9161 you can inquire the readiness of the servo drive controller with Module 9162. The servo drive controller can only be switched on if there is a voltage supply to connection X50.

```
Call:
CM   9162
PL   B/W/D      <Drive ready>
                Bit:    15               0         0 = not ready
                Axis    Sxxxxxxxxxx54321           1 = ready
```

4.11.10 Instantaneous Utilization of the Servo Drives (Digital Axes Only)

Module 9166 provides you with the present utilization of the given servo motor as a percentage of the rated utilization. The utilization is calculated as:

Speed range	n_{Actl} < rated speed	n_{Actl} >= rated speed
Asynchronous motor	$\frac{\lvert M\rvert}{\lvert M_{Noml}\rvert}$	$\frac{\lvert P\rvert}{\lvert P_{Noml}\rvert}$.
Synchronous motor	$\frac{\lvert M\rvert}{\lvert M_{Noml}\rvert}$	–

Instead of the drive moment, the effective component I_q of the current that is proportional to it is used. I_{qMean} is calculated as the mean of the individual current values I_{qx} of the last 20 ms according to the formula:

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

where: Utilization = $100\% \bullet \frac{I_{qMean}}{I_{qRated}}$

Asynchronous motor:

$I_{qRate} = \sqrt{\text{<Reference value for utilization display>}^2 - \text{<Magnetization current>}^2}$

Synchronous motor:

I_{qRated} = <Reference value for utilization display>

Only with spindle DSP (as of NC software 280 472 04):

The maximum torque is limited to 2.5 • rated torque.

Call:

PS	B/W/D/K	<Axis>	0 to 4, 15 = axis 1 to 5, spindle
CM	9166		
PL	B/W/D	<Utilization of the drive motor in %> For disconnected axes, the value zero is transferred.	

MP2312.0-5 Reference value for utilization display of the feed drive motors
Input: 0 to 1000.000 [• motor rated current]
0 or 1 = reference value is the motor rated current

MP2313 Reference value for utilization display of the spindle drive motor
Input: 0 to 1000.000 [• motor rated current]
0 or 1 = reference value is the motor rated current

4.11.11 EMERGENCY STOP Monitoring

A PLC-input (X42 / 4) and a PLC-output (X41 / 34) with the designation "control is ready" are available in the control for the EMERGENCY STOP-routine. If a malfunction is recognized in the control, the TNC switches the control-is-ready output signal off, a flashing error message appears on the display screen and the PLC-program is stopped. You cannot cancel this error message with the CE key. You must first remove the fault and repeat the switch-on routine.

If the control-is-ready signal is switched off by an event outside the control, the error message **`EMERGENCY STOP`** is displayed and the NC sets M4177 and M4178. Also, a zero nominal speed value is output and the drives are switched off. To erase this error message, you must first switch the control voltage back on.

If marker M4580 is set, an EMERGENCY STOP (control-is-ready input) is not reported to the NC. Instead, the control loops of all axes and the spindle are opened and an NC stop is carried out.

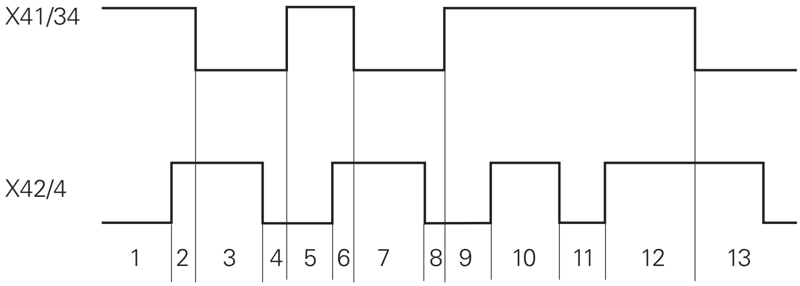
		Set	Reset
M4177	Erasable error message is displayed	NC	NC
M4178	Error message **`EMERGENCY STOP`** is displayed	NC	NC
M4580	Suppress EMERGENCY STOP, open all position loops, NC stop	PLC	PLC

Connection diagram

Under fault conditions the control-is-ready output must trip an EMERGENCY STOP. Because of the great importance of this function, this output is tested by the control every time the line power is switched on. Refer to the basic circuit diagram (Appendix) for the wiring recommended by HEIDENHAIN.

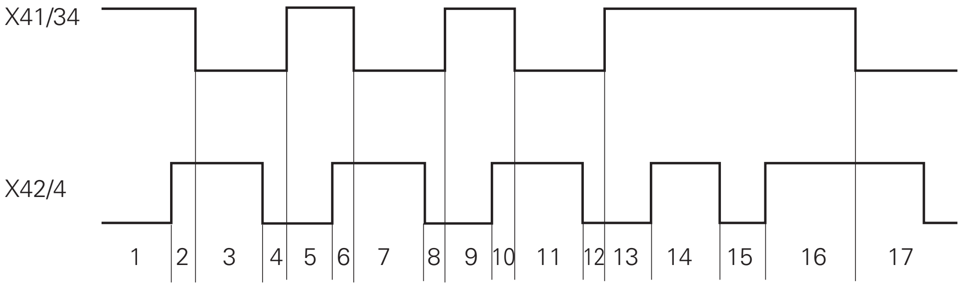
The external electronics must meet the specified conditions. In particular, the acknowledgment for the control-is-ready signal must reach the TNC within 380 ms.

TNC 426: Flowchart



		Display on screen
1	Wait for control voltage.	`Relay ext. dc voltage missing`
2	Recognize the control voltage on X42/4 and switch-off the control-is-ready signal on X41/34 by the main microprocessor (t < 66 ms).	
3	Maximum time during which the acknowledgment of control readiness on X42/4 must go to zero (t < 380 ms).	If exceeded: `EMERGENCY STOP defective`
4	Recognize the acknowledgment and set X41/34 (t < 20 ms).	
5	Wait for control voltage.	`Relay ext. dc voltage`
6	Recognize the control voltage on X42/4. Digital Signal Processor (DSP) switches off the control-is-ready signal on X41/34 (t < 120 ms).	
7	Maximum time during which the acknowledgment of control readiness on X42/4 must go to zero. (t < 380 ms).	If exceeded: `EMERGENCY STOP defective`
8	Recognize the acknowledgment and set X41/34 (t < 120 ms).	
9	Wait for control voltage.	`Relay ext. dc voltage missing`
10	Normal control operation. Control-is-ready output signal and acknowledgment signal are high.	
11	Control voltage is switched off externally.	`EMERGENCY STOP`
12	After switching on the control voltage again, the error message can be erased, followed by normal control operation.	
13	After recognizing a fault, the control switches off the control-is-ready output (X41/34).	Flashing error message

TNC 430: Flowchart



		Display on screen
1	Wait for control voltage.	`Relay ext. dc voltage missing`
2	Recognize the control voltage on X42/4 and switch-off the control-is-ready signal on X41/34 by the main microprocessor (t < 66 ms).	
3	Maximum time during which the acknowledgment of control readiness on X42/4 must go to zero (t < 380 ms).	If exceeded: `EMERGENCY STOP defective`
4	Recognize the acknowledgment and set X41/34 (t < 20 ms).	
5	Wait for control voltage.	`relay ext. dc voltage`
6	Recognize the control voltage on X42/4. Digital Signal Processor (DSP) switches off the control-is-ready signal on X41/34 (t < 120 ms).	
7	Maximum time during which the acknowledgment of control readiness on X42/4 must go to zero. (t < 380 ms).	If exceeded: `EMERGENCY STOP defective`
8	Recognize the acknowledgment and set X41/34 (t < 120 ms).	
9	Wait for control voltage.	`Relay ext. dc voltage missing`
10	Recognize the control voltage on X42/4 and switch-off the control-is-ready signal on X41/34 by DSP 2 (t < 120 ms).	
11	Maximum time during which the acknowledgment of control readiness on X42/4 must go to zero (t < 380 ms).	If exceeded: `EMERGENCY STOP defective`

12	Recognize the acknowledgment and set X41/34 (t < 120 ms).	
13	Wait for control voltage.	`Relay ext. dc voltage missing`
14	Normal control operation. Control-is-ready output signal and its acknowledgment signal are high.	
15	Control voltage is switched off externally.	`EMERGENCY STOP`
16	After switching on the control voltage again, the error message can be erased, followed by normal control operation.	
17	After recognizing a fault, the control switches off the control-is-ready output (X41/34).	Flashing error message

4.12 Spindle

You control the spindle through the PLC. You can output the programmed rotational speed as:

- Code via PLC outputs
- Analog nominal speed value for an analog spindle
- Digital nominal speed value for digital spindle

The analog or digital spindle can also be operated with closed position control loop. In this case, you need a separate position encoder for the spindle. In machine parameter MP3010 you define how you wish to operate the spindle.

MP3010 Output of rotational speed, gear range
Input: 0 = no output of spindle speed
1 = speed code only if the spindle speed changes
2 = speed code for every TOOL CALL
3 = analog output of spindle speed, G code only if the gear stage changes
4 = analog output of spindle speed, G code for every TOOL CALL
5 = analog output of spindle speed, no G code
6 = same as input value 3 but with controlled spindle for orientation
7 = same as input value 4 but with controlled spindle for orientation
8 = same as input value 5 but with controlled spindle for orientation

4.12.1 Position Encoder of the Spindle

The analog or digital spindle can also be operated with closed-loop position controller. In this case you need a separate position encoder for the spindle. The input for this position encoder can be defined in MP111. If you wish to use the speed encoder on a digital spindle as a position encoder, MP110 must be 0. Enter the line count of the rotary encoder in MP3142. Signals with levels of 1 V_{PP} are subdivided by 1024.

In MP3143 you enter the mounting configuration of the spindle position encoder . To ensure adequate accuracy, the position encoder should be mounted directly on the spindle. If the machine design prevents this, you must enter in MP3450.x and MP3451.x the encoder-to-spindle transmission ratio for each gear range. In this case, the encoder outputs several reference pulses per revolution of the spindle. The reference mark evaluation is then carried out with Module 9020 (see Reference Marks).

If MP3143 = 2, the reference pulse release for the spindle position encoder is set with X30 pin 1. In this way you can be sure that the same reference signal is always evaluated. If MP3143 = 1, then X30 pin 1 is evaluated as a reference signal. In this case the reference mark of the position encoder is not evaluated. Due to the low accuracy of such a solution, this setting is not to be recommended. Input value 3 has the same effect as input value 1, however in this case the TNC waits to evaluate the second reference pulse.

MP3142 Line count of the position encoder on the spindle
Input: 100 to 9 999 [lines]

MP3143 Mounting configuration of spindle position encoder
Input:
0 = position encoder directly on the spindle
1 = position encoder via transmission (transmission ratios in MP3450.x and MP3451.x). X30 pin 1 = reference pulse
2 = position encoder via transmission (transmission ratios in MP3450.x and MP3451.x). X30 pin 1 = reference pulse release
3 = as input value 1, however the TNC waits for the second reference pulse

MP3450.0-7 Number of revolutions of the spindle position encoder
Input: 0 to 255
0 = no transmission

MP3451.0-7 Number of revolutions of the spindle
Input: 0 to 255
0 = no transmission

Module 9042: Reading the spindle coordinates

With Module 9042 you can read the spindle coordinates. The coordinate values for actual value, nominal value, actual value in the reference system, servo lag, and spindle distance-to-go are stored in five consecutive doublewords beginning from the given target address. The values for actual value, nominal value, 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°. The display is in 1/1000 degree format.

If MP3010 <6 (no controlled spindle), all coordinates are read as zero. During controlled operation (M03/M04 active or M05 and opened position control loop) the nominal value corresponds to the actual value; servo lag and distance-to-go are zero.

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

4.12.2 Analog and Digital Spindle Control

You can program spindle speeds of 0 to 99 999.999 rpm both for analog and digital nominal speed value output. If you have selected nominal speed value output in MP3010 (entry 3 to 8), then M4003 is set. The programmed speed is stored in D356 (in units of 0.001 rpm). The nominal speed value is stored in W320 (in rpm). You will find the actual speed value in rpm in W322.

Analog spindle: The nominal speed value of the motor is output as an analog dc voltage of ±10 V to socket X8.

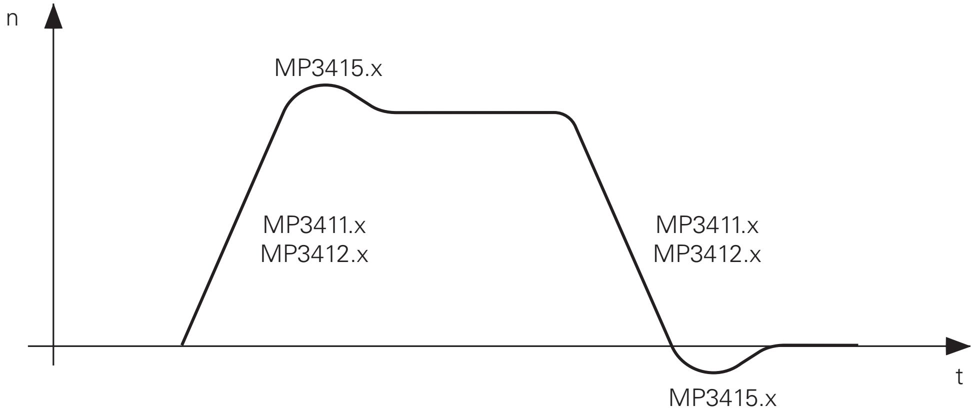
Digital spindle: The nominal speed value of the motor is transferred to the internal speed controller.

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

Controlled spindle nominal speed

The nominal speed is controlled in a closed loop only for spindle orientation. In all other modes of operation it is controlled in an open loop, which means that the actual speed of the spindle (of its position encoder) is not measured. The NC also outputs a nominal speed value when the drive is not switched on (Module 9161).

With MP3411.x you define for each gear range the ramp gradient for the nominal speed command signal with M03 and M04. With MP3412.x you then set a multiplier for MP3411.x for M05, spindle orientation and tapping, because a different curve is usually desired in these modes of operation. The same factor applies for all gear ranges. With MP3415.x you determine the transient response for each operating mode individually.



For M03, M04 and M05, set MP3411.x such that the motor accelerates and decelerates within the current limitation. Enter a value for MP3415.0 that results in only one overshoot. If the spindle speed nominal value is in the acceleration or deceleration ramp, marker M4001 is reset. This also applies when the speed is changed through the override potentiometer. If the spindle speed nominal value zero is output, M4002 is set.

MP3411.x Ramp gradient of the spindle with M03 and M04
Input: **analog axes:** 0 to 1.999 [V/ms]
digital axes: 0 to 1.999 [(1000/min)/ms]

MP3412.0 Multiplier for MP3411.x for M05
Input: 0.000 to 1.999

MP3415.0 Transient response of the spindle with M03, M04 and M05
Input: 0 to 1000 [ms]

		Set	Reset
M4001	Spindle speed nominal value not in the ramp	NC	NC
M4002	Spindle speed nominal value = 0	NC	NC

Direction of spindle rotation

With MP3130 you define the polarity of the nominal speed command signal. In MP3140 you enter the counting direction of the position encoder output signals. The nominal speed value is output as soon as you set M4005 for M03 or M4006 for M04. With M4007 for M05 the nominal speed value zero is output (spindle stop).

With M4005 to M4007 you also control the display of the miscellaneous function in the status window (see section "Display and Operation"). If more than one of these markers is set at the same time, the error message `PLC: M4005, M4006, M4007 incorrect` appears. With M4014 you can reverse the direction of spindle rotation, i.e. the polarity of the nominal speed value is inverted (e.g. to set the transmission ratio for differing horizontal and vertical spindles). M4019 inverts the count direction of the position encoder on the spindle.

MP3130 Polarity of the nominal speed command signal for the spindle
Input: 0 = M03 positive; M04 negative
1 = M03 negative; M04 positive
2 = M03 and M04 positive
3 = M03 and M04 negative

MP3140 Counting direction of spindle position encoder 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	Invert the count direction of the spindle position encoder	PLC	PLC

Disable speed output for spindle

With M4008 you disable the speed output for the spindle. At the same time, M03, M04 or M05 is shown inverted. 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 you enter for each gear range the rated speed for S-override at 100%. If you do not need all gear ranges, enter a zero in the remaining machine parameters. In MP3210.x you enter for each gear range the S-analog voltage or the motor revolutions at rated speed. With MP3240.1 you define the minimum nominal speed value for the motor. With MP3120 you define whether the programmed speed zero is permitted. If an illegal speed is programmed, M4004 is set and the error message `WRONG RPM` appears.

MP3510.0-7 Rated speed for gear ranges
Input: 0 to 99 999.999 [rpm]

MP3210.0-7 **Analog spindle:** Nominal value voltage at rated speed
Input: 0 to 9.999 [V]
Digital spindle: Motor revolutions at rated speed
Input: 0 to 100.000 (1000/min)

MP3240.1 **Analog spindle:** Minimum nominal value voltage
Input: 0 to 9.999 [V]
Digital spindle: Minimum motor speed
Input: 0 to 9.999 [1000/min]

MP3120 Permissibility of zero speed value
Input: 0:S = 0 permitted
1:S = 0 not permitted

		Set	Reset
M4004	Illegal speed was programmed	NC	NC

Gear changing

You control gear changing through the PLC. In accordance with the programmed speed the NC sets the current gear range in W256. The gear range is calculated using MP3510.x. With MP3010 you must select the output of the gear range. When the gear range is changed the NC set the G-strobe (M4070). The NC program is not continued until you send the "gear change completed" acknowledgment with M4090. As soon as you set M4090 the G-strobe (M4070) is reset by the NC.

In the PLC program you can change the programmed speed and the gear range calculated by the NC. This may be necessary, for example, with horizontal-/vertical spindles with two separate transmissions. The NC stores the programmed speed in doublewords D356 and D756. Enter the speed that you desire in D756 and the desired gear range in W256. With M4134 you activate your entries in D756 and W256. After the NC has reset M4134 you shift the gears and then acknowledge with M4090 "gear change completed." Please note that the speed that you choose falls within the limits of the gear range that you have chosen.

A changing nominal speed value can be output to shift gears. To do this you must alternately set and reset markers M4009 and M4010. It is best to do this 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 you define the nominal speed value to be output with M4009 / M4010 to the spindle motor.

			Set	Reset
W256	Gear code		NC/PLC	NC/PLC
D356	Programmed speed [0.001 rpm]	NC	NC	
D756	Programmed speed or speed of the PLC [0.001 rpm]		NC/PLC	NC/PLC
M4070	Strobe signal for gear code		NC	NC
M4090	Acknowledgment "gear change completed"		PLC	PLC
M4134	Activation of a gear range and speed through the PLC		PLC	NC
M4009	Spindle rotation left (for gear shifting)		PLC	PLC
M4010	Spindle rotation right (for gear shifting)		PLC	PLC

MP3240.2 **Analog spindle:** Jog voltage for gear shifting (M4009/M4010)
Input: 0 to 9.999 [V]
Digital spindle: Motor speed for gear shifting (M4009/M4010)
Input: 0 to 9.999 [1000/min]

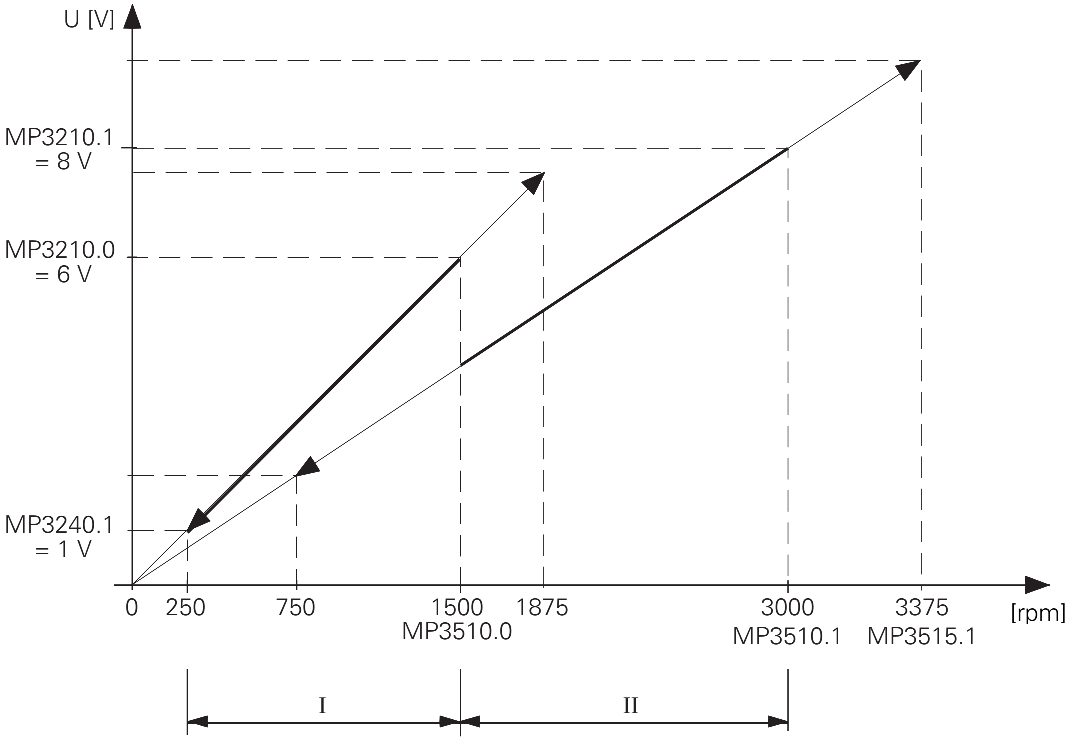
Spindle Override

With the S-override potentiometer you can change the spindle speed within certain limits. You define these limits in MP3310.x. In MP3515.x you enter a maximum attainable speed for each gear range. This speed cannot be exceeded with the spindle override.

The NC enters the % factor adjusted by the spindle override in W492 and W764. However, you can also change the % factor through the PLC by entering the desired % factor in W764. As soon as another value is entered here it is adopted by the NC.

With MP7620 bit 3 you define whether the override functions in 1% steps or according to a nonlinear curve. With a nonlinear setting, 0.01% steps are in effect below 2.5% override, and 0.75% steps are in effect above it. The range of override values in W492 and W764 lies between 1 and 150 in 1% steps for the linear characteristic, and between 0 and 15 000 for the nonlinear curve.

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: 1V (MP3240.1 = 1)



MP3310.0-1 Limits for spindle override

Input: 0 to 150 [%]

MP3310.0 Upper limit

MP3310.1 Lower limit

MP3515.0-7 Maximum spindle speed

Input: 0 to 99 999.999 [rpm]

MP7620 Feed-rate override and spindle override

Input: %xxxx

Bit 3 Feed-rate override and spindle override in 1% steps or according to a nonlinear curve

0 = 1% steps

1 = **nonlinear curve**

		Set	Reset
W492	% factor for spindle override (NC → PLC)	NC	NC
W764	% factor for spindle override (PLC → NC)	NC/PLC	NC/PLC

4.12.3 Coded Output of Spindle Speed

If in MP3010 you have selected spindle-speed code output (input value 1 or 2), an S code is entered in W258. You must send this speed code through the PLC outputs to the spindle drive.

If the speed code changes, the NC sets the S strobe marker (M4071). The NC program does not continue until you acknowledge the S code with M4091. As soon as you set M4091, the NC resets the S strobe (M4071). The NC may round off the programmed spindle speed to the next standard value. Spindle speeds of 0 to 9000 are possible.

The NC indicates the programmed spindle speed in the S code in accordance with ISO 6983 (see S code table). With MP3020 you define the permissible speed range and the speed step. The S code for the minimum speed is stored in W1008.

Example: Minimum speed = 1 rpm (S-Code 20); Maximum speed = 1000 rpm (S code 80);
Speed step = 2:

MP3020 = 20802
W1008 = 20

MP3020 Speed range for S code output
Input: 0 to 99 999
Format: xxyyz
xx = S code for minimum speed
yy = S code for maximum speed
z = speed step

		Set	Reset
W258	S code	NC	NC
M4071	Strobe signal S code	NC	NC
M4091	Acknowledgment S code	PLC	PLC
W1008	S code for minimum rpm	NC	NC

S Code Table

S function 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 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 function Code	rpm
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 82	1250
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

4.12.4 Oriented Spindle Stop

To carry out an oriented spindle stop, the axis S must be kept in the position control loop. You must therefore mount a position encoder for the spindle. In MP3010 (input values 6 to 8) you define whether the control works with spindle orientation. For special NC functions (probe cycles, rigid tapping) the spindle is oriented directly from the NC. In such cases the NC sets M4017, and you need only reset M4012 in the PLC. In most cases, an oriented spindle stop is requested through an M function (e.g. M19) or a cycle. Then you must activate it in the PLC.

Spindle orientation functions asynchronously to the NC positioning movements. Therefore you must not acknowledge the orientation until the spindle is in position (M4000). The spindle orientation will not be started by the NC until the drive is switched on (Module 9161).

You have three possible methods of orienting the spindle in the PLC:

- With Module 9171
- With marker M4130
- With a proximity switch and marker M4011

Process of spindle orientation with marker M4130 or Module 9171

The spindle speed is reduced under control along the ramp from MP3412.1 to the speed for spindle orientation (MP3520.1). Once this speed has been reached the position control loop is closed and the spindle is servo-controlled along the ramp from MP3412.1 to the nominal position. As long as the servo-controlled spindle is moving, M4017 is set.

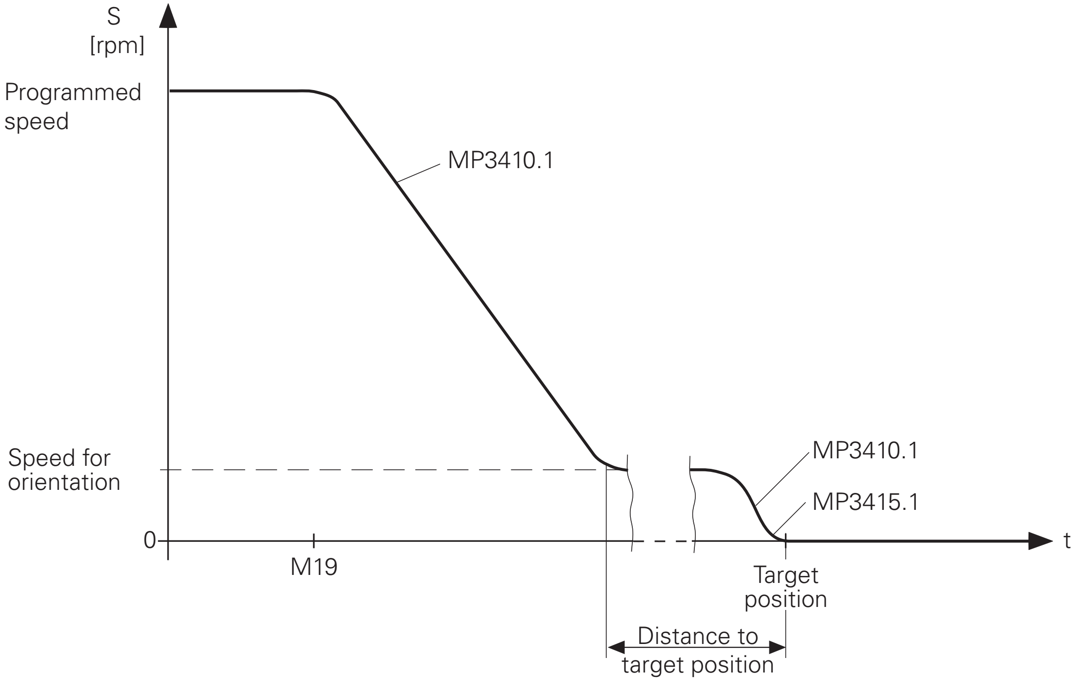
To adjust the gear ranges, enter a k_v factor for each of the axes in MP3440.x. With MP3415.1 you define the transient response of the spindle for spindle orientation. With MP3420 you define the positioning window. If the spindle is inside the positioning window after the oriented stop, M4000 is set.

If you do not want the spindle to remain in the position control loop after it has reached the nominal position, then you must set marker M4012. The spindle is not free until this marker is set. If M4012 is always set, the control loop is opened immediately after every oriented spindle stop, provided that the positioning window has been reached. With MP3430 you can easily compensate any misalignment of the rotary encoder resulting from imprecise mounting. The offset between the nominal and actual position of the reference mark entered here is then accounted for during spindle orientation.

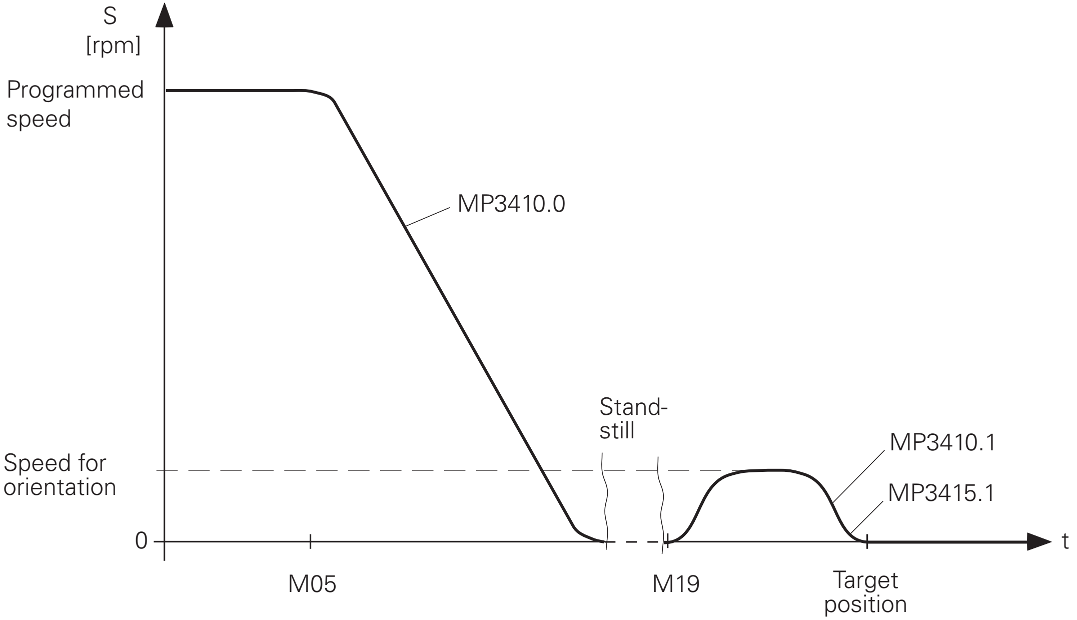
When the spindle is first switched on, the NC immediately evaluates the reference mark (even if the position control loop is not closed). M4018 is set until the reference mark is evaluated. For special applications you can evaluate the reference mark again by setting M4015. The NC resets M4015 when the reference mark has been evaluated.

With MP7291 you activate a display of the spindle position. The spindle position is displayed only if neither M03 nor M04 is active. It is displayed as a value below 360°.

Orientation of a moving spindle:



Orientation of a stationary spindle:



MP3412.1 Multiplier for MP3411.x with oriented stop
Input: 0 to 1.999

MP3415.1 Transient response of the spindle for oriented stop
Input: 0 to 1000 [ms]

MP3420 Positioning window for the spindle
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
Input: 0.1 to 10 $\left(\frac{1000°/\text{min}}{°}\right)$

MP3520.1 Spindle speed for oriented stop
Input: 0 to 99 999.999 [rpm]

		Set	Reset
M4000	Spindle in position	NC	NC
M4012	Open the spindle control loop	PLC	PLC
M4015	Reevaluation of the spindle reference mark	PLC	NC
M4017	Servo-controlled spindle in motion	NC	NC
M4018	Reference mark for spindle not yet traversed	NC	NC

Spindle orientation with module 9171

Module 9171 starts a spindle orientation for which you enter the speed, nominal position, and direction of rotation. M4130 is set as long as spindle is being positioned. The module functions only in the cyclic PLC program. If you call the module while the spindle is rotating, the transferred direction is disregarded and the spindle is oriented in the direction of rotation.

If you enter the values 2 to 4 as direction of rotation, the spindle is oriented to the angle last defined through CYCL DEF 13. The transferred angle is added to the value from CYCL DEF 13.

Call only from sequential PLC program:

```
PS   B/W/D/K    <Angle> [1/10 000 °]
                or additional preset with value from CYCL DEF 13
PS   B/W/D/K    <Speed> [1/1 000 rpm]
                0 =    MP3520.1 is used
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 with angle from CYCL DEF 13
                3:     Same as 0 but with angle from CYCL DEF 13
                4:     Same as +1 but with angle from CYCL DEF 13
CM   9171
```

Spindle orientation with M4130

With M4130 you can start an oriented spindle stop. The nominal position is taken over from D592 and the speed from MP3520.1. The nominal position is referenced to the reference point. The nominal position can be transferred to the PLC with, for example, MP4210.x.
The nominal position can also be taken from taken from the Oriented Spindle Stop cycle (CYCL DEF 13). Then you must set the MSB of D592 = 1 and the other bits to 0. When Cycle 13 is run, M4016 is set.

If a stationary spindle is oriented, it takes the shortest path to the nominal position if the distance from the actual starting position to the nominal position is not greater than the positioning window (MP3420). If the distance is greater than the positioning window it will be 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 stop 0 = M03 ; 1 = M04	PLC	PLC
M4016	Cycle 13 is being run	NC	NC
M4130	Activation of the spindle orientation, or spindle orientation has been started with 9171	NC,PLC	NC

Spindle orientation through proximity switch with M4011

The spindle can also be oriented by means of a proximity switch. To do this you must set M4011. After you have set M4011 the spindle is moved in the direction of rotation from M4013 at the speed from MP3520.0. As soon as you reset M4011 the spindle is stopped. The current position value is displayed in the status window.

MP3520.0 Spindle speed activated by marker M4011
Input: 0 to 99 999.999 [rpm]

		Set	Reset
M4011	Activate spindle speed from MP3520.0 and direction of rotation from M4013	PLC	PLC

Offset adjustment (only analog spindles)

After the spindle has been brought to an oriented stop, any remaining offset is automatically adjusted. In order to give the spindle enough time to settle to a stop the offset adjustment is delayed until the spindle has been in position for at least two seconds. After this time is over the offset in compensated by 0.152 mV cyclically in intervals of one second.

4.12.5 Tapping with Floating Tap Holder and Nominal Speed Output

During tapping with floating tap holder the spindle position control loop is open. During the tapping cycle M4030 is set. After the spindle is switched on with M03 and acknowledged with M4092 the nominal speed must be attained before feed motion begins. After switch-on the spindle follows the ramp from MP3411.x; after switch-off it follows the ramp from MP3412.2. With MP3415.2 you can define the transient response of the spindle.

You must acknowledge the M functions immediately. An NC stop cannot be carried out until a preceding M function has been acknowledged. If the feed rate ramp is flatter than the spindle ramp, the spindle follows the feed rate ramp. With MP7120.2 you can delay switch-off.

Example:
From speed s = 1000 [rpm] = 1.8 [V] and MP3410.2 = 0.05 [V/ms] follows:

$$\frac{1.8\ [\mathrm{V}]}{0.05\ [\mathrm{V/ms}]} = 36\ \mathrm{ms}$$

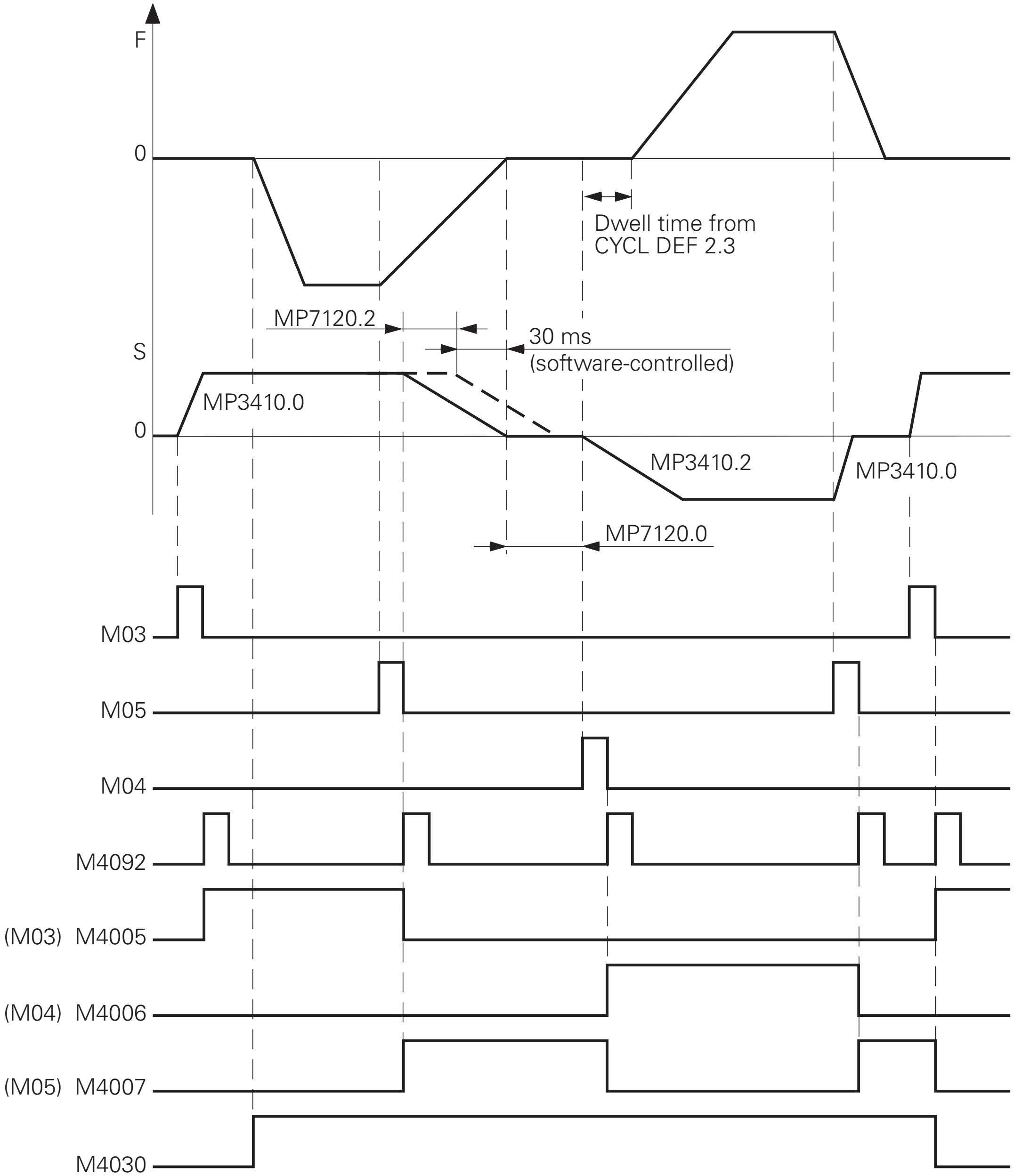
The spindle is decelerated 36 ms before reaching the total hole depth. If the spindle deceleration is delayed by the slow-down time (MP7120.2), then it is possible only to delay up to 30 ms before reaching the total hole depth. Any greater delay will be ignored.

With MP7120.0 you can delay the restarting of the spindle with M04. The ramp follows MP3412.2 again. You can delay the restarting of the feed by programming a longer dwell time in the cycle. The dwell times enable you to make an optimal adjustment of the floating tap holder.

The switch-off of the spindle is again performed by the NC with M05. The switch-off ramp follows MP3412.0. After this, the spindle is switched on again with M03.

The feed-rate override can only be effective within limits when tapping, otherwise the floating tap holder may be damaged. You enter the limits in MP7110.x.

The following diagram shows the temporal sequence of operations of the cycle.



MP3412.2 Multiplier for MP3411.x with tapping
Input: 0 to 1.999

MP3415.2 Transient response of the spindle for tapping
Input: 0 to 1 000 [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 [%]

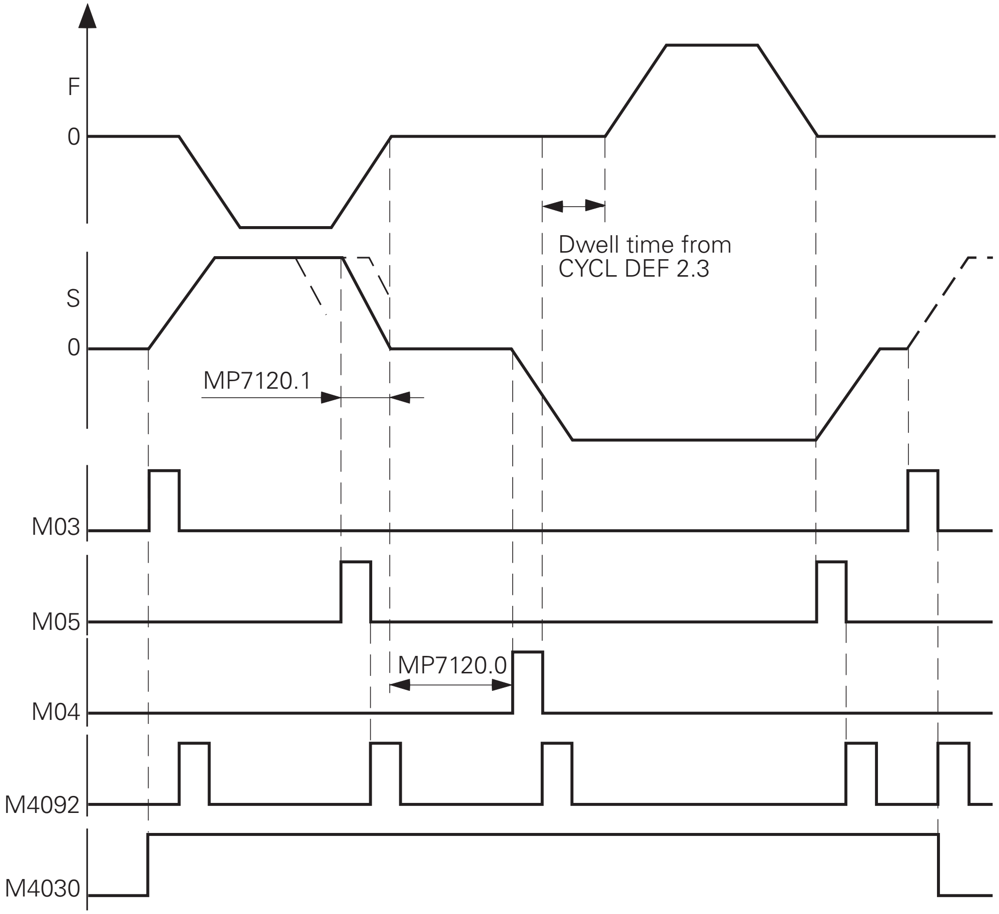
MP7120.0 Dwell time for reversal of spindle rotation direction
Input: 0 to 65.535 [s]

MP7120.2 Spindle slow-down time after reaching the total hole depth
Input: 0 to 65.535 [s]

		Set	Reset
M4030	Cycle 2 or Cycle 17 active	NC	NC

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

The following diagram shows the temporal sequence of operations of the cycle:



Since, when using the coded spindle speed output, the spindle- and feed ramps cannot be synchronized by the NC, you must enter the advanced switching time for the spindle in MP7120.1. MP7120.0 (dwell time for reversing the direction of rotation) and the programmable dwell time have the same effect at the nominal-speed-value output.

MP7120.1 Advanced switching time of the spindle for the tapping with coded output of the spindle speed

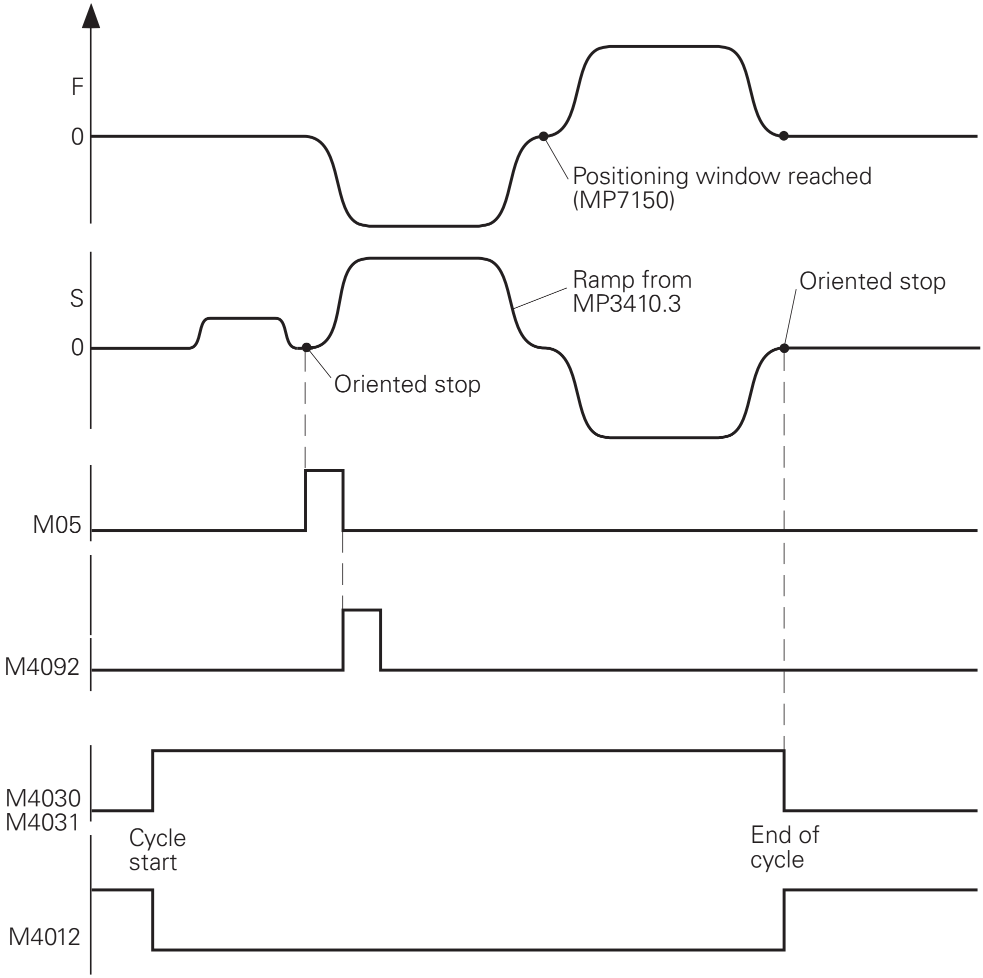
Input: 0 to 65.535 [s]

4.12.7 Rigid Tapping

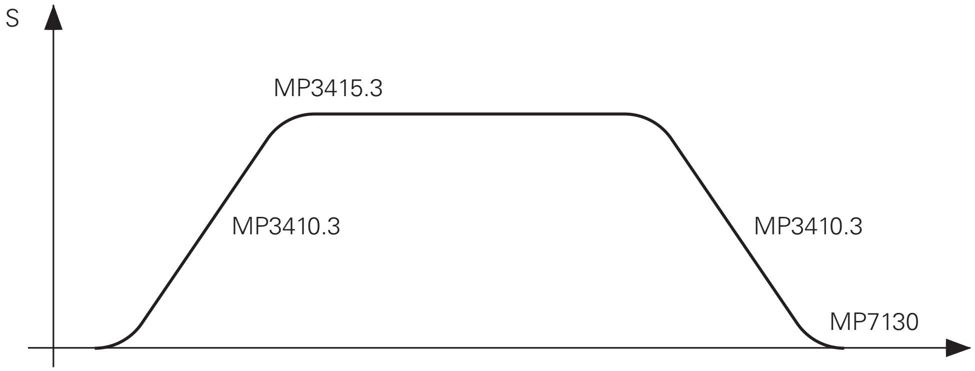
Cycle 17

During rigid tapping the spindle position control loop is open. The machine tool operator defines rigid tapping with Cycle 17 in the NC part program. While Cycle 17 is being run the TNC automatically switches to operation with velocity feedforward. You define the dynamic behavior of the spindle and the tool axis in machine parameters. The tool axis tracks the actual position of the spindle during tapping. Before tapping begins, the axes, for example Z and S, are synchronized by means of an oriented spindle stop. This means the every Z position is assigned to a corresponding spindle position. The NC carries out the oriented spindle stop. M4017 is set by the NC and in the PLC the spindle position control loop must be closed (M4012).

Synchronization makes it possible to cut the same thread more than once. The permanently assigned spindle position depends on the thread pitch entered in the cycle. To save machining time you can deselect this function with MP7160. Then, however, it is no longer possible to cut the thread more than once. M4031 and M4030 are set while the cycle is running.



The positioning window from MP7150 is effective for the tool axis while Cycle 17 is being run. The input value must be less than or equal to MP1030.x. You define the spindle acceleration and deceleration process in MP3412.3, MP3415.3 and MP7130.



MP3412.3 Multiplier for MP3411.x with rigid tapping
Input: 0 to 1.999

MP3415.3 Transient response of the spindle for rigid tapping
Input: 0 to 1000 [ms]

MP7130 Approach behavior of the spindle for rigid tapping
Input: 0.001 to 10 [°/min]

MP7150 Positioning window of the tool axis for rigid tapping
Input: 0.0001 to 2 [mm]

MP7160 Spindle orientation with Cycle 17
Input: 0= spindle orientation before execution of Cycle 17
1= no spindle orientation before execution of Cycle 17

		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 current position. The target position is the total hole depth. Approaching and departing movements must be programmed separately. During Cycle 18 marker M4031 is set. M4012 must be reset so that the cycle is run. MP3410.3, MP3415.3, MP7130 and MP7150 have the same function as for Cycle 17.

4.12.8 Speed Controller (Only TNC 426 PB, TNC 430 PA)

A digital speed controller for the spindle is integrated in the TNC 426 PB and TNC 430 PA. With MP2501.x you set the proportional factor and with MP2511.x the integral factor for the speed controller of the spindle.

With Module 9164 you can read the actual speed of the motor (see section "Control Loop"). As with the axes, you can also define a differential factor, a PT2 element and a frequency filter for the spindle speed controller (see sections "Control Loop" and "Commissioning").

MP2501.0-1 Proportional factor of the speed controller of the spindle
Input: 0 to 1000.000 [As]

MP2511.0-1 Integral factor of the speed controller of the spindle
Input: 0 to 100 000 [A]

MP2521.0-1 Differential factor of the speed controller of the spindle
Input: 0 to 1.0000 [As²]

MP2531.0-1 PT2 element of the speed controller of the spindle
Input: 0 to 1.0000 [s]

MP2541 Damping for frequency filter (spindle)
Input: 0.0 to 18.0 [dB]

MP2551 Nominal frequency for frequency filter (spindle)
Input: 0.0 to 999.9 [Hz]

4.12.9 Current Controller (Only TNC 426 PB, TNC 430 PA)

Depending on the hardware, the TNC may have different types of current controller. We differentiate between logic units with or without spindle DSP:

TNC 426 PB max. spindle speed 9 000 rpm	without spindle DSP
TNC 426 PB max. spindle speed 15 000 rpm	with spindle DSP
TNC 430 PA max. spindle speed 15 000 rpm	with spindle DSP

Current controller without spindle DSP

With MP2401 you define the current gain for the spindle when stationary. Some asynchronous motors tend to run unsteadily at high speeds. To correct this, enter a higher gain in MP2403 than in MP2401. MP2403 defines the current gain reached at maximum speed. The gain is interpolated linearly between standstill and maximum speed. If you enter the value zero in MP2403, the current gain from MP2401 will be in effect for the entire speed range.

MP2401 Gain for current controllers of the spindle at standstill
Input: 0.00 to 9 999.99 [V/A]
0 = disable the controller

MP2403 Gain for current controllers of the spindle at maximum rotational speed
Input: 0.00 to 9 999.99 [V/A]
0 = value from MP2401

Current controller with spindle DSP

With MP2421 you define the proportional factor, with MP2431 the integral factor for the current controller of the spindle (see section "Commissioning").

MP2421.0-1 Proportional factor for the current controller of the spindle
Input: 0.00 to 9 999.99 [V/A]

MP2431.0-1 Integral factor for the current controller of the spindle
Input: 0.00 to 9 999.99 [V/As]

4.12.10 Wye Connection / Delta Connection (Only with Spindle DSP)

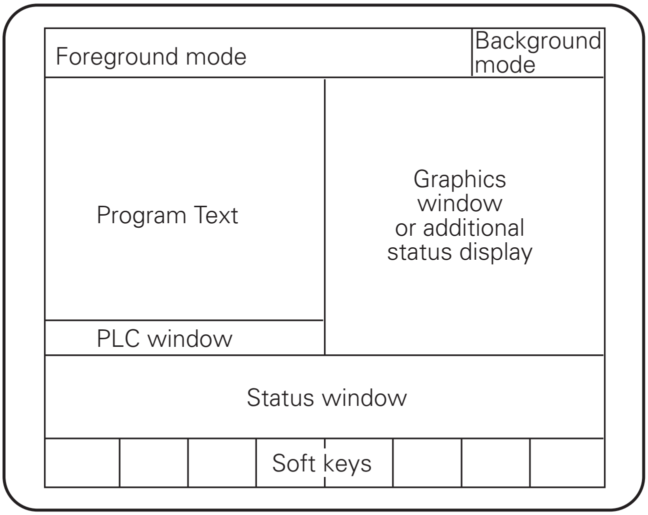
You can run the spindle motor either in a wye (Y) or delta (Δ) connection. A delta connection enables you to run the motor at higher speeds than the wye connection, but at low speeds the motor develops less power than with the wye connection. For this reason, you will switch between a wye and delta connection depending on the desired speed. The technical data of the motor are stored internally both for a wye and a delta connection. Also, there are two sets of machine parameters available for the current and speed controllers: index 0 is for wye connection and index 1 for delta connection.

With Module 9163 you activate the change between a wye and delta connection. As soon as Module 9163 is called, the NC disables the spindle drive and activates the motor data and machine parameters of the selected circuit (wye or delta). You can check this with Module 9162. After an external relay has changed the motor circuit, you enable the spindle with Module 9161.

```
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
```

4.13 Display and Operation

You can modify the display and operation modes of the TNC by editing the machine parameters. The display is divided into separate windows to provide you immediately with the information most important to you. With soft keys you can adjust the desired layout of the screen (see the User's Manual).



4.13.1 Machine Datum

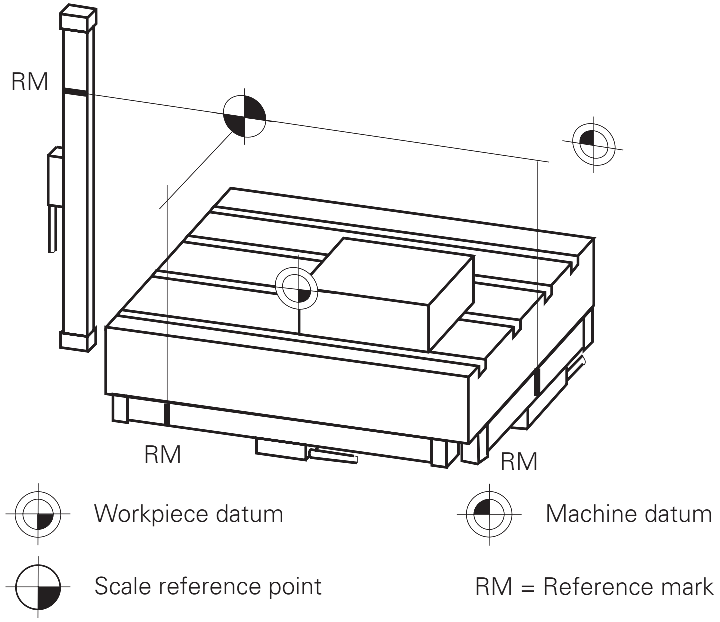
You can set a workpiece datum in the "Manual" and "Electronic handwheel" operating modes. The position values in NC positioning blocks are normally referenced to this datum. If you want the values to be referenced to machine datum instead of the workpiece datum, then the machine operator must program M91. Enter the distance between the machine datum and the scale datum in MP960.x. All REF-based displays and positions are then referenced to the machine datum.

If you wish the values in the NC positioning blocks to always be referenced to the machine datum, with MP7295 you can disable the datum setting function for specific axes. With MP950.x you can define an additional machine-fixed datum. NC blocks with M92 are then referenced to this position. In MP950.x you enter the distance of this machine-referenced position from the machine datum.

M91 and M92 are active only in the block in which they are programmed.

With MP7296 you define whether the datum can be set only by the DATUM SET soft key or also with the axis keys.
If you enter the code number 555 343 you can change the datum via the NC program using FN25 (OEM cycles).

Input format:	FN25: PRESET <Axis> /Q<Number> /<Value>
<Axis> =	Axis for which the datum is to be set
Q<Number> =	Number of the Q parameter where the reference position for the datum is stored
<Value> =	Desired datum value



MP950.0-8 Datum point for positioning blocks with M92
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
Values referenced to the machine datum

MP960.0-8 Machine datum
Input: –99 999.9999 to +99 999.9999 [mm] or [°]
Values referenced to the scale reference point

MP7295 Disable the datum-setting function
Input: %xxxxxxxxx

Bit	0 to 8	0 = not disabled
Axis	1 to 9	1 = disabled

MP7296 Datum setting with axis keys
Input: 0 = datum point can be set with axis keys and soft key
1 = datum point can be set only with soft key

4.13.2 Color Adjustment

The colors in the display can be configured by machine parameter. You may wish, for example, to change the colors to suit your company's corporate image or the design of the machine.

You cannot alter the following color settings by machine parameter:

- HEIDENHAIN company logo after switching on the machine (standard color)
- Blinking error messages (red)
- Error message for invalid machine parameters (red)
- Plan view in the graphics display (blue)
- Cursor (always inverse)

You mix the desired color by combining the basic colors red, green and blue. You can enter each of these elementary colors in 64 different intensity levels. The input values for color adjustment are byte-oriented. We recommend hexadecimal input.

Color	**Red**		**Green**		**Blue**	
Adjustment	Coarse	Fine	Coarse	Fine	Coarse	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

Since it is possible to make mistakes when setting the colors (red error messages on red background, for example) HEIDENHAIN supplies the controls with a standard color adjustment. This is the setting used by HEIDENHAIN. It is used by the control system when you edit the MP list.

The standard color adjustment is shown in the following list.

MP7350	Window frame	$030200C
MP7351	Error messages	$03F3F0F
MP7352	"Machine" operating mode display	
MP7352.0	Background	$0000000
MP7352.1	Text for operating mode	$0342008
MP7352.2	Dialog	$03F3828
MP7353	"Programming" operating mode display	
MP7353.0	Background	$0000000
MP7353.1	Text for operating mode	$0342008
MP7353.2	Dialog	$03F3828
MP7354	"Machine" program-text display	
MP7354.0	Background	$0080400
MP7354.1	General program text	$038240C
MP7354.2	Current block	$038341C
MP7354.3	Background not current window	$00C0800

MP7355	"Programming" program-text display	
MP7355.0	Background	$0080400
MP7355.1	General program text	$038240C
MP7355.2	Current block	$038341C
MP7355.3	Background not current window	$00C0800
MP7356	Status and PLC windows	
MP7356.0	Background	$00C0800
MP7356.1	Axis positions in the status display	$03F2C18
MP7356.2	Status display except for axis positions	$03F280C
MP7357	"Machine" soft-key display	
MP7357.0	Background	$0000000
MP7357.1	Symbols	$03F3828
MP7358	"Programming" soft-key display	
MP7358.0	Background	$0000000
MP7358.1	Symbols	$03F3828
MP7360	Graphics: 3-D view	
MP7360.0	Background	$0000000
MP7360.1	Top surface	$0203038
MP7360.2	Front face	$00C1820
MP7360.3	Text displays in the graphic window	$03F3F3F
MP7360.4	Side surfaces	$0102028
MP7361	Graphics: view in three planes	
MP7361.0	Background	$0000000
MP7361.1	Top view	$0203038
MP7361.2	Front and side view	$0203038
MP7361.3	Axis cross and text in the graphic display	$03F3F3F
MP7361.4	Cursor	$03F0000
MP7362	Additional status display in the graphic window and pocket calculator	
MP7362.0	Background graphic window and pocket calculator	$0080400
MP7362.1	Background status display and keys of the pocket calculator	$00C0800
MP7362.2	Status symbols and symbols of the pocket calculator (c in cos)	$038240C
MP7362.3	Status values and texts of the pocket calculator (os in cos)	$03F2C18
MP7363	Programming graphics	
MP7363.0	Background	$0000000
MP7363.1	Resolved contour	$03F3F3F
MP7363.2	Subprograms and frame for zooming	$0003F00
MP7363.3	Alternative solutions	$0003F00
MP7363.4	Non-resolved contours	$03F0000

MP7364	Colors of the help illustrations for cycles	
MP7364.0 to MP7364.6	Colors 1 to 7 of the graphic program used	$0000000
MP7364.7	Line colors (color 8 of the graphic program used)	$038240C
MP7364.8	Color for highlighted graphic elements if defined in the help illustration	$038341C
MP7364.9	Background	$0000000
MP7365	Oscilloscope	
MP7365.0	Background	$0000000
MP7365.1	Channel 1	$0203038
MP7365.2	Channel 2	$0003F00
MP7365.3	Channel 3	$03F3F00
MP7365.4	Channel 4	$03F003F
MP7365.5	Selected channel ($00000FF = color from MP7365.1-4)	$03F0000
MP7365.6	Grid	$030200C
MP7365.7	Cursor and Text	$03F3F3F
MP7366	Superimposed window (HELP key, pop-up menus, etc.)	
MP7366.0	Background	$0333333
MP7366.1	Text or foreground	$0281408
MP7366.2	Current line	$0140A04
MP7366.3	Headline	$02F2818
MP7366.4	Scroll bar field	$0100C08
MP7366.5	Scroll bar	$02F2818
MP7366.6-14	*Reserved*	Gray shade

4.13.3 Graphic Display

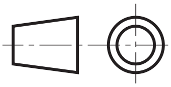
In the graphics window you can view the following graphics:

- Test graphics
- Parallel graphics
- Programming graphics
- Help illustration

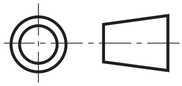
For the test graphics and parallel graphics can you choose one of three display modes:

- View in three planes
- Plan view
- 3-D view

View in three planes
The projection can be set either to the preferred German projection (1st angle) or preferred American (3rd angle).

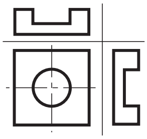


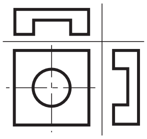
German preferrred



U.S. preferred

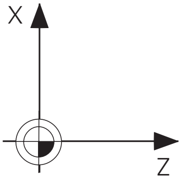
Example
:



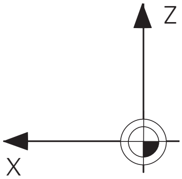


Position of the cursor
In the 3-plane display mode you can show the cursor position. You must activate this function with machine parameter MP7310, bit 3.

Rotating the coordinate system
You can rotate the coordinate system for graphic display by 90°. This can be useful, for example, when the Y axis is defined as the tool axis.



No rotation



90° rotation

Graphic display for datum shift
You can program several BLK forms in one NC program. With MP7310 bit 2 you define whether the definition of a subsequent blank form is moved after a datum shift with Cycle 7.

MP7310	Graphic display mode	
Input:	%xxxx	
Bit 0	Display mode in three planes	0 = projection preferred in Germany 1 = projection preferred in America
Bit 1	Rotating the coordinate system in the working plane by +90 °	0 = no rotation 1 = coordinate system rotated by +90 °
Bit 2	BLK form after a datum shift	0 = BLK form is not shifted 1 = BLK form is shifted
Bit 3	Display of cursor position during view in three planes	0 = no display of cursor position 1 = cursor position is displayed

4.13.4 Status Display

The status of the control (axis positions, tools, feed, M-functions, etc.) is displayed on the control screen in the status window. A soft key can be used to activate an additional status-display in the graphics window instead of the graphics.

Position display
With MP7290.x you select the display step for the individual axes. The position loop resolution is not affected by this machine parameter. With MP7285 you define whether the tool length should be calculated into the position display (i.e. tool length offset). If it is, the displayed position value then refers to the tool point.

MP7285 Offset of tool length in the position display in the tool axis
Input: 0 = tool length is not offset
1 = tool length is offset

MP7290.0-8 Position display step
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 °

Positions display for rotary axes and PLC auxiliary axes
With MP810 you define the position display for rotary axes and PLC auxiliary axes. You set the modulo value for the counting mode (i.e. the value after which the axis display returns to zero). At the same time you define whether the software limit switches of the traverse range limits should be in effect.

With MP7682 bit 2 you can set whether rotary axes with modulo display should always be positioned by the shorter path or without crossing over zero. If bit 2 is set, you don't need to program M126.

MP810.0-8 Display mode for rotary axes and PLC auxiliary axes
Input: 0.0000 to 99 999.9999[°]
0 = display ±99 999.9999; software limit switch active
≠ 0 = modulo value for display; software limit switch inactive

MP7682 Machine parameter with multiple function
Input: %xxx
Bit 2 Traverse path of rotary axes with modulo display
0 = positioning without crossing over zero
1 = shorter path positioning

Reading the axis coordinates

With Modules 9040 and 9041 you can read the axis coordinates. The values are stored in five doublewords, beginning with the given target address. Regardless of whether individual axes have been excluded through MP10, the coordinate values are always read for all axes (TNC 426: 5 doublewords; TNC 430: 9 doublewords). The values for excluded axes remain undefined. The coordinate value of an axis remains undefined until the reference point has been traversed.

Module 9040: Reading the axis coordinates (Format 1/1000 mm)
Module 9041: Reading the axis coordinates (Format·1/10 000 mm)
Call:

```
PS   K/B/W/D    <Target address Dxxxx>
PS   K/B/W/D    <Coordinate type>
                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)
CM   9040 or CM 9041
```

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 define the free-rotation function in the PLC program either in words (only axes 4 and 5) or with Module 9223 (axes 1 to 9). The max. feed rate is 300 000 °/min. The feed rate is not displayed in the status window. You can vary the feed rate continuously through the override percentage (W754). To do this, copy W494 (current feed-rate override) to W754.

Free rotation defined by means of words:

- W566 — feed rate in axis 4 for free rotation
- W568 — feed rate in axis 5 for free rotation
- W754 — feed-rate override percentage for free rotation
- B518 — set the free-rotation function
- B519 — traverse direction for free rotation
- M4133 — starting and stopping the free-rotation function

You start and stop the free-rotation function with M4133. If you set M4133, the NC takes the information from B518 and B519, and resets M4133.

Free rotation defined with Module 9223:

Call this module only when no program has been started or an M, S, T, Q strobe is about to start. Calling this module sets M4133 (both when starting and stopping).

```
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:    Feed rate not allowed
                                    3:    The axis did not cross over REF
                                    4:    No M,S,T,Q strobe after program was started
```

			Set	Reset
M4133	Starting and stopping the free rotation function		PLC	NC
B518	Definition of the free-rotation function		PLC	PLC
	0 =	Cancel the function		
	8 =	Free rotation for axis 4		
	16 =	Free rotation for axis 5		
B519	Traverse direction for free rotation		PLC	PLC
	0:	Axis 4 and axis 5 = +		
	8:	Axis 4 = –, axis 5 = +		
	16:	Axis 4 = +, axis 5 = –		
	24:	Axis 4 and axis 5 = –		
W754	Percentage of feed-rate override for free rotation		PLC	PLC
W566 to W568		Feed rate for free rotation	PLC	PLC
W566	to	W568		
Axis 4	to	Axis 5		

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 NC enters the percentage set with the feed-rate override into W494 and W766. You can change the percentage through the PLC by entering the desired percentage in W766. The NC adopts the new value as soon as it is entered.

With MP7620 bit 3 you define whether the override becomes effective in 1%-steps or according to a nonlinear curve. The nonlinear setting uses a resolution of 0.01% steps in the range below 2.5% and a resolution of 0.75% steps in the range above it. The range of the override values in W494 and W766 lie between 1 and 150 for the 1% steps. For the nonlinear curve they lie between 0 and 15 000.

In the manual operating modes the axis feed rate is shown, not the contouring feed rate. You have the choice between two types of displays:

- The axis-specific feed rate (MP1020.x or value programmed via soft key F) is displayed only after pressing an axis-direction key. If two axis-direction keys are pushed simultaneously, no feed is displayed.
- Even when none of the axis-direction keys is operated one feed rate will be displayed, which can also be adjusted by the feed potentiometer. The smallest feed rate from MP1020.x (or value programmed via soft key F) is effective for all axes. PLC axes are not taken into account in the selection of the smallest feed rate. The axis feed rate will also be shown if several axis-direction keys are pressed simultaneously.

MP7270 Feed-rate display in the "Manual operation" and "Electronic handwheel" modes
Input: 0 = display of the axis feed rate only when an axis-direction key is pressed (axis-specific feed from MP1020.x)
1 = display of the axis feed-rate even before operating an axis-direction key (smallest value from MP1020.x for all axes)

MP7620 Feed-rate override and spindle override
Input: %xxxx
Bit 3 Feed rate and spindle override in 1% steps or according to nonlinear curve
0 = 1% steps
1 = nonlinear curve

		Set	Reset
W494	Percentage for feed rate override (NC → PLC)	NC	NC
W766	Percentage for feed rate override (PLC → 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 therefore depends on the distance of the tool center from the center of the rotary axis. When M function M116 is activated, the feed rate is interpreted in mm per minute, i.e. the contour feed rate is now independent of the distance of the tool center from the center of the rotary axis.

M116 is automatically canceled by PGM END. M116 is active only if you define the center of rotation of a rotary axis in machine parameters MP7510 and following.

Display of the M functions

The miscellaneous functions for control of the spindle (M03, M04, M05) and the coolant (M07, M08, M09) are displayed in the status window. You control the display of these M-functions through the PLC. M4005 and M4006 change the polarity of the analog voltage for the spindle. With M4008 you disable the speed output for the spindle. The programmed spindle speed is, however, still displayed. At the same time, M03, M04 or M05 is highlighted on the screen. 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, 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

Control in operation

If the control is at work, i.e. if it is positioning an axis or executing an M function, the status window displays a star-like symbol known as the control-in-operation symbol. If a running NC program is interrupted with the machine stop button, this symbol starts to blink. With M4175 and M4176 these conditions are reported to the PLC. M4175 and M4176 are effective in the "Positioning with MDI," "Program run, single block" and "Program run, full sequence" modes of operation.

With Module 9089 you can use the PLC to display or erase the control-in-operation symbol. You can activate the control-in-operation symbol only if it has not already been shown or set to blinking by the NC. The NC has priority over the PLC for setting, resetting, or blinking the control-in-operation symbol.

If the NC displays the control-in-operation symbol, you cannot erase it. If the PLC displays the control-in-operation symbol, this has no effect on marker M4176.

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 erased/displayed.
                 1:  Incorrect command code.
                 2:  Control-in-operation symbol is already displayed by the NC .
                 3   Control-in-operation symbol flashes.
                 4:  Because it was displayed by the NC, control-in-operation symbol could not be
                     erased.
```

		Set	Reset
M4175	Program interruption (control-in-operation symbol flashes)	NC	NC

M4176 Control is in operation (control-in-operation symbol is on or is blinking NC NC

Clearing the status display

With MP7300 you define when the status display, the tool data and the contents of the Q parameters will be erased. At this time all the programmed values, such as scaling factor, datum shift, and feed rate will be reset. The Q parameters and tool data are set to zero.

MP7300 Clear the status display and the 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 with M02, M30, END PGM and 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 with M02, M30, END PGM, and when a program is selected
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 with M02, M30, END PGM
6 = Erase the status display when a program is selected
7 = Erase the status display when a program is selected and with M02, M30, END PGM

Input	**Erase with PGM MGT**	**Erase with M02, M30, END PGM**	**Status display**	**Tool data**	**Q parameters**
0	x	–	x	x	x
1	x	x	x	x	x
2	x	–	x	x	–
3	x	x	x	x	–
4	x	–	x		x
5	x	x	x	–	x
6	x	–	x	–	–
7	x	x	x	–	–

Interrogating the status display through the PLC

In addition to the markers and words described above, with Module 9035 you can interrogate the status display.

Module 9035: Reading the status information
You transfer a number designating the desired information.

Transferred number:		Requested values:
0	Main mode Editor	0: "Programming and editing" 1: "Test run"
1	Main mode Machine	0: "Traverse the reference marks" 1: "Manual operation" 2: "Electronic handwheel" 3: "Positioning the Manual Data Input" 4: "Program run, single block" 5: "Program run, full sequence"
2	Overlaid mode Editor	0: None (main mode is active) 1: MOD active 2: Directory/Ext screen active 3: MP editor active 4: PLC editor active
3	Overlaid mode Machine	0: None (main mode is active) 1: MOD active 2: Directory/Ext screen active 3: Tool table selected 4: Pocket table selected[1] 5: PLC table selected[1]
4	Displayed screen window	Bit-coded Bit 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/graphics window active Bit 6/Bit 7: *Reserved* Bit 8 to Bit 15: Machine screen Bit 8 =1: Machine screen is displayed Bit 9 =1: Mode window active Bit 10=1: Block display/program select/setup window active Bit 11=1: Position display active Bit 12=1: PLC status window active Bit 13=1: Status/Graphics window active Bit 14/Bit 15: *Reserved*

[1] As of NC software 280 472 01

5	Selected file in "Programming and editing" and "Test run" modes	0: No file 1: .H (plain language 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 (compensation value table)
6	Selected file in "Program run, full sequence" and "Program run, single block"	0: No file 1: .H (conversational part PGM) 2: .I (ISO part PGM)
7	Selected axis for actual position capture in "Programming and editing" mode	0 to 8: Axes 1 to 9
8	Selected axis for actual position capture in "Positioning with MDI" mode	0 to 8: Axes 1 to 9
9	Handwheel axis	–1: None or several 0 to 8: axes 1 to 9
10	Handwheel axis bit-coded	Bit 0 to bit 8: Axes 1 to 9
	Handwheel interpolation factor	
11	X-key	0 to 10
12	Y-key	0 to 10
13	Z-key	0 to 10
14	IV-key	0 to 10
15	V-key	0 to 10
16	Input format for the $MDI file	0: .H file type (plain language part PGM) 1: .I file type (ISO part PGM)
17	Display format	0: MM 1: INCH
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 velocity	0: Low 1: Medium 2: High

21	Model of control	0: TNC 310 1: TNC 370 2: TNC 410 3: TNC 426 CA/PA 4: TNC 426 CB/PB or TNC 430 CA/PA
	Handwheel interpolation factor	
31	Axis 1	0 to 10
32	Axis 2	0 to 10
33	Axis 3	0 to 10
34	Axis 4	0 to 10
35	Axis 5	0 to 10
36	Axis 6	0 to 10
37	Axis 7	0 to 10
38	Axis 8	0 to 10
39	Axis 9	0 to 10
1000[1]	PLC table editor (only in submit or SPAWN)	≥0: current line in PLC table editor –1: PLC table editor not active
1001[1]	Pallet table (only in submit or SPAWN)	≥0: current line in pallet table –1: pallet table not active

Call:

```
PS   B/W/D/K    <Number of the desired status information>
CM   9035
PL   B/W/D      <Status information>
```

W1022	1:	Status information not valid
	20:	Call was not made in submit or SPAWN

[1] as of NC software 280 472 01

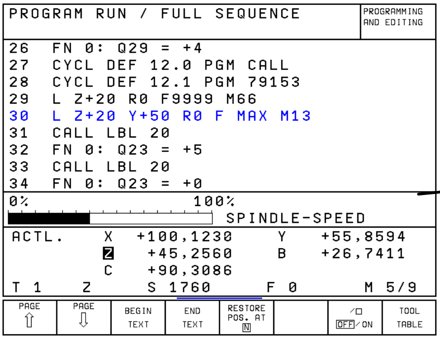
4.13.5 PLC Display

Through the PLC you can display the following information on the Visual Display Unit:

- Display of a moving-bar graphic or text in the small PLC window
- Display of graphics or text in the large PLC window (instead of the graphics window)
- Display of PLC soft keys
- Display of text in the HELP file (via MOD call)
- Superimpose PLC window

4.13.6 Small PLC Window

In the "Manual," "Electronic handwheel," "Positioning with MDI," "Program run, single block" and "Program run, full sequence" modes the small PLC window is always displayed.



```
PROGRAM RUN / FULL SEQUENCE                PROGRAMMING
                                           AND EDITING
26   FN 0: Q29 = +4
27   CYCL DEF 12.0 PGM CALL
28   CYCL DEF 12.1 PGM 79153
29   L Z+20 R0 F9999 M66
30   L Z+20 Y+50 R0 F MAX M13
31   CALL LBL 20
32   FN 0: Q23 = +5
33   CALL LBL 20
34   FN 0: Q23 = +0
0%                              100%
                                  SPINDLE-SPEED
ACTL.      X   +100,1230     Y    +55,8594
           Z    +45,2560     B    +26,7411
           C    +90,3086
T 1     Z      S 1760       F 0          M 5/9
```

Small PLC window

You can show any ASCII text in two lines, each with 38 characters. In the left half of the line you can also show a moving-bar diagram. Text and moving-bar diagram can be mixed. In the PLC program you can use the following modules to determine what is shown in this window:

- 9080: Delete the small PLC window
- 9081: Interrogate the status of the small PLC window
- 9082: Show string in the small PLC window
- 9083: Show moving-bar graphic in the small PLC window

Module 9080: Erase the small PLC window
You must use a submit job to call the module. While the module is being executed you must not abort the submit job with a CAN command. The contents of the small PLC window is erased. You have defined the background color of the window in MP7320.2 and MP7356.0. The module is also effective if the presently selected screen does not contain the small PLC window (e.g. large graphic display) or when the screen with the PLC window is in the background.

Call:
CM 9080

Module 9081: Interrogate status of the small PLC window

With Module 9081 you inquire whether a small PLC windows is presently being displayed.

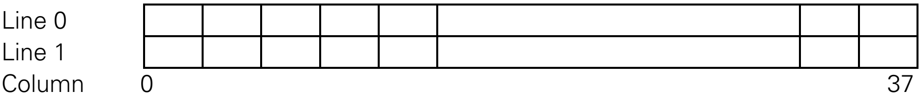
Call:

```
CM   9081
PL   B/W/D       <Status of the small PLC window>
                 Bit 0 = 1:   A small PLC window is in the selected screen (background or
                              foreground).
                 Bit 1 = 1:   The display with the small PLC window is in the foreground
```

Module 9082: Show string in the small PLC window

You must call the module in a submit job. While the module is being run you must not abort the SUBMIT job with a CAN command. The module is also effective when the presently selected screen does not contain a small window (e.g. large graphic display) or when the screen with the PLC window is in the background.

In case of error the string in the PLC window is not shown. The string that is designated with the string number and ends with the ASCII character <NULL> is displayed in the small PLC window in line 0 (upper line) or 1 (lower line) and from column 0 (left margin) to 37 (right margin) in the specified colors (1 to 15).



References to PLC dialogs or PLC error messages are deleted. If the specified dialog or error number is not found, then the ASCII character @ is displayed instead. If the text contains a non-displayable character except the string end, then the ASCII character ^ is displayed instead.

The character size is determined by the size of the screen window, i.e. by the current mode of operation, and cannot be altered. The color refers to MP735x according to the following table:

Number	Machine parameters
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

The color is the background color of the PLC window and therefore cannot be used as foreground color. If color zero is specified, then the text appears in the same color as the last displayed character. Because the complete line is always displayed again in the window when a string is displayed (even when a column greater than 0 is specified), a text (e.g. "Spindle Output") with the color 0 is always displayed in the color of the numerical value to its left.

If the color of the number changes, so does also the color of the text. If, however, the color zero is entered for the first character of a line, then the color of this character remains undefined and can change between two displays of the line.

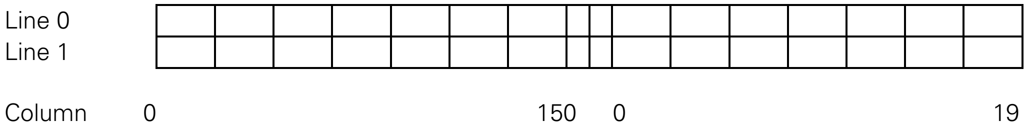
Call only in the submit job:

```
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    <String number>
                0 to 3)
CM   9082
```

Module 9083: Show moving-bar graphic in the small PLC window

You must call this module in a submit job. While the module is running you must not abort the SUBMIT job with a CAN command. The module is also effective when the presently selected screen does not contain a small window (e.g. large graphic display) or when the screen with the PLC window is in the background.

In case of error the string is not displayed in the PLC window. The moving-bar diagram is shown in the small PLC window in the specified line, lengths and colors. It can also be displayed only in the left half of each line. In this case the ASCII text is shown only in the right half of each line (max. 19 lines).



You must specify the line, maximum length (0...150), current length (<= maximum length) and the colors of the bars or the margin and scale graduation (0...15). If the maximum length exceeds 150 it is limited to 150. If current length exceeds maximum length then it is limited to the maximum length.

The chart comprises a rectangular grid with the maximum length and height of the ASCII characters. 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.

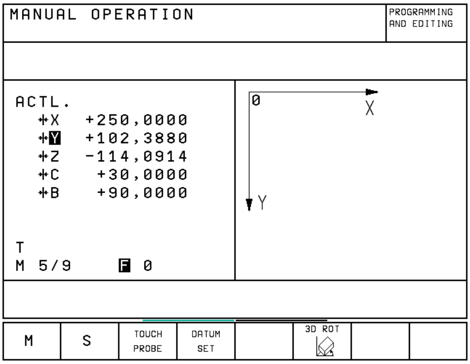
The specified color refers to MP735x and the table for Module 9082. The color zero uses the background color of the currently selected graphic (MP736x.1). The color two is the background color of the PLC window and can be used for margin and scale graduation if you do not want them to be visible.

Call only in the submit job:

```
PS   K/B/W/D    <Line number>
                0 or 1
PS   K/B/W/D    <Color of bar >
                0 to 15
PS   K/B/W/D    <Color of margin and scale graduation >
                0 to 15
PS   K/B/W/D    <Current length of bar>
                0 to 150
PS   K/B/W/D    <Maximum length of bar>
                0 to 150
CM   9083
```

4.13.7 Large PLC Window

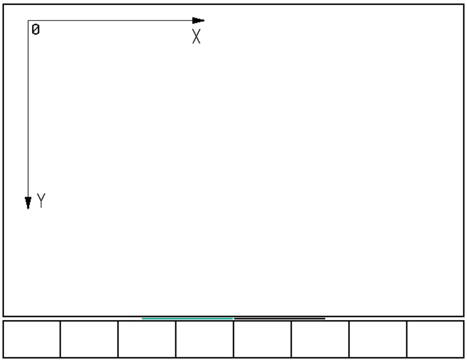
A large PLC window can be shown in place of the graphic window. The large PLC window can be combined with the PLC soft keys. Depending on the type of display, the PLC window is shown in place of the graphic/status window, or it takes up the entire screen. You can select the type of display with the Screen Management key or with Module 9202.



```
MANUAL OPERATION                PROGRAMMING
                                AND EDITING

ACTL.
  +X   +250,0000
  +Y   +102,3880
  +Z   -114,0914
  +C    +30,0000
  +B    +90,0000

T
M 5/9    F 0
```

321 x-pixels, 277 y-pixels

SMALL: 17 lines, 39 columns
0 = y16/x8
MEDIUM: 11 lines, 19 columns
0 = y24/x8
LARGE: 5 lines, 9 columns
0 = y48/x8



639 x-pixels, 433 y-pixels

SMALL: 27 lines, 79 columns
0 = y16/x8
MEDIUM: 18 lines, 39 columns
0 = y24/x8
LARGE: 9 lines, 19 columns
0 = y48/x8

You determine the character size with *charsize=*. The specified position 0 refers to the lower left corner of the first character.

The contents of the PLC window are defined in a screen mask. The screen mask is an ASCII file containing format instructions and special commands. In Module 9210 the name of the screen mask for activating the PLC window is indicated.

Format instructions:
The format instructions are similar to the programming language C and are written in quotation marks ("). Variable names are transferred as parameters.

For example: `"PLC WORD 100: %d PLC MARKER 20: %d",W100,M20;`

Unlike in the standard C language, the letters in the format instruction that specify the variable type can be written in capitals. For example, `"%D"` can be used instead of `"%d"` for a natural number.

The variable types in the format instructions must agree with the indicated variables. Internally, the TNC assigns fixed variable types (see list below). Please remember that, unlike PCs with MS-DOS/WINDOWS, the integer variables in the TNC have a length of 32 bits. If you wish to display a PLC variable as a number with decimal places you must first convert the type from integer to double with the switch */n=x*.

Special characters:

"\n" (Newline) moves the cursor to the left edge and downward by the distance set or preset in LINEDIST.

"\f" (Form feed) has the same effect as "\n" but also inserts a page break when the cursor moves past this special character (otherwise the window is scrolled).

Variable names:

B0 to B4095	PLC bytes _integer_		
W0 to W4094	PLC words _integer_		
D0 to D4092	PLC doublewords _integer_		
M0 to M4999	PLC marker _integer 0/1_		
I0 to I383	PLC inputs _integer 0/1_		
O0 to O191	PLC outputs _integer 0/1_		
T0 to T96	PLC timer _integer 0/1_		
C0 to C143	PLC counter _integer 0/1_		
S0 to S3	PLC strings _char[128]_		
S#D0 to S#D999	PLC dialogs _char*_		
S#E0 to S#E999	PLC error texts _char*_		
TIME[0] to TIME[15]	System time as in Module 9055 _char*_		
AXISCHAR[0] to AXISCHAR[4]	Letter for NC axis _char_		
MP to ..	Machine parameters, format: "MP910.1"		
	Input value:	Decimal places	_double_
		Hex or binary	_integer_
		Text	_char*_

Time:

HOUR	int	Number of hours from the real-time clock
MIN	int	Number of minutes from the real-time clock
SEC	int	Number of seconds from the real-time clock
DAY	int	Day from the real-time clock
MONTH	int	Month as number from the real-time clock
STR_MONTH	string	Month as string abbreviation from the real-time clock
YEAR2	int	Two-digit year two digits from the real-time clock
YEAR4	int	Four digit year from the real-time clock

Settings for the tool touch probe:

TT.RAD	double	Calibrated radius TT
TT.CENTER[3]	double	Calibrated center TT
TT.PNT1[3]	double	Calibrated touch point 0 TT
TT.PNT2[3]	double	Calibrated touch point 1 TT
TT.PNT3[3]	double	Calibrated touch point 2 TT
TT.PNT4[3]	double	Calibrated touch point 3 TT

Settings for RS-232:

RS232.FEBAUD	string	Baud rate FE on RS-232
RS232.EXT1BAUD	string	Baud rate EXT 1 on RS-232
RS232.EXT2BAUD	string	Baud rate EXT 2 on RS-232
RS232.LSV2BAUD	string	Baud rate LSV2 on RS-232
RS232.MODE	string	RS-232 mode

Settings for RS422:

RS422.FEBAUD	string	Baud rate FE on RS422
RS422.EXT1BAUD	string	Baud rate EXT 1 on RS422
RS422.EXT2BAUD	string	Baud rate EXT 2 on RS422
RS422.LSV2BAUD	string	Baud rate LSV2 on RS422
RS422.MODE	string	RS422 mode

Settings for simulation:

SIMU.ENAPRESET	string	Preset enable
SIMU.ENALIMIT	string	Limit switch enable
SIMU.LIMITPL[5]	double	Positive Limit switches
SIMU.LIMITMI[5]	double	Negative limit switches
SIMU.PRESET[5]	double	Preset values

Settings for machine:

MACHINE.LIMIT1PL[5]	double	Positive limit-switch set 1
MACHINE.LIMIT1MI[5]	double	Negative limit-switch set 1
MACHINE.LIMIT2PL[5]	double	Positive limit-switch set 2
MACHINE.LIMIT2MI[5]	double	Negative limit-switch set 2
MACHINE.LIMIT3PL[5]	double	Positive limit-switch set 3
MACHINE.LIMIT3MI[5]	double	Negative limit-switch set 3
MACHINE.PRESET1[5]	double	Preset values 1
MACHINE.PRESET2[5]	double	Preset values 2
MACHINE.PRESET3[5]	double	Preset values 3
MACHINE.POSINC	double	
MACHINE.DRVOFFSET[5]	double	
MACHINE.HANDW_FACTOR[5]	double	Handwheel interpolation 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.PRESAXIS2C[3]	double
MATRANS.PRESZ3_ABC[3]	double
MATRANS.PRESAXIS3	string
MATRANS.TRLPRES1[6]	double
MATRANS.TRLPRES2[6]	double
MATRANS.TRLPRES3[6]	double
MATRANS.MANUAL	string
MATRANS.PGMRUN	string
MATRANS.ANGLE[3]	double

Settings for display:

DISPLAY.AXIS1	string	
DISPLAY.AXIS2	string	
DISPLAY.SCREEN	int	
DISPLAY.FORMAT	string	MM/INCH switchover

Settings for oscilloscope:

`OSC.AXIS`	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

Switches for variables:

`/n=x` for B/W/D; reformats the integer into a floating comma number with *x* decimal points. This switch is always needed, for example, when a position (in the format 0.1 µm) must be shown in millimeters.

`/mi` for B/W/D; reformats the number as an inch expression when the inch conversion is active.

`/e` for B/W/D/M/S; editable field: The original content of the variable is shown and can be changed. You must define the field length in the format string.

`/i` for B/W/D/M/S; Entry field: A value can be entered in the field, which is empty at first. You must define the field length in the format string.

`/c` for B/W/D/M/S/TIME; The field content is cyclically updated. The field length should be defined in the format string, since otherwise the following text could be overwritten if the field length changes due to a change in the numerical value.

Special commands:

`/*<comment>*/` You can enter any comment between the two stars.

`MMINCH` variables that contain a position value (or `/mi` switch) are converted and displayed in inches, provided that this was selected under *Mode*. The default setting is "no conversion."

`POS=xpix,ypix`
Write the next text or graphic at the designated position (`xpix` = distance in pixels from the upper edge of the current page, `ypix` = distance in pixels from the left edge of the window). The default setting is to begin writing at the upper edge and then line by line.

`IPOS=xpix,ypix`
Write the next text or graphic shifted by the indicated number of pixels from the current position (`xpix` = distance in pixels from the current *x* position, `ypix` = distance in pixels from the current *y* position).

`CPOS=column, line`
Write the next text or graphic at the designated position (`line` = line on the current page, `column` = column from the left edge of the window). 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`.

`ICPOS= column, line`

Write the next text or graphic shifted by the indicated number of line and columns from the current position (`line` = line spacing of the current line, `column` = column spacing of the current column). 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`.

`LINEDIST=xpix`

Sets the line spacing. The default setting depends on the character size and is reset every time `CHARSIZE` is called.

`COLOR=[f]` or `COLOUR=[f]`

Sets the foreground color. The value range for `f` is 0 to 15. The default setting is color 11. The colors are describe in Module 9082.

`CURSOR=ON/OFF` Switches the highlighted depiction of the cursor on or off. The default setting is `OFF`

`CHARSIZE=SMALL/MEDIUM/LARGE/AUTO`

Sets the character size. With `AUTO` the character size changes with the size of the window. With a split screen the size is `SMALL`; with a large PLC window it is `MEDIUM`. The default setting is `AUTO`. Every time that `CHARSIZE` is called the value from `LINEDIST` is overwritten by a presetting that is not affected by the character set. The default spacing values are:

	Lines	Columns	
`SMALL`	16	8	pixels
`MEDIUM`	24	16	pixels
`LARGE`	48	32	pixels

`GRAPHICS=<fname>[,p0[,p1[` to `.,p9]]][/c]`

Includes a graphic in the window. The left lower corner is then located at the noted position. It can be set with `POS, IPOS` or `ICPOS`. The entries `p0` to `p9` are optional parameters for transfer to the graphic code. The switch `/c` causes the graphic to be cyclically regenerated in case one of the parameters changes. Graphics can be created either with the soft-key drawing program integrated in the PLC programming software PLCdesign or with a CAD program that can save its files in DXF format. DXF files can then be converted by PLCdesign for use in the TNC.

`<fname>` can contain a file name complete with path, or only the file name. In this case the path is adjusted according to the entry in MP7230.3 (language for help files).

`TEXTFILE=<fname>`

Includes a text file in the PLC window. The file identified with `<fname>` is shown in the window as a text with the previously set attributes. The current position is used for the first line of text, every further line begins at the same *x* position and is shifted downward by the `LINEDIST`. The line break is automatically adjusted to the space available in the window. Any "Linefeed," "Carriage Return," "Horizontal Tab" and "Vertical Tab" characters are converted to spaces. The backslash character "\" is used as an special symbol. It can be used to carry out the following functions:

`"\n","\N"`	Insert a manual line feed (end of paragraph)
`"\f","\F"`	Insert a page feed (separate into several screen pages)
`"\\"`	Backslash character "\" in the Text.

`<fname>` can contain a file name complete with path, or only the file name. In this case the path is adjusted according to the entry in MP7230.3 (language for help files).

`ERRQUE=n[/c][/e][/l][/n][/s]`

Includes a table with the messages momentarily in the PLC error queue. The character `n` indicates the number of lines for the table. With `/c` the table is cyclically updated, `/e` makes it possible to page in the table and acknowledge a specific message with CE. With `/l` a line number (= position in the error queue) appears in front of the error text. The characters `/n`, as an alternative to `/l`, show the error number (= line number in the .PET table). Only `/l` or `/n` are permitted. `/l` has priority. With `/s` a three-digit status field with the following information is shown.

`C`	CE possible
`S`	Message causes a stop,
`E`	Message causes an EMERGENCY STOP
`F`	Resets the feed-rate enable
0 to 2	Priority

`REFRESH=n`

Time interval (in ms), in which all variables with the `/c` switch are checked and, if necessary, displayed again. Values between 100 and 100 000 [ms] are permitted. The default value is 400 ms.

`KBD` The keyboard is assigned to the PLC window as long as it is visible on the screen. The command is needed only if you wish to move a long text in the PLC window with the cursor keys. You can then use the cursor keys to scroll the PLC window or page through the screens (if the page limits were defined with `\f`). If the mask contains elements with an `/e` or `/i` switch, the keyboard is automatically assigned to the PLC window. The cursor keys then move the cursor among the input fields. The soft keys, screen shift keys, operating mode keys, and 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 entered position. The command `xpix=` pixel distance from the upper edge of the current page, `ypix=` is the pixel distance from the left edge of the window. Afterwards the entered position becomes the actual position.

`ILINE=xpix, ypix`

Draws a line from the current position to a position at the distance `xpix, ypix` from it. The values `xpix` and `ypix` represent the line lengths in x and y, respectively. Afterwards the current position is shifted by `xpix, ypix`.

`LINESTYLE=SOLID/DASH/LDASH`

Definition of the line type for the `LINE/ILINE` command. The default setting is `SOLID`. The width of the lines is always on pixel and cannot be changed.

`SOLID` = solid line

`DASH` = dotted line

`LDASH` = dot-dash line

`FILE=<table name>`

Open a table which can then be read with `TABREAD`. Only one table can be open at any time. If you call up the menu item `FILE=` several times, the previous table will be closed each time. At the end of the mask the table is closed automatically.

`TABREAD`(line,column)

Read the table which you have just opened with `FILE=`.

Mathematical expressions for display positions

Wherever a numerical value is expected as parameter with special functions, a mathematical expression may also be written in integer arithmetic. Here the operators and priority rules of the programming language C apply.
Available operations: +, –, *, /, %, &, |, ^.

The expression may also contain the following variables:

PAGE	Number of the current page (beginning with zero)
XPOS	*x* position of the cursor in pixels
YPOS	*y* position of the cursor in pixels
LINEDIST	Current line spacing in pixels
ROWDIST	Current 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 assign input fields to the variables. The switch `/e` shows the present value of the variable that can then be overwritten. The switch `/i` shows an empty field in which a new value can be entered. As soon as the switch `/e` or `/i` is used, the cursor keys move the cursor among the input fields. If necessary the current page is scrolled. Under certain circumstances, text that is located between the input fields may no longer be visible. For this reason, especially, you should refrain from editing any text before the first input field or after the last one.

The format instruction stored in the mask file must contain a format suited for the C command "printf", which defines the exact length of the numerical field (highlight). Otherwise the length of the input field depends on the coincidental contents of the accompanying variable. For the input function this format instruction is converted to a form suitable for the C command "scanf":

```
printf:      %[flags][digits1][.[digits2]][l]conversion_char

scanf:       %[digits1][size]conversion_char
```

Please note the special characteristics of the following formats:

%d, %e The normally required size data "`l`" can be omitted, since all floating comma variables are of the double type and this information is therefore automatically added.

%g Do not use. Results in errors.

%i Should be avoided since a number with leading zeros will be interpreted as an octal number.

%u Can be used, but it works correctly only in the range of the numbers (with algebraic sign) possible for the respective variable.

The Size indicator `h` (short integer) of the *scanf* function cannot be written. It is not necessary, however, since all integer variables are expanded for 32-bit input/output.

The data in entered in the input field through the ASCII keyboard and the numeric keypad. The following keys have a special function:

CE Deletes either an displayed error message or the input field.

ENT Takes the input value as the variable and moves the highlight to the next input field. If the input value is syntactically incorrect or exceeds the numerical range of the assigned variable, the error message **ENTRY VALUE INCORRECT** appears.

NOENT Returns the previous value to the field and moves the highlight to the next input field.

–/+ If the input value begins with + or – , the sign will be switched.

<x If the field was already edited, the last character of the entry is erased. Otherwise the displayed value is put into the editing memory and the pointer is placed at the end of the input value.

Module 9210: Open or erase screen mask for PLC window

With Module 9210 you activate or erase the display in the large PLC window. You define the file name and path of the screen mask in one of the string memories S0 to S3. If you do not indicate the path name, the path for the language indicated in MP7230.3 (Help files) is used. The transfer value –1 causes the PLC window to be erased.

```
Call:
PS   B/W/D/K    <No. of string memory / Erase>
                0 to 3 =     String memory S0 to S3
                –1 =         Erase PLC window
CM   9210
PL   B/W/D      <Status/Error>
                0:      No error: Mask opened / Mask erased
                1:      PLC window not ready yet
                –1:     Error
```

If an erroneous mask file was activated an error message is shown 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`	Illegal format statement
`STRING NOT CLOSED`	String end is missing
`TOKEN TOO LONG (>32 CHAR)`	Variable name too long
`PARAMETER INDEX MISSING`	Index is missing Closing bracket "]" is missing
`SOURCE FILE NOT OPENED`	Source file has not been opened
`TEMPORARY FILE NOT OPENED`	Temporary destination file not open
`TOO FEW PARAMETERS`	Too few parameters for format statement
`WRONG COMMAND PARAMETER`	Parameter doesn't match format
`WRONG PARAMETER SWITCH`	Incorrect switch data

Module 9211: Status of the large PLC window

With Module 9211 you interrogate the status of the large PLC window.

Number	Read value
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

Call:

```
PS    B/W/D/K    <Number>
CM    9211
PL    B/W/D      <Status information>
                 –1: error
```

4.13.8 PLC Soft Keys

Through the PLC you can display your own soft keys in the "Manual," "Electronic handwheel," "Positioning with MDI," "Program run, full sequence" and "Program run, single block" operating modes. You can draw the soft keys with the drawing program integrated in the PLC programming software PLCdesign.

You must enter the name and complete path of the soft-key file names in the system file PLC:\ PLCSOFTK.SYS. The soft-key number is defined with sequence of entries in PLCSOFTK.SYS (line 0 = soft key 0 etc.). Each level can contain up to 32 soft keys, which equals four soft-key rows. When calling the module you indicate which soft-key row should be shown first.

The transfer parameter defines whether the PLC soft keys are shown after selection of the PLC window (screen manager key) or are shown immediately in the current machine mode. In the second case the NC soft keys may be overwritten (you must also select whether the NC soft keys should be overwritten or whether the PLC keys should be added to the NC keys. In the latter case a separate row is opened for the PLC keys. Only one PLC row can be added).

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 display with PLC soft keys and PLC window. This module functions like the screen management key.

Module 9200: Display or erase PLC soft-key row

The line number of the soft keys to be activated are entered in a constant field. If there is no file by the name of PLCSOFTK.SYS or the lines entered in the constant field do not exist, no soft-key row is generated.

Call only from sequential program:

```
PS   B/W/D/K/KF <Select or erase address of soft key>
                –1 =   Erase 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 with visible PLC window
                1 =    Soft-key row in current operating mode
                2 =    Add soft-key row to NC soft keys[1]
CM   9200
```

W1022 =	1:	Incorrect transfer parameter (e.g. KF address not in the address range of the PLC code
	2:	Line no. < 0 (however not –1) in the constant field
	24:	Module was called in submit job
	25:	More than 32 elements in the constant field

[1] ***NC software 280 470 ..***
NC and PLC soft keys cannot be displayed together

Module 9201: Display or erase PLC soft key

If there is no file with the name PLCSOFTK.SYS or the indicated line does not exist, no soft key is generated. If a PLC soft-key level already exists the soft key will be shown or erased at the indicated position.

Call only from sequential program:

```
PS   B/W/D/K    < Soft-key no. (line no.) / Erase >
                ≥ 0 =   Line no.
                –1=     Erase soft key
PS   B/W/D/K    <Position no.>
                0 to 31
PS   B/W/D/K    <Soft-key mode>
                0 =     Soft key with visible PLC window
                1 =     Soft key in current mode of operation
                2 =     Add soft-key to NC soft keys1)
CM   9201

W1022 =         1:      Transfer parameter out of value range
                2:      Line no. < –1
                24:     Module was called in submit job
```

Module 9202: Selecting and deselecting PLC soft keys and PLC windows

With Module 9202 you activate the display with PLC window or the display with PLC soft keys. This module functions like the screen management key.

Call:

```
PS   B/W/D/K    <Display mode>  0 = Deselect PLC soft key / window
                                1 = Select PLC soft key / window (small)
                                2 = Select PLC soft key / window (large)
CM   9202

W1022 =         1:      Transferred parameter is out of range
```

		Set	Reset
W302	Number of the activated PLC soft key	NC	NC

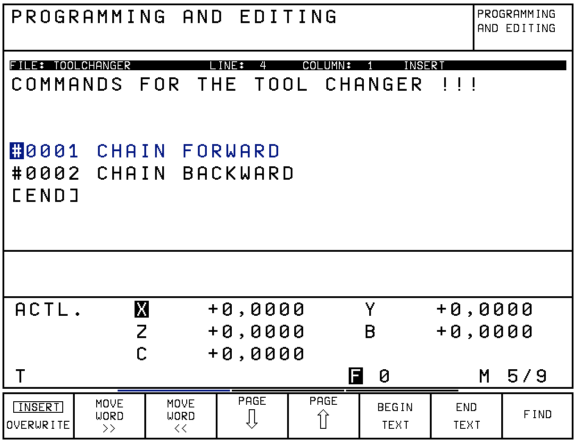
[1] ***NC software 280 470 ..***
NC and PLC soft keys cannot be displayed together

4.13.9 Help

Help soft key in MOD:
With the help file you can display useful hints and operating instructions or machine commands. If a help file with the extension .HLP is defined in the system file OEM.SYS with the command **MODEHELP=**, the soft key HELP appears when the MOD key is pressed. With this soft key you can call the information.

You can edit the help file in the PLC editor (code number 807667). For machine commands you define a numerical value in the format #xxxx at the beginning of the line. As soon as the user moves the cursor to a line with a numerical value, this number is displayed in W270. You can then interrogate W270 in the PLC program and execute the corresponding command. If the cursor is moved to a line without a valid numerical value, then the value –2 is entered in W270. If no help file is selected, the value –1 is in W270.

Example:



```
PROGRAMMING AND EDITING                    PROGRAMMING
                                           AND EDITING
FILE: TOOLCHANGER        LINE: 4    COLUMN: 1    INSERT
COMMANDS FOR THE TOOL CHANGER !!!


#0001 CHAIN FORWARD
#0002 CHAIN BACKWARD
[END]



ACTL.      X     +0,0000     Y     +0,0000
           Z     +0,0000     B     +0,0000
           C     +0,0000
T                          F 0         M 5/9
```

You can create several help files. To select the corresponding file, the user presses first the PGM MGT key and then the HELP soft key. With MP7230.3 you select the dialog language for the help files, i.e. the entry behind **MODEHELP=** in the file OEM.SYS is overwritten with the path for the desired language. If a HELP file is selected both in the foreground and background modes, then the message **BACKGROUND PROGRAMMING NOT POSSIBLE** will appear.

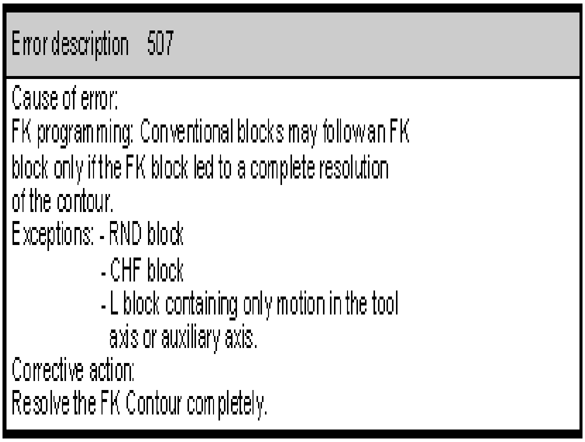
HELP files are stored externally with the identifier "J".

			Set	Reset
W270	Help-file line number		NC	NC
	–1 =	No help file selected		
	–2 =	No valid value		
	0 to 9 999 =	Line number		

Help window with HELP key:[1]

If an NC error message is displayed, you can call up a help window with the HELP key. This window offers you information on the possible causes of the error and some suggestions for correcting it.

Example:



Error description 507

Cause of error:
FK programming: Conventional blocks may follow an FK
block only if the FK block led to a complete resolution
of the contour.
Exceptions: - RND block
- CHF block
- L block containing only motion in the tool
axis or auxiliary axis.
Corrective action:
Resolve the FK Contour completely.

If you wish the same to happen with PLC error messages, you must store the corresponding error texts in files. Set up two files: one for the "causes of error" and one for the "correction of error". The names of the two files must be defined in the system file OEM.SYS with the commands `PLCERRREASON=` (cause of error) and `PLCERRFIX=` (correction of error). Store the files in the corresponding language directory (PLC:\LANGUAGE\<language>), and using MP7230.3 select the current language (file).

The files are divided into text blocks. Each text block can have a maximum of 10 lines with each 60 characters and is ended with <FF>. Use the keys SHIFT and RETURN to enter <FF> on the TNC. The TNC locates the appropriate text block in the files (with description of error and suggestions for correction) via the error number (line number) in the PET table. Error number 0 is the first text block. The error number is also displayed in the header of the help window.

A simpler alternative would be to create a PET table with the appropriate text files and transfer it to the TNC using the PLC development software PLCdesign.

[1] As of NC software 280 472 01

4.13.10 Superimpose PLC Window

As of NC software 280 472 01:

With Module 9215 you can activate a pop-up window where the user can choose settings from a list. This PLC window is only displayed in the operating modes "Manual", "Positioning with MDI", "Program run, single block" and "Program run, full sequence". You can make your selections using the cursor keys and ENTER or using key combinations. The module sends the line number of the selected menu item (line 1 = number 0).

The selection list you require is defined and transferred to the module as a file. Individual entries in the files are separated with <LF>. If you send a file name without indicating the path, the TNC will search for the file in the language-specific directory PLC:\LANGUAGE\<language>. The current language (file) can be selected using MP7230.3.

Other pop-up windows (e.g. HELP window) are placed in the background when the PLC window is called, and become active again once it is closed. Likewise the PLC window can be moved to the background by calling another pop-up window.

As the module only returns when the window is closed, it must be called in a SPAWN process. If it were called in a submit job, the following submit jobs would only be executed when the window is closed again.

Module 9215 uses the following events:

$01 000 000	Window design
$00 010 000	Closing the pop-up window

The event for window design is generated internally and may not be set externally. When the event for closing the pop-up window is transferred, the module is ended without any keyboard input.

Call only in SPAWN process:

```
PS   B/W/D/K/S  <String with window heading>
PS   B/W/D/K/S  <String with file name of the list>         [with directory or just file name]
PS   B/W/D/K    <Starting position>                          [line that is highlighted]
PS   B/W/D/K    <Mode>
                                 Bit0/Bit1:  Size of characters
                                       00:   Automatic
                                       01:   Small characters
                                       10:   Medium-size characters
                                       01:   Large characters
                                 Bit2: Frame
                                       0:    With frame
                                       1:    Without frame
                                 Bit3: List name
                                       0:    Display name
                                       1:    Do not display name
                                 Bit4: Key combinations
                                       0:    No key combinations
                                       1:    Key combinations (0 to 9 and A to Z) before menu items
                                 Bit5: Vertical/horizontal
                                       0:    Arrange vertically
                                       1:    Arrange horizontally (bit 4 = 0)
```

CM 9215
PL B/W/D <Selected line>

0 to n:	Line number from list
–1:	No selection (END, NOENT)
–2:	Error see W1022

W1022 =

2:	Incorrect mode transferred
3:	No valid string for file name or heading
6:	Window cannot be shown (internal error, e.g. problem with memory capacity or operating system)
20:	Module was not called from a SPAWN process
28:	Another PLC pop-up window is already open
36:	The file containing the list could not be opened.

4.13.11 M Functions

In the TNC you can program miscellaneous functions (M functions). The code for these M functions is transferred to the PLC either before or after execution of the NC block. Some of these 1M functions have a fixed meaning for the NC. These M functions are marked with * in the following table. The other M-functions are freely available.

	M function	Effective at: Start of block	End of block
*	M 00		●
	M 01		●
*	M 02		●
*	M 03	●	
*	M 04	●	
*	M 05		●
[1]	M 06		●
	M 07	●	
*	M 08	●	
*	M 09		●
	M 10		●
	M 11	●	
	M 12		●
*	M 13	●	
*	M 14	●	
	M 15	●	
	M 16	●	
	M 17	●	
	M 18	●	
	M 19		●
	M 20	●	
	M 21	●	
	M 22	●	
	M 23	●	
	M 24	●	
	M 25	●	
	M 26	●	
	M 27	●	
	M 28	●	
	M 29	●	
*	M 30		●
	M 31	●	
	M 32		●
	M 33		●

M function	Effective at: Start of block	End of block
M 34		●
M 35		●
M 36	●	
M 37	●	
M 38	●	
M 39	●	
M 40	●	
M 41	●	
M 42	●	
M 43	●	
M 44	●	
M 45	●	
M 46	●	
M 47	●	
M 48	●	
M 49	●	
M 50	●	
M 51	●	
M 52		●
M 53		●
M 54		●
M 55	●	
M 56	●	
M 57	●	
M 58	●	
M 59	●	
M 60		●
M 61	●	
M 62	●	
M 63		●
M 64		●
M 65		●
M 66		●
M 67		●

	M function	Effective at: Start of block	End of block
	M 68		●
	M 69		●
	M 70		●
	M 71	●	
	M 72	●	
	M 73	●	
	M 74	●	
	M 75	●	
	M 76	●	
	M 77	●	
	M 78	●	
	M 79	●	
	M 80	●	
	M 81	●	
	M 82	●	
	M 83	●	
	M 84	●	
	M 85	●	
	M 86	●	
	M 87	●	
	M 88	●	
[1]	M 89		●
*	M 90	●	
*	M 91	●	
*	M 92	●	
*	M 93	●	
*	M 94	●	
	M 95		●
	M 96		●
*	M 97		●
*	M 98		●
*	M 99		●

[1] Function is dependent on MP7440

If an M function does not have a fixed meaning for the NC, you must evaluate it in the PLC. When an M function is transferred to the PLC, its code is stored in W260 and the strobe marker M4072 is set.

The next NC block will not be run until you have acknowledged the execution of the M function by setting M4092. As soon as you set M4092, the NC resets M4072. The M functions M00 to M99 can also be transferred decoded to the markers M1900 to M1999. You must activate this function in M4571. All M functions over M99 are not transferred to the PLC. However, they have a fixed meaning for the activation of functions (see User's Manual).

		Set	Reset
W260	Code for M function	NC	NC
M4072	Strobe signal for M function	NC	NC
M4092	Acknowledgment of M function	PLC	PLC
M4571	Activation of decoded M-code transfer in M1900 to M1999	PLC	PLC
M1900 to M1999	Decoded M function if M4571 is set	NC	NC

Program stop on M functions

In the "Program Run, Full Sequence" and "Program Run, Single Block" operating modes the next NC block is not run until you have acknowledged the execution of the M function by setting M4092. You can deselect this program stop function for special machines by using MP7440 bit 2 (see chapter "Special Functions for Laser Cutting Machines").

Program stop on M06

According to ISO 6983, the M function M06 commands a tool change. With machine parameter MP7440, bit 0 you can select whether the program should stop when M06 is transferred to the PLC. If you have configured the control to stop the program on M06, then you must restart the program after the tool change. This can also be carried out by the PLC.

Modal cycle call M89

You can use the M function M89 for a modal cycle call.
The possibilities for calling a cycle are:

- With the NC block CYCL CALL
- With the miscellaneous function M99. M99 is only effective for a single block and must be reprogrammed for each execution.
- With the miscellaneous function M89 (depending on the MP7440 bit 1). M89 as a cycle call is modally effective, i.e. for every following positioning block there will be a call of the machining cycle last programmed. M89 is canceled by M99 or a CYCL CALL-block.

If M89 is not defined as a modal cycle call, then M89 will be 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 contour feed rate for movements of the tool axis in the negative direction. The feed-rate component of the tool axis is limited to a value that the TNC computes from the last programmed feed rate.
$F_{max} = F_{prog} * F_{\%}$
F_{max} = Maximum feed rate in negative direction of tool axis
F_{prog} = Last programmed feed rate
$F_{\%}$ = Programmed factor after M103 in %
M103 F... is canceled by entering M103 without a factor.
You can enable M103 F... with MP7440 bit 4.

MP7440	Output of M functions	
Input:	%xxxxx	
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 M function
	1 =	no program stop (do not wait for acknowledgment)
Bit 3	Select k_v factors with M105/M106	
	0 =	Function not effective
	1 =	Function effective
Bit 4	Reduced feed rate in the tool axis with M103	
	0 =	Function not effective
	1 =	Function effective

4.13.12 Error Messages

Under certain conditions error messages from the NC or the PLC will be displayed on the screen under the display for the operating mode. In the event of a blinking error message the machine must be switched off and the fault corrected. If a non-blinking error message is displayed on the screen, marker M4177 will be set.

You define the PLC error messages in the .PET table. A PLC error message is called either with Module 9085 or through activation of a marker (M4800 to M4899). If you use the markers, however, you must first assign them to the error messages in the .PET table. With Module 9086 you can erase the PLC error message; with Module 9087 you can interrogate the current status of the error message.

With the command **PLCERRTAB=** you must enter the name of the .PET table in the OEM.SYS file. With COMPILE xxxxx.PET the entry is made automatically in OEM.SYS. If several PLC error messages are activated at the same time, you can read them in sequence by pressing the CE key. With the special command **ERRQUE=** you can display the list of active error messages in the large PLC window.

In the individual columns you can assign special attributes to the PLC error messages. The PLC error message table (.PET) consists of the following columns:

NR
Line number in the table. The PLC error message is addressed in the modules by assigning the line number.

ERROR
Error text. There are three possibilities for defining the error text:

- Directly entering the error text (max. 32 characters)
- Line number of the PLC error text file (#<line no.>) defined in OEM.SYS with PLCERROR =.
- Number of the string memory where the error text is located (#S<string no.>)

You can call up additional information on the error messages by pressing the HELP key (see section "Help".)

MARKER
The PLC error message can be activated without a module call by setting the marker defined here. Only markers in the range of M4800 to M4899 may be entered. The marker is also set if the error message was activated through Module 9085. The entry value 0 means no error marker.

RESET
0 = No NC reset upon activation of the error message. Non-blinking error message.
1 = NC reset upon activation of the error message. Blinking error message.

NC STOP
0 = No NC stop when error message is activated.
1 = NC stop when error message is activated.

F STOP
0 = Feed-rate enable.
1 = Feed-rate enable is reset when error message is activated.

`EMER.STOP`

0 =	No EMERGENCY STOP when error message is activated.
1 =	EMERGENCY STOP when error message is activated.

`CE`

0 =	Error message can be deleted with the CE key.
1 =	Error message cannot be deleted with the CE key.

`PRIOR`

A priority level of 0 to 2 can be entered for the error message. Priority 0 is the highest priority level. The active error messages are displayed in the sequence of their priority.

Module 9085: Display the PLC error message

Up to 32 error messages can be placed in the queue, up to eight of which can be from the string memory. A blinking error message is displayed immediately without an entry in the queue. If you enter the error number –1, the blinking error message **`EMERGENCY STOP PLC`** is shown. This message is shown even if you have not defined a .PET table. If no .PET table was selected and you have entered an error message other than –1, the blinking error message **`PLC: error table missing`** is shown.

Call:

```
PS    B/W/D/K     <Line no. in PET table>
                  0 to 999:    Line no.
                  –1:          Blinking error message EMERGENCY STOP PLC
CM    9085
```

```
M4203 =           0:      Error message is displayed or is in the queue
                  1:      Error code in W1022

W1022 =           1:  Line number not found
                  8:  Incorrect operating mode, i.e. error marker compatibility is set
                  23: Overflow of PLC error-message queue or too many error messages from string
                      memory
```

Module 9086: Erase the PLC error message

With Module 9086 you can erase all set PLC error messages, or specifically erase a displayed error message or a (non-blinking) error message waiting in the queue.

Call:

```
PS    B/W/D/K     <Line no. in PET table>
                  0 to 999:    Line no.
                  –1:          Erase all PLC error messages
CM    9086
```

```
M4203 =           0:      No error
                  1:      Error code in W1022

W1022:            1:  Line number not found
                  8:  Incorrect mode of operation, i.e. error marker compatibility set
```

Module 9087: Status of PLC error message

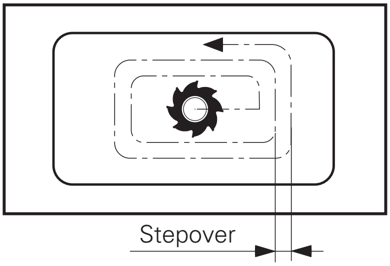
Call:

```
PS    B/W/D/K     <Line no. in PET table >
                  0 to 999:    Line no.
                  –1:          PLC error message general
CM    9087
PL    B/W/D       <Status/Error code>
                  0:       No error message with this number, or it was erased
                  Bit 0:   PLC error message displayed
                  Bit 1:   PLC error message in queue
                  –1:      Line number not found
```

4.13.13 Cycles

With HEIDENHAIN contouring controls it is possible to call standard cycles (e.g. pecking, tapping, pocket milling, etc.) from within the part program. You can also program you own Original Equipment Manufacturer's (OEM) cycles (see section "OEM Cycles"). You can influence the function of many HEIDENHAIN standard cycles through machine parameters. For more information on the tapping cycle and the oriented spindle stop cycle see the section "Spindle." See the section "Touch Probes" for more on the touch probe cycles.

Pocket milling
You enter the overlap factor for clearing out a rectangular or circular pocket (Cycle 4 and Cycle 5) MP7430.



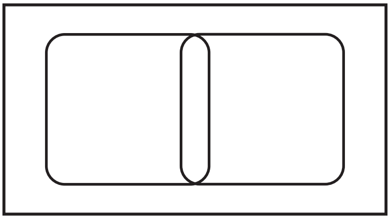
Stepover = (MP7430) • cutter radius

MP7430	Overlap factor for pocket milling
Input:	0.1 to 1.414

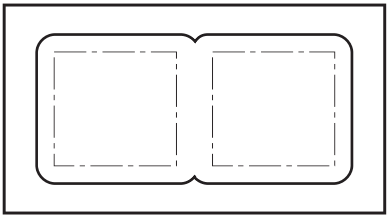
Milling cycles for pockets with combined contours
In MP7420 you can alter the function of cycles for milling pockets with variable contours (Cycles 6, 14, 15, 16).

You can decide:

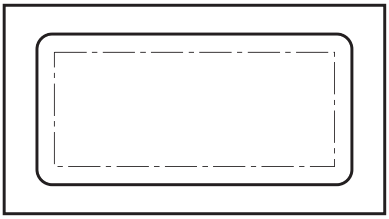
- Whether a channel should first be milled on the contour and the pocket cleared out afterwards, or the pocket cleared out first and then a channel milled on the contour.
- Whether the channel should be milled in a clockwise or counterclockwise direction.
- Under which conditions programmed pockets should be merged. You can select between merging programmed pockets when the programmed contours overlap, or when the tool center paths intersect.
- Whether each process (channel-milling or clearing) is completed for all pecking depths before performing the other process, or both are performed for each pecking depth.
- Whether at the end of the cycle the tool should return to its previous position, or simply to the "clearance height."



The programmed contours of two pockets intersect slightly.



MP7420 Bit 2 = 0:
The control clears out the pockets separately, since the tool center paths do not intersect. Material will remain in inside corners.



MP7420 Bit 2 = 1:
The control clears out the pockets jointly, since the programmed contours overlap. No material will remain in inside corners.

MP7420 Cycles for milling pockets with combined contours
Input: %xxxx

Bit 0 Traverse direction for channels
0 = pockets counterclockwise, islands clockwise
1 = pockets clockwise, islands counterclockwise
Bit 1 Sequence for clearing and channel-milling
0 = first mill the channel, the clear the pocket
1 = first clear the pocket, then mill the channel
Bit 2 Merging of listed contours
0 = contours are combined only if the tool center paths intersect
1 = contours are combined if the programmed contours intersect
Bit 3 Clearing and pocket-milling to pocket depth or for each pecking depth
0 = each process uninterrupted to pocket depth
1 = both processes for each pecking depth before proceeding to the next depth.
Bit 4 Position after machining the cycle
0 = tool moves to the same position as before the cycle was called
1 = tool moves only to the "clearance height"

Scaling factor
In MP7410 you indicate whether Cycle 11 "scaling factor" should be effective only in the working plane or also in the tool axis.

MP7410 Scaling factor cycle in two or three axes
Input: 0 = Scaling factor cycle effective in all three primary axes
1 = Scaling factor cycle effective only in the working plane

Cylindrical surface
With Cycle 27 "cylinder surface" it is possible to machine a contour on a cylindrical surface (see the User's Manual). For this cycle to be effective you must first define the center of rotation of a rotary axes in MP7510 and following (see section "Tilting Axes").

4.13.14 Returning to the Contour

With this HEIDENHAIN contouring control it is possible to resume an interrupted program, or to make a block scan up to a predetermined block number (see User's Manual).

These functions must be enabled by machine parameters and the PLC program must be adapted accordingly.

Markers inform the PLC about individual conditions during mid-program startup (also called *block scan*). Depending on these markers, you can enable certain functions in the PLC program (e.g. operating the axis-direction keys in MANUAL OPERATION).

M4156 is set if the MANUAL OPERATION soft key is set. M4157 is set if the return to contour function is activated with the RESTORE POSITION soft key. M4158 is set if the RESTORE POS. AT N soft key is pressed M4158 is reset if the RESTORE POSITION or INTERNAL STOP soft key is pressed.

During a block scan, PLC positioning commands are included in the calculation only if they are also executed. The TOOL CALL block normally releases PLC commands to move the tool to the tool-change position. If you want these positioning commands to be calculated in the block scan you must enter in MP951.x the absolute tool-change position referenced to the machine datum and activate the calculation for the specific axes with MP7450.

With flexible tool pocket coding in the central tool file (see section "Tool Changer"), a change in the position numbers in the tool memory must be prevented during block scan. This is ensured by setting M4542.

With MP7680 you define whether the block scan should be interrupted through a programmed STOP or M06, and whether the programmed dwell time should be considered during the block scan. You define the feed rate for returning to the contour in MP7451.x.

The tool data cannot be correctly offset in the block scan if you change them in the PLC or update them with M4538.

MP951.0-8 Simulated tool change position for TOOL CALL during block scan
Input: –99 999.9999 to +99 999.9999 [mm] or [°]

MP7450 Calculate the tool change position from MP951 in block scan
Input: %xxxxx

Bit	0 to 8	0 = do not calculate
Axis	1 to 9	1 = calculate

MP7451.0-4 Feed rate for returning to the contour
Input: 10 to 300 000 [mm/min]

MP7680 Machine parameter with multiple function
Input: %x xxx xxx
Bit1 Return to the contour
0 = not active
1 = active
Bit2 Block scan
0 = not active
1 = active
Bit3 Interruption of block scan with STOP or M06
0 = interruption
1= no interruption
Bit4 Inclusion of programmed dwell time during block scan
0 = include dwell time
1 = do not include 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 OPERATION soft key was pressed	NC	NC
M4157	Returning-to-contour function (RESTORE POSITION) active	NC	NC
M4158	Block scan is active	NC, PLC	NC, PLC
M4542	Do not update the pocket number in the pocket table	PLC	PLC

M/S/T/Q transfer during block scan

With MP7681 you define whether the M/S/T/Q signals should be collected during the block scan in order to output them after it is ended. If MP7681 ≠ 0, after a block scan the dialog **`Restore machine status:`** is displayed with the M/S/T/Q signals to be output. With the external start key you then output the displayed signals. M4161 remains set for the duration of output.

You define the M functions that you wish to be output after a block scan in the system files PLC:\MGROUPS.SYS and PLC:\MSPLIT.SYS. The M/S/T/Q signals are normally output in the following sequence (for exceptions see the instruction ORDER.PRIO):

1. M function that was defined with MFIRST
2. M/S/T/Q signals in the sequence in which they were entered in the NC program.
3. M function that was defined with MLAST

Instructions in MGROUPS.SYS:

GROUP=

With this instruction you divide the 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=

With SPECIAL = you define all M functions that should be sent to the PLC after a block scan and are not defined in a group.

Example: SPECIAL=M19,M22,M25

MFIRST=

MLAST=

With MFIRST= and MLAST= you define the M functions to be sent to the PLC at the start and end, respectively, of an output sequence after a block scan. In this way you can recognize from the PLC program that a sequence of M/S/T/Q strobes is being transferred that was collected during a block scan. You can omit these instructions if you do not need this information.

Example: MFIRST=M80
MLAST=M81

REMAIN=OUTPUT

If you enter this instruction, all M functions that are not defined in MGROUPS.SYS are transferred to the PLC during a block scan. If you do not enter this instruction such M functions are ignored. With 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

ORDER=PRIO

If you enter this instruction, the M functions will be transferred in the sequence in which they are entered in the file MGROUPS.SYS. 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 not use this instruction.

TOOLGROUP, TDEFGROUP, SPINDLEGROUP, FN19GROUP

With these instructions, in conjunction with ORDER=PRIO, the sequence of output of the S/T/Q strobes after a block scan is defined in the file MGROUPS.SYS. HEIDENHAIN recommends that you not use these instructions.

NCMACRO=TC

During a TOOL CALL you can also call an NC program instead of the T strobe (see chapter "Tool Changer"). With the instruction NCMACRO=TC you define that this tool-change program is not run during the block scan, but rather is started at the end of the block scan instead of the T strobe.

Instructions in MSPLIT.SYS:

M functions that are effective in several groups are divided in the file MSPLIT.SYS into function components.

Example: M13=M3,M8

MP7681 M/S/T/Q transfer to PLC during block scan
Input: %xxxx

Bit 0
0 = Output the M functions to the PLC during block scan.
1 = Collect the M functions and output them to the PLC after a block scan.

Bit 1
0 = Output the T code to the PLC during a block scan.
1 = Output the last T code to the PLC after a block scan.

Bit 2
0 = Output the S or G code to the PLC during a block scan.
1 = Output the last S or G code to the PLC after a block scan.

Bit 3
0 = Output the FN19 outputs to the PLC during a block scan.
1 = Output the last FN19 outputs to the PLC after a block scan.

		Set	Reset
M4161	M/S/T/Q transfer after mid-program start-up	NC	NC

4.13.15 Files

The TNC enables you to edit various file types. File types are designated with an extension after the file name. The file name can consist of up to eight characters (letters and numbers).

Disabling soft keys for file types
With the soft key SELECT TYPE, you can display a soft key for each file type. These soft keys can be disabled individually with MP7224.0.

Disabling file types for editing
You can protect individual file types with MP7224.1 so that they cannot be edited or changed. Protected files are displayed in the file directory with the colors defined in MP7354.1 or MP7355.1.

MP7224.0	Disable soft keys for file types		
Input:	%xxx xxxxx		
Bit 0	HEIDENHAIN program	.H	0 = do not disable
Bit 1	ISO program	.I	1 = disable
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	

MP7224.1	Protect file types		
Input:	%xxx xxxxx		
Bit 0	HEIDENHAIN programs	.H	0 = do not protect
Bit 1	ISO programs	.I	1 = protect
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	

Block-number increment with ISO programs
In MP7220 you enter the block-number increment with ISO programs.

MP7220	Block number increment for ISO programs
Input:	0 to 250

4.13.16 Datum Tables (.D)

In a datum table you can define up to 255 datums. You define the size of the datum tables in MP7226.1. With Cycle 7 "datum shift" you can enter either the absolute value of the new datum or the line number from the datum table (see the User's Manual). With Modules 9092 to 9094 you can use the PLC to read and write to the current datum table. With FN17 and FN18 you can read and overwrite the values in the datum table (OEM cycles).

MP7226.1 Size of the datum tables
Input: 0 to 255 [lines]

Reference for values in the datum table
With MP7475 you define whether the values in the datum table refer to the set workpiece datum or the machine datum (MP960.x).

MP7475 Reference in datum table
Input: 0 = Reference is the workpiece datum
1 = Reference is the machine datum (MP960.x)

4.13.17 Pallet Management

As of NC software 280 472 01

Configuring a pallet table:
The pallet table is a freely defined table. Define the prototype for this table in the directory PLC:\PROTO, giving it the file name extension P. If you don't define your own prototype, the standard form will automatically be used. This standard table format contains the fields PAL/PGM, NAME, DATUM, X, Y, Z. If you have already defined several prototypes with file name extension P, you will be given a menu from which you must select the desired format for your pallet table. Of course the various formats must be suited to your PLC program. Refer to the section "Freely Defined Tables" for information on how to define prototypes.

Field designations
The following different types of fields are differentiated:

- Obligatory fields:
  Values must be entered in these fields
- Optional fields:
  You do not have to enter values in these fields. However they have a fixed predefined meaning for the NC.
- Freely defined fields:
  You also have the opportunity to define your own additional fields and assign them a meaning. These entries serve as information only, or can be interrogated and changed via the PLC.

Name	Meaning
PAL/PGM	(Obligatory) Definition of the entry (PAL=Pallet / PGM=NC program).
NAME	(Obligatory) Name of the pallet or the NC program. If an NC program name has no path indication it will be searched for in the same directory as the pallet table can be found. You should only allow decimals be used for pallet names, so that the name can then be interrogated with FN18 in the pallet change macro.
DATUM	(Optional field) Name of the datum table. If a datum table has no path indication it will be searched for in the same directory as the pallet table can be found..
X, Y, Z, U, V, W, A, B, C	(Optional fields) Definition of the datum. For pallet entries these 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.
LOCATION	(Optional field, not used in standard format) Location of the pallet. Entries are only necessary in pallet lines. If the LOCATION field exists, an NC program can only be run if this field contains the entry MA (= pallet in machine).
Others	Freely defined

Executing a pallet table

The pallet table is selected in the operating mode "Program Run, Full Sequence" or "Program Run, Single Block" with PGM MGT (as for a normal machining program) and started. Pallet entries (PAL) call the NC macro for pallet change. Program entries (PGM) are executed like a PGM CALL. With MP7683 you can define the behavior at NC start.

M4160 is set as soon as a pallet table is selected. Via the PLC you can display the status of the pallet changer graphically in the PLC window. The user is able to control the changer using PLC soft keys. Sections from the pallet table can also be made available to the user for editing (see Section "Freely Defined Tables"). Module 9035 can be used to interrogate the current line of the pallet table, and Module 9090 to select a particular line in the table.

Example:

NR	PAL/PGM	NAME	DATUM	X	Y	Z
0	PAL	120	DATUM1	0	0	0
1	PGM	PART1.H				
2	PAL	130		10	20	30
3	PGM	PART21.H	DATUM2	5	3	1
4	PGM	PART22.I		100	100	100

Line 0:
The pallet with name "120" is defined. The NC macro for changing the pallet is activated. The datum table "DATUM1" is also active. The current datum corresponds with the machine datum.

Line 1:
The NC program "PART1.H" is called. The datum table "DATUM1" is still active. The current datum still corresponds with the machine datum.

Line 2:
The pallet with name "130" is defined. The NC macro for changing the pallet is activated. The datum table "DATUM1" is still active. The current datum is displaced from the machine datum by the given values.

Line 3:
The NC program "PART22.I" is called. The datum table "DATUM2" is now active. The current datum is displaced from the pallet datum by the given values. The pallet datum was defined in Line 2.

Line 4:
The NC program "PART21.H" is called. The datum table "DATUM2" is still active. The current datum is displaced from the pallet datum by the given values. The pallet datum was defined in Line 2.

		Set	Reset
M4160	Pallet table selected	NC	NC

MP7683 Executing pallet tables
Input: %xxx
Bit 0 Operating mode "Program Run/Single Block"
0: One line of the NC program is run at every start. The pallet change macro is completely executed
1: A complete NC program is run at every start
Bit 1 Operating mode "Program Run/Full Sequence"
0: A complete NC program is run at every start
1: All of the NC programs up to the next pallet are run at every start
Bit 2 Operating mode "Program Run/Full Sequence"
0: As defined in Bit 1
1: All of the NC programs and pallets to the end of the table are run

Module 9090: Selecting a line in the pallet table
With Module 9090 you can set the cursor on a particular line of the pallet table you selected in the Program Run operating mode. If the TNC is currently running in another mode, e.g. Manual, the line will only be selected once the Full Sequence mode is once again active. If the pallet table has already been started, no selection is possible.

Call in submit job only:

```
PS   B/W/D/K    <Line number in pallet table>
CM   9090
PL   B/W/D      <Error code>    0:   No error. Line was selected
                                1:   Call was not from a submit job
                                2:   Program was already started before call
                                3:   Pallet table not selected in "Program run, full sequence"
                                4:   Line does not exist
```

NC macro for pallet change
Using the entry PALETT = in the NCMACRO.SYS file, you can define the complete path indication for the NC macro that is called when a pallet entry (PAL) is executed. In this NC macro you can interrogate the current line and the name of the pallet using FN18.

End of program run
When the program reaches its end in the "Program run, single block" or "Program run, full sequence" modes, the NC sets M4170. The marker is not reset until the next program start. The information in this marker can be useful, for example, when you are working with a pallet changer.

		Set	Reset
M4170	END-PGM, M02 or M30 was executed	NC	NC

NC software: 280470..:

In the pallet tables you can assign NC programs and datum tables to specific workpieces. Each table can contain up to 255 assignments. You can define the size of the pallet table in MP7226.0.

With Module 9090 you select the desired assignment by entering the line number. The next NC START then runs the selected program with the assigned datum table. Module 9090 must be called in a submit job and the desired pallet table must have the status M. The pallet table will be in the M status if you select it with PGM MGT in the "Program run, full sequence" mode.

If no datum table is entered in the pallet table, the previous file is retained. Files that you have locked in MP7224.0 will be ignored in the pallet table. If you have locked HEIDENHAIN programs or ISO programs and you select a line with such a program, the error message **`NC PROGRAM NOT FOUND`** *will appear.*

Call only in the submit job:

```
PS     B/W/D/K        <Line number in pallet table>
CM     9090
PL     B/W/D          <Error condition>
                      0: Files were selected
                      1: Call was not from a submit job
                      2: Call during running program
                      3: Keyboard evaluator not in the default state
                      4: Pallet table not found
                      5: Line does not exist in pallet table
                      6: Incorrect type for NC program, or point is missing
                      7: NC program not found
                      8: Ambiguous NC program name
                      9: Datum table not found
                      12: Incorrect file name extension
```

MP7226.0 Size of the pallet tables
Input: 0 to 255 [lines]

4.13.18 Freely Defined Tables

As of NC software 280 472 01

You can define tables to suit your own specific applications. The names and numbers of your fields are defined in prototypes. If you wish to edit these entries, they can be interrogated and changed via special modules. Freely defined tables always have the file name extension TAB (except pallet tables and cutting data tables).

Creating the prototype

- Set up a table with the file extension TAB in the PLC:\PROTO directory.
- If you have defined several prototypes with the extension TAB, you will have to choose your desired format from a proposed menu when you are creating your table. Press the soft key NEW FORMAT to define your new prototype. If you wish to edit an existing prototype, you can do so using the soft key EDIT FORMAT, after the prototype has been selected.
- You can display the structure commands for the individual columns by pressing the following soft keys.
  `NAME:` Title of the column. This can have a max. of 8 characters and cannot be longer than WIDTH. Spaces are not permitted.
  `TYPE:` N = numerical input / C = alphanumeric input
  `WIDTH:` Width of the column. For TYPE = N this includes algebraic sign, decimal point and decimal places.
  `DEC:` Number of decimal places. For TYPE = C this has no meaning.
  `ENGLISH` to `HUNGARIA:` Language-dependent dialogs that are displayed in the dialog line when the column is being edited. A max. of 32 characters are permitted per language. A dialog does not have to be entered for every language.
- Enter your desired structure commands. If you wish to add further columns it is easiest to use the INSERT LINE soft key.
- Press END to exit the structure definition display. The table you have just created is displayed.

Data transfer

When a freely defined table (file extensions TAB, P or CDT) is being transferred via data interface, the defined structure format is stored between the lines #STRUCTBEGIN and #STRUCTEND in the external storage device. The contents of the table follow the line #STRUCTEND.

Reading and changing table fields in the PLC

You can read and overwrite individual fields of your table using special modules in the PLC. You can also make various sections available to the user for editing in the different operating modes.

Module 9245/9255: Reading a field out of a table

This module must be called up in a submit job. The table (file extension TAB or P) must already have been opened with Module 9040 (not in "buffered" mode). The field to be read is defined via column name and line number. Pay attention to small/capital letters for the column name. If an error occurs the result is no longer defined. Module 9245 reads the contents as a string, and Module 9255 as an integer. Module 9255 can only be applied to fields containing numerical values. Values with decimal places are written without the decimal point.

Call only in the submit job:

```
PS   D           <File handle>                    [from Module 9240]
PS   B/W/D/K     <Line>                           [0 to 65 535]
PS   B/W/D/K/S   <String no. of column name>      [0 to 3]
PS   B/W/D/K/S   <String no. of result>           [0 to 3]
CM   9245

PS   D           <File handle>                    [from Module 9240]
PS   B/W/D/K     <Line>                           [0 to 65 535]
PS   B/W/D/K/S   <String no. of column name>      [0 to 3]
CM   9255
PL   B/W/D       <Result>
```

Errors:	M4203 = 0:	Field was read
	M4203 = 1:	Error code in W1022
	W1022 = 1:	Line in table does not exist
	W1022 = 2:	"File handle" not correct or table opened in "buffered" mode
	W1022 = 3:	Incorrect string number
	W1022 = 7:	Could not read from table
	W1022 = 20:	Module was not called in submit job
	W1022 = 29:	The file opened is not a table (file extension TAB, P)
	W1022 = 30:	Name of column not found

Module 9246/9256: Writing a field in a table

This module must be called up in a submit job. The table (file extension TAB or P) must already have been opened with Module 9040 (not in "buffered" mode). The field defined via column name and line number is overwritten. Pay attention to small/capital letters for the column name.

Module 9246 writes a string and Module 9256 an integer. Module 9256 can only be applied to fields containing numerical values. Values with decimal places are written without the decimal point.

Call only in the submit job:

```
PS   D          <File handle>                         [from Module 9240]
PS   B/W/D/K    <Line>                                [0 to 65 535]
PS   B/W/D/K/S  <String no. of column name>           [0 to 3]
PS   B/W/D/K/S  <String no. of contents to be written> [0 to 3]
CM   9246

PS   D          <File handle>                         [from Module 9240]
PS   B/W/D/K    <Line>                                [0 to 65 535]
PS   B/W/D/K/S  <String no. of column name>           [0 to 3]
PS   B/W/D/K    <Numerical value to be written>
CM   9256
```

Errors:	M4203 = 0:	Field was overwritten
	M4203 = 1:	Error code in W1022
	W1022 = 1:	Line in table does not exist
	W1022 = 2:	"File handle" not correct or table opened in "buffered" mode
	W1022 = 3:	Incorrect string number
	W1022 = 6:	Table is write-protected
	W1022 = 7:	No numerical-value field (Module 9256)
	W1022 = 11:	The transferred value cannot be stored in the target field (incorrect format)
	W1022 = 20:	Module was not called in submit job
	W1022 = 29:	The file opened is not a table (file extension TAB)
	W1022 = 30:	Name of column not found

Module 9247: Looking for a condition in a table

This module must be called up in a submit job. The table (file extension TAB or P) must already have been opened with Module 9040 (not in "buffered" mode). You are looking for a field entry which meets one specific or several conditions. The conditions are formulated using SQL (System Query Language) data bank language. Pay attention to small/capital letters for the column name. If you indicate a starting line, you can search for several suitable field entries.

Permitted SQL commands:

• +, –, *, /	Arithmetical operators
• NOT, AND, OR,	Logical operators
• <, >, <=, >=, ==, <>	Comparisons
• LIKE’abc’	Text comparisons
• LIKE’*abc*’	Rapid string
• ( )	Brackets
• MIN(column name)	Minimum value from column
• MAX(column name)	Maximum value from column

Example: Search in pallet table for the line containing NC program 1.H and the set datum
X= –10.
String contents: (PAL/PGM LIKE’PGM’) AND (NAME LIKE’1.H’) AND (X==–10)

Call only in the submit job:

```
PS    D          <File handle>                             [from Module 9240]
PS    B/W/D/K    <Start line>                              [0 to 65 535]
PS    B/W/D/K/S  <String no. of condition>                 [0 to 3]
CM    9247
PS    B/W/D      <Line which fulfills the condition>
                 –1:     Error recognition in W1022
                 W1022 = 1:     Starting line in the table does not exist
                 W1022 = 2:     "File handle" not correct or table opened in "buffered" mode
                 W1022 = 3:     Incorrect string number
                 W1022 = 7:     Could not read from table
                 W1022 = 20:    Module was not called in submit job
                 W1022 = 31:    Syntax error in condition
                 W1022 = 32:    No suitable field entry found
```

Module 9250: Start PLC editor for tables

This module must be called up in a submit job. It starts a table editor in the different operating modes (as with tool table). Set the lines and columns that are to be displayed. Only tables (file extension TAB or P) can be edited in the PLC. A temporary file is created with the name SYS:\TEMP\PLCTABED.TAB, which you can check with the modules 9240, 9241, 9245 and 9247 before the changed data is stored again in the original file with Module 9251.

Enter the columns to be edited in the order you wish them to appear on the screen. Don't give the line numbers for the display as these always appears automatically. The different column names must be separated by a space. If you input an empty string, all of the columns of the original table are displayed.

Enter the first and last line that are to be displayed on the screen. Line numbering begins with zero. If you enter = –1 as the last line, the display will continue to the end of the table. Only when all lines and columns have been released for editing can you select whether lines can be deleted and inserted, and whether editing should be carried out directly in the original table. If the editing is done directly in the original table, the changes can no longer be reversed with Module 9251. If you do not edit in the original table and the PLC program is retranslated while the editor is opened, the editor is closed without the changes being saved. If you press the END soft key while the editor is open, the NC sets M4159. The PLC editor is not closed by the NC, but must be closed by the PLC with Module 9251. M4159 is reset when Module 9250 is called. You can activate the current active line in the PLC editor with Module 9035.

		Set	Reset
M4159	PLC editor: END button or END soft key activated	NC	NC, PLC

Call only in the submit job:

```
PS   B/W/D/K/S  <String/table name>                   [complete path and name]
PS   B/W/D/K/S  <String/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:   Inserting and deleting lines allowed
                             (only when all lines and columns are selected)
                Bit 1 = 1:   Edit the original file
                             (only when all lines and columns are selected)
                Bit 2 = 0:   Representation as table
                Bit 2 = 1:   Representation as formula
CM   9250
Error recognition: M4203 = 0:    Editor was opened
                   M4203 = 1:    Error code in W1022
                   W1022 = 1:    First and last line do not make realistic range, or mode value
                                 incorrect
                   W1022 = 3:    Incorrect string number
                   W1022 = 7:    Could not read from table, or temporary file could not be
                                 opened
                   W1022 = 20:   Module was not called in submit job
                   W1022 = 28:   PLC editor already open for another table
                   W1022 = 29:   The file open is not a table (file extension TAB, P)
                   W1022 = 30:   Column name not found
```

Module 9251: End PLC editor

This module must be called in a submit job. The PLC editor is ended and you must decide whether you wish the changes to be stored in the original file. The changed values are not automatically checked, but before calling up Module 9251 you can read the temporary file into the PLC and check it.

```
Call only in the submit job:
PS   B/W/D/K     <Mode>
                 0:      Do not store changes in original file
                 1:      Store changes in original file
CM   9251
Error recognition: M4203 = 0:     Editor was opened
                 M4203 = 1:      Error code in W1022
                 W1022 = 3:      Incorrect mode value
                 W1022 = 6:      The changes could not be stored in the original table (cause:
                                 see Module 9249)
                 W1022 = 20:     Module was not called in submit job
                 W1022 = 28:     PLC editor was not opened with Module 9250
```

Module 9252: Position the cursor in the PLC editor

This module must be called in a submit job. With this module you can position the input field of the PLC editor on a particular line and column. The designated line is relative to the starting line from Module 9250. The designated column must be defined in Module 9250.

```
Call only in the submit job:
PS   B/W/D/K/S   <String/column name>
PS   B/W/D/K     <Line>
CM   9252
Error recognition: M4203 = 0:     Cursor was set
                 M4203 = 1:      Error code in W1022
                 W1022 = 1:      Incorrect line number
                 W1022 = 3:      Incorrect string number
                 W1022 = 20:     Module was not called in submit job
                 W1022 = 30:     Incorrect column name
                 W1022 = 35:     PLC editor is not open (Module 9250)
```

4.13.19 PLC Files

It is possible to create PLC files using Modules. You can then write in and read from these files. PLC files are ASCII files. You might use these files to store PLC-specific data.

Module 9240: Opening a file

This module must be called up in a submit job. Up to eight files can be opened at the same time. Each file can be accessed only in the process during which it was called up (submit job). To prevent the file being opened in several processes at the same time, use the mode "inhibit file". For reading and writing in PLC files it is best to use the mode "buffered", for reasons of speed. In this mode part of the file is stored in the clipboard. This mode cannot however be used with tables. When the process is ended (EM in the submit job) all of the files open in the process are closed.

After a file has been opened, Module 9240 sets a "file handle". The "file handle" is a designation number that can be used to select this file again in other modules.

Call only in the submit job:

```
PS    B/W/D/K     <Mode>
                  Bit 0 = 0:    Read only
                  Bit 0 = 1:    Read and write
                  Bit 1 = 0:    Do not inhibit file
                  Bit 1 = 1:    Inhibit file
                  Bit 2 = 0:    Record-oriented (tables)
                  Bit 2 = 1:    Buffered (ASCII files)
PS    B/W/D/K/S   <String for 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: W1022 = 1:       Incorrect mode
                  W1022 = 3:        Incorrect string number
                  W1022 = 7:        File cannot be opened (Module 9249)
                  W1022 = 20:       Module was not called in a submit job
```

Module 9241: Closing a file

This module must be called in a submit job. The module is used to close a file which was opened with Module 9240. The file must be closed in the same process (submit job) as it was opened in.

Call only in the submit job:

```
PS    D           <File handle>           [Number from Module 9240]
CM    9241
Error recognition: M4203 = 0:       File was closed
                  M4203 = 1:        Error code in W1022
                  W1022 = 2:        Incorrect file handle
                  W1022 = 20:       Module was not called in a submit job
```

Module 9242: Positioning within a file

This module must be called in a submit job. With this module you can change the position of the writing/reading pointer in a file opened with Module 9240. The new position is sent as a result of Module 9242. If you opened the file in the "record-oriented" mode (tables), positioning is line by line. If you opened the file in "buffered" mode (ASCII files), positioning is character by character. If you indicate a position before the beginning or after the end of the file, the pointer is positioned on the first or last character respectively. The new position is referenced to either the beginning of the file, the current position, or the end of the file. You can interrogate the current position by entering zero.

```
Call only in the submit job:
PS   D                  <File handle>                      [number from Module 9240]
PS   B/W/D/K            <Desired position>
PS   B/W/D/K            <Mode>
                        0:   "Desired position" relative to beginning of file
                        1    "Desired position" relative to current position
                        2    "Desired position" relative to end of file
CM   9242
PL   B/W/D/K            <New position>
                        –1:  Error code in W1022
Error recognition: W1022 = 1:     Incorrect mode
                   W1022 = 2:     Incorrect file handle
                   W1022 = 7:     File system error (Module 9249)
                   W1022 = 20:    Module not called in submit job
```

Module 9243: Line-by-line reading from a file

Use Module 9245 to read tables.

Module 9243 must be called in a submit job. This module is used to read line-by-line from an ASCII file. This file must already have been opened with Module 9240. If you open the file in "buffered" mode, access times are quicker. The result is stored in a string. The module reads to the next line break (LF), or a maximum of 126 characters.

```
Call only in the submit job:
PS   D                  <File handle>                      [number from 9240]
PS   B/W/D/K/S          <String no. of result>             [0 to 3]
CM   9243
PL   B/W/D              <Number of bytes read>
                        >0:  Line read
                        0:   File end reached
                        –1:  Error code in W1022
Error recognition: W1022 = 2:     Incorrect file handle
                   W1022 = 3:     Incorrect string number
                   W1022 = 7:     File system error (Module 9249)
                   W1022 = 20:    Module was not called in submit job
```

Module 9244: Line-by-line writing in a file

Use Module 9246 to write in tables.

Module 9244 must be called in a submit job. This module is used to write line-by-line in an ASCII file. You must already have opened the file with Module 9240.

File opened in "buffered" mode:

- Short processing time
- The data is only stored on the hard disk if more than 512 bytes are overwritten in several calls, or if the file is closed.
- Exactly that number of data is overwritten that is defined in the transfer string.

File opened in record-oriented mode:

- Long processing time
- The data is stored on the hard disk immediately.
- Exactly one line is overwritten. If there is a difference in length, the following data is displaced by the difference.

Call only in the submit job:

```
PS   D                  <File handle>                 [number from Module 9240]
PS   B/W/D/K/S          <String no. of source data>   [0 to 3]
CM   9244
PL   B/W/D              <Number of bytes written (including LF)>
                        –1:   Error code in W1022
Error recognition: W1022 = 2:     Incorrect file handle
                   W1022 = 3:     Incorrect string number
                   W1022 = 7:     File system error (Module 9249)
                   W1022 = 20:    Module was not called in submit job
```

4.13.20 User Parameters

You can provide the machine tool operator with easy access to up to 16 machine parameters. He can then call them through the MOD function by simply pressing the USER PARAMETER soft key. In MP7330.x you define which parameters are to be user parameters. If, for example, you wish to define MP7230.1 as the first user parameter, you must enter 7230.01 in MP7330.0.

When the user selects a user parameter, a dialog appears on the screen. In MP7340.x you define that dialog by assigning a line number 0 to 4095 from the PLC dialog file. You must have identified the name of the PLC dialog file with the command **PLCDIALOG=** in the system file OEM.SYS.

MP7330.0-15 Definition of user parameters
Input: 0 to 9999.00 (no. of the desired machine parameter)

MP7340.0-15 Dialogs for user parameters
Input: 0 to 4095 (line number in the PLC dialog file)

4.13.21 Code Numbers

You can enter predefined code numbers in the MOD functions in order to activate certain functions. The following code numbers have a fixed meaning:

Code number	Function
95 148	Machine parameter list
807 667	PLC modes
75 368	Automatic offset adjustment
123	Machine parameters accessible to the user
531 210	Erase M0 to M999 and B0 to B127
688 379	Oscilloscope
555 343	FN17: Overwrite system data, FN25: Overwrite datum
NET123	Ethernet settings (option)

The entered code number is registered in doubleword D276. You can evaluate this code and define your own functions for code numbers, or disable the code numbers with fixed meanings.

		Set	Reset
D276	Code number last entered via MOD	NC	NC

4.13.22 Programming Station

With MP7210 you can set the control for use as a programming station without a machine. In a programming station only the "Programming and editing" and "Test run" operating modes function. You can also select whether the PLC should be active.

MP7210 Programming station
Input: 0 = controlling and programming
1 = programming station "PLC active"
2 = programming station "PLC not active"

4.13.23 Conversational Language

The TNC is delivered with all eleven NC dialog human languages. With MP7230.0 you select the language in which the operator will work. If the selected language is not located on the hard disk, the error message **LANGUAGE LOAD ERROR** appears. You can then still work in the basic language English.

You can also save the dialogs that you write in several languages and select the desired language with MP7230.1-3. For this purpose your dialogs must be stored in permanently assigned language directories in the PLC partition. These directories are:

```
PLC:\LANGUAGE\ CZECH\
               DANISH\
               DUTCH\
               ENGLISH\
               FINNISH\
               FRENCH\
               GERMAN\
               ITALIAN\
               POLISH\
               PORTUGUE\
               SPANISH\
               SWEDISH\
               HUNGARIAN\
```

This makes it possible for you to store PLC dialog files, PLC error files, and help files with the identical file names in the different languages.

In the system file OEM.SYS you enter only the file names after the commands **PLCDIALOG=** and **PLCERROR=**. The NC looks for the files in the paths given in MP7230.1 and MP7230.2, respectively. The entry **MODEHELP=** is overwritten with the selected path whenever MP7230.3 is changed (see above).

MP7230	Changing the dialog 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

MP7230.0	NC dialog language
MP7230.1	PLC dialog language (user parameters), soft keys for OEM cycles
MP7230.2	PLC Error messages
MP7230.3	Help files

Decimal sign
With MP7280 you define whether to use a comma or point as decimal sign.

MP7280	Decimal sign
Input:	0 = decimal comma
	1 = decimal point

4.13.24 Memory Test

With MP7690 you select the data storage media to be tested during switch-on. The message MEMORY TEST indicates that the test is in progress.

MP7690	MEMORY TEST during power-on	
Input:	%xxx	
Bit 0	Test the RAM	0 = MEMORY TEST during power-on
Bit 1	Test the EPROM	1 = no MEMORY TEST during power-on
Bit 2	Test the hard disk	

4.13.25 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. If the tolerance defined in MP7431 is exceeded, the error message `CIRCLE END POS. INCORRECT` will appear.

MP7431	Arc end-point tolerance
Input:	0.0001 to 0.016 [mm]

4.13.26 Radius Compensation R+, R-

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 one after another. To avoid erroneous entries, MP7246 can be used to disable the input of paraxial positioning blocks.

MP7246	Disable paraxial positioning blocks	
Input:	0 =	enable paraxial positioning blocks
	1 =	disable paraxial positioning blocks

4.13.27 Power Interrupted Message

When the control voltage is disconnected the TNC issues the message **POWER INTERRUPTED**. The PLC will not become active until you acknowledge the message by pressing the CE key. With MP7212 you can suppress this message if it is not needed, for example for unattended operation.

MP7212 Power interrupted message
Input: 0 = `Power interrupted` message must be acknowledged with the CE key
1 = `Power interrupted` message does not appear

4.13.28 Operating Times

The TNC can measure up to eleven operating times and store them in a file in the SYS partition:

TNCTIME	Control on
MACHINETIME	Machine on
PROGTIME	Program run
PLCTIME0 to PLCTIME7	Freely defined times of the PLC

The operating times are displayed in the MOD function "machine time." With MP7237 you identify which times you can reset with the code number 857282 and which PLC times you wish to display. You define the dialogs which shall be displayed for the individual PLC operating times in MP7238.x. The time is measured in seconds. During measurement the time information is updated every minute so that if the control is switched off no more that one minute will be lost. The NC measures the time for TNCTIME, MACHINETIME and PROGTIME. You start measuring the operating times for PLCTIME0 to PLCTIME7 with Module 9190, and stop it with Module 9191. Except for TNCTIME, all operating times are saved during a hard-disk backup with the program TNCBACK (see section "Software Exchange").

You can evaluate and change the operating times with the following modules:

- Module 9190: Start operating times
- Module 9191: Stop operating times
- Module 9192: Read operating times
- Module 9193: Set operating times
- Module 9194: Alarm when operating times exceeded

MP7237.0-1 Display and reset the operating times
Input: %xxxxxxxx

MP7237.0 Display PLC operating times

Bit	0 to 7	0 = do not display
PLC operating times	1 to 8	1 = display

MP7237.1 Reset the PLC operating times code number 857282

Bit	0 to 7	0 = do not reset
PLC operating time	1 to 8	1 = reset

MP7237.2	Reset the NC operating times with code number 857282	
Bit 0	Without function	0 = do not reset
Bit 1	MACHINE ON time	1 = reset
Bit 2	PROGRAM RUN time	

MP7238.0-7 Dialogs for PLC operating times
Input: 0 to 4095 [Dialog no. from the PLCDIALOG= (OEM.SYS) file]

Module 9190: Start operating times

Here you start one or more of the PLC operating times.

Call also in the submit job:

```
PS   B/W/D/K <PLC operating time>      Bits 0 to 7 = PLC operating time 1 to 8
CM   9190
```

Module 9191: Stop operating times

Here you stop one or more of the PLC operating times.

Call also in the submit job:

```
PS   B/W/D/K <PLC operating time>      Bits 0 to 7 = PLC operating time 1 to 8
CM   9191
```

Module 9192: Read operating times

Here you read the current value of one of the operating times. The current value in seconds is transferred. If the value is greater than 2 147 483 648 (approx. 69 years), a negative number is transferred.

Call only in the submit job:

```
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
```

Module 9193: Set operating times

You overwrite the current operating time value. The old value is irretrievably lost. The control-on time (TNCTIME) cannot be overwritten. You must transfer values greater than 2 147 483 648 (approx. 69 years) as a negative number.

Call only in the submit job:

```
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
```

Module 9194: Alarm when operating time is exceeded

You define a marker that is set when a certain operating time limit (alarm threshold) is exceeded. After the limit is exceeded the defined marker is set cyclically once every minute. Therefore the delay before the marker is set for the first time cannot exceed 59 seconds. You must transfer values greater than 2 147 483 648 (approx. 69 years) as a negative number. If you enter the value zero as alarm threshold, the function will be deactivated.

Call only in the submit job:

```
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 marker >
CM   9194
```

System time

The TNC operates with UNIX system time. This system time contains the number of seconds accumulated since 0:00 hours of January 1, 1970. When the TNC is shipped it is calibrated for Universal Time (also known as Greenwich Mean Time). To ensure that the time data in your program management reflects your local time, you must enter in MP7235 the difference between local time and Universal 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 reflects Universal Time. With Module 9055 you can convert the value read in Module 9195 into a readable ASCII format. Module 9055 compensates the transferred value by the difference to local time, which was entered in MP7235. The value calculated by Module 9055 therefore represents the local time.

Call:

```
CM   9195
PL   D           <System time>   [Number of seconds since Jan.1, 1970, 0.00 hours]
```

Call:

PS B/W/D/K <System time> [Number of seconds since Jan.1, 1970, 0.00 hours]
PS B/W/D/K <String number>
0 to 3
PS B/W/D/K <Format>

0:	"04.10.1996 09:16:12"
1:	"4.10.1996 9:16:12"
2:	"4.10.1996 9:16"
3:	"4.10.96 9:16"
4:	"1996-10-04 09:16:12"
5:	"1996-10-04 09:16"
6:	"1996-10-04 9:16"
7:	"96-10-04 9:16"
8:	"04.10.1996"
9:	"4.10.1996"
10:	"4.10.96"
11:	"1996-1004"
12:	"96-10-04"
13:	"09:16:12"
14:	"9:16:12"
15:	"9:16"

CM 9055

MP7235 Time difference from Universal Time (Greenwich Mean Time)
Input: –23 to +23 [hours]

–8 =	Pacific Time
–7 =	Mountain Time
–6 =	Central Time
–5 =	Eastern Standard Time
0 =	Universal Time
+1 =	Central European Time
+2 =	Eastern European Time
+3 =	Moscow Time
+6 =	Indian Time
+8 =	China Time
+9 =	Japan Time
+10 =	Eastern Australian Time

If your country has adopted daylight savings time, these example values must be adjusted correspondingly.

4.14 Keystroke Simulation

HEIDENHAIN contouring controls are operated through the keys on the keyboard unit and through the machine tool builder's own control panel. The two control panels are connected to sockets X45 and X46 of the logic unit.

The key code from the TNC keyboard is directly evaluated by the NC. PLC inputs and outputs for the machine control panel are available on socket X46. The PLC must evaluate these PLC inputs and outputs and set the appropriate markers (e.g. for traverse direction).

4.14.1 TNC Keyboard

The key code from the TNC keyboard is directly evaluated by the NC. The key code is displayed in W274 as long as a key is depressed. If you press a disabled key, marker M4577 is also set.

HEIDENHAIN provides the following modules to help you influence the effect of keys and soft keys:

- Module 9180: Simulation of NC keys
- Module 9181: Disabling individual NC keys
- Module 9182: Re-enabling individual NC keys
- Module 9183: Disabling groups of NC keys
- Module 9184: Re-enabling groups of NC keys
- Module 9186: Calling a soft-key function
- Module 9187: Status of a soft-key function call

		Set	Reset
W274	Code of the depressed key	NC	NC
M4577	Disabled key was pressed	NC	PLC

Module 9180: Simulation of NC keys
With Module 9180 you can simulate NC keys and soft keys by transferring the code of the desired key. If you transfer the code value zero, only the number of occupied elements in the keystroke queue is transferred and no key simulation is carried out.

Call:

PS	B/W/D/K	<Key code>	
CM	9180		
PL	B/W/D	<Number of the occupied elements / Error status>	
		0:	Key code was transferred, keystroke queue empty
		1 to 16 :	Key code was not yet simulated (max. 16 entries in the keystroke queue are possible)
		–1:	For error see W1022

W1022 :	1:	Transferred parameter greater than maximum value
	2:	Transferred parameter invalid
	22:	Keystroke queue overflow

Module 9181: Disabling individual NC keys
With Module 9181 you can disable individual NC keys. If you press an disabled key, the PLC sets M4577.

```
Call:
PS   B/W/D/K          <Key code>
CM   9181
PL   B/W/D            <Error status>
                      0 :   NC key was disabled
                      –1:   For error see W1022

W1022 :               1:    Transferred parameter greater than maximum value
                      2:    Transferred parameter invalid
```

Module 9182: Re-enabling individual NC keys
With Module 9182 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 was released
                      –1:   For error see W1022

W1022 :               1:    Transferred parameter greater than maximum value
                      2:    Transferred parameter invalid
```

Module 9183: Disabling groups of NC keys
With Module 9183 you can inhibit whole groups of NC keys.
The key-group codes are:

0	All keys
1	ASCII keys
2	Soft keys, page keys
3	Cursor keys, ENT, NOENT, DEL, END, GOTO
4	Numbers, algebraic sign key, Decimal point key, Actual-value-capture key
5	Operating mode keys
6	Block initiation keys

```
Call:
PS   B/W/D/K     <Key-group code>
CM   9183
PL   B/W/D       <Error status>
                 0 :      The group of NC keys was disabled
                 – 1 :    Transferred parameter greater than maximum value
```

Module 9184: Re-enabling groups of NC keys
With Module 9184 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 was released
                – 1 :   Transferred parameter greater than maximum value
```

Module 9186: Calling a soft-key function
With Module 9186 you can call certain soft-key functions of the machine operating mode. A new soft-key function must not be called until the previous function is completed. This information can be requested with Module 9187.

For a soft-key function to be simulated it must be displayed either in the foreground or background operating mode. If it is not, the module call has no effect. Module 9187 informs you of this error condition.

```
Call:
PS   B/W/D/K    <Number of the soft-key function>
                0:      INTERNAL STOP
                1:      M output
                2:      S output
                3:      TOUCH PROBE
                4:      PASS OVER REFERENCE
                5:      RESTORE POSITION
                6:      INCREMENT
CM   9186

W1022:          1:      Parameter out of range
                28:     Previous call not yet completed
```

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. This does not yet mean that the function cannot be carried out in the current operating mode. Module 9186 cannot be called again until status 0 or 2 is set. Error status 2 can be erased only by calling Module 9186 or switching power 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: Because the soft key is not available (wrong operating mode) the
                        soft-key function cannot be executed
```

Key code 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	Number
$31	1	Number
$32	2	Number
$33	3	Number
$34	4	Number
$35	5	Number
$36	6	Number
$37	7	Number
$38	8	Number
$39	9	Number
$3A	:	ASCII
$3B	;	ASCII
$3C	<	ASCII
$3D	=	ASCII
$3E	>	ASCII
$3F	?	ASCII
$41	A	ASCII
$42	B	ASCII
$45	E	ASCII
$43	C	ASCII
$44	D	ASCII
$46	F	ASCII
$47	G	ASCII
$48	H	ASCII
$49	I	ASCII
$4A	J	ASCII
$4B	K	ASCII
$4C	L	ASCII
$4D	M	ASCII
$4E	N	ASCII
$4F	O	ASCII
$50	P	ASCII
$51	Q	ASCII
$52	R	ASCII
$53	S	ASCII
$54	T	ASCII
$55	U	ASCII
$56	V	ASCII
$57	W	ASCII
$58	X	ASCII
$59	Y	ASCII
$5A	Z	ASCII
$5E	^	ASCII
$180	Soft key 0	Soft key
$181	Soft key 1	Soft key
$182	Soft key 2	Soft key
$183	Soft key 3	Soft key
$184	Soft key 4	Soft key
$185	Soft key 5	Soft key
$186	Soft key 6	Soft key
$187	Soft key 7	Soft key
$19C	FBACK	Soft key

Code	Key	Group
$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
$1BD	.	Number
$1C0	MANUAL	Operating mode
$1C1	POS MDI	Operating mode
$1C2	SINGLE	Operating mode
$1C3	AUTO	Operating mode
$1C4	EDIT	Operating mode
$1C5	HANDWHEEL	Operating mode
$1C6	TEST	Operating mode
$1C7	MOD	
$1CB	PGM MGT	
$1D0	PGM CALL	Block initiation
$1D1	TOOL DEF	Block initiation
$1D2	TOOL CALL	Block initiation
$1D3	CYCL DEF	Block initiation
$1D4	CYCL CALL	Block initiation
$1D5	LBL SET	Block initiation
$1D6	LBL CALL	Block initiation
$1D7	L	Block initiation

Code	Key	Group
$1AE	CE	
$1B0	X	
$1B1	Y	
$1B2	Z	
$1B3	IV	
$1B4	V	
$1B8	POLAR	
$1B9	INCREMENT	
$1BA	Q	
$1BB	ACT POS	Number
$1BC	–	Number
$1D8	C	Block initiation
$1D9	CR	Block initiation
$1DA	CT	Block initiation
$1DB	CC	Block initiation
$1DC	RND	Block initiation
$1DD	CHF	Block initiation
$1DE	FK	Block initiation
$1DF	TOUCH PROBE	Block initiation
$1E0	STOP	Block initiation
$1E1	APPR/DEP	Block initiation
$1EA	DIA	
$1EB	FIG	
$1EC	Switch-over key	
$1ED	HELP	
$1EE	INFO	
$1EF	CALC	
$1F0	NC START	

4.14.2 Machine Operating Panel

On socket X46 there are 25 PLC inputs (I128 To I152) and eight PLC outputs (O0 To O7) at your disposal for evaluating the keys on the machine operating panel. To activate the desired functions, link the PLC inputs with the corresponding markers and words. You can store a depressed axis-direction key with M4562 for "latched traverse." This means that the axis will keep moving until an NC STOP occurs. You must enable this memory function with MP7680 bit 0.

MP7680 Machine parameter with multiple function
Input: %xxxxxxx

Bit 0 Memory function for axis-direction keys with M4562
0 = no memory
1 = memory if M4562 is set

				Set	Reset	
W1046	Manual traverse in positive direction				PLC	PLC
	Bit	0 to 8	0 = do not move axis			
	Axis	1 to 9	1 = move axis			
W1048	Manual traverse in negative direction				PLC	PLC
	Bit	0 to 8	0 = do not move axis			
	Axis	1 to 9	1 = move axis			
M4562	Memory function for axis direction keys (MP7680 bit 0 = 1)				PLC	PLC
M4560	NC stop ("0" corresponds to stop)			PLC	PLC	
M4561	Rapid traverse			PLC	PLC	
M4564	NC START				PLC	

4.15 Touch Probe

The following touch probes can be connected:
The triggering touch probes

TS 120 or TS 220 With cable connection. Primarily for digitizing, but also for workpiece setup and workpiece measurement during machining

or

TS 630 With infrared transmission, for workpiece setup and measurement during machining.

and

TT 120 For workpiece measurement

and

One measuring touch probe.

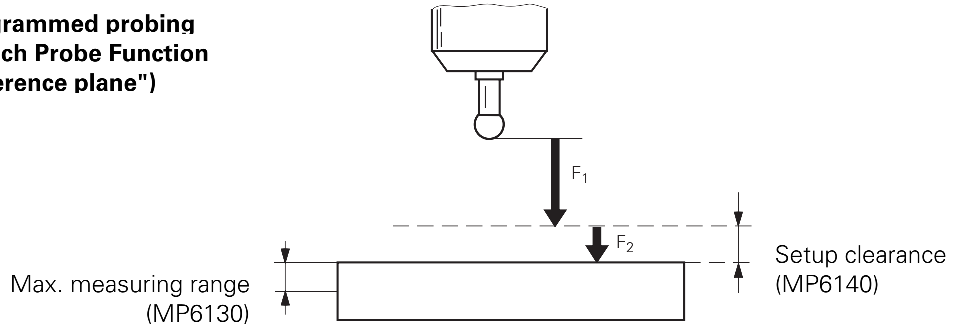
See chapter "Mounting and Electrical Installation" for information on connecting the touch probes.

With MP6010, MP6200 and MP6500 you can define which touch probe is connected. You must ensure that the spindle is locked during the measuring process. The current touch probe data can be read with FN18.

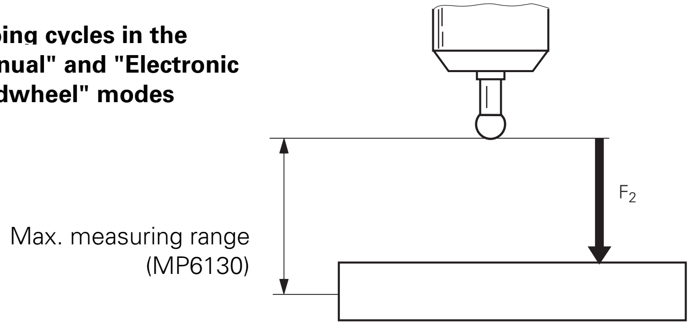
4.15.1 Standard Probing Cycles

You can use the touch probe in the "Manual" and "Electronic Handwheel" operating modes through the probing cycles and in the NC program through the "Touch Probe" function (see the User's Manual). You must set the proper machine parameters to adapt the touch probe to the measuring conditions.

Programmed probing (Touch Probe Function "reference plane")



Probing cycles in the "Manual" and "Electronic Handwheel" modes



F_1 = rapid traverse during programmed probing:	MP6150 for triggering touch probe MP6200 = 0 MP6361 for measuring touch probe MP6200 = 1
F_2 = probing feed rate:	MP6120 for triggering touch probe MP6200 = 0 MP6360 for measuring touch probe MP6200 = 1

If the maximum measuring range (MP6130) is exceeded, the error message `TOUCH POINT INACCESSIBLE` appears. For the probing cycles in the "Manual" and "Electronic Handwheel" modes, MP6140, MP6150 and MP6361 have no function.

M4055 is set by the NC before the probing process is started. The NC then waits until you reset M4055 before it performs the probe function. This allows you, for example, to clean the measured object with compressed air before probing. The probing function is controlled entirely from the NC.

M4050 to M4054 inform you of certain conditions and allows you to process this information if desired. If you set M4056, the NC stops the machine in all operating modes as soon as the stylus is deflected. Also, the maximum feed rate is limited to the value in MP6150 or MP6361. If M4056 is not set, the control recognizes stylus deflection only if the probing function has been started. HEIDENHAIN therefore recommends setting M4056 as soon as the touch probe is in the spindle.

With MP7411 you select whether the tool data (length, radius, axis) used in a touch probe block are taken from the last TOOL CALL block or from the calibrated data of the probe system. When MP7411 = 1 is set, you can use soft keys to have the effective length and the effective radius taken over by the tool table. Enter the tool number in the menu for touch probe calibration.

The TNC can store the calibration data for up to three touch probes at the same time. The current data can be activated using M4574/M4575. For three sets of calibration data to be stored at the same time, bit 2 must be set in MP7490.

Programmed probing (TOUCH PROBE function "reference plane")

The touch probe behavior can be set with FN17:SYSWRITE ID990 NR1. With input value = 0 the setup clearance form MP6140 and the effective radius are used. With input value > 0 the effective radius and setup clearance are assumed to be zero. This function might be used for the measurement of small holes.

Center misalignment

You can find the center offset of the probe system while calibrating. This center offset is then automatically compensated in all probing operations (see User's Manual). With MP6160 you define whether spindle orientation for a rotation by 180° will be performed directly by the NC or through the PLC. If the NC orients the spindle directly, you need only reset M4012 (see the section "Spindle Orientation"). The PLC orients the spindle, you must enter the number of the M function in MP6160. The respective position is then transferred as in the "spindle orientation" cycle.

For the triggering touch probe, the rotation is activated by pressing a soft key. For the measuring touch probe, the rotation is automatically activated during measurement. You can deselect this with MP6321. During every spindle orientation the Marker M4017 is set.

Special case: tilted axes:
The actual position of the encoder for spindle position can vary with tilted axes. This depends on the construction of the machine. Since the TNC refers to the actual position of the spindle when compensating the center offset, it would be necessary to recalibrate the touch probe for each new tilt in position. If however you enter the current offset to the initial position in D760, then calibration is only necessary in the initial position. The TNC then takes this offset value into consideration when compensating the center misalignment. D760 must be = 0 in the initial position.

Measuring tolerance
During programmed probing with the TOUCH PROBE function you can define a measuring tolerance via machine parameters. In MP6170 enter the number of measurements that are to be carried out with each probe function. The average value is calculated from these measurements and this is recorded. If the results of each individual measurement differ by more than the value from MP6171, an error message is output. This function can be used to identify if a measurement result is being influenced by dirt or chips.

MP6010 Selecting the touch probe
Input: 0 = touch probe with cable transmission
1 = touch probe with infrared transmission

MP6200 Choosing between 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

MP6120 Probing feed rate (triggering touch probe)
Input: 10 to 3000 [mm/min]

MP6360 Probing feed rate (measuring touch probe)
Input: 10 to 3000 [mm/min]

MP6130 Maximum measuring range
Input: 0.001 to 99 999.9999 [mm]

MP6140 Setup clearance above measuring point
Input: 0.001 to 99 999.9999 [mm]

MP6150 Rapid traverse in probing cycle (triggering touch probe)
Input: 10 to 10 000 [mm/min]

MP6361 Rapid traverse in the probing cycle (measuring touch probe)
Input: 10 to 10 000 [mm/min]

MP6160 M function for probing from opposite orientations
Input: –1 = Spindle orientation directly through the NC
0 = Function inactive
1 to 88 = Number of the M function for spindle orientation through the PLC

MP6321 Finding the center offset while calibrating the measuring touch probe
Input: 0 = calibration with center offset determination
1 = calibration without center offset determination

MP6170 Number of measurements in a programmed measurement (touch probe block)

Input: 1 to 3

MP6171 Confidence range in a programmed measurement (MP6170 > 1)
Input: 0.002 to 0.999 [mm]

MP7411 Tool data in touch probe block
Input: 0 = the calibrated touch probe data is used
1 = the current data from the last tool call is used

MP7490 Functions for traverse ranges
Input: %xxxx
Bit 2 Calibration data: touch probe for workpiece measurement
0 = one set of calibration data for all traverse ranges
1 = one set of calibration for each individual traverse range

		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 deflected (probing process not executed)	NC	PLC
M4053	Probing sequence ended or interrupted	NC	NC
M4054	Battery voltage too low (battery warning at touch probe connection); evaluated only during the probing process)	NC	NC
M4055	Enabling the probing process	NC	PLC
M4056	NC stop in all operating modes if stylus is deflected	PLC	PLC

D760 Offset for tilted axes (touch probe center misalignment)
[1/10 000 °]

4.15.2 Logging Probe Measurements

For every manual scanning cycle there is one print mask for every language. Standard print masks for each language are already stored on the hard disk when the control is shipped from the factory.

With the print masks, the output format of the measurement results is defined in the **%TCHPRNT.A** file. You define the path for the %TCHPRNT.A file in the MOD menu "RS232/RS422 Setup" in the PRINT line. If the path name begins with RS232:\ or RS422:\ the measurement results will be transmitted through the data interface. If no path is entered here, the file is stored in the root directory TNC:\.

To store the measured data in the %TCHPRNT.A file, press the PRINT soft key in the manual probing cycle. If you do not wish to use the standard print mask you can create your own. You can then save these masks in the language paths in the PLC partition. See the "Dialog Language" section.

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 measurement:	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 print masks

Print masks consists of text lines that are set in quotation marks and concluded with a semicolon. The text lines can contain format instructions, as are known from the programming language "C." Format instructions are fed from variables set after the text string and separated by commas. The special commands MM and INCH switch the display to millimeters or inches. The commands are effective only on types of numbers that permit representation in inches.

Variable names:

Time management

Name	Format type	Description
HOUR	Int	Number of hours from the real-time clock
MIN	Int	Number of minutes from the real-time clock
SEC	Int	Number of seconds from the real-time clock
DAY	Int	Day from the real-time clock
MONTH	Int	Month as number from the real-time clock
STR_MONTH	String	Month as string abbreviation from the real-time clock
YEAR2	Int	Two-digit year from the real-time clock
YEAR4	Int	Four-digit year from the real-time clock

Results/input from the manual measuring cycles in the control data

Name	Format type	Description
TCH.AXIS	String	Selected touch 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 of main axis of the trig. touch probe
TS.OFF2	Double	Calibrated center offset of aux. axis of trig. 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 of main axis of meas. touch probe
TM.OFF2	Double	Calibrated center offset of aux. axis of meas. touch probe
TM.CORSTAT[0]	Double	Calibrated stylus bending in X axis of the meas. touch probe
TM.CORSTAT[1]	Double	Calibrated stylus bending in Y axis of the meas. touch probe
TM.CORSTAT[2]	Double	Calibrated bending of the Z axis of the meas. touch probe
TM.CORDYN[0]	Double	Power ratio X/Z axis of measuring touch probe
TM.CORDYN[1]	Double	Power ratio Y/Z axis of measuring touch probe

Results/input from the manual measuring cycles
Setting the datum:

BZ	Double	Datum
BEZA	String	String datum axis

Datum at corner, circle, 4 holes, 3 holes on a circle

BZ_HA	Double	Datum of main axis
BZ_NA	Double	Datum of auxiliary axis
LKALBEZ	Double	Datum entered with calibrated probe length
HA	String	Main axis character
NA	String	Auxiliary axis character
TA	String	Probe axis character

Calculated straight lines from straight line probing

GE_HA[2]	Double	Straight-line axis section of main axis
GE_HA[2]	Double	Straight-line axis section of auxiliary axis
GE_WI[2]	Double	Straight-line angle

Calculated radii from circle probing

RAD[8]	Double	8 radii

Calculated centers from circle probing

MP_HA[8]	Double	Center in main axis 1
MP_NA[8]	Double	Center in main axis 2

Accumulated touch points from probes

AP_HA[32]	double	Touch points in main axis X
AP_NA[32]	double	Touch points in main axis Y
AP_NA[32]	double	Touch points in main axis Z

Example of a print mask

```
"Calibrate measuring touch probe:";
"----------------------";
"%02.2d-%02.2d-%4d, %2d:%02.2d:%02.2d Uhr",DAY,MONTH,YEAR4,HOUR,MIN,SEC;
"Probe axis     : %s“,TA;
"Probe axis 1   : %4.3lf“,TM.RAD;
"Probe axis 2             : %4.3lf“,TM.RAD2;
"Ring diameter            : %4.3lf“,TM.RINGDIA;
"Factors            : X = %4.4lf“,TM.CORSTAT[0];
"                   : Y = %4.4lf“,TM.CORSTAT[1];
"                   : Z = %4.4lf“,TM.CORSTAT[2];
"Power ratio: FX/FZ = %4.4lf“,TM.CORDYN[0];
"                   :  FY/FZ = %4.4lf“,TM.CORDYN[1];
```

4.15.3 Digitizing with the Triggering Touch Probe

HEIDENHAIN recommends the TS 220 touch trigger probe for digitizing. The touch probes with infrared transmission are not suited for digitizing: they are powered by accumulator and provide no more than eight hours of continuous operation.

Technical prerequisites

- The software module for digitizing with the TS must be installed. The ID number (Id. Nr. ....) of the logic unit indicates whether the module is already installed (see the "Software Options" section). Also, if the module is already installed the message `OPT: %00000001` will appear beneath the NC and PLC software numbers when you press the MOD key.
- Interfaced TS 220 touch probe
- The machine must be optimized for operation with servo lag.

You must optimize the digitizing process by setting machine parameters.

Machine parameter MP6210 is the value for the oscillations executed by the touch probe as it scans the form. It is governed by the dynamic response of the machine. The dynamic response is in turn determined by the k_v factor (operation with servo lag). The greater the k_v factor, the greater the number of oscillations.

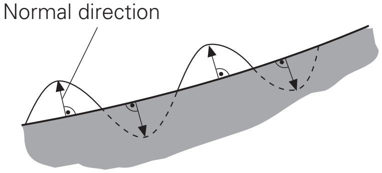
Machine parameter MP6210 determines the maximum probing feed rate in conjunction with the programmed probe point interval, PP.INT from the scanning cycles "Meander" and "Contour Lines":

F_{scan} [mm/min] = PP.INT [mm] • oscillations [1/s] • 60 [s/min]

This provides the formula for calculating the input value of MP6210, whereby the optimized scanning feed rate depends on the feed rate in the surface-normal direction (MP6230).

$$\text{MP6210 [1/s]} = \frac{\text{optimized } F_{scan} \text{ [mm/min]}}{\text{PP.INT [mm]} \bullet \text{60 [s/min]}}$$

The feed rate in normal direction (MP6230) is the velocity that results when the touch probe moves perpendicularly to the contour from the non-deflected to the deflected condition, and vice versa.



Aside from the oscillation amplitude, the feed rate in the normal direction also determines the 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 will be too high. 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.

You define the maximum stylus deflection in machine parameter MP6240. The value in MP6240 will depend on the length of the stylus used. MP6240 defines 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 out by the travel defined in MP6240, the touch probe axis (e.g. Z) is retracted in the positive direction. Digitizing continues as soon as the touch probe is clear.

If your input value is too small, the touch probe may get caught up in an endless repetitive attempt to come free of an inside corner.

Machine parameter MP6260 defines whether an M90 is appended to each NC block in the output digitized data (see also "Contouring Behavior").

Machine parameter MP6270 defines the output format of the digitized data, i.e. 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 stylus deflection
Input: 0 to 10.000 [mm]

MP6260 Output of M90 for NC blocks with digitized data
Input: 0 = no output of M90
1 = output of M90 in each NC block

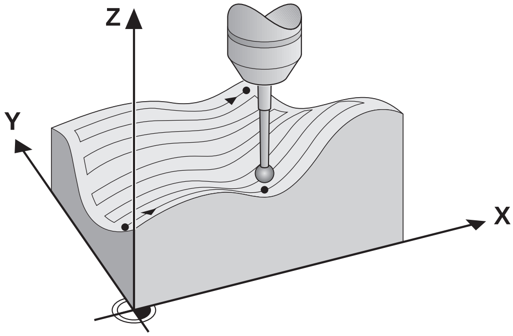
MP6270 Rounding decimal places
Input: 0 = output in 0.001-mm steps (1 µm)
1 = output in 0.01-mm steps (10 µm)
2 = output in 0.0001-mm steps (0.1 µm)

Scanning cycles

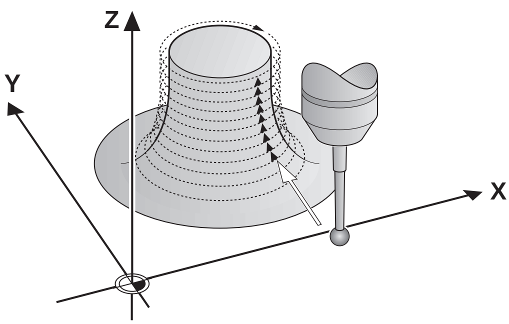
Because it has direct access to the position control loop of the TNC controller, the touch probe can record measured values very rapidly (3 to 5 values per second). With a programmed probe point interval of 1 mm, this produces a scanning feed rate of 180 to 300 mm/min. Three scanning cycles are used for digitizing: "Range," "Meander" and "Contour Lines."

The "Range" cycle defines the cuboid scanning range and the file where the digitized data are stored. You can save the digitized data in the program memory of the control or on a PC.

The "Meander" cycle digitizes a 3-D form meander-wise (line by line) in the pre-defined range.



The "Contour Lines" cycle digitizes a 3-D form level-by-level in contour lines within a pre-defined range. Level-by-level digitizing is mainly used 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 you can lubricate the axis at the end of the lines.

MP6220 Traverse distance for lubricating the probe axis at end of line
Input: 0.000 to 999.999 [mm]

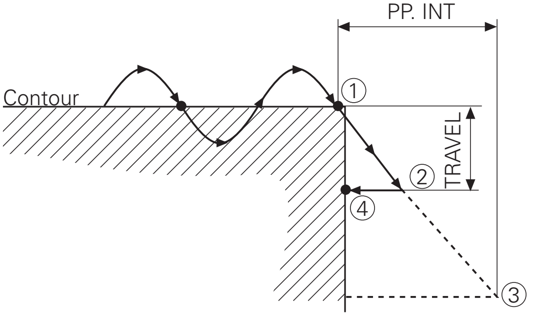
MP6221 Time after which the probe axis must be lubricated
Input: 0 to 65 535 [min]

Scanning process at corners

The scanning sequence responds differently for inside and outside corners. Here the two parameters PP.INT (maximum probe point interval) and TRAVEL from the "Meander" and "Contour Lines" scanning cycles operate like a travel limiter. Depending on the values that are entered for these parameters, either the travel or the probe point interval are limited.

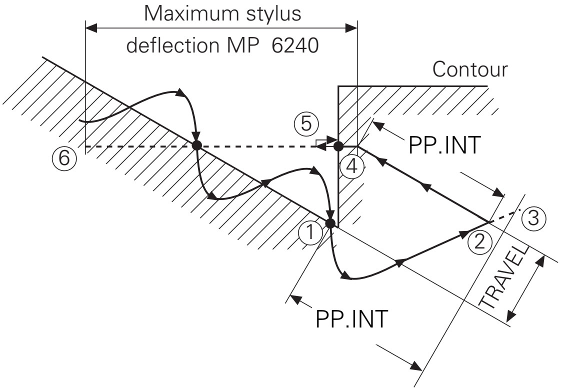
The travel is also responsible for geometrical accuracy at the corners. The smaller the stroke, the greater the accuracy of corner resolution. If too small a stroke is defined however, it may affect clearance at narrow inside corners (minimum travel = 0.1 mm).

Outside Corners



On outside corners, having probed the last point ①, the touch probe moves down the resultant straight line until it either makes workpiece contact again or hits one of the two limits TRAVEL ② or PP.INT ③. In the illustrated example TRAVEL is the limit, and the touch probe returns to the contour ④ in the inverse scanning 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 hits one of the limits TRAVEL ② or PP.INT ③.

The touch probe moves in inverse scanning direction to get clear. 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 (maximum stylus deflection). As soon as the touch probe is clear it returns to the contour ⑤ in the inverse travel direction. The new scanning direction is defined by the probed points ① and ⑤. If the touch probe has failed to clear even after it has backed out by the value of MP6240 ⑥, the touch probe axis (e.g. Z) is cleared in the positive direction. If the stylus is still deflected after it reaches the "clearance height" (see "Range" scanning cycle) the scanning sequence is aborted and an error message is displayed.

Optimizing the digitizing sequence
Preparation:

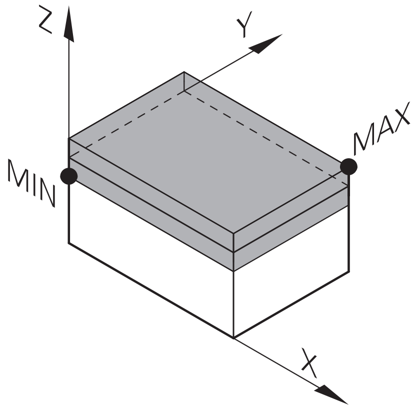
- Set up the flat workpiece with vertical face and plane surface in the machining plane (e.g. XY plane).
- Probe the surface with probing function "Surface = Datum" ("Manual" or "Electronic Handwheel" mode) and enter the reference plane as +0 mm.
- Default setting of the relevant machine parameters for digitizing:

MP6210 =5	[1/s]	Oscillations in normal direction
MP6230 =30	[mm/min]	Feed rate in normal direction
MP6240 =5	[mm]	Maximum stylus deflection

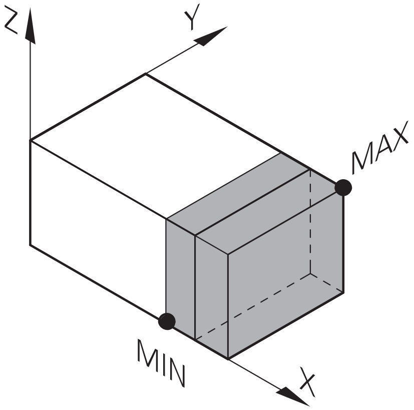
- Enter the NC program with the scanning cycles "Range" and "Meander" and the scanning direction X and the probe point interval 1 mm.
  Example:

```
0  BEGIN PGM OPTIDIGI MM
1  BLK FORM 0.1 Z X+0 Y+0 Z–10     ;REQUIRED FOR THE TNC'S
2  BLK FORM 0.2 X+100 Y+100 Z+10   ;PARALLEL GRAPHICS
3  TOOL DEF 1 L+0 R+4
4  TOOL CALL 1 Z S1000

5  TCH PROBE 5.0 RANGE             ;DEFINITION OF THE RANGE
6  TCH PROBE 5.1 PGM NAME:DIGIDAT  ;TO BE DIGITIZED WITH THE
7  TCH PROBE 5.2 Z X+0 Y+0 Z–10    ;PROGRAM NAME FOR THE
8  TCH PROBE 5.3 X+100 Y+100 Z+10  ;DIGITIZED DATA AND FOR THE
9  TCH PROBE 5.4 HEIGHT: +25       ;CLEARANCE HEIGHT(ABSOLUTE)

10 TCH PROBE 6.0 MEANDER           ;MEANDER SCANNING IN
11 TCH PROBE 6.1 DIRECTN:X         ;X DIRECTION WITH PROBE
12 TCH PROBE 6.2 TRAVEL:0.5 L.SPAC:1 PP.INT:1;POINT INTERVAL
                                   ;AND THE TRAVEL
13 END PGM OPTIDIGI MM
```

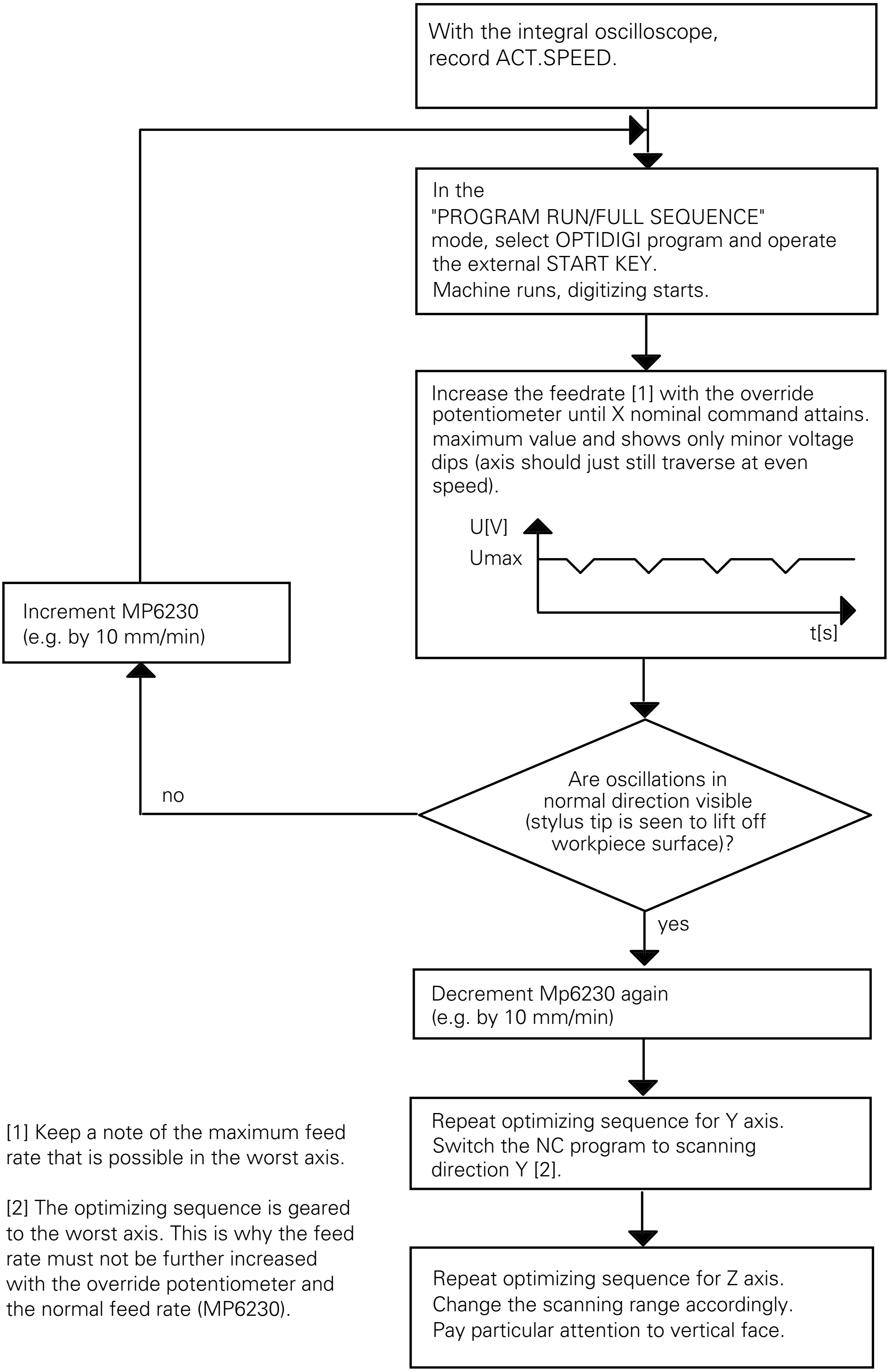
Optimize the X and Y axes by defining the range such that only the level surface of the component is scanned.



Optimize the Z axis by defining the range such that mainly the vertical face is scanned.



Procedure for Optimizing Machine Parameters


With the integral oscilloscope, record ACT.SPEED.
In the "PROGRAM RUN/FULL SEQUENCE" mode, select OPTIDIGI program and operate the external START KEY. Machine runs, digitizing starts.
Increase the feedrate [1] with the override potentiometer until X nominal command attains. maximum value and shows only minor voltage dips (axis should just still traverse at even speed).
U[V]
Umax
t[s]
Increment MP6230 (e.g. by 10 mm/min)
no
Are oscillations in normal direction visible (stylus tip is seen to lift off workpiece surface)?
yes
Decrement Mp6230 again (e.g. by 10 mm/min)
Repeat optimizing sequence for Y axis. Switch the NC program to scanning direction Y [2].
Repeat optimizing sequence for Z axis. Change the scanning range accordingly. Pay particular attention to vertical face.

[1] Keep a note of the maximum feed rate that is possible in the worst axis.

[2] The optimizing sequence is geared to the worst axis. This is why the feed rate must not be further increased with the override potentiometer and the normal feed rate (MP6230).

Calculation of Possible Oscillations in Normal Direction

$$\text{MP6210 [1/s]} = \frac{\text{optimierter } F_{scan} \text{ [mm/min]}}{\text{PP.INT [mm]} \bullet \text{60 [s/min]}}$$

When the calculated machine parameter MP 6210 is entered the feed override potentiometer is trimmed to the "attained feed rate."

4.15.4 Digitizing with the Measuring Touch Probe

The measuring touch probe permits high digitizing speeds up to 3 m/min (118 ipm). The stylus deflection is measured in every direction directly by integral measuring systems and evaluated in the TNC.

Technical prerequisites

- Interfaced measuring touch probe
- Adapter kit for digitizing with the measuring touch probe

The machine must be prepared for the use of the measuring touch probe. You must mechanically clamp the ballscrew and must ensure that the spindle drive cannot be started while the probe is in use.

The adapter kit for digitizing with the measuring touch probe also enables digitizing with the TS touch trigger probe.

Interfacing the measuring touch probe

With MP6200 you define whether to use the measuring or the triggering touch probe.

DANGER OF BREAKAGE
If you wish to use both the triggering and the measuring touch probe, you must make quite sure that the type of touch probe in use is entered in MP6200.

The counting direction of the measuring system signals must match the counting direction of the measuring systems for the machine tool axes (MP210). In the "Positioning with MDI" mode you can position the machine by pressing the PNT soft key. The machine must move in the direction in which the stylus was deflected. If this is not the case, you must change the counting direction in MP6320.

With MP6322 you assign the touch probe axes (the measuring systems in the probe) to the machine axes. For machines with swivel heads you must enter the respective mounting position of the touch probe in MP6322. If the touch probe is in a horizontal position, the undefined deflection resulting from the stylus's own weight makes it impossible to find the center of the stylus tip. It is therefore not possible to use the measuring touch probe in a horizontal attitude to locate a workpiece. In a horizontal attitude the measuring touch probe can be used only for digitizing.

DANGER OF BREAKAGE
The mounting position of the touch probe must be entered correctly in MP6322, otherwise the calculation of the maximum deflection from MP6330 may be incorrect.

If the stylus is deflected by a distance greater than the value defined in MP 6330, the blinking error message `Stylus deflection exceeds max.` appears.

With MP 6310 you define the mean constant deflection depth during digitizing. On standard parts an entry value of 1 mm has proven to be useful. On parts with sharp changes in directions (steep edges) that are scanned at high speed you must increase the deflection depth. The probing for can also be adjusted with the deflection depth.

After the "Meander" or "Contour line" cycle has been started, the probe moves at the feed rate defined in MP6361 to the clearance height, and then in the working plain to the point above the starting point. It then moves at the feed rate defined in MP6350 to the MIN point. If no touch point is reached, the probe moves to the first touch point in the direction defined in the cycle at the feed rate given in MP6350.

MP6360 (probing feed rate) and MP6361 (rapid traverse in the probe cycle) are effective in the standard touch cycles. With MP6362 you can automatically reduce the probing feed rate if the ball tip moves too far from the path.

During contour-line scanning the probe sometimes ends the contour line at a point located near but not exactly at the starting point. MP6390 defines a target window within which the probe is considered to have returned to the starting point. The target window is a square. The entry value is half the length of one side of the square.

MP6200 Selection of triggering or measuring touch probe (only with the digitizing-with-measuring-probe option)
Input: 0 = triggering touch probe (e.g. TS 220)
1 = measuring touch probe (e.g. TM 110)

MP6310 Stylus deflection depth (measuring touch probe)
Input: 0.1000 to 2.0000 [mm]

MP6320 Counting direction of the measuring system signals (measuring touch probe)
Input: %xxx
Bit 0 to 2 0 = positive
Axis X to Z 1 = negative

MP6321 Measuring the center offset while calibrating the measuring touch probe
Input: 0 = Calibrate and measure center offset
1 = Calibrate without measuring center offset

MP6322.0-2 Assignment of touch probe axes to the machine axes
Input: 0 = Touch probe axis X
1 = Touch probe axis Y
2 = Touch probe axis Z

MP6330 Maximum stylus deflection (measuring touch probe)
Input: 0.1 to 4.000 [mm]

MP6350 Feed rate for positioning to the MIN point and contour approach (measuring touch probe)
Input: 10 to 3000 [mm/min]

MP6360 Probing feed rate (measuring touch probe)
Input: 10 to 3000 [mm/min]

MP6361 Rapid traverse in scanning cycle (measuring touch probe)
Input: 10 to 10 000 [mm/min]

MP6362 Feed rate reduction if stylus of the measuring touch probe is deflected away from its path
Input: 0 = Feed rate reduction not active
1 = Feed rate reduction active

MP6370 Radial acceleration for digitizing with the measuring touch probe
Input: 0.001 to 3.000 [m/s^2]
Recommended input value: 0.1

MP6390 Target window for contour-line end point
Input: 0.1000 to 4.0000 [mm]

4.15.5 Tool Measurement

With the HEIDENHAIN tool touch probe TT 120 you can measure and inspect tools. HEIDENHAIN provides standard cycles for measuring and inspecting tool automatically with the TT 120 (see User's Manual).

Technical Prerequisites:

- TT 120
- Central tool file TOOL.T must be active (via machine parameter)

The TNC can store the calibration data for up to three different touch probes. The current data can be activated using M4574/M4575. For three sets of calibration data to be stored at the same time, bit 3 must be set in MP7490.

Standard measuring cycles
With MP6500 bit 0 you enable the cycles for tool measurement. Of course you should do this only after the TT 120 has been mounted and interfaced.

Calibrating tool radius and tool length
With MP6500 bit 1 and bit 2 you define whether tool radius and tool length measurement are permitted, and whether the individual tool teeth are to be measured.

Oriented spindle stop
With MP6500 bit 3 you define whether the single-tooth measurement should be performed with or without an oriented spindle stop. Single-tooth measurement is not possible without spindle orientation, and under certain circumstances a tool radius measurement is also incorrect. With MP6560 you define whether the spindle should be oriented directly by the NC or through the PLC. If the spindle is oriented directly by the NC, you need only reset M4012 (see "Spindle Orientation." If the spindle is oriented by the PLC, you must enter the number of the M function in MP6560. The respective positions are then transferred as in the "spindle orientation" cycle. M4017 is set during every spindle orientation.

Probing direction
You define the probing direction for tool radius measurement in MP6505.x.

Offset of probe contact to the tool
In MP6530.x you enter the distance from the tool bottom to the top of the probe contact during tool radius measurement. In the tool table you enter an additional tool-specific offset in the L-OFFS field.

Safety zone
In MP6540 you define a safety zone around the contact of the TT 120. 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.

Probe contact

Enter the diameter (disk) or edge length (cube) of the probe contact in MP6531.x. In MP6580 you enter the coordinates of the stylus center referenced to the machine datum. After calibration, the NC internally stores the exact center of the probe contact. If your probe contact is a cube, it suffices to approach from one direction only. This can be set in MP6500 bit 8 = 1. With bit 9 you can set whether the basic rotation of the cube should be determined automatically or whether the cube should be aligned axis-parallel. In the first case the edge of the touch probe is approached twice and the basic rotation is then calculated. All of the subsequent probing is done automatically at a right angle to the touch probe edge.

If you set bit 10, the only axes moved by the TNC during probing are the tool axis and the axis from MP6505 (probing direction). Bit 9 is reset when bit 10 is set.

Probing feed rate and spindle speed

For tool measurement of a non-rotating tool the probing feed rate is taken from MP6520. For tool measurement of a rotating tool the probing feed rate and the spindle speed are automatically calculated by the TNC. The rotational speed is calculated from the maximum permissible surface cutting speed (MP6570) and the tool radius from the tool table. With MP6500 bit 4 you can set whether the speed should be limited to a max. 1000 rpm or not.

$$n = \frac{\text{MP6570}}{2 \bullet \pi \bullet r \bullet 10^{-3}}$$

n = Rotational speed [rpm]
MP6570 = Maximum permissible surface speed of the tool edge [m/min]
r = Tool radius [mm]

The probing feed rate is calculated from the calculated rotational speed and the measuring tolerance given in MP6510.

$v = \text{meas. tolerance} \bullet n$

meas. tolerance = meas. tolerance [mm] depending on MP6507
n = Rotational speed [rpm]

With MP6507 you define the method of calculating the probing feed rate:

MP6507=0: Calculation of the feed rate with constant tolerance

This setting guarantees that the measuring tolerance remains constant regardless of the tool radius (MP6510). If the tool is very large, however, the necessary probing feed rate comes so close to zero that it falls below the lowest programmable increment. The smaller the maximum surface cutting speed and the permissible measuring error, the sooner this effect becomes noticeable.

MP6507=1: Calculation of the feed rate with variable tolerance

In this setting the permissible measuring tolerance changes depending on the tool radius. This ensures that there is a probing feed rate even for large tool radii. The measuring tolerance changes according to the following table.

Tool radius	Measuring tolerance
Up to 30 mm	MP6510
30 to 60 mm	2 • MP6510
60 to 90 mm	3 • MP6510
90 to 120 mm	4 • MP6510

etc.

MP6507=2: Constant probing feed rate

The probing feed rate remains constant regardless of the tool. The absolute measuring error grows linearly with increasing tool radius.

$$\text{Meas. tolerance} = \frac{r}{5\,[\text{mm}]} \bullet \text{MP6510}$$

r = Tool radius [mm]
MP6510 = Max. permissible measuring error [mm]

$$v = \text{meas. tolerance} \bullet n$$

$$v = \frac{\text{MP6570} \bullet \text{MP6510}}{10 \bullet \pi \bullet 10^{-3}}$$

v = Probing feed rate [m/min]
MP6570 = Maximum permissible surface speed at the cutting edge [m/min]

Tool breakage

With MP6500 bit 5 and bit 6, you can set whether or not the NC program should be stopped if the breakage tolerance is exceeded. Marker M4063 is always set if the breakage tolerance is exceeded. If bit 11 is set, the result of "Check tool" is not entered in the tool table.

Markers in the PLC

Marker M4060 is set when a tool measuring cycle is started. Marker M4061 indicates whether a cycle for tool measurement or tool inspection was activated. If inspection shows that one of the entered tolerances is exceeded, the tool is disabled and Marker M2392 or M2393 is set.

Markers M4050, M4051, M4052, M4053, M4055 and M4056 function as in the standard probing cycles. The cycles for tool measurement must therefore also be released by the PLC with Marker M4055. If the spindle is oriented directly by the NC (MP6560 = –1), you must reset marker M4012.

MP6500 Tool measurement with TT 120

Input: %xxxx

Bit 0 0: Cycles for tool measurement disabled
1: Cycles for tool measurement not disabled

Bit 1 0: Tool radius measurement permitted. Tool length measurement with rotating spindle
1: Tool radius measurement and single tooth measurement disabled

Bit 2 0: Tool length measurement with stationary spindle (Bit 1 = 1)
1: Tool length measurement with rotating spindle. The tool length is then calibrated with the spindle rotating only if a tool radius offset (TT:R-OFFS) is entered in the tool table.

Bit 3 0: Tool measurement with oriented spindle stop
1: Tool measurement without oriented spindle stop. Single-tooth measurement is not possible. Tool radius measurement might be incorrect.

Bit 4 0: Measuring speed is limited to max. 1000 rpm
1: Measuring speed is not limited

Bit 5 NC stop during tool inspection
0: The NC program is not stopped if breakage tolerance exceeded
1: If the breakage tolerance is exceeded, the NC program is stopped and the error message **TOOL BROKEN** is output

Bit 6 NC stop during tool measurement
0: The NC program is not stopped if breakage tolerance exceeded
1: If the breakage tolerance is exceeded, the NC program is stopped and the error message **TOOL BROKEN** is output

Bit 7 *Reserved*

Bit 8 Probing routine
0: The probe contact is approached from several directions
1: The probe contact is approached from one direction only

Bit 9 Automatic determination of the basic rotation for the probe contact (bit 8 = 1)
0: Basic rotation is not determined
1: Basic rotation for the probe contact is determined automatically

Bit10 Probing routine (bit 8 = 1)
0: The starting point is prepositioned in all three axes
1: The starting point is only prepositioned in the tool axis and the axis for probing direction (MP6505) (bit 9 = 0)

Bit11 "Check tool" and edit the tool table
0: The tool table is edited after tool check
1: The tool table is not edited after tool check

MP6505 Probing direction for tool radius measurement
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

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 Max. permissible measuring error for tool measurement with rotating tool
Input: 0.002 to 0.999 [mm]

MP6520 Probing feed rate for tool measurement with non-rotating tool
Input: 10 to 3000 [mm/min]

MP6530 Distance from tool lower edge to probe contract upper edge for tool radius measurement
Input: 0.001 to 99.9999 [mm]
MP6530.0 Traverse range 1
MP6530.1 Traverse range 2
MP6530.2 Traverse range 3

MP6531 Diameter or edge length of the TT 120 probe contact
Input: 0.001 to 99 999.9999 [mm]
MP6531.0 Traverse range 1
MP6531.1 Traverse range 2
MP6531.2 Traverse range 3

MP6540 Safety zone around the probe contact TT 120 for pre-positioning
Input: 0.001 to 99 999.9999 [mm]

MP6550 Rapid traverse in the probing cycle for TT 120
Input: 10 to 10 000 [mm/min]

MP6560 M function for spindle orientation with individual-tooth calibration
Input: –1 = Spindle orientation directly via NC
0 = Function inactive
1 to 88 = Number of the M function for spindle orientation via PLC

MP6570 Max. permissible surface cutting speed at the tool edge
Input: 1.0000 to 120.0000 [m/min]

MP6580.0-2 Coordinates of the TT 120 probe center referenced to the machine datum (traverse range 1)
Input: –99 999.9999 to +99 999.9999 [mm]

MP6581.0-2 Coordinates of the TT 120 probe center referenced to the machine datum (traverse range 2)
Input: –99 999.9999 to +99 999.9999 [mm]

MP6582.0-2 Coordinates of the TT 120 probe center referenced to the machine datum (traverse range 3)
Input: –99 999.9999 to +99 999.9999 [mm]

MP7490 Functions for traverse ranges
Input: %xxxx
Bit 3 Calibration data: touch probe for workpiece measurement
0 = one set of calibration data for all traverse ranges
1 = one set of calibration data for each individual traverse range

		Set	Reset
M4060	Cycle for tool measurement started	NC	NC
M4061	0 = tool measurement 1 = tool inspection	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

4.16 Electronic Handwheel

The following handwheels can be connected to HEIDENHAIN contouring controls:
- One HR 130 panel-mounted handwheel or
- Three HR 150 panel-mounted handwheels via HRA 110 handwheel adapter, or
- One HR 410 portable handwheel

For information on connecting the handwheels, see the chapter "Mounting and Electrical Installation." Handwheel operation is described in the User's Manual.

In MP7640 you indicate which handwheel is connected to the control. If you enter a value greater than zero but no handwheel is connected, the error message **`Handwheel?`** appears. Shock and vibration can result in a slight motion of the handwheel and therefore cause undesired motion in the machine axis. To prevent this, enter a threshold sensitivity in MP7660.

You can block traverse with the handwheel by setting M4576. In the "Handwheel" mode of operation you can enter in interpolation factor, which defines the traverse distance per handwheel revolution. In order to ensure that the rapid traverse rates fixed in machine parameter MP1010.x are not exceeded, the NC determines the minimum entry value for the interpolation factor. With MP7641 you determine whether the interpolation factor can be entered directly through the TNC keyboard or through the PLC Module 9036.

Interpolation factor	Traverse distance in per revolution [mm]	Becomes effective beginning from rapid traverse: MP1010.x [mm/min]
0	20	12000
1	10	6000
2	5	3000
3	2.5	1500
4	1.25	750
5	0.625	80
6	0.312	80
7	0.156	80
8	0.078	80
9	0.039	80
10	0.019	80

With MP7670.x you enter an even higher limit than that calculated by the NC. In MP7645.x you enter initializing parameters for the handwheel. These initializing parameters are presently evaluated only by the HRA 110 and HR 410. The functions are described in the corresponding chapters.

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

MP7641	Entry of interpolation factor
Input:	0 = via TNC keyboard
	1 = via PLC Module 9036

MP7650	Counting direction for handwheel
Input:	0 = negative counting direction
	1 = positive counting direction

MP7660	Threshold sensitivity for electronic handwheel
Input:	0 to 65 535 [increments]

MP7670.0-2	Interpolation factor for handwheel
Input:	0 to 10
MP7670.0	Interpolation factor for low velocity
MP7670.1	Interpolation factor for medium velocity (only HR 410)
MP7670.2	Interpolation factor for high velocity (only HR 410)

		Set	Reset
M4576	Handwheel lock	PLC	PLC

Module 9036: Write Status Information
The information to be written is designated by a transferred number. Handwheel interpolation factors are limited to the smallest possible value in accordance with the rapid traverse of the corresponding axis. This does not result in an error message, however. Handwheel interpolation can be transferred only if MP7641 = 1.

Number	Function	Value
0	Handwheel interpolation for key X	0 to 10
1	Handwheel interpolation for key Y	0 to 10
2	Handwheel interpolation for key Z	0 to 10
3	Handwheel interpolation for key IV (MP410.3)	0 to 10
4	Handwheel interpolation for key V (MP410.4)	0 to 10
5	Handwheel interpolation for axes	0 to 10
6	Choosing handwheel axis (not for HRA 110)	0 to 8 Axes 1 to 9
10	See "Incremental Jog Positioning" below	
11	Handwheel interpolation for axis 1	0 to 10
12	Handwheel interpolation for axis 2	0 to 10
13	Handwheel interpolation for axis 3	0 to 10
14	Handwheel interpolation for axis 4	0 to 10
15	Handwheel interpolation for axis 5	0 to 10
16	Handwheel interpolation for axis 6	0 to 10
17	Handwheel interpolation for axis 7	0 to 10
18	Handwheel interpolation for axis 8	0 to 10
19	Handwheel interpolation for 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 identifier>
                0:      Status was written: no error
                1:      Incorrect status identifier
                2:      Transferred value out of range
                3:      Entry disabled (e.g. per MP)
```

4.16.1 HR 130 Panel-Mounted Handwheel

MP7640 = 2
Pressing the axis direction keys moves the corresponding highlight and handwheel symbol on the screen.

4.16.2 HR 410 Portable Handwheel

MP7640 = 6: **HR 410**
With MP7645.0 you define whether the keys on the handwheel are evaluated by the NC or the PLC.

MP7645.0 = 0
Keys are evaluated by NC

X		IV
Y		V
Z		ACTUAL- POSITION CAPTURE
FEED RATE SLOW	FEED RATE MEDIUM	FEED RATE FAST
–		+
O109 I173	O110 I174	O111 I175

All keys except for the functions keys A, B, and C, are evaluated by the NC. MP7670.x defines the interpolation factor for the slow, medium and fast settings. MP7671.x defines the velocity for the slow, medium and fast settings. The speed is entered as a percentage of the manual feed rate (MP1020.x).

MP7645.0 =1
Keys are evaluated by PLC

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's axis and interpolation. W766 makes it possible to influence the feed rate by pressing the direction keys.

MP7645 Initializing parameters for handwheel
When an HR 410 is installed, MP7645.0 has the following meaning:
MP7645.0 Evaluation of HR 410 handwheel keypad
Input: 0 = Keys evaluated by NC
1 = Keys evaluated by PLC

MP7645.1 to MP7645.7 are without 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 Manual feed rate in handwheel mode with HR 410
Input: 0 to 1000 [% of MP1020]
MP7671.0 Low speed
MP7671.1 Medium speed
MP7671.2 High speed

4.16.3 HR 150 Panel-Mounted Handwheels with HRA 110 Adapter

MP7640 = 5

If the step switch is used for the selection of the interpolation factor (S1) then inputs I160 to I167 must be evaluated in the PLC and the result must be displayed with the aid of PLC Module 9036.

The third handwheel can be assigned to the axes X, Y, Z, IV, or V. The designation for axes IV and V is taken from MP410.x. MP7645.2 defines whether the axis for the third handwheel is selected with the axis selector switch (switch S2, see MP7645.0) or is permanently assigned in machine parameter MP7645.1.

If you have not connected an axis selection key, define the axis for the third handwheel in MP7645.1 and enter the value one in MP7645.2. If you have entered the value one in MP7645.2, the PLC inputs I168 to I175 are set with the axis selection switch. In this way the axis selector switch can also fulfill other tasks.

MP7645 Initialization parameters for handwheel
MP7645.0 to MP7645.2 have the following meaning when an HRA 110 is connected:

MP7645.0 Third handwheel assigned by axis selector switch

Entry	Switch position	3rd handwheel
0	1 (left stop)	Axis Z
	2	Axis IV (MP410.3)
	3	Axis V (MP410.4)
1	1 (left stop)	Axis X
	2	Axis Y
	3	Axis Z
	4	Axis IV (MP410.3)
	5	Axis V (MP410.4)
2	3	Axis Z
	4	Axis IV (MP410.3)
	5	Axis V (MP410.4)

MP7645.1 Assignment of 3rd handwheel if MP7645.2 = 1

Input: 0 = Simulation of switch position 1 (left stop) the third handwheel is assigned from MP7645.0 (Input: 0 or 1)
1 = Axis X
2 = Axis Y
4 = Axis Z
8 = Axis IV (MP410.3)
16 = Axis V (MP410.4)

MP7645.2 Assignment of a third handwheel via axis selector switch or MP7645.1

Entry 0 = Assigned by axis selector switch or according to MP7645.0
1 = Assigned by MP7645.1

MP7645.3 to MP7645.7 are without function.

Assignment of switch positions to PLC inputs

The tables below give the assignment of the switch positions of S1 and S2 to PLC inputs I160 to I175. Both switches operate with a 0 V logic, e.g. if switch S1 is in position 3 then input I162 is logic 0 and inputs I160, I161 and I163 to I167 are logically 1.

Step switch S1

Step switch for selecting interpolation factor

Switch Position	PLC Input
1 (left stop)	I160
2	I161
3	I162
4	I163
5	I164
6	I165
7	I166
8 (right stop)	I167

Step switch S2

Step switch for axis selection

Switch position	PLC input
1 (left stop)	I168
2	I169
3	I170
4	I171
5	I172
6	I173
7	I174
8 (right stop)	I175

4.17 PLC Inputs/Outputs

The logic unit provides you with digital inputs/outputs and analog inputs/outputs for the PLC. If the available number of inputs and outputs is not enough, you can add up to four PL 410 B input/output units for more I/O. For information on connecting the PL 410 B, see the chapter "Mounting and Electrical Installation."

To interrogate and set the inputs and outputs of the PLC I/O unit you need PLC modules.

	Logic Unit				**PLC I/O unit**	
	X9	**X41**	**X42**	**X48**	**Without analog inputs**	**With analog inputs**
Switching inputs 24 Vdc	–	–	56	–	64	56
Switching outputs 24 Vdc	–	31	–	–	31	29
Analog inputs ±10 Vdc	–	–	–	3	–	4
Inputs for Pt 100 thermistors	–	–	–	3	–	4
Analog outputs ±10 Vdc	13 [1)]	–	–	–	–	–
Control-is-ready output	–	2	–	–	1	1
Control-is-ready input	–	–	2	–	–	–

1) for each analog axis you need one analog output.

4.17.1 24-Vdc Switching Inputs/Outputs

The current conditions of the switching inputs and outputs are available for you in PLC addresses (see the Mounting and Electrical Installation Chapter). You must download the current conditions of the PLC I/O unit inputs with Module 9002 and update the outputs with Module 9005. With Module 9004 you can evaluate the rising or falling edges of the PLC inputs.

Module 9002: Read the inputs of a PLC input/output unit
The current conditions of the PLC inputs of the PLC input/output unit are read into the PLC addresses (see chapter "Mounting and Electrical Connection"). The addresses remain unchanged until you recall Module 9002. The module does not recognize whether a PLC I/O unit is actually connected.

```
Call:
PS   B/W/D/K    <Number of the PL>
                0:     First PLC I/O unit
                1:     Second PLC I/O unit
                2:     Third PLC I/O unit
                3:     Fourth PLC I/O unit
CM   9002
```

Module 9005: Update the outputs of a PLC I/O unit
The outputs of the PLC I/O unit are overwritten with the values from the PLC addresses (see chapter "Mounting and Electrical Connection"). The outputs are set or reset immediately at the time of module processing. The module does not recognize whether a PLC I/O unit is actually connected. You must not call Module 9005 as long as the Module 9003 is active through a Submit command.

```
Call:
PS   B/W/D/K    <Number of the PL>
                0:      First PLC I/O unit
                1:      Second PLC I/O unit
                2:      Third PLC I/O unit
                3:      Fourth PLC I/O unit
CM   9005
```

Module 9004: Read the edges of PLC inputs
With Module 9004 you set the edge markers or bits corresponding to the rising or falling edges of PLC markers in the specified byte range. Changes in the inputs are only recognized if a change occurs also in the PLC addresses (see Module 9002). Make sure that the given edge markers or edge bytes are in an unassigned area. The edge bytes are written beginning from the least significant bit. Unused 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
```

4.17.2 Analog Inputs

At socket X48, the logic unit provides you with ±10 Vdc analog inputs and analog inputs for connecting Pt 100 thermistors (see the chapter "Mounting and Electrical Connection"). The PLC input/output unit is available in a version with additional analog inputs (see the chapter "Mounting and Electrical Connection"). You must read the current conditions of the analog inputs with Module 9003.

Module 9003: Read the analog input

This module reads the current value of the given analog input. It does not recognize whether the given input is also present. You must not call Module 9003 as long as the Module 9005 is active through a Submit command.

PS	B/W/D/K	<Number of the analog input>	
		0 to 7:	Analog inputs X15 to X22 on the first PLC I/O unit
		8 to 15:	Analog inputs X15 to X22 on the second PLC I/O unit
		16 to 23:	Analog inputs X15 to X22 on the third PLC I/O unit
		24 to 31:	Analog inputs X15 to X22 on the fourth PLC I/O unit
		64 to 66:	±10V input on socket X48
		67 to 69:	Pt100 input on socket X48
CM	9003		
PL	W/D	<Analog value>	
		Numbers 0 to 31:	Natural number with the unit 0.1 V or 0.5°
		Numbers 64 to 69:	Natural number with the unit 0.01 V or 0.1°

4.17.3 Analog Outputs

With Module 9130 you apply an analog voltage to an analog output. You can control the analog outputs 1 to 13 at sockets X8 and X9 (see chapter "Mounting and Electrical Connection"). Please note that for each analog axis or an analog spindle you need one analog output, and that these outputs are then not available for the PLC.

The voltage is applied with a slight delay after the end of the PLC run. The module can be called only once for each output per PLC run. The voltage must be transferred in 1 mV format. Voltage greater than 10 V or less than –10 V are limited to their respective maximum values.

Module 9130: Output of an analog voltage
Call:

```
PS   B/W/D/K    <Number the analog output>
                1 to 6:     Analog output 1 to 6 (X8)
                7 to 13:    Analog output 7 to 13 (X9)
PS   B/W/D/K    <Analog voltage in millivolts>
CM   9130
```

4.18 Incremental Jog Positioning

The incremental jog positioning function can be switched on or off with the soft key INCREMENT ON/OFF. You can interrogate the current state using M4579. Jog positioning is carried out with W1046/W1048 (direction keys). The words W1050/W1052 remain functional (for compatibility reasons), but do not have to be used. You can limit the jog increment with Module 9036. Using Module 9186 the jog positioning function can be switched on or off via the PLC.

		Set	Reset
M4579	INCREMENT ON/OFF soft key set to on	NC	NC

NC software: 280470..:

With M4572 you can enable the "electronic handwheel" mode and the "incremental jog" function. Besides the "interpolation factor" prompt the "jog increment" prompt is also displayed. To activate incremental jog positioning you must set the corresponding bit in W1050 or W1051. Activation of incremental jog positioning is normally linked to the axis direction keys.

				Set	*Reset*
M4572	*Enabling of incremental jog positioning*			*PLC*	*PLC*
W1050	*Incremental jog positioning in positive direction*			*PLC*	*PLC*
	Bit	*0 to 8*	*0 = not active*		
	Axis	*1 to 9*	*1 = active*		
W1052	*Incremental jog positioning in negative direction*			*PLC*	*PLC*
	Bit	*0 to 8*	*0 = not active*		
	Axis	*1 to 9*	*1 = active*		

Module 9036: Writing the status information

The information to be overwritten is designated with a transferred number.

Number	Function	Value
0 to 6	(See section "Handwheel")	
10	Incremental jog limiting	0 to 50 mm = Jog increment limiting –1; <–2; >50 = Cancellation of Incremental jog limiting and activation of the last entered incremental jog –2 = Cancellation of Incremental jog limiting and activation of the minimum from the last entered jog increment and last limit
11 to 19	(See section "Handwheel")	

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 identifier>
                0:       Status written
                1:       Incorrect error identifier
                2:       Transferred value out of range
```

4.19 Hirth Coupling

A frequently applied method of locking rotary axes and swivel heads is the so-called Hirth coupling, in which finely splined plates engage to form a rigid connection. You must implement the exact positioning of the Hirth coupling in the PLC program. The NC merely rounds the datum point off to the corresponding grid measure from MP430.

"Manual" positioning
As soon as an axis direction key is pressed, the NC resets the corresponding bit in W1026 (axis in position). From it you derive the unclamping of the PLC in the Hirth axis. As soon as the axis-in-position bit is set, you check the nominal position with the Hirth grid and derive from that a PLC positioning to the next grid point.

"Electronic handwheel" positioning
For the current handwheel axis the corresponding bit in W1026 (axis in position) is reset. As long as the axis-in-position bit remains reset, the Hirth axis remains unclamped and the Hirth axis can be move with the handwheel. As soon as you set another handwheel axis, the axis-in-position bit is set for the previous axis. You 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. You must check the positions in the PLC during the program run. As soon as the "axis-in-position" bit is reset, check the target position with the Hirth grid. If the target position is not in the Hirth grid, you put out a PLC error message.

MP420.0-8 Hirth coupling
Input: 0 = no Hirth coupling
1= Hirth coupling

MP430.0-8 Prescribed increment for Hirth coupling
Input: 0.0000 to 30.0000 [°]

4.20 Datum Shift

With the datum shift function you can offset the defined datum point. The same starting position must be used for the datum shift as for the description of the machine geometry via MP7510 and following (see "Tilting Axes"). The datum shift function can only be activated during an M/S/T/Q strobe.

Datum shift with D528 to D544

In D528 to D544 you enter for each axis the distance by which the datum is to be shifted. For axes 6 to 9 you must use Module 9230.

With M4132 you activate the datum shift. After the datum is shifted, 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 position display for X axis without offset = 50
Offset value in D528 = +20
M4132 is set, i.e. the offset is active
New actual-value display X= +70

A description of the machine geometry via machine parameters MP7510 and following (see "Tilting axes") must have the same reference position as any datum shift.

		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 Module 9230 is called	PLC	NC

Datum shift with Module 9230

With Module 9230 you transfer the axis and distance by which the datum is to be shifted. When Module 9230 is called M4132 is set and the NC resets the datum shift M4132 after execution.

Call:

```
PS   B/W/D/K    <Axis>          [0 to 8]
PS   B/W/D/K    <Shift>       [0.0001 mm]
CM   9230
```

4.21 Tool Changer

You control the tool changer through the PLC. If the tool changer is operated with controlled axes, then use the PLC axes (see the section "PLC Axes"). However, you can also control the tool changer simply with proximity switches.

The information on the tool is stored in the tool table and the information on the tool changer is kept in the tool pocket table. The NC manages all tool information (replacement tool, tool age, etc.). The NC provides you with markers and words containing the information that you need to control the tool changer.

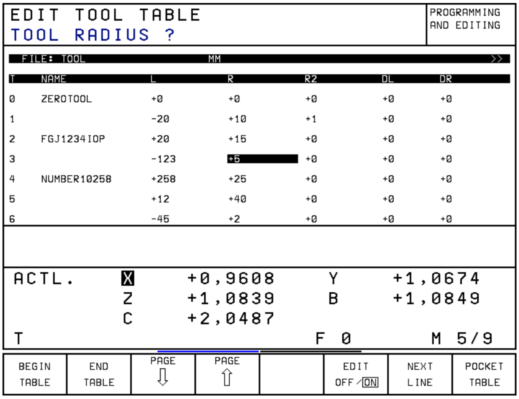
4.21.1 Tool Table, Pocket Table

In the "Program run" mode of operation you can edit the tool table. Press the TOOL TABLE soft key. From the tool table you can press the POCKET TABLE soft key to access the pocket table (see the User's Manual).

The current tool table has the name TOOL.T and the pocket table is called TOOL_P.TCH. Both are filed in the root directory TNC:\. In the Programming and Editing mode you can upload or download the tool table through the data interface after pressing the PGM MGT key. The pocket table is always automatically transferred as well. The identifiers T and R are assigned for the tool table and pocket table, respectively, in the external memory.

In MP7266.x you can define the elements of the tool table that are to be displayed and the sequence in which they appear.

Example:



EDIT TOOL TABLE
TOOL RADIUS ?

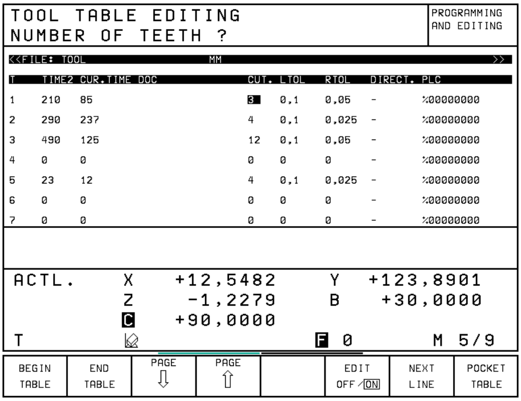
PROGRAMMING AND EDITING

FILE: TOOL MM >>

T	NAME	L	R	R2	DL	DR
0	ZEROTOOL	+0	+0	+0	+0	+0
1		-20	+10	+1	+0	+0
2	FGJ1234IOP	+20	+15	+0	+0	+0
3		-123	+5	+0	+0	+0
4	NUMBER10258	+258	+25	+0	+0	+0
5		+12	+40	+0	+0	+0
6		-45	+2	+0	+0	+0

```
ACTL.   X   +0,9608   Y   +1,0674
        Z   +1,0839   B   +1,0849
        C   +2,0487
T                     F 0       M 5/9
```

Left side of the tool table



TOOL TABLE EDITING
NUMBER OF TEETH ?

PROGRAMMING AND EDITING

<<FILE: TOOL MM >>

T	TIME2	CUR.TIME	DOC	CUT.	LTOL	RTOL	DIRECT.	PLC
1	210	85		3	0,1	0,05	-	%00000000
2	290	237		4	0,1	0,025	-	%00000000
3	490	125		12	0,1	0,05	-	%00000000
4	0	0		0	0	0	-	%00000000
5	23	12		4	0,1	0,025	-	%00000000
6	0	0		0	0	0	-	%00000000
7	0	0		0	0	0	-	%00000000

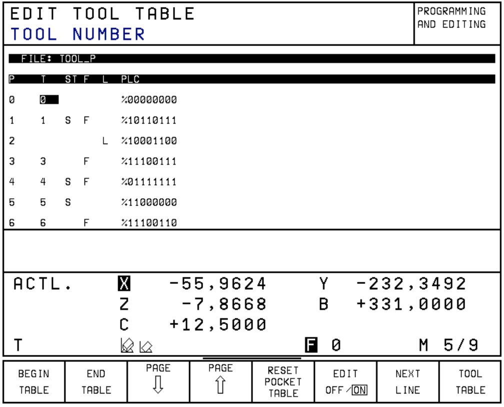
```
ACTL.   X   +12,5482   Y   +123,8901
        Z    -1,2279   B    +30,0000
        C   +90,0000
T                      F 0        M 5/9
```

Right side of the tool table

When configuring the tool table, note that the complete width cannot be more than 250 characters. Wider tables cannot be transferred via data interface. The width of individual columns is included in the description of MP7266.x.

In MP7267.x you can define the elements of the pocket table that are to be displayed and the sequence in which they appear.

Example:



EDIT TOOL TABLE
TOOL NUMBER

PROGRAMMING AND EDITING

FILE: TOOL_P

P	T	ST	F	L	PLC
0	0				%00000000
1	1	S	F		%10110111
2				L	%10001100
3	3		F		%11100111
4	4	S	F		%01111111
5	5	S			%11000000
6	6		F		%11100110

```
ACTL.   X   -55,9624    Y  -232,3492
        Z    -7,8668    B  +331,0000
        C   +12,5000
T                      F 0       M 5/9
```

If you wish to edit the tool and pocket tables, they must not be locked or protected through MP7224.x (see files). In MP7260 you define the number of tools in the tool table. If you enter the value zero in MP7260, there will be no tool table (TOOL.T will not exist). You then use TOOL DEF to program the tool length and tool radius in the NC part program (see the User's Manual). If the control is used without a tool table, there will also be no automatic tool management. You define the number of pockets in the tool magazine in MP7261. If you enter the value zero in MP7261, no pocket table will be generated. With Modules 9092, 9093, 9094 and 9096 you can read the tool and pocket tables and overwrite them. If an input field has been opened with the editor at the time the modules are called, this field is closed automatically. The current tool data are shown in the additional status display (graphic window).

MP7260 Number of the tools in the tool table
Input: 0 to 32 767

MP7261 Number of the pockets in the tool magazine
Input: 0 to 32 767

MP7266.0-26 Elements of the tool table
Input: 0 = no display
1 to 99 = position in the tool table

MP	**Description**	**Column name**	**Column width**
7266.0	16-character alphanumeric tool name	NAME	16
7266.1	Tool length	L	11
7266.2	Tool radius	R	11
7266.3	Tool radius 2 for toroidal cutters	R2	11
7266.4	Oversize for tool length	DL	8
7266.5	Oversize for tool radius	DR	8
7266.6	Oversize for tool radius 2	DR2	8
7266.7	Tool locked?	TL	2
7266.8	Replacement tool	RT	3
7266.9	Max. tool life (M4543)	TIME1	5
7266.10	Max. tool life TOOL CALL	TIME2	5
7266.11	Current tool age	CUR. TIME	8
7266.12	Comment on the tool	DOC	16
7266.13	Number of tool teeth	CUT	4
7266.14	Wear tolerance for tool length	LTOL	6
7266.15	Wear tolerance for tool radius	RTOL	6
7266.16	Cutting direction of the tool	DIRECT	7
7266.17	Additional information for PLC (Module 9093)	PLC	9
7266.18	Tool length offset	TT: L-OFFS	11
7266.19	Tool radius offset	TT: R-OFFS	11
7266.20	Breakage tolerance for tool length	LBREAK	6
7266.21	Breakage tolerance for tool radius	RBREAK	6
7266.22	Tooth length	LCUTS	11
7266.23	Plunge angle	ANGLE	7
7266.24[1]	Tool type	TYP	5
7266.25[1]	Tool material	TMAT	16
7266.26[1]	Cutting data table	CDT	16

MP7267.0-4 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)

[1] *As of NC software 280 472 01*

Module 9092: Seek an entry in the table selected for program run (.T/.D/.TCH)

You can search for certain value in certain column in the table selected for program run (M status is set). The function replies with the number of the line in which the value was found. This makes it possible for you, for example, to find the vacant pocket (corresponds to T0) in the pocket table (.TCH).

You must start the module through a Submit job. You must enter the value that your looking for as a natural number, shifted by the number of decimal places that can be entered. If you wish to look for more occurrences of the same value, you must enter as the starting line the line number of the last occurrence plus one.

Call only in a Submit job:

```
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 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 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 length (RBREAK)
                            22:  Tooth length (LCUTS)
                            23:  Plunge angle (ANGLE)
                .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 in entered)
                           1:    Special tool (ST); (0= no, 1= yes)
                           2:    Fixed pocket (F); (0= no, 1= yes)
                           3:    Locked pocket (L); (0= no, 1= yes)
                           4:    PLC status (PLC)

PS   B/W/D/K   <Line number for beginning of search >
CM   9092
PL   B/W/D     <Line number > (if error –1)
PL   B/W/D     <Error number>
               0:     No error, element was found
               1:     Call did not come from a Submit job
               2:     File type does not exist
               3:     No file of the given type found with M status
               4:     Line number not contained in file
               5:     Incorrect element number
               6:     Element value not found
```

Module 9093: Read data from the tables selected for program run (.T/.D/.TCH)
You can read the content of certain elements in the tables selected for program run (M status is set). For this purpose you transfer the line number (= tool number for .T, vector number for .D, or pocket number for .TCH) and the element number of the element to be read.

You must start the module through a Submit job. You must enter the new values as a natural number, shifted by the number of decimal places that can be entered.

Call only in a Submit job:

```
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
PL   B/W/D     <Element value >
PL   B/W/D     <Error number>
               0:     No error, element was read
               1:     Call did not come from a Submit job
               2:     File type does not exist
               3:     No file of the given type found with M status
               4:     Line number not contained in file
               5:     Incorrect element number
```

Module 9094: Write data in a tool datum table

You can overwrite the content of certain elements in a table selected for program run (M status is set). For this purpose you must transfer the line number and the element number of the element to be overwritten. When Module 9094 is run, it initializes the geometry.

You must start the module through a Submit job. You must enter the new values as a natural number, shifted by the number of decimal places that can be entered.

Call only in a Submit job:

```
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 status>
                0:     No error, element was written
                1:     Call did not come from a Submit job
                2:     File type does not exist
                3:     No file of the given type found with M status
                4:     Line number not contained in file
                5:     Incorrect element number
                6:     Element value outside the permissible range
```

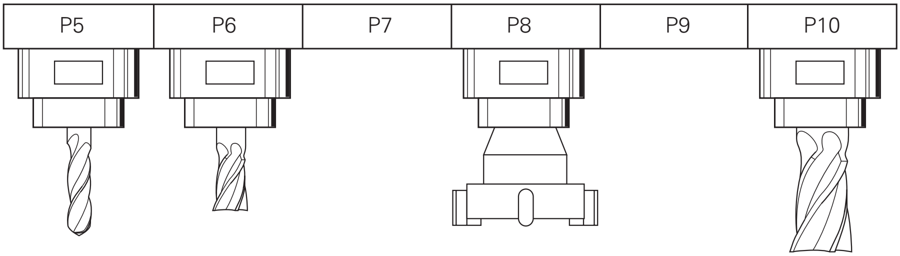
Module 9096: Delete a line in the tool table

With Module 9096 you can delete a line from the tool table, and break a link with a replacement tool if one exists.

Call in the submit job only:

```
PS   B/W/D/K    <Tool number / Pocket number>
PS   B/W/D/K    <Mode>
                Bit 0:  Delete entries in the pocket table
                        0: Pocket table is not changed
                        1: Tool number is deleted in the pocket table
                Bit 1:  Tool number or pocket number
                        0:    Transferred value = tool number
                        1:    Transferred value = pocket number
CM          9096
```

Special tools
In the pocket table you define tools as special tools in the "ST" field. If these tools are oversize tools for which one pocket is not large enough, you must leave space free on both sides of the pocket. You lock the pockets to be left free in the "L" field.



You deselect the variable pocket coding for special tools with M4541. As soon as M4541 is set, all special tools are returned to their original places in spite of the "variable pocket coding" setting. With the "F" field (fixed pocket) you can also define this function selectively for individual tools (including normal tools).

		Set	Reset
M4541	In spite of the variable pocket coding, return special tool to its original pocket	PLC	PLC

Tool life, replacement tool

In the tool table you can enter for each tool two tool life values (TIME1 and TIME2) and one replacement tool (RT). If the current tool age (CUR.TIME) for a TOOL CALL is greater than TIME2, the pocket number or tool number (MP7480) of the replacement tool is automatically downloaded. If TIME2 is greater than zero and no replacement tool is defined, the error message **`Max. tool age expired`** will appear in the event of a TOOL CALL after this time elapses. If the current tool age becomes greater than TIME1, the NC sets marker M4543. You decide in the PLC what should happen when M4543 is set (e.g. display a PLC error message).

With the M function M101 you activate the automatic insertion of the replacement tool after expiration of the tool life (TIME1 of TIME2). M101 is deactivated with M102. The replacement tool is not changed exactly upon expiration of the tool life; it can vary by a few NC blocks depending on the degree of utilization of the processor. A T-strobe M4073 is sent to the PLC and the marker M4525 is set. For automatic tool changing in standard NC programs (NC blocks with RR, RL or R0), the replacement tool must have the same radius defined as for the original tool.

No radius compensation is entered in NC blocks with surface-normal vectors. For each tool a delta value is entered for radius (DR) and length (DL). These delta values are offset by the TNC in NC programs with surface-normal vectors. If the radius of the replacement tool is different from the radius of the original tool, you must enter the difference in the DR field in the tool table. The delta value must always be negative. Entering a positive delta value will provoke the error message **`TOOL RADIUS TOO LARGE.`** You can suppress this error message with the M function M107. You can cancel M107 with M108. With MP7680 you define for NC blocks with surface-normal vectors whether calculation of the tool length includes the oversize for the tool radius (DR2).

The current tool age between tool calls is counted only in the "Program run, full sequence" and "Program run, single block" modes. Spindle operation time and machine traversing time play no role in the current tool age. The tool time counter is not stopped until the program is terminated with an "internal stop," M02, M30, or END PGM. The tool time counter does not run in the "Manual operation," "Electronic Handwheel," or "Positioning with Manual Data Input" modes. The operator can reset the current tool age by entering zero.

		Set	Reset
M4543	Tool life expired (TIME1 in the tool table)	NC	NC, PLC

MP7680 Machine parameters with multiple function
Input: %xxxxxxx

Bit 6	Tool length for blocks with surface-normal vectors
0 =	Without DR2 from the tool table
1 =	With DR2 from the tool table

4.21.2 Automatic Calculation of Cutting Data

As of NC software 280 472 01

The optimal speed and its corresponding feed rate can be calculated from the values entered in cutting data tables for workpiece and tool material. Depending on the type of tool (for milling or drilling), the values for cutting speed, feed rate per tooth or feed rate per revolution may be included in these tables.

	$S = \frac{v_c \bullet 1\,000}{d \bullet \pi}$	$S =$	spindle speed [rpm]
		$v_c =$	cutting speed [m/min]
		$d =$	tool diameter [mm]
Mill:	$F = f_z \bullet z \bullet S$	$F =$	feed rate [mm/min]
		$f_z =$	feed rate per tooth [mm]
		$z =$	number of teeth
Drill:	$F = f_u \bullet S$	$f_u =$	feed rate per revolution [mm]

Tool table

In the tool table enter the name of the cutting data table that is to be used for that tool (under the column CDT). Define the tool type under TYP (DRILL/TAP/MILL). You also need to indicate the radius of the tool (R), the tool material (TMAT) and, in the case of milling machines, the number of teeth (CUT).

The three 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

Your tool manufacturer will provide you with the necessary tool-specific cutting data.

Cutting data tables have the file extension CDT. Each line of the cutting data table contains the cutting data for a particular combination of workpiece and tool material. With milling machines you can enter up to four different cutting speeds with the corresponding feed rates per tooth. In the cutting data tables of the tool manufacturers these data are given for various infeeds and for climb and up-cut milling. With drilling machines only enter one cutting speed and its corresponding feed rate.

There is a standard design for cutting data tables stored in the TNC root directory (TNC:\). You can set up as many tables as you wish. If you want to change the standard format, you must store this under another path. Otherwise your changes will be overwritten with the standard HEIDENHAIN settings at the next software update. Using the command PCDT =, enter the path where you have stored your cutting data tables in the system file TNC.SYS.

Material tables

The workpiece and tool materials you are using are defined in the tables TMAT.TAB (tool) and WMAT.TAB (workpiece). The standard formats for these tables are stored in the root directory TNC:\. You can edit or expand these tables as you wish. However the new tables need to be stored under a different path, otherwise they will once again be overwritten with the standard HEIDENHAIN settings at the next software update. You need to enter the path and file names for these tables in the system file TNC.SYS using the commands TMAT= and WMAT=. In the material tables enter the short form of the tool material name under the column Name (e.g. HSS). In the DOC column you can enter some additional information about the material.

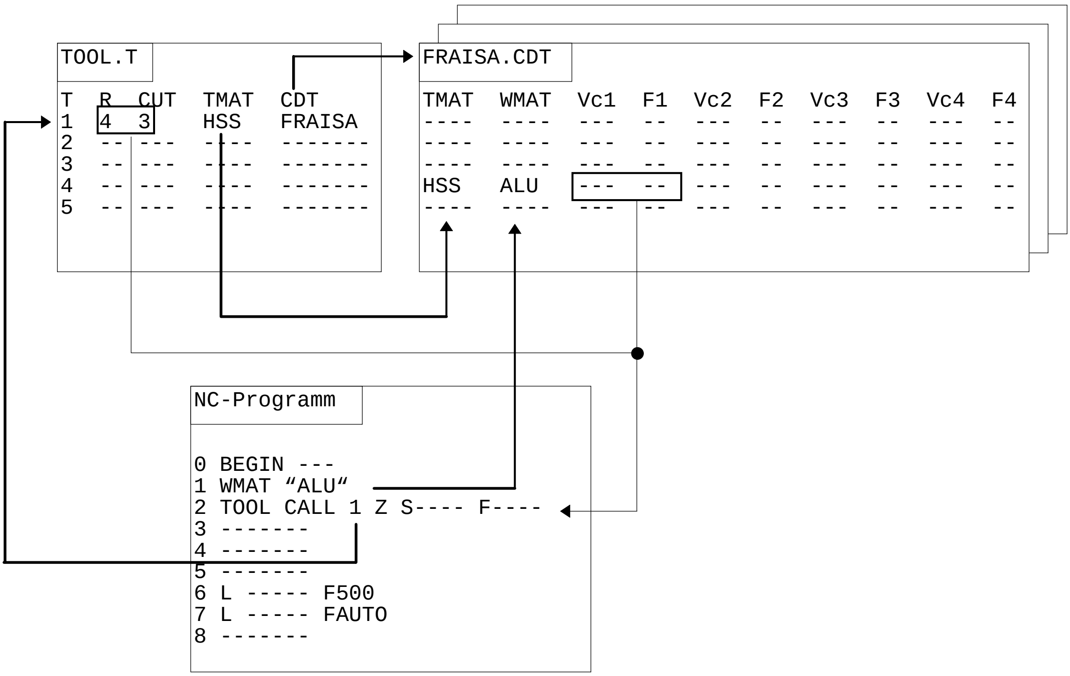
When defining the material in the tool table or in the NC program, the contents of the corresponding material table will be displayed when you press the SELECT MATERIAL soft key.

Calculating the cutting data

Define in the NC program the workpiece material being used using the WMAT soft key. In the TOOL CALL block you will be offered a number of soft keys to choose between S1 to S4 when entering the spindle speed S. You can also choose between one of four set feed rates. If you enter the speed manually, this same value will be taken into consideration when calculating the feed rate. The opposite does not hold true (input F and calculating S).

The feed rate entered in the TOOL CALL block is valid until you program another feed rate. You can take over the feed rate from the TOOL CALL block at any time using the F AUTO soft key.

Principle



```
TOOL.T

T  R  CUT  TMAT  CDT
1  4  3    HSS   FRAISA
2  -- ---  ----  -------
3  -- ---  ----  -------
4  -- ---  ----  -------
5  -- ---  ----  -------
```

```
FRAISA.CDT

TMAT  WMAT  Vc1  F1  Vc2  F2  Vc3  F3  Vc4  F4
----  ----  ---  --  ---  --  ---  --  ---  --
----  ----  ---  --  ---  --  ---  --  ---  --
----  ----  ---  --  ---  --  ---  --  ---  --
HSS   ALU   ---  --  ---  --  ---  --  ---  --
----  ----  ---  --  ---  --  ---  --  ---  --
```

```
NC-Programm

0 BEGIN ---
1 WMAT “ALU“
2 TOOL CALL 1 Z S---- F----
3 -------
4 -------
5 -------
6 L ----- F500
7 L ----- FAUTO
8 -------
```

4.21.3 Automatic Tool Recognition

Automatic tool identification is possible with the Balluff tool identification system (BIS). Please contact HEIDENHAIN for more information.

4.21.4 Controlling the Tool Changer

You must program the control of a tool changer, i.e., the positioning of the changing arm and carousel and the complete tool-change sequence, in the PLC. The NC handles tool management, i.e. tool life, tool pocket assignment and evaluation of the TOOL DEF and TOOL CALL blocks. The NC and PLC communicate by means of markers and words.

When a TOOL CALL BLOCK is executed the tool geometry of the defined tool is taken from the tool table. With M4538 you activate the tool geometry of the tool defined in W264. M4538 can be activated only together with an M/S/T/Q strobe or when the axis is stationary (control-in-operation symbol "*" off). With this marker you can ensure, for example, that the current tool geometry is active even when the tool changing process is interrupted.

With the TOOL DEF block you can pre-position the tool changer. After a tool has been changed, the next tool is programmed with TOOL DEF. The PLC evaluates the tool and pocket number and pre-positions the tool changer to the follow-up tool.

Calling an NC program with TOOL CALL

With the NC block TOOL CALL you can also call an NC program of your own definition. The tool geometry is then not taken over with this TOOL CALL and you must program a TOOL CALL at another location for updating the tool data.

Define the name of the calling NC program in the file PLC:\NCMACRO.SYS as follows:
TC = <Path name>\<File name>

With such a program you can, for example, very easily program a positioning motion to the tool-change position. It is best to file the program in the PLC partition to protect it against being changed by the end user. Since the tool data, as mentioned above, are not active in the activated program, you must interrogate them in with FN18 (see the chapter "PLC Programming"). With these data you can program a TOOL CALL at an appropriate location in the called program so that the tool data become active and a T-strobe is transferred to the PLC.

With FN18 you can also overwrite the software limit switch for tool-change positioning. If you use FN18 to also call the programmed position after the TOOL CALL, you can program a continuous movement from the tool magazine to the connecting position (M112).

With FN20: WAIT FOR you can suspend execution of the NC program until the entered condition is fulfilled. Comparisons of a PLC variable with a constant are permitted (see "PLC Programming" chapter). To ensure that during a block scan the tool-change program is not run until the end you must enter the instruction NCMACRO=TC in the MGROUPS.SYS file. See "Reapproaching the Contour." If no NC program is specified in the file NCMACRO.SYS, the TOOL CALL is executed as before.

Example:

```
0   BEGIN PGM TCALL MM
1   M112 T4 ; INSERT AN ARC IN ORDER TO POSITION CONTINUOUSLY
2   FN 18: SYSREAD Q1 = ID60 NR1 IDX0 ; TOOL NUMBER
3   FN 18: SYSREAD Q2 = ID60 NR2 IDX0 ; TOOL AXIS
4   FN 18: SYSREAD Q3 = ID60 NR3 IDX0 ; ROTATIONAL SPEED
5   FN 18: SYSREAD Q4 = ID60 NR4 IDX0 ; OVERSIZE FOR TOOL LENGTH DL
6   FN 18: SYSREAD Q5 = ID60 NR5 IDX0 ; OVERSIZE FOR TOOL RADIUS DR
7   FN 19: PLC=+Q1 / +0 ; INFO FOR PLC FOR PRE-POSITIONING THE MAGAZINE
8   LBL 5 ; CHECK WHETHER THE TOOL IS ALREADY IN THE SPINDLE
9   FN 18: SYSREAD Q18 = ID2000 NR60 IDX2301
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 ALREADY IN SPINDLE
12  FN 18: SYSREAD Q10 = ID1000 NR4210 IDX0 ; TOOL-CHANGE POSITION AXIS X
13  FN 18: SYSREAD Q11 = ID1000 NR4210 IDX2 ; TOOL-CHANGE POSITION 1 AXIS Y
14  FN 18: SYSREAD Q12 = ID1000 NR4210 IDX5 ; TOOL-CHANGE POSITION AXIS Z
15  FN 18: SYSREAD Q15 = ID1000 NR4210 IDX3 ; TOOL-CHANGE END POSITION AXIS Y
16  L X+Q10 Y+Q11 Z+Q12 R0 F MAX M91 ; MOVE TO TOOL-CHANGE POSITION
17  LBL 4 ; BYTE2300=1: SPINDLE AND MAGAZINE IN POSITION ?
18  FN 18: SYSREAD Q18 = ID2000 NR60 IDX2300
19  FN 10: IF +Q18 NE +1 GOTO LBL 4
20  L Y+Q15 R0 F MAX M91 ; TOOL IN TOOL CHANGER
21  L Y+Q11 M71 ; CLAMP THE TOOL AND RETURN TO TOOL-CHANGE POSITION
22  LBL 3
22  TOOL CALL Q1 Z SQ3 DL+Q4 DR+Q5 ; TOOL CALL WITH T-STROBE
23  M113 ; M112 SWITCH OFF
24  END PGM TCALL MM
```

Variable and fixed pocket coding

You can operate the system with either variable or fixed tool pocket coding. Machine parameter MP7480.x defines whether the tool number or the pocket number is transferred to the PLC. For variable pocket coding the pocket number must be transferred to the PLC (MP7480.x = 3 or 4). With fixed pocket coding it is advisable to work with the tool number (MP7480.x = 1 or 2).

When executing TOOL CALL and TOOL DEF blocks, depending on the setting of MP7480.x, the NC transfers either only the tool number of the programmed tool to the word W264, or the pocket number and the tool number to words W262 and W264, respectively. The NC sets the strobe marker M4073 (TOOL CALL) or M4074 (TOOL DEF). The NC does not reset these strobe markers until you set marker M4093 (TOOL CALL) or M4094 (TOOL DEF) after processing the tool or pocket number. The machining program is resumed when you acknowledge the strobe markers.

If tool number zero is processed, the NC sets marker M4521 and this is not reset until there is a TOOL CALL for another tool.

MP7480.0 Output of tool or pocket number with TOOL CALL block

Input: 0 = No output

1 = Tool number output only when tool number changes

2 = Tool number output with every TOOL CALL block

3 = Output of pocket number and tool number only when tool number changes

4 = Output of pocket number and tool number with every TOOL CALL block

5 = Output of pocket number and tool number only when tool number changes. Pocket table does not change.

6 = Output of pocket number (W262) and tool number (W264) with every TOOL CALL block. Pocket table does not change.

MP7480.1 Output of tool or pocket number with TOOL DEF block

Input:	0 =	No output
	1 =	Tool number output only when tool number changes
	2 =	Tool number output with every TOOL DEF block
	3 =	Output of pocket number and tool number only when tool number changes
	4 =	Output of pocket number and tool number with every TOOL DEF block

		Set	Reset
W262	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 no. zero programmed	NC	NC
M4538	Geometry of the tool from W264	PLC	NC

Output of tool number (fixed pocket coding)

Evaluating the tool number is adequate for fixed tool pocket coding. With MP7480.x you define whether the tool number should be transferred to the PLC with every TOOL CALL (TOOL DEF) block or only when the tool number changes (input values for MP7480.x = 2 or 1). With this setting the tool number is transferred to the word W262 when a TOOL CALL or TOOL DEF block is executed. W264 is not used. By entering 5 or 6 in MP7480.0 you can transfer the pocket number to W262 without changing the assignment of tool and pocket numbers in the pocket.

Output of pocket number (variable pocket coding)

With variable pocket coding (MP7480 = 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. In addition to the pocket number, the NC also transfers the current tool number in W264. Variable pocket management (the assignment of tool number to pocket number in the tool table) is handled by the NC. If you set M4542 the assignment of tool and pocket numbers in the pocket table is not changed, even though variable pocket coding is selected. You set this marker, for example, during a mid-program startup.

You define the number of tools with a pocket number in machine parameter MP7261. The input value for MP7261 matches the number of pockets in the tool magazine. This means that you can define more tools in the tool table than there is room for in the tool magazine [(MP7260) > (MP7261)]. If a tool number is programmed and no pocket is defined for it, pocket number –1 (W262) is transferred on TOOL CALL, and marker M4523 is set.

Only the tool number and the pocket number are transferred when TOOL DEF is programmed. A TOOL DEF for a manual tool has no relevance in the PLC. You can define a fixed pocket in the field "F" of the pocket table. Tools for which a fixed pocket is defined are returned to the same pocket despite variable coding.

		Set	Reset
M4520	Another T code (P code) follows with TOOL CALL 0 = Normal tool follows normal tool (N → N) or manual tool follows manual tool (M → M) or special tool follows special tool (S → S), when M4541 = 0 1 = Special tool follows manual tool (M → S) or special tool follows special tool (S → S), when M4541 = 1 or manual tool follows special tool (S → M) or manual tool follows normal tool (N → M) or normal tool follows manual tool (M → N) or normal tool follows special tool (S → N) see M4540	NC	NC
M4522	Tool programmed with pocket number (effective only with MP7480.0 = 3 or 4 and TOOL CALL)	NC	NC
M4523	Tool programmed without pocket number (effective only for MP7480.0 = 3 or 4 and TOOL CALL)	NC	NC
M4524	Call special tool (TOOL CALL)	NC	NC
M4525	TOOL CALL programmed or after expiration of tool life 0 = programmed TOOL CALL 1 = TOOL CALL after expiration of tool life	NC	NC
M4540	Sequence of tool numbers — 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 to original pocket despite variable pocket coding	PLC	PLC
M4542	Do not update pocket number in pocket table	PLC	PLC

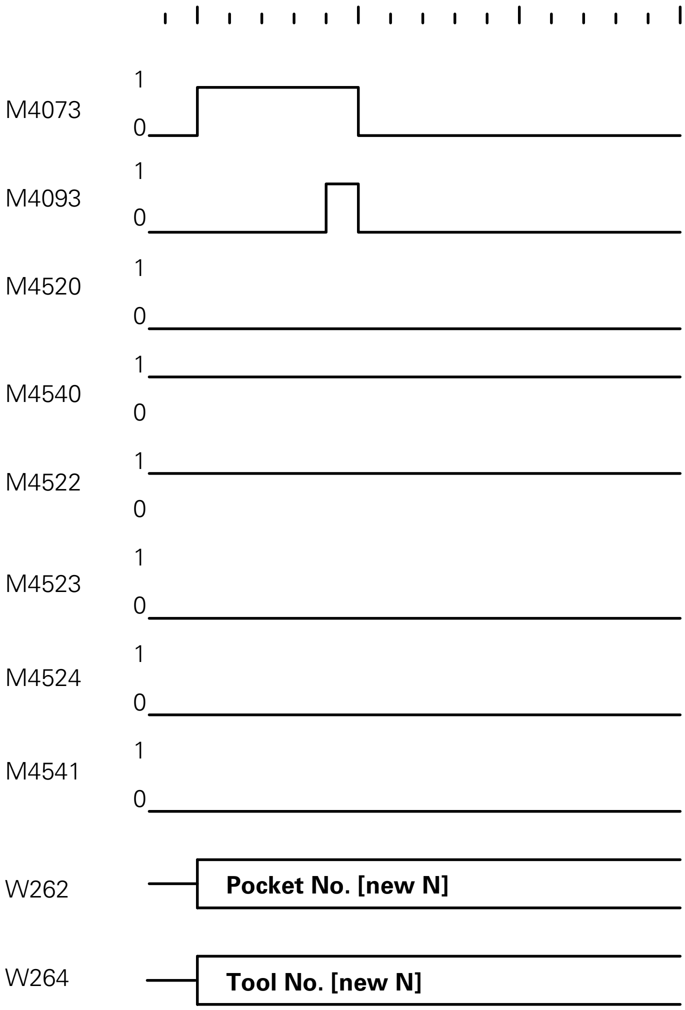
A variety of tool types can be called from the machining program. The following definitions are used in the examples that follow:

N = Tool for which a pocket number is defined in the tool table (**N**ormal).
M = Tool for which no pocket number is defined in the tool table. You must change these tools by hand (**M**anual).
S = **S**pecial tool (defined in tool table).

Nine different combinations of tool-change sequence are therefore possible. For many tool-change sequences, for the tool magazine to be controlled, two pocket numbers (tool numbers) must be output in succession for one TOOL CALL. You can recognize this with M4520 and M4540. You must evaluate and acknowledge both pocket numbers (tool numbers).

The logic diagrams for the nine different tool-change sequences are shown on the following pages (activated by TOOL CALL).

N → N: Normal tool follows normal tool
The pocket number and tool number of the called tool are transferred.



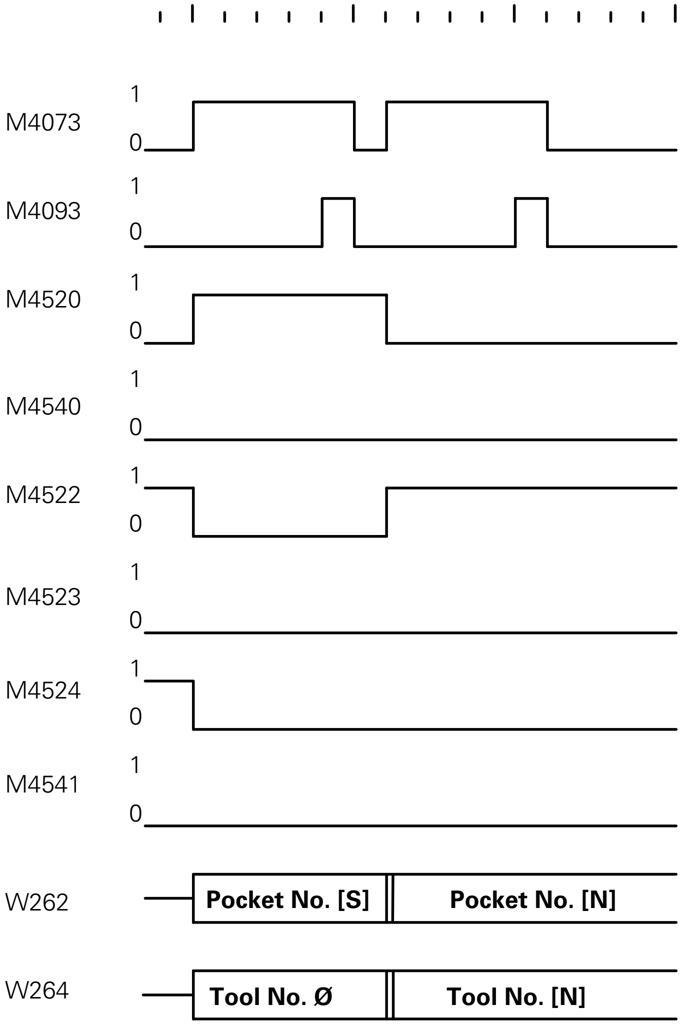
S → N: Normal tool follows Special tool

With this tool change, two pocket numbers (or tool numbers) must be transferred in succession. M4520 indicates that another TOOL CALL strobe (M4073) follows.

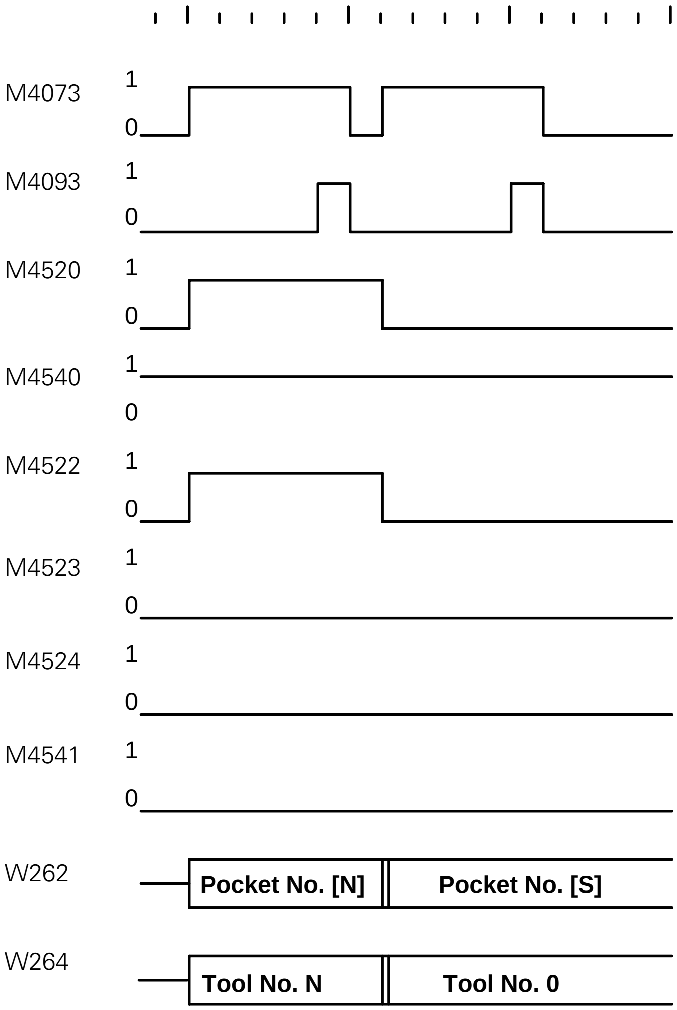
With M4540 you can determine the sequence in which the pocket numbers are transferred. The decision will depend on whether you are using a single or double changing arm.

S → N, Single Changing Arm (M4540 = 0):

The pocket number of the old tool and tool number zero are transferred first. Tool number zero is your indication to clear the spindle. After you have acknowledged with M4093 the pocket number and tool number of the new tool are transferred.



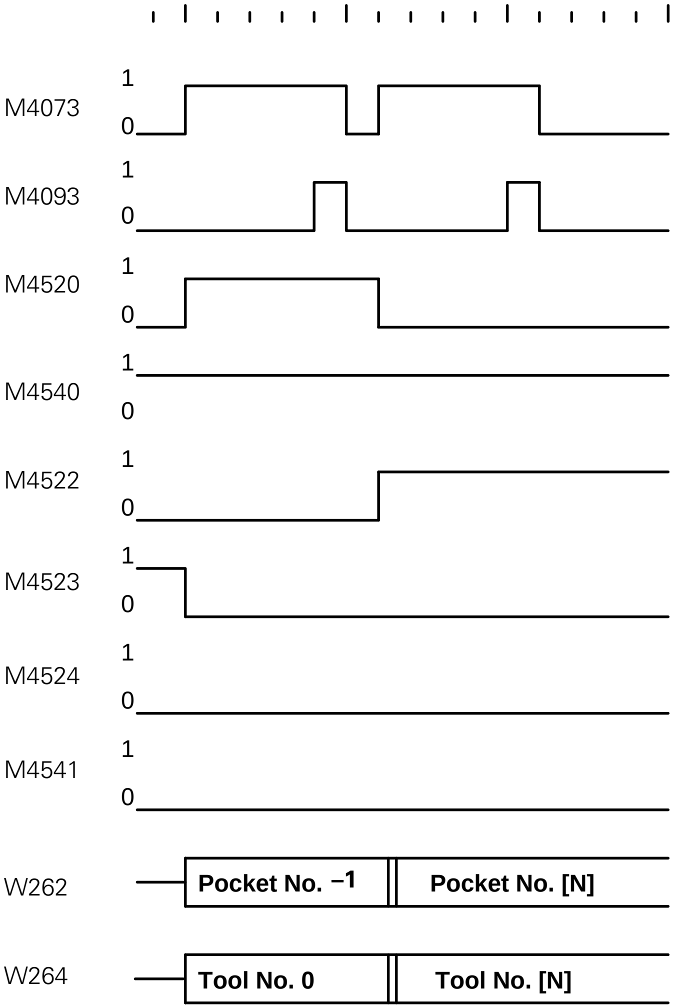
S → N, Double Changing Arm (M4540 = 1):
The pocket number and tool number of the new tool are transferred first. After you have acknowledged with M4093, the pocket number of the old tool and tool number zero are transferred. Tool number zero is your indication to clear the spindle.



M → N: Normal tool follows Manual tool

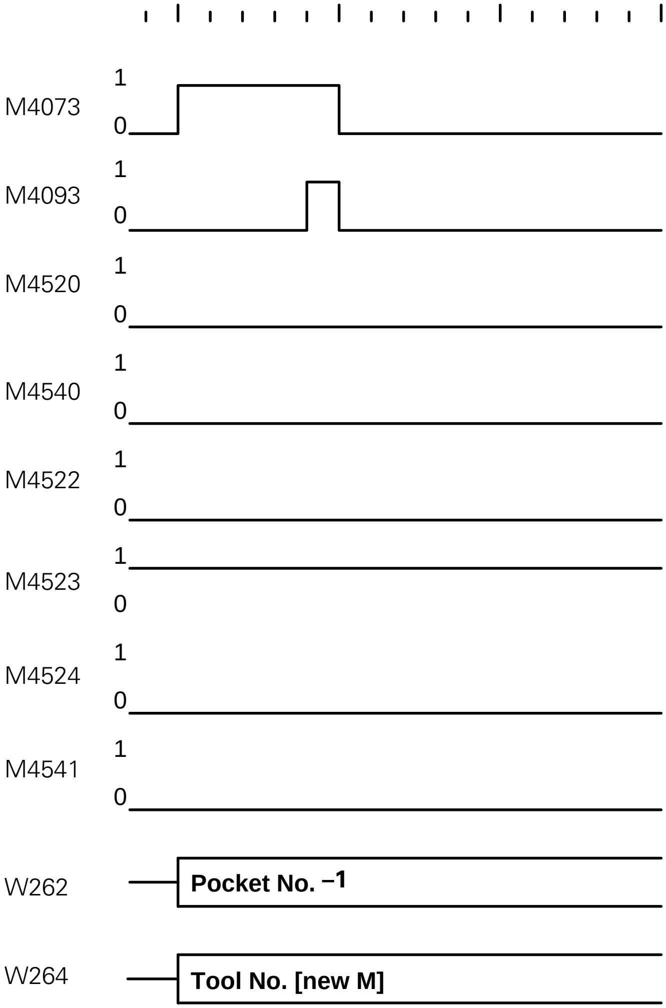
With this tool-change sequence two pocket numbers (or tool numbers) must be transferred in succession. M4520 indicates that another TOOL CALL strobe (M4073) follows.

Irrespective of marker M4540, pocket number –1 and tool number zero are transferred first. Tool number zero tells you to clear the spindle. Pocket number –1 means that there is no pocket in the tool magazine for the called tool. After you have acknowledged with M4093, the pocket number and tool number of the new tool (called tool) are transferred.



M → M: Manual tool follows Manual tool

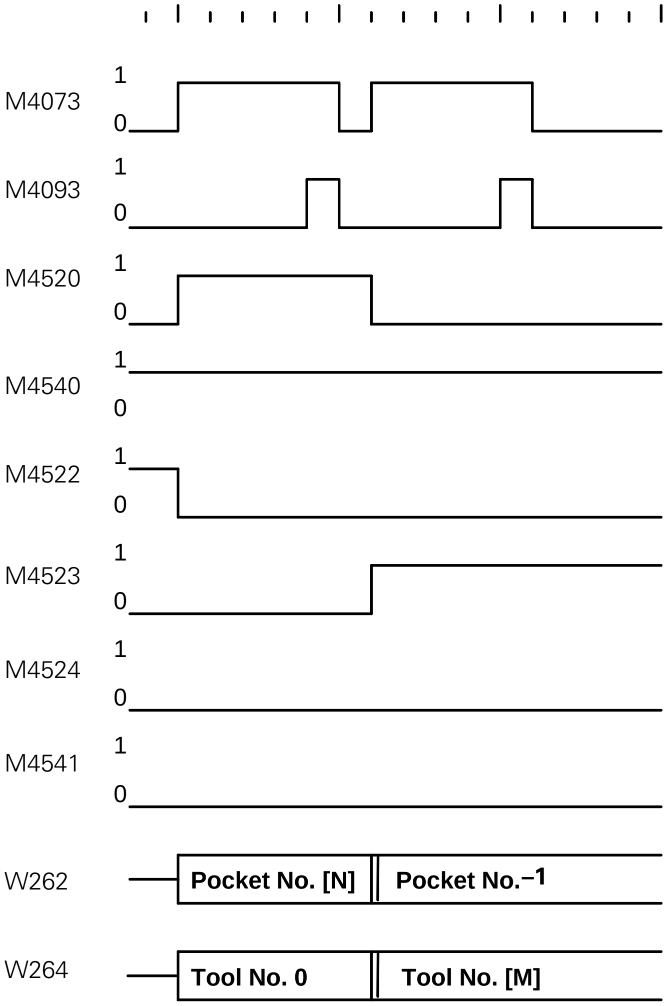
The pocket number –1 tells you that there is no pocket in the tool magazine for the called tool.



N → M: Manual tool follows Normal tool

With this tool-change sequence two pocket numbers (or tool numbers) must be transferred in succession. M4520 indicates that another TOOL CALL strobe (M4073) follows.

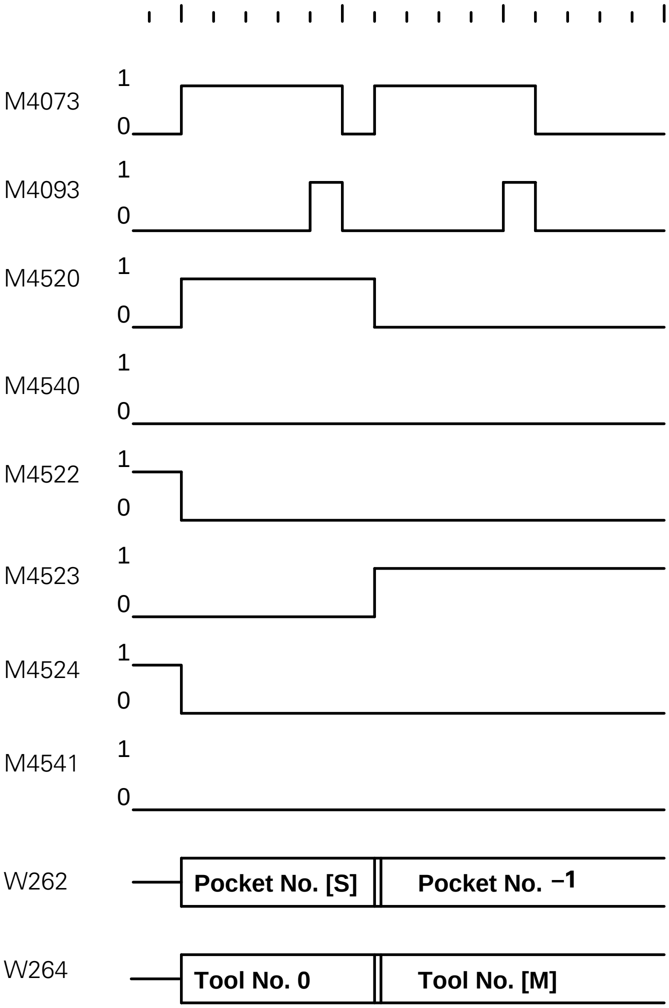
Irrespective of marker M4540, the pocket number of the old tool and tool number zero are transferred first. Tool number zero is your indication to clear the spindle. After acknowledging with M4093, pocket number –1 and the tool number of the called tool are transferred. Pocket number –1 tells the PLC that there is no pocket in the tool magazine for the called tool.



S → M: Manual tool follows Special tool

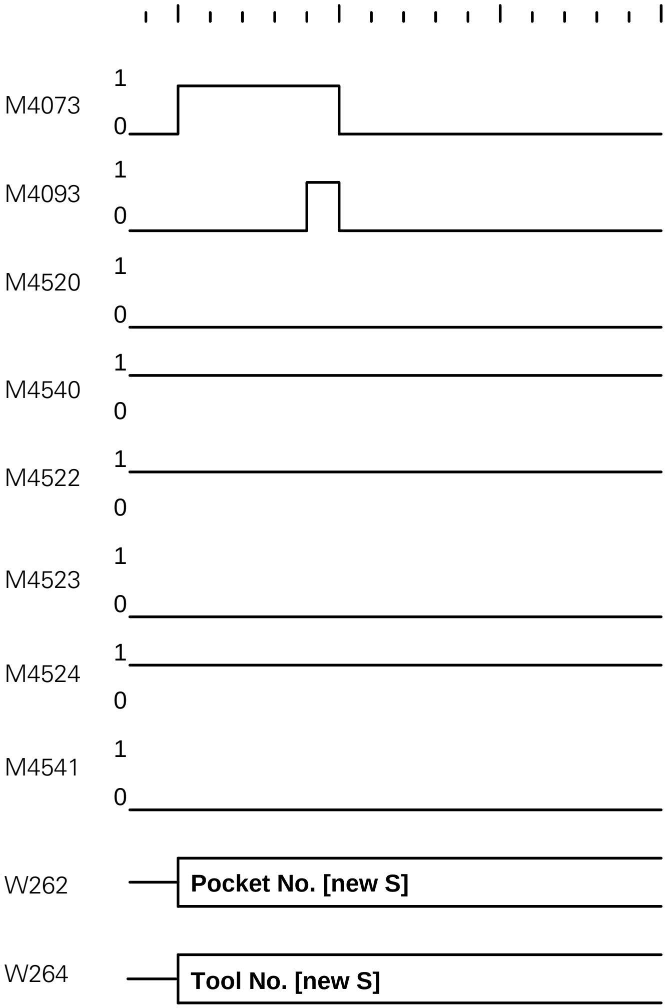
With this tool-change sequence two pocket numbers (or tool numbers) must be transferred in succession. M4520 indicates to you that another TOOL CALL strobe (M4073) follows.

Irrespective of marker M4540, the pocket number of the old tool and tool number zero are transferred first. Tool number zero is your indication to clear the spindle. After acknowledging with M4093, pocket number –1 and the tool number of the called tool are transferred. Pocket number –1 tells you that there is no pocket in the tool magazine for the called tool.



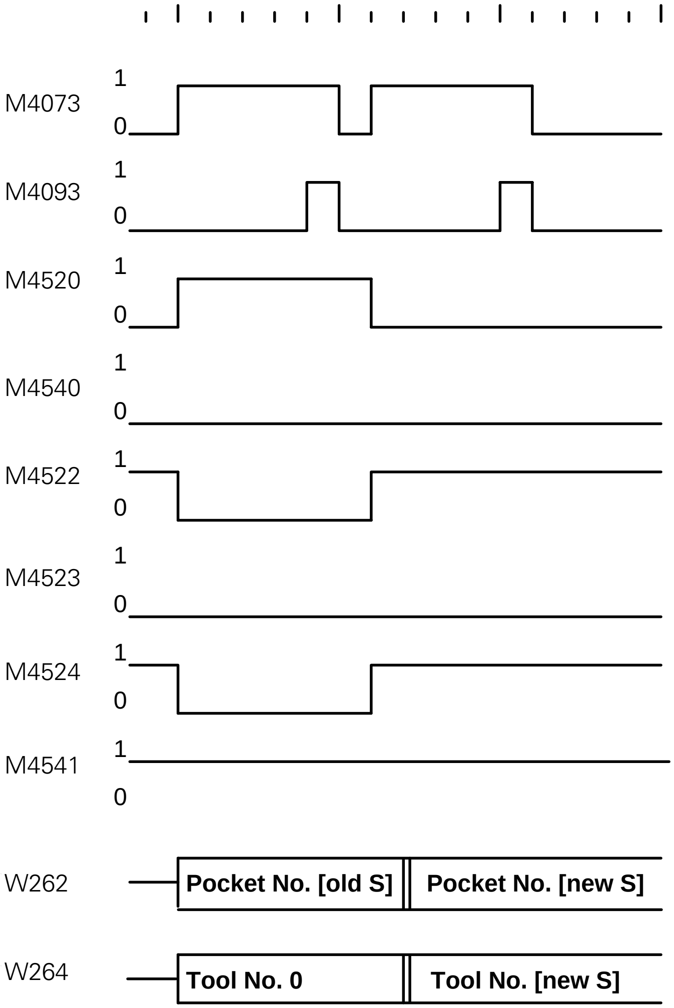
S → S: Special tool follows Special tool

With marker M4541or field "F" in the pocket table you can set whether the special tool is to be returned to its original pocket despite variable pocket coding. With variable pocket coding for special tools (M4541 = 0), the same logic diagram applies to single as to double changing arms (M4540 = 0 and 1).

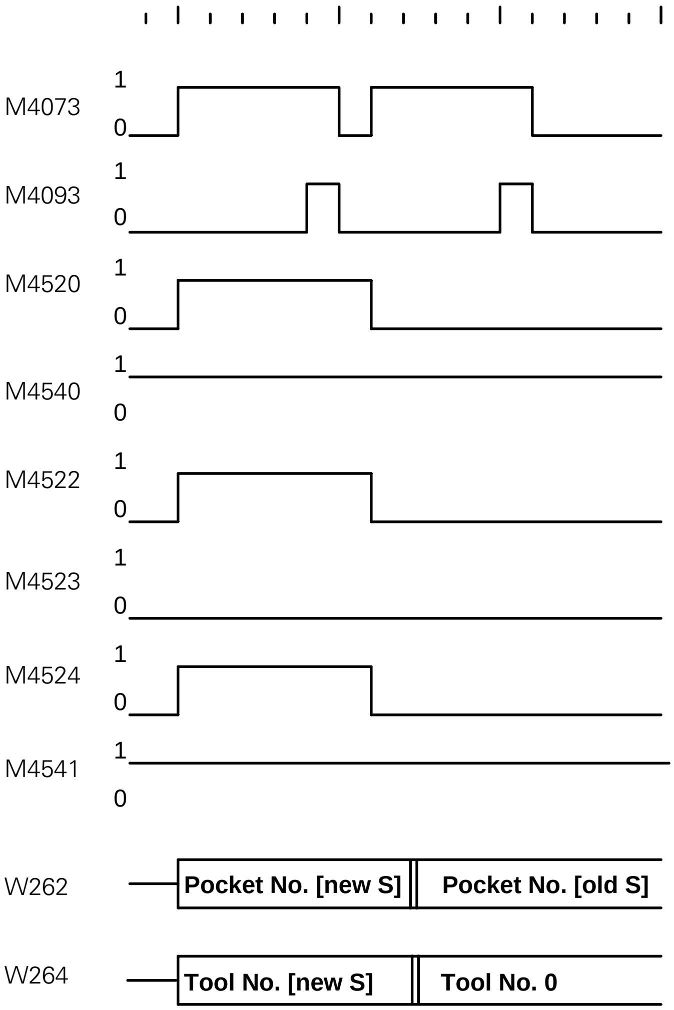


If the special tool is to return to its original pocket despite variable pocket coding (M4541 = 1), there is a different pocket number transfer sequence for single and double changing arms (M4540).

S → S, Single Changing Arm (M4540 = 0)
The pocket number of the old tool and tool number zero are transferred first. Tool number zero tells the PLC to clear the spindle. After you have acknowledged with M4093, the pocket number and tool number of the new tool are transferred.



S → S, Double Changing Arm (M4540 = 1)
The pocket number and tool number of the new tool are transferred first. After you have acknowledged with M4093, the pocket number of the old tool and tool number zero are transferred. Tool number zero is your indication to clear the spindle.



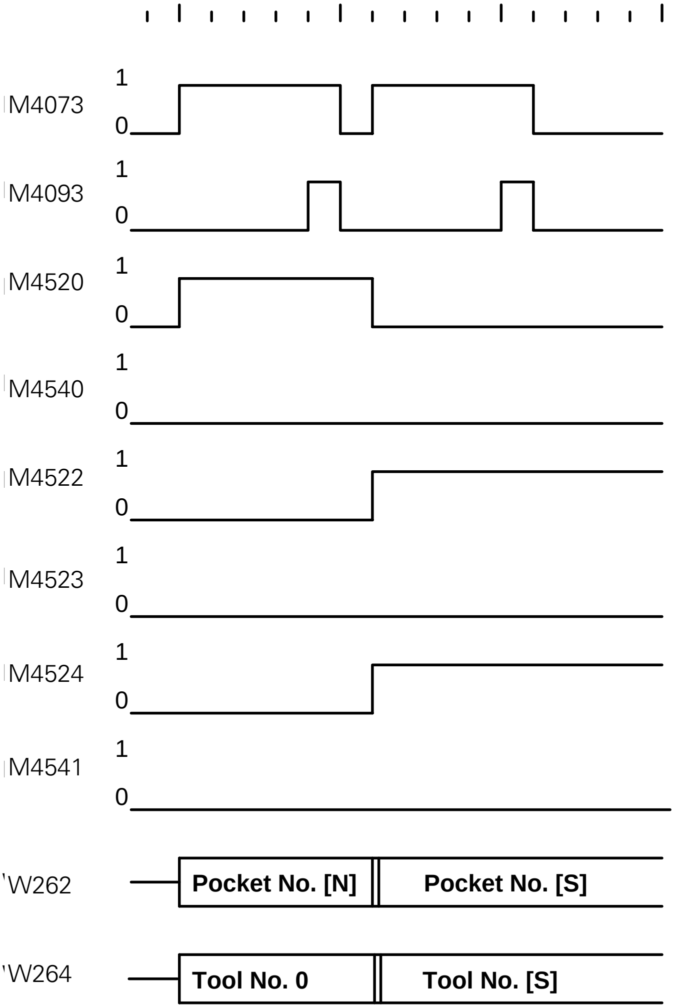
N → S: Special tool follows Normal tool

With this tool-change sequence two pocket numbers (or tool numbers) must be transferred in succession. M4520 indicates to you that another TOOL CALL strobe (M4073) follows.

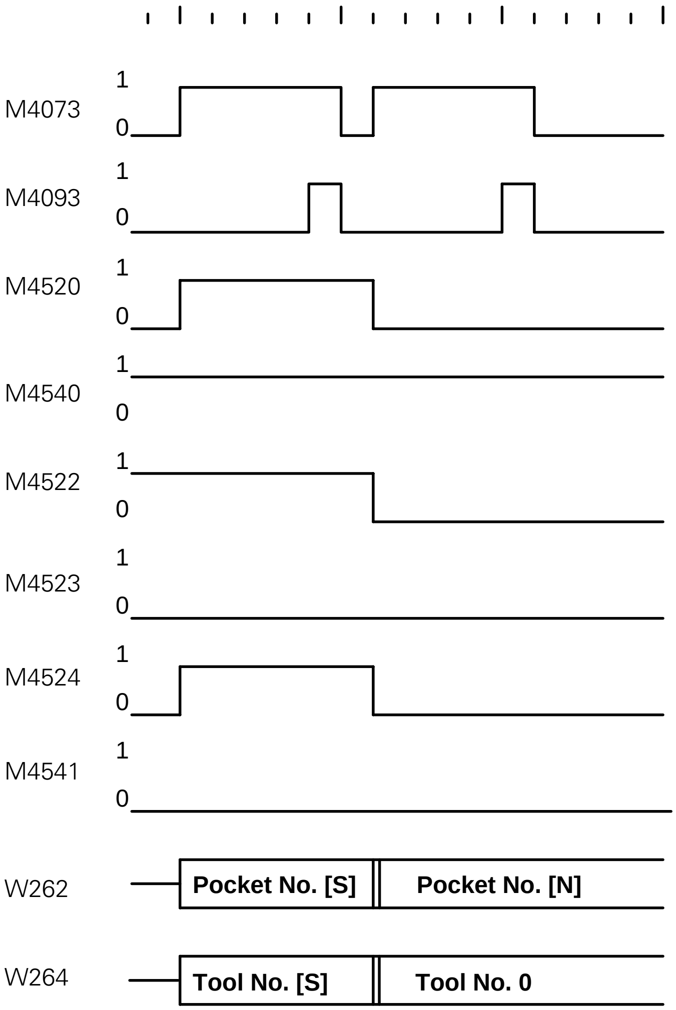
There is a different pocket number transfer sequence depending on M4540 (single/double changing arm). M4541 is not relevant.

N → S, Single Changing Arm (M4540 = 0)

The pocket number of the old tool and tool number zero are transferred first. Tool number zero tells the PLC to clear the spindle. After you have acknowledged with M4093, the pocket number and tool number of the new tool are transferred.

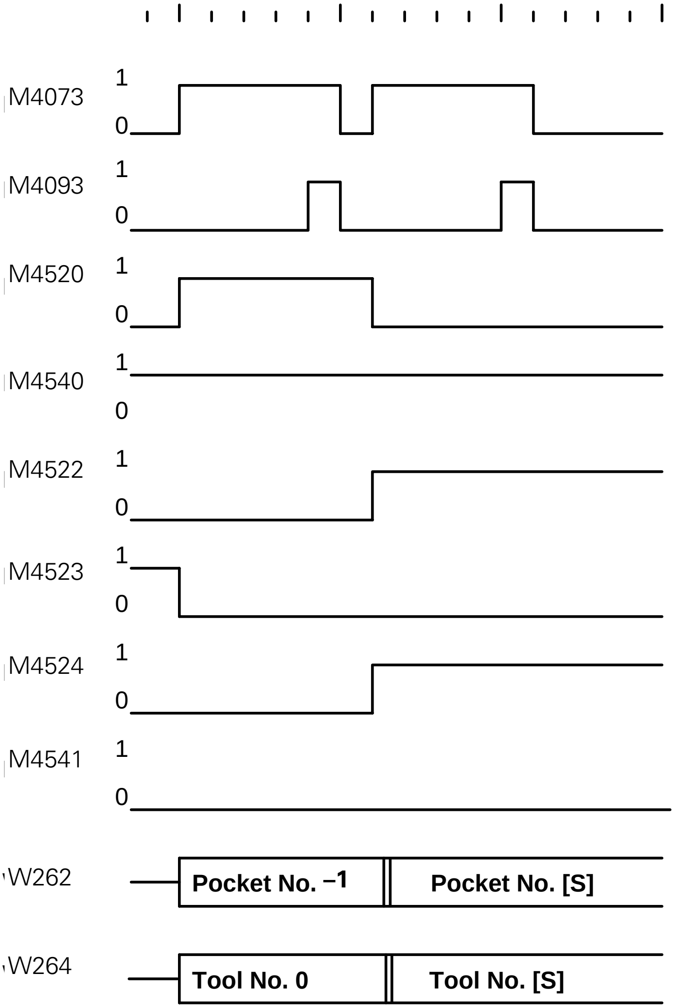


N → S, Double Changing Arm (M4540 = 1)
The pocket number and tool number of the new tool are transferred first. After you have acknowledged with M4093, the pocket number of the old tool and tool number zero are transferred. Tool number zero is your indication to clear the spindle.



M → S: Special tool follows Manual tool
With this tool-change sequence two pocket numbers (or tool numbers) must be transferred in succession. M4520 indicates to you that another TOOL CALL strobe (M4073) follows.

Irrespective of markers M4540 and M4541, pocket number –1 and tool number zero are transferred first. Tool number zero is your indication to clear the spindle. Pocket number –1 means that there is no pocket in the tool magazine for the called tool. After you have acknowledged with M4093, the pocket number and tool number of the new tool (called tool) are transferred.

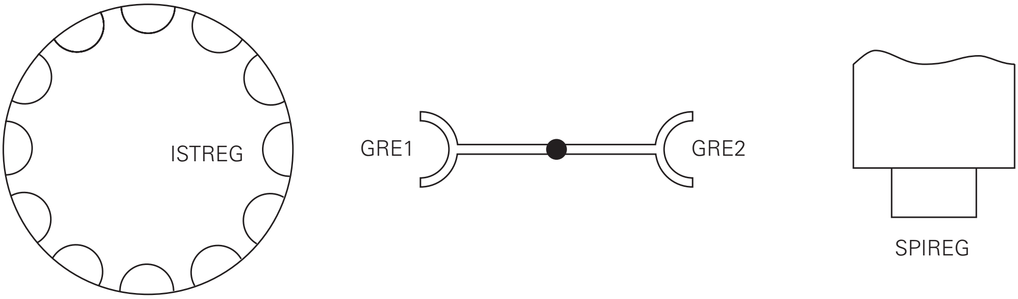


4.21.5 PLC Programming Example

This section describes a tool changer and contains the basic sequence diagrams of the corresponding PLC programs. The most effective way to create the PLC program is with the PLC programming software PLC.EXE.

The tool changer treated here has the following features:

- Up to 254 tools
- Variable pocket coding (MP7480.x = 4)
- Special tools are permitted
- Next tool standby with TOOL DEF
- Tool change with TOOL CALL
- Tools with no pocket number defined in the tool table can be changed by hand.
- Double changing arm
- Special tools variable (M4541 = 0)



The sequence diagram below uses the following variables for greater clarity:

- ISTREG = 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 tool in changing arm facing spindle
- SPIREG = Pocket number of tool in spindle

Other addresses that are used:

		Set	Reset
W262	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 T code (P code) with TOOL CALL	NC	NC
M4094	Acknowledgment T code (P code) with TOOL DEF	NC	NC
M4520	Another T code (P code) follows with TOOL CALL	NC	NC
M4524	Special tool called (TOOL CALL)	NC	NC
M4540	Sequence of tool number or pocket number transfer (M4520 = 1)	PLC	PLC
M4541	Special tool to original pocket despite variable pocket coding	PLC	PLC

Machines parameter that are used:

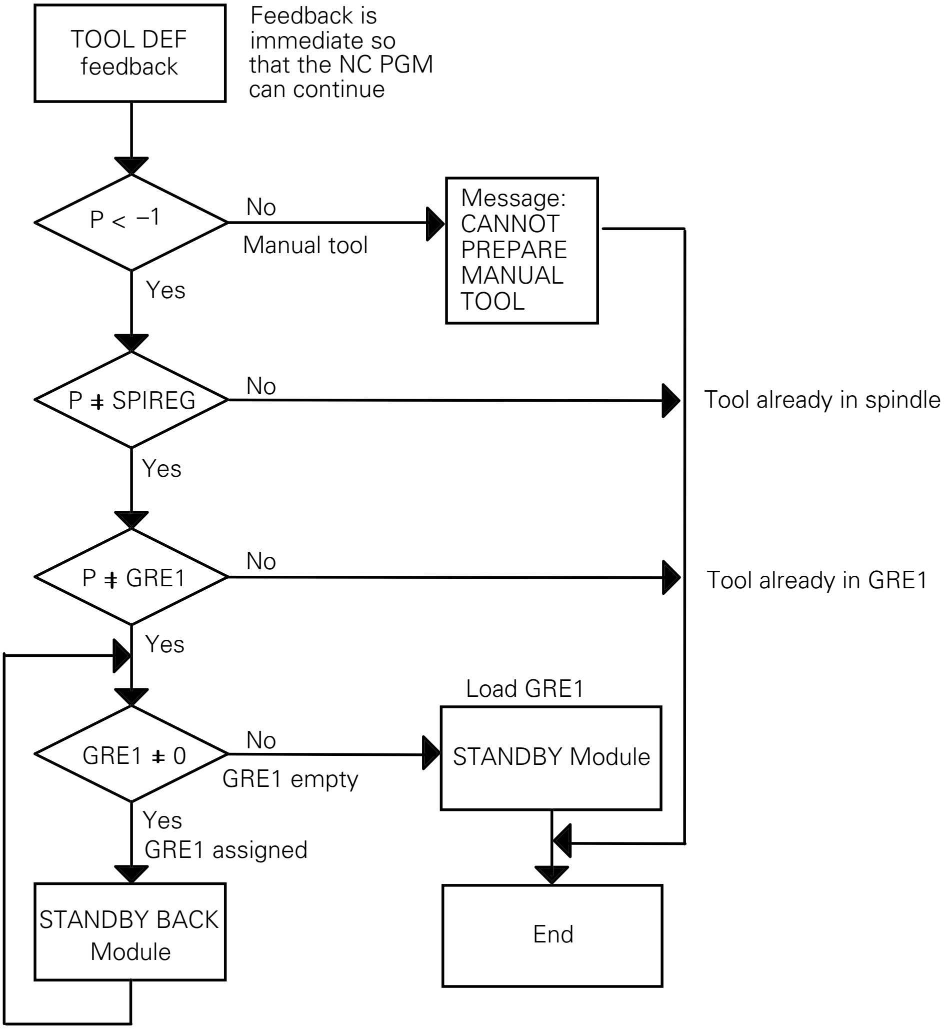
MP7260 = 90	Number of tools in tool table
MP7261 = 12	Number of pockets in tool magazine
MP7480.0 = 4	Output of pocket number and tool number with every TOOL CALL block
MP7480.1 = 4	Output of pocket number and tool number with every TOOL DEF block

The sequence diagram for this tool changer is subdivided into modules.
List of modules (subroutines):

• TOOL DEF	Search tool and load in GRE1
• TOOL CALL	Automatic tool-change
• STANDBY	Search tool and load in GRE1
• STANDBY BACK	Return tool from GRE1 to magazine
• MANUAL TOOL IN	Manual tool follows Normal or Special tool
• MANUAL TOOL OUT	Normal or Special tool follows Manual tool
• MANUAL OUT/IN	Manual tool follows Manual tool
• INSERT	Take old tool out and put new tool in
• COMPUTE SHORTEST PATH	
• COMPARE P-CODE WITH ISTREG	
• COMPARE GRE1 WITH ISTREG	

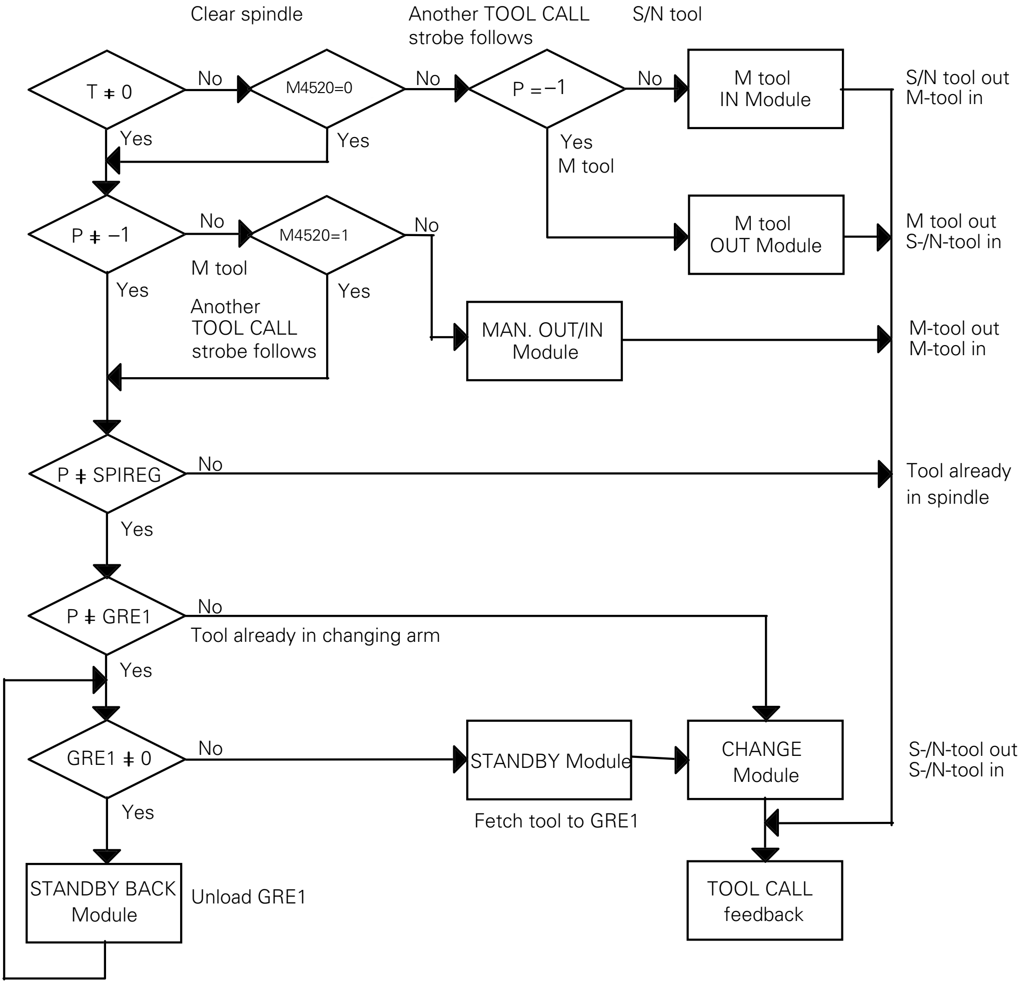
Program Module TOOL DEF

Look for tool and load in GRE1

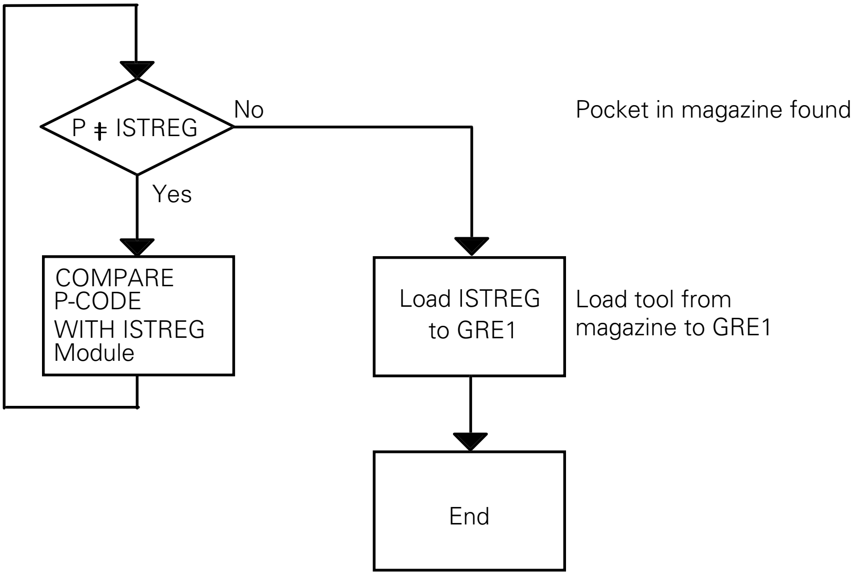


Program Module TOOL CALL

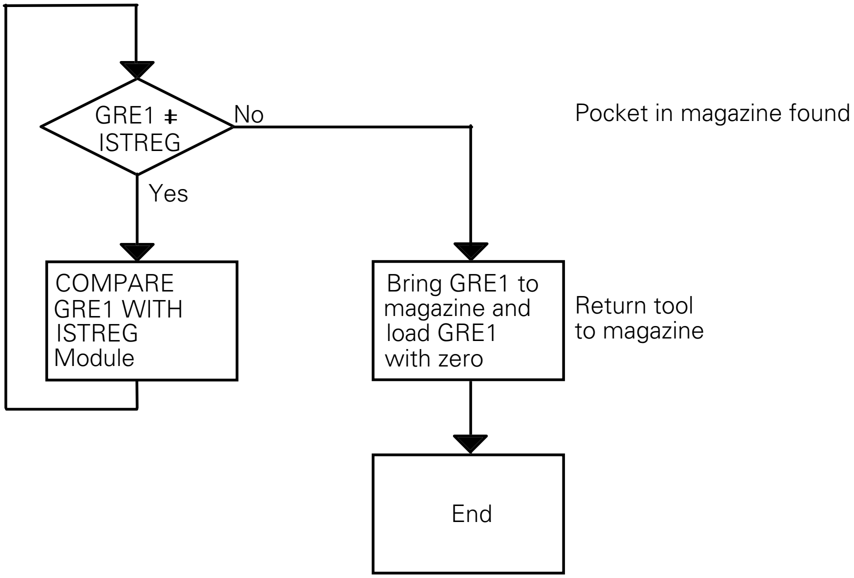
Automatic tool change (main program)



Program Module STANDBY
Look for tool and bring it into GRE1

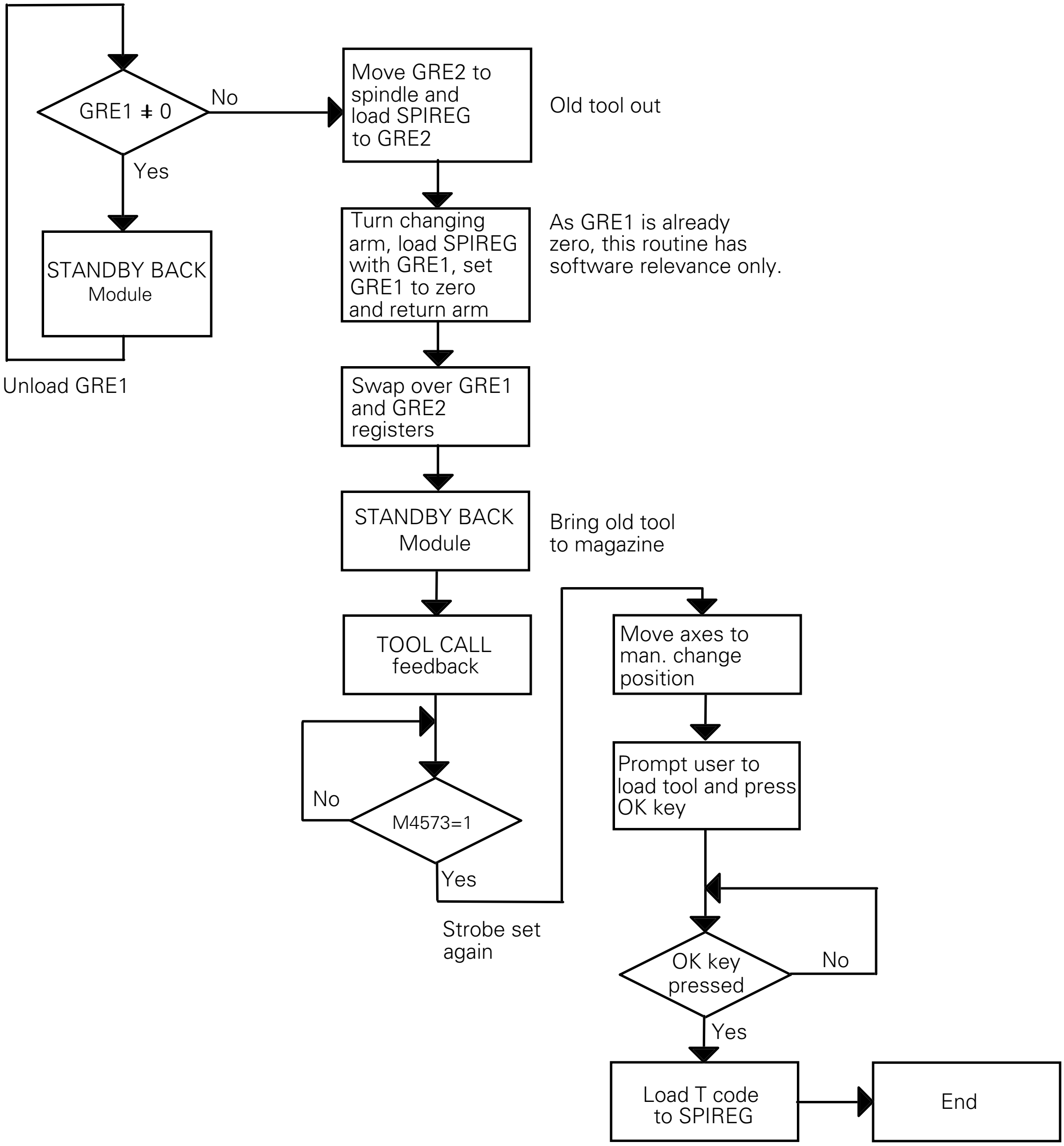

P ≠ ISTREG
No
Yes
COMPARE P-CODE WITH ISTREG Module
Load ISTREG to GRE1
End
Pocket in magazine found
Load tool from magazine to GRE1

Program Module STANDBY BACK
Tool back from GRE1 into tool magazine


GRE1 ≠ ISTREG
No
Yes
COMPARE GRE1 WITH ISTREG Module
Bring GRE1 to magazine and load GRE1 with zero
End
Pocket in magazine found
Return tool to magazine

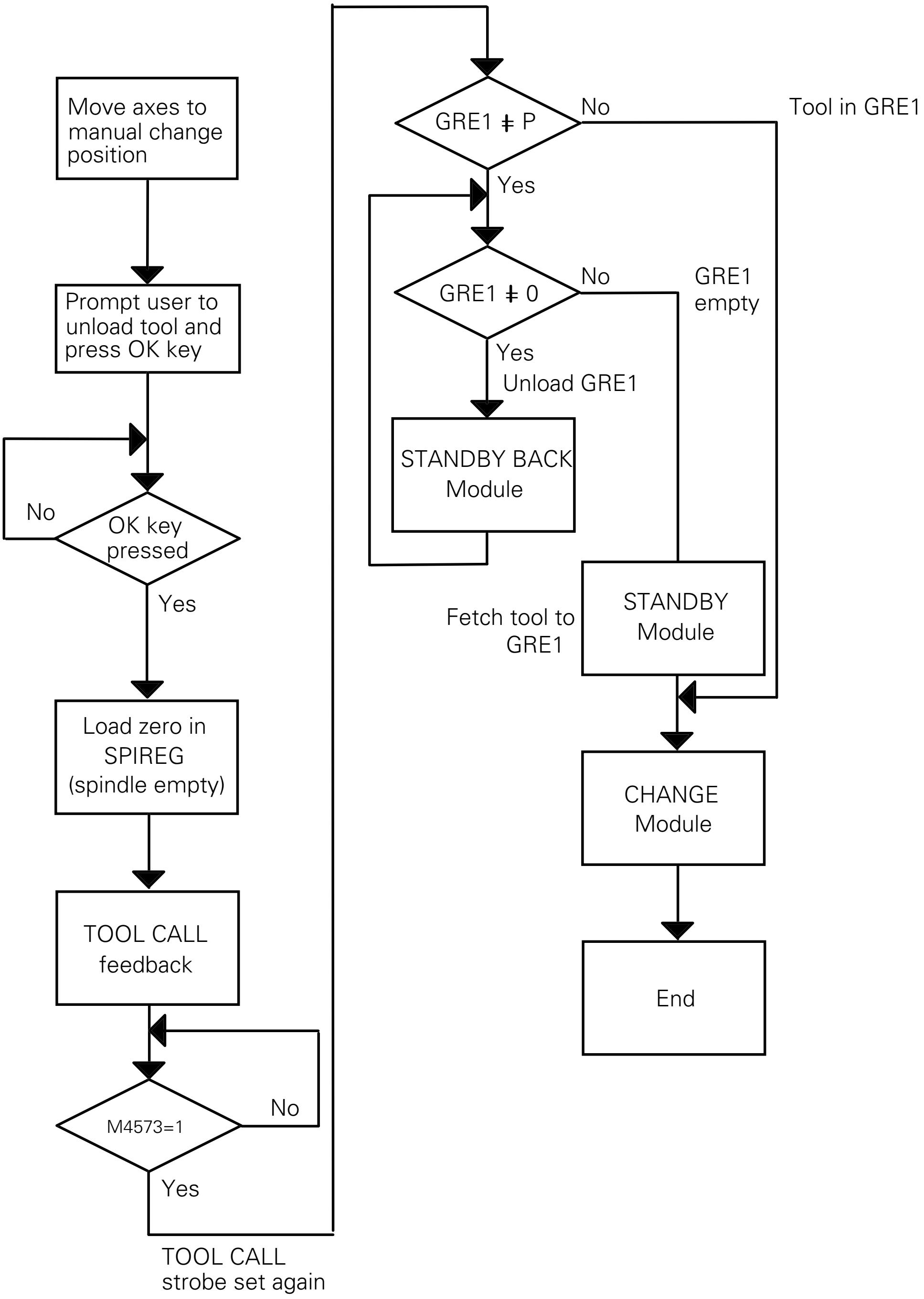
Program Module MANUAL TOOL IN

N → M or S → M: Manual tool follows normal tool or special tool. The old tool is placed in the tool magazine and the operator is prompted to insert a manual tool that is not in the tool magazine.


GRE1 ≠ 0
No
Yes
STANDBY BACK Module
Unload GRE1
Move GRE2 to spindle and load SPIREG to GRE2
Old tool out
Turn changing arm, load SPIREG with GRE1, set GRE1 to zero and return arm
As GRE1 is already zero, this routine has software relevance only.
Swap over GRE1 and GRE2 registers
STANDBY BACK Module
Bring old tool to magazine
TOOL CALL feedback
No
M4573=1
Yes
Strobe set again
Move axes to man. change position
Prompt user to load tool and press OK key
OK key pressed
No
Yes
Load T code to SPIREG
End

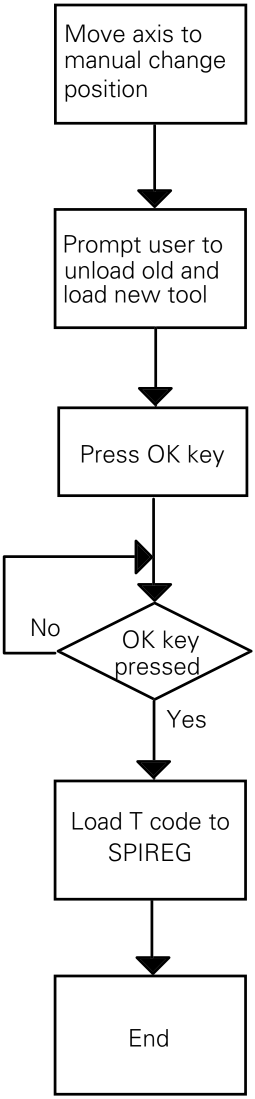
Program Module MANUAL TOOL OUT

M → N or M → S: A normal or special tool follows a manual tool. The operator is prompted to empty the spindle manually, since there is no room in the tool magazine for the current tool. The called tool is automatically inserted.



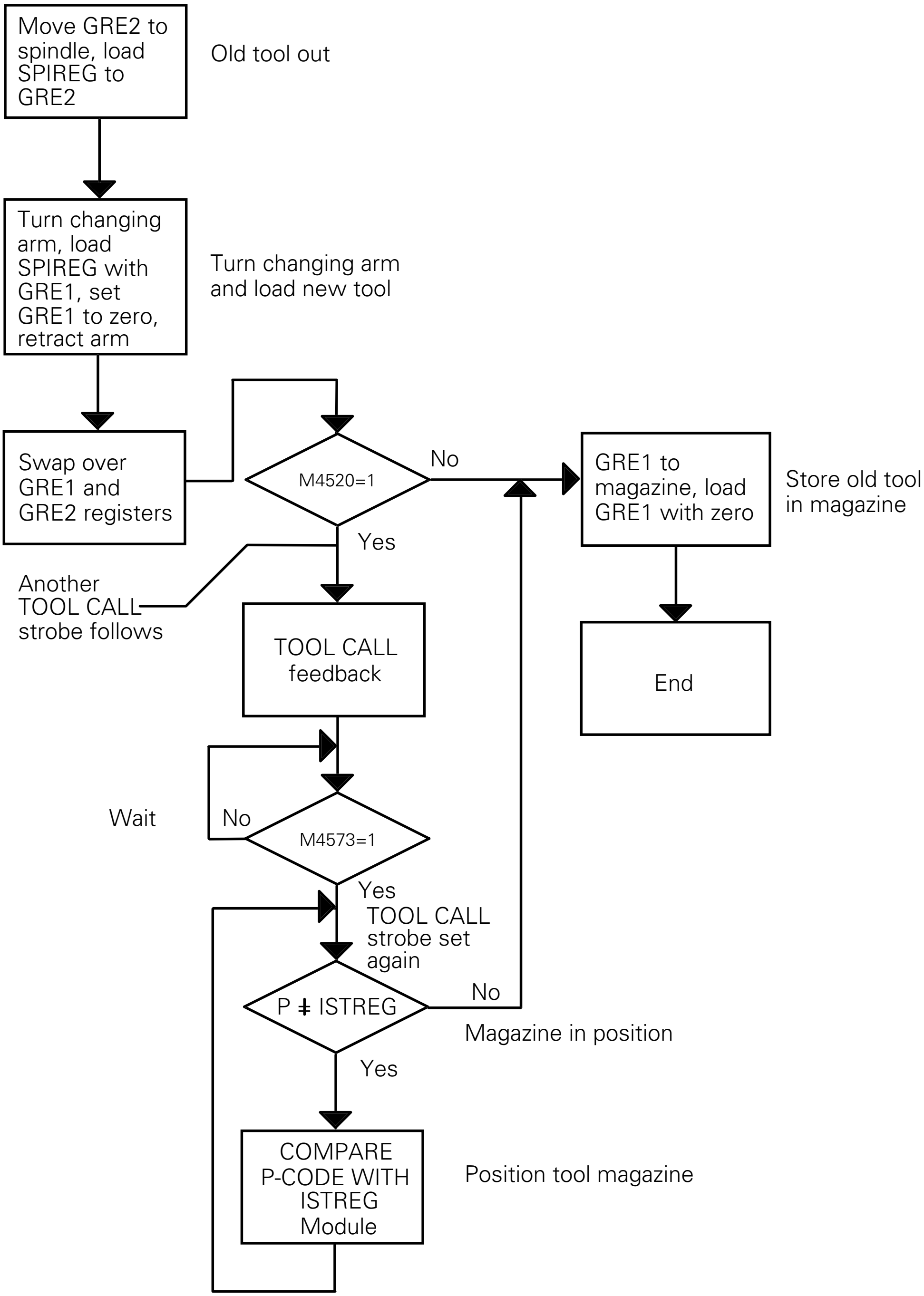
Program Module MANUAL TOOL OUT/IN

M → M: A manual tool follows a manual tool. The operator is prompted to manually empty the spindle and insert the new tool, since there are no pockets for the tools in the magazine.


Move axis to manual change position
Prompt user to unload old and load new tool
Press OK key
No
OK key pressed
Yes
Load T code to SPIREG
End

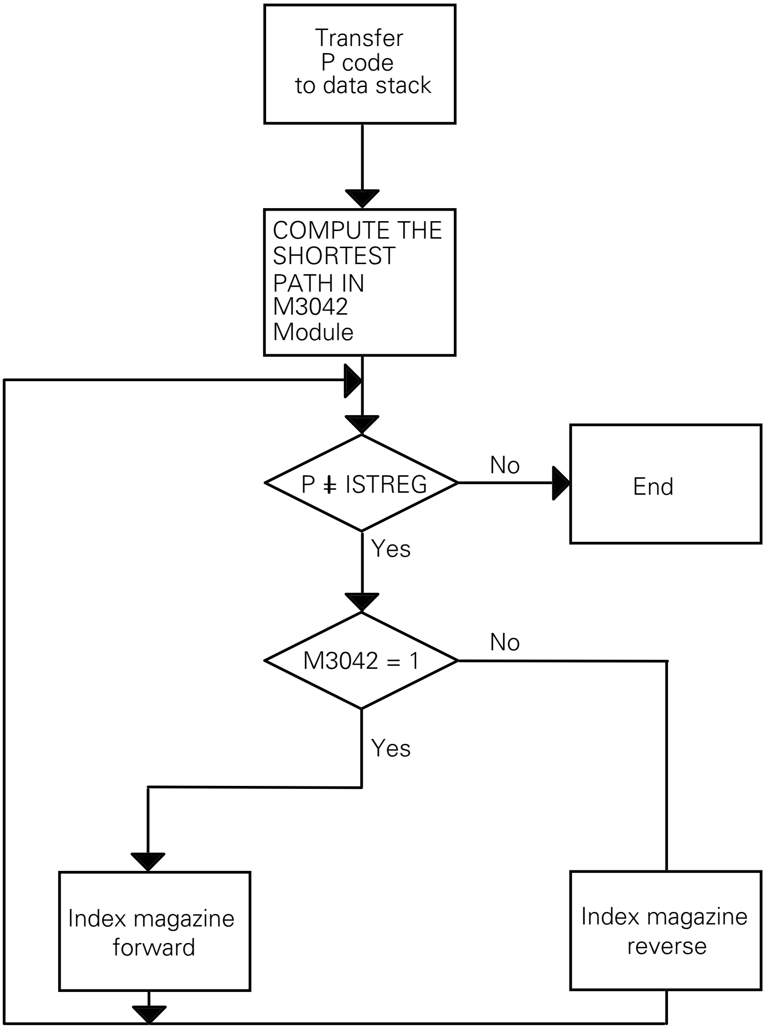
Program Module INSERT

The spindle is emptied and the new tool is automatically inserted. The program takes into account whether the tool should be returned to its old pocket in the tool magazine (e.g. special tool).


Move GRE2 to spindle, load SPIREG to GRE2
Old tool out
Turn changing arm, load SPIREG with GRE1, set GRE1 to zero, retract arm
Turn changing arm and load new tool
Swap over GRE1 and GRE2 registers
M4520=1
No
Yes
GRE1 to magazine, load GRE1 with zero
Store old tool in magazine
Another TOOL CALL strobe follows
TOOL CALL feedback
End
Wait
No
M4573=1
Yes
TOOL CALL strobe set again
P ≠ ISTREG
No
Magazine in position
Yes
COMPARE P-CODE WITH ISTREG Module
Position tool magazine

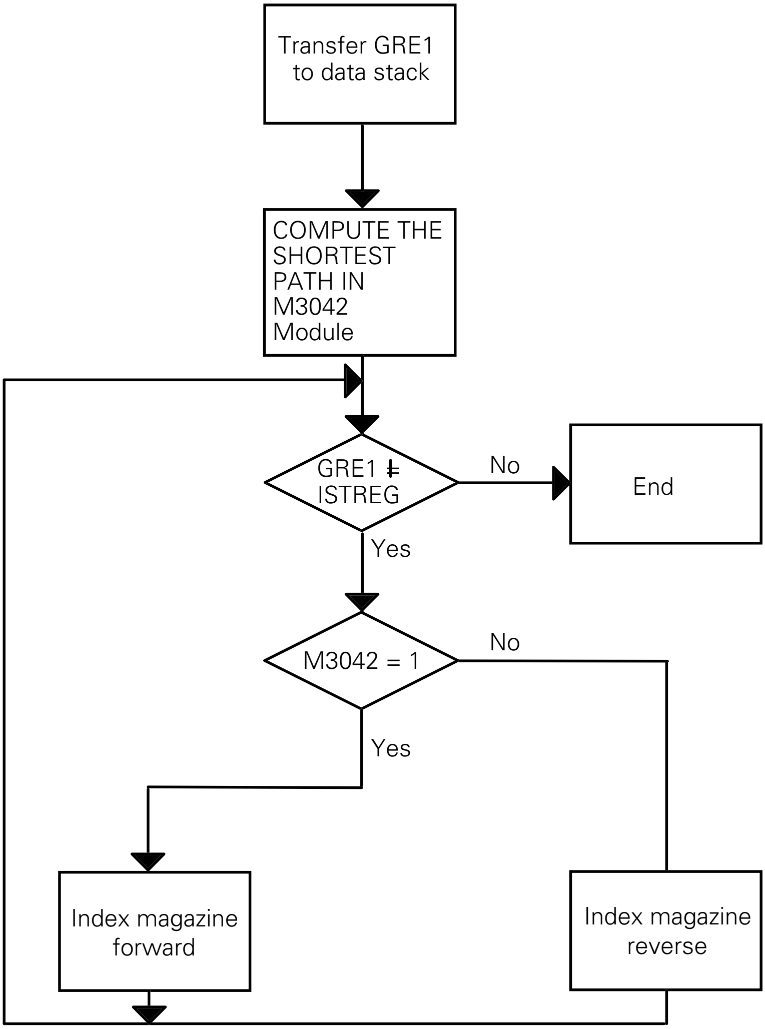
Program Module COMPARE P CODE WITH ISTREG

The tool magazine is positioned in the shortest direction to the desired pocket number.



Program Module COMPARE GRE1 WITH ISTREG

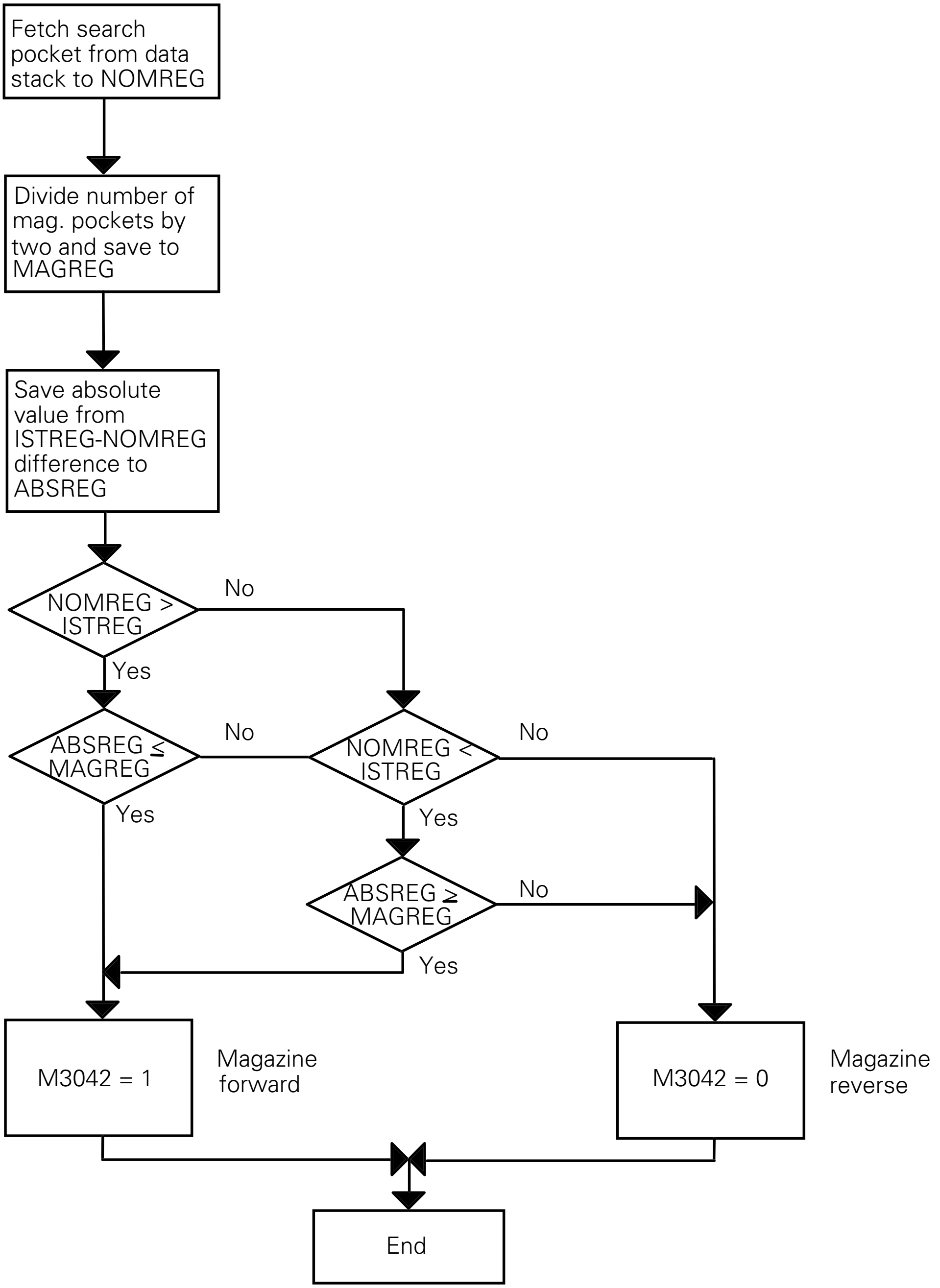
The tool magazine is positioned in the shortest direction to the pocket number of the tool that is located in GRE1.



Program Module CALCULATE SHORTEST DIRECTION

The program calculates the tool magazine traverse direction for the shortest distance to the desired pocket number. The direction is stored in marker M3042.

M3042 = 0: backward
M3042 = 1: forward


Fetch search pocket from data stack to NOMREG
Divide number of mag. pockets by two and save to MAGREG
Save absolute value from ISTREG-NOMREG difference to ABSREG
NOMREG > ISTREG
No
Yes
ABSREG ≤ MAGREG
No
Yes
NOMREG < ISTREG
No
Yes
ABSREG ≥ MAGREG
No
Yes
M3042 = 1
Magazine forward
M3042 = 0
Magazine reverse
End

4.22 Special Functions for Laser Cutting Machines

You can activate special functions to interface the TNC to laser cutting machines and water jet machines.

4.22.1 Analog Voltage Output

If you do not need the analog output (X8, pin 8) for the spindle, you can define with MP3011 another function for this analog output. The input values in MP3011 show an effect only if the value in MP3010 is less than 3.

MP3011 Function of analog output S, if MP3010 < 3
Input: 0 = No special function
1 = Voltage proportional to current contour feed rate (depending on MP3012)
2 = Voltage as defined by Module 9130
3 = Voltage is defined via M function (M200 to M204)

Voltage proportional to the feed rate (MP3011 = 1)
A voltage proportional to the current contour feed rate is output. In MP3012 you enter the feed rate achieved when 10 V is output.

MP3012 Feed rate for 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 issued.

Definition of the voltage via M functions (MP3011 = 3)
The analog voltage output can be defined in the positioning block with the miscellaneous functions M200 to M204. These M functions are available only if you have entered the value 3 in MP3011. The M functions are executed synchronously to the positioning blocks and are effective at the beginning of the block.

Direct output of the programmed voltage: M200 V...
The TNC outputs the value programmed behind M200 V... as a voltage.
Input: 0 to 9.999 [V]
Duration: M201 V... is effective unit a new voltage is output through M200 to M204.

Voltage varies with distance: M201 V...
The TNC outputs the voltage as a function of the traversed distance. The TNC increases or decreases the voltage linearly from the active voltage to the value programmed behind M201 V...
Input: 0 to 9.999 [V]
Duration: M201 V... is effective until a new voltage is output through M200 to M204.

Voltage varies with the velocity: M202 FNR.

The TNC outputs the voltage as a function of the velocity. With machine parameters MP3013.x and MP3014.x you can define up to three characteristic curves in a table. In the table certain analog voltages are assigned to certain feed rates. M202 FNR. selects the characteristic curve in which the TNC finds the voltage to be output.

Input: 1 to 3
Duration: M202 FNR. is effective until a new voltage is output through M200 to M204.

You can enter in the table up to four kink points per characteristic curve. The output values are interpolated linearly between the kink points. The first kink point must have the value zero. The entry values of the following kink points must increase in sequence. The TNC recognizes the beginning of a new characteristic curve from the entry value zero.

Example:

Velocity		**Voltage**		**Characteristic**
MP3013.0	0	MP3014.0	0	1
MP3013.1	25	MP3014.1	0	1
MP3013.2	500	MP3014.2	4.5	1
MP3013.3	1000	MP3014.3	9.999	1
MP3013.4	0	MP3014.4	0	2
MP3013.5	10 000	MP3014.5	9.999	2
MP3013.6	0	MP3014.6	9	3
MP3013.7	50	MP3014.7	0.5	3
MP3013.8	300	MP3014.8	1.5	3
MP3013.9	5000	MP3014.9	9.999	3
MP3013.10	0	MP3014.10	0	Not used
MP3013.11	0	MP3014.11	0	Not used

MP3013.0-11 Characteristic kink points (velocity) for analog voltage output with M202
Input: 10 to 300 000 [mm/min]

MP3014.0-11 Characteristic kink points (voltage) for analog voltage output with M202
Input: 0.000 to 9.999 [V]

Voltage varies with the time (time-voltage ramp): M203 V... TIME...

The TNC outputs the voltage as a function of the time. It increases or decreases the voltage linearly in the time programmed behind TIME from the current voltage to the voltage value programmed behind V...

Input: Voltage V: 0 to 9.999 [V]
TIME: 0 to 1.999 [sec]
Duration: M203 V... TIME... is effective until a new voltage is output through M200 to M204.

Voltage for a specific time (time pulse): M204 V... TIME...
The TNC outputs the voltage programmed behind V... as a pulse. The duration of the pulse is entered with TIME....

Entry:	Voltage V:	0 to 9.999 [V]
	TIME:	0 to 1.999 [sec]
Duration:	M204 V... TIME... is effective until a new voltage is output through M200 to M204.	

4.22.2 Graphic Simulation Without TOOL CALL

Graphic simulation is also available on machines that operate without tool definition (e.g. water jet and laser cutters). You define the tool radius for graphic simulation in machine parameter MP7315. With MP7316 you define the depth of penetration of the simulated tool. You mark the program sections to be displayed with M functions that you define in machine parameters 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]

MP7317.0 M function at start of graphic display
Input: 0 to 88

MP7317.1 M function to interrupt graphic display
Input: 0 to 88

4.22.3 Program Stop with M Functions and TOOL CALL S

When an M function is output in the "Program run, full sequence" and "Program run, single block" modes, the program run is normally interrupted until you report with M4092 that the M function was executed. The same applies for a TOOL CALL with which only a spindle speed is programmed (TOOL CALL S).

This can be a disadvantage in certain applications, such as with laser cutting machines. For such applications it is more desirable not to wait for the acknowledgment, but rather to run the program continuously. You can therefore select with machine parameter MP7440 bit 2 and MP3030 whether program run should be interrupted in these cases. There must be no PLC positioning, datum shift, spindle orientation or changes in limit-switch range during M function output.

This function must not be used on milling machines or boring machines.

MP3030 Axis stops with TOOL CALL S...
Input: 0 = Axis stops with TOOL CALL S...
1 = Axis does not stop with TOOL CALL S...

MP7440 Output of M functions
Input: %xxxxx
Bit 2 Program stops for M functions
0 = Program stops until M function is acknowledged
1 = Program run continues (does not wait for acknowledgment)

4.23 Integrated Oscilloscope

The TNC features an integrated oscilloscope. To activate it, enter the code number **688 379**. It enables you to record and store the following axis characteristics in up to four channels:

`ACTL.SPEED` Actual value of the axis feed rate (mm/min).

`NOML SPEED` Nominal value of the axis feed rate (mm/min). The axis feed rate as calculated from the differences of the nominal position values. The servo lag is not included.

`FEED RATE` Machining feed rate (mm/min)

`ACTUAL POS` Actual position (mm)

`NOML.POS` Nominal position (mm)

`LAG` Servo lag of the position controller (µm)

`ENCODER: I1` Signal 1 of the position encoder

`ENCODER: I2` Signal 2 of the position encoder

`SAVED` The signal recorded last gets stored

`PLC` The PLC operands (B, W, D, I, O, T, C) are recorded. Enter the desired operands in the input field next to `PLC`.

Analog axes: **`VOLT.ANLOG`** Analog voltage = nominal velocity value (mV)

Digital axes: **`V (ACT RPM)`** Rotational speed actual value (mm/min). Calculated via tachometer and standardized with MP2020.

`V(NOM RPM)` Nominal velocity value (mm/min). Output quantity of the position controller.

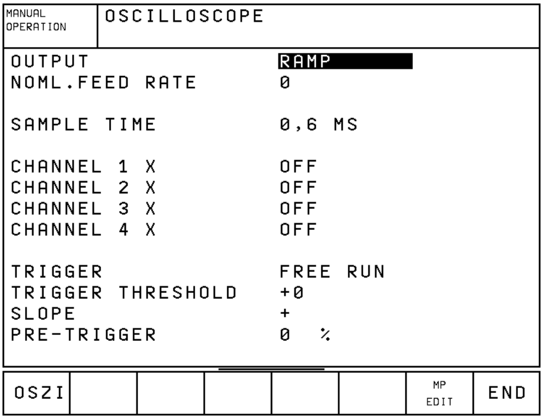
`I(INT RPM)` Integral component of the nominal current value (A)

`I NOMINAL` Nominal current value that determines torque (A)

The oscilloscope features additional functions for commissioning the current controller. See the "Commissioning" section.

The recorded data remains stored until you start recording again or activate another graphic function. You can configure the colors for the oscilloscope in MP7356.x.

After you have entered the code number the **Setup Menu** appears.



```
MANUAL
OPERATION   OSCILLOSCOPE

OUTPUT                  RAMP
NOML.FEED RATE          0

SAMPLE TIME             0,6 MS

CHANNEL 1 X             OFF
CHANNEL 2 X             OFF
CHANNEL 3 X             OFF
CHANNEL 4 X             OFF

TRIGGER                 FREE RUN
TRIGGER THRESHOLD       +0
SLOPE                   +
PRE-TRIGGER             0  %
```

Use the cursor keys to select the desired position and set the parameters.

OUTPUT

Here you can define whether the nominal velocity value is output as a step or as a ramp. For output as a ramp, the programmed feed rate and the k_V factors and acceleration rates set in the machine parameters go into effect.

If you have selected step output, a step will be output as nominal velocity value when you press the axis direction buttons in the "Manual" operating mode. The position control loop is open while the step is being output. You define the height of the step in the **NOML. FEED RATE** input field.

NOML. FEED RATE

Here you enter the height of the step for the nominal velocity value (in mm/min). This entry has no significance for the output of a ramp.

SAMPLE TIME

You can set the time interval for recording the signals to between 0.6 and 6 ms. 4096 samples are stored. The signals are therefore stored for a time duration of 2.4576 s to 24.576 seconds.

CHANNEL 1 to **CHANNEL 4**

Here you assign to each of the four channels the signal and axis that you wish to record. The input possibilities are described above.

TRIGGER

Here you define the type of recording. You have the following possibilities:

- FREE RUN — The recording is started and stopped manually via soft keys. When you press the STOP key, the last 4096 events are recorded.
- SINGLE SHOT — When you press the START key, the next 4096 events are recorded.
- CHANNEL 1 to 4 — Recording begins when the triggering threshold of the selected channel is exceeded.

TRIGGER THRESHOLD

Here you enter the trigger threshold in the following dimensions:

- Velocity [mm/min]
- Position [mm]
- Rotational speed [mm/min]
- Servo lag [µm]
- Analog voltage [mV]
- Current [A]

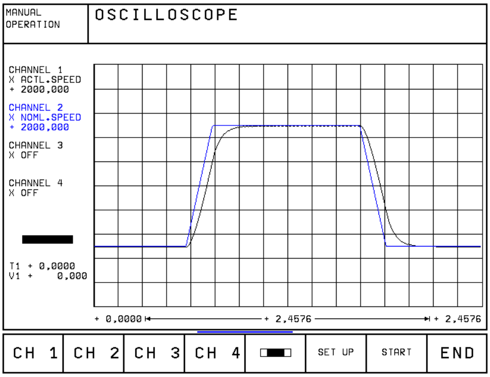
SLOPE

Here you set whether the rising (positive) or falling (negative) edge should be triggered.

PRE-TRIGGER

The stored recording starts before the trigger time point by the value defined here.

Press the OSCI soft key to call the **oscilloscope display.**



MANUAL OPERATION

OSCILLOSCOPE

CHANNEL 1
X ACTL.SPEED
+ 2000,000

CHANNEL 2
X NOML.SPEED
+ 2000,000

CHANNEL 3
X OFF

CHANNEL 4
X OFF

T1 + 0,0000
V1 + 0,000

The selected signals are continually displayed during recording. The memory contents are displayed after completion of the recording. For every channel the manner of the signal and the resolution are also shown. The length of the recorded range, relative to the entire memory content, is shown as a bar in the status field.

To move the cursor, press the arrow keys. The status field shows the amplitude of the selected channel and the time from the beginning of recording. With the CURSOR 1/2 soft key you activated a second cursor. The oscilloscope shows you the current amplitude and time for this cursor as well. The time of the second cursor refers to the position of the first cursor. The time of the second cursor refers to the position of the first cursor. You can use this function to measure the acceleration time of an axis, for example.

Meaning of the soft keys:

CH 1	You select one of the four channels and get a new soft-key row with the following soft keys:	
	INVERT	Inverts the signal
	Arrows	Moves the signal up or below
	↕⎍	Increasing vertical resolution
	↕⎽⎍⎽	Reducing vertical resolution
	⊓⊔	Optimal vertical resolution. The signal is centered in the screen. With NO ENT you return to the originally selected resolution.
	CURSOR 1/2	Changes over on second cursor
	END	Back to the oscilloscope display
▭■▭	Select the memory area to be displayed. A new soft-key row with the following soft keys appears:	
	Arrows	Move the signal to the left or on the right
	■■■ ← →	Reducing horizontal resolution
	▭■▭ → ←	Increasing horizontal resolution.
	END	Back to the oscilloscope display
SET UP	Back to the 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

You can store the last signal recorded with "Saved". You can also store the recorded signals with all the settings as a file on the hard disk using the SAVE SCREEN soft key. The file must have the extension DTA. This data can then be called up again with the PLC.EXE program.

4.24 Commissioning

4.24.1 Preparation

- Check the wiring against the grounding diagram and the safety concept (see "Basic Circuit Diagram")
- Check the "control-is-ready" function (see "EMERGENCY STOP Monitoring").
- Check the EMERGENCY STOP circuit by pressing the EMERGENCY STOP keys and the EMERGENCY STOP limit switch.
- The current machine parameter file must be selected. Most of the input values can be clearly determined from the present documentation. Enter preliminary values for those machine parameters that are determined during commissioning.
- The PLC program for integration to the machine must be written prior to commissioning. Ensure that in the system file OEM.SYS the instruction **`PLCMAIN=`** refers to the current PLC program. To write the PLC program, use the PLC programming software PLCdesign.

4.24.2 Digital Axis

With MP120 you define digital and analog axes. The current controller, speed controller and position controller must be interfaced in sequence. You can complete all interfacing procedures directly at the TNC. You can record the signals that you need from the TNC's integrated oscilloscope.

NC software: 280470..:
Digital and analog axes are defined in MP2000.x.

MP2000.0-5	*Type of drive*
Input:	*0 = output of nominal speed command signal (analog axis)*
	1 = output of current pulses (digital axis)
MP2001	*Type of drive for spindle*
Input:	*0 = output of nominal speed command signal (analog spindle)*
	1 = output of current pulses (digital spindle)

Motor and power stage

In the machine parameter editor you select the installed power stages and motors using special menus, which you call using special soft keys.

▷		Call list of power stages.
	SELECT AXIS	Select axis and confirm selected power stage with SELECT.
	EDIT	Change the data for the power stages.
	APPEND AMPLIFIER	Add power stage.
	END	Return to machine parameter editor.
-(M)-		Call list of motors.
	SELECT AXIS	Select axis and confirm selected motor with SELECT.
	EDIT	Change the data for the motors. The TNC shows either the list of synchronous or asynchronous motors, depending on the selected motor.
	APPEND SYNC	Add synchronous motor.
	APPEND ASYNC	Add asynchronous motor.
	END	Return to machine parameter editor.

After you have selected the motor and the power stage, the correct data is automatically entered in MP2100.x, MP2101, MP2200.x and MP2201.

If you are using motors or power stages that are not listed in the menus, please contact your HEIDENHAIN representative.

You can overwrite the standard data or add additional models to the tables. If you change the list of motor models or power stages, the altered tables are saved in the PLC partition.

PLC:\MP\MOTOR.ASY	list of asynchronous motors
PLC:\MP\MOTOR.SN	list of synchronous motors
PLC:\MP\MOTOR.AMP	list of power stages

The TNC then uses these tables. If at any time you wish to use the HEIDENHAIN standard table again, you must erase the tables in the PLC partition.

MP2100.0-5 Model of power stage for the axes
Input: Name of the selected power stage (is registered by the TNC)

MP2101 Model of power stage for the spindle
Input: Name of the selected power stage (is registered by the TNC)

MP2200.0-5 Model of motor for the axes
Input: Name of the selected motor (is registered by the TNC)

MP2201 Model of motor for the spindle
Input: Name of the selected motor (is registered by the TNC)

Maximum motor speed

		Max. speed
Axis drives	TNC 426 PB TNC 430 PA	$\frac{18\ 000}{\text{No. pole pairs}}\frac{1}{\text{min}}$
Spindle drives	TNC 426 PB standard	$\frac{18\ 000}{\text{No. pole pairs}}\frac{1}{\text{min}}$
Spindle drives	TNC 426 PB option TNC 430 PA	$\frac{30\ 000}{\text{No. pole pairs}}\frac{1}{\text{min}}$

The maximum speed indicated in motor data sheets always depends on a specific dc-link voltage supply. If you are working with a lower dc-link voltage, this speed will not be reached. You can combat this effect on synchronous motors by entering a field angle offset. This means however that more current is required starting from the speed at which the field angle begins to shift. The thermal limit curve is shifted as a result.

MP2340.0-5 Speed starting from which the field angle begins to shift on synchronous motors
Input: 0 to 100 000 [rpm]
0 = no field angle offset

MP2350.0-5 Maximum field angle offset for synchronous motors
Input: 0 to 60 [°]

Logic unit up to Id. Nr. xxx xxx 3x:
Due to the differing characteristics of HEIDENHAIN and SIEMENS current controllers, the maximum speed for synchronous motors attainable with the TNC lies 15% below the value given in the SIEMENS data sheets. Please keep this in mind when selecting the motors. By entering a field angle offset, however, you can attain the maximum speed specified in the SIEMENS data sheet. To do this, enter the following values:

- *MP2340 = rated speed / 1.2*
- *MP2350 = 30°*

Please note that, with these data, starting from the speed at which the field angle begins to shift, the motors draw 16% more current than the SIEMENS data specify. This shifts the thermal limit curve.

DC link voltage

In MP2190 you must enter the dc-link voltage applied to the power stage.

MP2190 DC link voltage
Input: 0 to 10 000 [V]

Temporary input values:

To start, enter the following temporary input values:

MP1030.x = 0.01	Positioning windows
MP1090.x= 1000	Acceleration rate-of-change limit
MP1092 = <greater than rapid traverse>	Feed-rate threshold from which MP1090.1 goes into effect
MP1095 = 0	Single filter
MP1096 = 0	Nominal position value filter off
MP1099.0 = 5	Minimum filtering order for single filter
MP1099.1 = 3	Minimum filtering order for double filter
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 the reference marks
MP1410.x = 0.5	Position monitoring in operation with velocity feedforward (erasable)
MP1420.x = 2	Position monitoring in velocity feedforward mode (EMERGENCY STOP)
MP1510.x =1	kv factor for velocity feedforward
MP1710.x = 50	Position monitoring in lag mode (erasable)
MP1720.x = 50	Position monitoring in lag mode (EMERGENCY STOP)
MP1810.x = 1	kv factor for lag mode
MP1820.x = 1	Multiplication factor for the kv factor
MP1830.x = 100	Characteristic kink
MP2000.x = 1	Digital axes
MP2020.x = ?	Traverse for one motor revolution (machine-specific)
MP2400.x = 0.1	Amplification for current controller
MP2500.x = 0.5	Proportional factor of the speed controller
MP2510.x = 0	Integral factor of the speed controller (for axes with holding moment, e.g. vertical axes, the value 1 must be entered, since otherwise the axis will drift.)
MP2512.x = 0	Limitation of the integral factor of the speed controller
MP2520.x = 0	Differential factor of the speed controller
MP2530.x = 0	2nd order time-delay (PT_2) element of the speed controller
MP2540.x = 0	Band-stop filter for damping
MP2550.x = 0	Band-stop filter for mid-frequency
MP2600.x = 0	Acceleration feed forward
MP2610.x = 0	Friction compensation at low speed
MP2612.x = 0	Delay of friction compensation
MP2620.x = 0	Friction compensation at nominal speed
MP2630.x = 0	Holding current
MP2800.x = 0	Motion monitoring for position and speed

Current controller

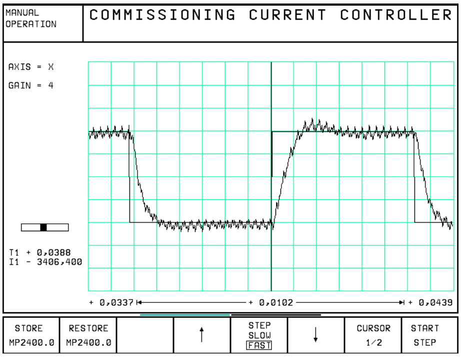
You can adjust the current controller with the integrated oscilloscope (code number 688 379). Since the speed and position controllers are open while the current controller is being adjusted, you must activate a special PLC commissioning program. Enter the name of this PLC program in the file OEM.SYS with the instruction **`PLCPWM=.`** It suffices to program only one EM (end module) in this PLC program. Please note, however, that the drive must be enabled externally and the TNC must receive the readiness signal through the interface card.

As soon as the PLC program defined in **`PLCPWM=`** is activated, you can interrogate the status of commissioning with Module 9168.

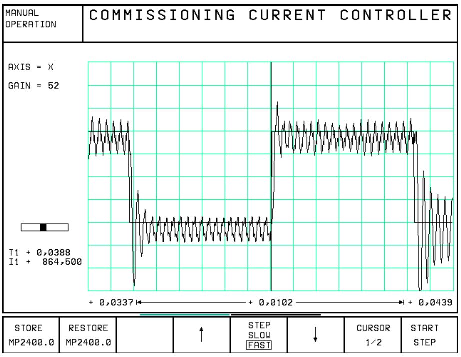
Call:
CM 9168
PL D <Status>
–1: Commissioning not active or no axis yet selected.
Bit 0 to Bit 5: Axis 1 to axis 6 selected
Bit 15: Spindle selected
Bit 16: Circuitry of the spindle
0: Wye connection
1: Delta connection

Procedure:

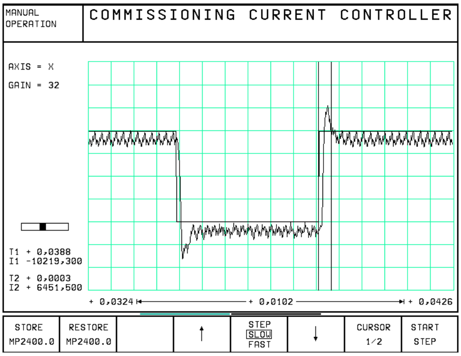
- ➤ Switch on the control.
- ➤ Do not acknowledge the message **`POWER INTERRUPTED`** and, in the editing mode (via MOD) enter the code number 688 379. This starts the integrated oscilloscope.
- ➤ Press the soft key I CONTROL.
- ➤ In the Manual operating mode, acknowledge the message `POWER INTERRUPTED.` This begins translation of the PLC program that is defined with the command `PLCPWM=` in the OEM.SYS file.
- ➤ In the oscilloscope mode, use 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, whose step response is measured. The height and length of the step function is calculated automatically by the TNC using the entered machine 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.


MANUAL OPERATION
COMMISSIONING CURRENT CONTROLLER
AXIS = X
GAIN = 4
T1 + 0,0388
I1 - 3406,400
+ 0,0337 + 0,0102 + 0,0439

MP2400 Too small


MANUAL OPERATION
COMMISSIONING CURRENT CONTROLLER
AXIS = X
GAIN = 52
T1 + 0,0388
I1 + 864,500
+ 0,0337 + 0,0102 + 0,0439

MP2400 too large



MANUAL OPERATION

COMMISSIONING CURRENT CONTROLLER

AXIS = X

GAIN = 32

T1 + 0,0388
I1 -10219,300

T2 + 0,0003
I2 + 6451,500

MP2400 at optimum setting

➤ When the current gain is properly adjusted, press the STORE MP2400 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

Before commissioning the speed controller you must first deselect the "Cross over reference mark" function (MP1340.x = 0). The loaded PLC program must contain the following functions:

- Opening the position control loop (W1038/W1040), since the NC opens the loop only during step function output. Since the position controller is not yet optimized, a closed control loop provokes undesired error messages.
- Releasing the drive controller (Module 9161)
- NC stop inactive (M4560 = 1)
- Axis direction buttons
- Clamping the axes

Procedure:

➤ MP2500.x = 0.5 Proportional factor of the speed controller
MP2510.x = 0 Integral factor of the speed controller (for axes requiring a holding force, such as vertical axes, you must enter the value 1 to prevent the axis from drifting)
MP2520.x = 0 Differential factor of the speed controller
MP2530.x = 0 2nd order time-delay (PT_2) element of the speed controller

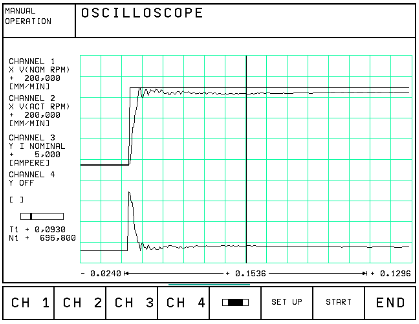
➤ In the manual operating mode, use the oscilloscope function to send a step function to the speed controller (approx. 500 mm/min). Display the nominal velocity value V (NOM RPM), the actual speed value V (ACT RPM) and the nominal current value (I NOMINAL).

➤ Choose a step height that will not overload the speed controller.

➤ Activate the step function by pressing the axis direction buttons.

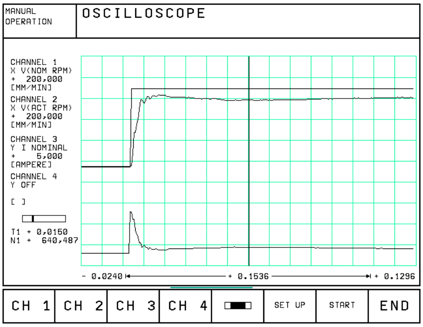
➤ To change the machine parameters, press the MP EDIT soft key in the setup menu.

➤ Increase MP2500.x (P factor) up to the oscillation limit.



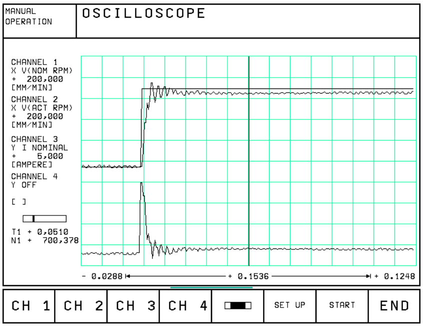
```
CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    5,000
[AMPERE]
CHANNEL 4
Y OFF
[ ]
T1 + 0,0930
N1 +  695,800
```

MP2500.x at the oscillation limit



```
CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    5,000
[AMPERE]
CHANNEL 4
Y OFF
[ ]
T1 + 0,0150
N1 +  640,487
```

MP2500.x too small

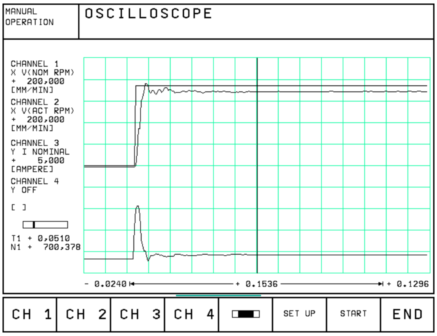


```
CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    5,000
[AMPERE]
CHANNEL 4
Y OFF
[ ]
T1 + 0,0510
N1 +  700,378
```

MP2500.x too large

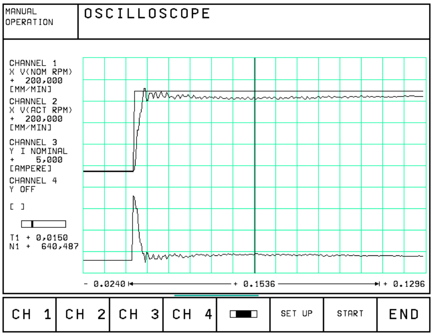
➤ Input value for MP2500.x = <determined value> • 0.6

➤ *You can compensate high frequency disturbance oscillations (> 400 Hz) with MP2530.x. However, this compensation dampens the control loop. Try first to correct the mechanical causes of the disturbance oscillations.*



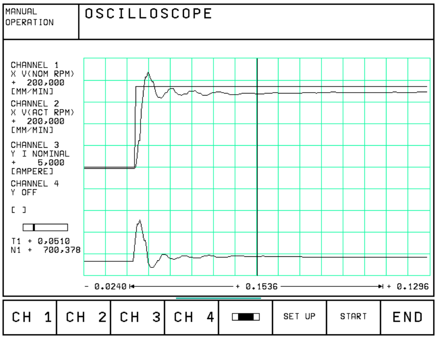
```
MANUAL
OPERATION        OSCILLOSCOPE

CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    5,000
[AMPERE]
CHANNEL 4
Y OFF
[ ]

T1 + 0,0510
N1 +  700,378
```

MP2530.x at optimum setting



```
MANUAL
OPERATION        OSCILLOSCOPE

CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    5,000
[AMPERE]
CHANNEL 4
Y OFF
[ ]

T1 + 0,0150
N1 +  640,487
```

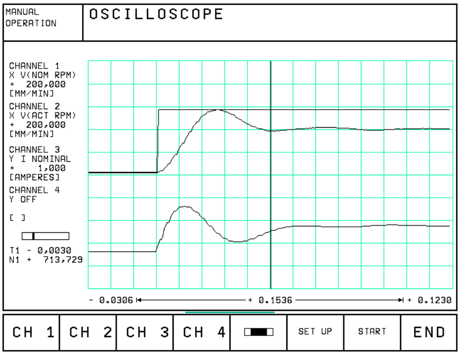
MP2530 too small



```
MANUAL
OPERATION        OSCILLOSCOPE

CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    5,000
[AMPERE]
CHANNEL 4
Y OFF
[ ]

T1 + 0,0510
N1 +  700,378
```

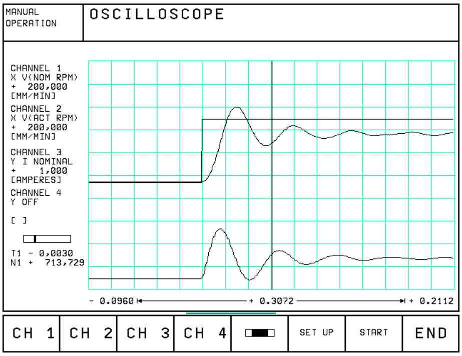
MP2530.x too large

➤ *HEIDENHAIN recommends avoiding the use of MP2520.x if at all possible. On mechanically problematic axes you can compensate low-frequency disturbance oscillations (< 200 Hz) with MP2520.x. Never use MP2520.x on belt-driven axes.*



```
MANUAL
OPERATION        OSCILLOSCOPE

CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    1,000
[AMPERES]
CHANNEL 4
Y OFF
[ ]

T1 - 0,0030
N1 +  713,729
```

MP2520.x at optimum setting



```
MANUAL
OPERATION        OSCILLOSCOPE

CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    1,000
[AMPERES]
CHANNEL 4
Y OFF
[ ]

T1 - 0,0030
N1 +  713,729
```

MP2520.x too small

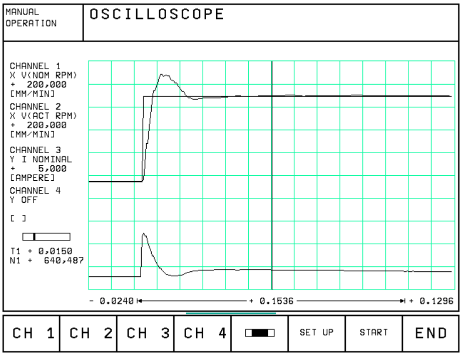
- ➤ *You can also compensate disturbance oscillations with the band-stop filter. To find the band-stop filter, calculate the frequency of the oscillation and enter it in MP2550.x.*

- ➤ *Increase the damping for the band-stop filter (MP2540.x) until the oscillation is minimized. Realistic inputs values lie between 3 to 9 dB.*

  This compensation dampens the control loop. For this reason it is best to attempt first to correct the mechanical causes of the oscillation.

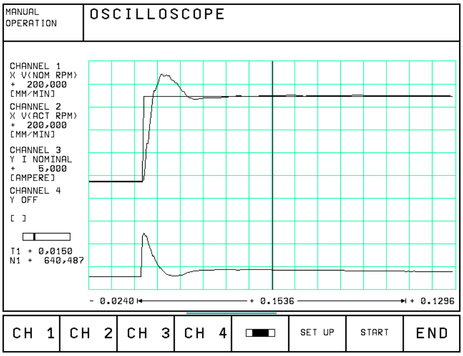
In order to reduce the occurrence of disturbance oscillations, HEIDENHAIN recommends the use of motor couplings with little tendency to oscillate (e.g. from Rotex).

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



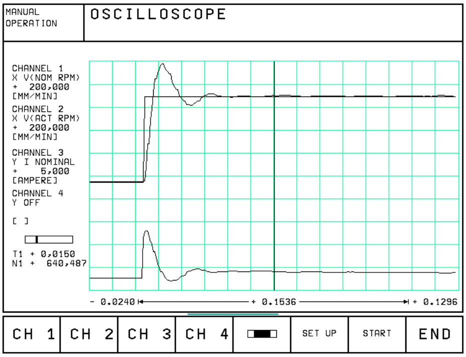
```
MANUAL
OPERATION        OSCILLOSCOPE

CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    5,000
[AMPERE]
CHANNEL 4
Y OFF

[ ]

T1 + 0,0150
N1 +  640,487

- 0.0240 |←———— + 0.1536 ————→| + 0.1296
```

MP2510.x at optimum setting



```
MANUAL
OPERATION        OSCILLOSCOPE

CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    5,000
[AMPERE]
CHANNEL 4
Y OFF

[ ]

T1 + 0,0150
N1 +  640,487

- 0.0240 |←———— + 0.1536 ————→| + 0.1296
```

MP2510.x too small



```
MANUAL
OPERATION        OSCILLOSCOPE

CHANNEL 1
X V(NOM RPM)
+  200,000
[MM/MIN]
CHANNEL 2
X V(ACT RPM)
+  200,000
[MM/MIN]
CHANNEL 3
Y I NOMINAL
+    5,000
[AMPERE]
CHANNEL 4
Y OFF

[ ]

T1 + 0,0150
N1 +  640,487

- 0.0240 |←———— + 0.1536 ————→| + 0.1296
```

MP2510.x too large

Determining the acceleration

➤ Clamp an object of maximum permissible weight on the machine table

➤ Enter the rapid traverse rate as step height

➤ During the step response, record the nominal speed 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 during acceleration.

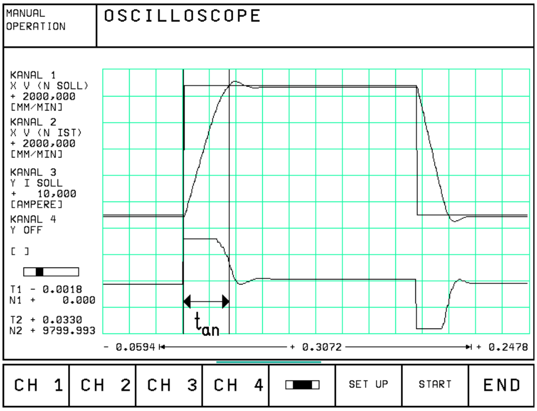
➤ From the step response of the speed controller you determine the maximum possible acceleration:

$$a = \frac{F_{max}}{t_r \bullet 66\,000}$$

$a =$ acceleration (MP1060.x) $\left(\frac{m}{s^2}\right)$

$F_{max} =$ maximal machining feed rate (MP1010.x) $\left(\frac{mm}{min}\right)$

$t_r =$ rise time [s]



```
MANUAL
OPERATION        OSCILLOSCOPE

KANAL 1
X V (N SOLL)
+ 2000,000
[MM/MIN]
KANAL 2
X V (N IST)
+ 2000,000
[MM/MIN]
KANAL 3
Y I SOLL
+   10,000
[AMPERE]
KANAL 4
Y OFF

[ ]

T1 - 0.0018
N1 +    0.000

T2 + 0.0330
N2 + 9799.993

- 0.0594          + 0.3072          + 0.2478
```

Counting direction

➤ On the oscilloscope, set the **TRIGGER** to **FREE RUN.**

➤ Start recording

➤ Change to Manual operating mode.

➤ Press the axis direction buttons.

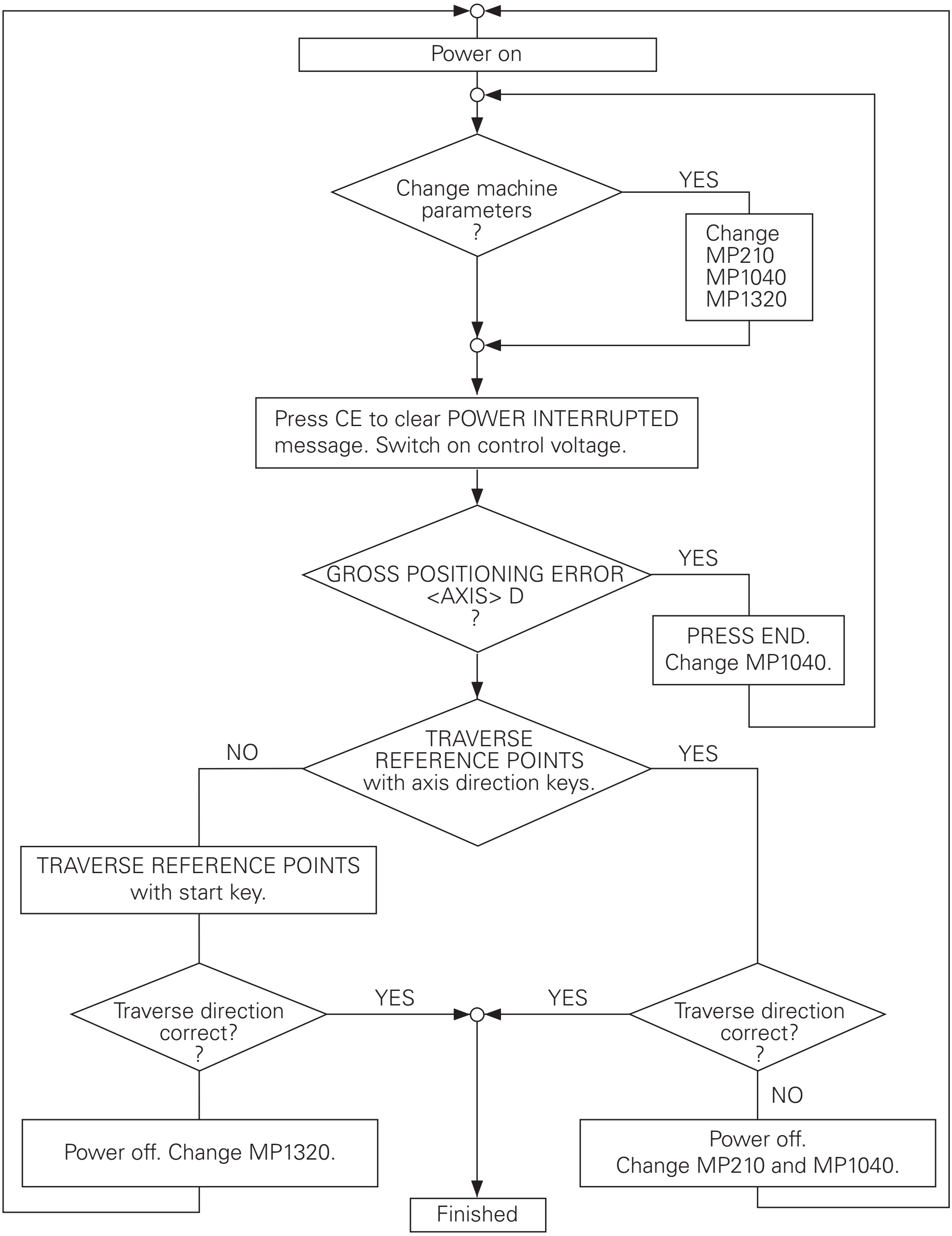
➤ Check the counting direction on the display and, if necessary, correct it with MP210.x.

Position controller

You must activate a PLC program that is interfaced to the machine. The position control loop must be closed (W1038/W1040) and all PLC inputs and outputs must be operated correctly. To optimize the position control loop, proceed as follows:

Checking the traversing direction

In MP1340.x enter the sequence in which you want the reference points to be traversed. Check the traversing direction with the following flowchart:


Power on
Change machine parameters ?
YES
Change MP210 MP1040 MP1320
Press CE to clear POWER INTERRUPTED message. Switch on control voltage.
GROSS POSITIONING ERROR <AXIS> D ?
YES
PRESS END. Change MP1040.
TRAVERSE REFERENCE POINTS with axis direction keys.
NO
YES
TRAVERSE REFERENCE POINTS with start key.
Traverse direction correct? ?
YES
YES
Traverse direction correct? ?
NO
Power off. Change MP1320.
Power off. Change MP210 and MP1040.
Finished

Setting the traverse range

You can define up to three traverse ranges (see "Traverse Ranges" at the beginning of this chapter). To define the software limit switches, proceed as follows:

➤ In the manual mode of operation, press the MOD key, then select REF display.

➤ Position displays show the distance to the machine datum (MP960.x).

➤ With the axis direction buttons or handwheel, move all axes in positive and negative direction until they almost reach the EMERGENCY STOP limit switch. Write down the displayed positions.

➤ Enter the noted values in MP91x.x or MP92x.x.

➤ Press the MOD key and select the ACTL display.

Control with velocity feedforward (MP1390 = 0)

k_v factor / acceleration rate-of-change limit:

➤ MP1390 = 0
MP1090.0 = 1000
MP1090.1 = 1000
MP1092 = <twice rapid traverse>
MP1095 = 0
MP1096 = 0
MP1099.0 = 5
MP1099.1 = 3

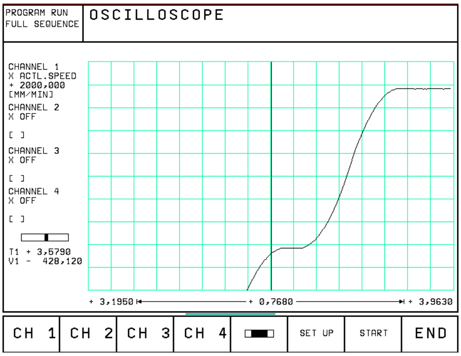
➤ Enter the following test program:

```
LBL 1
L X<maximum traverse> R0 FMAX
L X0 FMAX
CALL LBL1 REP 100/100
```

➤ With the internal oscilloscope, show the actual feed rate (ACTL.SPEED)

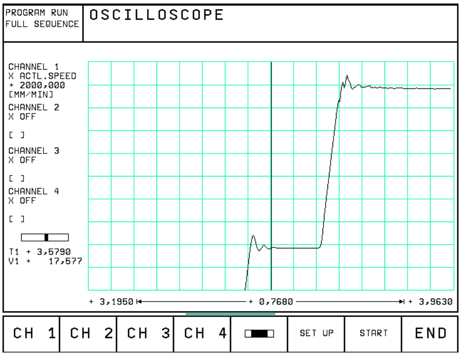
➤ Start the test program. Feed rate override = 100 %

➤ With MP1090.0, reduce the acceleration rate of change until the overshoot disappears.



```
PROGRAM RUN
FULL SEQUENCE      OSCILLOSCOPE

CHANNEL 1
X ACTL.SPEED
+ 2000,000
[MM/MIN]
CHANNEL 2
X OFF
[ ]
CHANNEL 3
X OFF
[ ]
CHANNEL 4
X OFF
[ ]

T1 + 3,5790
V1 -  428,120
```

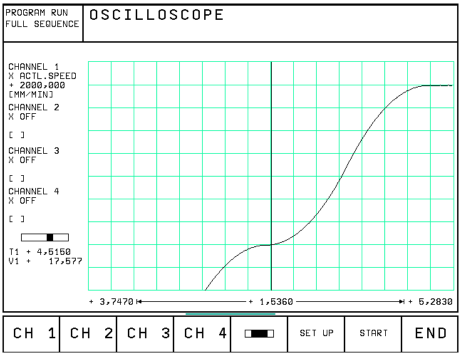
MP1090.0 at optimum setting



PROGRAM RUN FULL SEQUENCE — OSCILLOSCOPE

CHANNEL 1 X ACTL.SPEED + 2000,000 [MM/MIN]
CHANNEL 2 X OFF [ ]
CHANNEL 3 X OFF [ ]
CHANNEL 4 X OFF [ ]
T1 + 3,5790
V1 + 17,577

MP1090.0 too large



PROGRAM RUN FULL SEQUENCE — OSCILLOSCOPE

CHANNEL 1 X ACTL.SPEED + 2000,000 [MM/MIN]
CHANNEL 2 X OFF [ ]
CHANNEL 3 X OFF [ ]
CHANNEL 4 X OFF [ ]
T1 + 4,5150
V1 + 17,577

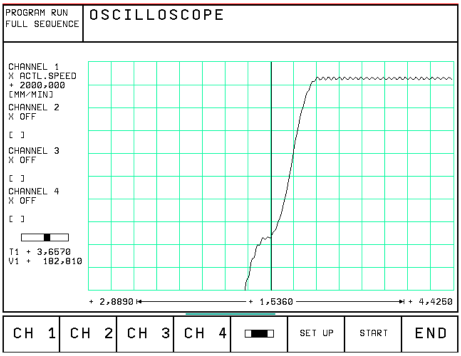
MP1090.0 too small

As a further aid, the servo lag can also be recorded on the oscilloscope. Or you determine the permissible jerk simply by placing your hand on the machine.

➤ Transfer the jerk from MP1090.0 to the axis-specific machine parameter MP1097.x.

➤ Enter double the value from MP1097.x in MP1098.x.

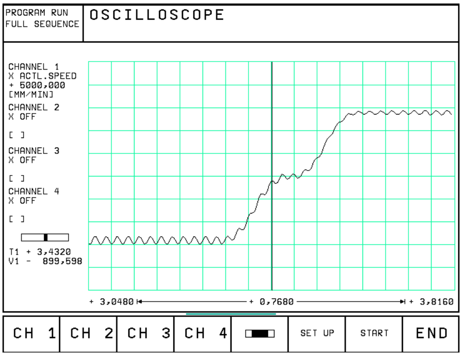
➤ Increase the k_v factor (MP1510.x) until the oscillation limit has been reached.



PROGRAM RUN FULL SEQUENCE — OSCILLOSCOPE

CHANNEL 1 X ACTL.SPEED + 2000,000 [MM/MIN]
CHANNEL 2 X OFF [ ]
CHANNEL 3 X OFF [ ]
CHANNEL 4 X OFF [ ]
T1 + 3,6570
V1 + 182,810

Oscillation limit has been reached



PROGRAM RUN FULL SEQUENCE — OSCILLOSCOPE

CHANNEL 1 X ACTL.SPEED + 5000,000 [MM/MIN]
CHANNEL 2 X OFF [ ]
CHANNEL 3 X OFF [ ]
CHANNEL 4 X OFF [ ]
T1 + 3,4320
V1 - 899,598

k_v factor too large

➤ MP1510.x = <calculated value> • 0.6

➤ Unlike operation with servo lag, velocity feedforward enables you to adjust the optimum k_v factor for each axis even if they are interpolated.

➤ You can store in the TNC differing k_v factors, which you can activate with M functions (see "Control Loop").

➤ Adjust all other axes in the same way.

➤ MP1090.x is effective for all axes. The worst axis determines the input value.

➤ The adjusted acceleration rate of change (MP1090.0) may have to be further reduced, depending on the mechanical design of the machine. Do not set the acceleration rate of change lower than necessary, however, because this very strongly reduces the acceleration performance at a small acceleration rate of change.

➤ If after you have optimized the acceleration rate-of-change the axis does not reach the maximum acceleration, enter the maximum feed rate in MP1092.

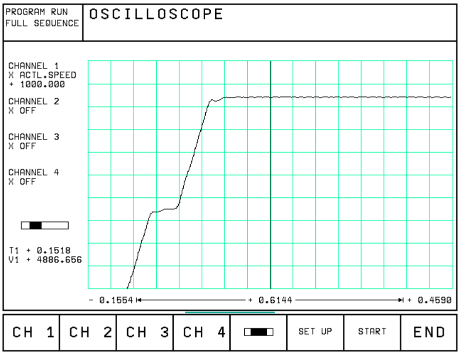
In MP1090.1 you then define a higher acceleration rate of change for high feed rates (greater than MP1092) in order to achieve a higher acceleration at these feed rates.

➤ Switch on nominal position value filter. Enter a tolerance value (e.g. 0.02 mm) in MP1096.

Control with servo lag (MP1390 = 1)

The max. defined jerk is also effective in control with servo lag. The values for MP1090.x, MP1097.x, MP1098.x, MP1095.x and MP1099.x remain the same. The nominal position value filters are not activated until the k_V factors are determined.

- MP1390 = 1
- MP1810 ≈ 1 (provisional input value)
- MP1096 = 0
- Enter the following test program:
  LBL 1
  L X<maximum traverse> R0 F<maximum machining feed rate>
  L X0 R0 F<maximum machining feed rate>
  CALL LBL1 REP 100/100
- With the internal oscilloscope, display the actual feed rate (ACTL.SPEED)
- Start the test program. Feed rate override = 100 %
- Increase MP1810.x until you can clearly recognize an oscillation tendency.



PROGRAM RUN
FULL SEQUENCE

OSCILLOSCOPE

CHANNEL 1
X ACTL.SPEED
+ 1000.000

CHANNEL 2
X OFF

CHANNEL 3
X OFF

CHANNEL 4
X OFF

T1 + 0.1518
V1 + 4886.656

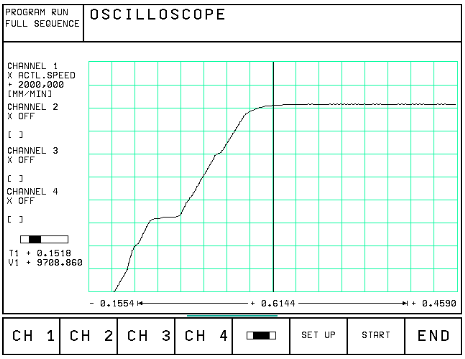
MP1810.x at the oscillation limit

- MP1810.x = <determined value> • 0.6
- The k_V factors for axes that that are to be interpolated together must be equal. The axis with the smallest k_V factor determines the input value for all axes.
- You can store differing k_V factors in the TNC memory and activate with M functions (see "Control Loop").

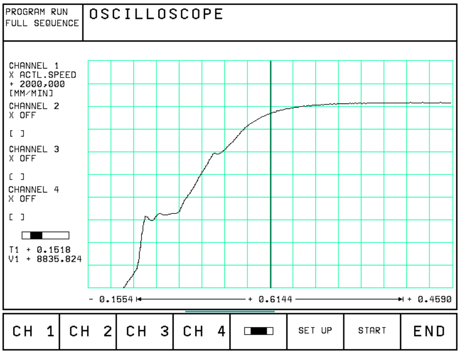
k_V factor for rapid traverse (characteristic kink):

- $\text{MP1830.x} = \frac{\text{max. feed rate} \bullet 100\ \%}{\text{rapid traverse}}$
  MP1820.x = 1
  MP1390 = 1
- Enter the following test program:
  LBL 2
  L X<maximum traverse> R0 FMAX
  L X0 R0 FMAX
  CALL LBL1 REP 100/100
- Start the test program.
- With the integral oscilloscope, show the actual feed rate (ACTL.SPEED).

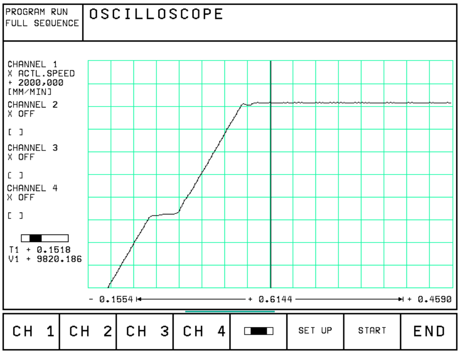
- ➤ If no oscillations are visible, a characteristic kink is not necessary. If oscillations are visible, you must decrease MP1820.x until the oscillations have disappeared.



MP1820.x at optimum setting



MP1820.x too small



MP1820.x too large

- ➤ Adjust all other axes in the same way

Switch on nominal position value filter

- ➤ Enter a defined tolerance in MP1096 (e.g. 0.02 mm)

Monitoring functions:

To ensure that the monitoring functions become active at the proper moment, you must first have entered meaningful values. HEIDENHAIN recommends the following input values, which you must then adjust slightly to fit the design of the individual machine.

MP1030.x = 0.01 mm	Positioning window
MP1110.x = 2 • MP1030.x	Standstill monitoring
MP1140.x = 0.03 [1000 Rpm]	Motion monitoring
MP2800.x = 0.5 mm	Motion monitor for position and speed
MP1410.x = 0.5 mm	Position monitoring in operation with velocity feedforward (erasable)
MP1420.x = 2 mm	Position monitoring in operation with velocity feedforward (EMERGENCY STOP)
MP1710.x = 1.2 • Servo lag in rapid traverse	Position monitoring in the operation with servo lag (erasable)
MP1720.x = 1.4 • Servo lag in rapid traverse	Position monitoring in the operation with servo lag (EMERGENCY STOP)

Compensation of backlash

Case 1: The cause is outside the controlled loop.

➤ Enter the backlash in MP710.0-8.

Case 2: The cause of the backlash is within the controlled loop.

➤ Enter the following test program:

```
LBL 1
L X100 R0 F10
L X0
CALL LBL 1 REP 100/100
```

➤ Record the V ACTL und V (ACT.RPM) with an internal oscilloscope

➤ At the turnaround point the actual feed rate trails the actual speed by time delay t.

➤ Input values: MP750 = t • ΔV ACTL
MP752 = approx. 20 ms (determined in test)

Compensation of static friction

➤ If the axis has backlash, it must be entered before you can find the static friction.

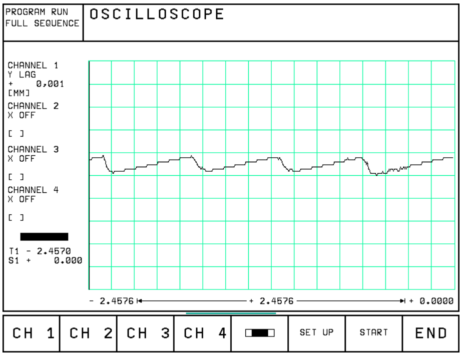
➤ Enter the following test program (static friction in axis Y):

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

➤ MP1511.x = 0
MP1512.x = 20
MP1513.x = 0

➤ With the internal oscilloscope, display the servo lag of the axis Y (Y SDIFF).

➤ Start the program and adjust the override so that the servo lag resulting from static friction is visible.

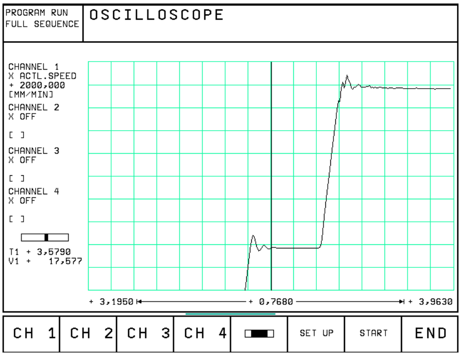


```
PROGRAM RUN
FULL SEQUENCE   OSCILLOSCOPE

CHANNEL 1
Y LAG
+     0,001
[MM]
CHANNEL 2
X OFF

[ ]
CHANNEL 3
X OFF

[ ]
CHANNEL 4
X OFF

[ ]

T1 - 2.4570
S1 +    0.000

- 2.4576 |<------------ + 2.4576 ------------>| + 0.0000
```

➤ Increase the feed rate till the servo lag is no longer measurable. From the current machining feed rate, calculate the feed rate specific to the axis Y and enter the value in MP1513.1.

➤ Decrease the feed rate until the servo lag is measurable again

➤ Increase MP1511.x (in steps of 10 000) until the servo lag is no longer measurable.



```
PROGRAM RUN
FULL SEQUENCE   OSCILLOSCOPE

CHANNEL 1
X ACTL.SPEED
+ 2000,000
[MM/MIN]
CHANNEL 2
X OFF

[ ]
CHANNEL 3
X OFF

[ ]
CHANNEL 4
X OFF

[ ]

T1 + 3,5790
V1 +   17,577

+ 3,1950 |<------------ + 0,7680 ------------>| + 3,9630
```

➤ If the machine oscillates at a standstill, decrease MP1512.x.

Limitation of the integral factor of the speed controller

➤ Very high stick-slip friction can cause an axis to jerk loose and "jump" around the target position. In this case, increase MP2512.x until the axis remains stationary.

Holding moment

See section "Control Loop" in this chapter.

➤ Enter the following test program (holding moment for axis Z):

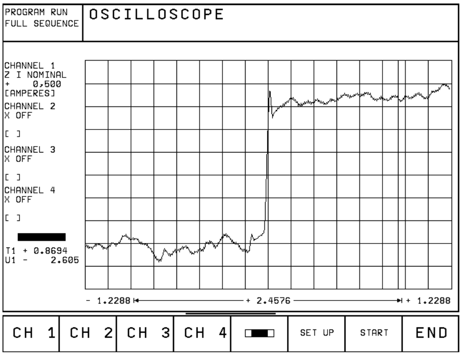
```
LBL 1
L Z+2 R0 F50
L Z–2 R0 F50
CALL LBL 1/10
```

➤ With the integrated oscilloscope, record the actual speed value V(ACT.RPM) and the nominal current value (I NOMINAL).

➤ Start the program.

➤ With the feed-rate override knob, adjust the motor speed to ±10 rpm (MP2020.x) in each of both traverse directions.

- ➤ Find the nominal current (I NOMINAL) in both directions.



PROGRAM RUN
FULL SEQUENCE

OSCILLOSCOPE

CHANNEL 1
Z I NOMINAL
+ 0,500
[AMPERES]
CHANNEL 2
X OFF
[ ]
CHANNEL 3
X OFF
[ ]
CHANNEL 4
X OFF
[ ]

T1 + 0.8694
U1 - 2.605

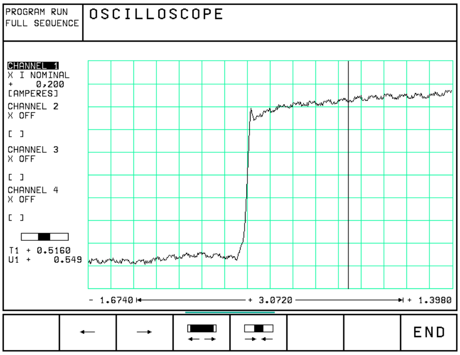
- ➤ MP2630.x = $\dfrac{\text{I NOML}_1 + \text{I NOML}_2}{2}$

Compensation of sliding friction

- ➤ MP1390 = 0 Operation with velocity feedforward.
- ➤ Enter the following test program (sliding friction in axis X):

```
LBL 1
L X+2 R0 F5
L X–2 R0 F5
CALL LBL 1/10
```

- ➤ With the integrated oscilloscope, record the actual speed value V(ACT.RPM) and the nominal current value (I NOMINAL).
- ➤ Start the program.
- ➤ With the feed-rate override knob, adjust the motor speed to ±10 rpm (MP2020.x).
- ➤ Find the nominal current (I NOMINAL) in both directions of rotation.



PROGRAM RUN
FULL SEQUENCE

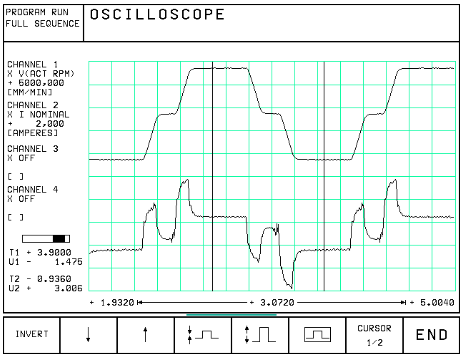
OSCILLOSCOPE

CHANNEL 1
X I NOMINAL
+ 0,200
[AMPERES]
CHANNEL 2
X OFF
[ ]
CHANNEL 3
X OFF
[ ]
CHANNEL 4
X OFF
[ ]

T1 + 0.5160
U1 + 0.549

- ➤ MP2610.x = $\dfrac{|\text{I NOML}_1| - |\text{I NOML}_2|}{2}$
- ➤ Change the test program so that the motor rotates at its rated speed.
- ➤ Start the program.

- Find the nominal current (I NOMINAL) in both directions of rotation.



```
PROGRAM RUN     OSCILLOSCOPE
FULL SEQUENCE

CHANNEL 1
X V(ACT RPM)
+ 5000,000
[MM/MIN]
CHANNEL 2
X I NOMINAL
+    2,000
[AMPERES]
CHANNEL 3
X OFF

[ ]
CHANNEL 4
X OFF

[ ]

T1 + 3.9000
U1 -    1.475

T2 - 0.9360
U2 +    3.006

+ 1.9320 |<-------- + 3.0720 -------->| + 5.0040
```

- $$\text{MP2620.x} = \frac{|\text{I NOML}_1| + |\text{I NOML}_2|}{2}$$
- In the event that the motor cannot be driven at the rated speed, measure I NOMINAL at the maximum speed (rapid traverse) and calculate the current at the rated speed as follows:

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

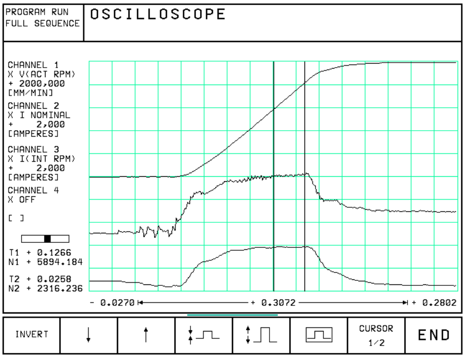
I_{nmax} = current at rapid traverse
n_{max} = motor speed at rapid traverse

Acceleration feedforward

- MP1390 = 0   Operation with velocity feedforward.
- Enter the following test program:

```
LBL 1
L X+100 R0 F5000
L X–100 R0 F5000
CALL LBL 1/10
```

- With the integral oscilloscope, record the actual speed value V (ACT RPM), the nominal current value (I NOMINAL) and the integral-action component of the nominal current value I (INT RPM).
- Start the program.
- With the feed-rate override knob, adjust the speed so that I NOMINAL is not limited.



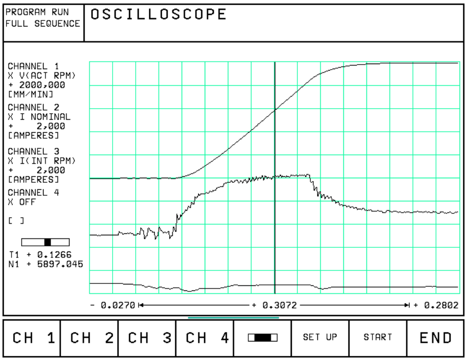
```
PROGRAM RUN     OSCILLOSCOPE
FULL SEQUENCE

CHANNEL 1
X V(ACT RPM)
+ 2000,000
[MM/MIN]
CHANNEL 2
X I NOMINAL
+    2,000
[AMPERES]
CHANNEL 3
X I(INT RPM)
+    2,000
[AMPERES]
CHANNEL 4
X OFF

[ ]

T1 + 0.1266
N1 + 5894.184

T2 + 0.0258
N2 + 2316.236

- 0.0270 |<-------- + 0.3072 -------->| + 0.2802
```

- Measure the gradient of the acceleration ramp in the part in which I (INT RPM) remains constant, and calculate MP2600.x as follows:

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

I (INT RPM) = integral-action component of the nominal current value
t = acceleration time (ramp)
Δ V (ACT RPM) = actual motor speed during t
MP2020.x = traverse per motor shaft revolution

➤ Repeat this measurement to check the input value from MP2600.x.
I (INT RPM) must have approached zero.



```
PROGRAM RUN
FULL SEQUENCE        OSCILLOSCOPE

CHANNEL 1
X V(ACT RPM)
+ 2000,000
[MM/MIN]
CHANNEL 2
X I NOMINAL
+    2,000
[AMPERES]
CHANNEL 3
X I(INT RPM)
+    2,000
[AMPERES]
CHANNEL 4
X OFF

[ ]

T1 + 0.1266
N1 + 5897.045
```

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, where MP1060.x = smallest acceleration in the machining plane

➤ At a mid-range feed rate (approx. 500 mm/min), check whether MP2610 is at the optimum setting. At the optimum setting the reversal peaks are at a minimum.

➤ At high feed rates (from approx. 6000 rpm), the reversal peaks might point inward as a result of overcompensation. Increase MP2612.x until the reversal peaks no longer point inward.

4.24.3 Analog Axes

Temporary input values:

To start, enter the following temporary values:

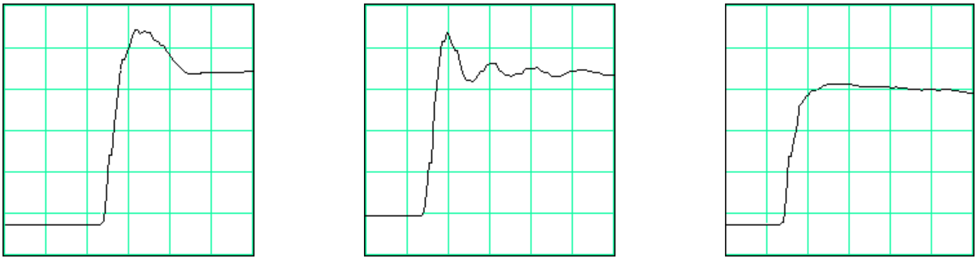
MP1030.x = 0.01	Positioning window
MP1090.x = 1000	Acceleration rate-of-change limit
MP1092 = <greater than rapid traverse>	Feed rate threshold from which MP1090.1 becomes effective
MP1110.x = 2.0	Standstill monitoring
MP1140.x = 10	Motion monitoring
MP1410.x = 0.5	Position monitoring in operation with velocity feedforward (erasable)
MP1420.x = 2	Position monitoring in operation with velocity feedforward (EMERGENCY STOP)
MP1510.x =1	k_V factor for velocity feedforward
MP1710.x = 50	Position monitoring in operation with servo lag (erasable)
MP1720.x = 50	Position monitoring in operation with servo lag (EMERGENCY STOP)
MP1810.x = 1	k_V factor operation with servo lag
MP1820.x = 1	Multiplication factor for the k_V factor
MP1830.x = 100	Characteristic kink

Interfacing the servo amplifier:

Before you optimize the position controller on the TNC, you must first adjust the servo amplifier. Procedure:

- Disconnect the nominal-value connection between the servo amplifier and logic unit.
- Short-circuit the nominal value input. The input must have a 0 V voltage.
- Activate the control enabling at the servo amplifier.
- Connect the power supply to the servo amplifier.
- 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 becomes stationary.
- Remove the jumper at the nominal value input and establish a nominal-value connection to the logic unit.
- Coarse speed adjustment:
  MP1010.x (rapid traverse) and MP1050.x (analog voltage for rapid traverse) must be correctly adjusted. With the internal oscilloscope function, transmit 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.
- Measure the step response on the tachometer during the step output.

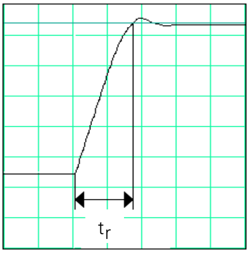
➤ Adjust the proportional component and integral-action component of the tachometer at the servo amplifier:


Optimum setting Gain too large Gain too small

Determining the acceleration

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

$$a = \frac{F_{max}}{t_r \bullet 66\,000}$$



a = acceleration (MP1060.x) $\left(\frac{m}{s^2}\right)$

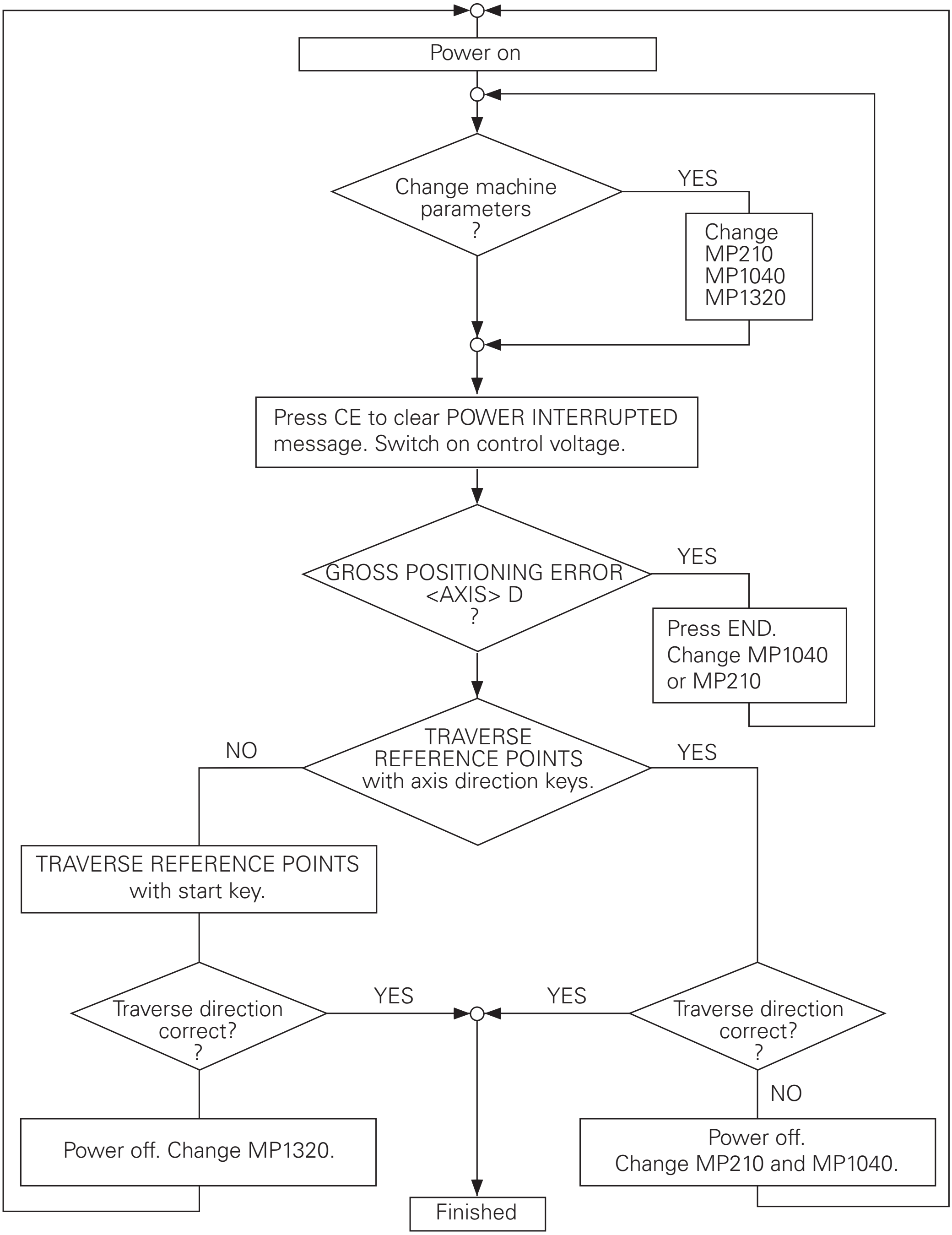
F_{max} = maximal machining feed rate (MP1010.x) $\left(\frac{mm}{min}\right)$

t_r = rise time [s]

Optimizing the position controller at the TNC

You must activate a PLC program that is interfaced to the machine. The position control loop must be closed (W1038/W1040) and all PLC inputs and outputs must be operated correctly. To optimize the position control loop, proceed as follows:

Checking the traversing direction


Power on
Change machine parameters ?
YES
Change MP210 MP1040 MP1320
Press CE to clear POWER INTERRUPTED message. Switch on control voltage.
GROSS POSITIONING ERROR <AXIS> D ?
YES
Press END. Change MP1040 or MP210
NO
TRAVERSE REFERENCE POINTS with axis direction keys.
YES
TRAVERSE REFERENCE POINTS with start key.
Traverse direction correct? ?
YES
YES
Traverse direction correct? ?
NO
Power off. Change MP1320.
Finished
Power off. Change MP210 and MP1040.

Setting the traverse range

Same procedure as for digital axes.

Control with servo lag (MP1390 = 1)

Same procedure as for digital axes.

Control with velocity feedforward (MP1390 = 0)

Same procedure as for digital axes.

Offset adjustment

You can carry out the fine adjustment of the offset at the TNC. See the "Position Control Loop" chapter above.

Monitoring functions:

To ensure that the monitoring functions become active at the proper moment, you must first have entered meaningful values. HEIDENHAIN recommends the following input values. You must then adjust these input values slightly to fit the design of the individual machine.

MP1030.x = 0.01 mm	Positioning window
MP1110.x = 2 • MP1030.x	Standstill monitoring
MP1140.x = 0.5 V	Motion monitoring
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 • servo lag in rapid traverse	Position monitoring in the operation with servo lag (erasable)
MP1720.x = 1.4 • servo lag in rapid traverse	Position monitoring in the operation with servo lag (EMERGENCY STOP)

Compensation of static friction

Same procedure as with digital axes.

4.24.4 Digital Spindle for TNC 426 without Spindle DSP

Temporary input values:

MP3010 = 3 to 8	Output of spindle speed, gear range
MP3020 = 991	Spindle speed range
MP3411.x = 1.999	Ramp gradient
MP3412.x = 1	Multiplier for MP3411.x
MP3415.x = 0	Transient response
MP3420 = 1	Positioning window
MP3440.x = 1	k_v factor

Current controller

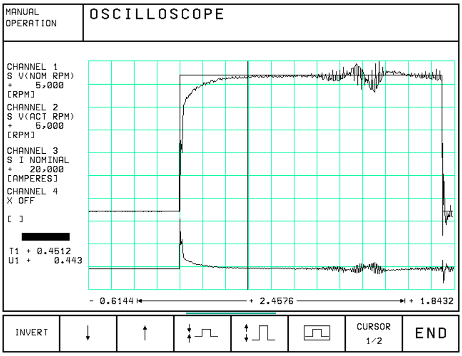
As with digital axes, except that MP2401 is adjusted instead of MP2400.x.

Speed controller

You do not use the internal oscilloscope to define the step function as with the axes. Rather, you simply enter the maximum acceleration (MP3411.x) and start the step by switching the spindle on. Procedure:

➤ 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 speed value V(NOM RPM), actual speed value V(ACT RPM) the nominal current value (I NOMINAL).
- Output a step by activating the spindle on function (M03/M04).
- Select the height of the step function for a very low speed so as not to overload the speed controller (I NOMINAL).
- To change the machine parameters, in the setup menu press the soft key MP EDIT.
- Increase MP2501 (P factor) until the system oscillates or until no change is visible:

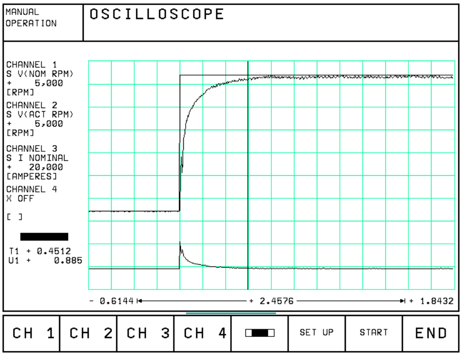


MANUAL OPERATION

OSCILLOSCOPE

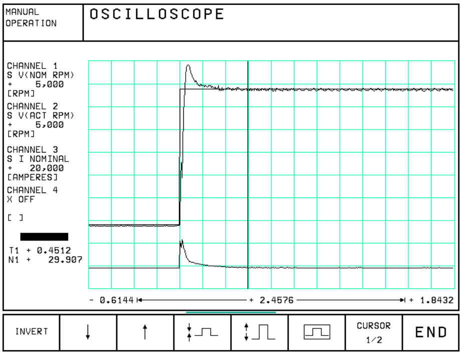
```
CHANNEL 1
S V(NOM RPM)
+    5,000
[RPM]
CHANNEL 2
S V(ACT RPM)
+    5,000
[RPM]
CHANNEL 3
S I NOMINAL
+   20,000
[AMPERES]
CHANNEL 4
X OFF

[ ]

T1 + 0.4512
U1 +    0.443
```

➤ MP2501 = MP2501 • 0.6



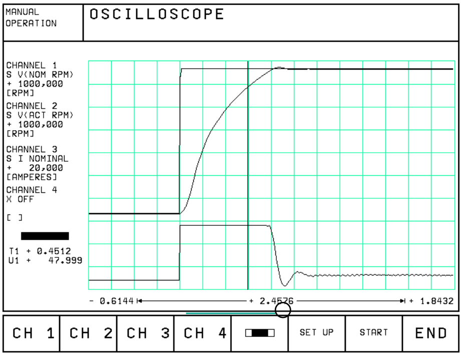
```
MANUAL
OPERATION     OSCILLOSCOPE

CHANNEL 1
S V(NOM RPM)
+    5,000
[RPM]
CHANNEL 2
S V(ACT RPM)
+    5,000
[RPM]
CHANNEL 3
S I NOMINAL
+   20,000
[AMPERES]
CHANNEL 4
X OFF
[ ]

T1 + 0.4512
U1 +    0.885

- 0.6144        + 2.4576        + 1.8432
```

➤ Increase MP2511 (I factor) until one overshoot is followed by a slight undershoot.



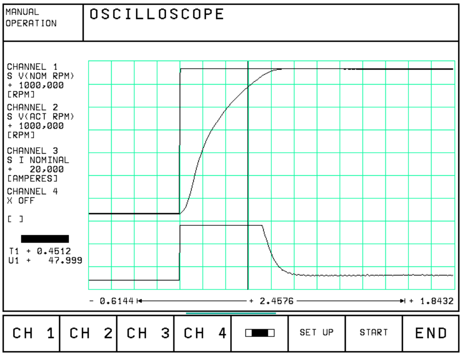
```
MANUAL
OPERATION     OSCILLOSCOPE

CHANNEL 1
S V(NOM RPM)
+    5,000
[RPM]
CHANNEL 2
S V(ACT RPM)
+    5,000
[RPM]
CHANNEL 3
S I NOMINAL
+   20,000
[AMPERES]
CHANNEL 4
X OFF
[ ]

T1 + 0.4512
N1 +   29.907

- 0.6144        + 2.4576        + 1.8432
```

➤ Output a step with maximum speed. I NOMINAL is within the limit during acceleration. After the maximum speed has been reached, I NOMINAL should not oscillate. If it does, you must decrease MP2501 and MP2511 by the same amounts until the overshoot is minimized.



```
MANUAL
OPERATION     OSCILLOSCOPE

CHANNEL 1
S V(NOM RPM)
+ 1000,000
[RPM]
CHANNEL 2
S V(ACT RPM)
+ 1000,000
[RPM]
CHANNEL 3
S I NOMINAL
+   20,000
[AMPERES]
CHANNEL 4
X OFF
[ ]

T1 + 0.4512
U1 +   47.999

- 0.6144        + 2.4576        + 1.8432
```

I NOMINAL oscillating

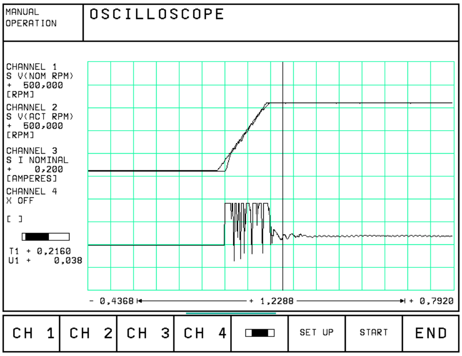


```
MANUAL
OPERATION     OSCILLOSCOPE

CHANNEL 1
S V(NOM RPM)
+ 1000,000
[RPM]
CHANNEL 2
S V(ACT RPM)
+ 1000,000
[RPM]
CHANNEL 3
S I NOMINAL
+   20,000
[AMPERES]
CHANNEL 4
X OFF
[ ]

T1 + 0.4512
U1 +   47.999

- 0.6144        + 2.4576        + 1.8432
```

Only one overshoot

Acceleration

- ➤ The acceleration must be optimized individually for each gear range. For operation with M03, M04 and M05, select the ramp gradient so that the motor almost reaches the limit of current. Enter the setting in MP3410.x.



MANUAL OPERATION

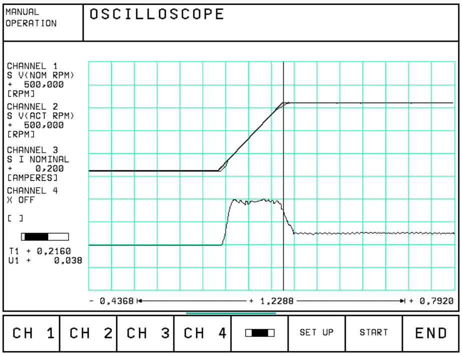
OSCILLOSCOPE

CHANNEL 1
S V(NOM RPM)
+ 500,000
[RPM]
CHANNEL 2
S V(ACT RPM)
+ 500,000
[RPM]
CHANNEL 3
S I NOMINAL
+ 0,200
[AMPERES]
CHANNEL 4
X OFF
[ ]

T1 + 0,2160
U1 + 0,038

- 0,4368 + 1,2288 + 0,7920

- ➤ In MP3412.0 you enter a factor for MP3411.x that takes effect in the braking ramp with M05. Braking also occurs in the current limit.
- ➤ In the tapping and spindle orientation operating modes, I NOMINAL must not be in the limit during acceleration. With MP3412.1 to MP 3412.3 you enter a factor for MP3411.x for these operating modes.



MANUAL OPERATION

OSCILLOSCOPE

CHANNEL 1
S V(NOM RPM)
+ 500,000
[RPM]
CHANNEL 2
S V(ACT RPM)
+ 500,000
[RPM]
CHANNEL 3
S I NOMINAL
+ 0,200
[AMPERES]
CHANNEL 4
X OFF
[ ]

T1 + 0,2160
U1 + 0,038

- 0,4368 + 1,2288 + 0,7920

- ➤ With MP3415.x you determine an individual transient response for each operating mode of the spindle. You adapt the nominal value curve to the actual value curve.

Direction of rotation

Check the rotational direction of the spindle when M03 is output. If the spindle does not rotate in clockwise direction, change MP3130.

Position controller

For 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.”
- ➤ If the error message **`Nominal speed value too high`** appears, you must change MP3140.
- ➤ Now optimize the k_v factor individually for each gear range (MP3440.x). A TOOL CALL must be run before the changed gear-specific MPs are transferred.

Higher current gain beginning with the nominal speed

Because counter EMF increases with increasing shaft speed, a greater current gain becomes necessary at high speeds.

- ➤ With the integrated oscilloscope, record V (ACT RPM) and activate a shaft speed greater than the nominal speed.
- ➤ If V (ACT RPM) oscillates only at high speeds, increase MP2403 until the spindle runs smoothly.
- ➤ If V (ACT RPM) also oscillates at low speeds, the cause of the problem is resonance, which you can compensate just as for the axes with the differential factor and the PT2 element.

4.24.5 Digital Spindle for TNC 430 / TNC 426 with Spindle DSP

Temporary input values:

MP3010 = 3 to 8	Output of the shaft speed, gear range
MP3020 = 991	Shaft speed range
MP3411.x = 1.999	Ramp gradient
MP3412.x = 1	Multiplier for MP3411.x
MP3415.x = 0	Transient response
MP3420 = 1	Positioning window
MP3440.x = 1	k_V factor

Wye (star) connection / Delta connection

During commissioning you can use soft keys to switch between a wye connection and delta connection. You can interrogate the current setting in the PLC with Module 9168 and then switch the motor through PLC outputs and activate the corresponding machine parameters with Module 9163. You must adjust both for the wye connection and the delta connection. If you do not use the delta connection, enter a zero in the corresponding machine parameter.

Current controller

➤ Switch on the control.

➤ Do not acknowledge the message **POWER INTERRUPTED** and, in the Programming and Editing mode, enter the code number 688 379. This starts the integrated oscilloscope.

➤ Press the I CONTROL soft key.

➤ In the Manual operating mode acknowledge the message **POWER INTERRUPTED**. This translates the PLC program defined in the OEM.SYS file with **PLCPWM=**.

➤ In the oscilloscope mode of operation, use the SELECT AXIS soft key to select the spindle.

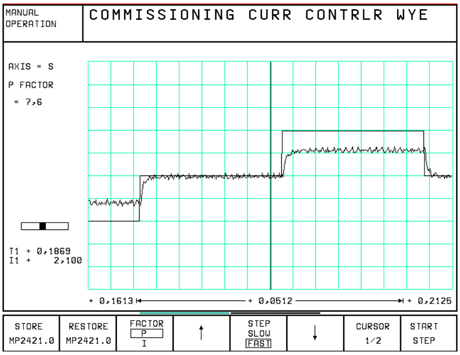
➤ With the STAR / DELTA soft key select either the wye (star) or delta connection.

➤ With the I FACTOR / P FACTOR soft key, select the I factor and set MP2431.x to zero.

➤ With the I FACTOR / P FACTOR soft key, select the P factor.

➤ Press START STEP. This sends a step function to the current controller and measures the step response. The height and length of the step function is calculated automatically by the TNC using the entered machine parameters.

➤ With the ↑ soft key, increase the P factor (MP2421.x) to the oscillation limit.



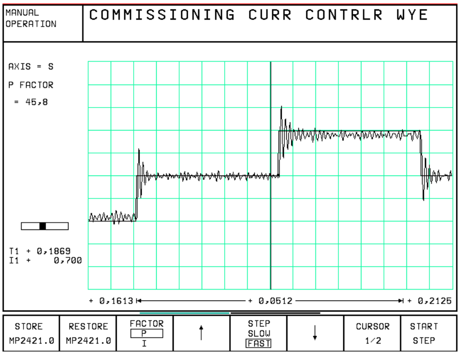
COMMISSIONING CURR CONTRLR WYE

AXIS = S
P FACTOR
= 7,6

T1 + 0,1869
I1 + 2,100

+ 0,1613 + 0,0512 + 0,2125

MP2421.x too small



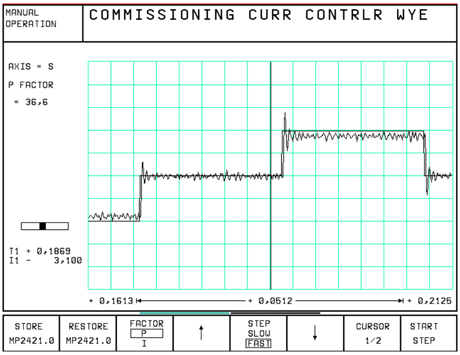
COMMISSIONING CURR CONTRLR WYE

AXIS = S
P FACTOR
= 45,8

T1 + 0,1869
I1 + 0,700

+ 0,1613 + 0,0512 + 0,2125

MP2421.x too large



MANUAL OPERATION

COMMISSIONING CURR CONTRLR WYE

AXIS = S

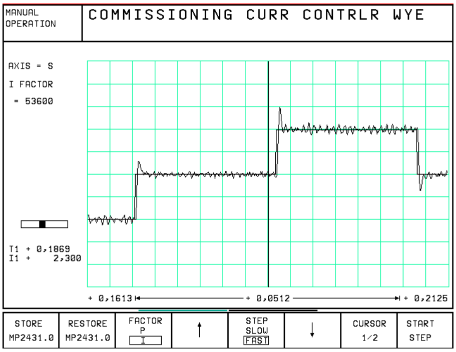
P FACTOR

= 36,6

T1 + 0,1869
I1 - 3,100

MP2421.x at the oscillation limit

- ➤ Input value for MP2421.x = <determined value> • 0.6
  Enter this value and save it with the STORE MP2421.x soft key.
- ➤ With the I-FAKTOR / P-FAKTOR soft key, select the I factor.
- ➤ With the ↑ soft key, increase the I factor (MP2431.x), until there is one overshoot but no undershoot.



MANUAL OPERATION

COMMISSIONING CURR CONTRLR WYE

AXIS = S

I FACTOR

= 53600

T1 + 0,1869
I1 + 2,300

MP2431.x at optimum setting

- ➤ Save this value with the STORE MP2431.x soft key.
- ➤ Switch off the machine to exit the I CONTROL mode.

Speed controllers

Same procedure as for a digital spindle with TNC 426.

Acceleration

Same procedure as for a digital spindle with TNC 426.

Direction of rotation

Same procedure as for a digital spindle with TNC 426.

Position controller

Same procedure as for a digital spindle with TNC 426.

4.24.6 Analog Spindle

Adjusting servo amplifiers:
Same procedure as for analog axes.

Acceleration
Procedure is same as for a digital spindle. You measure the signals directly at the servo amplifier with an external oscilloscope.

Direction of rotation:
Same procedure as for a digital spindle.

Position controller:
Same procedure as for a digital spindle.

5 PLC Programming

5.1 PLC Functions

The integrated PLC in the TNC contains its own Text Editor for creating the list of statements for the PLC program. Commands and comments are entered via the ASCII keyboard on the TNC. It's easier, however, to create your PLC programs with the PLC compiler software PLCdesign. For information on PLCdesign, contact HEIDENHAIN.

You can use the functions TRACE and TABLE, as well as a syntax check on entering the PLC commands and a logical test with the COMPILE function to make it easier to find faults in the PLC program.

For the compiled PLC program you have 256 KB (approx. 32 000 blocks) available in the sequential program. A new PLC run begins every 21 ms (PLC cycle time). This means that every 21 ms the inputs are read and outputs are set. A PLC run must not take more than 7 ms.

5.1.1 Select PLC Operation

PLC operation covers all functions for creating and testing the PLC programs, for creating the PLC error messages and the dialogue texts for OEM cycles, the Help files and compensation lists for non-linear axis error compensation.

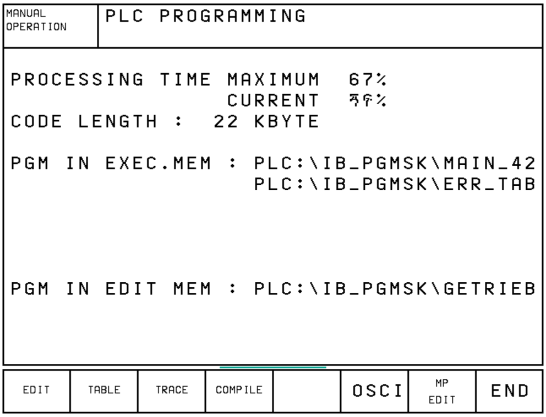
To select PLC operation:

➤ Select the NC "Programming and Editing" mode of operation.
➤ Press the MOD key
➤ Enter the code number **807 667**, or press the soft key PLC EDIT.
➤ PLC operation is now active (Main menu)
➤ You can exit PLC operation by pressing the END key or the END soft key.

[1] NC software 280 470 xx: 128 KB (approx. 16 000 blocks)

5.1.2 PLC Main Menu

After entering the code number (or soft key PLC EDIT) the following screen display will appear (main menu):



```
MANUAL
OPERATION    PLC PROGRAMMING

PROCESSING TIME MAXIMUM   67%
                CURRENT   [illegible]%
CODE LENGTH :  22 KBYTE

PGM IN EXEC.MEM : PLC:\IB_PGMSK\MAIN_42
                  PLC:\IB_PGMSK\ERR_TAB



PGM IN EDIT MEM : PLC:\IB_PGMSK\GETRIEB
```

The above information has the following meaning:

PROCESSING TIME MAXIMUM
The PLC processing time (time for a PLC run) is given as a percentage, whereby 3.5 ms is the equivalent of 100%

The maximum run time must not exceed 10.5 ms. If it is higher, the error message **PLC: Time out** will appear.

PROCESSING TIME CURRENT
The time for the latest PLC run, displayed in %.

CODE LENGTH
This is the length of the translated sequential program in KB. A maximum of 256 KB is possible.

PROGRAM IN EXECUTIVE MEMORY
The last compiled PLC program is displayed here (program in process memory). A program that was already selected as an executable program before switching on will be compiled automatically. The PLC program is active only after compilation!

PROGRAM IN EDITOR MEMORY
The name of a file which was selected with the soft key SELECT can be seen in the line PGM IN EDIT MEM.

From the main menu you can use soft keys to access the following PLC functions:

EDIT	Load PLC program into main memory for editing.
TABLE	Show logical states of the M/I/O/T/C or contents of the B/W/D (see section "Table Function" below).
TRACE	Show trace function or logic diagram.
COMPILE	Compile PLC program.
OSCI	Activate integrated oscilloscope
MP EDIT	Select machine parameters.
END	Exit PLC operation.

5.1.3 File Management

File management in PLC operation is nearly the same as in the normal programming mode. For an exact description, see the User's Manual for the control.

In PLC operation, pressing the PGM MGT key displays the directory of the PLC partition in addition to the TNC partition.

Unlike the normal programming mode, however, pressing the soft key SELECT TYPE calls the following soft-key row:

SELECT TYPE		
	SHOW ALL	List all files.
	.PLC FILES	Show only PLC files (*.PLC).
	SHOW .A	Show only ASCII files (*.A).
	.HLP FILES	Show only help files (*.HLP).
	.SYS FILES	Show only SYS files(*.SYS).
	.COM FILES	Show only COM files (*.COM).
	.CMA FILES	Show only CMA files (*.CMA).
	.PET FILES	Show only PET files (*.PET).
	END	Return to the higher-level menu.

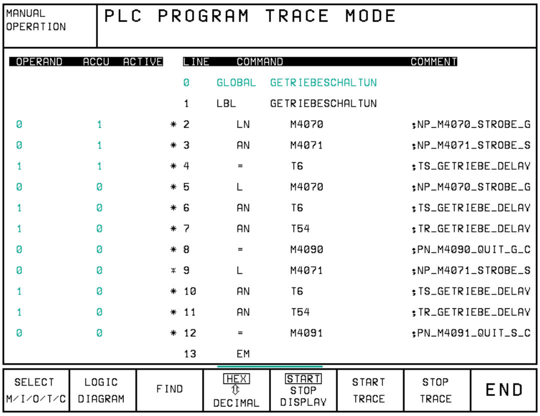
Pressing the MORE FUNCTIONS soft keys calls the following additional soft keys:

MORE FUNCTIONS		
	CONV. PLC CROSSREF	Generates a cross-reference list; the file name extension must not be one already used by the system (e.g. .PLC, .CMA, etc.).
	CONV. PLC LISTING	Generates a program listing numbered by line.
	CONV. PLC LISTING+ CROSSREF	Generates a cross-reference list and a program listing numbers by line.

5.1.4 TRACE Functions

The TRACE function makes it possible to check the logical states of the markers, inputs, outputs, timers and counters as well as to test the contents of byte, word and doubleword. These functions are available from the main menu by using the soft key TRACE.

The list of statements for the program will then be displayed. In addition, the contents of the operand and accumulator for each line of the program are shown in HEX or decimal code (selectable by soft key). Every cyclically executed command is identified with a "*". The cursor keys or the GOTO function can be used to display the required portion of the program.



```
MANUAL       PLC PROGRAM TRACE MODE
OPERATION

 OPERAND   ACCU  ACTIVE   LINE   COMMAND                      COMMENT
                           0    GLOBAL   GETRIEBESCHALTUN
                           1    LBL      GETRIEBESCHALTUN
  0          1         *  2        LN       M4070           ;NP_M4070_STROBE_G
  0          1         *  3        AN       M4071           ;NP_M4071_STROBE_S
  1          1         *  4        =        T6              ;TS_GETRIEBE_DELAY
  0          0         *  5        L        M4070           ;NP_M4070_STROBE_G
  1          0         *  6        AN       T6              ;TS_GETRIEBE_DELAY
  1          0         *  7        AN       T54             ;TR_GETRIEBE_DELAY
  0          0         *  8        =        M4090           ;PN_M4090_QUIT_G_C
  0          0         *  9        L        M4071           ;NP_M4071_STROBE_S
  1          0         *  10       AN       T6              ;TS_GETRIEBE_DELAY
  1          0         *  11       AN       T54             ;TR_GETRIEBE_DELAY
  0          0         *  12       =        M4091           ;PN_M4091_QUIT_S_C
                           13      EM

SELECT    LOGIC     FIND    HEX       START    START   STOP    END
M/I/O/T/C DIAGRAM           ⇕         STOP     TRACE   TRACE
                            DECIMAL   DISPLAY
```

Pressing the TRACE soft key calls a series of new soft keys with the following meanings:

TRACE	
SELECT M/I/O/T/C	Select M/I/O/T/C for 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	This soft key can start and stop the dynamic display of the operand, accumulator, and logic diagrams.
START TRACE	Start trace.
STOP TRACE	End trace.
END	Return to the higher-level menu.

5.1.5 Logic Diagram

The LOGIC DIAGRAM soft key calls a new row of soft keys with the following meanings:

LOGIC DIAGRAM		
	SELECT M/I/O/T/C	Select M/I/O/T/C for logic diagram.
	TRACE IN-CODE	Show trace in code.
	SAVE TRACE BUFFER	Assign (*.A) and save file names for diagram.
	RESTORE TRACE BUFFER	Show desired diagram.
	START STOP DISPLAY	This soft key can start and stop the dynamic display of the operand, accumulator, and logic diagrams.
	START TRACE	Start trace.
	STOP TRACE	End trace.
	END	Return to the higher-level menu.

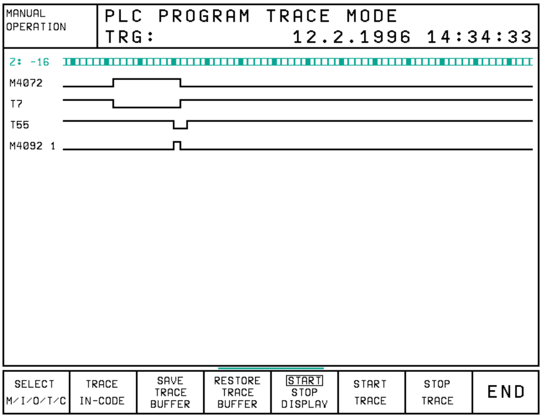
In this way it is possible to show the logical states of up to 16 operands (M, I, O, T, C) graphically and simultaneously on the VDU screen. 1024 PLC runs can be recorded. The selection is made by soft key SELECT M/I/O/T/C, which makes it possible to create a table with the required operands. The individual positions in the table are determined by dialog. Incorrect entries can be erased with the DEL key. For each operand a trigger condition can be entered. 512 states are recorded before and 512 after a trigger event. The following trigger conditions are possible:

1 → Record when operand is logical one (Triggering on positive edge)
0 → Record when operand is logical zero (Triggering on negative edge)

If no trigger condition is wanted, then confirm with NO ENT. If no trigger condition is entered for any of the operands, then the operand states will be continuously recorded and the last 1024 remain in memory.

A recording commences with START TRACE and is ended either with STOP TRACE or when the trigger event takes place. During recording of the logical states, the message PCTR will blink in the status window. The blinking will stop when the recording is finished. The cursor keys can be used to select the desired range in the TRACE buffer.

Example for logic diagram:



MANUAL OPERATION

PLC PROGRAM TRACE MODE

TRG: 12.2.1996 14:34:33

5.1.6 TABLE Function

The table of markers, inputs, outputs, timers, counters, bytes, words, and doublewords can be dynamically displayed on the VDU screen by using the soft key TABLE from within the main menu. The cursor keys or the GOTO key are used to select the positions within the table. Return to the main menu by pressing the END key.

The following soft keys are available for selecting, setting and resetting the markers, inputs, outputs, timers and counters:

TABLE		
	SET	Set M/I/O/T/C.
	RESET	Reset M/I/O/T/C.
	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	Return to the higher-level menu.

BYTE	Show list of bytes.
WORD	Show list of words.
DOUBLE	Show list of doublewords.
HEX ⇕ DECIMAL	Show contents of operands in hexadecimal or decimal.
SAVE M/B/W/D	Save operand ranges as file. The ranges of several operands can be saved, e.g. M0 to M100, W100 to W118.
RESTORE M/B/W/D	Show file of saved operands.
END	Return to the higher-level memory.

5.1.7 COMPILE Function

Compiling a completed PLC program transfers it to the process memory where it can then become active. The name of this program then appears in the main menu in the line PROGRAM IN EXEC MEM.

You can select the PLC program to be compiled by pressing the soft key COMPILE in the overview of existing PLC programs. You must confirm your selection with SELECT. During compilation, error messages may appear indicating programming errors. See the Appendix for a list of these error messages.

5.2 Operands

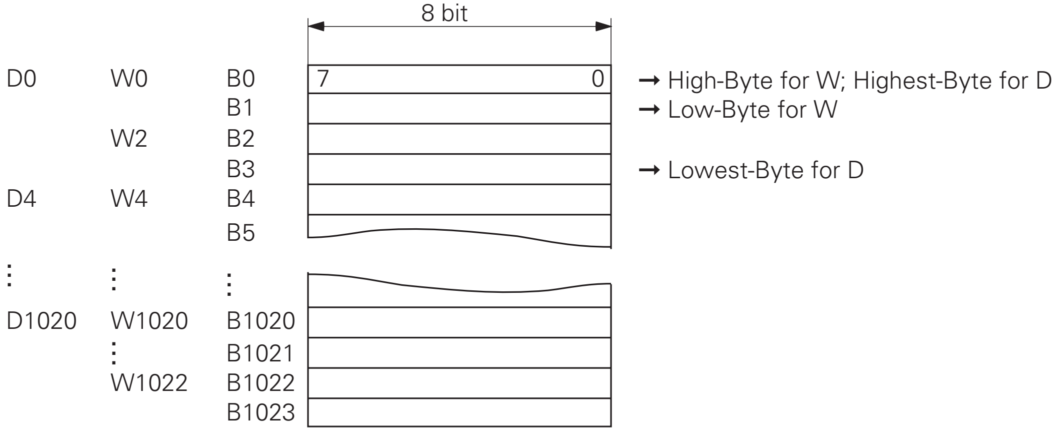
5.2.1 Operand Directory

Operand	Abbreviation	Address range
Marker	M (Marker)	M0 to M4999 M0 to M999* free, are not erased during reset M1000 to M3999 free, are erased during reset M4000 to M4999 reserved for NC/PLC interface
Input	I (Input)	I0 to I31; I128 to I152 I64 to I126 (first PL) I192 to I255 (second PL) I256 to I319 (third PL) I320 to I383 (fourth PL)
Output	O (Output)	O0 to O30; O32 to O62 (first PL) O64 to O94 (second PL) O128 to O158 (third PL) O160 to O190 (fourth PL)
Counter	C (Counter)	Set counter: C0 to C31 Counter contents: C48 to C79 Release count pulse: C96 to C127
Timer	T (Timer)	Timer start: T0 to T47 Timer running: T48 to T95
Byte Word Doubleword	B (Byte) W (Word) D (Doubleword)	B0 to B4095 (8 bit) B0 to B127* free, are not erased during reset B128 to B2047 reserved for NC/PLC interface B2048 to B4095 free, are erased during reset
Constant	K	- 2 147 483 647 to + 2 147 483 647
String	S	S0 to S3

* After entry of the code number **531 210** markers M0 to M999 and Byte B0 to B127 are erased.

5.2.2 Operand Addressing

The memory for the 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 overlap of the memory areas will occur, which must be taken into account in addressing the memory.



In byte addressing every address from 0 to 4095 is accessible. In word addressing, every second address from 0 to 4094 is accessible and in doubleword addressing every fourth from 0 to 4092. The address parameter gives the high byte for a word address (W) , or the highest byte for a doubleword address (D).

The markers, timer and counters are addressed by the corresponding letters M, T or C followed by the operand number (e.g. M500, T7, C18).

5.2.3 Data Transfer

Information is exchanged between PLC and NC by markers, bytes, words and doublewords. The function of the individual markers, bytes, words and doublewords is fixed.

The transfer of several numerical values is controlled by strobes.

5.2.4 Data Transfer NC → PLC

Transfer with FN19
The parameter function FN19 can be used in an NC program to transfer two numbers to the PLC. The transferred values are deposited in the doublewords D280 and D284.
During the transfer, the NC sets marker M4075. The PLC must acknowledge the transfer by setting marker M4095. Marker M4570 determines whether dimensions are in mm or inches. The transferred value is stored as an integer number in units of 1/10 000.

		Set	Reset
M4075	Transfer with FN19 active	NC	NC
M4095	Acknowledgment of transfer with FN19	PLC	PLC
M4570	Unit of measure through transfer with FN19 0 = mm; 1 = inch	NC	NC
D280	1st value from FN 19	NC	NC
D284	2nd value from FN 19	NC	NC

Transfer with FN17: Write system data

After entering the code number **555 343** you can use the soft key FN17 SYS-DATUM WRITE to overwrite system data that you find in the table below. After a control system reset the soft key FN17 SYS-DATUM WRITE is erased again, however the function remains active. FN17 was introduced primarily for certain tasks in OEM cycles. The system datum is selected by group number and index. The syntax is as follows:
FN17: SYSWRITE IDxxxx NRxxxx IDXxxxx = Qxxx or numerical value; comment

Group name	**Group number ID......**	**Sys. data number NR......**	**Sys. data index IDX.....**	**Input value for system datum**
Data in tool table	50	1	-	Tool length
		2	-	Tool radius
		3	-	Tool radius R2
		4	-	Oversize in tool length DL
		5	-	Oversize in tool radius DR
		6	-	Oversize in tool radius DR2
		7	-	Tool locked 0 = not locked, 1 = locked
		8	-	Number of replacement tool
		9	-	Maximum tool life TIME1
		10	-	Maximum tool life TIME2
		11	-	Current tool life CUR. TIME
		12	-	PLC status
		13	-	Maximum tooth length LCUTS
		14	-	Maximum plunge angle ANGLE
		15	-	TT: Number of teeth CUT
		16	-	TT: Tolerance for wear detection in tool length LTOL
		17	-	TT: Tolerance for wear detection in tool radius RTOL
		18	-	TT: Direction of rotation DIRECT 0 = pos; –1 = neg
		19	-	TT: Offset in the plane R-OFFS R = 99999.9999
		20	-	TT: Offset in tool length L-OFFS
		21	-	TT: Breakage tolerance in tool length LBREAK
		22	-	TT: Breakage tolerance in tool radius RBREAK

Group name	Group number ID......	Sys. data number NR......	Sys. data index IDX.....	Input value for system datum
Transformation	210	1	-	Basic rotation (manual)
		3	–	Active mirrored axis Bits 0 to 2 and 6 to 8: Axis X, Y, Z and U, V, W
		6	–	Tilt working plane 0 = not active / –1 = active
Exchange tool axes	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 switch axis 1 to 9
		3	1 to 9	Positive software limit switch axis 1 to 9
Triggering touch probe TS	350	10	-	Tool axis
		11	-	Effective radius
		12	-	Effective length
		13	-	Radius of calibration ring
		14	1	Center offset (main axis)
			2	Center offset (secondary axis)
		15	-	Center offset in direction
TT (tool measurement)		20	1	Center point axis 1
			2	Center point axis 2
			3	Center point axis 3
		21	-	Effective radius
		22	1	Probing position 1 axis X
			2	Probing position 1 axis Y
			3	Probing position 1 axis Z
		23	1	Probing position 2 axis X
			2	Probing position 2 axis Y
			3	Probing position 2 axis Z
		24	1	Probing position 3 axis X
			2	Probing position 3 axis Y
			3	Probing position 3 axis Z

Group name	Group number ID......	Sys. data number NR......	Sys. data index IDX.....	Input value for system datum
		25	1	Probing position 4 axis X
			2	Probing position 4 axis Y
			3	Probing position 4 axis Z
Measuring touch probe		30		Effective length
		31		Effective radius 1
		32		Effective radius 2
		33		Diameter of calibration ring
		34	1	Center offset (main axis)
			2	Center offset (secondary axis)
		35	1	Compensation factor axis 1
			2	Compensation factor axis 2
			3	Compensation factor axis 3
		36	1	Power ratio axis 1
			2	Power ratio axis 2
			3	Power ratio axis 3
Transformations from OEM cycle	420	0	0	0 = not global 1 = global
Overwrite current datum table	500	Line	Column	
Approach behavior during programmed probing	990	1	–	0 = standard behavior 1 = effective radius and setup clearance zero

Transfer with Machine Parameter

122 different machine parameters are reserved for data transfer in the PLC. MP4210.x, MP4220.x and MP4310.x are stored in PLC words. The contents of MP4230.x and 4231.x must be called by module 9032. For example, PLC positioning, datum shifts, feed rates for PLC positioning or coding for the release of certain PLC functions can be filed in these machine parameters. These numerical values are evaluated in the PLC program.

The control rounds input values of < 0.001 mm (or °) internally to 0.001 mm (or °).

		Set	Reset
D768	Value from MP4210.0	NC	NC
D772	Value from MP4210.1		
D776	Value from MP4210.2		
D780	Value from MP4210.3		
D784	Value from MP4210.4		
D788	Value from MP4210.5		
D792	Value from MP4210.6		
D796	Value from MP4210.7		
D800	Value from MP4210.8		
D804	Value from MP4210.9		
D808	Value from MP4210.10		
D812	Value from MP4210.11		
D816	Value from MP4210.12		
D820	Value from MP4210.13		
D824	Value from MP4210.14		
D828	Value from MP4210.15		
D832	Value from MP4210.16		
D836	Value from MP4210.17		
D840	Value from MP4210.18		
D844	Value from MP4210.19		
D848	Value from MP4210.20		
D852	Value from MP4210.21		
D856	Value from MP4210.22		
D860	Value from MP4210.23		
D864	Value from MP4210.24		
D868	Value from MP4210.25		
D872	Value from MP4210.26		
D876	Value from MP4210.27		
D880	Value from MP4210.28		
D884	Value from MP4210.29		
D888	Value from MP4210.30		
D892	Value from MP4210.31		
D896	Value from MP4210.32		
D900	Value from MP4210.33		
D904	Value from MP4210.34		
D908	Value from MP4210.35		
D912	Value from MP4210.36		
D916	Value from MP4210.37		
D920	Value from MP4210.38		
D924	Value from MP4210.39		
D928	Value from MP4210.40		
D932	Value from MP4210.41		

D936	Value from MP4210.42
D940	Value from MP4210.43
D944	Value from MP4210.44
D948	Value from MP4210.45
D952	Value from MP4210.46
D956	Value from MP4210.47

W960	Value from MP4220.0
W962	Value from MP4220.1
W964	Value from MP4220.2
W966	Value from MP4220.3
W968	Value from MP4220.4

W976	Value from MP4310.0
W978	Value from MP4310.1
W980	Value from MP4310.2
W982	Value from MP4310.3
W984	Value from MP4310.4
W986	Value from MP4310.5
W988	Value from MP4310.6

MP4210.0-47 Set a number in the PLC
Input: –99 999.9999 to +99 999.9999

MP4220.0-4 Machine parameter with multiple function
Input: 10 to 30 000
- Set a number in the PLC. In word range W960 to W968.
- Feed rate for re-approaching the contour

MP4310.0-6 Set a number in the PLC, in the word range W 976 to W 988
Input: 0 to 65 535

MP4230.0-31 Set a number in the PLC
Input: –99 999.9999 to +99 999.9999

MP4231.0-31 Set a number in the PLC
Input: –99 999.9999 to +99 999.9999

Data transfer with strobes

The transfer of certain data to the PLC is controlled by strobes. M codes, S codes , T codes, G codes and Q code are transferred in this manner.

Example:
When an M function is output, the NC sets the strobe signal M4072. After evaluating the M function, the PLC sets the acknowledge marker M4092. The PLC must reset M4092 otherwise no further strobes can be transferred by the NC.

5.2.5 Data Transfer PLC → NC

Transfer with FN18
With function FN18: SYS-DATUM READ you can read system data that you find in the table below and save them in Q parameters. FN18 was introduced primarily for certain tasks in OEM cycles. The system data are selected by group number and index.
Syntax: FN18: SYSREAD Qxxx = IDxxxx NRxxxx IDXxxxx or numerical value; comment

Group name	Group number	Sys. data number	Sys. data index	System data for
Program information	10	1	-	mm=0, inch=1
		2	-	Overlap factor at the pocket milling
		3	-	Number of active fixed cycle
		4	-	Number of the last DEF-active OEM cycle
Machine status	20	1	-	Active tool number
		2	-	Prepared tool number
		3	-	Active tool axis 0 = X 6 = U 1 = Y 7 = V 2 = Z 8 = W
		4	-	Programmed spindle speed
		5	-	Active spindle state 0 = off, 1 = on
		8	-	Coolant state 0 = off, 1 = on
		9	-	Active feed rate
Cycle parameters	30	1	-	Setup clearance
		2	-	Total hole depth / milling depth
		3	-	Pecking depth
		4	-	Feed rate for plunging
		5	-	1st side length for pocket
		6	-	2nd side length for pocket
		7	-	1st side length for slot
		8	-	2nd side length for slot
		9	-	Radius for circular pocket
		10	-	Feed rate for milling

Group name	Group number	Sys. data number	Sys. data index	System data for
		11	-	Directional sense of the milling path
		12	-	Dwell time
		13	-	Thread pitch
		14	-	Finishing allowance
		15	-	Rough-out angle
Data from the tool table	50	1	-	Tool length
		2	-	Tool radius
		3	-	Tool radius R2
		4	-	Oversize for tool length DL
		5	-	Oversize for tool radius DR
		6	-	Oversize for tool radius DR2
		7	-	Tool locked 0 = not locked; 1 = locked
		8	-	Number of the replacement tool
		9	-	Maximum tool life TIME1
		10	-	Maximum tool life TIME2
		11	-	Current tool time CUR. TIME
		12	-	PLC status
		13	-	Maximum tool length LCUTS
		14	-	Maximum plunge angle ANGLE
		15	-	TT: Number of teeth CUT
		16	-	TT: Tolerance for wear in length LTOL
		17	-	TT: Tolerance for wear in radius RTOL
		18	-	TT: Direction of rotation DIRECT 0 = pos; –1 = neg
		19	-	TT: Offset in plane R-OFFS R = 99999.9999

Group name	Group number	Sys. data number	Sys. data index	System data for
		20	-	TT: Offset in length L-OFFS
		21	-	TT: Breakage tolerance in length LBREAK
		22	-	TT: Breakage tolerance in radius RBREAK
Data from the pocket table	51	1	Pocket no.	Tool no.
		2	Pocket no.	0 = no special tool 1 = special tool
		3	Pocket no.	0 = no fixed pocket 1 = fixed pocket
		4	Pocket no.	0 = no locked pocket 1 = locked pocket
		5	Pocket no.	PLC status
Tool pocket	52	1	Tool no.	Pocket no.
Values programmed in the Tool Call	60	1	-	Tool number
		2	-	Tool axis 0 = X 6 = U 1 = Y 7 = V 2 = Z 8 = W
		3	-	Spindle speed
		4	-	Oversize for tool length DL
		5	-	Oversize for tool radius DR
		6	-	Automatic TOOL CALL 0= yes 1= no
		7	-	Oversize in tool radius DR2
Programmed position after TOOL CALL	70	1	-	1 = valid position
		2	1	Position axis X
		2	2	Position axis Y
		2	3	Position axis Z
		3	-	Feed rate (–1= no programmed feed rate)

Group name	Group number	Sys. data number	Sys. data index	System data for
Tool compensation	200	1	-	Current radius (including oversize)
		2	-	Current length (including oversize)
Active transformations	210	1	-	Basic rotation (manual)
		2	-	Programmed rotation
		3	-	Active mirrored axis Bit 0 to 2 and 6 to 8: Axis X, Y, Z and U, V, W
		4	1	Active scaling factor axis X
			2	Active scaling factor axis Y
			3	Active scaling factor axis Z
			7	Active scaling factor axis U
			8	Active scaling factor axis V
			9	Active scaling factor axis W
		5	1	3-D ROTATION A
			2	3-D ROTATION B
			3	3-D ROTATION C
		6	-	Tilt working plane (0= not active; –1= active)
Traverse range	230	2	1 to 9	Negative software limit switch axis 1 to 9
		3	1 to 9	Positive software limit switch axis 1 to 9
Nominal position of REF system	240	1	1 to 9	Axis 1 to 9
Current position	270	1	1 to 9	Axis 1 to 9
TS touch trigger probe	350	10	-	Tool axis
		11	-	Effective radius
		12	-	Effective length
		13	-	Radius calibration ring
		14	1	Center offset (main axis)
			2	Center offset (secondary axis)
		15	-	Center offset in direction

Group name	Group number	Sys. data number	Sys. data index	System data for
TT (tool measurement)		20	1	Center point axis 1
			2	Center point axis 2
			3	Center point axis 3
		21	-	Effective radius
		22	1	Probing pos. 1 axis X
			2	Probing pos. 1 axis Y
			3	Probing pos. 1 axis Z
		23	1	Probing pos. 2 axis X
			2	Probing pos. 2 axis Y
			3	Probing pos. 2 axis Z
		24	1	Probing pos. 3 axis X
			2	Probing pos. 3 axis Y
			3	Probing pos. 3 axis Z
		25	1	Probing pos. 4 axis X
			2	Probing pos. 4 axis Y
			3	Probing pos. 4 axis Z
Measuring touch probe		30		Effective length
		31		Effective radius 1
		32		Effective radius 2
		33		Diameter of calibration ring
		34	1	Center offset (main axis)
			2	Center offset (auxiliary axis)
		35	1	Compensation factor axis 1
			2	Compensation factor axis 2
			3	Compensation factor axis 3
		36	1	Power ratio axis 1
			2	Power ratio axis 2
			3	Power ratio axis 3
Datum table	500	Line	Column	
Pallet table	510	1	-	Active line
		2	-	Pallet no. from column "Name"
Approach behavior during programmed probing	990	1	-	0 = standard behavior 1 = effective radius and setup clearance zero

Group name	Group number	Sys. data number	Sys. data index	System data for
Machine parameters	1000	MP number	MP index	Machine parameter value
PLC data	2000	10	Marker no.	PLC marker
		20	Input no.	PLC input
		30	Output no.	PLC output
		40	Counter no.	PLC counter
		50	Timer no.	PLC timer
		60	Byte no.	PLC byte
		70	Word no.	PLC word
		80	Doubleword no.	PLC doubleword

Interrogate PLC operands in the NC program; FN20: WAIT FOR

With the NC block FN20: WAIT FOR you can interrupt the NC program until a programmed condition is fulfilled. Permissible conditions are comparisons of a PLC operand with a constant.

Permissible PLC operands: M, B, W, D, T, C, I, O

Permissible conditions:		
	==	equal to
	!= or <>	not equal to
	<	less than
	>	greater than
	<=	less than or equal to
	>=	greater than or equal to

If there is no condition given, the interruption will continue until the operand = 0.

Examples:

`FN20: WAIT FOR I10==1`	NC program will only continue if PLC input I10 is set
`FN20: WAIT FOR I10`	NC program will only continue if PLC input I10 is equal to 0
`FN20: WAIT FOR B3000>255`	NC program will only continue if byte 3000 is set to a value greater than 255.

Transfer by Q parameters

Numerical values are transferred from the PLC to the part program by means of Q parameters Q100 to Q107, which means that the Q parameters Q100 to Q107 can be overwritten by the PLC. The numerical value is registered in double word D528 and the Q parameter numbers 0 to 7 are defined in W516. Transfer is activated with the strobe marker M4131. The Q parameter values are taken over with the next M/S/T strobe.

		Set	Reset
M4131	Activate Q-parameter transfer to the NC Data from D528; Q-no. from W516	PLC	NC
D528	Doubleword with multiple-function. Here: data from transfer from the PLC to the NC (Q-no. from W516, strobe marker M4131)	PLC	PLC
W516	Q nos. 0 to 7 with numerical transfer from PLC to NC (Transfer of the value from D528, strobe marker M4131)	PLC	PLC

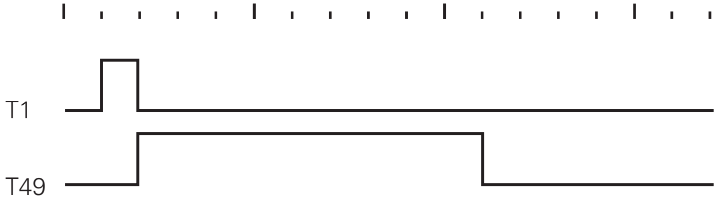
5.2.6 Timer

48 timers are available in the PLC. These 48 timers are controlled by special markers with the abbreviation symbol T. The time period for the timer is defined in the machine parameter MP4110.X. The time unit corresponds to the PLC cycle time (21 ms).

The timers are started by setting the markers T0 to T47 which also sets the timer to the value from MP4110.X. This activation may only be performed for a single PLC run, as otherwise the timers will be restarted by every succeeding run.

Markers T48 to T95 (timer running) will remain set for the period defined in the machine parameters.

Example:
Start of Timer 1
Period in MP4110.1 = 9 (PLC cycles)



Timer start	Timer is running	Machine Parameters
T0	T48	MP4110.0
T1	T49	MP4110.1
T2	T50	MP4110.2
T3	T51	MP4110.3
T4	T52	MP4110.4
T5	T53	MP4110.5
T6	T54	MP4110.6
T7	T55	MP4110.7
T8	T56	MP4110.8
T9	T57	MP4110.9
T10	T58	MP4110.10
T11	T59	MP4110.11
T12	T60	MP4110.12
T13	T61	MP4110.13
T14	T62	MP4110.14
T15	T63	MP4110.15
T16	T64	MP4110.16
T17	T65	MP4110.17

Timer start	Timer is running	Machine parameters
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

MP4110.0-31 Preset value for timers T0 to T47
Input range: 0 to 65 535 [PLC cycle times]

Setting and starting PLC timers (Module 9006)
Module 9006 sets the cycle time for a PLC timer T0 to T47 and starts the timer.

Constraints:

- If during a PLC scan a timer 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.
- The corresponding timer from T48 to T96 is set immediately after the module is called. T0 to T47 is not set.
- The cycle time is transferred with the unit of measure millisecond [ms].
- The effective cycle time is rounded upward to whole-number PLC cycle times.
- The PLC scan can be interrupted by resetting the timers T48 to T95.

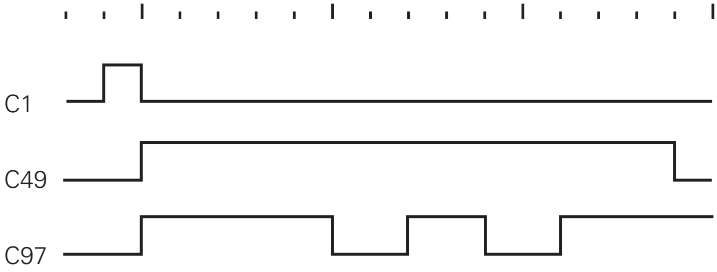
Call:

```
PS   B/W/D/K    <Timer no.>
                Input value: 0 to 47
PS   B/W/D/K    <Cycle time>
                Input value: 0 to 1 000 000 ms
CM   9006
```

5.2.7 Counters

Thirty-two counters are available in the PLC. Each of these 32 counters is controlled by special markers with the abbreviation symbol C. After setting a marker from the range C0 to C31 the counter is loaded with the value from machine parameter MP4120.X. The marker range C48 to C79 indicates whether the count has been completed or not. The marker range C96 to C127 is used to start the counter (counter release pulse).

Example: Logic diagram for counter C1
Contents of machine parameter MP4120.1 = 10 (PLC cycles)



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
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 Preset value for counters C0 to C31
Input range: 0 to 65 535 [PLC cycles]

5.2.8 Fast PLC Inputs

With MP4130 you can define PLC inputs that are not interrogated in the PLC cycle (21 ms), but rather in the control loop cycle (3 ms). The markers M4590 to M4593 show the current state of the fast PLC inputs. The function of the fast PLC inputs must be activated in the PLC program with W522 bit 2 to bit 5. In order to ensure that a signal change be recognized, the duration of the signal at the fast PLC input must be at least 4 ms.

MP4130 Numerical designation of fast PLC inputs
Input: 0 to 255 [No. of the PLC input]

MP4130.2 Fast PLC input sets M4590
MP4130.3 Fast PLC input sets M4591
MP4130.4 Fast PLC input sets M4592
MP4130.5 Fast PLC input sets M4593

MP4131.2-5 Condition for activating the fast PLC inputs
Input: 0 = activation at low level
1 = activation at high level

		Set	Reset
W522	Activate the fast PLC inputs	PLC	PLC
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		

5.3 Hard-Disk Organization

The hard disk of the TNC is divided into three partitions:

TNC: 1500 megabytes[1] for user data. Here you store the part programs, tool tables, datum tables and pallet tables.

PLC: 64 megabytes for manufacturer's data. This partition is for system files, PLC programs, machine parameters, help files, PLC dialogs, PLC error tables, compensation value tables, and OEM cycles. To access the PLC partition you must enter the code number 807667.

SYS: 128 megabytes for system-specific files (system files, NC dialogs, HEIDENHAIN cycles, etc.). The SYS partition is not visible and cannot be accessed!

Note: Changes in the SYS partition can impair the proper functioning of your system!

The PLC partition is most significant for the machine tool builder. The following is an example of a meaningful directory structure for 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 (generated automatically)
ENGLISH	PLC dialogs and error messages ∗.A; Help files ∗.HLP
FRENCH	PLC dialogs and error messages ∗.A; Help files ∗.HLP
GERMAN	PLC dialogs and error messages ∗.A; Help files ∗.HLP
ITALIAN	PLC dialogs and error messages ∗.A; Help files ∗.HLP
SPANISH	PLC dialogs and error messages ∗.A; Help files ∗.HLP
MP	Machine parameters ∗.MP
NC_MACRO	NC macros
OEMCYC	Directory for manufacturer's cycles, generated by **CycleDesign**
DES	Generated by CycleDesign
ELE	Generated by CycleDesign
NC	Generated by CycleDesign
SK	Generated by CycleDesign
PLCSOFTK	Graphics for PLC soft keys

[1] NC software 280 470 xx: TNC 900 megabytes, PLC 32 megabytes, SYS 64 megabytes

Description of the system files (∗.SYS):

OEM.SYS

Code words for calling certain functions are registered in OEM.SYS. 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. Most entries in OEM.SYS must be made manually or with Module 9271 (see below for exceptions). Module 9271 overwrites the contents of existing code words and adds new code words at the end of the file. MPFILE and PLCMAIN cannot be written with Module 9271. You can read the entries in OEM.SYS with Module 9270.

The following code words are available:

- **MPFILE**
  Path for active MP file. Mandatory. If you have loaded an MP file and you exit the editor, the MP file is automatically registered in OEM.SYS!
  Example for entry: MPFILE = PLC:\MP\NC430V02.MP
- **PLCMAIN**
  Path for active PLC program. Mandatory. If you compile a new PLC program it is automatically entered in OEM.SYS!
  Example for entry: PLCMAIN = PLC:\PLC_PGM\MAIN_430.PLC
- **PLCPWM**
  Path for PLC program for putting digital axes into operation
  Example for entry: PLCPWM = PLC:\IB_PGM\IB430.PLC
- **PLCERRTAB**
  Path for PLC error-message table. Mandatory for PLC error messages. If you compile a new PLC program it is automatically entered in OEM.SYS!
  Example for entry: PLCERRTAB = PLC:\ PLC_PGM \ERR_TAB.PET
- **PLCERROR**
  Name of text file for PLC error messages. The path for the text file is fixed.
  Example for entry: PLCERROR = PLC_ERR.A
- **PLCDIALOG**
  Name of text file for PLC dialogs. The path for the text file is fixed. Example for entry:
  PLCDIALOG = DIALOG.A
- **PLCSOFTVERS**
  PLC software version, is displayed on screen with MOD. Mandatory.
- **TABCMA**
  Path for the compensation value table for axis error compensation (see nonlinear axis-error compensation)
  Example for entry: TABCMA = PLC:\AXIS_COR\CORRECT.CMA;
- **MODEHELP**
  Path for help texts and machine commands
  Example for entry: MODEHELP = PLC:\LANGUAGE\GERMAN\OPTIMIER.HLP
- **PLCPASSWORD**
  Code number for calling PLC operation (instead of 807667)
  Example for entry: PLCPASSWORD = 123456789
  Note: Do not enter a code number that was already assigned by HEIDENHAIN!
- **TTYP** Path and file name for list of machine tool types
- **PLCERRFIX** Path for help text "Corrective action"
- **PLCERRREASON** Path for help text "Cause of error"
- **PLCEVENTS** Path for list of events (SPAWN command)
- **LSV2TIME0** Timeout for receiving block (STX to ETX)
- **LSV2TIME1** Timeout for acknowledging ENQ or check sum
- **LSV2TIME2** Timeout when sending DLE 0, DLE 1 or NAK until valid character is sent

Call only in submit or SPAWN:

```
PS   B/W/D/K/S  <String with code word>
PS   B/W/D/K    <String number for result>                [0 to 3]
CM   9270
                M4203 =   0: Entry read
                          1: Error see W1022
                W1022 =   3: String not valid for code word or result
                          12: String for code word too long
                          20: Call was not in submit or SPAWN
                          30: Code word was not found
```

Call only in submit or SPAWN:

```
PS   B/W/D/K/S  <String with code word >
PS   B/W/D/K    <String number for entry>
CM   9271
                M4203 =   0: Entry was written
                          1: Error see W1022
                W1022 =   3: String not valid for code word or entry
                          6: PLCMAIN or MPFILE transferred
                          12: String for code word too long
                          20: Call was not in submit or SPAWN
```

MGROUPS.SYS
The M functions to be output after a block scan are defined in the system files PLC:\MGROUPS.SYS and PLC:\MSPLIT.SYS (see section "Returning the Contour).

MSPLIT.SYS
M functions that affect several groups are resolved into partial functions in the file MSPLIT.SYS (see section "Returning to the Contour").

PLCSOFTK.SYS
Path for the file names of the PLC soft-key graphics (see section "PLC Soft Keys").

CYCLE.SYS
Definition of the soft-key structure if OEM cycles were integrated. This file is created automatically by the PC software **CycleDesign** (see OEMCYC directory).

NCMACRO.SYS
Name of the NC macro for tool and pallet change.
TC= <program name of the tool change program>
PALETT= < program name of the pallet change program >
Example for entry: TC =PLC:\NC_MACRO\TOOLCALL.H

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>

5.4 Program Creation

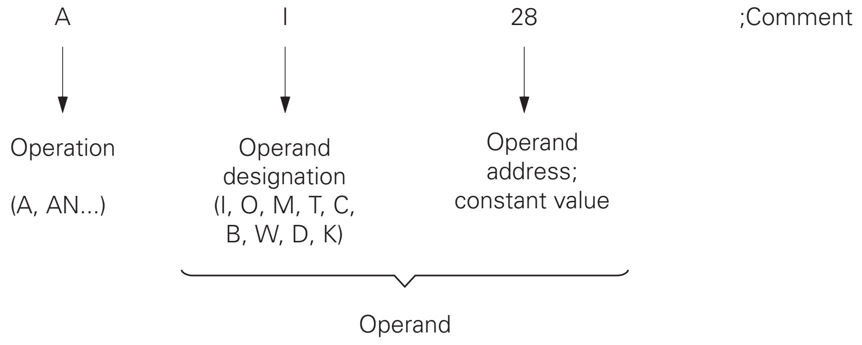
5.4.1 ASCII Editor

With the integrated editor you can create the PLC program and the other necessary files using the ASCII keyboard of the control.

For a detailed description of the editor including the associated soft keys see the User's Manual for the control.

5.4.2 Program Format

Command
A command is the smallest unit of a PLC program. It consists of an operation part and an operand part.



The operation describes the function to be performed. It explains what is done with the operands.

The operand shows what is to be operated on. It consists of the operand abbreviation and a parameter (Address). Register and memory contents can be gated, erased and loaded by using PLC commands.

Both bit and word execution are possible. In word execution it is possible to address memory contents with a length of 8 bits (byte), 16 bits (word) or 32 bits (doubleword).

The control can recognize an input error immediately upon entry and respond by displaying an error message. Refer to the Appendix for a list of these error messages.

5.4.3 Program structure

In order to give the program a transparent structure and make it easier to maintain and expand the PLC program it is important for it to have a modular structure. This means that for each function you write a program module.

You can then call the module from the main program (see section "PLC Program Example"). Improper functioning of the machine should be interrogated in the PLC program and indicated by a plain-language error message on the screen.

For debug purposes you can interrogate the contents of the processing stack using Module 9019. This function indicates the number of bytes that are currently present in the processing stack of the PLC. If the stack is empty, the value zero is sent. One byte, word or doubleword occupies four bytes in the stack, one marker, input, output, timer or counter occupies two bytes.

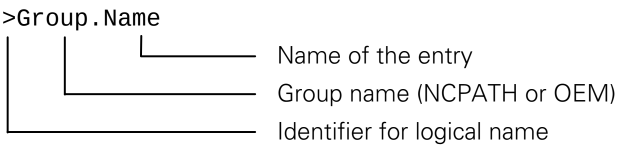
Call only in the submit job:

```
CM   9019
PL   B/W/D        <No. of bytes in processing stack>
```

5.4.4 Logical Names for Files

Instead of the file name you can also enter a logical name. This will be of most use when transferring file names to PLC modules

Syntax:



```
>Group.Name
```

Name of the entry

Group name (NCPATH or OEM)

Identifier for logical name

Example:

>NCPATH.NCEDIT	Transfers the complete name and path of the file which is currently selected in the editing mode.
>OEM.PLCMAIN	Transfers the complete name and path of the PLC program that was entered in the file OEM.SYS with the command PLCMAIN.

List of the logical names:

Group	Entry	Meaning
NCPATH	PLCEDIT	Selected file in PLC programming mode
	NCEDIT	Selected file in programming mode
	RUNPGM	Selected file in executing mode
	RUNDATUM	Selected datum table in executing mode
	SIMPGM	Selected file in test run mode
	SIMDATUM	Selected datum table in test run mode
	SIMTOOL	Selected tool table in test run mode
	RUNBRKPGM	Target file in the block scan in executing mode
	SIMBRKPGM	Target file in the block scan in test run mode
	MDIPGM	Selected file in positioning with handwheel mode
	TCHPATH	Selected datum table for manual probe function
OEM	TABCMA	Current compensation value table
	MODEHELP	Current help file
	PLCMAIN	Current PLC main program
	PLCPWM	Current PLC commissioning program for digital axes
	PLCEVENTS	Current event list for SPAWN command
	PLCERRTAB	Current PLC error message list (PET)
	WMAT	Current material list
	TMAT	Current workpiece material list
	MPFILE	Current machine parameter list
	Your own entry	In the file OEM.SYS you can enter the desired file name with path after your own entry. e.g. HUGO=TNC:\HUGO\TEST.H

5.4.5 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
Input: %xxxxx

Bit 0= Convert "axis" words (W1024 and following) into markers
Bit 1= Convert "new" markers (4000 and following) into "old" markers (2000 and following)
Bit 2= Convert configuration bits from MP4310 into markers (M2192 to M2239 and M3200 to M3263)
Bit 3= Errors markers are available
Bit 4= Nonvolatile markers in range M1000 to M1999

5.5 PLC Program Example

The following PLC program example was written on a PC with the PLC programming environment PLCdesign. This software is supplied together with additional comprehensive PLC program examples.

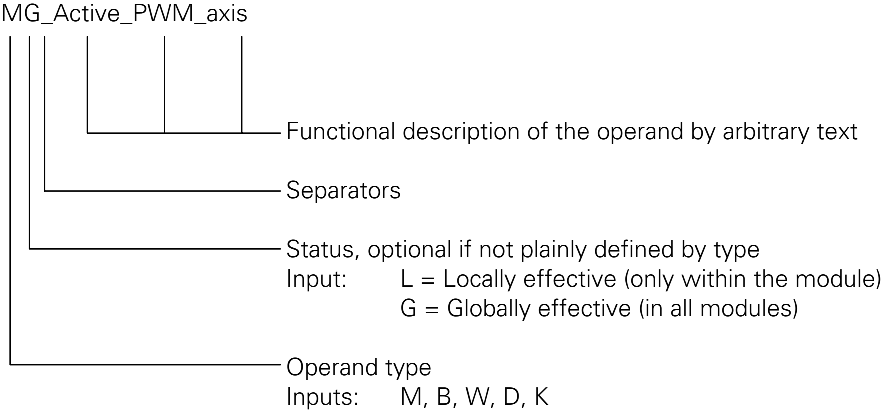
The PLC program is divided into modules that perform individual functions. This gives it a structure that thereby ensures program transparency. A structured, and therefore readable, program format makes it possible at a later time to edit or add to the program.

The program is described in a so-called "documentation file." The program PLCdesign can output this file in addition to the individual PLC programs (see the User's Manual for the PLC Programming Environment PLCdesign).

The right column of the "documentation file" shows the source code of the individual modules as it was written by the programmer with symbolic operands and label names. The left column shows the associated list of statements as they are needed by the TNC. These statement lists are generated automatically by the compiler. 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

Functional description of the operand by arbitrary text

Separators

Status, optional if not plainly defined by type
Input: L = Locally effective (only within the module)
G = Globally effective (in all modules)

Operand type
Inputs: 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 global and are therefore not indicated as such.

Example:	NP_M2008_X_InPos
	I_release_tool
	TS_5_clamp_unclmp
	CS_RS_Err_ReStart

```
PLCCOMP V2.9M  MAIN_PGM.SRC                          Fri Nov 14 08:33:56 1997
                            1
                            2 *+----------------------------------------------------+
                            3 *¦  Main program for TNC 430                          ¦
                            4 *+----------------------------------------------------+
                            5
                            6 #plcpath PLC:\EXAMPLE\
                            7
                            8 *+----------------------------------------------------+
                            9 *¦  Marker range definition                           ¦
                           10 *+----------------------------------------------------+
                           11
                           12 #define /MN 3200 3999
                           13 #define /MR 200   999
                           14 #define /BN 2048 4095
                           15 #define /BR 4     127
                           16
                           17 *+----------------------------------------------------+
                           18 *¦  Global file definition                            ¦
                           19 *+----------------------------------------------------+
                           20
                           21 #define /g  GLB_TCMB.Def

                            1 *+----------------------------------------------------+
                            2 *¦  Global markers Bytes Words  DWords                ¦
                            3 *+----------------------------------------------------+
                            4
                            5 #Type M
    M3999                   6    MG_one_marker
    M3998                   7    MG_zero_marker
    M3997                   8    MG_spindle_on_M03
    M3996                   9    MG_spindle_on_M04
    M3995                  10    MG_spindle_off_M05
    M3994                  11    MG_spi_Pos_M19_R_M0X
    M3993                  12    MG_T_I_N_supervision
    M3992                  13    MG_Spindle_RPM_Zero
                           14
                           15 #Type
    M992                   16 /r MG_closed_loop            M[8]
    M992                   17 /c MG_1_clamp_mode_activ   M &MG_closed_loop +  0
    M993                   18 /c MG_2_clamp_mode_activ   M &MG_closed_loop +  1
    M994                   19 /c MG_3_clamp_mode_activ   M &MG_closed_loop +  2
    M995                   20 /c MG_4_clamp_mode_activ   M &MG_closed_loop +  3
    M996                   21 /c MG_5_clamp_mode_activ   M &MG_closed_loop +  4
    M997                   22 /c MG_S_clamp_mode_activ   M &MG_closed_loop +  5
                           23
    M3984                  24    MG_active_PWM_axis       M[8]
    M3984                  25 /c MG_active_PWM_axis_1     M &MG_active_PWM_axis + 0
    M3985                  26 /c MG_active_PWM_axis_2     M &MG_active_PWM_axis + 1
    M3986                  27 /c MG_active_PWM_axis_3     M &MG_active_PWM_axis + 2
    M3987                  28 /c MG_active_PWM_axis_4     M &MG_active_PWM_axis + 3
    M3988                  29 /c MG_active_PWM_axis_5     M &MG_active_PWM_axis + 4
    M3989                  30 /c MG_active_PWM_axis_S     M &MG_active_PWM_axis + 5
                           31
                           32 #Type
    B4088                  33    BG_MPAxis.x_CA_PA         B[6]
    B4088                  34 /c BG_MPAxis.0_CA_PA_1       B &BG_MPAxis.x_CA_PA + 0
    B4089                  35 /c BG_MPAxis.1_CA_PA_2       B &BG_MPAxis.x_CA_PA + 1
    B4090                  36 /c BG_MPAxis.2_CA_PA_3       B &BG_MPAxis.x_CA_PA + 2
    B4091                  37 /c BG_MPAxis.3_CA_PA_4       B &BG_MPAxis.x_CA_PA + 3
    B4092                  38 /c BG_MPAxis.4_CA_PA_5       B &BG_MPAxis.x_CA_PA + 4
    B4093                  39 /c BG_MPSpin.0_CA_PA_S       B &BG_MPAxis.x_CA_PA + 5
                           40
                           41 #Type W
    W4086                  42    WG_MP10_Active_Axis
    W4084                  43    WG_servo_enable_internal_servo
    W4082                  44    WG_Active_PWM_Axis
                           45
                           46 #Type
    W4068                  47    WG_motor_temp        W[6]
    W4068                  48 /c WG_motor_temp_1      W &WG_motor_temp +  0
    W4070                  49 /c WG_motor_temp_2      W &WG_motor_temp +  2
    W4072                  50 /c WG_motor_temp_3      W &WG_motor_temp +  4
    W4074                  51 /c WG_motor_temp_4      W &WG_motor_temp +  6
    W4076                  52 /c WG_motor_temp_5      W &WG_motor_temp +  8
    W4078                  53 /c WG_motor_temp_S      W &WG_motor_temp + 10
                           54
```

```
                          *+----------------------------------------------------+
                          *¦  Error markers                                     ¦
                          *+----------------------------------------------------+

                          #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

                          *+----------------------------------------------------+
                          *¦  Definition of timers and counters                 ¦
                          *+----------------------------------------------------+

                          #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_IO.Def
                          *
                          *+-----------------------------------------------------+
                          *¦  Assign PLC inputs                                  ¦
                          *+-----------------------------------------------------+
                          *
                          #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
I5                           I_feed_enable        I5
I138                         I_1_axis_Plus            I138
I133                         I_1_axis_Minus           I133
I137                         I_2_axis_Plus            I137
I134                         I_2_axis_Minus           I134
I136                         I_3_axis_Plus            I136
I135                         I_3_axis_Minus           I135
I139                         I_4_axis_Plus            I139
I140                         I_4_axis_Minus           I140
I146                         I_5_axis_Plus            I146
I147                         I_5_axis_Minus           I147
I132                         I_NC_Start               I132
I131                         I_NC_Stop                I131
I141                         I_rapid_button           I141

I129                         I_servo_ready_1          I129
I144                         I_servo_ready_2          I144

M3983                        MG_spindle_on_gear                 M
M3982                        MG_gear_change_activ               M

M991                      /r I_gear_range_1                     M
M990                      /r I_gear_range_2                     M
                          *
                          *+----------------------------------------------------+
                          *¦ Assign 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

                          #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 signals 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 signals                             ¦
                          *+----------------------------------------------------+

M4090                        PN_M4090_quit_G_code                     M4090
M4091                        PN_M4091_quit_S_code                     M4091
M4092                        PN_M4092_quit_M_function                 M4092

                          *+----------------------------------------------------+
                          *¦  Strobe signals 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

                          *+----------------------------------------------------+
                          *¦  Arithmetic or module error in PLC                 ¦
                          *+----------------------------------------------------+

M4203                        NP_M4203_error_Modul_9xxx                M4203

                          *+----------------------------------------------------+
                          *¦  Markers that can be influenced by MPs             ¦
                          *+----------------------------------------------------+

M4300                        NP_M4300_PowerFailOn_MP4310.0_Bit_00     M4300

                          *+----------------------------------------------------+
                          *¦  Additional keys                                   ¦
                          *+----------------------------------------------------+

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 (Words bit-coded 54ZYX)             ¦
                          *+----------------------------------------------------+
                          *+----------------------------------------------------+
                          *¦  Axes                                              ¦
                          *+----------------------------------------------------+

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
```

```
                        94 *+-----------------------------------------------------+
                        95 *¦  Data for strobe signals                            ¦
                        96 *+-----------------------------------------------------+
     D756               97    PN_D756_S_nominal_rpm_PLC               D756
     W260               98    NP_W260_M_code                          W260
     W256               99    NP_W256_G_code_spindle                  W256
                       100
     W302              101    NP_W302_Number_PLC_Soft_Key             W302
     W274              102    NP_W274_Button_Key_Code                 W274

                        24 #define /g  Config.Def

                         1
                         2 *+----------------------------------------------------+
                         3 *¦ Configuartion file for this PLC program            ¦
                         4 *+----------------------------------------------------+
                         5
                         6 * Read MP MP2000 or MP120
                         7 * for NC-SW 280470 xx or 280472 yy
     K280472             8    #define MP_Read_NC_SW        280472
                         9
     K0                 10    #define First_PL    0       * PL boards
                        11
     K0                 12    #define NC_Type_PA  0       * Control_type_CA_PA
                        13
     K5                 14    #define Max_NC_Axis 5       * NC-axis without spindle
                        15
     K255               16    #define Motor_Temp_1   255  * supervison motor temperatur
axis 1
     K255               17    #define Motor_Temp_2   255  * supervison motor temperatur
axis 2
     K255               18    #define Motor_Temp_3   255  * supervison motor temperatur
axis 3
     K255               19    #define Motor_Temp_4   255  * supervison motor temperatur
axis 4
     K255               20    #define Motor_Temp_5   255  * supervison motor temperatur
axis 5
     K255               21    #define Motor_Temp_S   255  * supervison motor temperatur
axis S

                        25
                        26 *+-----------------------------------------------------+
                        27 *¦  List of include files                              ¦
                        28 *+-----------------------------------------------------+
                        29
                        30 #if  MP_Read_NC_SW = 280470
                        31 USES Initi470.Src
                        32 #else
   1 USES PLC:\EXAMPLE\IN   33 USES Initi472.Src
                        34 #endif
   2 EXTERN INITIALIZATIO   35 EXTERN initialization
                        36
   3 USES PLC:\EXAMPLE\M_   37 USES M_Funct.Src
   4 EXTERN M_FUNCTION      38 EXTERN M_Function
                        39
   5 USES PLC:\EXAMPLE\RE   40 USES Ref_Endl.Src
   6 EXTERN REFERENCE_END   41 EXTERN reference_endswitch
                        42
   7 USES PLC:\EXAMPLE\DI   43 USES DircBut.Src
   8 EXTERN MANUEL_BUTTON   44 EXTERN Manuel_button_funcktion
                        45
   9 USES PLC:\EXAMPLE\AX   46 USES Axis.Src
  10 EXTERN NC_AXIS         47 EXTERN NC_Axis
                        48
  11 USES PLC:\EXAMPLE\SP   49 USES Spindle.Src
  12 EXTERN SPINDLE_FUNCT   50 EXTERN spindle_function
                        51
  13 USES PLC:\EXAMPLE\GE   52 USES Gear.Src
  14 EXTERN GEAR_CHANGING   53 EXTERN Gear_Changing
                        54
  15 USES PLC:\EXAMPLE\HE   55 USES HelpDiag.Src
  16 EXTERN AXIS_SUPERVIS   56 EXTERN Axis_Supervision
                        57
  17 USES PLC:\EXAMPLE\SO   58 Uses Softkeys.Src
  18 EXTERN PLC_SOFT_KEYS   59 Extern PLC_Soft_keys
                        60
```

The actual program code begins here

```
                              61 *+----------------------------------------------------+
                              62 *¦  PLC program                                       ¦
                              63 *+----------------------------------------------------+
                              64
                              65 #ifdef First_PL
  19 PS K0                    66   PS     K+0
  20 CM 9002                  67   CM     9002
  21 L M4203                  68   L      NP_M4203_error_Modul_9xxx
  22 S M4801                  69   S      PN_error_mod_9002
                              70 #endif
                              71
  23 L M4172                  72   L      NP_M4172_1_PLC_after_power_on
  24 O M4173                  73   O      NP_M4172_1_PLC_after_compile
  25 O M4174                  74   O      NP_M4172_1_PLC_after_MP_edit
  26 CMT INITIALIZATION       75   CMT    initialization
                              76
  27 CM M_FUNCTION            77   CM     M_Function
                              78
  28 L W1032                  79   L      NP_W1032_reference_necessary
  29 <> K0                    80   <>     K+0
  30 O M4155                  81   O      NP_M4155_reference_mode
  31 CMT REFERENCE_ENDSWI     82   CMT    reference_endswitch
                              83
  32 CM MANUEL_BUTTON_FU      84   CM     Manuel_button_funcktion
  33 CM NC_AXIS               85   CM     NC_Axis
  34 CM SPINDLE_FUNCTION      86   CM     spindle_function
  35 CM GEAR_CHANGING         87   CM     Gear_Changing
  36 CM AXIS_SUPERVISION      88   CM     Axis_Supervision
  37 CM PLC_SOFT_KEYS         89   CM     PLC_Soft_keys
                              90
                              91 #ifdef First_PL
  38 PS K0                    92   PS     K+0
  39 CM 9005                  93   CM     9005
  40 L M4203                  94   L      NP_M4203_error_Modul_9xxx
  41 S M4802                  95   S      PN_error_mod_9005
                              96 #endif
  42 EM                       97 EM
________________________________ Local Labels _________________________________

9002                                                       : 9002     LBL 9002
     67
9005                                                       : 9005     LBL 9005
     93

                               1
                               2 *+----------------------------------------------------+
                               3 *¦  Initialize PLC program                            ¦
                               4 *+----------------------------------------------------+
                               5
   1 GLOBAL INITIALIZATIO      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
   2 LBL INITIALIZATION       15 LBL initialization
   3 LN M3999                 16   LN     MG_one_marker
   4 S M3999                  17   S      MG_one_marker
                              18
   5 L M3998                  19   L      MG_zero_marker
   6 R M3998                  20   R      MG_zero_marker
                              21
   7 L M3999                  22   L      MG_one_marker
   8 S M4572                  23   S      NP_M4572_enable_jog_mode_Posit
   9 R M992                   24   R      MG_1_clamp_mode_activ
  10 R M993                   25   R      MG_2_clamp_mode_activ
  11 R M994                   26   R      MG_3_clamp_mode_activ
  12 S M995                   27   S      MG_4_clamp_mode_activ
  13 S M996                   28   S      MG_5_clamp_mode_activ
                              29
  14 L M4300                  30   L      NP_M4300_PowerFailOn_MP4310.0_Bit_00
  15 IFT                      31   IFT
  16 PS K1                    32     PS     KL_On_Power_Fail
  17 ELSE                     33   ELSE
  18 PS K0                    34     PS     KL_Off_Power_Fail
  19 ENDI                     35   ENDI
  20 CM 9167                  36   CM     9167
  21 PLW                      37   PLW
  22 <> K0                    38   <>     K+0
  23 S M4800                  39   S      PN_error_mod_9167
                              40
  24 L K255                 41   L      Motor_Temp_1
  25 = W4068                42   =      WG_motor_temp_1
  26 L K255                 43   L      Motor_Temp_2
  27 = W4070                44   =      WG_motor_temp_2
  28 L K255                 45   L      Motor_Temp_3
  29 = W4072                46   =      WG_motor_temp_3
  30 L K255                 47   L      Motor_Temp_4
  31 = W4074                48   =      WG_motor_temp_4
  32 L K255                 49   L      Motor_Temp_5
  33 = W4076                50   =      WG_motor_temp_5
  34 L K255                 51   L      Motor_Temp_S
  35 = W4078                52   =      WG_motor_temp_S
                            53
  36 RPLY B4067             54   RPLY   BL_MPs_read_identify
  37 <> K0                  55   <>     K+0
  38 EMT                    56   EMT
  39 SUBM MPS_READ_SUBMIT   57   SUBM   MPs_read_Submit
  40 = B4067                58   =      BL_MPs_read_identify
  41 == K0                  59   ==     K+0
  42 S M4804                60   S      PN_error_Submit_Queue_Full
  43 EM                     61 EM
                            62
  44 LBL MPS_READ_SUBMIT    63 LBL MPs_read_Submit
  45 L K0                   64   L      K+0
  46 = W4080                65   =      WL_Index_Reg
  47 =X                     66   =X
  48 REPEAT                 67   REPEAT
  49 PS KF MP_READ_TABLE[   68     PS     KF MP_Read_Table[X]
  50 INCX                   69     INCX
  51 PS KF MP_READ_TABLE[   70     PS     KF MP_Read_Table[X]
  52 CM 9032                71     CM     9032
  53 INCX                   72     INCX
  54 L KF MP_READ_TABLE[X   73     L      KF MP_Read_Table[X]
  55 = B4066                74     =      BL_Case
  56 INCX                   75     INCX
  57 L KF MP_READ_TABLE[X   76     L      KF MP_Read_Table[X]
  58 =X                     77     =X
  59 CASE B4066             78     CASE   BL_Case
  60 CM PL_BYTE_INDEX       79       CM     PL_Byte_Index
  61 CM PL_WORD_INDEX       80       CM     PL_Word_Index
  62 CM PL_DWORD_INDEX      81       CM     PL_DWord_Index
  63 ENDC                   82     ENDC
  64 L W4080                83     L      WL_Index_Reg
  65 + K4                   84     +      K+4
  66 = W4080                85     =      WL_Index_Reg
  67 =X                     86     =X
  68 L KF MP_READ_TABLE[X   87     L      KF MP_Read_Table[X]
  69 < K0                   88     <      K+0
  70 UNTILT                 89   UNTILT
                            90
  71 L K0                   91   L      K+0
  72 =X                     92   =X
  73 REPEAT                 93   REPEAT
  74 PS KF AXISNUMBER[X]    94     PS     KF AxisNumber[X]
  75 PS K2 ; AXIS UNDER C   95     PS     K+2                   ; Axis is controlled (0=no,
1=yes)
  76 CM 9038                96     CM     9038
  77 PS KF AXISNUMBER[X]    97     PS     KF AxisNumber[X]
  78 PS K8 ; AXIS DIGITAL   98     PS     K+8                   ; Axis is controlled digitally
  79 CM 9038                99     CM     9038
                           100
  80 PLW                   101     PLW
  81 A[                    102     A[
  82 PLW                   103       PLW
  83 ]                     104     ]
  84 <> K0                 105     <>     K+0
  85 S M3984[X]            106     S      MG_active_PWM_axis[X]
  86 INCX                  107     INCX
  87 LX                    108     LX
  88 > K5                  109     >      K+5
  89 UNTILT                110   UNTILT
                           111
  90 ;-----Spindle Bit fr   112   ;-----Spindle Bit from Bit 5 into Bit 15 copieren
  91 LB M3984              113   LB     MG_active_PWM_axis
  92 = W4082               114   =      WG_Active_PWM_Axis
  93 L M3989               115   L      MG_active_PWM_axis_S
  94 IFT                   116   IFT
  95 L W4082               117     L      WG_Active_PWM_Axis
  96 A K$1F                118     A      K$001F
  97 O K$8000              119     O      K$8000
  98 = W4082               120     =      WG_Active_PWM_Axis
  99 ENDI                  121   ENDI
 100 EM                    122 EM
                           123
 101 LBL PL_BYTE_INDEX     124 LBL PL_Byte_Index
 102 PL B0[X]              125   PL     B0[X]
 103 EM                     126 EM
                            127
 104 LBL PL_WORD_INDEX      128 LBL PL_Word_Index
 105 LX                     129   LX
 106 / K2 ;TYPE CASTING F   130   /      K+2         ;Adapt index type to word index:=byte
address
 107 =X                     131   =X
 108 PL W0[X]               132   PL     W0[X]
 109 EM                     133 EM
                            134
 110 LBL PL_DWORD_INDEX     135 LBL PL_DWord_Index
 111 LX                     136   LX
 112 / K4 ;TYPE CASTING F   137   /      K+4         ;Adapt index type to Dword index:=byte
address
 113 =X                     138   =X
 114 PL D0[X]               139   PL     D0[X]
 115 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
 116 KFIELD MP_READ_TABLE   153 KFIELD MP_Read_Table
 117  K10                   154   K+10
 118  K$0                   155   KL_Index_0
 119  K$1                   156   KL_Word_Type
 120  K4086                 157   K&WG_MP10_Active_Axis
                            158
 121  K-1                   159   K-1
 122 ENDK                   160 ENDK
                            161
 123 KFIELD AXISNUMBER      162 KFIELD AxisNumber
 124  K0 ;AXIS 1            163   K+0                               ;Axis 1
 125  K1 ;AXIS 2            164   K+1                               ;Axis 2
 126  K2 ;ACIS 3            165   K+2                               ;Acis 3
 127  K3 ;AXIS 4            166   K+3                               ;Axis 4
 128  K4 ;AXIS 5            167   K+4                               ;Axis 5
 129  K15 ;AXIS S           168   K+15                              ;Axis S
 130 ENDK                   169 ENDK
________________________________ Local Symbols _______________________________

KL_BYTE_TYPE                                           :  142     K0

________________________________ Local Symbols _______________________________

KL_DWORD_TYPE                                          :  144     K2
KL_INDEX_0                                             :  146     K0
       KF:118
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:18
KL_ON_POWER_FAIL                                       :   13     K1
       PS:16
KL_WORD_TYPE                                           :  143     K1
       KF:119
________________________________Static Symbols _______________________________

BL_CASE                                                :    9     B4066
        =:55  CASE:59
BL_MPS_READ_IDENTIFY                                   :    8     B4067
      RPLY:36     =:40
WL_INDEX_REG                                           :   10     W4080
        =:46     L:64     =:66
```

```
________________________________ Local Labels ________________________________

9032                                                     : 9032     LBL 9032
     71
9038                                                     : 9038     LBL 9038
     96   99
9167                                                     : 9167     LBL 9167
     36
AXISNUMBER                                               :  162     LBL 14
     94   97
MPS_READ_SUBMIT                                          :   63     LBL 9
     57
MP_READ_TABLE                                            :  153     LBL 10
     68   70   73   76   87


________________________________ Local Labels ________________________________

PL_BYTE_INDEX                                            :  124     LBL 11
     79
PL_DWORD_INDEX                                           :  135     LBL 13
     81
PL_WORD_INDEX                                            :  128     LBL 12
     80
                            1
                            2 *+----------------------------------------------------+
                            3 *¦  M functions                                       ¦
                            4 *+----------------------------------------------------+
                            5
   1 GLOBAL M_FUNCTION      6 GLOBAL M_Function
                            7
   2 LBL M_FUNCTION         8 LBL M_Function
   3 L W260                 9   L      NP_W260_M_code
   4 < K30                 10   <      K+30
   5 IFT                   11   IFT
   6 L W260                12     L      NP_W260_M_code
   7 =X                    13     =X
   8 L KF M_FUNK_TAB[X]    14     L      KF M_Funk_Tab[X]
   9 =X                    15     =X
  10 L M4072               16     L      NP_M4072_strobe_M_function
  11 = M0[X]               17     =      M0[X]
  12 ENDI                  18   ENDI
  13 EM                    19 EM
                           20
  14 KFIELD M_FUNK_TAB     21 KFIELD M_Funk_Tab
  15  K3995 ; 0            22   K &MG_spindle_off_M05          ; 0
  16  K4805 ; 1            23   K &PN_error_not_used_M_function; 1
  17  K3995 ; 2            24   K &MG_spindle_off_M05          ; 2
  18  K3997 ; 3            25   K &MG_spindle_on_M03           ; 3
  19  K3996 ; 4            26   K &MG_spindle_on_M04           ; 4
  20  K3995 ; 5            27   K &MG_spindle_off_M05          ; 5
  21  K4805 ; 6            28   K &PN_error_not_used_M_function; 6
  22  K4805 ; 7            29   K &PN_error_not_used_M_function; 7
  23  K4805 ; 8            30   K &PN_error_not_used_M_function; 8
  24  K4805 ; 9            31   K &PN_error_not_used_M_function; 9
  25  K4805 ;10            32   K &PN_error_not_used_M_function;10
  26  K4805 ; 1            33   K &PN_error_not_used_M_function; 1
  27  K4805 ; 2            34   K &PN_error_not_used_M_function; 2
  28  K3997 ; 3            35   K &MG_spindle_on_M03           ; 3
  29  K3996 ; 4            36   K &MG_spindle_on_M04           ; 4
  30  K4805 ; 5            37   K &PN_error_not_used_M_function; 5
  31  K4805 ; 6            38   K &PN_error_not_used_M_function; 6
  32  K4805 ; 7            39   K &PN_error_not_used_M_function; 7
  33  K4805 ; 8            40   K &PN_error_not_used_M_function; 8
  34  K3994 ; 9            41   K &MG_spi_Pos_M19_R_M0X        ; 9
  35  K4805 ;20            42   K &PN_error_not_used_M_function;20
  36  K4805 ; 1            43   K &PN_error_not_used_M_function; 1
  37  K4805 ; 2            44   K &PN_error_not_used_M_function; 2
  38  K4805 ; 3            45   K &PN_error_not_used_M_function; 3
  39  K4805 ; 4            46   K &PN_error_not_used_M_function; 4
  40  K4805 ; 5            47   K &PN_error_not_used_M_function; 5
  41  K4805 ; 6            48   K &PN_error_not_used_M_function; 6
  42  K4805 ; 7            49   K &PN_error_not_used_M_function; 7
  43  K4805 ; 8            50   K &PN_error_not_used_M_function; 8
  44  K4805 ; 9            51   K &PN_error_not_used_M_function; 9
  45  K3995 ;30            52   K &MG_spindle_off_M05          ;30
  46 ENDK                  53 ENDK
________________________________ Local Labels ________________________________

M_FUNK_TAB                                               :   21     LBL 15
     14
```

```
                               1
                               2 *+----------------------------------------------------+
                               3 *¦  Control reference end switches                    ¦
                               4 *+----------------------------------------------------+
                               5
   1 GLOBAL REFERENCE_END      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
   2 LBL REFERENCE_ENDSWI     15 LBL reference_endswitch
   3 L I0                     16   L      I_Ref_Endswitch_1_axis
   4 = M3200                  17   =      Input_Bit0
   5 L I1                     18   L      I_Ref_Endswitch_2_axis
   6 = M3201                  19   =      Input_Bit1
   7 L I2                     20   L      I_Ref_Endswitch_3_axis
   8 = M3202                  21   =      Input_Bit2
   9 L I4                     22   L      I_Ref_Endswitch_4_axis
  10 = M3203                  23   =      Input_Bit3
  11 L I6                     24   L      I_Ref_Endswitch_5_axis
  12 = M3204                  25   =      Input_Bit4
                              26
  13 LB M3200                 27   LB     Inputs
  14 = W1054                  28   =      PN_W1054_reference_endswitch
  15 EM                       29 EM
                              30
________________________________ Local Symbols _______________________________

INPUTS                                                 :    8     M3200
      LB:13
INPUT_BIT0                                             :    9     M3200
        =:4
INPUT_BIT1                                             :   10     M3201
        =:6
INPUT_BIT2                                             :   11     M3202
        =:8
INPUT_BIT3                                             :   12     M3203
        =:10
INPUT_BIT4                                             :   13     M3204
        =:12
                               1
                               2 *+----------------------------------------------------+
                               3 *¦  Direction keys                                    ¦
                               4 *¦  Jog mode                                          ¦
                               5 *¦  NC start                                          ¦
                               6 *¦  NC stop                                           ¦
                               7 *¦  Rapid traverse key                                ¦
                               8 *+----------------------------------------------------+
                               9
   1 GLOBAL MANUEL_BUTTON     10 GLOBAL Manuel_button_funcktion
                              11
     M3200                    12 #define ML_XYZ45_Plus              M[8]
     M3200                    13 #define /c ML_1_Plus                  M &ML_XYZ45_Plus + 0
     M3201                    14 #define /c ML_2_Plus                  M &ML_XYZ45_Plus + 1
     M3202                    15 #define /c ML_3_Plus                  M &ML_XYZ45_Plus + 2
     M3203                    16 #define /c ML_4_Plus                  M &ML_XYZ45_Plus + 3
     M3204                    17 #define /c ML_5_Plus                  M &ML_XYZ45_Plus + 4
                              18
     M3208                    19 #define ML_XYZ45_Minus             M[8]
     M3208                    20 #define /c ML_1_Minus                 M &ML_XYZ45_Minus + 0
     M3209                    21 #define /c ML_2_Minus                 M &ML_XYZ45_Minus + 1
     M3210                    22 #define /c ML_3_Minus                 M &ML_XYZ45_Minus + 2
     M3211                    23 #define /c ML_4_Minus                 M &ML_XYZ45_Minus + 3
     M3212                    24 #define /c ML_5_Minus                 M &ML_XYZ45_Minus + 4
                              25
   2 LBL MANUEL_BUTTON_FU     26 LBL Manuel_button_funcktion
   3 L I131                   27   L      I_NC_Stop
   4 AN M3993                 28   AN     MG_T_I_N_supervision
   5 = M4560                  29   =      PN_M4560_NC_STOP_0_activ
                              30
   6 L I141                   31   L      I_rapid_button
   7 = M4561                  32   =      PN_M4561_rapide
                              33
   8 L I5                     34   L      I_feed_enable
   9 = M4563                  35   =      PN_M4563_feed_enable
                              36
  10 L I132                   37   L      I_NC_Start
  11 = M4564                  38   =      PN_M4564_NC_start
  12 = M4562                  39   =      PN_M4562_axis_button_latch
                              40
  13 L M4150                41   L      NP_M4150_manuel_mode
  14 O M4151                42   O      NP_M4151_electronic_handwhell
  15 O M4155                43   O      NP_M4155_reference_mode
  16 O M4156                44   O      NP_M4156_Softkey_Manual_Operation
  17 CMT INPUT_KEYBOARD     45   CMT    Input_keyboard
  18 CMT JOG_DIRECTION_BU   46   CMT    Jog_Direction_Button
  19 EM                     47 EM
                            48
  20 LBL INPUT_KEYBOARD     49 LBL Input_keyboard
  21 L I138                 50   L      I_1_axis_Plus
  22 = M3200                51   =      ML_1_Plus
  23 L I137                 52   L      I_2_axis_Plus
  24 = M3201                53   =      ML_2_Plus
  25 L I136                 54   L      I_3_axis_Plus
  26 = M3202                55   =      ML_3_Plus
  27 L I139                 56   L      I_4_axis_Plus
  28 = M3203                57   =      ML_4_Plus
  29 L I146                 58   L      I_5_axis_Plus
  30 = M3204                59   =      ML_5_Plus
                            60
  31 L I133                 61   L      I_1_axis_Minus
  32 = M3208                62   =      ML_1_Minus
  33 L I134                 63   L      I_2_axis_Minus
  34 = M3209                64   =      ML_2_Minus
  35 L I135                 65   L      I_3_axis_Minus
  36 = M3210                66   =      ML_3_Minus
  37 L I140                 67   L      I_4_axis_Minus
  38 = M3211                68   =      ML_4_Minus
  39 L I147                 69   L      I_5_axis_Minus
  40 = M3212                70   =      ML_5_Minus
  41 EM                     71 EM
                            72
  42 LBL JOG_DIRECTION_BU   73 LBL Jog_Direction_Button
  43 L M4572                74   L      NP_M4572_enable_jog_mode_Posit
  44 A M4151                75   A      NP_M4151_electronic_handwhell
  45 IFT                    76   IFT
  46 LB M3200               77     LB     ML_XYZ45_Plus
  47 = W1050                78     =      PN_W1050_jog_mode_Posit_plus
  48 LB M3208               79     LB     ML_XYZ45_Minus
  49 = W1052                80     =      PN_W1052_jog_mode_Posit_minus
  50 ELSE                   81   ELSE
  51 LB M3200               82     LB     ML_XYZ45_Plus
  52 = W1046                83     =      PN_W1046_manuel_dircetion_plus
  53 LB M3208               84     LB     ML_XYZ45_Minus
  54 = W1048                85     =      PN_W1048_manuel_direction_minus
  55 ENDI                   86   ENDI
  56 EM                     87 EM
________________________________ Local Symbols _______________________________

ML_1_MINUS                                              :   20     M3208
        =:32
ML_1_PLUS                                               :   13     M3200
        =:22
ML_2_MINUS                                              :   21     M3209
        =:34
ML_2_PLUS                                               :   14     M3201
        =:24
ML_3_MINUS                                              :   22     M3210
        =:36
ML_3_PLUS                                               :   15     M3202
        =:26
ML_4_MINUS                                              :   23     M3211
        =:38
ML_4_PLUS                                               :   16     M3203
        =:28
ML_5_MINUS                                              :   24     M3212
        =:40
ML_5_PLUS                                               :   17     M3204
        =:30
ML_XYZ45_MINUS                                          :   19     M3208
       LB:48    LB:53
ML_XYZ45_PLUS                                           :   12     M3200
       LB:46    LB:51

________________________________ Local Labels ________________________________

INPUT_KEYBOARD                                          :   49     LBL 16
     45
JOG_DIRECTION_BU                                        :   73     LBL 17
     46
```

```
                             1
                             2 * +-----------------------------------------------------+
                             3 * ¦ Axis control 5,4,Z,Y,X,                             ¦
                             4 * +-----------------------------------------------------+
                             5
   1 GLOBAL NC_AXIS          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
     M3977                  14 #define /s  ML_servo_enable_axis        M[5]
     M3977                  15 #define /c  ML_1_servo_enable_axis      M &ML_servo_enable_axis
+ 0
     M3978                  16 #define /c  ML_2_servo_enable_axis      M &ML_servo_enable_axis
+ 1
     M3979                  17 #define /c  ML_3_servo_enable_axis      M &ML_servo_enable_axis
+ 2
     M3980                  18 #define /c  ML_4_servo_enable_axis      M &ML_servo_enable_axis
+ 3
     M3981                  19 #define /c  ML_5_servo_enable_axis      M &ML_servo_enable_axis
+ 4
                            20
     M3972                  21 #define /s  ML_clamping_Achsen          M[5]
     M3972                  22 #define /c  ML_clamping_1_axis          M &ML_clamping_Achsen +
0
     M3973                  23 #define /c  ML_clamping_2_axis          M &ML_clamping_Achsen +
1
     M3974                  24 #define /c  ML_clamping_3_axis          M &ML_clamping_Achsen +
2
     M3975                  25 #define /c  ML_clamping_4_axis          M &ML_clamping_Achsen +
3
     M3976                  26 #define /c  ML_clamping_5_axis          M &ML_clamping_Achsen +
4
                            27
   2 LBL NC_AXIS            28 LBL NC_Axis
   3 L K1                   29   L      K+1
   4 = W2048                30   =      WL_Axis_Mask
   5 L K0                   31   L      K+0
   6 =X                     32   =X
   7 REPEAT                 33   REPEAT
   8 LN I3                  34     LN     I_not_emergency_stop
   9 ON I129                35     ON     I_servo_ready_1
  10 ON I144                36     ON     I_servo_ready_2
  11 IFT                    37     IFT
  12 L K0                   38       L      K+0
  13 = B4052[X]             39       =      BL_Axis_Step[X]
  14 ENDI                   40     ENDI
  15 CASE B4052[X]          41     CASE   BL_Axis_Step[X]
  16 CM INITIAL_AXIS        42       CM     Initial_Axis
  17 CM WAITING_POS_STAR    43       CM     Waiting_Pos_Start
  18 CM ON_CURRENTRPML_C    44       CM     On_currentRPMl_control
  19 CM CLAMPING_OPEN       45       CM     Clamping_open
  20 CM CLOSE_LOOP_CLOSE    46       CM     close_loop_close
  21 CM POSITIONING         47       CM     positioning
  22 CM CLOSE_LOOP_OPEN     48       CM     close_loop_open
  23 CM CLAMPING_CLOSE      49       CM     clamping_close
  24 CM OFF_CURRENTRPML_    50       CM     off_currentRPMl_control
  25 CM STEP_CHAIN_END      51       CM     Step_chain_end
  26 ENDC                   52     ENDC
  27 L W2048                53     L      WL_Axis_Mask
  28 << K1                  54     <<     K+1
  29 = W2048                55     =      WL_Axis_Mask
  30 INCX                   56     INCX
  31 LX                     57     LX
  32 >= K5                  58     >=     Max_NC_Axis
  33 UNTILT                 59   UNTILT
                            60
  34 CM WRITE_OUTPUTS       61   CM     Write_Outputs
                            62
                            63 #ifdef NC_Type_PA
  35 L O15                  64   L      O_Spindle_servo_enable
  36 IFT                    65   IFT
  37 L W4062                66     L      WL_current_rpm_control
  38 BS K15                 67     BS     K+15
  39 = W4062                68     =      WL_current_rpm_control
  40 ELSE                   69   ELSE
  41 L W4062                70     L      WL_current_rpm_control
  42 BC K15                 71     BC     K+15
  43 = W4062                72     =      WL_current_rpm_control
  44 ENDI                   73   ENDI
                            74
  45 L W4062                75   L      WL_current_rpm_control
  46 <> W4060               76   <>     WL_old_current_rpm_control
```

```
IFT                         IFT
L W4062                         L      WL_current_rpm_control
= W4060                         =      WL_old_current_rpm_control
A W4082                         A      WG_Active_PWM_Axis
= W4084                         =      WG_servo_enable_internal_servo
PSW                             PSW
CM 9161                         CM     9161
L M4203                         L      NP_M4203_error_Modul_9xxx
S M4803                         S      PN_error_mod_9161
ENDI                          ENDI
                          #endif
EM                        EM

LBL INITIAL_AXIS          LBL Initial_Axis
L W1038                     L      PN_W1038_closed_loop_open_active
O W2048                     O      WL_Axis_Mask
= W1038                     =      PN_W1038_closed_loop_open_active

L W1040                     L      PN_W1040_closed_loop_open
O W2048                     O      WL_Axis_Mask
= W1040                     =      PN_W1040_closed_loop_open

L W1042                     L      PN_W1042_supervision_inactiv
O W2048                     O      WL_Axis_Mask
= W1042                     =      PN_W1042_supervision_inactiv

L W1044                     L      PN_W1044_actul_nominal_transfer
O W2048                     O      WL_Axis_Mask
= W1044                     =      PN_W1044_actul_nominal_transfer

L M3999                     L      MG_one_marker
R M3977[X]                  R      ML_servo_enable_axis[X]
R M3972[X]                  R      ML_clamping_Achsen[X]

L W4062                     L      WL_current_rpm_control
AN W2048                    AN     WL_Axis_Mask
= W4062                     =      WL_current_rpm_control

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 M3977[X]                 LN     ML_servo_enable_axis[X]
S M3977[X]                  S      ML_servo_enable_axis[X]

INC B4052[X]                INC    BL_Axis_Step[X]
EM                        EM

LBL CLAMPING_OPEN         LBL Clamping_open
LN M3972[X]                 LN     ML_clamping_Achsen[X]
S M3972[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 I129                   ON     I_servo_ready_1
ON I144                   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 M3972[X]              L      ML_clamping_Achsen[X]
R M3972[X]              R      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_OPEN   LBL close_loop_open
L W1040                 L      PN_W1040_closed_loop_open
O W2048                 O      WL_Axis_Mask
= W1040                 =      PN_W1040_closed_loop_open

L W1038                 L      PN_W1038_closed_loop_open_active
O W2048                 O      WL_Axis_Mask
= W1038                 =      PN_W1038_closed_loop_open_active

INC B4052[X]            INC    BL_Axis_Step[X]
EM                    EM

LBL OFF_CURRENTRPML_  LBL off_currentRPMl_control
L W4062                 L      WL_current_rpm_control
AN W2048                AN     WL_Axis_Mask
= W4062                 =      WL_current_rpm_control

L M3977[X]              L      ML_servo_enable_axis[X]
R M3977[X]              R      ML_servo_enable_axis[X]

INC B4052[X]            INC    BL_Axis_Step[X]
EM                    EM

LBL STEP_CHAIN_END    LBL Step_chain_end
L K1                    L      K+1
= B4052[X]              =      BL_Axis_Step[X]
EM                    EM

LBL WRITE_OUTPUTS     LBL Write_Outputs
L M3977                 L      ML_1_servo_enable_axis
= O0                    =      O_1_axis_enable
L M3978                 L      ML_2_servo_enable_axis
= O1                    =      O_2_axis_enable
L M3979                 L      ML_3_servo_enable_axis
= O2                    =      O_3_axis_enable
L M3980                 L      ML_4_servo_enable_axis
= O3                    =      O_4_axis_enable
L M3981                 L      ML_5_servo_enable_axis
= O4                    =      O_5_axis_enable
```

```
  188 L M3972               249   L      ML_clamping_1_axis
 189 = O7                  250   =      O_1_clamping
 190 L M3973               251   L      ML_clamping_2_axis
 191 = O8                  252   =      O_2_clamping
 192 L M3974               253   L      ML_clamping_3_axis
 193 = O9                  254   =      O_3_clamping
 194 L M3975               255   L      ML_clamping_4_axis
 195 = O10                 256   =      O_4_clamping
 196 L M3976               257   L      ML_clamping_5_axis
 197 = O11                 258   =      O_5_clamping
 198 EM                    259 EM

________________________________ Local Symbols _______________________________

ML_1_SERVO_ENABLE_AXIS                                  :   15     M3977
        L:178
ML_2_SERVO_ENABLE_AXIS                                  :   16     M3978
        L:180
ML_3_SERVO_ENABLE_AXIS                                  :   17     M3979
        L:182
ML_4_SERVO_ENABLE_AXIS                                  :   18     M3980
        L:184
ML_5_SERVO_ENABLE_AXIS                                  :   19     M3981
        L:186
ML_CLAMPING_1_AXIS                                      :   22     M3972
        L:188
ML_CLAMPING_2_AXIS                                      :   23     M3973
        L:190
ML_CLAMPING_3_AXIS                                      :   24     M3974
        L:192
ML_CLAMPING_4_AXIS                                      :   25     M3975
        L:194
ML_CLAMPING_5_AXIS                                      :   26     M3976
        L:196
WL_AXIS_MASK                                            :   12     W2048
        =:4      L:27     =:29     O:60     O:63     O:66     O:69
       AN:75     A:79     A:88     O:96    AN:116   AN:120   AN:123
       AN:126    A:133    O:158    O:161   AN:167
________________________________Static Symbols _______________________________

BL_AXIS_STEP                                            :   11     B4052
        =:13  CASE:15  INCW:82  INCW:91  INCW:100 INCW:109 INCW:128

________________________________Static Symbols _______________________________

     INCW:143 INCW:153 INCW:163 INCW:171    =:175
ML_CLAMPING_ACHSEN                                      :   21     M3972
        R:73    LN:103    S:104    L:147    R:148
ML_SERVO_ENABLE_AXIS                                    :   14     M3977
        R:72    LN:98     S:99     L:169    R:170
WL_CURRENT_RPM_CONTROL                                  :    8     W4062
        L:37     =:39     L:41     =:43     L:45     L:48     L:74
        =:76     L:95     =:97     L:166    =:168
WL_OLD_CURRENT_RPM_CONTR                                :    9     W4060
       <>:46     =:49
________________________________ Local Labels ________________________________

9161                                                    : 9161     LBL 9161
     83
CLAMPING_CLOSE                                          :  197     LBL 25
     49
CLAMPING_OPEN                                           :  145     LBL 21
     45
CLOSE_LOOP_CLOSE                                        :  157     LBL 22
     46
CLOSE_LOOP_OPEN                                         :  209     LBL 24
     48
INITIAL_AXIS                                            :   90     LBL 18
     42
OFF_CURRENTRPML_                                        :  221     LBL 26
     50
ON_CURRENTRPML_C                                        :  134     LBL 20
     44
POSITIONING                                             :  180     LBL 23
     47
STEP_CHAIN_END                                          :  232     LBL 27
     51
```

```
________________________________ Local Labels ________________________________

WAITING_POS_STAR                                          :  124     LBL 19
     43
WRITE_OUTPUTS                                             :  237     LBL 28
     61


                              1
                              2 *+---------------------------------------------------+
                              3 *¦  Spindle function                                 ¦
                              4 *+---------------------------------------------------+
                              5
   1 GLOBAL SPINDLE_FUNCT     6 GLOBAL spindle_function
                              7
     M3971                    8 #define /s ML_spi_pos_start           M
     M3970                    9 #define /s ML_servo_activ_poweron     M
                             10
   2 LBL SPINDLE_FUNCTION    11 LBL spindle_function
   3 L M3994                 12   L       MG_spi_Pos_M19_R_M0X
   4 AN M3971                13   AN      ML_spi_pos_start
   5 CMT M19_START_SPI_PO    14   CMT     M19_start_spi_pos
   6 S M3971                 15   S       ML_spi_pos_start
                             16
   7 LN M3994                17   LN      MG_spi_Pos_M19_R_M0X
   8 A M4000                 18   A       NP_M4000_S_in_position
   9 R M3971                 19   R       ML_spi_pos_start
                             20
  10 L M3997                 21   L       MG_spindle_on_M03
  11 S M4005                 22   S       PN_M4005_S_M03_analog_volt_status
  12 R M4006                 23   R       PN_M4006_S_M04_analog_volt_status
                             24
  13 L M3996                 25   L       MG_spindle_on_M04
  14 R M4005                 26   R       PN_M4005_S_M03_analog_volt_status
  15 S M4006                 27   S       PN_M4006_S_M04_analog_volt_status
                             28
  16 L M3994                 29   L       MG_spi_Pos_M19_R_M0X
  17 O M3995                 30   O       MG_spindle_off_M05
  18 O M3993                 31   O       MG_T_I_N_supervision
  19 ON I3                   32   ON      I_not_emergency_stop
  20 R M4005                 33   R       PN_M4005_S_M03_analog_volt_status
  21 R M4006                 34   R       PN_M4006_S_M04_analog_volt_status
                             35
  22 LN M4005                36   LN      PN_M4005_S_M03_analog_volt_status
  23 AN M4006                37   AN      PN_M4006_S_M04_analog_volt_status
  24 = M4007                 38   =       PN_M4007_S_M05_0V_status
                             39
  25 L M4012                 40   L       PN_M4012_S_close_loop_open
  26 R M4012                 41   R       PN_M4012_S_close_loop_open
                             42
  27 L M3995                 43   L       MG_spindle_off_M05
  28 O M3993                 44   O       MG_T_I_N_supervision
  29 O M4005                 45   O       PN_M4005_S_M03_analog_volt_status
  30 O M4006                 46   O       PN_M4006_S_M04_analog_volt_status
  31 S M4012                 47   S       PN_M4012_S_close_loop_open
                             48
  32 L M4002                 49   L       NP_M4002_S_analog_0_V
  33 S M3970                 50   S       ML_servo_activ_poweron
  34 L M4005                 51   L       PN_M4005_S_M03_analog_volt_status
  35 O M4006                 52   O       PN_M4006_S_M04_analog_volt_status
  36 ON M4002                53   ON      NP_M4002_S_analog_0_V
  37 O M4130                 54   O       PN_M4130_Strobe_PLC_pos_spindle
  38 A M3970                 55   A       ML_servo_activ_poweron
  39 S O15                   56   S       O_Spindle_servo_enable
                             57
  40 L M3995                 58   L       MG_spindle_off_M05
  41 O M3993                 59   O       MG_T_I_N_supervision
  42 ON I3                   60   ON      I_not_emergency_stop
  43 ON I129                 61   ON      I_servo_ready_1
  44 ON I144                 62   ON      I_servo_ready_2
  45 R O15                   63   R       O_Spindle_servo_enable
                             64
  46 LN M4072                65   LN      NP_M4072_strobe_M_function
  47 = T7                    66   =       TS_M_func_delay
                             67
  48 L M4072                 68   L       NP_M4072_strobe_M_function
  49 A M4001                 69   A       NP_M4001_S_analog_not_in_ramp
  50 AN M4130                70   AN      PN_M4130_Strobe_PLC_pos_spindle
  51 AN T7                   71   AN      TS_M_func_delay
  52 AN T55                  72   AN      TR_M_func_delay
  53 AN M4805                73   AN      PN_error_not_used_M_function
  54 = M4092                 74   =       PN_M4092_quit_M_function
  55 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
    56 LBL M19_START_SPI_PO   81 LBL M19_start_spi_pos
  57 PS K0                  82   PS     KL_angle_spindle_pos
  58 PS K100000             83   PS     KL_RPM_spindle_pos
  59 PS K0                  84   PS     KL_direction_spindle_pos
  60 CM 9171                85   CM     9171
  61 L M4203                86   L      NP_M4203_error_Modul_9xxx
  62 S M4806                87   S      PN_error_9171_Spi_Pos
  63 EM                     88 EM

________________________________ Local Symbols _______________________________

KL_ANGLE_SPINDLE_POS                                    :   77     K0
       PS:57
KL_DIRECTION_SPINDLE_POS                                :   79     K0
       PS:59
KL_RPM_SPINDLE_POS                                      :   78     K100000
       PS:58
________________________________Static Symbols _______________________________

ML_SERVO_ACTIV_POWERON                                  :    9     M3970
        S:33     A:38
ML_SPI_POS_START                                        :    8     M3971
       AN:4      S:6      R:9
________________________________ Local Labels ________________________________

9171                                                    : 9171     LBL 9171
     85
M19_START_SPI_PO                                        :   81     LBL 29
     14
                             1 *+---------------------------------------------------+
                             2 *| Gear changing                                     |
                             3 *+---------------------------------------------------+
                             4
     B4065                   5 #define /s    BL_step_gear                      B
     B127                    6 #define /s /r BL_G_code                         B
                             7
     D120                    8 #define /s /r DL_N_programmed                   D
                             9
     M3200                  10 #define       ML_Gear_swicth_done               M
                            11
   1 GLOBAL GEAR_CHANGING   12 GLOBAL Gear_Changing
                            13
   2 LBL GEAR_CHANGING      14 LBL Gear_Changing
   3 L M4172                15   L      NP_M4172_1_PLC_after_power_on
   4 O M4173                16   O      NP_M4172_1_PLC_after_compile
   5 S M4134                17   S      PN_M4134_strobe_G_step_rpm
   6 IFT                    18   IFT
   7 L B127                 19     L      BL_G_code
   8 = W256                 20     =      NP_W256_G_code_spindle
   9 L D120                 21     L      DL_N_programmed
  10 = D756                 22     =      PN_D756_S_nominal_rpm_PLC
  11 ENDI                   23   ENDI
                            24
  12 LN M4070               25   LN     NP_M4070_strobe_G_code
  13 R M4090                26   R      PN_M4090_quit_G_code
                            27
  14 L B4065                28   L      BL_step_gear
  15 == K0                  29   ==     K+0
  16 = T20                  30   =      TS_gear_timeout
                            31
  17 CASE B4065             32   CASE   BL_step_gear
  18 CM ACTIVATION ;00      33     CM     Activation                 ;00
  19 CM SPINDLE_ZERO ;01    34     CM     spindle_zero               ;01
  20 CM GEAR_RANGE_SWITC    35     CM     gear_range_switch          ;02
  21 CM QUIT ;03            36     CM     quit                       ;03
  22 CM END ;04             37     CM     end                        ;04
  23 ENDC                   38   ENDC
                            39
  24 PLL                    40   PLL
  25 IFT                    41   IFT
  26 INC B4065              42     INC    BL_step_gear
  27 ENDI                   43   ENDI
                            44
  28 LN M991                45   LN     I_gear_range_1
  29 XO M990                46   XO     I_gear_range_2
  30 AN M4070               47   AN     NP_M4070_strobe_G_code
  31 AN M4134               48   AN     PN_M4134_strobe_G_step_rpm
  32 AN M3982               49   AN     MG_gear_change_activ
  33 O[                     50   O[
  34 AN T20                 51     AN     TS_gear_timeout
  35 AN T68                 52     AN     TR_gear_timout
  36 ]                      53   ]
  37 = M4815                54   =      PN_Error_gear_switching
                            55
```

```
L M4815                     L      PN_Error_gear_switching
AN M4070                    AN     NP_M4070_strobe_G_code
AN M3982                    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 M3982                     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

LBL GEAR_RANGE_1 ;+0      LBL gear_range_1                      ;+00
LN M991                     LN     I_gear_range_1
O M990                      O      I_gear_range_2
S M991                      S      O_Gear_Range_1
R M990                      R      O_Gear_Range_2
=N M3200                    =N     ML_Gear_swicth_done

L M3200                     L      ML_Gear_swicth_done
PSL                         PSL
EM                        EM

LBL GEAR_RANGFE_2 ;+      LBL gear_rangfe_2                     ;+01
L M991                      L      I_gear_range_1
ON M990                     ON     I_gear_range_2
R M991                      R      O_Gear_Range_1
S M990                      S      O_Gear_Range_2
=N M3200                    =N     ML_Gear_swicth_done

L M3200                     L      ML_Gear_swicth_done
PSL                         PSL
EM                        EM

LBL QUIT ;03              LBL quit                              ;03
L M4070                     L      NP_M4070_strobe_G_code
S M4090                     S      PN_M4090_quit_G_code

LN M4070                    LN     NP_M4070_strobe_G_code
S M3983                     S      MG_spindle_on_gear
PSL                         PSL
EMF                         EMF

L W256                      L      NP_W256_G_code_spindle
= B127                      =      BL_G_code

L D756                      L      PN_D756_S_nominal_rpm_PLC
= D120                      =      DL_N_programmed
EM                        EM

LBL END ;04               LBL end                               ;04
PS M3998                    PS     MG_zero_marker
LBL RESET                 LBL reset
L M3999                     L      MG_one_marker
R M3982                     R      MG_gear_change_activ
R M3983                     R      MG_spindle_on_gear
R M4010                     R      PN_M4010_S_swing_right
R M4009                     R      PN_M4009_S_swing_left

L K0                        L      K+0
= B4065                     =      BL_step_gear
EM                        EM
```

```
________________________________ Local Symbols ________________________________

ML_GEAR_SWICTH_DONE                                                 :   10     M3200
     WHIL:72     L:73  WHIL:81     L:82
________________________________Static Symbols ________________________________

BL_G_CODE                                                           :    6     B127
        L:7      =:93
BL_STEP_GEAR                                                        :    5     B4065
        L:14  CASE:17  INCW:26     =:106
DL_N_PROGRAMMED                                                     :    8     D120
        L:9      =:95
________________________________ Local Labels ________________________________

ACTIVATION                                                          :   63     LBL 30
     33
END                                                                 :  130     LBL 34
     37
GEAR_RANGE_1                                                        :   92     LBL 36
     87
GEAR_RANGE_SWITC                                                    :   77     LBL 32
     35
GEAR_RANGFE_2                                                       :  103     LBL 37
     88
QUIT                                                                :  114     LBL 33
     36
RESET                                                               :  132     LBL 35
     60
SPINDLE_ZERO                                                        :   70     LBL 31
     34

                              1
   1 GLOBAL AXIS_SUPERVIS     2 GLOBAL Axis_Supervision
                              3
                              4 *+---------------------------------------------------+
                              5 *¦ Supervison of the drives by the TNC 430           ¦
                              6 *+---------------------------------------------------+
                              7
   2 LBL AXIS_SUPERVISION     8 LBL Axis_Supervision
   3 CM 9160                  9   CM     9160
   4 L M4203                 10   L      NP_M4203_error_Modul_9xxx
   5 S M4810                 11   S      PN_error_modul_9xxx_Supervision
   6 PLW                     12   PLW
   7 <> K0                   13   <>     K+0
   8 S M4808                 14   S      PN_error_Temp_powersupply
   9 PLW                     15   PLW
  10 PSW                     16   PSW
  11 A K$FFFF                17   A      K$0000FFFF
  12 <> K0                   18   <>     K+0
  13 S M4809                 19   S      PN_error_I2T_caution
  14 PLW                     20   PLW
  15 A K$FFFF0000            21   A      K$FFFF0000
  16 <> K0                   22   <>     K+0
  17 S M4813                 23   S      PN_error_I2T_limitation
                             24
  18 CM 9162                 25   CM     9162
  19 L M4203                 26   L      NP_M4203_error_Modul_9xxx
  20 S M4810                 27   S      PN_error_modul_9xxx_Supervision
  21 LN W4084                28   LN     WG_servo_enable_internal_servo
  22 B= T8                   29   B=     TS_1_servo_supervison
  23 L W4084                 30   L      WG_servo_enable_internal_servo
  24 AN[                     31   AN[
  25 PLW                     32     PLW
  26 ]                       33   ]
  27 AN[                     34   AN[
  28 LB T56                  35     LB     TR_1_servo_supervison
  29 ]                       36   ]
  30 <> K0                   37   <>     K+0
  31 S M4807                 38   S      PN_error_servo_activ
                             39
  32 L K0                    40   L      K+0
  33 =X                      41   =X
  34 REPEAT                  42   REPEAT
  35 L M3984                 43     L      MG_active_PWM_axis
  36 IFT                     44     IFT
  37 PS KF AXIS_BIT_CODE[    45       PS     KF Axis_Bit_Code[X]
  38 CM 9165                 46       CM     9165
  39 PLW                     47       PLW
  40 >= W4068[X]             48       >=     WG_motor_temp[X]
  41 S M4812                 49       S      PN_error_motor_temp
                             50
  42 L M4203                 51       L      NP_M4203_error_Modul_9xxx
  43 S M4810                 52       S      PN_error_modul_9xxx_Supervision
                             53
  44 PS KF AXIS_BIT_CODE[    54       PS     KF Axis_Bit_Code[X]
  45 CM 9166                55       CM     9166
  46 PLW                    56       PLW
  47 > K100                 57       >      K+100
  48 S M4811                58       S      PN_error_utilization_motor
                            59
  49 L M4203                60       L      NP_M4203_error_Modul_9xxx
  50 S M4810                61       S      PN_error_modul_9xxx_Supervision
  51 ENDI                   62     ENDI
  52 INCX                   63     INCX
  53 LX                     64     LX
  54 > K5                   65     >      K+5
  55 UNTILT                 66   UNTILT
                            67
  56 L M4808                68   L      PN_error_Temp_powersupply
  57 O M4809                69   O      PN_error_I2T_caution
  58 O M4812                70   O      PN_error_motor_temp
  59 O M4811                71   O      PN_error_utilization_motor
  60 O M4807                72   O      PN_error_servo_activ
  61 = M3993                73   =      MG_T_I_N_supervision
  62 EM                     74 EM
                            75
  63 KFIELD AXIS_BIT_CODE   76 KFIELD Axis_Bit_Code
  64  K0                    77   K+00
  65  K1                    78   K+01
  66  K2                    79   K+02
  67  K3                    80   K+03
  68  K4                    81   K+04
  69  K15                   82   K+15
  70 ENDK                   83 ENDK
________________________________ Local Labels ________________________________

9160                                                    : 9160     LBL 9160
     9
9162                                                    : 9162     LBL 9162
     25
9165                                                    : 9165     LBL 9165
     46
9166                                                    : 9166     LBL 9166
     55
AXIS_BIT_CODE                                           :   76     LBL 38
     45   54
                             1
                             2 *+----------------------------------------------------+
                             3 *¦  PLC soft keys                                     ¦
                             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
     M3969                  14 #define /s    ML_NC_soft_key_On                       M
                            15
                            16 #define /i    Mac_lib.Def

                             1 *+--------------------------------------------------------------
-+

                             2 *¦ Macro for library

¦

                             3 *+--------------------------------------------------------------

-¦

                             4 *¦ Macros for the 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
```

```
GLOBAL PLC_SOFT_KEYS      GLOBAL PLC_Soft_keys

LBL PLC_SOFT_KEYS         LBL PLC_Soft_keys
L W302                      L      NP_W302_Number_PLC_Soft_Key
>= K0                       >=     K+0
IFT                         IFT
CASE W302                     CASE   NP_W302_Number_PLC_Soft_Key
CM EMPTY_SK                     CM     Empty_SK
CM OFF_SK                       CM     Off_SK
CM ON_SK                        CM     On_SK
CM MASKE1_SK                    CM     Maske1_SK
CM MASKE2_SK                    CM     Maske2_SK
CM DELMASKE_SK                  CM     DelMaske_SK
ENDC                          ENDC
L K-1                         L      K-1
= W302                        =      NP_W302_Number_PLC_Soft_Key
ENDI                        ENDI

L W274                      L      NP_W274_Button_Key_Code
== K$53                     ==     KL_ASCII_Key
IFT                         IFT
L K-1 ;IMPULS                 L      K-1                         ;Impuls
= W274                        =      NP_W274_Button_Key_Code

LN M3969                      LN     ML_NC_soft_key_On
= M3969                       =      ML_NC_soft_key_On
CMT SOFT_KEY_DISPLAY          CMT    Soft_key_displayOn
CMF SOFT_KEY_DIPSLAY          CMF    Soft_key_dipslayOff
ENDI                        ENDI
EM                        EM

LBL EMPTY_SK              LBL Empty_SK
EM                        EM

LBL OFF_SK                LBL Off_SK
L M4814                     L      MG_Function_On
R M4814                     R      MG_Function_On
EM                        EM

LBL ON_SK                 LBL On_SK
LN M4814                    LN     MG_Function_On
S M4814                     S      MG_Function_On
EM                        EM

LBL SOFT_KEY_DISPLAY      LBL Soft_key_displayOn
PS KF SOFT_KEY_ROW          PS     KF Soft_key_row
PS K0                       PS     K+0
PS K1                       PS     K+1
CM 9200                     CM     9200
------ Macro ----->         M_Modul_Error_display(KG_Error_Modul_9200)

L M4203                   L      NP_M4203_error_Modul_9xxx
IFT                       IFT
PS K15                    PS     KG_ERROR_MODUL_9200
CM 9085                   CM     9085
ENDI                      ENDI
EM                        EM

LBL SOFT_KEY_DIPSLAY      LBL Soft_key_dipslayOff
PS K-1                      PS     K-1
PS K0                       PS     K+0
PS K1                       PS     K+1
CM 9200                     CM     9200
------ Macro ----->         M_Modul_Error_display(KG_Error_Modul_9200)

L M4203                   L      NP_M4203_error_Modul_9xxx
IFT                       IFT
PS K15                    PS     KG_ERROR_MODUL_9200
CM 9085                   CM     9085
ENDI                      ENDI
EM                        EM

KFIELD SOFT_KEY_ROW       KFIELD Soft_key_row
 K0                         KL_Empty
 K1                         KL_soft_key_Off
 K2                         KL_soft_key_On
 K0                         KL_Empty
 K3                         KL_Mask_1_On
 K4                         KL_Mask_2_On
 K5                         KL_Mask_Off
 K0                         KL_Empty
ENDK                      ENDK

                          #define /i    Masken.Src

LBL MASKE1_SK             LBL Maske1_SK
```

```
  71 PS K1 ; BIG PLC WIND    3   PS     K+1   ; Big PLC Window open
  72 CM 9202                 4   CM     9202
     ------ Macro ----->     5   M_Modul_Error_display(KG_Error_Modul_9202)
                             5
  73 L M4203                 5 L      NP_M4203_error_Modul_9xxx
  74 IFT                     5 IFT
  75 PS K18                  5 PS     KG_ERROR_MODUL_9202
  76 CM 9085                 5 CM     9085
  77 ENDI                    5 ENDI
                             6
  78 L S"Maske1.A"           7   L      S"Maske1.A"
  79 = S0                    8   =      S0
                             9
  80 PS K0 ;NR. STRING BU   10   PS     K+0                           ;No. string buffer /
clear>
  81 CM 9210                11   CM     9210
  82 PLW ;<STATUS/ERROR>    12   PLW                                  ;<Status/Error>
  83 >= K1 ; FEHLER         13   >=     K+1                           ; Error
     ------ Macro ----->    14   M_Error_display(KG_Error_Modul_9210)
                            14
  84 IFT                    14 IFT
  85 PS K17                 14 PS     KG_ERROR_MODUL_9210
  86 CM 9085                14 CM     9085
  87 ENDI                   14 ENDI
  88 EM                     15 EM
                            16
  89 LBL MASKE2_SK          17 LBL Maske2_SK
  90 PS K1 ; BIG PLC WIND   18   PS     K+1   ; Big PLC Window open
  91 CM 9202                19   CM     9202
     ------ Macro ----->    20   M_Modul_Error_display(KG_Error_Modul_9202)
                            20
  92 L M4203                20 L      NP_M4203_error_Modul_9xxx
  93 IFT                    20 IFT
  94 PS K18                 20 PS     KG_ERROR_MODUL_9202
  95 CM 9085                20 CM     9085
  96 ENDI                   20 ENDI
                            21
  97 L S"Maske2.A"          22   L      S"Maske2.A"
  98 = S0                   23   =      S0
                            24
  99 PS K0 ;NR. STRING BU   25   PS     K+0                           ;No. string buffer /
clear>
 100 CM 9210                26   CM     9210
 101 PLW ;<STATUS/ERROR>    27   PLW                                  ;<Status/Error>
 102 >= K1 ; FEHLER         28   >=     K+1                           ; Error
     ------ Macro ----->    29   M_Error_display(KG_Error_Modul_9210)
                            29
 103 IFT                    29 IFT
 104 PS K17                 29 PS     KG_ERROR_MODUL_9210
 105 CM 9085                29 CM     9085
 106 ENDI                   29 ENDI
 107 EM                     30 EM
                            31
 108 LBL DELMASKE_SK        32 LBL DelMaske_SK
 109 PS K0 ; BIG PLC WIND   33   PS     K+0   ; Big PLC Window clear
 110 CM 9202                34   CM     9202
     ------ Macro ----->    35   M_Modul_Error_display(KG_Error_Modul_9202)
                            35
 111 L M4203                35 L      NP_M4203_error_Modul_9xxx
 112 IFT                    35 IFT
 113 PS K18                 35 PS     KG_ERROR_MODUL_9202
 114 CM 9085                35 CM     9085
 115 ENDI                   35 ENDI
                            36
 116 PS K-1 ;NR. STRING B   37   PS     K-1                           ;No. string buffer /
clear>
 117 CM 9210                38   CM     9210
 118 PLW ;<STATUS/ERROR>    39   PLW                                  ;<Status/Error>
 119 >= K1 ; FEHLER         40   >=     K+1                           ; Error
     ------ Macro ----->    41   M_Error_display(KG_Error_Modul_9210)
                            41
 120 IFT                    41 IFT
 121 PS K17                 41 PS     KG_ERROR_MODUL_9210
 122 CM 9085                41 CM     9085
 123 ENDI                   41 ENDI
 124 EM                     42 EM

                            90
________________________________ Local Symbols _______________________________

KL_ASCII_KEY                                            :   12     K83
       ==:18
KL_EMPTY                                                :   11     K0
       KF:61    KF:64    KF:68
KL_MASK_1_ON                                            :    8     K3
```

```
       KF:65
KL_MASK_2_ON                                            :    9     K4
       KF:66
KL_MASK_OFF                                             :   10     K5
       KF:67
KL_SOFT_KEY_OFF                                         :    6     K1
       KF:62
KL_SOFT_KEY_ON                                          :    7     K2
       KF:63
________________________________Static Symbols ________________________________

ML_NC_SOFT_KEY_ON                                       :   14     M3969
       LN:22     =:23
________________________________ Local Labels ________________________________

9085                                                    : 9085     LBL 9085
     67   75   5    14   20   29   35   41
9200                                                    : 9200     LBL 9200
     66   74
9202                                                    : 9202     LBL 9202
     4    19   34
9210                                                    : 9210     LBL 9210
     11   26   38
DELMASKE_SK                                             :   32     LBL 44
     30
EMPTY_SK                                                :   49     LBL 39
     25
MASKE1_SK                                               :    2     LBL 42
     28
MASKE2_SK                                               :   17     LBL 43
     29
________________________________ Local Labels ________________________________

OFF_SK                                                  :   52     LBL 40
     26
ON_SK                                                   :   57     LBL 41
     27
SOFT_KEY_DIPSLAY                                        :   70     LBL 46
     45
SOFT_KEY_DISPLAY                                        :   62     LBL 45
     44
SOFT_KEY_ROW                                            :   78     LBL 47
     63

________________________________ Global Labels ________________________________

AXIS_SUPERVISION                          HELPDIAG.SRC  :    8     LBL 7
     MAIN_PGM.SRC  88
GEAR_CHANGING                             GEAR.SRC      :   14     LBL 6
     MAIN_PGM.SRC  87
INITIALIZATION                            INITI472.SRC  :   15     LBL 0
     MAIN_PGM.SRC  75
MANUEL_BUTTON_FU                          DIRCBUT.SRC   :   26     LBL 3
     MAIN_PGM.SRC  84
M_FUNCTION                                M_FUNCT.SRC   :    8     LBL 1
     MAIN_PGM.SRC  77
NC_AXIS                                   AXIS.SRC      :   28     LBL 4
     MAIN_PGM.SRC  85
PLC_SOFT_KEYS                             SOFTKEYS.SRC  :   20     LBL 8
     MAIN_PGM.SRC  89
REFERENCE_ENDSWI                          REF_ENDL.SRC  :   15     LBL 2
     MAIN_PGM.SRC  82
SPINDLE_FUNCTION                          SPINDLE.SRC   :   11     LBL 5
     MAIN_PGM.SRC  86
________________________________Global Symbols ________________________________

BG_MPAXIS.0_CA_PA_1                       GLB_TCMB.DEF  :   34     B4088
     - not used -
BG_MPAXIS.1_CA_PA_2                       GLB_TCMB.DEF  :   35     B4089
     - not used -
BG_MPAXIS.2_CA_PA_3                       GLB_TCMB.DEF  :   36     B4090
     - not used -
BG_MPAXIS.3_CA_PA_4                       GLB_TCMB.DEF  :   37     B4091
     - not used -
BG_MPAXIS.4_CA_PA_5                       GLB_TCMB.DEF  :   38     B4092
     - not used -
BG_MPAXIS.X_CA_PA                         GLB_TCMB.DEF  :   33     B4088
     - not used -
BG_MPSPIN.0_CA_PA_S                       GLB_TCMB.DEF  :   39     B4093
     - not used -
FIRST_PL                                  CONFIG.DEF    :   10     K0
     - not used -
I_1_AXIS_MINUS                            GLB_IO.DEF    :   15     I133
     DIRCBUT.SRC    L:31

I_1_AXIS_PLUS                             GLB_IO.DEF    :   14     I138
```

```
     DIRCBUT.SRC    L:21

I_2_AXIS_MINUS                            GLB_IO.DEF    :   17     I134
     DIRCBUT.SRC    L:33

I_2_AXIS_PLUS                             GLB_IO.DEF    :   16     I137
     DIRCBUT.SRC    L:23

I_3_AXIS_MINUS                            GLB_IO.DEF    :   19     I135
     DIRCBUT.SRC    L:35

I_3_AXIS_PLUS                             GLB_IO.DEF    :   18     I136
     DIRCBUT.SRC    L:25

I_4_AXIS_MINUS                            GLB_IO.DEF    :   21     I140
     DIRCBUT.SRC    L:37

I_4_AXIS_PLUS                             GLB_IO.DEF    :   20     I139
     DIRCBUT.SRC    L:27

I_5_AXIS_MINUS                            GLB_IO.DEF    :   23     I147
     DIRCBUT.SRC    L:39

I_5_AXIS_PLUS                             GLB_IO.DEF    :   22     I146
     DIRCBUT.SRC    L:29

I_FEED_ENABLE                             GLB_IO.DEF    :   13     I5
     DIRCBUT.SRC    L:8

I_GEAR_RANGE_1                            GLB_IO.DEF    :   34     M991
     GEAR.SRC      LN:28    LN:68     L:77

I_GEAR_RANGE_2                            GLB_IO.DEF    :   35     M990
     GEAR.SRC      XO:29     O:69    ON:78

I_NC_START                                GLB_IO.DEF    :   24     I132
     DIRCBUT.SRC    L:10

________________________________Global Symbols _______________________________

I_NC_STOP                                 GLB_IO.DEF    :   25     I131
     DIRCBUT.SRC    L:3

I_NOT_EMERGENCY_STOP                      GLB_IO.DEF    :   10     I3
     AXIS.SRC      LN:8     ON:137
     SPINDLE.SRC   ON:19    ON:42
     GEAR.SRC      ON:41

I_RAPID_BUTTON                            GLB_IO.DEF    :   26     I141
     DIRCBUT.SRC    L:6

I_REF_ENDSWITCH_1_AXIS                    GLB_IO.DEF    :    7     I0
     REF_ENDL.SRC   L:3

I_REF_ENDSWITCH_2_AXIS                    GLB_IO.DEF    :    8     I1
     REF_ENDL.SRC   L:5

I_REF_ENDSWITCH_3_AXIS                    GLB_IO.DEF    :    9     I2
     REF_ENDL.SRC   L:7

I_REF_ENDSWITCH_4_AXIS                    GLB_IO.DEF    :   11     I4
     REF_ENDL.SRC   L:9

I_REF_ENDSWITCH_5_AXIS                    GLB_IO.DEF    :   12     I6
     REF_ENDL.SRC   L:11

I_SERVO_READY_1                           GLB_IO.DEF    :   28     I129
     AXIS.SRC      ON:9     ON:138
     SPINDLE.SRC   ON:43

I_SERVO_READY_2                           GLB_IO.DEF    :   29     I144
     AXIS.SRC      ON:10    ON:139
     SPINDLE.SRC   ON:44

KG_ERROR_MODUL_9200                       GLB_TCMB.DEF  :   75     K15
     SOFTKEYS.SRC  PS:45    PS:56

KG_ERROR_MODUL_9202                       GLB_TCMB.DEF  :   78     K18
     SOFTKEYS.SRC  PS:75    PS:94    PS:113

KG_ERROR_MODUL_9210                       GLB_TCMB.DEF  :   77     K17
     SOFTKEYS.SRC  PS:85    PS:104   PS:121

KG_ERROR_MODUL_9220                       GLB_TCMB.DEF  :   76     K16
     - not used -
MAX_NC_AXIS                               CONFIG.DEF    :   14     K5
```

```
     AXIS.SRC      >=:32

MG_1_CLAMP_MODE_ACTIV                     GLB_TCMB.DEF  :   17     M992
     INITI472.SRC   R:9
     AXIS.SRC      LN:113    L:136

MG_2_CLAMP_MODE_ACTIV                     GLB_TCMB.DEF  :   18     M993
     INITI472.SRC   R:10

MG_3_CLAMP_MODE_ACTIV                     GLB_TCMB.DEF  :   19     M994
     INITI472.SRC   R:11

________________________________Global Symbols _______________________________

MG_4_CLAMP_MODE_ACTIV                     GLB_TCMB.DEF  :   20     M995
     INITI472.SRC   S:12

MG_5_CLAMP_MODE_ACTIV                     GLB_TCMB.DEF  :   21     M996
     INITI472.SRC   S:13

MG_ACTIVE_PWM_AXIS                        GLB_TCMB.DEF  :   24     M3984
     INITI472.SRC   S:85    LB:91
     HELPDIAG.SRC   L:35

MG_ACTIVE_PWM_AXIS_1                      GLB_TCMB.DEF  :   25     M3984
     - not used -
MG_ACTIVE_PWM_AXIS_2                      GLB_TCMB.DEF  :   26     M3985
     - not used -
MG_ACTIVE_PWM_AXIS_3                      GLB_TCMB.DEF  :   27     M3986
     - not used -
MG_ACTIVE_PWM_AXIS_4                      GLB_TCMB.DEF  :   28     M3987
     - not used -
MG_ACTIVE_PWM_AXIS_5                      GLB_TCMB.DEF  :   29     M3988
     - not used -
MG_ACTIVE_PWM_AXIS_S                      GLB_TCMB.DEF  :   30     M3989
     INITI472.SRC   L:93

MG_CLOSED_LOOP                            GLB_TCMB.DEF  :   16     M992
     - not used -
MG_FUNCTION_ON                            GLB_TCMB.DEF  :   74     M4814
     SOFTKEYS.SRC   L:31     R:32    LN:35     S:36

MG_GEAR_CHANGE_ACTIV                      GLB_IO.DEF    :   32     M3982
     GEAR.SRC      AN:32    AN:40     S:48     R:101

MG_ONE_MARKER                             GLB_TCMB.DEF  :    6     M3999
     INITI472.SRC  LN:3      S:4       L:7
     AXIS.SRC       L:71
     GEAR.SRC       L:100

MG_SPINDLE_OFF_M05                        GLB_TCMB.DEF  :   10     M3995
     SPINDLE.SRC    O:17     L:27      L:40

MG_SPINDLE_ON_GEAR                        GLB_IO.DEF    :   31     M3983
     GEAR.SRC       S:89     R:102

MG_SPINDLE_ON_M03                         GLB_TCMB.DEF  :    8     M3997
     SPINDLE.SRC    L:10

MG_SPINDLE_ON_M04                         GLB_TCMB.DEF  :    9     M3996
     SPINDLE.SRC    L:13

MG_SPINDLE_RPM_ZERO                       GLB_TCMB.DEF  :   13     M3992
     - not used -
MG_SPI_POS_M19_R_M0X                      GLB_TCMB.DEF  :   11     M3994
     SPINDLE.SRC    L:3     LN:7       L:16

MG_S_CLAMP_MODE_ACTIV                     GLB_TCMB.DEF  :   22     M997
     - not used -
MG_T_I_N_SUPERVISION                      GLB_TCMB.DEF  :   12     M3993
     DIRCBUT.SRC   AN:4

________________________________Global Symbols _______________________________

     AXIS.SRC       O:140
     SPINDLE.SRC    O:18     O:28      O:41
     HELPDIAG.SRC   =:61

MG_ZERO_MARKER                            GLB_TCMB.DEF  :    7     M3998
     INITI472.SRC   L:5      R:6
     GEAR.SRC      PS:98

MOTOR_TEMP_1                              CONFIG.DEF    :   16     K255
     INITI472.SRC   L:24

MOTOR_TEMP_2                              CONFIG.DEF    :   17     K255
```

```
     INITI472.SRC   L:26

MOTOR_TEMP_3                              CONFIG.DEF    :   18     K255
     INITI472.SRC   L:28

MOTOR_TEMP_4                              CONFIG.DEF    :   19     K255
     INITI472.SRC   L:30

MOTOR_TEMP_5                              CONFIG.DEF    :   20     K255
     INITI472.SRC   L:32

MOTOR_TEMP_S                              CONFIG.DEF    :   21     K255
     INITI472.SRC   L:34

MP_READ_NC_SW                             CONFIG.DEF    :    8     K280472
     - not used -
NC_TYPE_PA                                CONFIG.DEF    :   12     K0
     - not used -
NP_M4000_S_IN_POSITION                    GLB_NC.DEF    :    7     M4000
     SPINDLE.SRC    A:8

NP_M4001_S_ANALOG_NOT_IN                  GLB_NC.DEF    :    8     M4001
     SPINDLE.SRC    A:49

NP_M4002_S_ANALOG_0_V                     GLB_NC.DEF    :    9     M4002
     SPINDLE.SRC    L:32    ON:36
     GEAR.SRC      PS:53

NP_M4070_STROBE_G_CODE                    GLB_NC.DEF    :   20     M4070
     GEAR.SRC      LN:12    AN:30    AN:39     O:46     L:86
                   LN:88

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:10
     SPINDLE.SRC   LN:46     L:48

NP_M4150_MANUEL_MODE                      GLB_NC.DEF    :   44     M4150
     DIRCBUT.SRC    L:13

NP_M4151_ELECTRONIC_HAND                  GLB_NC.DEF    :   45     M4151
     DIRCBUT.SRC    O:14     A:44

NP_M4155_REFERENCE_MODE                   GLB_NC.DEF    :   46     M4155
     MAIN_PGM.SRC   O:30

________________________________Global Symbols _______________________________

     DIRCBUT.SRC    O:15

NP_M4156_SOFTKEY_MANUAL_                  GLB_NC.DEF    :   47     M4156
     DIRCBUT.SRC    O:16

NP_M4172_1_PLC_AFTER_COM                  GLB_NC.DEF    :   49     M4173
     MAIN_PGM.SRC   O:24
     GEAR.SRC       O:4

NP_M4172_1_PLC_AFTER_MP_                  GLB_NC.DEF    :   50     M4174
     MAIN_PGM.SRC   O:25

NP_M4172_1_PLC_AFTER_POW                  GLB_NC.DEF    :   48     M4172
     MAIN_PGM.SRC   L:23
     GEAR.SRC       L:3

NP_M4203_ERROR_MODUL_9XX                  GLB_NC.DEF    :   56     M4203
     MAIN_PGM.SRC   L:21     L:40
     AXIS.SRC       L:54
     SPINDLE.SRC    L:61
     HELPDIAG.SRC   L:4      L:19     L:42     L:49
     SOFTKEYS.SRC   L:43     L:54     L:73     L:92     L:111

NP_M4300_POWERFAILON_MP4                  GLB_NC.DEF    :   62     M4300
     INITI472.SRC   L:14

NP_M4572_ENABLE_JOG_MODE                  GLB_NC.DEF    :   74     M4572
     INITI472.SRC   S:8
     DIRCBUT.SRC    L:43

NP_W1026_AXIS_IN_POSITIO                  GLB_NC.DEF    :   83     W1026
     AXIS.SRC       O:78     O:87     O:132

NP_W1032_REFERENCE_NECES                  GLB_NC.DEF    :   84     W1032
     MAIN_PGM.SRC   L:28

NP_W256_G_CODE_SPINDLE                    GLB_NC.DEF    :   99     W256
```

```
     GEAR.SRC       =:8   CASE:62     L:92

NP_W260_M_CODE                            GLB_NC.DEF    :   98     W260
     M_FUNCT.SRC    L:3      L:6

NP_W274_BUTTON_KEY_CODE                   GLB_NC.DEF    :  102     W274
     SOFTKEYS.SRC   L:17     =:21

NP_W302_NUMBER_PLC_SOFT_                  GLB_NC.DEF    :  101     W302
     SOFTKEYS.SRC   L:3   CASE:6      =:15

O_1_AXIS_ENABLE                           GLB_IO.DEF    :   42     O0
     AXIS.SRC       =:179

O_1_CLAMPING                              GLB_IO.DEF    :   48     O7
     AXIS.SRC       =:189

O_2_AXIS_ENABLE                           GLB_IO.DEF    :   43     O1
     AXIS.SRC       =:181

________________________________Global Symbols _______________________________

O_2_CLAMPING                              GLB_IO.DEF    :   49     O8
     AXIS.SRC       =:191

O_3_AXIS_ENABLE                           GLB_IO.DEF    :   44     O2
     AXIS.SRC       =:183

O_3_CLAMPING                              GLB_IO.DEF    :   50     O9
     AXIS.SRC       =:193

O_4_AXIS_ENABLE                           GLB_IO.DEF    :   45     O3
     AXIS.SRC       =:185

O_4_CLAMPING                              GLB_IO.DEF    :   51     O10
     AXIS.SRC       =:195

O_5_AXIS_ENABLE                           GLB_IO.DEF    :   46     O4
     AXIS.SRC       =:187

O_5_CLAMPING                              GLB_IO.DEF    :   52     O11
     AXIS.SRC       =:197

O_GEAR_RANGE_1                            GLB_IO.DEF    :   56     M991
     GEAR.SRC       S:70     R:79

O_GEAR_RANGE_2                            GLB_IO.DEF    :   57     M990
     GEAR.SRC       R:71     S:80

O_SPINDLE_SERVO_ENABLE                    GLB_IO.DEF    :   54     O15
     AXIS.SRC       L:35
     SPINDLE.SRC    S:39     R:45

PN_D756_S_NOMINAL_RPM_PL                  GLB_NC.DEF    :   97     D756
     GEAR.SRC       =:10     L:94

PN_ERROR_9171_SPI_POS                     GLB_TCMB.DEF  :   66     M4806
     SPINDLE.SRC    S:62

PN_ERROR_GEAR_SWITCHING                   GLB_TCMB.DEF  :   79     M4815
     GEAR.SRC       =:37     L:38

PN_ERROR_I2T_CAUTION                      GLB_TCMB.DEF  :   69     M4809
     HELPDIAG.SRC   S:13     O:57

PN_ERROR_I2T_LIMITATION                   GLB_TCMB.DEF  :   73     M4813
     HELPDIAG.SRC   S:17

PN_ERROR_MODUL_9XXX_SUPE                  GLB_TCMB.DEF  :   70     M4810
     HELPDIAG.SRC   S:5      S:20     S:43     S:50

PN_ERROR_MOD_9002                         GLB_TCMB.DEF  :   61     M4801
     MAIN_PGM.SRC   S:22

PN_ERROR_MOD_9005                         GLB_TCMB.DEF  :   62     M4802
     MAIN_PGM.SRC   S:41

PN_ERROR_MOD_9161                         GLB_TCMB.DEF  :   63     M4803
     AXIS.SRC       S:55
________________________________Global Symbols _______________________________

PN_ERROR_MOD_9167                         GLB_TCMB.DEF  :   60     M4800
     INITI472.SRC   S:23

PN_ERROR_MOTOR_TEMP                       GLB_TCMB.DEF  :   72     M4812
     HELPDIAG.SRC   S:41     O:58
```

```
PN_ERROR_NOT_USED_M_FUNC                  GLB_TCMB.DEF  :   65     M4805
     SPINDLE.SRC   AN:53

PN_ERROR_SERVO_ACTIV                      GLB_TCMB.DEF  :   67     M4807
     HELPDIAG.SRC   S:31     O:60

PN_ERROR_SPINDLE_ZERO                     GLB_TCMB.DEF  :   80     M4816
     GEAR.SRC       S:52

PN_ERROR_SUBMIT_QUEUE_FU                  GLB_TCMB.DEF  :   64     M4804
     INITI472.SRC   S:42

PN_ERROR_TEMP_POWERSUPPL                  GLB_TCMB.DEF  :   68     M4808
     HELPDIAG.SRC   S:8      L:56

PN_ERROR_UTILIZATION_MOT                  GLB_TCMB.DEF  :   71     M4811
     HELPDIAG.SRC   S:48     O:59

PN_M4005_S_M03_ANALOG_VO                  GLB_NC.DEF    :   11     M4005
     SPINDLE.SRC    S:11     R:14     R:20    LN:22     O:29
                    L:34

PN_M4006_S_M04_ANALOG_VO                  GLB_NC.DEF    :   12     M4006
     SPINDLE.SRC    R:12     S:15     R:21    AN:23     O:30
                    O:35

PN_M4007_S_M05_0V_STATUS                  GLB_NC.DEF    :   13     M4007
     SPINDLE.SRC    =:24

PN_M4009_S_SWING_LEFT                     GLB_NC.DEF    :   39     M4009
     GEAR.SRC    WHIL:61     R:104

PN_M4010_S_SWING_RIGHT                    GLB_NC.DEF    :   38     M4010
     GEAR.SRC       =:60     R:103

PN_M4012_S_CLOSE_LOOP_OP                  GLB_NC.DEF    :   14     M4012
     SPINDLE.SRC    L:25     R:26     S:31

PN_M4090_QUIT_G_CODE                      GLB_NC.DEF    :   28     M4090
     GEAR.SRC       R:13     S:87

PN_M4091_QUIT_S_CODE                      GLB_NC.DEF    :   29     M4091
     - not used -
PN_M4092_QUIT_M_FUNCTION                  GLB_NC.DEF    :   30     M4092
     SPINDLE.SRC    =:54

PN_M4130_STROBE_PLC_POS_                  GLB_NC.DEF    :   36     M4130
     SPINDLE.SRC    O:37    AN:50

PN_M4134_STROBE_G_STEP_R                  GLB_NC.DEF    :   37     M4134
     GEAR.SRC       S:5     AN:31     L:45

________________________________Global Symbols _______________________________

PN_M4560_NC_STOP_0_ACTIV                  GLB_NC.DEF    :   68     M4560
     DIRCBUT.SRC    =:5

PN_M4561_RAPIDE                           GLB_NC.DEF    :   69     M4561
     DIRCBUT.SRC    =:7

PN_M4562_AXIS_BUTTON_LAT                  GLB_NC.DEF    :   70     M4562
     DIRCBUT.SRC    =:12

PN_M4563_FEED_ENABLE                      GLB_NC.DEF    :   71     M4563
     DIRCBUT.SRC    =:9

PN_M4564_NC_START                         GLB_NC.DEF    :   72     M4564
     DIRCBUT.SRC    =:11

PN_W1038_CLOSED_LOOP_OPE                  GLB_NC.DEF    :   85     W1038
     AXIS.SRC       L:59     =:61     L:115    =:117    L:160
                    =:162

PN_W1040_CLOSED_LOOP_OPE                  GLB_NC.DEF    :   86     W1040
     AXIS.SRC       L:62     =:64     L:119    =:121    L:157
                    =:159

PN_W1042_SUPERVISION_INA                  GLB_NC.DEF    :   87     W1042
     AXIS.SRC       L:65     =:67     L:125    =:127

PN_W1044_ACTUL_NOMINAL_T                  GLB_NC.DEF    :   88     W1044
     AXIS.SRC       L:68     =:70     L:122    =:124

PN_W1046_MANUEL_DIRCETIO                  GLB_NC.DEF    :   89     W1046
     DIRCBUT.SRC    =:52
```

```
PN_W1048_MANUEL_DIRECTIO                  GLB_NC.DEF    :   90     W1048
     DIRCBUT.SRC    =:54

PN_W1050_JOG_MODE_POSIT_                  GLB_NC.DEF    :   91     W1050
     DIRCBUT.SRC    =:47

PN_W1052_JOG_MODE_POSIT_                  GLB_NC.DEF    :   92     W1052
     DIRCBUT.SRC    =:49

PN_W1054_REFERENCE_ENDSW                  GLB_NC.DEF    :   93     W1054
     REF_ENDL.SRC   =:14

TR_1_CLAMPING                             GLB_TCMB.DEF  :   89     T48
     AXIS.SRC      AN:107   AN:151

TR_1_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  101     T56
     HELPDIAG.SRC  LB:28

TR_2_CLAMPING                             GLB_TCMB.DEF  :   91     T49
     - not used -
TR_2_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  103     T57
     - not used -
TR_3_CLAMPING                             GLB_TCMB.DEF  :   93     T50
     - not used -
TR_3_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  105     T58
     - not used -
________________________________Global Symbols _______________________________

TR_4_CLAMPING                             GLB_TCMB.DEF  :   95     T51
     - not used -
TR_4_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  107     T59
     - not used -
TR_5_CLAMPING                             GLB_TCMB.DEF  :   97     T52
     - not used -
TR_5_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  109     T60
     - not used -
TR_6_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  111     T61
     - not used -
TR_7_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  113     T62
     - not used -
TR_8_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  115     T63
     - not used -
TR_GEAR_TIMOUT                            GLB_TCMB.DEF  :  118     T68
     GEAR.SRC      AN:35    LN:51

TR_GREAR_TOGGEL_ALL                       GLB_TCMB.DEF  :  120     T69
     GEAR.SRC      LN:56

TR_GREAR_TOGGEL_RIGHT                     GLB_TCMB.DEF  :  122     T70
     GEAR.SRC       L:59

TR_M_FUNC_DELAY                           GLB_TCMB.DEF  :   99     T55
     SPINDLE.SRC   AN:52

TS_1_CLAMPING                             GLB_TCMB.DEF  :   88     T0
     AXIS.SRC       =:105   LN:106    =:149   LN:150

TS_1_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  100     T8
     HELPDIAG.SRC  B=:22

TS_2_CLAMPING                             GLB_TCMB.DEF  :   90     T1
     - not used -
TS_2_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  102     T9
     - not used -
TS_3_CLAMPING                             GLB_TCMB.DEF  :   92     T2
     - not used -
TS_3_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  104     T10
     - not used -
TS_4_CLAMPING                             GLB_TCMB.DEF  :   94     T3
     - not used -
TS_4_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  106     T11
     - not used -
TS_5_CLAMPING                             GLB_TCMB.DEF  :   96     T4
     - not used -
TS_5_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  108     T12
     - not used -
TS_6_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  110     T13
     - not used -
TS_7_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  112     T14
     - not used -
TS_8_SERVO_SUPERVISON                     GLB_TCMB.DEF  :  114     T15
     - not used -
TS_GEAR_TIMEOUT                           GLB_TCMB.DEF  :  117     T20
     GEAR.SRC       =:16    AN:34
```

```
TS_GREAR_TOGGEL_ALL                       GLB_TCMB.DEF  :  119     T21
     GEAR.SRC       =:57

TS_GREAR_TOGGEL_RIGHT                     GLB_TCMB.DEF  :  121     T22
     GEAR.SRC       =:58

TS_M_FUNC_DELAY                           GLB_TCMB.DEF  :   98     T7
     SPINDLE.SRC    =:47    AN:51

WG_ACTIVE_PWM_AXIS                        GLB_TCMB.DEF  :   44     W4082
     INITI472.SRC   =:92     L:95     =:98
     AXIS.SRC       A:50

WG_MOTOR_TEMP                             GLB_TCMB.DEF  :   47     W4068
     HELPDIAG.SRC  >=:40

WG_MOTOR_TEMP_1                           GLB_TCMB.DEF  :   48     W4068
     INITI472.SRC   =:25

WG_MOTOR_TEMP_2                           GLB_TCMB.DEF  :   49     W4070
     INITI472.SRC   =:27

WG_MOTOR_TEMP_3                           GLB_TCMB.DEF  :   50     W4072
     INITI472.SRC   =:29

WG_MOTOR_TEMP_4                           GLB_TCMB.DEF  :   51     W4074
     INITI472.SRC   =:31

WG_MOTOR_TEMP_5                           GLB_TCMB.DEF  :   52     W4076
     INITI472.SRC   =:33

WG_MOTOR_TEMP_S                           GLB_TCMB.DEF  :   53     W4078
     INITI472.SRC   =:35

WG_MP10_ACTIVE_AXIS                       GLB_TCMB.DEF  :   42     W4086
     - not used -
WG_SERVO_ENABLE_INTERNAL                  GLB_TCMB.DEF  :   43     W4084
     AXIS.SRC       =:51
     HELPDIAG.SRC  LN:21     L:23
```

```
 ________________________________ Project Info ________________________________

Used Files:
File                           Class    Date
MAIN_PGM.SRC                   Module  14.11.97
GLB_TCMB.DEF                   Define  14.11.97
GLB_IO.DEF                     Define  14.11.97
GLB_NC.DEF                     Define  14.11.97
CONFIG.DEF                     Define  14.11.97
INITI472.SRC                   Module  14.11.97
M_FUNCT.SRC                    Module  14.11.97
REF_ENDL.SRC                   Module  14.11.97
DIRCBUT.SRC                    Module  14.11.97
AXIS.SRC                       Module  14.11.97
SPINDLE.SRC                    Module  14.11.97
GEAR.SRC                       Module  14.11.97
HELPDIAG.SRC                   Module  14.11.97
SOFTKEYS.SRC                   Module  14.11.97
MAC_LIB.DEF                    Include 14.11.97
MASKEN.SRC                     Include 14.11.97

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 3969-3999
Byte   (remanent) :    4- 127      -                   120- 127
Byte   (nonrem. ) : 2048-4095  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
```

5.6 Commands

5.6.1 Execution Times

Commands

PLC commands are divided into three groups according to the execution time.

- Group A: 0.0 to 0.5 µs
- Group B: 0.1 to 1.5 µs
- Group C: 1.0 to 15 µs
- Group 0: 0 µs

If the index register is used, the execution times increase by 0.05 to 0.2 µs per indexed command.

Command	IOMCT	BWDK	String	LBL/other
L,LN,L-,=,=N,=-	A	A	C	-
A,AN,O,ON,XO,XON	A	A	-	-
S,SN,R,RN	A	-	-	-
OVWR	-	-	C	-
+	-	A	C	-
-,X	-	A	-	-
/,MOD	-	B	-	-
==,<,>,<=,>=,<>	-	A	C	-
<<,>>	-	A	-	-
BT,BS,BR	-	A	-	-
LB,LW	B	-	-	-
LD	C	-	-	-
=B,=W	C	-	-	-
=D	C	-	-	-
PL,PS	B	B	-	-
PLL,PLW,PSL,PSW,PSX,PLX	-	-	-	A
A[.. XON[	-	-	-	A
] for these commands	-	-	-	A
+[,-[,X[,	-	-	-	A
] for these commands	-	-	-	A
/[,MOD[	-	-	-	A
] for these commands	-	-	-	B
==[.. <>[	-	-	-	A
] for these commands	-	-	-	A
LBL	-	-	-	0
JP,JPT,JPF	-	-	-	A

Command	IOMCT	BWDK	S	LBL/no op.
CM,CMT,CMF in its own source module	-	-	-	B
CM,CMT,CMF on global label	-	-	-	C
EM,EMT,EMF	-	-	-	A
IFc,ELSE,UNTILc,WHILEc	-	-	-	A
ENDI,REPEAT,ENDW	-	-	-	0
CASE	-	B	-	-
CM for Case	-	-	-	0
ENDC	-	-	-	0
SUBM,RPLY,CAN	C	-	-	-
LX,=X	-	-	-	A
INCW,DECW,INCX,DECX	-	-	-	A
INC,DEC	-	A	-	-

Modules
Execution times for modules that run as a submit job:
These modules communicate with other parts of the control software (e.g. screen display, file system). This results in unavoidable waiting times and response times that make it impossible to specify the execution times. They must therefore be determined empirically. Some factors that influence response times are:

- CPU load from processing an NC program
- Load on the file system, for example from copying
- Load of the CPU and the video system, e.g. due to a PLC window

Execution times for modules that run in the cyclical program:
Unless indicated otherwise, the execution time of this module lies between 10 µs and 100 µs. Unnecessary calls should therefore be avoided (e.g. cyclical setting of the pulse release, of a soft-key row, etc.).

The execution time are even longer for certain comprehensive modules:

Numbers	Function	Time
9002	Read 64 inputs from a PL	450 µs
9005	Write to 32 outputs of a PL	280 µs
9004	Generate edge markers (e.g. 100 pieces)	150 µs
9003	Read analog input of a PL	150 µs

5.6.2 LOAD (L)

Abbreviation for the PLC Editor: L (LOAD)

Logic execution with LOAD command
Operands: M, I, O, T, C

Operation:
The addressed operand is copied into the Accumulator. A load command is always used at the start of a logic chain, in order to enable subsequent gating commands.

Example:
Input I4 and Input I5 is to be gated with AND and the result assigned to Output O2. Thus the logic state of Input I4 is loaded into the Accumulator to enable subsequent gating commands.

Initial state:	Input	I4	= 1
	Input	I5	= 0
	Output	O2	= ?

Line	Instruction	Accumulator Contents			Operand-Contents
	Bit	31 . . .	7	0	
		... x x x x x x	X	x x x x x x x	
1	L I4	... x x x x x x	1	x x x x x x x	1
2	A I5	... x x x x x x	0	x x x x x x x	0
3	= O2	... x x x x x x	0	x x x x x x x	0

Line 1: The operand contents are loaded into the Logic Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with AND.
Line 3: The gating result is assigned to output O2.

Word execution with the LOAD command
Operands: B, W, D, K

Operation:
The addressed Operand (B, W, D) or a Constant (K) is copied into the Word Accumulator. In addition, the Accumulator is filled, if necessary, according to the sign bit. In contrast to logic execution the start of a word gating chain must always be with the L command. It is not possible to use a gating command.

Example:
A Constant and Byte B5 is to be gated with AND and the result assigned to Byte B8.

Initial state:	Byte	B5 = 2A (hex)
	Constant:	54 = 36 (hex)
	Byte	B8 = ?

Line	Instruction	Accumulator Contents			Operand Contents
	Bit	31 . . . 15	7	0	7 0
		... x x x x x x x x x x	x	x x x x x x x	
1	L K+54	... 0 0 0 0 0 0 0 0 0 0	0	0 1 1 0 1 1 0	
2	A B5	... 0 0 0 0 0 0 0 0 0 0	0	0 1 0 0 0 1 0	00101010
3	= B8	... 0 0 0 0 0 0 0 0 0 0	0	0 1 0 0 0 1 0	00100010

Line 1: The Constant is loaded into the Word Accumulator.
Line 2: The contents of the Word Accumulator and Byte B5 are gated with AND.
Line 3: The gating result is assigned to Byte B8.

5.6.3 LOAD NOT (LN)

Abbreviation for the PLC Editor: LN (LOAD NOT)

Logic execution with the LOAD NOT command
Operands: M, I, O, T, C

Operation:
The complement of the addressed operand is loaded into the Logic Accumulator. A load command is always used at the start of a logic chain in order to enable subsequent gating commands.

Example:
The inverted logic state of Input I4 and Input I5 is to be gated with AND and the result assigned to Output O2. Thus the inverted logic state of Input I4 is loaded into the Accumulator to enable subsequent gating commands.

Initial state:		
	Input	I4 = 0
	Input	I5 = 1
	Output	O2 = ?

Line	Instruction	Accumulator Contents			Operand Contents
	Bit 31	. . .	7	0	
		... x x x x x x	X	x x x x x x x	
1	LN I4	... x x x x x x	1	x x x x x x x	0
2	A I5	... x x x x x x	1	x x x x x x x	1
3	= O2	... x x x x x x	1	x x x x x x x	1

Line 1: The inverted operand contents are loaded into the Logic Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with AND.
Line 3: The gating result is assigned to Output O2.

Word execution with the LOAD NOT command

Operands: B, W, D, K

Operation:
The complement of the contents of the addressed Operand (B, W, D) or Constant (K) is loaded into the Word Accumulator. In addition, the Accumulator is filled, if necessary, according to the sign bit. In contrast to logic execution a word gating chain must always start with a load command. It is not possible to use a gating command.

Example:
The complement of Byte B6 and Byte B5 is to be gated with AND and the result assigned to Byte B8.

Initial state:	Byte B5 = 2A (hex) Byte B6 = B6 (hex) Byte B8 = ?

Line	Instruction		Accumulator Contents	Operand Contents
	Bit	31	. . . 15 7 0	7 0
			... x x x x x x x x x x x x x x x x x x	
1	LN B6		... 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1	10110110
2	A B5		... 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	00101010
3	= B8		... 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	00001000

Line 1: The inverted contents of Byte B6 are loaded into the Word Accumulator.
Line 2: The contents of the Word Accumulator and Byte B5 are gated with AND.
Line 3: The gating result is assigned to Byte B8.

5.6.4 LOAD TWO'S-COMPLEMENT (L–)

Abbreviation for the PLC Editor: L– (LOAD MINUS)

Operands: B, W, D, K

Operation:
The contents of the addressed Operand (B, W, D) or a Constant (K) are loaded into the Word Accumulator as a two's complement. In addition, the Accumulator is filled, if necessary, according to the sign bit. 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 may only be used with Word execution.

Example:
The contents of Byte B5 is to be negated, added to Byte B6 and the result assigned to Byte B8.

Initial state:
Byte B5 = 15 (dec)
Byte B6 = 20 (dec)
Byte B8 = ?

Line	Instruction	Accumulator Contents		Operand Contents	
	Bit	31 . . . 15 7 0			7 0
		... x x x x x x x x x x x x x x x x x x			
1	L– B5	... 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1	(–15)	(+15)	00001111
2	+ B6	... 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	(+ 5)	(+20)	00010100
3	= B8	... 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	(+ 5)	(+5)	00000101

To aid understanding of this example, the contents of the Accumulator and operands are shown as decimal values in parentheses.

Line 1: The contents of Byte B5 are loaded into the Accumulator and the sign of the value is inverted.
Line 2: The contents of the Word Accumulator and Byte B6 are added.
Line 3: The result is assigned to Byte B8.

5.6.5 LOAD BYTE (LB)

Abbreviation for the PLC Editor: LB (LOAD BYTE)

Operands: M, I, O, T, C

Operation:
With the command LB, 8 Markers, Inputs, Outputs, Timers or Counters with ascending numbering are loaded into the Word Accumulator. Each operand occupies 1 bit in the Accumulator. The designated operand address occupies the LSB in the Accumulator, the designated address + 1 the LSB + 1 and so on. In this way, the last affected operand occupies the MSB! If necessary, the Accumulator is filled according to the sign bit.

5.6.6 LOAD WORD (LW)

Abbreviation for the PLC Editor: LW (LOAD WORD)

Operands: M, I, O, T, C

Operation:
With the command LW, 16 Markers, Inputs, Outputs, Timers or Counters with ascending numbering are loaded into the Word Accumulator. Each operand occupies 1 bit in the Accumulator. The designated operand address occupies the LSB in the Accumulator, the designated address + 1 the LSB + 1 and so on. In this way, the last affected operand occupies the MSB! If necessary, the Accumulator is filled according to the sign bit.

5.6.7 LOAD DOUBLEWORD (LD)

Abbreviation for the PLC Editor: LD (LOAD DOUBLE WORD)

Operands: M, I, O, T, C

Operation:
With the command LD, 32 Markers, Inputs, Outputs, Timers or Counters with ascending numbering are loaded into the Word Accumulator. Each operand occupies 1 bit in the Accumulator. The designated operand address occupies the LSB in the Accumulator, the designated address + 1 the LSB + 1 and so on. In this way, the last affected operand occupies the MSB! If necessary, the Accumulator is filled according to the sign bit.

Example for the Commands LB, LW and LD:
Via the Inputs I3 to I10, a binary coded value is to be read in and assigned to Byte B8 for further use.

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

Line	Instruction		Accumulator Contents	Operand Contents
		Bit	31 . . . 15 7 0	I10 I3
			... x x x x x x x x x x x x x x x x x x	
1	LB I3		... 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1	01100111
				7 0
2	= B8		... 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1	01100111

Line 1: Inputs I3 to I10 are loaded into the Word Accumulator (Bit 0 to Bit 7).
Line 2: The Accumulator Contents are assigned to Byte 8.

The Commands LW and LD are processed in the same way except that 16 or 32 operands are used accordingly.

5.6.8 ASSIGN (=)

Abbreviation for the PLC Editor: = (STORE)

Logic execution with the ASSIGN command
Operands: M, I, O, T, C

Operation:
ASSIGN in conjunction with the Logic-Operands (M, I, O, T, C) copies the contents of the Logic Accumulator to the addressed operand. The = command is only used at the end of a logic chain in order that a gating result is available. The command may be used several times in succession (see example).

Example:
Input I4 and Input I5 should be gated with AND and the result assigned to Outputs O2 and O5.

Initial state:	Input	I4 = 1
	Input	I5 = 0
	Output	O5 = ?

Line	Instruction	Accumulator Contents			Operand Contents
	Bit	31 . . .	7	0	
		... x x x x x x	X	x x x x x x x	
1	L I4	... x x x x x x	1	x x x x x x x	1
2	A I5	... x x x x x x	0	x x x x x x x	0
3	= O2	... x x x x x x	0	x x x x x x x	0
4	= O5	... x x x x x x	0	x x x x x x x	0

Line 1: The operand contents are loaded into the Logic Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with AND.
Line 3: The gating result is assigned to Output O2.
Line 4: The gating result is assigned to Output O5.

Word execution with the ASSIGN command
Operands: B, W, D

Operation:
ASSIGN in conjunction with the Word Operands (B, W, D) copies the contents of the Word Accumulator to the addressed operand. The = command is only used at the end of a logic chain in order that a gating result is available. The command may be used several times in succession (see example).

Example:
A Constant and the contents of Byte B5 are to be gated with UND and the result assigned to Bytes B8 and B10.

Initial state:	Byte	B5 = 2A	(hex)
	Constant	54 = 36	(hex)
	Byte	B8 = ?	
	Byte	B10 = ?	

Line	Instruction	Accumulator Contents	Operand contents
	Bit	31 . . . 15 7 0	7 0
		... x x x x x x x x x x x x x x x x x x	
1	L K+54	... 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	
2	= B8	... 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	00110110
3	A B5	... 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0	00101010
4	= B8	... 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0	01010000
5	= B10	... 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0	01010000

Line 1: The Constant is loaded into the Logic Accumulator.
Line 2: The contents of the Word Accumulator is assigned to Byte B5.
Line 3: The contents of the Word Accumulator and Byte B5 are gated with AND.
Line 4: The gating result is assigned to Byte B8.
Line 5: The gating result is assigned to Byte B10.

5.6.9 ASSIGN BYTE (B=)

Abbreviation for the PLC Editor: B= (STORE BYTE)

Operands: M, I, O, T, C

Operation:
With the command B=, 8 bits are copied from the Word Accumulator to Markers, Inputs, Outputs, Timers or Counters with ascending numbering. Each bit corresponds to 1 operand. The LSB in the Accumulator is copied to the designated operand address, the LSB + 1 to the designated address + 1 and so on. The last affected operand is occupied by the MSB.

5.6.10 ASSIGN WORD (W=)

Abbreviation for the PLC Editor: W= (STORE WORD)

Operands: M, I, O, T, C

Operation:
With the command W=, 16 bits are copied from the Word Accumulator to Markers, Inputs, Outputs, Timers or Counters with ascending numbering. Each bit corresponds to 1 operand. The LSB in the Accumulator is copied to the designated operand address, the LSB + 1 to the designated address + 1 and so on. The last affected operand is occupied by the MSB.

5.6.11 ASSIGN DOUBLEWORD (D=)

Abbreviation for the PLC Editor: D= (STORE DOUBLE)

Operands: M, I, O, T, C

Operation:
With the command D=, 32 bits are copied from the Word Accumulator to Markers, Inputs, Outputs, Timers or Counters with ascending numbering. Each bit corresponds to 1 operand. The LSB in the Accumulator is copied to the designated operand address, the LSB + 1 to the designated address + 1 and so on. The last affected operand is occupied by the MSB.

Example:
A bit pattern, as defined in Word W8 , is to be assigned to Outputs O5 to O20.

Initial state: Word W8: 36 FF (hex)

Line	Instruction		Accumulator Contents			Operand Contents	
		Bit	31 . . . 15	7	0	15 8	7 0
			... x x x x x x x x x x	x	x x x x x x x		
1	L W8		... 0 0 0 0 1 1 0 1 1 0	1	1 1 1 1 1 1 1	00110110	11111111
						O20 ...	... O5
2	W= O5		... 0 0 0 0 1 1 0 1 1 0	1	1 1 1 1 1 1 1	00110110	11111111

Line 1: The contents of Word W8 are loaded into the Accumulator.
Line 2: The contents of the Accumulator are assigned to the Outputs O5 to O20.

The Commands B= and D= are processed in the same way except that 8 or 32 bits are used accordingly.

5.6.12 ASSIGN NOT (=N)

Abbreviation for the PLC Editor: =N (STORE NOT)

Logic processing
Operands: M,I,O,T,C

Operation:
An ASSIGN NOT in conjunction with a logic operand (M,I,O,T,C) copies the one's complement of the contents of the logic accumulator to the addressed operand.
For example see ASSIGN command (=).

Word processing
Operands: B,W;D

Operation:
An ASSIGN NOT in conjunction with a word operand (B,W,D) copies the one's complement of the contents of the word accumulator to the addressed operand. For an example, see ASSIGN command (=)

5.6.13 ASSIGN TWO'S COMPLEMENT (=-)

Abbreviation for the PLC Editor: = - (STORE MINUS)
Operands: B, W, D

Operation:
An ASSIGN TWO'S COMPLEMENT copies the two's complement of the contents of the word accumulator to the addressed operand. For example see ASSIGN command (=).

5.6.14 SET (S)

Abbreviation for the PLC Editor: S (SET)

Byte value in parentheses:
With certain preceding program sequences, the command may be shortened.

Operands: M, I, O, T, C

Operation:
The function of the command depends on the contents of the Logic Accumulator. If the Logic Accumulator = 1, the addressed operand is set to 1, otherwise the operand remains unchanged. An S-command is used at the end of a logic chain so that the gating result may influence the operand. The command may be used several times in succession (see example).

Example:
Input I4 and input I5 should be gated with OR.
If the gating result is 1, output O2 and marker M500 should be set.

Initial state:			
	Input	I4	= 1
	Input	I5	= 0
	Output	O2	= ?
	Marker	M500	= ?

Line	Instruction	Accumulator Contents	Operand Contents
	Bit	31 . . . 7 0	
		... x x x x x x X x x x x x x x	
1	L I4	... x x x x x x 1 x x x x x x x	1
2	O I5	... x x x x x x 1 x x x x x x x	0
3	S O2	... x x x x x x 1 x x x x x x x	1
4	S M500	... x x x x x x 1 x x x x x x x	1

Line 1: The contents of the operand are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with OR.
Line 3: The gating result = 1: output O2 is set.
Line 4: The gating result = 1: marker M500 is set.

5.6.15 RESET (R)

Abbreviation for the PLC Editor: R (RESET)

Byte value in parentheses:
With certain preceding program sequences the command may be shortened.

Operands: M, I, O, T, C

Operation:
The function of the command is dependent on the contents of the Logic Accumulator. If the Logic Accumulator = 1, the addressed operand is set to 0, otherwise the operand remains unchanged. An R command is used at the end of a logic chain, in order that a gating result may influence the operand. The command may be used several times in succession (see example).

Example:
Input I4 and Input I5 should be gated with OR.
If the gating result = 1, Output O2 and Marker M500 should be reset.

Initial state:	Input	I4	= 1
	Input	I5	= 0
	Output	O2	= ?
	Marker	M500	= ?

Line	Instruction	Accumulator Contents	Operand Contents
	Bit	31 . . . 7 0	
		... x x x x x x | X | x x x x x x x	
1	L I4	... x x x x x x | 1 | x x x x x x x	1
2	O I5	... x x x x x x | 1 | x x x x x x x	0
3	R O2	... x x x x x x | 1 | x x x x x x x	0
4	R M500	... x x x x x x | 1 | x x x x x x x	0

Line 1: The operand contents are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and input I5 are gated with OR.
Line 3: The gating result = 1: Output O2 is reset.
Line 4: The gating result = 1: Marker 500 is reset.

5.6.16 SET NOT (SN)

Abbreviation for the PLC Editor: SN (SET NOT)

Byte value in parentheses:
With certain preceding program sequences the command may be shortened.

Operands: M, I, O, T, C

Operation:
The function of the command is dependent upon the contents of the Logic Accumulator. If the Logic Accumulator = 0 , then the addressed operand is set to 1, otherwise the operand remains unchanged. An SN command is used at the end of a logic chain, in order that a gating result may influence the operand. The command may be used several times in succession (see example).

Example:
Input I4 and Input I5 are to be gated with OR.
If the gating result = 0, Output O2 and Marker M500 are set.

Initial state:	Input	I4	= 0
	Input	I5	= 0
	Output	O2	= ?
	Marker	M500	= ?

Line	Instruction	Accumulator Contents			Operand Contents
	Bit	31 . . .	7	0	
		... x x x x x x	X	x x x x x x x	
1	L I4	... x x x x x x	0	x x x x x x x	0
2	O I5	... x x x x x x	0	x x x x x x x	0
3	SN O2	... x x x x x x	0	x x x x x x x	1
4	SN M500	... x x x x x x	0	x x x x x x x	1

Line 1: The operand contents are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with OR.
Line 3: The gating result = 0: Output O2 is set.
Line 4: The gating result = 0: Marker 500 is set.

5.6.17 RESET NOT (RN)

Abbreviation for the PLC Editor: RN (RESET NOT)

Byte value in parentheses:
With certain preceding program sequences the command may be shortened.

Operands: M, I, O, T, C

Operation:
The function of the command is dependent upon the contents of the Logic Accumulator. If the Logic Accumulator = 0, then the addressed operand is set to 0, otherwise the operand remains unchanged. An RN command is used at the end of a logic chain, in order that a gating result may influence the operand. The command may be used several times in succession (see example).

Example:
Input I4 and Input I5 are to be gated with OR.
If the gating result = 0, Output O2 and Marker M500 are reset.

Initial state:	Input	I4	= 0
	Input	I5	= 0
	Output	O2	= ?
	Marker	M500	= ?

Line	Instruction		Accumulator Contents	Operand Contents
		Bit	31 . . . 7 0	
			... x x x x x x X x x x x x x x	
1	L I4		... x x x x x x 0 x x x x x x x	0
2	O I5		... x x x x x x 0 x x x x x x x	0
3	RN O2		... x x x x x x 0 x x x x x x x	0
4	RN M500		... x x x x x x 0 x x x x x x x	0

Line 1: The operand contents are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with OR.
Line 3: The gating result = 0: Output O2 is reset.
Line 4: The gating result = 0: Marker M500 is reset.

5.6.18 AND (A)

Abbreviation for the PLC Editor: A (AND)

Logic execution with AND command
Operands: M, I, O, T, C

Operation:
This command functions in different ways according to its position in the program:

a) At the start of a logic chain the command functions as an L command, i.e. the logic state of the operand is loaded into the Logic Accumulator. This is to ensure compatibility with the TNC 355 control which did not have the special L command.
In PLC programs, a logic chain should always be started with a load command (see L, LN, L–).

b) Within a logic chain the contents of the Logic Accumulator and the logic state of the operand (M, I, O, T, C) are gated with AND. The gating result is stored in the Logic Accumulator.

Example:
Input I4 and Input I5 are to be gated with AND and the result assigned to Output O2.

Initial state:	Input	I4	= 1
	Input	I5	= 0
	Output	O2	= ?

```
Line  Instruction            Accumulator Contents                 Operand Contents

                  Bit   31            .  .  .    7                      0
                           ... x x x x x x | x | x x x x x x x

1     L  I4                ... x x x x x x | 1 | x x x x x x x          [1]

2     A  I5                ... x x x x x x | 0 | x x x x x x x          [0]

3     =  O2                ... x x x x x x | 0 | x x x x x x x          [0]
```

Line 1: The operand contents are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with AND.
Line 3: The gating result is assigned to Output O2.

Word execution with the AND command
Operands: B, W, D, K

Operation:
The contents of the Word Accumulator and the contents of the operand (B, W, D, K) are gated with AND. 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 is gated with bit 1 of the operand and so on.
The result of the operation is stored in the Word Accumulator.

Example:
The contents of Byte B5 and Byte B6 should be gated with AND and the result assigned to Byte B8.

Initial state:	Byte B5 = 2A	(hex)
	Byte B6 = 36	(hex)
	Byte B8 = ?	

Line	Instruction	Accumulator Contents			Operand Contents
	Bit	31 . . . 15	7	0	15 8 7 0
		... x x x x x x x x x x	x	x x x x x x x	
1	L B6	... 0 0 0 0 0 0 0 0 0 0	0	0 1 1 0 1 1 0	00110110
2	A B5	... 0 0 0 0 0 0 0 0 0 0	0	0 1 0 0 0 1 0	00101010
3	= B8	... 0 0 0 0 0 0 0 0 0 0	0	0 1 0 0 0 1 0	00100010

Line 1: The contents of Byte B6 are loaded into the Accumulator.
Line 2: The contents of the Word Accumulator and Byte B5 are gated with AND.
Line 3: The gating result is assigned to Byte B8.

5.6.19 AND NOT (AN)

Abbreviation for the PLC Editor: AN (AND NOT)

Logic execution with the AND NOT command
Operands: M, I, O, T, C

Operation:
This command functions in different ways according to its position in the program:
a) At the start of a logic chain the command functions as an LN command, i.e. the complement of the operand is loaded into the Logic Accumulator. A logic chain, however, should always be started with a load command (see L, LN, L–).
b) Within a logic chain, the contents of the Logic Accumulator and the logic state of the operand (M, I, O, T, C) are gated with AND NOT.
The gating result is stored in the Logic Accumulator.

Example:
Input I4 and Input I5 should be gated with AND NOT and the result assigned to Output O2.

Initial state:	Input	I4	= 1
	Input	I5	= 1
	Output	O2	= ?

```
Line  Instruction            Accumulator Contents                 Operand Contents

               Bit   31         .  .  .   7                   0
                     ... x x x x x x | x | x x x x x x x

1     L  I4          ... x x x x x x | 1 | x x x x x x x              [1]

2     AN  I5         ... x x x x x x | 0 | x x x x x x x              [1]

3     =  O2          ... x x x x x x | 0 | x x x x x x x              [0]
```

Line 1: The operand contents are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with AND NOT.
Line 3: The gating result is assigned to Output O2.

Word execution with the AND NOT command

Operands: B, W, D, K

Operation:
The contents of the Word Accumulator and the contents of the operand (B, W, D, K) are gated with AND 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 in the Accumulator is gated with bit 0 in the operand.
Bit 1 in the Accumulator is gated with bit 1 in the operand and so on.
The result of the operation is stored in the Word Accumulator.

Example:
The contents of Word W4 and Word W6 should be gated with AND NOT and the result assigned to Word W8.

Initial state: Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Line	Instruction		Accumulator Contents			Operand Contents	
		Bit	31 . . . 15	7	0	15 8	7 0
			... x x x x x x x x x x	x	x x x x x x x		
1	L W6		... 0 0 0 0 1 1 1 1 0 0	0	0 1 1 0 1 1 0	00111100	00110110
2	AN W4		... 0 0 0 0 0 0 1 0 0 0	0	0 0 1 0 1 0 0	00110110	10101010
3	= W8		... 0 0 0 0 0 0 1 0 0 0	0	0 0 1 0 1 0 0	00001000	00010100

Line 1: The contents of Word W6 are loaded into the Accumulator.
Line 2: The contents of the Word Accumulator and Word W4 are gated with AND NOT.
Line 3: The gating result is assigned to Word W8.

5.6.20 OR (O)

Abbreviation for the PLC Editor: O (OR)

Logic execution with OR command
Operands: M, I, O, T, C

Operation:
This command functions in different ways according to its position in the program:

a) At the start of a logic chain the command functions as an L command, i.e. the logic state of the operand is loaded into the Logic Accumulator. A logic chain, however, should always be started with a load command (see L, LN, L–).

b) Within a logic chain, the contents of the Logic Accumulator and the logic state of the operand (M, I, O, T, C) are gated with OR.
The result of the operation is stored in the Logic Accumulator.

Example:
Input I4 and Input I5 are to be gated with OR and the result assigned to Output O2.

Initial state:	Input	I4 = 0
	Input	I5 = 1
	Output	O2 = ?

Line	Instruction		Accumulator Contents	Operand Contents
		Bit	31 . . . 7 0	
			... x x x x x x x x x x x x x x	
1	L I4		... x x x x x x 0 x x x x x x x	0
2	O I5		... x x x x x x 1 x x x x x x x	1
3	= O2		... x x x x x x 1 x x x x x x x	1

Line 1: The operand contents are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with OR.
Line 3: The gating result is assigned to Output O2.

Word execution with the OR command

Operands: B, W, D, K

Operation:
The contents of the Word Accumulator and the contents of the operand (B, W, D, K) are gated with 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 in the Accumulator is gated with bit 0 in the operand
Bit 1 in the Accumulator is gated with bit 1 in the operand and so on.

The result of the operation is stored in the Word Accumulator.

Example:
The contents of Byte B5 and Byte B6 are to be gated with OR and the result assigned to Word W8.

Initial state:				
	Byte	B5	= 2A	(hex)
	Byte	B6	= 36	(hex)
	Word	W8	= ?	

Line	Instruction	Accumulator Contents			Operand Contents	
	Bit	31 . . . 15	7	0	15 8	7 0
		... x x x x x x x x x x	x	x x x x x x x		
1	L B6	... 0 0 0 0 0 0 0 0 0 0	0	0 1 1 0 1 1 0		00110110
2	O B5	... 0 0 0 0 0 0 0 0 0 0	0	0 1 1 1 1 1 0		00101010
3	= W8	... 0 0 0 0 0 0 0 0 0 0	0	0 1 1 1 1 1 0	00000000	00111110

Line 1: The contents of Byte B6 are loaded into the Accumulator.
Line 2: The contents of the Word Accumulator and Byte B5 are gated with OR.
Line 3: The gating result is assigned to Word W8.

5.6.21 OR NOT (ON)

Abbreviation for the PLC Editor: ON (OR NOT)

Logic execution with OR NOT command
Operands: M, I, O, T, C

Operation:
This command functions in different ways according to its position in the program:
a) At the start of a logic chain this command functions as an LN command, i.e. the complement of the operand is loaded into the Logic Accumulator. A logic chain, however, should always be started with a load command (see L, LN, L–).
b) Within a logic chain, the contents of the Logic Accumulator and the logic state of the operand (M, I, O, T, C) are gated with OR NOT. The result of the operation is stored in the Logic Accumulator.

Example:
Input I4 and Input I5 are to be gated with OR NOT and the result assigned to Output O2.

Initial state:	Input	I4	= 0
	Input	I5	= 0
	Output	O2	= ?

Line	Instruction	Accumulator Contents			Operand Contents
	Bit	31 . . .	7	0	
		... x x x x x x	x	x x x x x x x	
1	L I4	... x x x x x x	0	x x x x x x x	0
2	ON I5	... x x x x x x	1	x x x x x x x	0
3	= O2	... x x x x x x	1	x x x x x x x	1

Line 1: The operand contents are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with OR NOT.
Line 3: The gating result is assigned to Output O2.

Word execution with the OR NOT command

Operands: B, W, D, K

Operation:

The contents of the Word Accumulator and the contents of the operand (B, W, D, K) are gated 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 in the Accumulator is gated with bit 0 in the operand
Bit 1 in the Accumulator is gated with bit 1 in the operand and so on.

The result of the operation is stored in the Word Accumulator.

Example:

The contents of Word W4 and Word W6 are to be gated with OR NOT and the result assigned to Word W8.

Initial state:	
	Word W4 = 36 AA (hex)
	Word W6 = 3C 36 (hex)
	Word W8 = ?

Line	Instruction		Accumulator Contents			Operand Contents	
		Bit	31 . . . 15	7	0	15 8	7 0
			... x x x x x x x x x x	x	x x x x x x x		
1	L W6		... 0 0 0 0 1 1 1 1 0 0	0	0 1 1 0 1 1 0	00111100	00110110
2	ON W4		... 1 1 1 1 1 1 1 1 0 1	0	1 1 1 0 1 1 1	00110110	10101010
3	= W8		... 1 1 1 1 1 1 1 1 0 1	0	1 1 1 0 1 1 1	11111101	01110111

Line 1: The contents of Word W6 are loaded into the Accumulator.
Line 2: The contents of the Word Accumulator and Word W4 are gated with OR NOT.
Line 3: The gating result is assigned to Word W8.

5.6.22 EXCLUSIVE OR (XO)

Abbreviation for the PLC Editor: XO (EXCLUSIVE OR)

Logic execution with the EXCLUSIVE OR command
Operands: M, I, O, T, C

Operation:
This command functions in different ways according to its position in the program:

a) At the start of a logic chain the command functions as an L command, i.e. the logic state of the operand is loaded into the Logic Accumulator. A logic chain, however, should always be started with a load command (see L, LN, L–).

b) Within a logic chain the contents of the Logic Accumulator and the logic state of the operand (M, I, O, T, C) are gated with EXCLUSIVE OR. The result of the operation is stored in the Logic Accumulator.

Example:
Input I4 and Input I5 are to be gated with EXCLUSIVE OR and the result assigned to Output O2.

Initial state:	Input	I4 = 1
	Input	I5 = 1
	Output	O2 = ?

Line	Instruction	Accumulator Contents	Operand Contents
	Bit	31 . . . 7 0	
		... x x x x x x | X | x x x x x x x	
1	L I4	... x x x x x x | 1 | x x x x x x x	1
2	XO I5	... x x x x x x | 0 | x x x x x x x	1
3	= O2	... x x x x x x | 0 | x x x x x x x	0

Line 1: The operand contents are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and Input I5 are gated with EXCLUSIVE OR.
Line 3: The gating result is assigned to Output O2.

Word execution with the EXCLUSIVE OR command

Operands: B, W, D, K

Operation:
The contents of the Word Accumulator and the contents of the operand (B, W, D, K) are gated 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 in the Accumulator is gated with bit 0 in the operand
Bit 1 in the Accumulator is gated with bit 1 in the operand and so on.
The result of the operation is stored in the Word Accumulator.

Example:
The contents of Byte B5 and Byte B6 are to be gated with EXCLUSIVE OR and the result assigned to Word W8.

Initial state				
	Byte	B5	= 2A	(hex)
	Byte	B6	= 36	(hex)
	Word	W8	= ?	

Line	Instruction		Accumulator Contents			Operand Contents	
		Bit	31 . . . 15	7	0	15 8	7 0
			... x x x x x x x x x x	x	x x x x x x x		
1	L B6		... 0 0 0 0 0 0 0 0 0 0	0	0 1 1 0 1 1 0		00110110
2	XO B5		... 0 0 0 0 0 0 0 0 0 0	0	0 0 1 1 1 0 0		00101010
3	= W8		... 0 0 0 0 0 0 0 0 0 0	0	0 0 1 1 1 0 0	00000000	00011100

Line 1: The contents of Byte B6 are loaded into the Accumulator.
Line 2: The contents of the Word Accumulator and Byte B5 are gated with EXCLUSIVE OR.
Line 3: The gating result is assigned to Word W8.

5.6.23 EXCLUSIVE OR NOT (XON)

Abbreviation for the PLC Editor: XON (EXCLUSIVE OR NOT)

Logic execution with the EXCLUSIVE OR NOT command
Operands: M, I, O, T, C

Operation:
This command functions in different ways according to its position in the program:
a) At the start of a logic chain this command functions as a LN command, i.e. the complement of the operand is loaded into the Logic Accumulator. A logic chain, however, should always be started with a load command (see L, LN, L–).
b) Within a logic chain the contents of the Logic Accumulator and the logic state of the operand (M, I, O, T, C) are gated with EXCLUSIVE OR NOT. The result of the operation is stored in the Logic Accumulator.

Example:
Input I4 and Marker M500 are to be gated with EXCLUSIVE OR NOT and the result assigned to Output O2.

Initial state:			
	Input	I4	= 0
	Marker	M500	= 0
	Output	O2	= ?

Line	Instruction		Accumulator Contents	Operand Contents
		Bit	31 . . . 7 0	
			... x x x x x x X x x x x x x x	
1	L M500		... x x x x x x 0 x x x x x x x	0
2	XON I4		... x x x x x x 1 x x x x x x x	0
3	= O2		... x x x x x x 1 x x x x x x x	1

Line 1: The operand contents are loaded into the Accumulator.
Line 2: The contents of the Logic Accumulator and the Input I4 are gated with EXCLUSIVE OR NOT.
Line 3: The gating result is assigned to Output O2.

Word execution with the EXCLUSIVE OR NOT command

Operands: B, W, D, K

Operation:
The contents of the Word Accumulator and the contents of the operand (B, W, D, K) are gated 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 in the Accumulator is gated with bit 0 in the operand
Bit 1 in the Accumulator is gated with bit 1 in the operand and so on.
The result of the operation is stored in the Logic Accumulator.

Example:
The contents of Word W4 and Word W6 are to be gated with EXCLUSIVE OR NOT and the result assigned to Word W8.

Initial state:	Word W4 = 36 AA	(hex)
	Word W6 = 3C 36	(hex)
	Word W8 = ?	

Line	Instruction	Accumulator Contents			Operand Contents	
	Bit	31 . . . 15	7	0	15 8	7 0
		... x x x x x x x x x x	x	x x x x x x x		
1	L W6	... 0 0 0 0 1 1 1 1 0 0	0	0 1 1 0 1 1 0	00111100	00110110
2	XON W4	... 1 1 1 1 1 1 0 1 0 1	0	1 1 0 0 0 1 1	00110110	10101010
3	= W8	... 1 1 1 1 1 1 0 1 0 1	0	1 1 0 0 0 1 1	11110101	01100011

Line 1: The contents of Word W6 are loaded into the Accumulator.
Line 2: The contents of the Word Accumulator and Word W4 are gated with EXCLUSIVE OR NOT.
Line 3: The gating result is assigned to Word W8.

5.6.24 ADDITION (+)

Abbreviation for the PLC-Editor: + (PLUS)

Operands: B, W, D, K

Operation:
With arithmetic functions the operand is first expanded to the size of the Accumulator (32 bits). Then the contents of the operand are added to the Word Accumulator. The result of the operation is stored in the Word Accumulator and may be processed further.

Example:
A constant and a stored value in Word W6 are to be added. The result is then stored in Doubleword D8.

Initial state:	Constant	= 100 000	(dec)
	Word W6	= 200	(dec)
	Doubleword D8	= ?	

In the interests of clarity the contents of the Accumulator and operand are shown in decimal notation.
The 10 bit wide Accumulator allows the entry of the highest possible Accumulator contents (2 147 483 647).

Line	Instruction	Accumulator Contents	Operand Contents
		x x x x x x x x x x	
1	L K100000	1 0 0 0 0 0	
2	+ W6	1 0 0 2 0 0	2 0 0
3	= D8	1 0 0 2 0 0	1 0 0 2 0 0

Line 1: The Constant is loaded into the Accumulator.
Line 2: The contents of the Accumulator and Word W6 are added.
Line 3: The result is assigned to Doubleword D8.

5.6.25 SUBTRACTION (–)

Abbreviation for the PLC-Editor: – (MINUS)

Operands: B, W, D, K

Operation:
With arithmetic functions the operand is first expanded to the size of the Accumulator (32 bits). Then the contents of the operand are subtracted from the contents of the Word Accumulator. The result of the operation is stored in the Word Accumulator and may be processed further.

Example:
A stored value in Word W6 is to be subtracted from a Constant. The result is then stored in Doubleword D8.

Initial state:				
	Constant		= 100 000	(dec)
	Word	W6	= 200	(dec)
	Doubleword	D8	= ?	

In the interests of clarity the contents of the Accumulator and the operand are shown in decimal notation. The 10 bit wide Accumulator allows the entry of the highest possible Accumulator contents (2 147 483 647).

Line	Instruction	Accumulator Contents	Operand Contents
		x xxx xxx xxx	
1	L K100000	100 000	
2	– W6	99 800	200
3	= D8	99 800	99 800

Line 1: The Constant is loaded into the Accumulator.
Line 2: The contents of Word W6 are subtracted from the Accumulator.
Line 3: The result is assigned to Doubleword D8.

5.6.26 MULTIPLICATION (x)

Abbreviation for the PLC-Editor: x (MULTIPLY)

	Logic	Byte/Word	Double	Constant
Execution time [µs]	----	3.5 to 4.3	3.2 to 3.8	3.0 to 3.8
Number of bytes	----	14	10	14

Operands: B, W, D, K

Operation:
With arithmetic functions the operand is firstly expanded to the size of the Accumulator (32 bits). Then the contents of the operand are multiplied with the contents of the Word Accumulator. The result of the operation is stored in the Word Accumulator and may be processed further. If the multiplication is not correctly executed the Marker M4200 is set, otherwise it is reset.

Example:
A Constant and a value stored in Word W6 are to be multiplied. The result is then stored in Doubleword D8.

Initial state	Constant		= 100	(dec)
	Word	W6	= 20	(dec)
	Doubleword	D8	= ?	

In the interests of clarity the contents of the Accumulator and the operand are shown in decimal notation. The 10 bit wide Accumulator allows the entry of the highest possible Accumulator contents (2 147 483 647).

Line	Instruction	Accumulator Contents	Operand Contents
		x x x x x x x x x x	
1	L K100	1 0 0	
2	x W6	2 0 0 0	2 0
3	= D8	2 0 0 0	2 0 0 0

Line 1: The Constant is loaded into the Accumulator.
Line 2: The contents of the Accumulator are multiplied by the contents of Word W6.
Line 3: The result is assigned to Doubleword D8.

5.6.27 DIVISION (/)

Abbreviation for the PLC-Editor: / (DIVIDE)

Operands: B, W, D, K

Operation:
With arithmetic functions the operand is firstly expanded to the size of the Accumulator (32 bits) . Then the contents of the Word Accumulator are divided by the contents of the operand. The result of the operation is stored in the Word Accumulator and may be processed further. If the division is not correctly executed the Marker M4201 is set, otherwise it is reset.

Example:
A Constant is to be divided by the value stored in Word W6 . The result is then assigned to Doubleword D8.

Initial state:	Constant		= 100	(dec)
	Word	W6	= 20	(dec)
	Doubleword	D8	= ?	

In the interests of clarity the contents of the Accumulator and the operand are shown in decimal notation. The 10 bit wide Accumulator allows the entry of the highest possible Accumulator contents (2 147 483 647).

Line	Instruction	Accumulator Contents	Operand Contents
		x xxx xxx xxx	
1	L K100	100	
2	/ W6	5	20
3	= D8	5	5

Line 1: The Constant is loaded into the Accumulator.
Line 2: The contents of the Accumulator are divided by the contents of Word W6.
Line 3: The result is assigned to Doubleword D8.

5.6.28 REMAINDER (MOD)

Abbreviation for the PLC-Editor: MOD (MODULO)

Operands: B, W, D, K

Operation:
With arithmetic functions the operand is firstly expanded to the size of the Accumulator (32 bits). Then the REMAINDER is determined from a division of the contents of the Word Accumulator by the contents of the operand. The REMAINDER is stored in the Word Accumulator and may be processed further. If the MOD command is not correctly executed then the Marker M4202 is set, otherwise it is reset.

Example:
The REMAINDER of a division of the value stored in Word W6 by a constant is to be determined. The REMAINDER is then stored in Doubleword D8.

Initial state:	Word	W6	= 50	(dec)
	Constant	K	= 15	(dec)
	Doubleword	D8	= ?	

In the interests of clarity the contents of the Accumulator and the operand are shown in decimal notation. The 10 bit wide Accumulator allows the entry of the highest possible Accumulator contents (2 147 483 647).

Line	Instruction	Accumulator Contents	Operand Contents
		x xxx xxx xxx	
1	L W6	50	
2	MOD K15	5	50
3	= D8	5	5

Line 1: The contents of Word W6 are loaded into the Accumulator.
Line 2: The contents of the Accumulator are divided by the constant and the integer REMAINDER is left in the Accumulator.
Line 3: The REMAINDER is assigned to Doubleword D8.

5.6.29 INCREMENT (INC)

INCREMENT Operand
Abbreviation for the PLC Editor: INC
Operands: B, W, D

Operation:
The contents of the addressed operand increases by one.

INCREMENT Word Accumulator
Abbreviation for the PLC Editor: INC W

Operation:
The contents of the word accumulator increases by one.

INCREMENT Index Register
Abbreviation for the PLC Editor: INC X

Operation:
The contents of the index register increases by one.

5.6.30 DECREMENT (DEC)

DECREMENT Operand
Abbreviation for the PLC Editor: DEC
Operands: B, W, D

Operation:
The contents of the addressed operand decreases by one.

DECREMENT Word Accumulator
Abbreviation for the PLC Editor: DEC W

Operation:
The contents of the word accumulator decreases by one.

DECREMENT Index Register
Abbreviation for the PLC Editor: DEC X

Operation:
The contents of the index register decreases by one.

5.6.31 EQUAL TO (==)

Abbreviation for the PLC-Editor: == (EQUAL)

Operands: B, W, D, K

Operation:
With this command, a direct transfer from Word to Logic processing occurs. The contents of the Word Accumulator and the contents of the addressed operand are compared. If the Word Accumulator and the operand are equal, the condition is true and the Logic Accumulator is set 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, i.e. B = 8 bit, W = 16 bit and D = K = 32 bit.

Example:
A constant is to be compared with the contents of Doubleword D8. The result is then assigned to Marker M500.

Initial state:	Constant	= 16 000
	Doubleword D8	= 15 000

The Accumulator and operand contents are shown in decimal notation. The 10 bit wide Accumulator allows the entry of the highest possible Accumulator contents (2 147 483 647).

Line	Instruction	Accumulator Contents	Operand Contents
		x xxx xxx xxx	
1	L K16000	16 000	
	Bit	31 . . . 7 0	
2	== D8	... x x x x x x 0 x x x x x x x	15 000
3	= M500	... x x x x x x 0 x x x x x x x	0

Line 1: The constant is loaded into the Accumulator.
Line 2: The contents of the Accumulator and the Doubleword D8 are compared (Accumulator = Operand ?). Because the condition is not fulfilled the Logic Accumulator is set to 0.
Line 3: The contents of the Logic Accumulator (The result of the comparison) are assigned to Marker M500.

5.6.32 LESS THAN (<)

Abbreviation for the PLC-Editor: < (LESS THAN)

Operands: B, W, D, K

Operation:
With this command, a direct transfer from Word to Logic processing occurs. The contents of the Word Accumulator are compared with the contents of the addressed operand. If the Word Accumulator is smaller than the operand, the condition is true and the Logic Accumulator is set to 1. If the Word Accumulator is smaller or equal to the operand, the Logic Accumulator is set to 0. The comparison takes place over the number of bits in the operand, i.e. B = 8 bit, W = 16 bit and D = K = 32 bit.

Example:
A constant is to be compared with the contents of Doubleword D8. The result is then assigned to Marker M500.

Initial state:	Constant	= 16 000
	Doubleword D8	= 15 000

The Accumulator and operand contents are shown in decimal notation. The 10 bit wide Accumulator allows the entry of the highest possible Accumulator contents (2 147 483 647).

Line	Instruction	Accumulator Contents			Operand Contents
		x xxx xxx xxx			
1	L K16000	16 000			
	Bit	31 . . .	7	0	
1	< D8	... x x x x x x	0	x x x x x x x	15 000
2	= M500	... x x x x x x	0	x x x x x x x	0

Line 1: The constant is loaded into the Accumulator

Line 2: The contents of the Accumulator and the Operand are compared (Accumulator < Operand ?). Because the condition is not fulfilled the Logic Accumulator is set to 0.

Line 3: The contents of the Logic Accumulator (The result of the comparison) are assigned to Marker M500.

5.6.33 GREATER THAN (>)

Abbreviation for the PLC-Editor: > (GREATER THAN)

Operands: B, W, D, K

Operation:
With this command, a direct transfer from Word to Logic processing occurs. The contents of the Word Accumulator are compared with the contents of the addressed operand. If the Word Accumulator is greater than the operand, the condition is true and the Logic Accumulator is set to 1. If the Word Accumulator is less than or equal to the operand, the Logic Accumulator is set to 0. The comparison takes place over the number of bits in the operand, i.e. B = 8 bit, W = 16 bit and D = K = 32 bit.

Example:
A constant is to be compared with the contents of Doubleword D8. The result is then assigned to Marker M500.

Initial state:	Constant	= 16 000
	Doubleword D8	= 15 000

The Accumulator and operand contents are shown in decimal notation. The 10 bit wide Accumulator allows the entry of the highest possible Accumulator contents (2 147 483 647).

Line	Instruction	Accumulator Contents	Operand Contents
		x x x x x x x x x x	
1	L K16000	1 6 0 0 0	
	Bit	31 . . . 7 0	
1	> D8	... x x x x x x | 1 | x x x x x x x	1 5 0 0 0
2	= M500	... x x x x x x | 1 | x x x x x x x	1

Line 1: The constant is loaded into the Accumulator

Line 2: The contents of the Accumulator and the Operand are compared (Accumulator > Operand ?).
Because this condition is fulfilled **the Logic Accumulator is set to 1.**

Line 3: The contents of the Logic Accumulator (The result of the comparison) are assigned to Marker M500.

5.6.34 LESS THAN OR EQUAL TO (<=)

Abbreviation for the PLC-Editor: <= (LESS EQUAL)

Operands: B, W, D, K

Operation:
With this command, a direct transfer from Word to Logic processing occurs. The contents of the Word Accumulator are compared with the contents of the addressed operand. If the Word Accumulator is less than or equal to the operand, the condition is true and the Logic Accumulator is set to 1. If the Word Accumulator is greater than the operand, the Logic Accumulator is set to 0. The comparison takes place over the number of bits in the operand, i.e. B = 8 bit, W = 16 bit and D = K = 32 bit.

Example:
A constant is to be compared with the contents of Doubleword D8. The result is then assigned to Marker M500.

Initial state:	Constant	= 16 000
	Doubleword D8	= 15 000

The Accumulator and operand contents are shown in decimal notation. The 10 bit wide Accumulator allows the entry of the highest possible Accumulator contents (2 147 483 647).

Line	Instruction	Accumulator Contents	Operand Contents
		x x x x x x x x x x	
1	L K16000	1 6 0 0 0	
	Bit	31 . . . 7 0	
1	<= D8	... x x x x x x 0 x x x x x x x	1 5 0 0 0
2	= M500	... x x x x x x 0 x x x x x x x	0

Line 1: The constant is loaded into the Accumulator.
Line 2: The contents of the Accumulator and the Operand are compared (Accumulator <= Operand). Because this condition is not fulfilled the Logic Accumulator is set to 0.
Line 3: The contents of the Logic Accumulator (The result of the comparison) are assigned to Marker M500.

5.6.35 GREATER THAN OR EQUAL TO (>=)

Abbreviation for PLC Editor: >= (GREATER EQUAL)

Operands: B, W, D, K

Operation:
With this command, a direct transfer from Word to Logic execution occurs. The content of the Word Accumulator is compared 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 Logic Accumulator is set to 1. If the Word Accumulator is smaller than the operand, the Logic Accumulator is set to 0. The comparison takes place over the number of bits corresponding to the operand i.e. B=8 bit, W=16 bit and D=K=32 bit.

Example:
A constant is to be compared with the content of Doubleword D8. The result is then assigned to marker M500.

Initial state:		
	Constant	= 16 000
	Doubleword D8	= 15 000

Accumulator and operand contents are entered here in decimal notation. The ten-position Accumulator thus permits the maximum possible Accumulator content (2 147 483 647).

Line	Instruction	Accumulator Content	Operand Content
		x x x x x x x x x x	
1	L K16000	1 6 0 0 0	
	Bit	31 . . . 7 0	
1	>= D8	... x x x x x x 1 x x x x x x x	1 5 0 0 0
2	= M500	... x x x x x x 1 x x x x x x x	1

Line 1: The constant is loaded into the Word Accumulator.
Line 2: The contents of the Word Accumulator and operand are compared according to the following criteria: Word Accumulator >= Operand. Because this condition is fulfilled, the Logic Accumulator is set to 1.
Line 3: The content of the Logic Accumulator (result of the comparison) is assigned to marker M500.

5.6.36 UNEQUAL (<>)

Abbreviation for PLC Editor: <> (NOT EQUAL)

Operands: B, W, D, K

Operation:
With this command, a direct transfer from Word to Logic execution occurs. The content of the Word Accumulator is compared with the content of the addressed operand. If the Word Accumulator and the operand are not equal, the condition is true and the Logic Accumulator is set to 1. If the Word Accumulator is equal to the operand, the Logic Accumulator is set to 0. The comparison takes place over the number of bits corresponding to the operand i.e. B=8 bit,W=16 bit and D=K=32 bit.

Example:
A constant is to be compared with the contents of Doubleword D8. The result is then assigned to marker M500.

Initial state	Constant	= 16 000
	Doubleword D8	= 15 000

Accumulator and operand contents are entered here in decimal notation. The ten position Accumulator thus permits the maximum possible Accumulator content (2 147 483 647).

Line	Instruction		Accumulator Content	Operand Content
			x xxx xxx xxx	
1	L K16000		16 000	
		Bit	31 . . . 7 0	
1	<> D8		... x x x x x x 1 x x x x x x x	15 000
2	= M500		... x x x x x x 1 x x x x x x x	1

Line 1: The constant is loaded into the Word Accumulator.
Line 2: Contents of the Word Accumulator and operand are compared according to the following criteria: Word Accumulator <> Operand. If this condition is fulfilled, the Logic Accumulator is set to 1.
Line 3: The contents of the Logic Accumulator [result of the comparison] is assigned to marker M500.

5.6.37 AND [ ] (A[ ])

Abbreviation for PLC Editor: A[ ] (AND [ ])

Operands: none

5.6.38 AND NOT [ ] (AN[ ]

Abbreviation for PLC Editor: AN[ ] (AND NOT [ ])

Operands: none

5.6.39 OR [ ] (O[ ])

Abbreviation for PLC Editor: O[ ] (OR [ ])

Operands: none

5.6.40 OR NOT [ ] (ON[ ])

Abbreviation for PLC Editor: ON[ ] (OR NOT [ ])

Operands: none

5.6.41 EXCLUSIVE OR [ ] (XO[ ])

Abbreviation for PLC Editor: XO[ ] (EXCL: OR [ ])

Operands: none

5.6.42 EXCLUSIVE OR NOT [ ] (XON[ ])

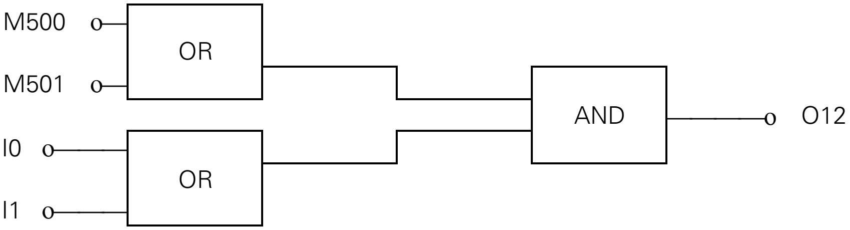
Abbreviation for PLC Editor: XON[ ] (EXCL: OR NOT [ ])

Operands: none

Function of Parentheses with Logic Commands:
The execution sequence in a ladder may be altered by the use of parentheses. The "open-parentheses" command loads the contents of the Accumulator onto the Program Stack. If the Logic Accumulator is addressed in the previous command, prior to a "parentheses-open" instruction, the content of the Logic Accumulator is loaded into the Program Stack. By addressing the Word Accumulator, the content of the Word Accumulator will be distributed.

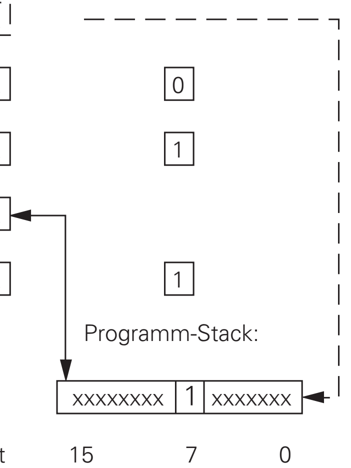
The "close-parentheses" instruction initiates the gating of the buffered value from the Program Stack with the Logic Accumulator and/or the Word Accumulator, depending on which Accumulator was addressed prior to the "parentheses-open" instruction. The result is then available in the corresponding Accumulator. The maximum nesting level is 16 pairs of parentheses.

Examples for the commands AND [ ], AND NOT [ ], OR [ ], OR NOT [ ], EXCLUSIVE OR [ ], EXCLUSIVE OR NOT [ ]. With the use of parentheses, an instruction listing may be developed according to the following logic block-diagram.



Initial state:	Marker M500 = 0Input I0 = 0 Marker M501 = 1Input I1 = 1	Output O12 = ?

Line	Instruction	Accumulator Contents	Operand Contents
	Bit 31 … 7 … 0	... x x x x x x X x x x x x x x	
1	L M500	... x x x x x x 0 x x x x x x x	0
2	O M501	... x x x x x x 1 x x x x x x x	1
3	A [	... x x x x x x 1 x x x x x x x	
4	L I0	... x x x x x x 0 x x x x x x x	0
5	O I1	... x x x x x x 1 x x x x x x x	1
6	]	... x x x x x x 1 x x x x x x x	
7	= O12	... x x x x x x 1 x x x x x x x	1



Line 1: Marker state M500 is loaded into the Logic Accumulator.
Line 2: The Logic Accumulator is gated with Marker M501.
Line 3: Open parentheses: the Accumulator contents are buffered on the Program Stack.
Line 4: Input state I0 is loaded into the Logic Accumulator.
Line 5: The Logic Accumulator is gated with Input I1.
Line 6: Close parentheses: Accumulator content is gated with the content of the Program Stack, according to the command (A[, O[, NO[...).
Line 7: The result of the complete logical process is assigned to Output O12.

Note:
The functional sequence is in principle the same for word execution, with the exception that the whole Accumulator is written onto the Stack.

5.6.43 ADD [ ] (+[ ])

Abbreviation for PLC Editor: + [ ] (PLUS [ ])

Operands: none

5.6.44 SUBTRACT [ ] (–[ ])

Abbreviation for PLC Editor: – [ ] (MINUS [ ])

Operands: none

5.6.45 MULTIPLICATION [ ] (x[ ])

Abbreviation for PLC Editor: x [ ] (MULTIPLY [ ])

Operands: none

Marker M4200 is set if an error occurs.

5.6.46 DIVISION [ ] (/[ ])

Abbreviation for PLC Editor: / [ ] (DIVIDE [ ])

Operands: none

Marker M4201 is set if an error occurs.

5.6.47 REMAINDER [ ] (MOD[ ])

Abbreviation for PLC Editor: MOD [ ] (MODULO [ ])

Operands: none

Function of Parentheses with Arithmetic Commands:
With arithmetic commands, only word execution comes into question. The execution sequence in a ladder may be altered by the use of parentheses. The "open-parentheses" command loads the content of the Word Accumulator onto the Program Stack. Then the Accumulator is available for the calculation of intermediate results. The "close-parentheses" instruction initiates the gating of the buffered value from the Program Stack with the content of the Word Accumulator. The result is again loaded into the Accumulator. The maximum nesting level is 16 pairs of parentheses.

Marker M4201 is set if an error occurs.

Example for the commands ADD [ ], SUBTRACT [ ], MULTIPLY [ ], DIVIDE [ ], DIVISION REMAINDER [ ]
The following example demonstrates how parentheses influence the result of the operation.

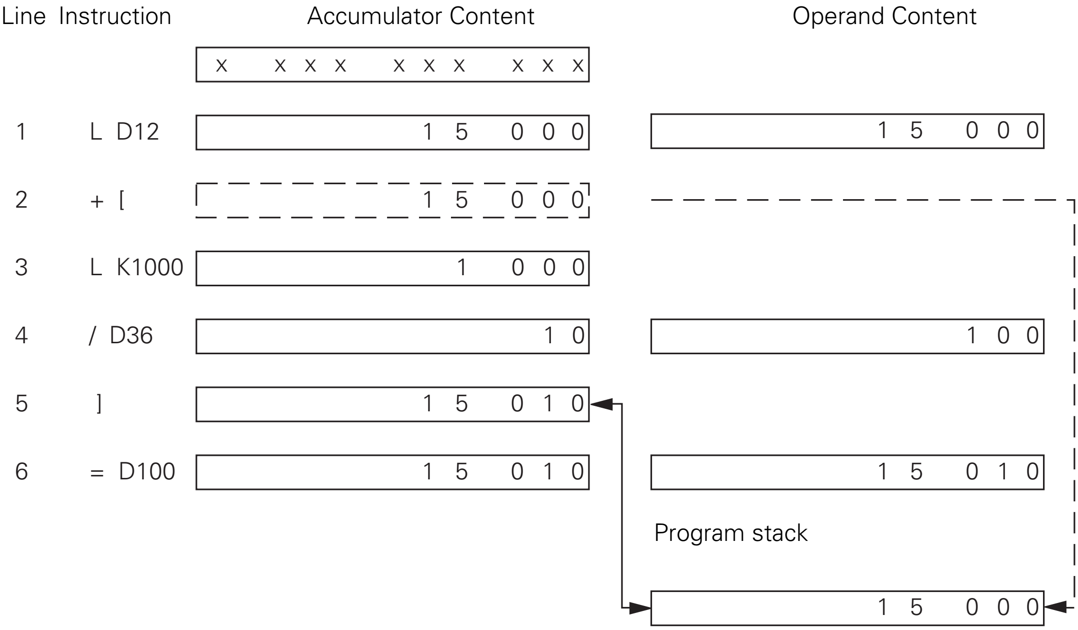
Initial state:	Constant	= 1000		(decimal)
	DoublewordD12	= 15000	(decimal)	
	DoublewordD36	= 100	(decimal)	
	DoublewordD100	= ?		

The specification of Accumulator and operand contents is given in decimal notation**.** The ten-place Accumulator thus permits the maximum possible Accumulator content of (2 147 483 647).

Command sequence without parentheses:

Line	Instruction	Accumulator Content	Operand Content
		x xxx xxx xxx	
1	L D12	15 000	15 000
2	+ K1000	16 000	
3	/ D36	160	100
4	= D100	160	100

Command sequence with parentheses:



Line	Instruction	Accumulator Content	Operand Content
		x xxx xxx xxx	
1	L D12	15 000	15 000
2	+ [	15 000	
3	L K1000	1 000	
4	/ D36	10	100
5	]	15 010	
6	= D100	15 010	15 010

Line 1: The content of Doubleword D12 is loaded into the Word Accumulator.
Line 2: Open parentheses: buffer the Accumulator content in the Program Stack.
Line 3: A constant is loaded into the Word Accumulator.
Line 4: The content of the Word Accumulator is divided by the content of Doubleword D12.
Line 5: Close parentheses: Accumulator content is gated, corresponding to the command (+[, -[, x[...) with the content of the Program Stack.
Line 6: The result of the complete logical process is assigned to Doubleword D100.

5.6.48 EQUAL TO [ ] (==[ ])

Abbreviation for PLC Editor: == [ ] (EQUAL [ ])

Operands: none

5.6.49 LESS THAN [ ] (<[ ])

Abbreviation for PLC Editor: < [ ] (LESS THAN [ ])

Operands: none

5.6.50 GREATER THAN [ ] (>[ ])

Abbreviation for PLC Editor: > [ ] (GREATER THAN [ ])

Operands: none

5.6.51 LESS THAN OR EQUAL TO [ ] (<=[ ])

Abbreviation for PLC Editor: <= [ ] (LESS EQUAL [ ])

Operands: none

5.6.52 GREATER THAN OR EQUAL TO [ ] (>=[ ])

Abbreviation for PLC Editor: >= [ ] (GREATER EQUAL [ ])

Operands: none

5.6.53 NOT EQUAL TO [ ] (<>[ ])

Abbreviation for PLC Editor: <> [ ] (NOT EQUAL [ ])

Operands: none

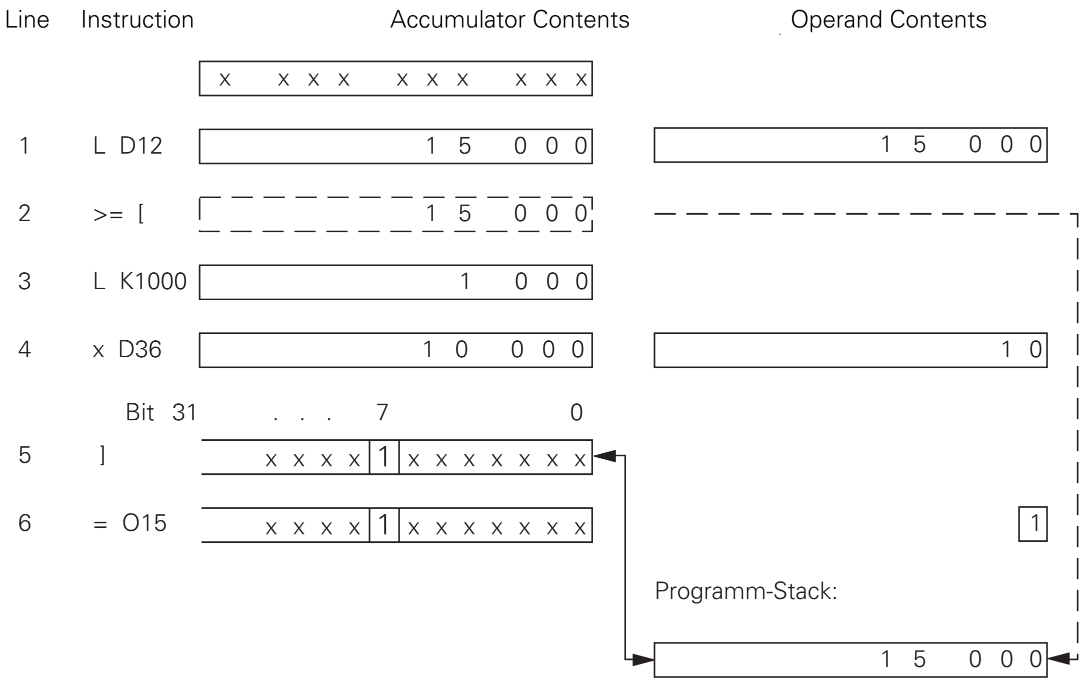
Function of parentheses with comparison commands:
The execution sequence in a ladder may be altered by the use of parentheses. The "open-parentheses" command loads the contents of the Word Accumulator onto the Program Stack. The Accumulator is now available for the calculation of intermediate results.

The "close-parentheses" instruction initiates the gating of the buffered value from the Program Stack with the content of the complete Word Accumulator. The result is loaded again into the Accumulator. The maximum nesting depth is 16 parentheses.
A direct transition from Word to Logic execution takes place with comparison commands. If the comparison condition is "true", the Logic Accumulator is set to "1". If the condition is not fulfilled, the Logic Accumulator is set to "0".

Example:

Initial state:	Constant	= 1000	(decimal)
	Doubleword D12	= 15000	(decimal)
	Doubleword D36	= 10	(decimal)
	Output O15	= ?	

The Accumulator contents and operand contents are shown in decimal notation. The ten-position Accumulator thus permits the maximum possible Accumulator content of 2 147 483 647.
The Accumulator is again represented in binary notation after program line 5, because the transition to logic execution occurs here.



Line 1: The content of Doubleword D12 is loaded into the Word Accumulator.
Line 2: Open parentheses: buffering of the Accumulator content in the Program Stack.
Line 3: Loading of a Constant into the Word Accumulator.
Line 4: The content of the Word Accumulator is multiplied by the content of Doubleword D12.
Line 5: Close parentheses: Word Accumulator content is gated, corresponding to the command(==[, >=[, <=[...) with the content of the Program Stack . The transition from Word to Logic processing occurs in this program line. The Logic Accumulator is set or reset, depending on the result of the comparison.
Line 6: The result of the complete logical process is assigned to output O15.

5.6.54 SHIFT LEFT (<<)

Abbreviation for PLC Editor: << (SHIFT LEFT)

Operands: B, W, D, K

Operation:
Since the sign bit (MSB) is included with this command, it is grouped in with arithmetic commands. For this reason and out of time considerations, this command should not be used for the isolation of bits. A SHIFT LEFT instruction causes the contents of the Word Accumulator to be multiplied by two. For this purpose, the bits in the Accumulator are simply shifted 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. The number of shift events is defined by the operand. The Accumulator is filled on the right side with zeros.

Example:
The content of the Doubleword D8 is to be shifted four times to the left and then assigned to D12.

Initial state: Doubleword D8 = 3E 80 (hex)
Doubleword D12 = ?

The Accumulator content is shown here in binary notation, and the operand content in hexadecimal notation.

Line	Instruction	Accumulator Content				Operand Content			
		xxxxxxxx	xxxxxxxx	xxxxxxxxx	xxxxxxxxx				
1	L D8	00000000	00000000	00111110	10000000	00	00	3E	80
2	<< K+1	00000000	00000000	01111101	00000000				
3	<< K+1	00000000	00000000	11111010	00000000				
4	<< K+1	00000000	00000001	11110100	00000000				
5	<< K+1	00000000	00000011	11101000	00000000				
6	= D12	00000000	00000011	11101000	00000000	00	03	E8	00

Line 1: Load Doubleword D8 into the Accumulator.
Line 2 to 5: The content of the Word Accumulator is shifted to the left by the number of bits specified in the operand. The complete operation can also be undertaken with the command << K+4.
Line 6: The result is stored in the Doubleword D12.

5.6.55 SHIFT RIGHT (>>)

Abbreviation for PLC Editor: >> (SHIFT RIGHT)

Operands: B, W, D, K

Operation:
Since the sign bit (MSB) is included with this command, it is grouped in with arithmetic commands. For this reason and out of time considerations, this command should not be used for the isolation of bits. A SHIFT RIGHT instruction causes the contents of the Word Accumulator to be divided by two. For this purpose, the bits in the Accumulator are simply shifted by one place to the right. The number of the shift operations is determined via the operand. Thus the set bits, which are shifted beyond the Accumulator to the right, are lost; the Accumulator is filled according to the sign, from the left-hand side. With operand contents greater than 32, the operand value Modulo 32 is used, i.e. the integer remainder from the division (operand value)/32.

Example:
The content of the Doubleword D8 is to be shifted four times to the right and then stored in D12.

Initial state: Doubleword D8 = 3E 80 (hex)
Doubleword D12 = ?

The Accumulator content is shown here in binary notation and the operand content in hexadecimal notation.

Line	Instruction	Accumulator Content	Operand Content
		xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx	
1	L D8	00000000 00000000 00111110 10000000	00 00 3E 80
2	>> K1	00000000 00000000 00011111 01000000	
3	>> K1	00000000 00000000 00001111 10100000	
4	>> K1	00000000 00000000 00000111 11010000	
5	>> K1	00000000 00000000 00000011 11101000	
6	= D12	00000000 00000000 00000011 11101000	00 00 03 E8

Line 1: Load Doubleword D8 into the Accumulator.
Line 2 to 5: The content of the Word Accumulator is shifted to the right by the number of bits specified in the operand. The complete operation can also be undertaken with the command >> K+4.
Line 6: The result is stored in Doubleword D12.

5.6.56 BIT SET (BS)

Abbreviation for PLC Editor: BS (BIT SET)

Operands: B, W, D, K, X

Operation:
With this command, each bit in the Accumulator can be acted on. The addressed bit is set to "1" through the use of the BS command. The selection (addressing) of the corresponding bit is derived from the content of the specified Operand or a Constant. In the bit-numbering, bit 0 corresponds to the LSB and bit 31 the MSB. For operand contents larger than 32, the operand value Modulo 32 is used, i.e. the integer remainder from the division (operand value)/32.

Example:
Load Doubleword D8 in the Accumulator, set the bit 0 of the Accumulator to "1" and store the result in Doubleword D12.

Initial state:	Doubleword D8 = 3E 80	(hex)
	Doubleword D12 = ?	

Accumulator and operand contents are shown here in hexadecimal notation.

Line	Instruction	Accumulator Content	Operand Content
		xx xx xx xx	
1	L D8	00 00 3E 80	00 00 3E 80
2	BS K+0	00 00 3E 81	
3	= D12	00 00 3E 81	00 00 3E 81

Line 1: Load Doubleword D8 into the Accumulator.
Line 2: The bit specified in the operand is set to 1.
Line 3: The result is stored in Doubleword D12.

5.6.57 BIT RESET (BC)

Abbreviation for PLC Editor: BC (BIT CLEAR)

Operands: B, W, D, K, X

Operation:
With this command, each bit in the Accumulator can be acted on. The addressed bit is set to "0" through the use of the BC command. The selection (addressing) of the corresponding bit is derived from the content of the specified Operand or a Constant. In the bit-numbering, bit 0 corresponds to the LSB and bit 31 the MSB. For operand contents larger than 32, the operand value Modulo 32 is used, i.e. the integer remainder from the division (operand value)/32.

Example:
Load Doubleword D8 in the Accumulator, set bit 0 of the Accumulator to "0" and store the result in Doubleword D12.

Initial state: Doubleword D8 = 3E 81 (hex)
Doubleword D12 = ?

Accumulator and operand contents are shown here in hexadecimal notation.

Line	Instruction	Accumulator Content	Operand Content
		xx xx xx xx	
1	L D8	00 00 3E 81	00 00 3E 81
2	BC K+0	00 00 3E 80	
3	= D12	00 00 3E 80	00 00 3E 80

Line 1: Load Doubleword D8 into the Accumulator.
Line 2: The bit specified in the operand is set to "0".
Line 3: The result is stored in Doubleword D12.

5.6.58 BIT TEST (BT)

Abbreviation for PLC Editor: BT (BIT TEST)

Operands: B, W, D, K, X

Operation:
With this command, the status of each individual bit in the Accumulator may be interrogated. With BT commands, a direct transition from Word to Logic execution takes place. The BIT TEST tests the status of a bit from the Word Accumulator and then acts correspondingly on the Logic Accumulator. If the tested bit is "1", then the Logic Accumulator is also set to "1"; if it is "0" ,it is set to "0". The program continues in logic execution. The selection (addressing) of the corresponding bit is derived from the content of the specified Operand or a Constant. In the bit-numbering, bit 0 corresponds to the LSB and bit 31 the MSB. For operand contents larger than 32, the operand value Modulo 32 is used, i.e. the integer remainder from the division (operand value)/32.

Example:
Load Doubleword D8 in the Accumulator, and assign the logic state of bit 0 to an Output.

Initial state: Doubleword D8 = 3E 81 (hex)
Output O12 = ?

Word Accumulator and operand contents are shown here in hexadecimal notation, the Logic Accumulator in binary representation.

Line	Instruction	Accumulator Content	Operand Content
		xx xx xx xx	
1	L D8	00 00 3E 81	00 00 3E 81
2	BT K+0	00 00 3E 81	
3	= O12	x x x x x x | 1 | x x x x x x x	1

Line 1: Load Doubleword D8 into the Accumulator.
Line 2: The bit specified in the operand is tested as to its status.
Line 3: The Logic Accumulator is assigned to Output O12.

5.6.59 Load Data onto the Data Stack (PS)

Abbreviation for PLC Editor: PS (PUSH)

Logic Execution with the PS Command

Operands: M, I, O, T, C

Operation:
With the PS command, data can be buffered. Thus the addressed operand is loaded onto the Data Stack. Since the Data Stack is organized as 16 bit, a minimum width of one Word must be used in writing to it. During this the operand value is copied into bit 7 of the current address in the Data Stack. The free bits of the reserved memory are undefined or unused. In the event of a stack overflow, an error message will be issued.

Memory allocation in the Data Stack:

Bit 15	7	0
x x x x x x x x	L	x x x x x x x

Word Execution with the PS Command

Operands: B, W, D, K

Operation:
With the PS command, data can be buffered. Thus the addressed memory area (B, W, D, K) is copied into the current address of the Data Stack. With Word execution, two Words are reserved as standard on the Data Stack per PS command. The operand is extended in the Stack with sign justification corresponding to the MSB. In the event of a Stack overflow, an error message will be issued.

Memory allocation in the Data Stack upon saving of:

	Bit 31	15 7 0
Byte	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
Word	X X X X X X X X X X X X X X X X	WWWWWWWWWWWWWWWW
Doubleword	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
Constant	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

5.6.60 Acquire Data from the Data Stack (PL)

Abbreviation for PLC Editor: PL (PULL)

Logic Execution with the PL Command

Operands: M, I, O, T, C

Operation:
The PL command complements the PS command. Data which are saved with PUSH can be taken from the Data Stack again with PULL. With logic execution, bit 7 is copied from the current address of the Data Stack into the addressed operand with a PL command. If the Stack is empty, an error message will be issued.

Logic Execution with the PL Command

Operands: B, W, D

Operation:
The PL command complements the PS command. Data which are saved with PUSH can be taken from the Data Stack again with PULL. With Word execution, two Words are copied from the current address of the Data Stack into the addressed memory area with a PL command. If the Stack is empty, an error message will be issued.

5.6.61 Load Logic Accumulator onto the Data Stack (PSL)

Abbreviation for PLC Editor: PSL (PUSH LOGICACCU)

Execution time [µs] 0.6 to 1.0
Number of bytes 20

Operands: none

Operation:
The Logic Accumulator can be buffered with the PSL command. For this purpose, the Logic Accumulator is loaded onto the Data Stack. Since the Data Stack is organized as 16 bits, it must be written to with a minimum width of one Word. During this the content of the Logic Accumulator is copied into the current address of the Data Stack. The free bits of the reserved memory are undefined or unused. In the event of a Stack overflow, an error message will be issued.

Memory allocation in the Data Stack:

Bit 15	7	0
x x x x x x x x	L	x x x x x x x

5.6.62 Load Word Accumulator onto the Data Stack (PSW)

Abbreviation for PLC Editor: PSW (PUSH WORDACCU)

Operands: none

Operation:
The content of the Word Accumulator can be buffered with the PSW command. For this purpose, the Word Accumulator is copied into the Data Stack. The content of the Word Accumulator (32 bit) reserves two Words on the Data Stack. In the event of a stack overflow, an error message will be issued.

5.6.63 Acquire Logic Accumulator from the Data Stack (PLL)

Abbreviation for PLC Editor: PLL (PULL LOGICACCU)

Operands: none

Operation:
The PLL command complements the PSL command. With a PLL instruction, bit 7 from the current address of the Data Stack is copied into the Logic Accumulator. If the stack is empty, an error message will be issued.

5.6.64 Acquire Word Accumulator from the Data Stack (PLW)

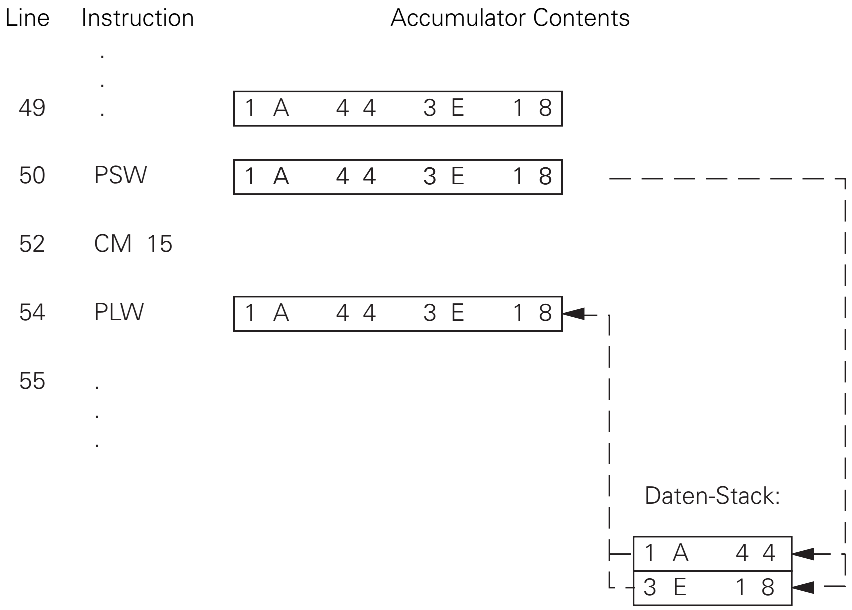
Abbreviation for PLC Editor: PLW (PULL WORDACCU)

Operands: none

Operation:
The PLW command complements the PSW command. With a PLW instruction, two Words are copied from the Data Stack into the Word Accumulator. If the stack is empty, an error message will be issued.

Examples for the commands PS, PL, PSL, PSW, PLL, PLW
The Module 15 is to be called at a specific point in the program. After the return into the main program, the original Accumulator content is again required for further program run.

Accumulator contents prior to the Call Module: 1A 44 3E 18



Line 50: Save the Word Accumulator onto the Data Stack.
Line 52: Subprogram 15 is called up.
Line 54: The original Accumulator contents are transferred back from the Data Stack and is available for further program run.

Note:
The sequence for stack operations is the same for all commands. Only the data width varies.

5.6.65 Unconditional Jump (JP)

Abbreviation for PLC Editor: JP (JUMP)

Operands: jump address (LBL)

Operation:
A JP command instructs the processor to continue the program at the specified jump address (Label). This command interrupts a logic sequence.

5.6.66 Jump if Logic Accumulator = 1 (JPT)

Abbreviation for PLC Editor: JPT (JUMP IF TRUE)

Operands: jump address (LBL)

Operation:
A JPT command is a conditional jump command. If the Logic Accumulator is "1", the program is continued from the specified jump address (Label). If the Logic Accumulator is "0" the jump is not processed. This command interrupts a logic sequence.

5.6.67 Jump if Logic Accumulator = 0 (JPF)

Abbreviation for PLC Editor: JPF (JUMP IF FALSE)

Operands: jump address (LBL)

Operation:
A JPF command is a conditional jump command. If the Logic Accumulator is "0", the program is continued from the specified jump address (Label). If the Logic Accumulator is "1", the jump is not processed. This command interrupts a logic sequence.

Example for the commands JP, JPT, JPF

A certain program section is to be skipped, depending on Input 15.

Initial state: Input I5 = 1

Line	Instruction	Accumulator Content	Operand Content
	Bit	31 . . . 7 0 ... x x x x x x | X | x x x x x x x	
1	L I5	... x x x x x x | 1 | x x x x x x x	1
2	JPT 10	... x x x x x x | 1 | x x x x x x x	
3	L I3		
4	O M500		
5	= O20		
6	LBL 10		
7	L M100	... x x x x x x | 0 | x x x x x x x	0

•
•
•

Line 1: Load the operand contents in the Accumulator.
Line 2: Dependent on Input I5, a program jump is processed.
Line 3: Skipped in this example.
Line 4: Skipped in this example.
Line 5: Skipped in this example.
Line 6: Jump address: The program run is continued from here.

5.6.68 Call Module (CM)

Abbreviation for PLC Editor: CM (CALL MODULE)

Operands: jump address (LBL)

Operation:
A Call Module instructs the processor to leave the main program and process the Module designated by the jump address (LBL). Modules are independent subprograms and are terminated by the command EM. They can also be called at multiple points in the main program. This command interrupts a logic sequence.

5.6.69 Call Module if Logic Accumulator = 1 (CMT)

Abbreviation for PLC Editor: CMT (CALL MODULE IF TRUE)

Operands: jump address (LBL)

Operation:
A CMT command is a conditional Call Module. If the Logic Accumulator is "1", the Module with the specified jump address (Label) is processed. If the Logic Accumulator is "0", the main program continues without a Call Module. This command interrupts a logic sequence.

5.6.70 Call Module if Logic Accumulator = 0 (CMF)

Abbreviation for PLC Editor: CMF (CALL MODULE IF FALSE)

Operands: jump address (LBL)

Operation:
A CMF command is a conditional Call Module. If the Logic Accumulator is "0", the Module with the specified jump address (Label) is processed. If the Logic Accumulator is "1", the main program continues without a Call Module. This command interrupts a logic sequence.

Example for the commands CM, CMT, CMF
A certain Module is to be called, depending on Input I5.

Initial state: Input I5 = 0

Line	Instruction	Accumulator Contents			Operand Contents
	Bit	31 . . .	7	0	
		... x x x x x x	X	x x x x x x x	
1	L I5	... x x x x x x	0	x x x x x x x	0
2	CMF 10	... x x x x x x	0	x x x x x x x	
3	L M100	... x x x x x x	1	x x x x x x x	1
	• • •				
499	EM				
500	LBL 10				
501	L I3	... x x x x x x	0	x x x x x x x	0
502	OM 500	... x x x x x x	1	x x x x x x x	1
503	= O20	... x x x x x x	1	x x x x x x x	1
504	EM				

Line 1: Load the operand contents in the Accumulator.
Line 2: Dependent on Input I5, the Call Module is processed.
Line 499: End Module of the main program.
Line 500: Start of the Module, identified by LBL.
Line 501: Instruction in the subprogram.
Line 502: Instruction in the subprogram.
Line 503: Instruction in the subprogram.
Line 504: End Module: Effects the return to the main program.
Line 3: The main program continues at this point once the Module is processed.

5.6.71 End of Module, Program End (EM)

Abbreviation for PLC Editor: EM (END OF MODULE)

Operands: none

Operation:
Every program and/or every subprogram (Module) is terminated with an EM command. An EM command in a Module initiates the return jump to the Call Module (CM, CMT, CMF). The program is continued with the instruction following the Call Module. The command EM is handled as program end criterion; thus subsequent program instructions can be reached using a jump address.

5.6.72 End of Module if Logic Accumulator = 1 (EMT)

Abbreviation for the PLC Editor: EMT (END OF MODULE IF TRUE)

Operation:
An EMT command only initiates a return jump to the Call Module (CM, CMT, CMF) when the Logic Accumulator is "1".

5.6.73 End of Module if Logic Accumulator = 0 (EMF)

Abbreviation for the PLC Editor: EMF (END OF MODULE IF FALSE)

Operation:
An EMF command only initiates a return jump to the Call Module (CM, CMT, CMF) when the Logic Accumulator is "0".

5.6.74 Jump Label (LBL)

Abbreviation for PLC Editor: LBL (LABEL)

Operands: ASCII name (up to 32 characters long)

Operation:
The jump label defines a program position as an entry point for the CM and JP commands. Jump labels may be allocated addresses in the range 0 to 511. Up to 1000 jump labels per file can be defined.

The ASCII name of the jump label may be up to 32 characters long, but only the first 16 characters are used to distinguish jump labels. For importing global jump labels, see EXTERN instruction.

5.7 INDEX Register

Under the control of the PLC programmer this register can be used for data transfer, intermediate storage of results and for indexed addressing of operands. The register is 32 bits wide but only the lower 16 bits are used for index addressing. The X register can be used anywhere in the program — there is no contents validity check — however there is a check for address space overflow with indexed write accesses.

Example: = B100[X]

If the address space is overshot the error message **`PLC: index range incorrect`** flashes in the display. Reset with END to display the error line in the PLC Editor.

Before using a command with the index-register it must be assigned a defined value. At the beginning of each PLC cycle the index register is set to 0.

The following operands can be addressed.

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	Contents of index register = operand
BCX	Contents of index register = operand
BSX	Contents of index register = operand
Sn[X]	String number = n+X
S#Dn[X]	Dialog text number = n+X
S#En[X]	Error text number = n+X
S#An[X]	ASCII code + X
Sn^X	Substring from X-th characters of n-th string

The types S"", K and K$ cannot be indexed.

Caution: When addressing S#Dn[X] and S#En[X] the sequence <SUB>Dnnn or <SUB>Ennn is loaded in the string accumulator, with nnn as the modified string number!

Commands for operating the Index Register:

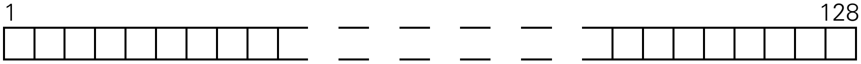
The following commands have been introduced to permit data interchange between the Word Accumulator and the Index Register or between the Stack and Index Register:

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)	

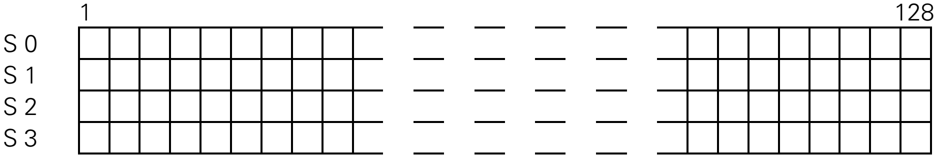
5.8 Commands for STRING Execution

STRING execution allows the creation and manipulation of any texts via the PLC program. These texts may be displayed in the PLC window of the screen by the use of Module 9082, and/or deleted again with Module 9080 (refer to PLC modules). A STRING Accumulator and four STRING memories are provided in the control for STRING execution. A maximum of 128 characters may be loaded into this.

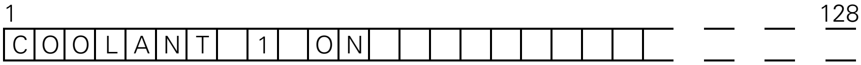
STRING accumulator: 128 Characters



STRING memory: 128 Characters



Example:



STRING Accumulator and STRING memory are volatile, and so are again deleted upon powering off. For STRING execution, the new operand "S" has been introduced. The operand "S" may be used with various arguments.

Explanation of the Operand:

The operand "S" is only used in STRING execution. The following locations may be addressed with the various arguments:

- STRING memory: Should a STRING memory be addressed, the number of the required memory (S0-S3) must be specified after the Operand-Designation.
- Part of a STRING: If only part of a STRING is to be addressed, then this can be done by addressing Sn^X (see INDEX REGISTER).
  The substring will be addressed from the X-th character in the specified STRING.
- Immediate STRING: A STRING can also be entered directly into the PLC program. The Text STRING, which may contain 0 – 37 characters, must be identified by quotation marks.
  Example: S "COOLANT 1 ON"

- Text from the PLC-Error Message files and/or from the PLC-dialog files: Text from the active error message or dialog files may be read by the input of the line number.
  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)
  The character sequence #Exx or #Dxx is entered in the Argument <arg> for the STRING-Command. A 5 Byte long character train <SUB> E0xx or <SUB> D0xx is loaded into the Accumulator (<SUB> = ASCII <SUB>). Instead of this character train, the line xx of the active error message or dialog file is read for display on the screen.
- ASCII character entered in string. The ASCII character is defined by its code S#Axxx.

5.8.1 LOAD (L)

Abbreviation for PLC Editor: L (LOAD)

Operands: S <arg>

Operation:
The STRING Accumulator is loaded with this L command. The selection of the STRINGS to be loaded, proceeds using the Argument <arg> after the operand designation.
Refer also to operand explanation.

5.8.2 ADD (+)

Abbreviation for PLC Editor: +

Operands: S <arg>

Operation:
With this command another STRING is added to the STRING in the STRING Accumulator. The selection of the STRINGS, which should be added, proceeds using the Argument <arg> after the operand designation. Refer also to operand explanation. The resultant STRING must not be longer than 128 characters.

5.8.3 Storing a STRING (=)

Abbreviation for PLC Editor: =

Operands: S <arg>

Operation:
With the = command a STRING from the STRING Accumulator is stored in a STRING memory. The selection of the memory, into which the STRING should be copied, proceeds using the Argument <arg> after the operand designation. Whereby only the Arguments 0 – 3, which address a STRING memory (S0 – S3) are valid here. Refer also to operand explanation.

5.8.4 Overwriting of a STRING (OVWR)

Abbreviation for PLC Editor: OVWR (OVERWRITE)

Operands: S <arg>

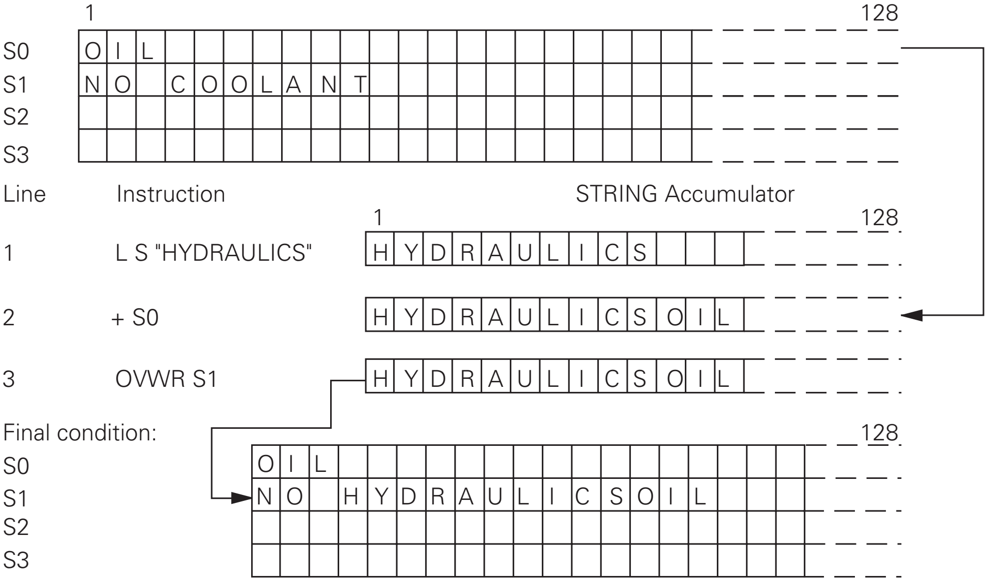
Operation:
With the OVWR command a STRING from the STRING Accumulator is stored in a STRING memory. This command functions in a similar manner to the = command, with the difference that the character "STRING-End" is not transferred alongside. By this means, the beginning of a STRING which is already in the STRING memory, can be overwritten.

The selection of the memory, into which the STRING should be copied, proceeds using the Argument <arg> after the operand designation. Whereby only the Arguments 0 – 3, which address a STRING memory (S0 – S3) are valid here.
Refer also to operand explanation.

Example of STRING execution:
A STRING from the STRING memory S0 is to be added to an Immediate STRING. This STRING must overwrite the contents STRING memory S1.

Initial condition:

Immediate STRING:	HYDRAULICS
STRING memory S0:	OIL
STRING memory S1:	NO COOLANT



	1 … 128
S0	OIL
S1	NO COOLANT
S2	
S3	

Line	Instruction	STRING Accumulator (1 … 128)
1	L S "HYDRAULICS"	HYDRAULICS
2	+ S0	HYDRAULICSOIL
3	OVWR S1	HYDRAULICSOIL

Final condition:

	1 … 128
S0	OIL
S1	NO HYDRAULICSOIL
S2	
S3	

Line 1: Load the immediate STRING into the STRING accumulator.
Line 2: The contents of the STRING memory S0 is added to the contents of the STRING accumulator.
Line 3: The STRING accumulator overwrites the contents of the STRING memory S1.

5.8.5 Logical Comparisons in STRING Execution

Two STRINGS are compared according to the argument as follows:
If STRING memory or Immediate STRING are entered in the command, both STRINGS are compared character for character. The Logic Accumulator is reset after the first character for which the comparison conditions are not fulfilled. The remaining characters are checked no further. For the purposes of comparison, the number of the character in the ASCII table is always used. This results in, for example:
A < B
AA > A

If PLC-Error messages or PLC-Dialog texts are entered, the position in the file (0 to 4095) is compared, not the actual text as with Immediate STRING.

The execution times depend on the length of the STRINGS. The quoted times represent maximum values. With the Immediate STRINGS, the length "n" of the STRINGS respectively must be added to the command length. In the event that this is odd, the next larger even-numbered length must be added.

5.8.6 EQUAL TO (==)

Abbreviation for PLC Editor: == (EQUAL)

Operation:
With this command a direct transition from STRING- to logic execution takes place. The content of the STRING Accumulator is compared with the STRING in the Argument. If the STRING Accumulator and the operand are equal, the condition is true and the Logic Accumulator is set to 1. If they are not equal the Logic Accumulator is set to 0.

5.8.7 LESS THAN (<)

Abbreviation for PLC Editor: < (LESS THAN)

Operands: S <arg>

Operation:
With this command a direct transition from STRING to Logic execution takes place. The content of the STRING Accumulator is compared with the STRING in the Argument. If the STRING Accumulator is smaller than the operand, the condition is true and the Logic Accumulator is set to 1. If the STRING Accumulator is greater than or equal to the operand the Logic Accumulator is set to 0.

5.8.8 GREATER THAN (>)

Abbreviation for PLC Editor: > (GREATER THAN)

Operands: S <arg>

Operation:
With this command a direct transition from STRING- to logic execution takes place. The content of the STRING Accumulator is compared with the STRING in the Argument. If the STRING Accumulator is greater than the operand, the condition is true and the Logic Accumulator is set to 1. If the STRING Accumulator is less than or equal to the operand the Logic Accumulator is set to 0.

5.8.9 LESS THAN OR EQUAL TO (<=)

Abbreviation for PLC Editor: <= (LESS EQUAL)

Operands: S <arg>

Operation:
With this command a direct transition from STRING- to logic execution takes place. The content of the STRING Accumulator is compared with the STRING in the Argument. If the STRING Accumulator is less than or equal to the operand, the condition is true and the Logic Accumulator is set to 1. If the STRING Accumulator is greater than the operand the Logic Accumulator is set to.

5.8.10 GREATER THAN OR EQUAL TO (>=)

Abbreviation for PLC Editor: >= (GREATER EQUAL)

Operands: S <arg>

Operation:
With this command a direct transition from STRING- to logic execution takes place. The content of the STRING Accumulator is compared with the STRING in the Argument. If the STRING Accumulator is greater than or equal to the operand, the condition is true and the Logic Accumulator is set to 1. If the STRING Accumulator is less than the operand the Logic Accumulator is set to 0.

5.8.11 UNEQUAL (<>)

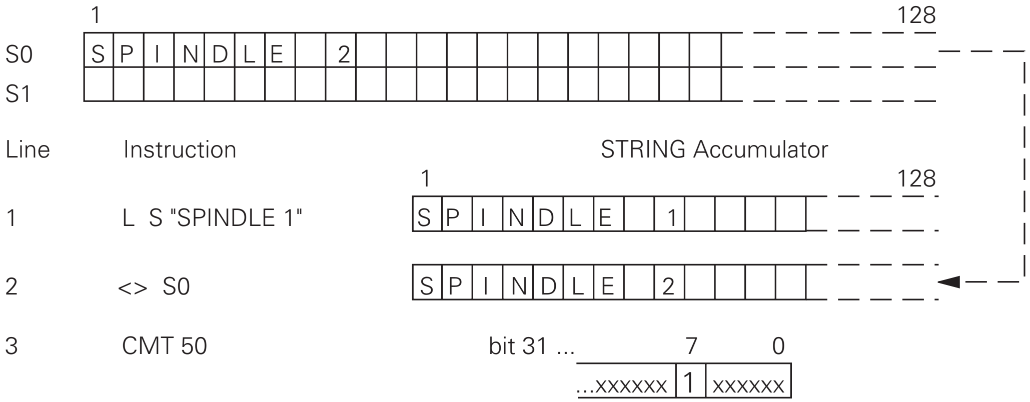
Abbreviation for PLC Editor: <> (NOT EQUAL)

Operands: S <arg>

Operation:
With this command a direct transition from STRING- to logic execution takes place. The content of the STRING Accumulator is compared with the STRING in the Argument. If the STRING Accumulator is not equal to the operand, the condition is true and the Logic Accumulator is set to 1. If the STRING Accumulator is equal to the operand the Logic Accumulator is set to 0.

Example of STRING execution:
An Immediate STRING is to be compared with the content of the STRING-Memory S0. Depending on the comparison result, Module 50 is called.

Initial state:	STRING memory S0:	SPINDLE 2
	Immediate STRING:	SPINDLE 1



```
Line    Instruction
1       L  S "SPINDLE 1"
2       <>  S0
3       CMT 50
```

Line 1: Load the Immediate STRING into the STRING Accumulator.
Line 2: The content of the STRING Accumulator is compared with the content of the STRING memory S0 according to the command.
Line 3: Since the result of the comparison is "true", the Logic Accumulator is set and the Call Module is processed.

5.8.12 Modules for String Execution

Copying a number from a string (Module 9070)
The source string in the string buffer with the given source string number is searched for a numerical value. The first numerical value found is copied into the string designated by the destination string number.

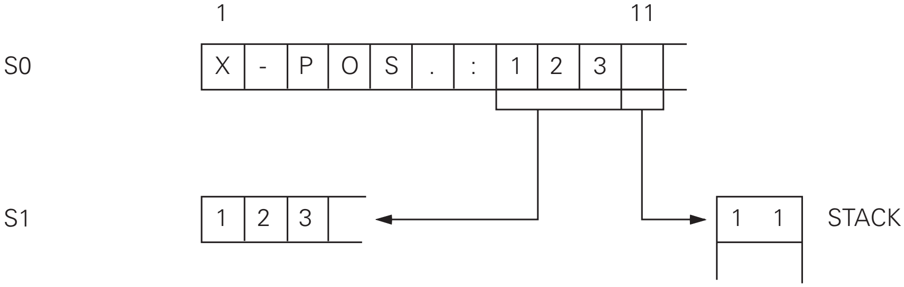
If any conflict between the source string and destination string is not examined, the source string may be overwritten. (Even if this occurs, the proper function of the module is ensured.) Numbers with and without algebraic signs are both recognized. Decimal characters can be either a point or comma. The distance (in characters) of the first character after the number found in the source string is sent back.

Possible errors:
- The numbers of the source or destination string are outside of the permissible range (0..3).
- There is no number in the source string.
- The source string was searched and no end was found.
- The numerical string has a length of more than 79 characters, which leads to an internal overflow.

Call:

```
PS   K/B/W/D     <Number of the source string>
PS   K/B/W/D     <Number of the destination string>
CM   9070
PL   B/W/D       <Offset end of number string in the source string >
```

Example:



Finding the string length (Module 9071)
The length of the string with the given number in the string buffer is calculated.

Possible errors:
- The number of the source string lies outside of the permissible range (0..3)
- The source string was searched, but no end (<NUL>) was found.

Call:

```
PS   K/B/W/D     <Number of the source string>
CM   9071
PL   B/W/D       <Length of the string>
```

5.9 Submit Programs

Submit programs are subprograms which the PLC submits to the NC for processing. This allows tasks to be performed which are very processor-intensive, require program loops, or must wait for external results. It is assumed, however, that these programs are not bound by a particular time frame. Depending on processor loading, each Submit program is allocated a certain computing power, but always at least 5% of the total power. Submit programs are started from the PLC program and can access all the same data memories (M/B/W/D) as can the main program. This can lead to problems in certain circumstances. Such problems can be avoided if the data processed by the PLC program are clearly separated from the data processed by the Submit program.

Up to eight Submit programs can be entered in a queue (Submit Queue). Each receives an "Identifier," a number between 1 and 255 assigned by the NC, which is transferred into the Word Accumulator. With this Identifier and the REPLY function, you can inquire whether or not the program is in the queue, is being processed, or is already complete. The Submit programs are executed in the order of their placement in the queue. Should an error occur during the execution of the Submit programs, the following Markers are set:

M4200	Overflow during Multiplication
M4201	Division by 0
M4202	MODULO incorrectly executed
M4203	Error status for PLC module
M4204	Reserved for errors that the PLC programmer would like to intercept

These markers are listed separately in the submit job. This means that the same markers can be edited as those in the PLC run program without changing the original markers.

Exact times cannot be given for the commands for the management of the Submit queue. The execution times denote maximum values.

5.9.1 Call up of the Submit Program (SUBM)

Abbreviation for PLC Editor: SUBM (SUBMIT)

Operands: jump address (LBL)

Operation:
The SUBM command allots an "Identifier" (1 to 255) to the subprogram, designated by the jump address (LBL). Simultaneously, the allocated number is written to the Word Accumulator. If there are already programs transferred into the Submit queue, the addressed program will not be processed until the program immediately prior to it is finished. A submission to the queue may only take place from a PLC program, a SUBM command in a Submit program is not possible.

If no location is free in the queue, or if the SUBM command is programmed in a Submit program (nesting), a "0" will be returned to the Word Accumulator.

5.9.2 Status Interrogation of a Submit Program (RPLY)

Abbreviation for PLC Editor: RPLY (REPLY)

Operands: B, W

Operation:
With the RPLY command the Status of the Submit program is interrogated with the specified Identifier. This Identifier must already be stored in a Byte or Word prior to the calling up of the Submit program. With the RPLY command and the memory address specified above, which contains the Identifier, one of the following messages about the status is transferred 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

5.9.3 Cancellation of a Submit Program (CAN)

Abbreviation for PLC Editor: CAN (CANCEL)

Operands: B, W

Operation:
With the CAN command the Submit-Program with the specified Identifier is canceled during execution or removed from the queue. This Identifier must already be stored in a Byte or Word prior to the calling up of the Submit-Program. After the cancellation of the Program, the next Submit program in the queue will immediately be processed.

The following PLC modules cannot be canceled with CANCEL at any desired point:
- PLC module for access to the screen (908X).
- PLC module for reading NC files (909X).

In these cases, the RPLY command must be used to check whether or not the CAN command may be used.

Example of the use of the SUBM command:
Dependent on Input I10 the subprogram with the Label LBL 300 is handed over to the NC for processing. In addition, the execution of the subprogram is checked in the main program with the RPLY command and canceled with the CAN command in conjunction with Input I11.

```
Line    Instruction         Program Comments:

1       L        I10        ;Interrogate state of Input I10
2       JPF      100        ;Dependent on Input I10 skip
                            ;Call Module
3       RPLY     B 128      ;Interrogate status of the Submit program
4       <>       K+0        ;Submit program already transferred to
                            ;the NC for processing ?
5       JPT      100        ;If program already transferred to the NC,
                            ;renewed program call skipped
6       SUBM     300        ;Call up Submit program
7       =        B 128      ;Store Identifier in Byte 128
8       LBL 100             ;Jump address
9       L        I 11       ;Interrogate state of Input I11
10      JPF      110        ;Dependent on Input I11, skip the deletion
                            ;of the Submit program
11      CAN      B 128      ;Interrupt execution of the Submit program
                            ;or remove program from the queue
12      LBL 110             ;Jump address
          •                 ;Continuation
          •                 ;Main program
          •                 ;
XX      EM                  ;End main program

XX      LBL      300        ;Begin Submit program (is added as with Modules
XX        •                 ;at the end of the main program)
XX        •                 ;
XX        •                 ;
XX      EM                  ;End Submit program
```

In this case, the contents of the Submit program could, for example, be a display in the PLC window, which can be done via a fixed PLC Module.

5.10 Cooperative Multitasking[1)]

It is possible to run several parallel processes in the PLC using the cooperative multitasking function. In comparison to real multitasking, information exchange and task change only occur at certain programmed instances in cooperative multitasking.

Cooperative multitasking comprises a maximum of 8 parallel PLC processes and the submit queue. This means that the commands for task change and event control (Module 926x) can also be used in a program started with SUBM. A task change is also inserted between the individual jobs in the submit queue, so that if parallel processes exist, they will be executed by the end of that job at the latest.

The cyclic PLC main program does not take part in cooperative multitasking, but interrupts a submit job and the parallel processes at whatever point they are at.

5.10.1 Starting a Parallel Process

A parallel process is started with the command:
SPAWN <jump label>
= D<number>
The identifier is sent back in the given doubleword (see submit job). If no process can be started, the value –1 is sent.

The SPAWN command can only be called from a submit job or from another SPAWN process. A maximum of 8 parallel processes can be started at the same time. If such a process is ended with EM, it is taken from the memory and this memory area is once again available.

[1)] As of NC software 280 472 01

5.10.2 Control of Events

The parallel processes can make events available to one another. This offers the advantage that no time is lost by the individual processes constantly interrogating the operating status.

One particular feature of the events control is the timeout. This timeout allows the process to "sleep" for a certain time, which is particularly useful for the repetition of program sections in a slow time-slot pattern (e.g. display functions, monitoring functions).

List of events

In the file OEM.SYS enter the complete name for the ASCII file containing your list of events using the command PLCEVENTS=.

The entries in the events file must have the following syntax:

<job name> ; <condition> ; <event> [;comment]

<job name>	This name is identical with the jump label in the SPAWN command. Only the first 16 characters are taken into consideration.
<condition>	Logical expression in accordance with C language convention, identical with the syntax used in function FN20: Operand: M/I/O/T/C/B/W/D with a permissible number 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 do not enter any condition, the condition = 0 will be tested .
<event mask>	Hexadecimal coded mask of events that are triggered if the condition is met. The limits defined in Module 9260 are valid for bits 16 to 31.

Example:

Entry in OEM.SYS:

```
PLCEVENTS=PLC:\EXAMPLE.PEV
```

Contents of 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
```

An event is triggered 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 that can be triggered at the same time is limited to 15.

When a PLC process is generated (with the SPAWN command), the events file is searched for entries for this process. All of the entries found are transferred to a list which is executed after each run of the cyclic PLC program. If a PLC process ends itself, or if is ended through a recompiling of the PLC program, the entries are deleted from the list and are once again available.

The entries in the events file are not monitored, which means that syntactically incorrect entries or incorrect job names do not lead to an error message being output. If an events file which actually does not exist is entered in OEM.SYS, a flashing error message is triggered with the first SPAWN command. An error message is also triggered if more events need to be monitored than the run time list contains (currently contains 15 entries).

Process monitor

The PROCESS MONITOR soft key is available in the PLC programming operating mode. This soft key opens a screen showing all of the parallel processes as well as the process for the submit queue. The current status (ready to run, running, waiting for event) of each process is shown in one second intervals. You can also see how many context changes were carried out by each process in the previous second, and how much CPU time it used. The distribution of the CPU time is shown in a bar chart.

Module 9260: Receiving and waiting for events

This module must be called in a submit or a SPAWN job. The module makes it possible for a SPAWN process or a SUBMIT job to interrogate or wait for the arrival of one or several events. At the same time the module triggers a contact change.

After the module call, the markers 4200 to 4202 and 4204 have undefined changes.

If the value zero is transferred for the events mask, all of the events are sent back without them being canceled. In a call with Wait, all of the events requested are sent back and are also canceled. In a call without Wait, only those events that have fulfilled the condition are sent back and canceled. In the case where the events are OR-gated, only the set events are sent and canceled. If you wish to specify which events should be canceled, you need to call without Wait and with OR-gating.

The event bits 16 to 31 are reserved for the operating system:

Bit16: BREAK	Aborts a function. May be set and read. Access to interfaces and network is interrupted if this event is sent in wait conditions!
Bit17:	*Reserved, do not use*.
Bit18:	*Reserved, do not use*.
Bit19: QUIT	Acknowledgment for a request. May only be used in immediate with a request.
Bit20 to Bit31:	*Reserved, do not use*.

Call in the SPAWN or submit job only:

```
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 = existing events are to be read only
CM   9260
PL   B/W/D      <Events>            read events
Error recognition: M4203 = 0:    Events were read
                   M4203 = 1:    Error code in W1022
                   W1022 = 2:    Incorrect transfer value for parameter <Wait>
                   W1022 = 20:   Module was not called in the submit job.
```

Module 9261: Sending events

With this module you can send events to a SPAWN or submit job. These events are then interrogated with Module 9260.

This module can be called up in the cyclical program section, in submit jobs and in the SPAWN processes. The identifier indicates the destination address. This identifier is then sent back with the SPAWN command. The submit queue is addressed with the identifier $80000000 (and not with the identifier sent back with the SUBM command!).

The events that are sent to the submit queue are always assigned to the job that is running at the time of arrival. If they are not read by this job, they are kept for the next job.

If the receiver process is also to start immediately, Module 9262 must be called immediately after Module 9261 so that a context change will be triggered.

The event bits 16 to 31 are reserved for the operating system (see Module 9260).

Call:

PS	D/K	<Identifier>	Identifier from SPAWN command of the receiver K$80000000 = submit queue
PS	B/W/D/K	<Events>	Events bit-coded
CM	9261		
Error recognition:		M4203 = 0:	Events were sent
		M4203 = 1:	Error code in W1022
		W1022 = 2:	Incorrect identifier

Module 9262: Context change between SPAWN processes

This module may only be called in a SPAWN or submit job. The module carries out a context change in another PLC process or in the submit queue if such a process exists which is not waiting for an event or a timeout.

After the module call, the markers 4200 to 4202 and 4204 have undefined changes.

Call in the SPAWN or submit job only:

CM	9262		
Error recognition:		M4203 = 0:	Context change was executed
		M4203 = 1:	Error code in W1022
		W1022 = 30:	Module was not called in the submit job.

Module 9263: Interrupt SPAWN process for a defined period of time

This module may only be called in a SPAWN or submit job. The module interrupts the calling process for at least the time indicated in the module. If other processes or the submit queue are ready to run, a context change is carried out to one of these processes.
After the module call, the markers 4200 to 4202 and 4204 have undefined changes.
The timeout is interpreted as an unsigned number, i.e. negative numbers result in extremely long timeouts.

Call only in the SPAWN or submit job:

```
PS   B/W/D/K      <Waiting time in ms>
CM   9263
Error recognition: M4203 = 0:     Timeout
                   M4203 = 1:     Error code in W1022
                   W1022 = 30:    Module was not called in the submit job.
```

5.11 Constants Field (KF)

The Constants Field data type can be used to access one of several constants defined in tabular form, depending on the value of the Index Register X.

Addressing is with KF <Name>[X], where <Name> is a jump label that identifies the beginning of the Constants Field.

Constants Fields start with the label KFIELD <Name> followed by a random (not zero) number of constants followed by the end label ENDK. Constants Fields may only be created when the program has been previously terminated with an EM or JP instruction.
The name of the Constants Fields conforms to the rules for jump labels.

Types of addressing:

L KF <Name> [X]: (X ≥ 0): The value of the constant defined by X from the Constants Field <Name>is transferred.

L KF <Name> [X]: (X = –1): The length of the Constants Field <Name> is transferred.

L KF <Name>: The absolute address of the Constants Field <Name> is transferred. This is only worthwhile in conjunction with modules (e.g. module 9200). This type of addressing can also be used within a Constants Field.

Example:

```
    L       KF VALUESFIELD    ;ACCESS TO VALUESFIELD WITH
                              VALUESFIELD[X]
                              ;X=[0 TO 3]
            = W0              ONE OF THE CONSTANTS IS
                              STORED
    EM
    KFIELD  VALUESFIELD
            K+10              CONSTANT TO BE LOADED IF X=0
            K+1
            K$ABC
            K-100000          CONSTANT TO BE LOADED IF X=3
            ENDK
```

Access to Constants Fields is checked in the same way as write access to indexed operands. This is why X may only assume positive values from 0 to <Length of Constants Field -1>.

5.12 Program Structures

A program is split up into program sequences so as to make it clearer. To do this the programmer uses jump labels (LBL) and conditional and unconditional jumps.

When structured instructions are used, the jump labels and jump commands are created by the Compiler. Remember that internal jump labels are generated to implement these structured commands, so the total number of available jump labels will be reduced accordingly. Structured instructions can be nested to up to 16 levels but there must be no "interleaving".

```
Right:  IFT                           Wrong:  IFT
        ...                                   ...
        WHILEF                                WHILEF
        ...                                   ...
        ENDW                                  END
        ...                                   ...
        ENDI                                  ENDW
```

Instructions IFT, IFF, WHILET, WHILEF, ENDW, UNTILT and UNTILF require a valid gating result in the Logic Accumulator. They terminate the gating chain. Instructions ELSE, ENDI and REPEAT require all gating chains to be terminated first.

5.12.1 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 can be omitted. 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 not fulfilled
• ENDI	(End of IF-Structure)	End of IF Structure

Example:

```
        L       I0
        IFT             ;If Logic Accu=1
        ....            ;Program code for I0 = 1
        ELSE            ;                           can be omitted

        ....            ;Program code for I0 = 0    can be omitted
        ENDI            ;end of conditional processing
```

5.12.2 REPEAT ... UNTIL Structure

The REPEAT ... UNTIL structure repeats a program sequence until a condition is fulfilled.
Under no circumstances may this structure wait for an external event in the cyclical PLC program to happen!

The following commands are available:

• REPEAT	(Repeat)	Repeat program sequence from here
• UNTILT	(Until True)	Repeat sequence until Logic Accumulator=1
• UNTILF	(Until False)	Repeat sequence until Logic Accumulator=0

A REPEAT ... UNTIL loop is always run at least once!

Example:

```
        =       M100            ;end of previous chain
REPEAT                          ;repeat following code
        .....                   ;code to be executed
        LX                      ;load Index Register
        >=      K100            ;check Index Register
UNTILT                          ;repeat until X>=100
```

5.12.3 WHILE ... ENDW Structure

The WHILE ... ENDW structure repeats a program sequence if a condition is fulfilled.
Under no circumstances must this structure wait for an external event in the cyclical PLC program to happen!

The following commands are available:

• WHILET	(While True)	Execute sequence if Logic Accumulator=1
• WHILEF	(While False)	Execute sequence if Logic Accumulator=0
• ENDW	(End While)	End of program sequence, go back to beginning

A WHILE ... ENDW loop is only run when the WHILE condition is fulfilled at the beginning. The execution condition must be repeated before the ENDW instruction. The condition can also be repeated differently than before the WHILE instruction!

Example:

```
      .....
      L     M100        ;create condition for 1st WHILE scan
WHILET                  ;execute following code if Logic Accumulator = 1
      .....             ;code to be executed
      L     M101        ;create condition for repeat processing
      A     M102        ;next condition
ENDW                    ;return to WHILE scan
```

Two internal jump labels are generated for the WHILE ... ENDW structure.

5.12.4 CASE Branch

Indexed Module Call (CASE)

Abbreviation for PLC Editor: CASE (CASE OF)

	Byte	Word
Execution time [µs]	3.3 to 3.8	3.3 to 3.8
Number of bytes	46	44

4 bytes must be added to the length for each entry in the jump table (CM).

Operands: B, W

Operation:
The CASE command is used to select a defined subprogram from a list of module calls (CM). These CM commands come directly after the CASE command and are numbered internally in ascending order from 0 to 127 maximum. The content of the operand (B, W) addresses the desired module.

End of Indexed Module Call (ENDC)

Abbreviation for PLC Editor: ENDC (ENDCASE)

	Byte	Word
Execution time [µs]	0	0
Number of bytes	0	0

Operands: none

Operation:
The ENDC command is used together with the CASE command. It must come directly after the list of CM commands.

Structure of a CASE instruction:

```
          Internal addressing (0 to 127 max.)
1         CASE B 150
2         CM 100    <-------    (0)
3         CM 200    <-------    (1)
4         CM 201    <-------    (2)
5         CM 202    <-------    (3)
6         CM 203    <-------    (4)
7         CM 204    <-------    (5)
8         CM 300    <-------    (6)
9         ENDC
```

Line 1: Command + Operand; the internal address of the required module must be filed in the operand
Line 2: Call Module if operand content 0
Line 3: Call Module if operand content 1
Line 4: Call Module if operand content 2
Line 5: Call Module if operand content 3
Line 6: Call Module if operand content 4
Line 7: Call Module if operand content 5
Line 8: Call Module if operand content 6
Line 9: End of CASE instruction

5.13 Linking Files

The source code of the PLC program can be stored in several different files that are managed with the USES, GLOBAL and EXTERN commands. These must be written at the beginning of the program, i.e. before any PLC instruction (see "PLC Program Example").

The USES command links another file to the program. The GLOBAL command supplies a jump label from its own file as an entry that can be used by all other files. The EXTERN command supplies a jump label defined in another file where it is declared as GLOBAL for use in its own file.

Splitting the source code up into a number of files helps improve clarity and overview by swapping out individual function groups.

The number of possible jump labels is 1000 for each file.

In all, up to 60 files can be linked to form a program. Each file can generate up to 64 KB of code. The total potential length of the code is limited to 128 KB.

Where several files exist, the main program must have the status flag "M" in the directory. In the RAM this is done by selecting the PLC program function "COMPILE" once, in the EPROM it is done by specifying the /M option after the main program in the Linker file for binary output.

5.13.1 USES Instruction

The USES instruction links other files to the main program. Files linked with USES can also link other files with the USES instruction. It is also acceptable for one file to be linked by several other files with USES, and code for that file is only generated once.

The USES instruction requires a file name as its argument.

The USES instruction only links the file, the program code for the file is not executed, i.e. USES cannot be compared to a CM instruction. The linked files must therefore contain individual modules which can then be called with CM instructions.

Example:

```
        USES    PLCMOD1                 ;
        USES    EPRUPG                  ;
        USES    RAMPLC                  ;
```

Example of file linking:

```
PLCMAIN.PLC
;main program
USES    SPINDEL.PLC
USES    TCHANGE.PLC
;code

                        TCHANGE.PLC                     SPINDEL.PLC
                        ;tool change                    ;spindle control
                        USES    PLCUPG.PLC              USES    PLCUPG.PLC
                        ;code                           ;code

                                        PLCUPG.PLC
                                        ;general subprograms
                                        ;Code
```

5.13.2 GLOBAL Instruction

Up to 1000 local jump labels can be defined in each of the files linked with USES. Modules defined in one file must be defined globally before they can be called from another file.

This is done with the GLOBAL instruction at the beginning of the file. Jump labels can only be defined globally when they are defined as LBL later on in the program (and not as KFIELD!).

Syntax:

```
GLOBAL          jump label          ;declaration of "jump label" beyond the file boundary
```

GLOBAL definitions must not be written in the main program. A single jump label cannot be declared globally by more than one module, however a name that has been declared globally by file A can be used again locally in file B.

In all, 1000 jump labels can be defined globally by all modules.

5.13.3 EXTERN Instruction

For a jump label to be able to access in one file modules which other files have declared as GLOBAL, it must be declared as EXTERN. The EXTERN instruction must be written at the beginning of the file.

The commands CM, CMT and CMF can then jump to this label in the program code.

The instructions JP, JPT, JPF, access to a Constants Field and linking with CM to a CASE Branch are not possible with external jump labels.

The name of the jump label cannot be assigned again in this file for a local jump label. Every external jump label reduces the number of available local jump labels.

Syntax:

```
EXTERN          Jump label        ;the "Jump Label" Module can now be called from another file
                                  with the CM instruction.
```

5.14 PLC Modules

A number of PLC modules are available for PLC functions that are very difficult, or even impossible, to perform only with PLC commands.

You will find a description of these modules in the corresponding function descripions. There is a list of all available modules in the chapter entitled "List of PLC Modules."

Improper execution of a module sets Marker 4203. This marker can then be used to evaluate an error message.

5.14.1 Markers, Bytes, Words, Doublewords

Copy in Marker or Word Range (Module 9000/9001)
Modules 9000 (Marker) and 9001 (Byte/Word/Double) copy a block with a certain number of markers or bytes beginning from the start address to the specified target address.
For module 9001 the length should always be defined in bytes.

Constraints:
Copying is sequential, starting with the first memory cell. This means that the function is not guaranteed when the source and destination blocks overlap and the source block begins at a lower address than the destination block. In this case the overlapping part of the source block is overwritten before copying takes place.

Possible errors:

- A block of the defined length cannot be read from the defined address in the marker or word RAM (address is too high or block is too long).
- A block of the defined length cannot be written to the defined address in the marker or word RAM (address is too high or block is too long).

Call:

```
PS   B/W/D/K     <Number 1st marker source block>
PS   B/W/D/K     <Number 1st marker destination block>
PS   B/W/D/K     <Length of block in markers>
CM   9000        Transfer in marker range
```

```
PS   B/W/D/K     <Number 1st marker source block>
PS   B/W/D/K     <Number 1st marker destination block>
PS   B/W/D/K     <Length of block in bytes>
CM   9001        Transfer in word range
```

Read in Word Range (Module 9010/9011/9012)
A byte, word or doubleword is read from the defined position in the word memory and returned to the stack as an output variable. Indexed reading in the memory is possible by specifying a variable as the name of the memory cell.

Possible errors:

- The defined address is outside the valid range (0..1023).
- Module 9011: The defined address is not a word address (not divisible by 2).
- Module 9012: The defined address is not a doubleword address (not divisible by 4).

Call:

```
PS   B/W/D/K     < Number of byte to be read >             (Address)
CM   9010        read byte
PL   B           <byte read>        (Value)
```

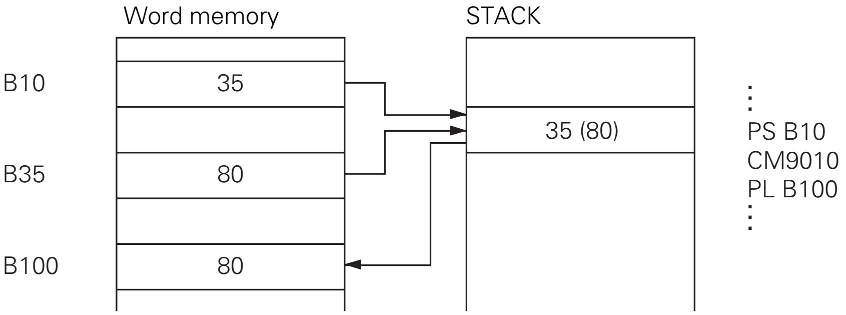
or

```
PS   B/W/D/K     <Number of the word to be read >          (Address)
CM   9011        read word
PL   W           <word read>              (Value)
```

or

```
PS   B/W/D/K     <Number of the doubleword to be read>   (Address)
CM   9012        read doubleword
PL   D           <doubleword read>        (Value)
```

Example of Module 9010



```
:
PS B10
CM9010
PL B100
:
```

Write in Word Range (Module 9020/ 9021/ 9022)
The defined byte, word or doubleword is written to the defined position in the word memory. Indexed reading in the memory is possible by specifying a variable as the name of the memory cell.

Possible errors:
- The defined address is outside the valid range (0..1023).
- Module 9021: The defined address is not a word address (not divisible by 2).
- Module 9022: The defined address is not a doubleword address (not divisible by 4).

Call:

```
PS   B/W/D/K    < Number of byte to be written >          (Address)
PS   B/W/D/K    <byte to be written>
CM   9020       write byte    (value)
```

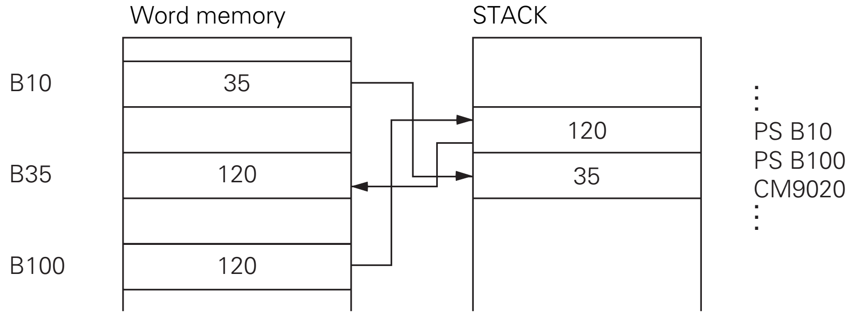
or

```
PS   B/W/D/K    < Number of word to be written >          (Address)
PS   B/W/D/K    <byte to be written>
CM   9021       write word   (value)
```

or

```
PS   B/W/D/K    < Number of doubleword to be written >   (Address)
PS   B/W/D/K    <byte to be written>
CM   9022       write doubleword         (Value)
```

Example for Module 9020:



5.14.2 Number Conversion

Conversion of Binary Number to ASCII (Module 9050)
Converts a binary numerical value consisting of mantissa and exponent to the base 10 to an ASCII-coded decimal number.

The number specified as mantissa and exponent is converted to a decimal number and stored at the specified address as a string. The exponent relates to the lowest-value place in the number. A negative number is detected when the mantissa corresponds to a negative number in the notation as a two's complement. A sign is only set in front of negative numbers. Trailing zeroes after the decimal point or leading zeroes before the decimal point are not converted, the string is written left-justified starting from the specified target address in the string buffer.

Constraints:
The decimal sign is defined by machine parameter MP7280 as a decimal comma (MP7280 = 0) or a decimal point (MP7280 = 1).

Possible errors:
- The number of the target string is outside the permitted range (0..3).
- The conversion would result in more than 10 places after the decimal point.
- The conversion would result in more than 10 places before the decimal point.

Call:

```
PS   K/B/W/D    <Mantissa of numerical value to be converted>
PS   K/B/W/D    <Exponent to base 10 of the value>
PS   K/B/W/D    <Number of target string>
CM   9050
```

Conversion of Binary Number to ASCII (Module 9051)

Converts a binary numerical value to an ASCII-coded decimal number in the format specified.

The specified number is converted to a decimal number and stored as a string in the specified address. The number is interpreted as a two's complement. When interpreted without a sign the absolute amount of the number is converted without a sign being put before the string. With the algebraically signed notation, the sign ("+" or "-") is placed before the string in any event.

With the inch notation the numerical value is divided by 25.4 before being converted. If the number has more decimal places than the total of specified places before and after the decimal point, then the highest-value decimal places are omitted. With right-justified notation leading zeroes before the decimal point are replaced by blanks, with left-justified notation they are suppressed. Trailing zeroes after the decimal point are always converted.

Constraints:
The decimal sign is defined by machine parameter MP7280 as a decimal comma (MP7280 = 0) or a decimal point (MP7280 = 1).

Possible errors:
- The number of the target string is outside the permitted range (0..3).
- There are more than 16 decimal places in all (before and after decimal point)
- No places before the decimal point are specified

Call:
```
PS   K/B/W/D     <Numerical value to >
PS   K/B/W/D     <display mode (bit coded)>
                 Bit #3: Display with sign
                 Bit #2: Display converted to INCH
                 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
PS   K/B/W/D      <Number of places after the decimal point>
PS   K/B/W/D      <Number of places before the decimal point>
PS   K/B/W/D     <Target address in string buffer>
CM   9051
```

Conversion of ASCII Number to Binary (Module 9052)

Converts an ASCII coded decimal number (possibly with places after the decimal point) into a mantissa and an exponent to the base 10.

The string identified by the source string number is read and converted to a signed number and an exponent to the base 10. If the sign is missing the number is detected as positive. Both the comma and the point are accepted as the decimal character. If the full extent of the mantissa cannot be represented in a doubleword, then the last places are omitted and the exponent corrected accordingly. If possible the exponent is adjusted to correspond with the ASCII notation of the number.

Possible errors:

- The number of the source string is outside the valid range (0..3).
- The source string does not contain a string that can be interpreted as a number.
- The string overflows the end of the string buffer, 128 characters were read without an end of string being found.

Call:

```
PS   K/B/W/D    <Number of the source string>
CM   9052
PL   B/W/D      <Numerical value>
PL   B/W/D      <Exponent 10Exx)
```

Conversion of Binary to ASCII/Hexadecimal (Module 9053)

Converts a block of binary values from the word marker range into a string of ASCII coded hexadecimal numbers.

The specified number of bytes is read from the place specified by the source address and converted to a hexadecimal-coded ASCII string. Each byte in the source block makes 2 characters in the destination string. the destination string is identified by the destination string number.

Possible errors:

- The address for the source block is outside the range 0 to 4095.
- The number of the destination string is outside the valid range (0 to 3).
- The number of data bytes is too high (0 to 63).

Call:

```
PS   K/B/W/D    <Source address in Word-RAM>
PS   K/B/W/D    <Number of destination string>
PS   K/B/W/D    <Number of data bytes>
CM   9053
```

Conversion of ASCII/Hexadecimal to Binary (Module 9054)

Converts a string of ASCII coded hexadecimal values to a block of binary values in the word marker range.

The string in the string buffer with the specified number is interpreted as a chain of ASCII coded hexadecimal numbers and converted into a block of corresponding binary bytes. Two ASCII characters make one binary byte. The binary block is stored in the word marker range starting from the specified destination address.

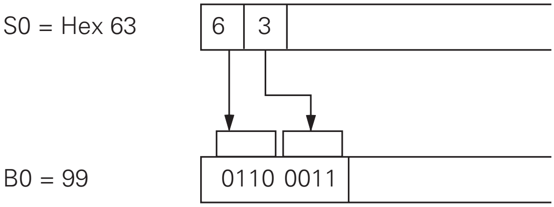
Possible errors:

- The number of the source string is outside the permitted range (0..3).
- The address for the destination block is outside the range 0 to 4095.
- The source string contains characters that cannot be interpreted as hexadecimal values (different characters 0 to 9, A to F).
- The source string contains an uneven number of characters (the last byte is not fully defined).
- The destination block has no room at the specified address

Call:

```
PS   K/B/W/D     <Number of source string>
PS   K/B/W/D     <Destination address in Word-RAM>
CM   9054
```

Example:



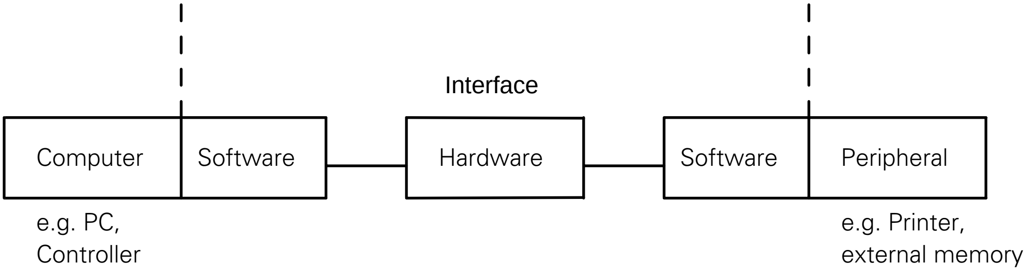
6 Data Interfaces

6.1 Introduction

In addition to their Central Processing Unit (CPU), computer systems such as PCs or machine controllers usually include a wide variety of peripheral devices such as printers and external storage devices (e.g. floppy-disk drives and hard disks). A computer system may also be connected with other computer systems.

A data interface makes it possible for the CPU and its peripheral devices to communicate. Communication requires facilities for transferring data to the peripherals and of course, physical connection by means of a transmission line. Peripheral device control and communication, via the interface, is generally the responsibility of the computer system. The computer system therefore has to meet certain requirements.

The interfaces, which primarily consist of the physical links between the computer system and the peripherals, need appropriate software in order to control the transfer of information between the individual units. The relationship between hardware and software, which fully defines an interface, is illustrated by the following diagram:



The "hardware" in the diagram covers all the physical components, such as circuit construction, pin layout, electrical characteristics, etc. The "software" includes, for example, the drivers for the output modules, which are part of the operating software both of the computer system and the peripherals.

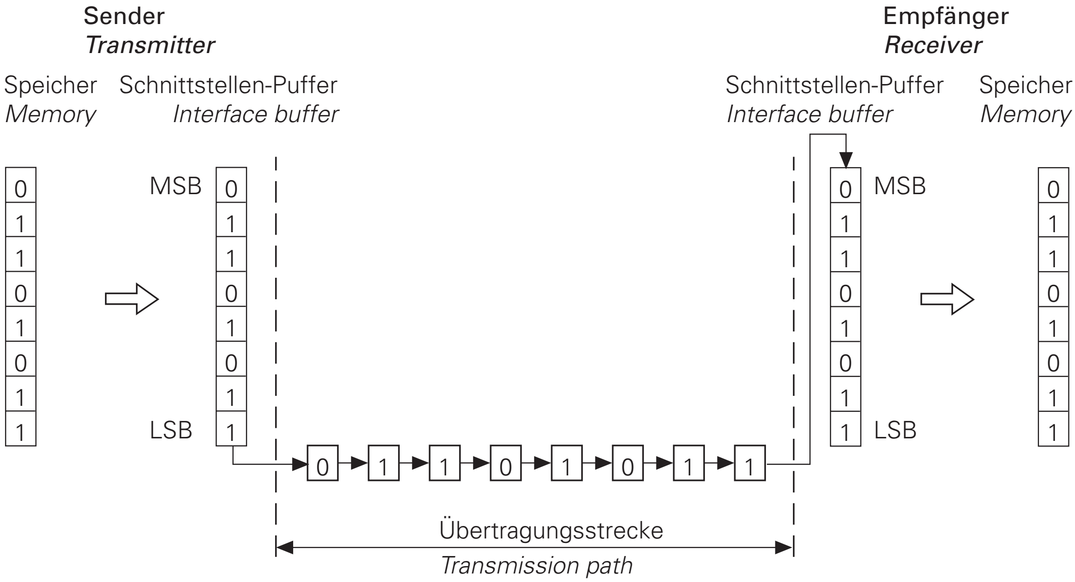
Standard interfaces were developed in an effort to enable the extremely wide variety of existing computers, controllers and peripherals to be connected to each other. Such standards include the
RS-232-C/V.24 and **RS-422/V.11** interfaces, which are described in detail later.

6.1.1 Principles of Data Transfer

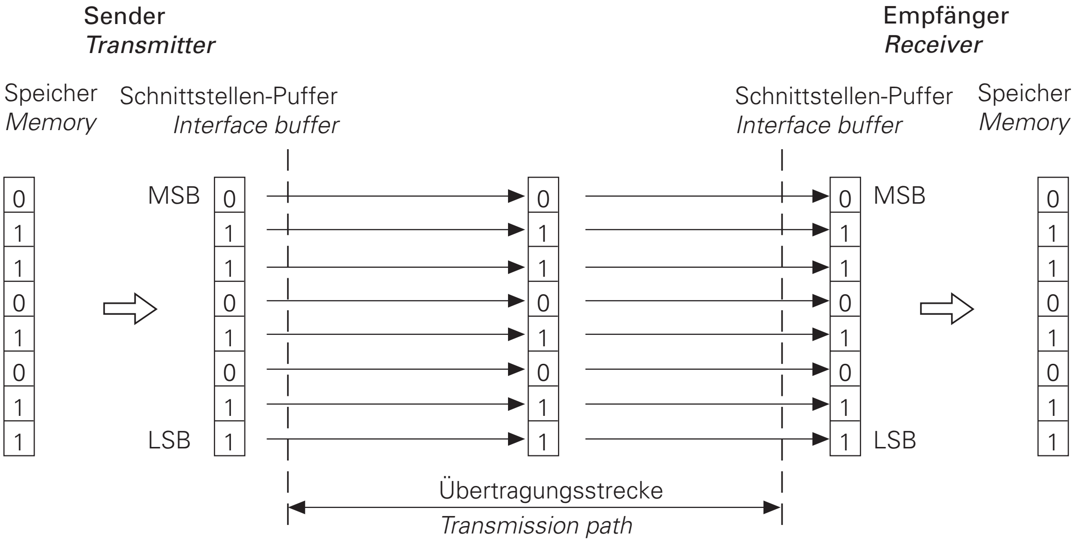
Since all information is conveyed as data, one first needs to become familiar with a few of the principles of data transfer. The term "data" is used to describe all of the information that the computer is capable of collecting and processing.

Serial/Parallel
Data can be transmitted in either serial or parallel format. Basically, data is coded in the computer system, e.g. as bytes (8 bits) and supplied to the interface in parallel.

In the case of serial data transmission, the parallel information from the computer system has to be converted into a serial data-flow by a USART (Universal Synchronous/Asynchronous Receiver/Transmitter). The receiver accepts the serial data-flow and converts it back again into parallel information.



A parallel interface, on the other hand, needs only line drivers, not a USART. Typically, the connection between the computer system and a peripheral consists of a 36-line ribbon cable. Its maximum length is generally about 3 meters.



One obvious advantage of serial data transmission becomes apparent when long distances have to be covered. With parallel transmission, the cost of the cable increases with every additional bit that has to be transmitted. In addition, the effect of interference on adjacent wires from sharp signal edges and electrical coupling is far greater over long lines than it is with serial transmission, which is relatively slower and uses fewer wires.

The comparatively slow speed of serial data transmission is, at the same time, its greatest drawback. Since the individual bits are sent along the line one after the other and each transfer takes a certain time, it takes far longer to send a binary word to the receiver than it would if conveyed by parallel transmission. As it happens, most peripheral devices work fairly slowly and cannot in fact cope with data transmitted at high speed. Serial data transmission is generally adequate for devices such as external memories or mechanical printers, unless such devices have a large internal buffer for incoming characters.

Asynchronous data format

In order for communication to be established between two devices involved in data interchange, they must use a common language. In the field of computer engineering, this language consists of digital coding of letters, numbers and control characters.

One of the most common codes is the ASCII code (American Standard Code for Information Interchange), which codes all characters with seven bits. In all, it is possible to code 2^7 = 128 characters. According to the ASCII code, the control character "Line Feed" or <LF> is coded with the following combination of bits:

$$\underset{\text{MSB}}{0\ 0\ 0\ 1}\ 0\ 1\ \underset{\text{LSB}}{0} \quad = \quad 10 \text{ dec} = 0\text{A hex}$$

The letter 'z' is represented by the following combination of bits:

$$\underset{\text{MSB}}{1}\ 1\ 1\ 1\ 0\ 1\ \underset{\text{LSB}}{0} = 122\text{ dec} = 7\text{A hex}$$

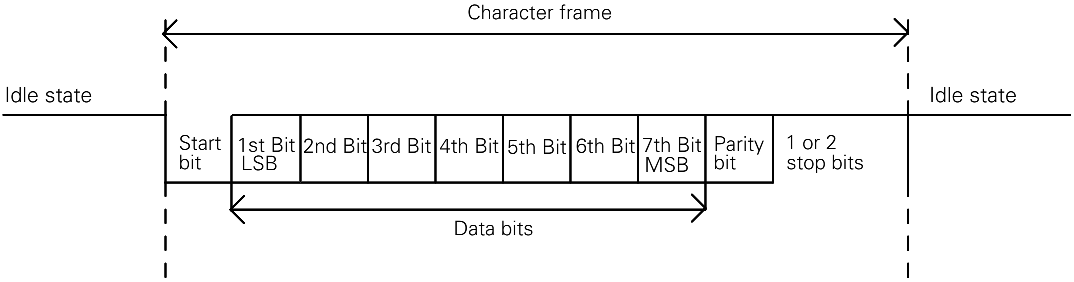
When the letter "z" is transmitted via a serial interface, the appropriate bits are sent one after the other.

Successful data transmission requires the device concerned to interpret incoming data correctly and, in particular, to recognize the start of a transmission. For this purpose, there is a synchronization process that ensures that the receiver detects the first bit of a character correctly. With an asynchronous data format, a start bit is sent before each data word and the word is then ended by one or two stop bits. One feature of this data format is that, starting from an idle state, transmission of a data word can begin at any time.

An idle state exists before switch-on and is returns after each transmission. Before a data bit can be transmitted this has to be communicated to the receiver. Otherwise, if the first bit of the data word has the same value as the idle state, the receiver will not notice any difference from the idle state.

A so-called "start bit" is used for this purpose: for the duration of a single bit, the transmitter emits a logic value that clearly differs from the idle state and gives the receiver an opportunity to prepare its polling logic to read in the data bit. After the start bit has been sent, the data word is transmitted, bit by bit, starting with the LSB (Least Significant Bit). After the MSB (Most Significant Bit) of the data word, a so-called "parity bit" is added in order to detect transmission errors.

The parity bit is followed by one or two stop bits. These final stop bits ensure that the receiver has enough time to recognize the transmitter again before the start of the next character. Synchronization is repeated before each character and applies for one character frame.



Checking data
With an asynchronous character frame, transmission errors can be detected by using a parity-check procedure. A parity bit is sent in addition to the data bits. The evaluation of this bit enables the receiver to check the parity of received data.

The parity check can take three different forms; the same type of parity check must be set at both interfaces.

- 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 the 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 this count yields an odd number, there is a transmission error and the data word must be repeated, or an error message will be displayed.
- Odd parity:
  In this case, the parity bit is so chosen by the transmitter that the total number of all the set bits is odd. In this case, an error will be detected if the receiver observes an even number of set bits in its evaluation.

Example:

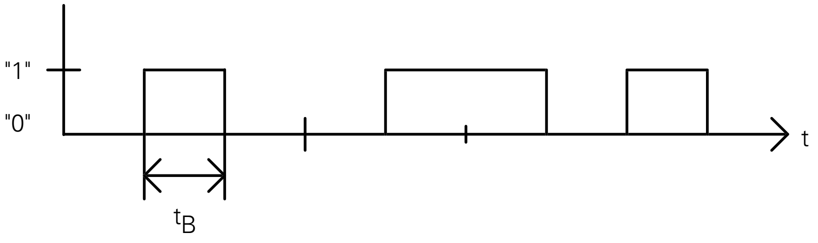
Letter "z" 1111010	Parity bit:	with even parity bit = 1
		with odd parity bit = 0

Data transfer rate
The data transfer rate of an interface is given in baud, which indicate the number of bits of data transmitted in one second.

$$1 \text{ baud} = 1 \frac{\text{bit}}{\text{s}}$$

Common baud rates are:
110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38 400, 57 600, 115 200 baud.

The time taken to transmit one bit (t_B) can be calculated from the baud rate:



$$t_B = \frac{1}{\text{Baud rate}[\frac{\text{bits}}{\text{s}}]}$$

For example, a baud rate of 19 200 baud will have a bit duration of t_B = 52.083 µs. The number of characters transmitted can be calculated from the baud rate and the transmission format:

$$\text{Characters per second} = \frac{\text{Baud rate}[\frac{\text{bits}}{\text{s}}]}{\text{Number of bits per character}}$$

For example: with a transmission format of one start bit, seven data bits, two stop bits and a data transfer rate of exactly 300 baud:

$$\frac{\text{300 baud}}{\text{10 bits}} = \text{30 characters per second will be transmitted.}$$

6.1.2 Handshaking

A "handshake" procedure is often used in connection with interfaces. This means that two devices are, as it were, working "hand in hand" in order to control data transfer. A distinction is drawn between "software" and "hardware" handshaking. Either hardware or software handshaking can be chosen for communication between two devices.

Hardware handshaking
With this procedure, data transfer is controlled by electrical signals. Important information, such as Clear To Send (CTS), Request To Send (RTS), "Start transmission" and "Stop transmission" is signaled by the hardware.

For example, when a computer character is to be transmitted, the CTS signal line is checked to see whether it is active (ON). If it is, the character is transmitted. Otherwise the computer will delay transmission until the CTS line is switched to active.

Hardware handshaking requires as a minimum the two data lines TXD and RXD, the RTS control line, the CTS signal line, and a ground connection.

Software Handshaking
With software handshake, control of data transfer is achieved by appropriate control characters transmitted via the data line. One such handshake is the XON/XOFF method, which is in widespread use with the RS-232-C/V.24 interface. The meaning XON is assigned to a 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, it delays transmission until it receives the character XON, indicating that the connected unit is ready to receive further characters.

Besides the ground line and the data lines TXD and RXD, no other lines are needed for software handshaking.

6.2 TNC Data Interfaces

6.2.1 General

The TNC features two data interfaces: one RS-232-C/V.24 and one RS-422/V.11 interface. Both interfaces differ only in the design of their hardware (signal lines, signal levels and pin layout).The data format and transmission protocol are the same for both interfaces.

You can connect the HEIDENHAIN FE 401 floppy disk unit and external devices with appropriate data interfaces (computers, printers, readers, punches) via either RS-232-C/V.24 or RS-422/V.11 interfaces. The two interfaces of the TNC can also be operated in parallel.

Three transmission protocols are available for data transfer:

- Standard data transmission protocol
- Data transfer with Block Check Character (BCC)
- LSV2 protocol

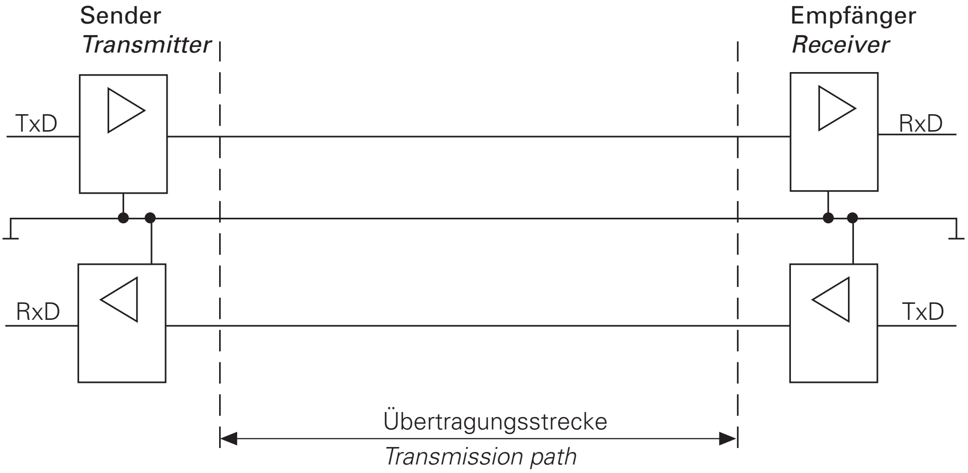
6.2.2 RS-232-C/V.24 Interface

RS-232-C is the designation of a serial interface based on the American EIA standard of the same name. Data transfer is executed asynchronously, with a start bit before each character and one or two stop bits after each character. The interface is designed for transmission distances of up to 30 meters.

The RS-232-C interface has been adopted with slight modifications and has been introduced into Europe as the V.24 interface. The relevant German standard is DIN 66020.

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.

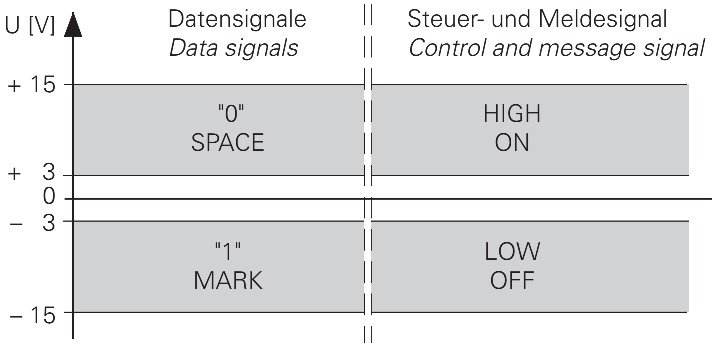
Physical connections:



Signal levels
With the RS-232-C/V.24 interface one must differentiate between two different signal lines and their levels.

- Data lines:
  The data signals are defined as being logic one (MARK) over the range –3 V to –15 V and as logic zero (SPACE) over the range +3 V to +15 V.
- Control and signal lines:
  These signals are defined as being ON (High) over the range +3 V to +15 V and as OFF (Low) over the range from –3 V to –15 V.

For all of the signals, the voltage range from –3 V to +3 V is not defined as a logic level and can therefore not be evaluated.



Signal Designations
The RS-232-C/V.24 interface distinguishes between the following types of lines.

- Data lines:
  TxD Transmitted data
  RxD Received data

- Control/signal lines:
  DCD Data Carrier Detect): Received signal level. With the DCD signal the receiver indicates that the information it has received lies within the defined level. The DCD signal (pin 8) is not used by the TNC, i.e. the TNC delivers no signal from pin 8.
  DTR (Data Terminal Ready): This signal shows that the TNC is ready for service (e.g. receiving buffer full → DTR = Low).
  DSR (Data Set Ready): Peripheral 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 is ready to receive.

- Ground conductor (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. See the chapter "Installation and Electrical Connection."

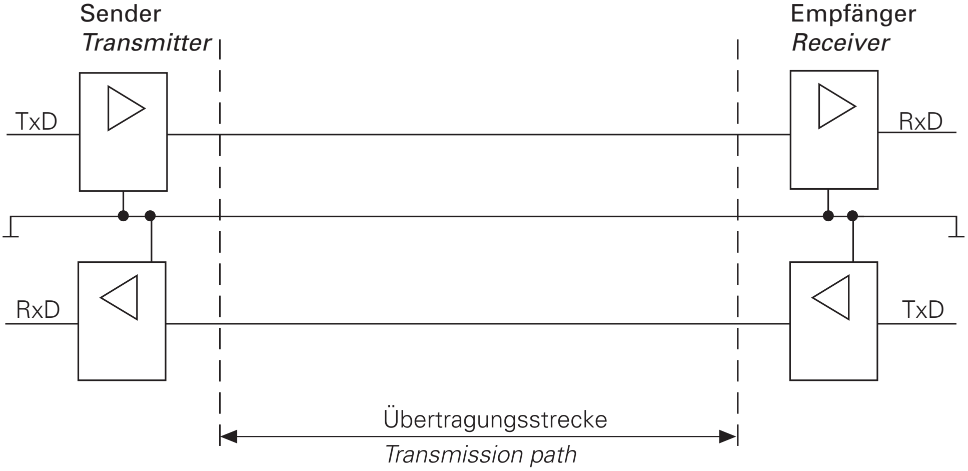
6.2.3 RS-422/V.11 Interface

The RS-422/V.11 interface was developed to improve on the capabilities of the RS-232-C/V.24 interface. Like the RS-232-C/V.24 interface, it is standardized and works symmetrically. The RS-422/V.11 interface is suitable for data transfer speeds up to 10 megabits/sec. The interface module of the TNC can transmit data at up to 115 200 baud. At this baud rate it is possible to transmit over a cable one kilometer long.

Hardware
The standard RS-422/V.11 works with differential voltages. The advantage of this method is that, on the transmission path, electromagnetic interference acts simultaneously and with the same effect on both signal lines. At the receiver only the difference in voltage of the two signal lines is evaluated and therefore the electromagnetic interference is unimportant.

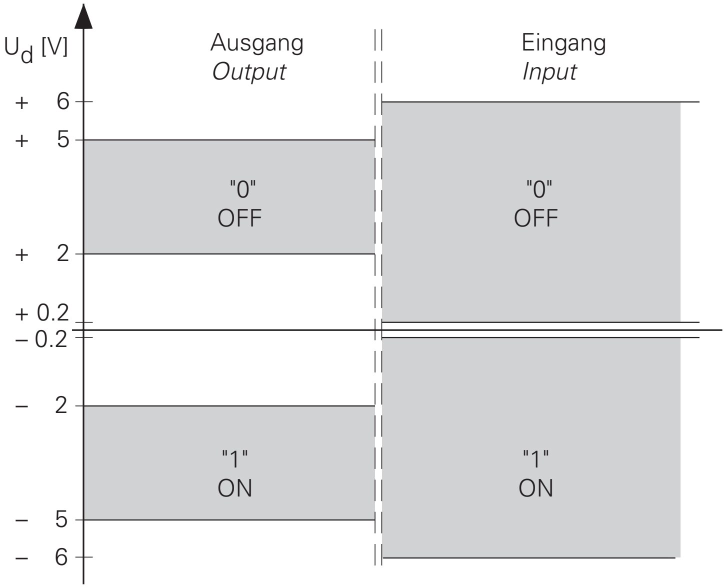
Considerably longer lines can therefore be used and, because of the suppression of interference, the transfer speed can be considerably higher.

Physical connections:



Signal levels
On an RS-422/V.11 interface, the signals are both transmitted and received as differential voltage. A positive differential voltage corresponds to logic zero (OFF), and a negative differential voltage to logic one (ON).

Differential voltages between
V_{dmin} = 2 V and V_{dmax} = 5 V
are transmitted and the control unit detects the differential voltages between
V_{dmin} = 0.2 V and V_{dmax} = 6 V
as a logically defined level.



Signal designations
With an RS-422/V.11 interface, the following signals are transmitted as differential signals:

• Data signals:	TxD, $\overline{\text{TxD}}$	RxD, $\overline{\text{RxD}}$
• Control and message signals:	RTS, $\overline{\text{RTS}}$	CTS, $\overline{\text{CTS}}$
	DSR, $\overline{\text{DSR}}$	DTR, $\overline{\text{DTR}}$

In addition, the protective ground connects the transmitter and receiver housings. The signal GND represents the differential voltage reference conductor.

These signals perform the same functions as those on the RS-232-C/V.24 interface. The transmission protocol is fully identical for the RS-232-C/V.24 and RS-422/V.11 interfaces.

Pin layouts
The TNC logic unit and adapter block have the same pin layout (see the chapter "Installation and Electrical Connection").

6.2.4 Saving/Reading Files

The following table lists all the files that can be saved to external memory (floppy disk unit or PC) and can be 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 tables	.PNT	U
PLC error table	.PET	F
System file	.SYS	O
Cutting data table	.CDT	–
Freely defined 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	–

After you have entered the appropriate code numbers for the PLC, the machine parameters, and the correction table, you can write to or read from these files via the data interfaces. You can also output the current values of Q parameters, PLC error messages and dialogs via the two interfaces (NC program: FN 15: PRINT). During transmission with a Block Check Character (BCC) each device outputs and again receives the appropriate file code.

If the file is stored in an external computer using HEIDENHAIN's **TNC.EXE** data transfer software, a new file extension is generated. This extension consists of the file code and the letters NC.

Example:
If a pallet table is stored, it is given the file extension *.LNC.

For data transmission with the HEIDENHAIN software **TNC Remote,** the file code has no significance. The files are saved on the PC with the same extension as on the TNC.

Files that have no code (–) can only be transmitted with LSV2 protocol (TNC Remote).

Output to external devices
Any external device, e.g. computers, printers, readers and punches, can be addressed through either of the two interfaces. For this purpose, the TNC has three freely configurable interface modes (EXT1/EXT2/EXT3) that, within certain limits, permit any setting of the data format of the required data transmission protocol.

The setting selected at external devices must of course match the TNC. On printers, this is done by setting the DIP switches or adjusting the transmission parameters. If data transfer to a computer is desired, appropriate data transfer software must be installed. To help in this, HEIDENHAIN offers the data transfer software TNC.EXE, which permits transfer between TNC and a PC using a fixed transmission protocol.

Communication between TNCs
For certain applications, it is necessary for TNCs to be able to exchange data or to communicate with each other. This is made possible both by RS-232-C/V.24 and RS-422/V.11 interfaces.

The simplest form of data exchange is the transfer of files (e.g. NC programs) from one TNC to another. To do this, 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).

6.3 Configuration of Interfaces

6.3.1 Selection of Interfaces

Either data interface can be inhibited with MP 5000. If neither of the interfaces is inhibited, you can use the RS-232-C RS 422 SETUP soft key to select the following settings:

Operating mode	External device
FE1	HEIDENHAIN Floppy Disk Unit: • FE 401B • FE 401 from Program no. 230 626 03
FE2	• HEIDENHAIN FE 401 Floppy Disk Unit up to program no. 230 626 02 • PC with HEIDENHAIN TNC.EXE data transfer software
LSV2	PC with HEIDENHAIN software TNC Remote
EXT1 and EXT2	Non-HEIDENHAIN device such as a printer, punch or PC with other data transfer software

MP5000 Disable data interfaces
Input: 0 = no interface disabled
1 = RS-232-C/V.24 interface disabled
2 = RS-422/V.11 interface disabled

6.3.2 Freely Configurable Interfaces

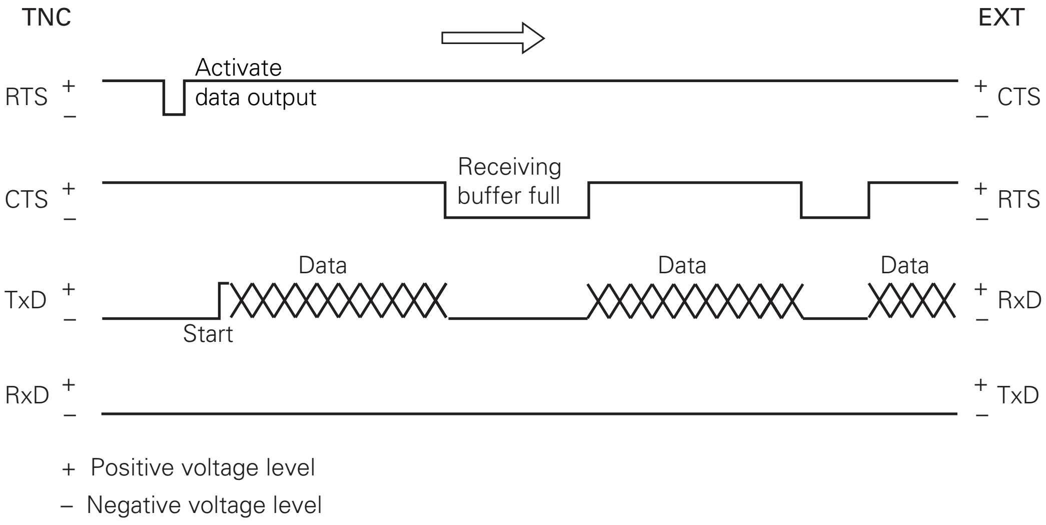
The three operating modes EXT1/EXT2/EXT3 (EXT3 only for the PLC) can be configured in MP5020.x.

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, but in some cases, especially for printer interfacing, eight bits are needed.

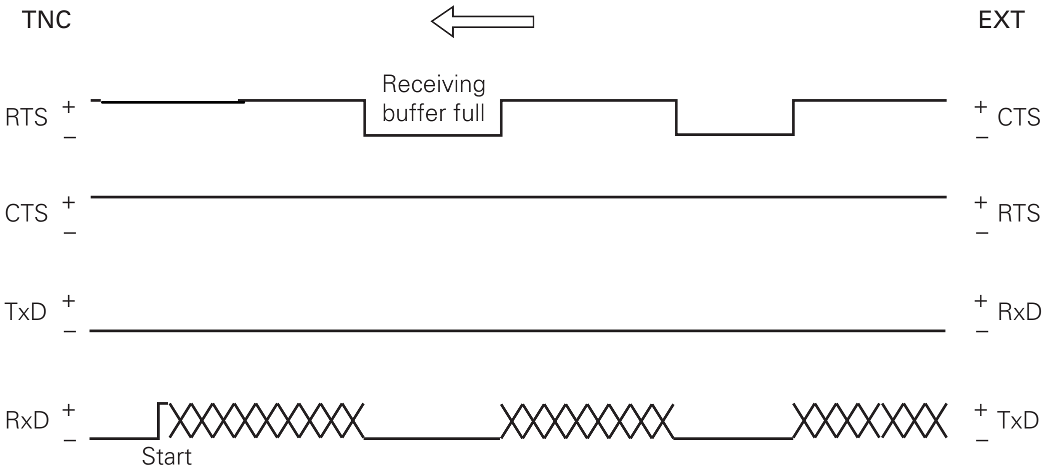
BCC
If calculation of the BCC produces a number less than $20 (i.e. a control character) then a "Space" character ($20) is sent in addition immediately before <ETB>. The BCC will consequently always be greater than $20 and cannot therefore be interpreted as a control character.

Hardware handshaking
Bit 2 can be set to determine whether the TNC stops transfer from an external device by sending an RTS signal.

- Data output TNC → EXT
  When the receiving buffer is full, the external device resets the RTS signal. The TNC thereby detects that the peripheral unit receiving buffer is full because of the CTS input.



- Data input EXT → TNC
  When the receiving buffer is full, the TNC removes the RTS signal, which is detected by the peripheral device at its CTS input.



+ Positive voltage level

– negative voltage level

The DTR and DSR signals from the TNC indicate the operational status of the TNC and peripheral (these cannot be set via the machine parameters).

DTR: Interrogated by peripheral; it is logic one if TNC is ready for service.
DSR: Interrogated by TNC.
LOW level → ext. data input/output not ready.
HIGH level → ext. data input/output 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>.

If transfer is stopped with a 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.

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 both RTS and by DC3 is active, 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.

MP 5020.2	Configuration of the data interface		
Entry:	%xxxxxxxx		
Bit 0	7 or 8 data bits	0 =	7 data bits
		1 =	8 data bits
Bit 1	Block check character	0 =	any BCC
		1 =	BCC not control character
Bit 2	Transmission stop by RTS	0 =	not active
		1 =	active
Bit 3	Transmission stop by DC3	0 =	not active
		1 =	active
Bit 4	Character parity	0 =	even
		1 =	odd
Bit 5	Character parity	0 =	not desired
		1 =	desired
Bit 6/7	Stop bits		
		Bit 6	Bit 7
	1 1/2 stop bits	0	0
	2 stop bits	1	0
	1 stop bit	0	1
	1 stop bit	1	1

MP5020.0	Operating mode EXT1
MP5020.1	Operating mode EXT2
MP5020.2	Operating mode EXT3 (PLC)

Transmission protocol
For operating modes EXT1/EXT2/EXT3, MP5030 defines the transmission protocol.

MP5030.0-2	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: NEC P7PLUS printer with EXT1

The following setting is selected at the printer itself (see the operating manual of the printer concerned):

- Serial interface
- Data bits
- Even character parity
- XON/XOFF protocol (software handshake)
- 9600 baud

The following settings (EXT1) are made at the TNC:

MP5000 = 0	No interface inhibited
MP5020.0 = %10101001	8 data bits Any BCC character Transfer stop by RTS not active Transfer stop by DC3 active Character parity even Character parity required 1 stop bit
MP5030.0 = 0	Standard data transfer

In the "RS-232/RS-422 Setup" of the TNC, you must assign the EXT1 operating mode to the RS-232 interface and set the baud rate to 9600.

6.4 Data Transmission Protocols

The TNC enables your data and files to be transferred using different protocols.

The transmission protocols can be selected as follows:

– FE1, FE2 Transmission with Block Check Character and with fixed control characters (7 data bits, 1 start bit, 1 stop bit)

– EXT1, EXT2, EXT3 Freely configurable operating modes: using machine parameters you can select data format, transmission protocol and control characters.

– LSV2 Two-way transfer for TNC diagnostics and remote operation conforms to DIN 66019. This protocol always runs in the background of the TNC and can be started from a PC or from the TNC.

The following applies to 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. You can continue the transmission by pressing a soft key.
- If you attempt to erase write-protected files, the TNC displays an error message. You can remove the write protection and continue transmission by pressing a soft key.
- If a file has been read out and the data transfer menu has been terminated with the END key, the TNC outputs the characters <ETX> and <EOT>.

If a transmission is terminated with the END key, the error message **`Program incomplete`** is issued.

6.4.1 Standard Transmission Protocol

General information

You can set this protocol as standard in operating modes EXT1/EXT2/EXT3. In the following, the control characters that are sent and received with this protocol are listed for the various transmission alternatives. When outputting a file, the <NUL> character is sent exactly 50 times at the start of the file. When reading in, however, the control unit ignores this character. It is therefore insignificant how often the peripheral unit sends the <NUL> character before the file.

If you wish to signal an error to the TNC, you must send the following sequence of instructions:

<ESC> <1> <Error number>

The blocks are not checked for correctness but are transmitted one after the other. If the receiver's data buffer is full, it can stop the transfer and resume it in one of two ways:

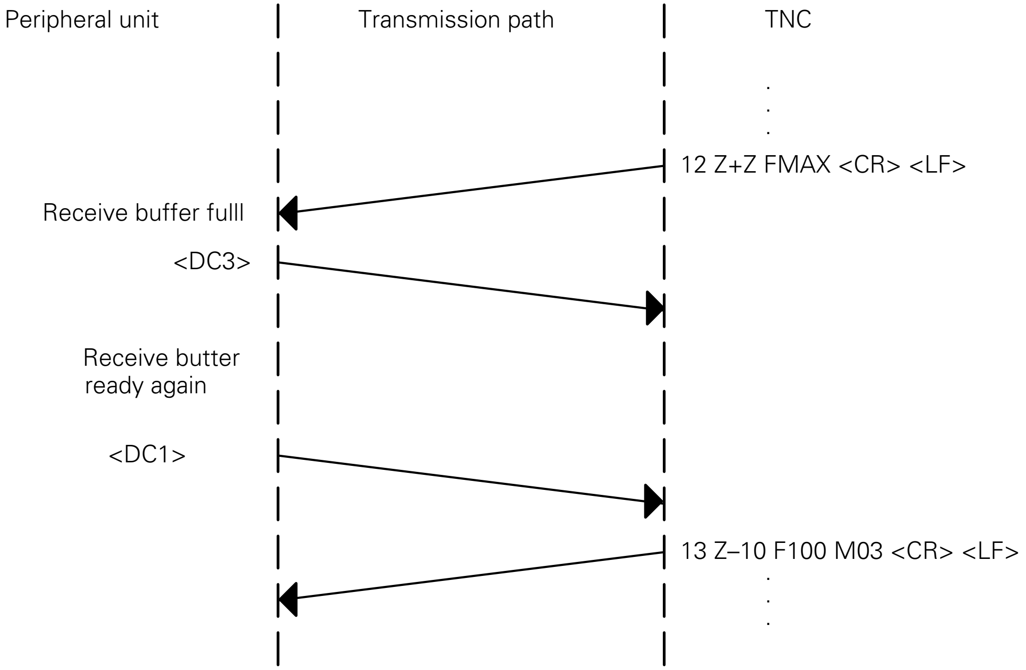
- Stop transfer by sending the character <DC3> (XOFF); continue by transmitting character <DC1> (XON) (software handshake).
- By suitable levels on the control and message lines RTS and CTS of interfaces RS-232-C/V.24 or RS-422/V.11 (hardware handshake).

Twelve characters before the receive 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><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
```

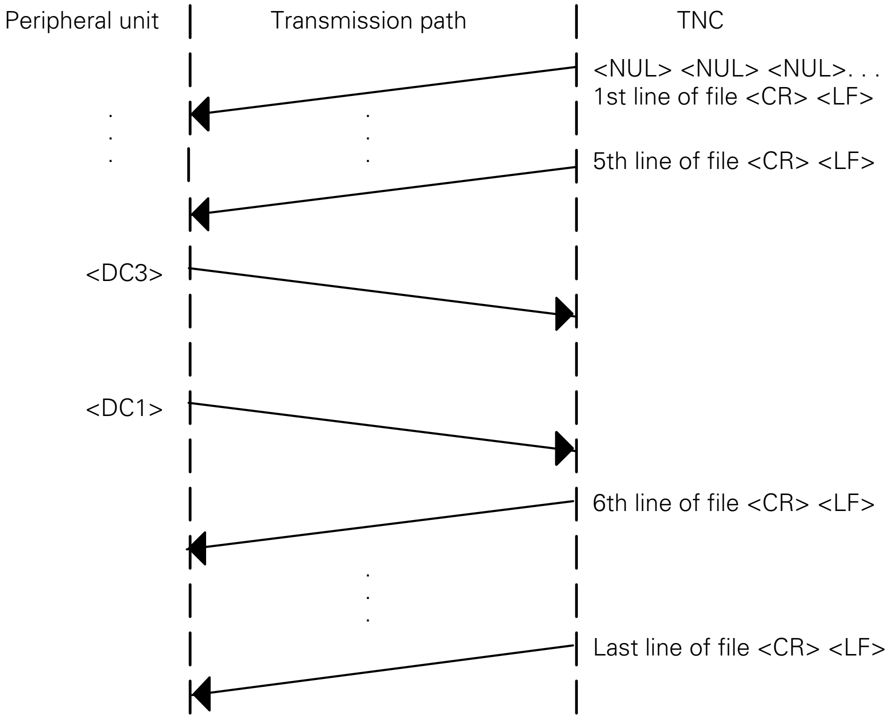
Example of software handshake:



The following section lists the transfer protocols for the various methods of data output and input. The EXT 1 mode is set with software handshake.

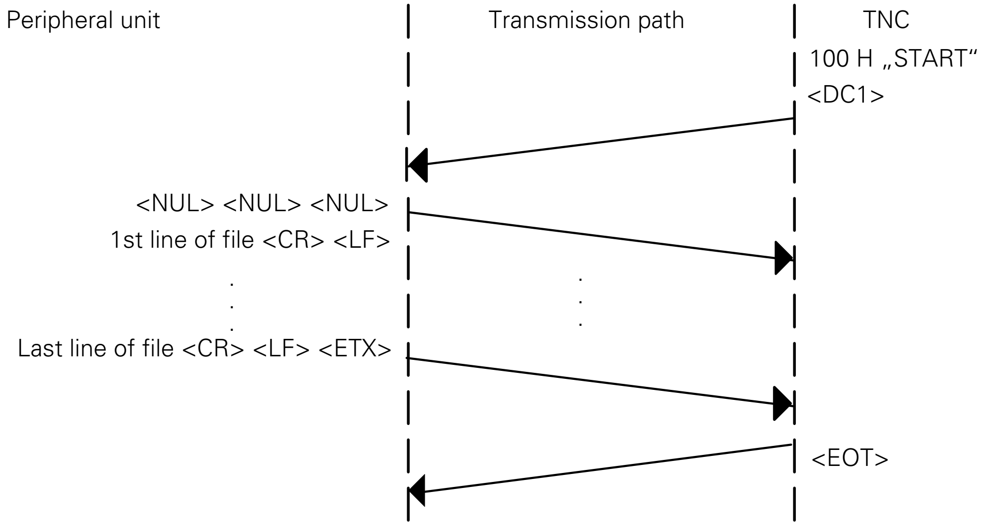
Output selected file

The TNC outputs all of the program lines in order. The peripheral unit can stop transmission with character <DC3> and start it again with character <DC1>.


Peripheral unit
Transmission path
TNC
<NUL> <NUL> <NUL>. . .
1st line of file <CR> <LF>
5th line of file <CR> <LF>
<DC3>
<DC1>
6th line of file <CR> <LF>
Last line of file <CR> <LF>

Read-in selected file

If you read-in a file from a peripheral unit (e.g. a PC), you must enter the corresponding name in the TNC.



In this transfer method, the TNC can stop transmission with <DC3> and continue it with <DC1>. If the file name in the first line of the file and the name indicated in the TNC are not identical, the TNC reads each block in and searches for the desired 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, and transfer must be terminated with the END key.

6.4.2 Data Transfer 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 fully identical for all these modes except for the FE1 mode, in which a command sequence is automatically output at the beginning to request the contents directory from the peripheral unit.

Header
When a file is transferred the first block — the so-called Header — is transmitted, consisting of the following characters:

<SOH><K><Name><M><ETB><BCC><DC1>

where:

- <SOH>: Start Of Header
- <K>: File code
- <Name>: File name
- <M>: Data transfer mode (E = input, A = output)
- <ETB>: End of Transmission Block — Indicates end of header
- <BCC>: Block Check Character
- <DC1>. XON

Block Check Character (BCC)
In addition to checking the parity of the individual characters (see chapter "Checking data") the parity of the complete transferred block is also checked. The BCC (Block Check Character) always rounds the individual bits of the transferred characters in a data transfer block to even parity.

Example of BCC generation:

Characters	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**

In this example, program 15, which has been written in HEIDENHAIN dialog (H) is input (E) through the data interface. A parity bit is also generated for the BCC (with even parity the BCC parity bit in this example is assigned the significance 1).

At the end of every block the receiver checks that it has 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 <ACK> character (= positive acknowledgment) to indicate that the data block has been transmitted without error.

If the two BCCs are not identical, then the receiver transmits the <NAK> character (= negative acknowledgment) to indicate that the data block was not transmitted correctly and must be re-transmitted. This process is repeated 15 times, then the error message **Transferred data incorrect E** is output and 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 always identified by the control character <STX>. The other control characters in this block are identical with the control characters in the header.

If the 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 transferred successfully (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. It is required by many devices to explicitly request an answer and to provoke the transmission once again.

You can suppress transmission of the <DC1> character in the EXT1, EXT2 and EXT3 modes by setting MP5020.x bit 3 = 0. The <DC1> character is not required for reading-in a file in the format with BCC.

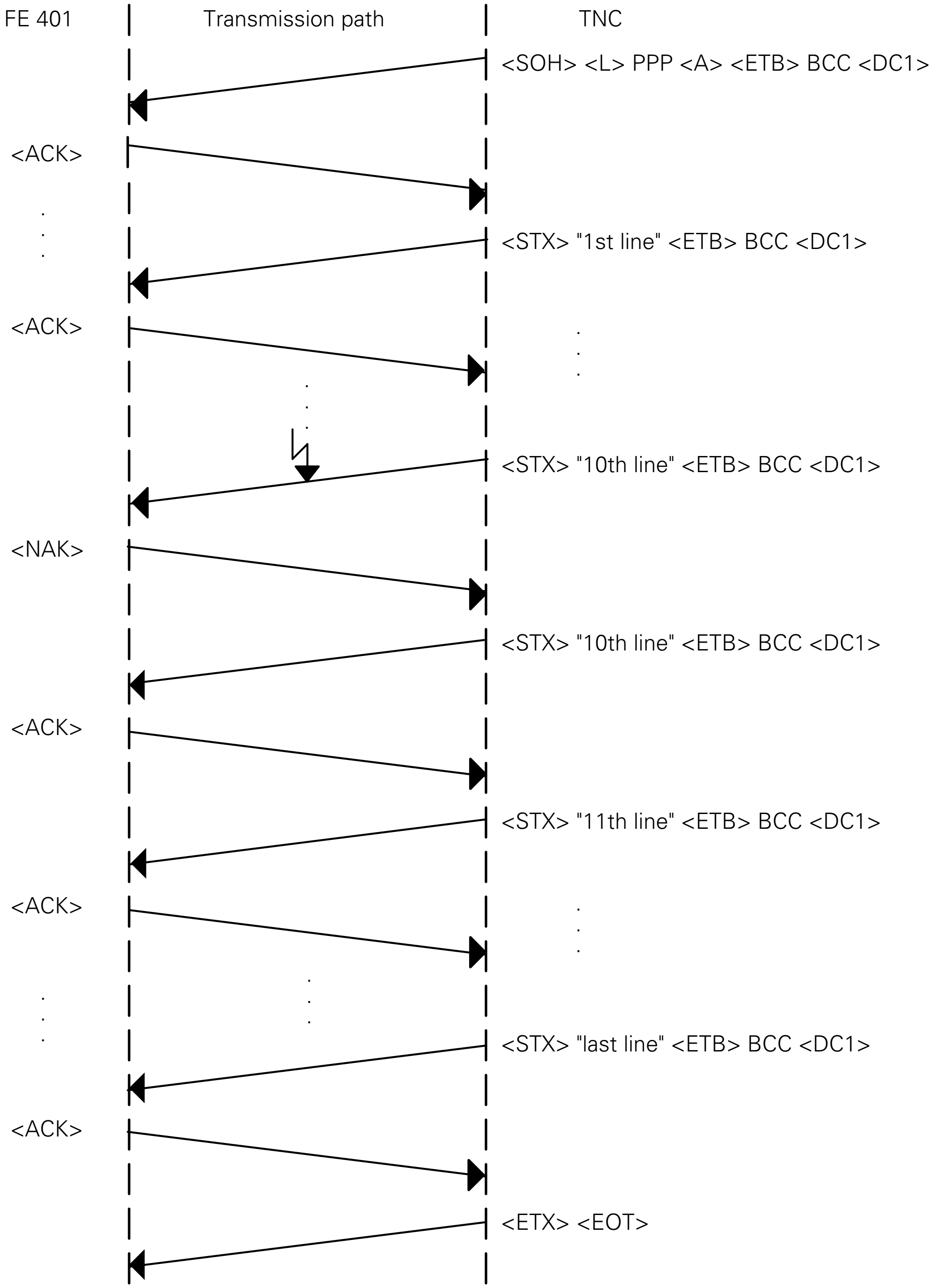
A software handshake is very easy to arrange when transmitting with BCC. The receiver transmits at first neither a positive acknowledgment (<ACK>) nor a negative one (<NAK>), and the transmitter waits to receive one of these characters. When the receiving device's receiving buffer is ready, it transmits an <ACK> and its partner resumes transmission.

Overview of control characters:

Character	Name	Description
SOH	Start of Header	SOH identifies the beginning of the data transfer header. This is a character string that contains the program number and information about the type of program and the transfer mode.
STX	Start of Text	STX identifies the beginning of a program block.
ETB	End of Text Block	ETB terminates a data transfer block. The character that follows ETB (BCC) is used for data checking.
DC1	Start data transfer (XON)	DC1 starts the transfer of data.
DC3	Stop data transfer (XOFF)	DC3 stops the transfer of data.
ETX	End of Text	ETX is transmitted at the end of a program.
EOT	End of Transmission	EOT terminates the data transfer and establishes the idle state. The 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	ACK is transmitted by the receiver when a data block has transferred without error.
NAK	Negative Acknowledgment	NAK is transmitted by the receiver when a data block has transferred with an error. The transmitter must re-transmit the block.

Example:

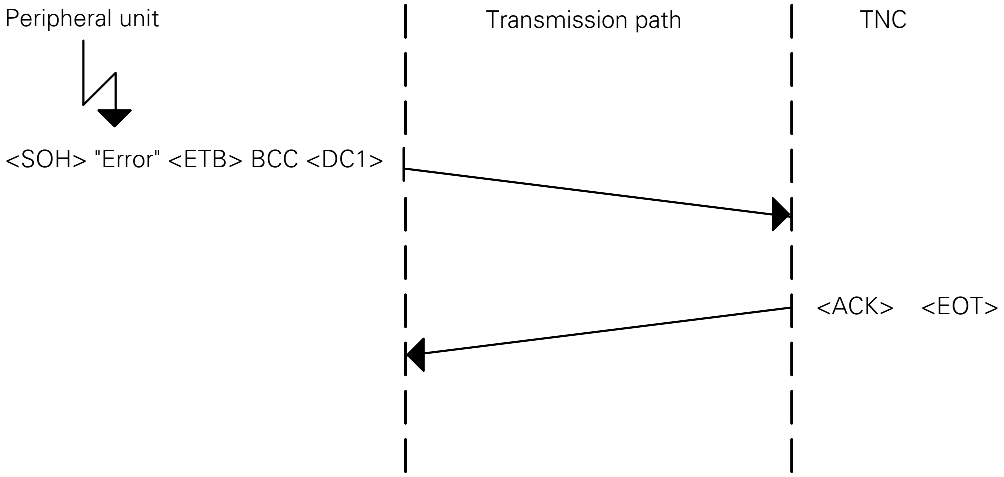
To read out a Pallet file with the name PPP to a peripheral device (e.g. FE 401).


FE 401
Transmission path
TNC
<SOH> <L> PPP <A> <ETB> BCC <DC1>
<ACK>
<STX> "1st line" <ETB> BCC <DC1>
<ACK>
<STX> "10th line" <ETB> BCC <DC1>
<NAK>
<STX> "10th line" <ETB> BCC <DC1>
<ACK>
<STX> "11th line" <ETB> BCC <DC1>
<ACK>
<STX> "last line" <ETB> BCC <DC1>
<ACK>
<ETX> <EOT>

In the following, the transmission protocols are listed for the various file input and output possibilities. **FE1** mode is set.

Report error to the TNC
If an error occurs at a peripheral device, the following block must be sent to the TNC:

<SOH>"Error text"<ETB>BCC



The received error message is displayed in the TNC, but can be acknowledged and erased with the CE key.

Request external directory
This protocol is not available in the FE2 and EXT mode. In FE1 mode, the following 'Escape' sequence is sent to request the external directory:

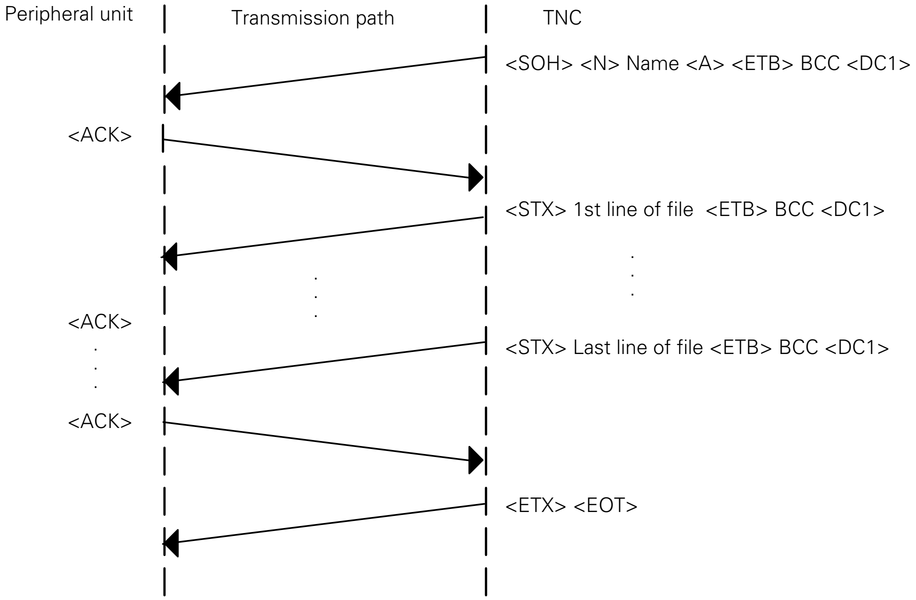
<DC3><ESC><DC1><0><SP><D><CR><LF>

The TNC expects the following input to this request: The first four lines, each ending in <CR><LF>, are ignored. In subsequent lines ending with <CR><LF>, only the program name and, after any number of blank characters, the number of occupied sectors are stored.

xxxxxx<Name>   <Sectors>xxxxxx<CR><LF>

If the character combination <FREE:> is detected, only a number (= number of free sectors) will be read in. The TNC always requests the complete directory. It then saves the directory, but displays only the files of the selected type.
The peripheral device ends transmission with <ETX>. In response the TNC sends an <EOT>.

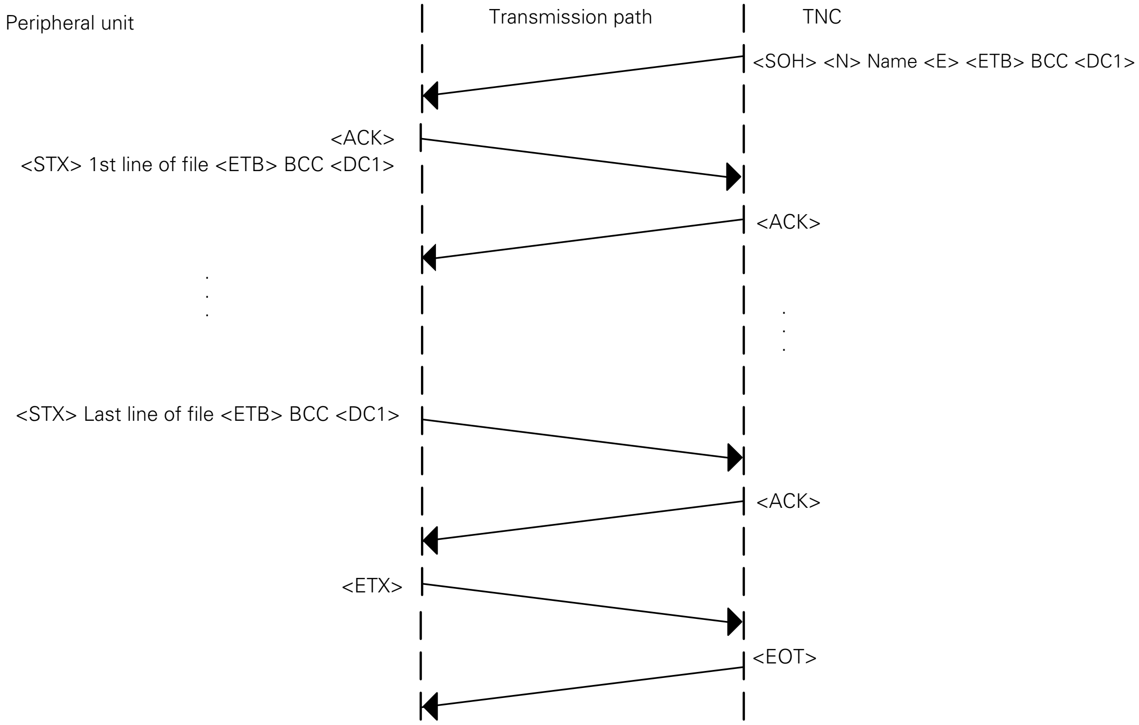
Output selected file


Peripheral unit
Transmission path
TNC
<SOH> <N> Name <A> <ETB> BCC <DC1>
<ACK>
<STX> 1st line of file <ETB> BCC <DC1>
<ACK>
<STX> Last line of file <ETB> BCC <DC1>
<ACK>
<ETX> <EOT>

Output marked files

The files are output in sequence in the same protocol as for outputting selected files. After each file the control characters <ETX><EOT> are sent to the peripheral device.

Read-in selected file
If a file is to be read in from an external memory, the TNC sends a header with the file name, whereupon the peripheral unit sends the file.


Peripheral unit
Transmission path
TNC
<SOH> <N> Name <E> <ETB> BCC <DC1>
<ACK>
<STX> 1st line of file <ETB> BCC <DC1>
<ACK>
<STX> Last line of file <ETB> BCC <DC1>
<ACK>
<ETX>
<EOT>

6.4.3 LSV2 Protocol

The LSV2 protocol is a data transfer protocol for the two-way transfer of commands and data according to DIN 66019. The commands and data are transferred in so-called telegrams, i.e. the data is split up into blocks (telegrams) and transmitted.

The following functions are possible:

- Data transfer
- File management (delete, copy and rename files)
- Write screen to a file (screen dump)
- Remote operation of the control functions, i.e. the TNC screen appears on the computer monitor and all TNC functions can be executed from the computer.
- Real DNC operation, i.e. 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).
- Changing, creating and deleting paths

HEIDENHAIN offer two LSV2 software packages:

- **TNC Remote:**
  Software for TNC remote control. Can be run on an AT compatible PC with MS-DOS. All the above functions are available with this software.
- **LSV2 TOOL BOX:**
  Software tools in C programming language for creating the transfer telegrams (library, executable files for telegrams, source codes, INCLUDE files for LSV2, MAKE files).

Timeouts
If you choose not to use the standard times for timeouts, you must define your own times 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 sent (standard 1 s)
Input range: 0.001 to 3.600 [s]
If the code words are not defined or if the value range is exceeded, the standard values are used.

6.5 Data Transfer by PLC

6.5.1 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.

When data is transmitted by the PLC, use of the interface is inhibited for the input/output program of the user interface. You select either a standard operating mode (FE1 or FE2) or you configure the data interface with MP5020.x to MP5040 in EXT3 mode.

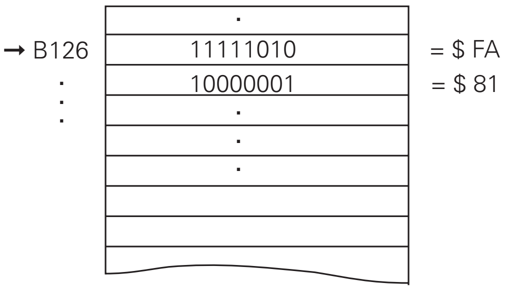
MP5040	Data transfer rate in operating mode EXT3 (data transfer via PLC)		
Input:	0 = 110 baud	5 = 2400 baud	10 = 57 600 baud
	1 = 150 baud	6 = 4800 baud	11 = 115 200 baud
	2 = 300 baud	7 = 9600 baud	
	3 = 600 baud	8 = 19 200 baud	
	4 = 1200 baud	9 = 38 400 baud	

6.5.2 PLC Modules

With PLC modules you can operate the data interfaces from the control. Modules 9100 and 9101 assign and release the data interfaces. With Module 9102 you interrogate the respective status of the data interface.

The transmit and receive buffers for the PLC are 128 characters long. Since every STRING ends with an END character a STRING in the transmit or receive buffer can only be up to 127 characters long. In addition to transmitting and receiving a STRING from the STRING memory (Modules 9103 and 9104), you can use Modules 9105 and 9106 to transfer a block of binary values (bytes) from the word memory. However, ASCII characters are transmitted and received through the interface in both cases (STRING and binary transmission).

Example: Transferring a binary block



When transferring binary data from the word memory from address B126 the ASCII characters <F><A><8><1> etc. are transmitted consecutively through the interface. Since each byte contains two ASCII characters when transferring binary data, the transmit and receive buffers hold 63 bytes. When transferring binary data you can use Module 9107 to read each byte (two ASCII characters) from the receive buffer without erasing the buffer.

Assign data interface (Module 9100)

Module 9100 configures the transfer parameters. It also 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 output program of the user interface. The assignment of an interface to the PLC is canceled when the PLC program is recompiled.

```
Call only in submit job:
PS   B/W/D/K     <Interface>        0:    RS232
                                    1:    RS422
PS   B/W/D/K     <Transmission parameters>
                                    0:    from MP50x0.2
                                    1:    from  MOD function
CM   9100
Error recognition: M4203 = 0:       Interface was assigned
                   M4203 = 1:       Error code in W1022
                   W1022 = 1:       Incorrect interface or incorrect transfer parameter
                   W1022 = 13:      No connection
                   W1022 = 14:      Interface already assigned or input/output not ready
                   W1022 = 17:      Incorrect baud rate
                   W1022 = 20:      Call was not from submit
```

Release data interface (Module 9101)

Module 9101 cancels the assignment of a serial interface to the PLC. The receive mode of the interface is canceled.

```
Call only in submit job:
PS   B/W/D/K     <Interface>        0:    RS232
                                    1:    RS422
CM   9101
Error recognition: M4203 = 0:       Interface was released
                   M4203 = 1:       Error code in W1022
                   W1022 = 1:       Incorrect interface
                   W1022 = 14:      Interface not assigned
                   W1022 = 20:      Call was not from submit
```

Status of data interface (Module 9102)

Module 9102 reads all the relevant status information about one of the two serial interfaces in bit-coded form.

The information "interface ready" is updated only when the interface is assigned to either the PLC or the NC. If the interface is not assigned, the module reads the status that was valid before the last interface release (whether by PLC or NC).

Call:

```
PS    B/W/D/K     <Interface>       0:    RS232
                                    1:    RS422
CM    9102
PL    B/W/D       <Interface status >
                  –1:     Incorrect module call
                  Bit 0:  Interface is assigned
                  Bit 1:  Interface is assigned to PLC
                  Bit 2:  Interface is ready (see above)
                  Bit 3:  Send 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: M4203 = 0:       Status read
                  M4203 = 1:        Error code in W1022
                  W1022 = 1:        Incorrect interface
```

Send string through data interface (Module 9103)

You must have already assigned the interface to the PLC and initialized it with MP 9100. Module 9103 transmits a string from one of the four string memories through one of the two serial interfaces. References to the PLC error file and PLC dialog file are deleted (see "PLC Error Messages").

Call only in submit job:

```
PS    B/W/D/K     <Interface>       0:    RS232
                                    1:    RS422
PS    K/B/W/D     <Number of source string in the string buffer (0 to 3)>
CM    9103
Error recognition: M4203 = 0:       String was sent
                  M4203 = 1:        Error code in W1022
                  W1022 = 1:        Incorrect interface or incorrect string
                  W1022 = 12:       String too long
                  W1022 = 13:       Interface not ready
                  W1022 = 14:       Interface not assigned
                  W1022 = 15:       Send buffer not empty
                  W1022 = 20:       Call was not from submit
```

Receive string through data interface (Module 9104)

Module 9104 reads a string from the receive buffer of a serial interface in one of the four string buffers and resets the receive buffer. You must have previously assigned the interface to the PLC and initialized it with MP 9100.

Call only in submit job:

```
PS    B/W/D/K     <Interface>       0:    RS232
                                    1:    RS422
PS    K/B/W/D     <Number of destination string in the string buffer (0 to 3)>
CM    9104
```

```
Error recognition: M4203 = 0:    String was received
                   M4203 = 1:    Error code in W1022
                   W1022 = 1:    Incorrect interface or incorrect string
                   W1022 = 12:   String too long
                   W1022 = 14:   Interface not assigned
                   W1022 = 16:   Receiving buffer empty
                   W1022 = 18:   Transfer error or input/output not ready
                   W1022 = 20:   Call was not from submit
```

Send binary data through data interface (Module 9105)
Module 9105 transmits a block of binary values from the word memory of the PLC to one of the two serial interfaces. The transfer is in the form of ASCII-coded hexadecimal values, so every byte in the source block makes two ASCII characters at the serial interface. You must have previously assigned the interface to the PLC and initialized it with MP 9100.

```
Call only in submit job:
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..63)>
CM   9105
Error recognition: M4203 = 0:    Data was sent
                   M4203 = 1:    Error code in W1022
                   W1022 = 1:    Incorrect interface or incorrect byte number or block too long
                   W1022 = 4:    Block outside value range
                   W1022 = 13:   Interface not ready or no connection
                   W1022 = 14:   Interface not assigned
                   W1022 = 15:   Send buffer not empty
                   W1022 = 20:   Call was not from submit
```

Receive binary data through data interface (Module 9106)
Module 9106 reads a block of binary values from one of the two serial interfaces to the word memory of the PLC. The transfer is in the form of ASCII-coded hexadecimal values, so 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. You must have previously assigned the interface to the PLC and initialized it with MP 9100.

```
Call only in submit job:
PS   B/W/D/K    <Interface>       0:   RS232
                                  1:   RS422
PS   K/B/W/D    <Number of first byte in binary block (0 to 1023)>
CM   9106
PL   B/W/D      <Length of binary block in bytes>        –1:        Incorrect module call
Error recognition: M4203 = 0:    Data was received
                   M4203 = 1:    Error code in W1022
                   W1022 = 1:    Incorrect interface or incorrect offset
                   W1022 = 4:    Block outside value range
                   W1022 = 11:   Odd number of characters or illegal character
                   W1022 = 12:   String too long
                   W1022 = 14    Interface not assigned
                   W1022 = 16:   Receiving buffer empty
                   W1022 = 18:   Transfer error or input/output not ready
                   W1022 = 20:   Call was not from submit
```

Read from the receiving buffer (Module 9107)

Module 9107 reads two ASCII characters from the receive buffer to one of the two serial 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 (i.e. half the offset in the ASCII string). The contents of the receive buffer are retained and can be read by Modules 9104 or 9106. You must have previously assigned the interface to the PLC and initialized it with MP 9100.

Call only in submit job:

```
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: M4203 = 0:     Receiving buffer was read
                   M4203 = 1:     Error code in W1022
                   W1022 = 1:     Incorrect interface or incorrect byte number or block too long
                   W1022 = 4:     Block outside value range
                   W1022 = 11:    Illegal character
                   W1022 = 12:    String too long or offset too large
                   W1022 = 14:    Interface not assigned
                   W1022 = 16:    Receiving buffer empty
                   W1022 = 18:    Transfer error or input/output not ready
                   W1022 = 20:    Call was not from submit
```

Send a message by LSV2 (Module 9110)

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 doubleword
                1:     String
PS   B/W/D/K    <Source address>
                With binary: number of doubleword (0 to 1020)
                With string: number of string (0 to 3)
CM   9110
PL   B/W/D      <Error code>
                0:     Message is being transmitted
                1:     No connection to host
                2:     Send buffer full
                3:     Wrong data type (not 0 or 1)
                4:     Wrong source address
Error recognition: M4203 = 0:     Message was transmitted
                   M4203 = 1:     Error code in W1022
                   W1022 = 2:     Incorrect data type
                   W1022 = 4:     No double word or incorrect string
                   W1022 = 11:    String too long
                   W1022 = 13:    No connection
                   W1022 = 15:    Send buffer not empty
                   W1022 = 16:    Receiving buffer empty
```

Receive a message by LSV2 (Module 9111)
Module 9111 reads a message (doubleword or string) that has been received from a host computer connected by LSV2 protocol. The message must be sent from the host by the LSV2 command "M PC<msg.I>".

```
Call:
PS    B/W/D/K     <Data type>
                  0:      Binary data doubleword
                  1:      String
PS    B/W/D/K     <Target address>
                  For binary: number of doubleword (0 to 1020)
                  For string: number of string (0 to 3)
CM    9111
PL    B/W/D       <Error code>
                  0:      Message was read
                  1:      No connection to host
                  2:      No message of this type in receive buffer
                  3:      Wrong data type (not 0 or 1))
                  4:      Wrong target address
Error recognition: M4203 = 0:     Message was received
                  M4203 = 1:      Error code in W1022
                  W1022 = 2:      Incorrect data type
                  W1022 = 4:      No double word or incorrect string
                  W1022 = 11:     String too long
                  W1022 = 13:     No connection
                  W1022 = 15:     Send buffer not empty
                  W1022 = 16:     Receiving buffer empty
```

Send ASCII characters via data interface (Module 9112)
The interface must already have been assigned to the PLC and initialized with Module 9100.
Send a single ASCII character with Module 9112. You need to set MP5030.2 = 2, so that the transmitted characters do not disturb the set protocol procedure. At least define the characters as a word so that the values to 255 can be recognized.

```
Call only in submit job:
PS    B/W/D/K     <Interface>     0:   RS232
                                  1:   RS422
PS    K/W/D       <ASCII code>    [0 to 255]
CM    9112
Error recognition: M4203 = 0:     Character was sent
                  M4203 = 1:      Error code in W1022
                  W1022 = 1:      Incorrect interface
                  W1022 = 13:     Interface not ready or no connection
                  W1022 = 14:     Interface not assigned
                  W1022 = 15:     Send buffer not empty
                  W1022 = 20:     Call was not from submit
```

Receive ASCII characters via data interface (Module 9113)

With Module 9113 read a single ASCII character from the receiving buffer of a serial interface and reset the receiving buffer. The interface must already have been assigned to the PLC and initialized with Module 9100. 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. Please store the result in a word at least, so that the values to 255 will be recognized.

Call only in submit job:

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:	M4203 = 0:	Character was received
	M4203 = 1:	Error code in W1022
	W1022 = 1:	Incorrect interface
	W1022 = 12:	String too long
	W1022 = 13:	Interface not ready or no connection
	W1022 = 14:	Interface not assigned
	W1022 = 16:	Receiving buffer empty
	W1022 = 18:	Transfer error or input/output not ready
	W1022 = 20:	Call was not from submit
	W1022 = 37:	Receiver queue full

6.6 External Programming

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 <ETX> 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.
- Block numbers need not be programmed. They are generated by the TNC (only for conversational programming).

6.7 Error Messages

6.7.1 TNC Error Messages

Listed below are the error messages for data transfer that are displayed by the TNC. In most cases the messages are self-explanatory.

General error messages:

`Interface already assigned`	Transfer is already taking place via interface, or data transfer has not been completed.
`Program incomplete`	A transfer has been interrupted or the file has not ended correctly (no END character or END block).
`Ext. output/input not ready`	Interface is not connected; peripheral unit is switched off or faulty.
`Transferred data incorrect X`	x = error code (see table)

Error codes:

E	During data transfer with Block Check Character (BCC), a "Not Acknowledged" control character (<NAK>) has been received 15 times in a row.
A to H except E	The receiving unit has detected an error from one of the following causes: • Different baud rates are set at TNC and peripheral unit • Parity bit wrong • Incorrect data frame (e.g. no stop bit) • Receiving component of interface faulty
K	During the transmission of an error to the TNC the <1> character was not transmitted after the <ESC> character.
L	An incorrect error number was received after the <ESC><1> error sequence (error numbers 0 to 7 are permitted).
N	An expected <ACK> or <NAK> acknowledgment was not transmitted within a certain time.
M	During data transfer with Block Check Character (BCC), a "Not Acknowledged" control character (<NAK>) has been transmitted 15 times in a row.

Error codes K and L are displayed only for transfer with the standard data transmission protocol.

6.7.2 Error Codes for HEIDENHAIN Peripherals

FE 401
The TNC display the following error messages:

Error code	Meaning
ERR: 001	Wrong instruction code
ERR: 002	Illegal program name
ERR: 003	Faulty data transmission
ERR: 004	Program not complete
ERR: 010	Program not on floppy disk
ERR: 011	Program is protected against erasure
ERR: 012	Program storage in progress
ERR: 013	Program directory full
ERR: 014	Floppy disk is full
ERR: 100	Floppy disk not formatted
ERR: 101	Sector number too large
ERR: 102	Drive not ready
ERR: 103	Floppy disk is write-protected
ERR: 104	Data on floppy disk are faulty
ERR: 105	Sectors not found
ERR: 106	Checksum erroneous
ERR: 107	Disk controller faulty
ERR: 108	DMA faulty

HEIDENHAIN data transmission software TNC.EXE

If data is transferred using the HEIDENHAIN TNC.EXE data transmission program, the following error messages might be displayed at the TNC. A comprehensive description of this software in the User's Manual for TNC.EXE.

Error message	Meaning
`Transferred data incorrect`	Attempt to transmit block to control unit has failed four times.
`Search feature not allowed`	Search feature not included in the set of permissible characters.
`Instruction not allowed`	The request instruction issued by control unit is not allowed.
`Program not present`	File requested by control unit does not exist in currently configured access path
`File name not program name`	Name of NC program and name of file do not match.
`Program incomplete`	NC program does not contain an end block.
`Protected file!`	File that is protected with read-only or hidden attribute is about to be overwritten.
`Data carrier @: is full`	Data carrier >@:< full.

6.8 Ethernet Interface (Option)

HEIDENHAIN offers an Ethernet interface as an option on the TNC. It can be ordered under the following part identification number.

Ethernet interface:	Id. Nr. 293 890 51

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 highly dependent on the amount of traffic at the time on the net.

Realistic values:	NC program to 200 Kbits/s, ASCII program to 1 Mbit/s

6.8.1 Hardware

The integrated Ethernet expansion card provides you with both the 10Base2 (BNC) port and the 10BaseT (twisted pair). You can use only one of the two ports. Both ports are metallically isolated from the control electronics. For the pin layout, see the chapter "Mounting and Electrical Installation."

X26 Ethernet interface BNC port (coax cable 10Base2)
The 10Base2 port is also known as **Thin-Ethernet** or **CheaperNet.** You connect the TNC with your network via BNC-T connector. The maximum cable length is 185 m (coax cable). The network topology is a linear bus. The "open" ends of the bus must be terminated with terminating resistors.

X25 Ethernet interface RJ45 port (10BaseT)
You can realize the twisted-pair cable on the 10Base-T connection either as shielded or unshielded.

Maximum cable length:	Unshielded:	100 m
	Shielded:	400 m

The network topology is a star configuration. This means central node establishes the connection to the other participants.

6.8.2 Software

In order for the TNC to be able to communicate with the remote station, the remote station must work according to the TCP/IP protocol principle. It must also be a NFS server (Network File System).

	OSI 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	

The TNC must be properly configured to be able to communicate with your network. Your network supervisor can provide you with the information you need to make these configurations. To access the settings, enter **NET123** on the TNC.

DEFINE NET	Settings on the TNC for networking.	
	ADDRESS	**Internet address of the TNC**: Enter the address as four decimal characters 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 (10BASET / 10BASE2):** Here you define which of the two ports you wish to use for your network.
	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 identification, the entries under UID, GID, DCM and FCM will be ignored.
DEFINE MOUNT	Here you define 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.
PATH	**Directory**: Enter here the complete directory (note the proper capitalization) of the NFS server that you wish to mount.
UID	**User ID**: Enter here the user identification for accessing the files in the network. The entry must be a decimal number.
GID	**Group ID**: Enter here 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 ..:	
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. If you work with null identification, the entries under UID, GID, DCM and FCM will be ignored.*

DEFINE PRINT	Define the name and address of the network printer. You can print directly from the TNC on the printer 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 keys have been activated.
PRINTER NAME	**Printer name:** Name of the printer for the printer server.

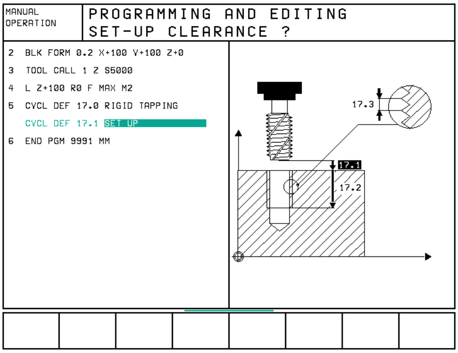
SHOW ERROR	Any errors occurring during network operation are displayed.
PING	If a ping is sent, the receiver sends it back to the sender. Thus a ping can be used to check whether a connection to a particular remote station is possible. The address is input as four decimal values separated by points (dotted 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

7 Original Equipment Manufacturer's (OEM) Cycles

7.1 HEIDENHAIN Standard Cycles

Many common machining tasks requiring several routine steps are stored ready-programmed in the TNC as fixed cycles. Instead of programming the task in all its steps, the user need only enter the dimensions and other parameters specific to the job at hand. Coordinate transformations and several special functions are also available as cycles.

The cycles are divided into groups and are called by soft key. Some cycles take effect immediately upon definition, while others must first be called with CYCL CALL. The TNC graphically illustrates the type of information required in the input parameters.



```
MANUAL       PROGRAMMING AND EDITING
OPERATION    SET-UP CLEARANCE ?

2   BLK FORM 0.2 X+100 Y+100 Z+0
3   TOOL CALL 1 Z S5000
4   L Z+100 R0 F MAX M2
5   CYCL DEF 17.0 RIGID TAPPING
    CYCL DEF 17.1 SET UP
6   END PGM 9991 MM
```

The User's Manual for the TNC provides a comprehensive description of the HEIDENHAIN fixed cycles.

7.2 CycleDesign

HEIDENHAIN offers a PC program called **CycleDesign** that enables you to add your own cycles, help graphics, and soft keys to the HEIDENHAIN standard cycles. The CycleDesign software includes all files from the HEIDENHAIN standard cycles, so that you can change the arrangement of soft keys, remove cycles, reorganize cycle groups, and add cycles that you have developed yourself. With its soft-key editor you can define you own soft-key contents. The help graphics for input parameter illustration can be created with any commercially available graphics program (not included), such as AutoSketch, that can export its files in DXF format. In machine parameter MP7364.x you can define the colors in which the graphic will be displayed on the TNC.

7.3 Application of OEM Cycles

The actual OEM cycle is a HEIDENHAIN conversational program that uses variables (Q parameters). You write and test the program on a TNC and define its transfer parameters using CycleDesign. Q parameters Q200 to Q299 are reserved as transfer parameters for OEM cycles.

Please note that Q200 to Q248 are assigned to the HEIDENHAIN standard cycles. For transfer parameters with the same meaning you will of course want to use the same parameter numbers. The TNC User's Manual provides comprehensive information on programming with Q parameters.

Functions that are not permitted in OEM cycles

- M functions M02, M30, M06 with program stop
- Programmed STOP call
- Program calls with PGM CALL
- Definition of Cycle 14 "Contour Geometry"
  Cycle 14 must be defined in the main program.

Q parameters with special meanings

Q parameters Q100 to Q 199 are reserved for special functions on the TNC. Many of these Q parameters have a special meaning such as tool radius, tool axis, etc. For a description of these parameters refer to the TNC User's Manual.

Global and local Q parameters

The content of the global Q parameters can be changed both in the calling NC program as well as in the OEM cycle being called. However, the content of the global Q parameter is not changed during transfer to and from the program being called.

Local Q parameters keep there contents only in the current program. When an OEM cycle is called, all contents of the local Q parameters are buffer stored and the same value is reassigned after they return from the OEM cycle. Any change in a local Q parameter in an OEM cycle is therefore effective only in the OEM cycle.

Q parameter	Effect
Q0 to Q59	Local
Q60 to Q99	Depends on MP7251
Q200 to Q299	Global

Because they are always local and can be changed within the program, Q parameters Q0 to Q59 are used in HEIDENHAIN standard cycles for mathematical operations. We recommend this for your OEM cycles as well.

MP7251 Definition of global Q parameters Q60 to Q99
Input: 0 = Q60 to Q99 local
1 to 40 = Q(100 – <Input value>) to Q99 global

FN functions with special meanings

With the following FN functions you can solve various tasks such as outputting error messages transferring data from the NC to the PLC:

- FN14 Output of error messages and dialogs on the screen.
- FN15 Output of error messages, dialogs and Q parameters to a file or through a data interface
- FN17 Overwrite system data: see "PLC programming"
- FN18 Read system data
- FN19 Assignment of two numerical values or Q parameters values from the OEM cycle to the PLC: see "PLC Programming."
- FN20 Wait for condition to become valid
- FN25 Overwrite 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 after definition. CALL-active cycles must first be activated with a separate CYCL CALL block. You can call DEF-active cycles from other OEM cycles, but not CALL-active cycles.

Loading the OEM-specific cycle structure

- ➤ On you PC, use CycleDesign to link the OEM cycles that you have developed with your soft keys and help graphics.
- ➤ Use CycleDesign to transfer these new cycle structures to the TNC.
- ➤ The TNC opens the system file `PLC:\CYCLE.SYS`. This system file contains the definition of the directories and files of the OEM cycles, soft keys and help graphics. 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 cannot find `PLC:\CYCLE.SYS` it uses the standard HEIDENHAIN cycle structures.
- ➤ Remember to use TNCBACK.EXE to make a backup copy of the PLC partition on a floppy disk, and to ship it together with your machine tool.

Example: HEIDENHAIN Standard Cycle 201 REAMING

The new HEIDENHAIN Standard Cycles (beginning from Cycle 200) are written like OEM cycles.

```
0  BEGIN PGM 201 MM P
1  FN 17: SYSWRITE ID212 = +3               Automatic compensation of 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 LBL 198       Inquiry whether tool is active
7  FN 14: ERROR = 1001                      Error message tool axis is missing
8  LBL 198
9  FN 12: IF +Q201 LT +0 GOTO LBL 197       Inquiry, whether the machining direction is
                                            negative
10  FN 0: Q30 = -1                          If not, the set constant to negative
11  FN 9: IF +0 EQU +0 GOTO LBL 194
12  LBL 197
13  FN 0: Q30 = +1                          Otherwise set constant to positive
14  LBL 194
15  FN 9: IF +Q97 EQU +1 GOTO LBL 193       Inquiry whether signs already were 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 to enter a 2nd safety clearance
20  FN 1: Q19 = +Q203 + +Q204               If so, set a new Z position
21  LBL 2
22  FN 1: Q20 = +Q203 + +Q200               Calculate Z pre-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 to enter feed rate for retraction
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 the sign was already reversed in the DEF cycle,
29  FN 3: Q200 = +Q200 * +Q30               skip negation
30  FN 3: Q204 = +Q204 * +Q30
31  LBL 0
32  LBL 192
33  L Z+Q20 R0 F MAX                        Move to pre-position in Z
34  L Z+Q24 R0 FQ206                        Move to total hole depth
35  CYCL DEF 9.0 VERWEILZEIT
36  CYCL DEF 9.1 V.ZEIT Q211                If desired, dwell at bottom
37  L Z+Q20 R0 FQ25                         Retraction to setup clearance
38  L Z+Q19 R0 F MAX                        If desired, retract to 2nd setup clearance
39  END PGM 201 MM P
```

7.4 Compatibility with “Old” OEM Cycles

Before the TNC 426 was introduced, OEM cycles were created directly at the TNC and then stored in the PLC EPROM. Such OEM cycles, which have the program numbers 99999968 to 99999999, are of limited use on the TNC 426, TNC 430.

In order to be able to run old NC programs in which these OEM cycles are called, you must use CycleDesign to save the old OEM cycles on the TNC's hard disk. CycleDesign provides a separate routine for this purpose. Such OEM cycles can be run on the TNC, but they cannot be defined. To enable your old OEM cycles to be defined on the TNC you must use CycleDesign to rewrite them in the new format.

8 Appendix

8.1 Error Messages

As of NC software 280 470 05 / 280 472 01
All NC error messages are displayed in dialog. With NC software 280 472 .. you can use the Help key to call up additional information on the error message shown.

Old/New error messages
In this new Technical Manual only the new dialog error messages are to be found throughout the text and in the Index at the end. These dialog error messages will be displayed only as of NC software version 280 470 05 and 280 472 01. If you are using one of the older controls, the following list may assist you by providing a cross reference between the old and the new error messages.

Old error message (coded)	New error message (dialog)
Gross positioning error <axis> A	Excessive servo lag 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 small <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 small
Positioning error	Excessive servo lag 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 1S	PLC: module 9008 not called
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

Old error message (coded)	New error message (dialog)
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 parenth 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: parenthesis 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
Error in PLC program 22	PLC: logic accum. 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 accum. 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

Old error message (coded)	New error message (dialog)
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: err. table format incorrect

Up to NC software 280 470 04
Part of the NC error messages is in coded form.

In the PLC editor or during compilation of the PLC program, errors are indicated by the message **`Input error X`***. During compilation of the program after the control is switched on, the flashing error message* **`Error in PLC program`** *is displayed.*

Classification of errors
Each error message indicates the time at which the error was recognized:

- *E recognized during editing, the line is not formatted.*
- *S recognized during syntax check in the PLC editor (soft-key compile).*
- *(S) recognized, under some circumstances, during the syntax check, otherwise during the compiler run.*
- *C recognized during the compiler run either after control switch-on or in the PLC programming mode.*
- *R recognized during the run time of the PLC program.*

Error Code	*Explanation*
0 E S C	*The line that was read cannot be interpreted as PLC command.*
2 E S C	*Invalid operand type* *An unknown operand type was entered.* *The command cannot be used with the entered operand type.*
3 E S C	*Operand not found. A type was entered for the operand, but no value.*
4 E S C	*Operand is outside of the permissible range. An operand number was entered that lies outside the value range available for this operand.*
5 E S C	*No limiter after the command. Additional characters found behind the PLC command cannot be interpreted.*
6 E S C	*End of line not found. The line is longer than 128 characters.*
7 S C	*Label not defined. A reference was made to a label that is not defined elsewhere with LBL, KFIELD or EXTERN.*

Error code	***Explanation***
8 S C	*No end of block found. At the end of the program file are PLC commands that are not concluded with an EM or JP command. The danger therefore exists that at run time an undefined program area will be run.*
9 S C	*Program too long (RAM overflow). The total length of the program code to be generated exceeds the memory space available in the control.*
10 S C	*Assignment to a parenthesis. An attempt was made to assign the result of a logic operation to an operand, although not all parentheses had been closed.*
11 S C	*Parentheses nested too deeply. An attempt was made to nest more than 16 parentheses.*
12 S C	*Jump within a logic chain. An unconditional jump was programmed, although the assignment chain previously begun was not yet assigned.*
13 S C	*Closing parenthesis without opening parenthesis. A "close parenthesis" command was programmed before writing the "open parenthesis" command.*
14 S C	*Label within a parenthesis. A label was set within a parenthesis gate. This is not permitted because "close parenthesis" commands cannot be executed before the corresponding "open parenthesis" commands.*
15 S C	*Label within a logic chain. A label has been programmed in a gate that has already begun. This is not permitted because the first command after the label would have to be interpreted either as a gate or as a load command depending on the program flow.*
16 S C	*Jump within a parenthesis. A jump command was programmed in a parenthesis. This is not possible because opened parentheses must always be closed again due to internal implementation, and this would not happen with the jump.*
17 S C	*Parenthesis opened at end of block. An EM instruction was programmed with parenthesis open. Parentheses must always be closed again.*
18 S C	*Label defined twice. A label name imported from another module with EXTERN was used again with an LBL or KFIELD instruction. A name reserved for internal modules (9000 – 9255) was used with an LBL, KFIELD or EXTERN instruction.*
19 S C	*No word assignment. A word gate was executed but the result had not been assigned to any operand, and a new gate had begun instead.*
20 S C	*No logic assignment. A logic gate was executed but the result had not been assigned to any operand, and a new gate had begun instead.*
21 S C	*Word accumulator not loaded. A command was programmed that gates, assigns or manipulates the already loaded word accumulator even though the accumulator was not previously loaded.*
22 S C	*Logic accumulator not loaded. A command was programmed that gates, assigns or manipulates the already loaded logic accumulator even though the accumulator was not previously loaded.*
23 S C	*Accumulators not loaded with open parenthesis. An "open parenthesis" command has been programmed even though neither a logic nor a word string was begun beforehand.*
24 S C	*Wrong type of parenthesis result. Depending on the gate formed before a parenthesis and the parenthesis command used, the system expects the string in the parenthesis to provide a result of the same type (word/logic). With different types the gate required by the "open parenthesis" command cannot be formed.*

Error code	***Explanation***
25 S C	*Conditional jump with invalid logic accumulator. A conditional jump was programmed (CMT/CMF/JPT/JPF/EMT/EMF), without first beginning a gating chain in the logic accumulator.*
26 S C	*ENDC/ENDK outside a CASE/KFIELD instruction. An ENDC command has been programmed without a prior CASE instruction. And ENDK command has been programmed without a prior KFIELD label.*
27 S C	*Wrong command within CASE table/KFIELD. A command other than CM was programmed after a CASE instruction and before the corresponding ENDC instruction. A command other than K was programmed after a KFIELD label and before the corresponding ENDK label.*
28 S C	*Too many table entries in CASE. A CASE table was programmed with more than 128 entries.*
29 S C	*Blank CASE instruction/KFIELD. A CASE instruction was programmed followed immediately by an ENDC label. A KFIELD instruction was programmed followed immediately by an ENDK label.*
30 S C	*String accumulator not loaded. A command was programmed that gates, assigns or manipulates the already loaded string accumulator even though the accumulator was not previously loaded.*
31 S C	*String instruction within parentheses. A string instruction was programmed within parentheses even though string gates cannot be nested with parentheses.*
32 S C	*No string assignment. A new gating chain was started without assigning the gating result previously formed in the string accumulator.*
33 S C	*GLOBAL/EXTERN not at start of file. The commands GLOBAL or EXTERN were written after another program code in the file. These commands must always come before the program code.*
34 (S) C	*Too many modules. An attempt was made to link more than 64 files into one program with the USES instruction.*
35 (S) C	*File not found. A file linked with USES cannot be found. An attempt was made to link a file of the .PLC type with MP4010=0 (EPROM).*
36 S C	*File too long. The program code for an individual file is bigger than 64 K and therefore cannot be compiled. The file must be split up into several files and linked with USES.*
37 S C	*Too many local labels. More than 1000 labels have been issued in a file. All LBL, KFIELD and EXTERN instructions are added together, also the (invisible) labels generated by structured commands. The file must be split up into several files and linked with USES.*
38 C	*Too many global labels. Over 1000 global labels were defined from all participating files.*
39 C	*External label not defined. A label declared as EXTERN was not defined as GLOBAL in any of the participating modules.*
40 S C	*External label in CASE instruction. A label declared as EXTERN was entered in the CM List of a CASE instruction. A local module must be defined that in the simplest case calls only the global module with CM.*
41 S C	*External label in JP instruction. An attempt was made to jump to a label defined as EXTERN with a JP/JPF/JPT instruction.*
42 (S) C	*Global label defined twice. The same label was defined as GLOBAL several times in the same or in different files.*

Error code	***Explanation***
43 S C	*Wrong instruction structure. An ELSE/ENDI/ENDW/UNTIL instruction was programmed without a preceding IF/ELSE/WHILE/REPEAT instruction. Instructions with different structures were interleaved instead of nested. The structures must always be closed in the reverse order in which they were opened!*
44 S C	*Structure open at end of file. A structured instruction was opened but not closed again by the end of file.*
45 S C	*GLOBAL instruction in main file. A module from the main file was defined as GLOBAL. Only modules from files linked with USES can be made accessible to other files using the GLOBAL instruction.*
50 R	*Nesting too deep. An attempt was made to nest more than 32 module calls. A recursive module call was programmed that exceeds the nesting depth limit of 32.*
51 R	*Stack Underflow. An attempt was made to fetch data from the stack even though they were not previously saved there.*
52 R	*Stack Overflow. An attempt was made to save 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 each.*
53 R	*Time Out. It took longer than 10.5 ms to process the program section that runs cyclically. Check the subprogram structure. It may be necessary to start very processor-intensive sections as SUBMIT jobs. The displayed computer time may be increased by RS-232-C transfers and handwheel mode. In case of doubt, select handwheel mode and start data transfer with RS-232-C (baud rate 115 000 if possible), then check "MAXIMUM PROCESSING TIME" in the PLC programming environment. 100% corresponds to 3.5 ms, the block processing speed is still achieved with this load. Values above 150% should not occur (safety margin for adverse operating conditions!).*
54 R	*CASE out of range. The operand for the CASE instruction contains a value that cannot be interpreted as an offset in the CM table (<0 or > table length –1).*
55 R	*Subprogram not defined. This error cannot occur at present.*
56 R	*Indexed access outside of the permissible range. The address for a write access to data types B/W/D/M/I/O/T/C is in a range invalid for this operand type owing to inclusion of the index register. While accessing a constants field the index register contains a value that is not possible for that field (<0 or > field length -1). The address of a string leads to a prohibited value owing to inclusion of the index register. The number of a dialog (S#Dn[X]) or an error message (S#En[X]) leads to a prohibited value owing to inclusion of the index register (<0 or >999). While addressing a substring (Sn^X) the value range for the index register (0..127) was exceeded.*
57 R	*No PLC error table. A PLC error module 9085/9086 was called although no error table was compiled, or the table contains no entries. A PLC error module 9085/9086 was called or an error marker was set, although the error table was changed or erased after compiling.*
58 R	*Fatal error during PLC module call. An MP was overwritten with the PLC module 9031 so that converting the MP resulted in an illegal value.*
90 C	*PLC error table. The 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 (file name or path incorrect).*
92 C	*PLC error table. The error table selected in OEM.SYS has no up-to-date binary format (e.g. software exchange).*

8.2 Tables

8.2.1 Seven-Bit ASCII Code

Character	Decimal	Octal	Hexadecimal
NUL	000	000	00
SOH	001	001	01
STX	002	002	02
ETX	003	003	03
EOT	004	004	04
ENQ	005	005	05
ACK	006	006	06
BEL	007	007	07
BS	008	010	08
HT	009	011	09
LF	010	012	0A
VT	011	013	0B
FF	012	014	0C
CR	013	015	0D
SO	014	016	0E
SI	015	017	0F
DLE	016	020	10
DC1 (X-ON)	017	021	11
DC2	018	022	12
DC3 (X-OFF)	019	023	13
DC4	020	024	14
NAK	021	025	15
SYN	022	026	16
ETB	023	027	17
CAN	024	030	18
EM	025	031	19
SUB	026	032	1A
ESC	027	033	1B
FS	028	034	1C
GS	029	035	1D
RS	030	036	1E
US	031	037	1F

Character	**Decimal**	**Octal**	**Hexadecimal**
SP	032	040	20
!	033	041	21
"	034	042	22
#	035	043	23
$	036	044	24
%	037	045	25
&	038	046	26
´	039	047	27
(	040	050	28
)	041	051	29
*	042	052	2A
+	043	053	2B
,	044	054	2C
-	045	055	2D
.	046	056	2E
/	047	057	2F
0	048	060	30
1	049	061	31
2	050	062	32
3	051	063	33
4	052	064	34
5	053	065	35
6	054	066	36
7	055	067	37
8	056	070	38
9	057	071	39
:	058	072	3A
;	059	073	3B
<	060	074	3C
=	061	075	3D
>	062	076	3E
?	063	077	3F
@	064	100	40
A	065	101	41
B	066	102	42
C	067	103	43

Character	Decimal	Octal	Hexadecimal
D	068	104	44
E	069	105	45
F	070	106	46
G	071	107	47
H	072	110	48
I	073	111	49
J	074	112	4A
K	075	113	4B
L	076	114	4C
M	077	115	4D
N	078	116	4E
O	079	117	4F
P	080	120	50
Q	081	121	51
R	082	122	52
S	083	123	53
T	084	124	54
U	085	125	55
V	086	126	56
W	087	127	57
X	088	130	58
Y	089	131	59
Z	090	132	5A
[	091	133	5B
\	092	134	5C
]	093	135	5D
^	094	136	5E
–	095	137	5F
`	096	140	60
a	097	141	61
b	098	142	62
c	099	143	63
d	100	144	64
e	101	145	65
f	102	146	66
g	103	147	67

Character	Decimal	Octal	Hexadecimal
h	104	150	68
i	105	151	69
j	106	152	6A
k	107	153	6B
l	108	154	6C
m	109	155	6D
n	110	156	6E
o	111	157	6F
p	112	160	70
q	113	161	71
r	114	162	72
s	115	163	73
t	116	164	74
u	117	165	75
v	118	166	76
w	119	167	77
x	120	170	78
y	121	171	79
z	122	172	7A
{	123	173	7B
¦	124	174	7C
}	125	175	7D
~	126	176	7E
DEL	127	177	7F

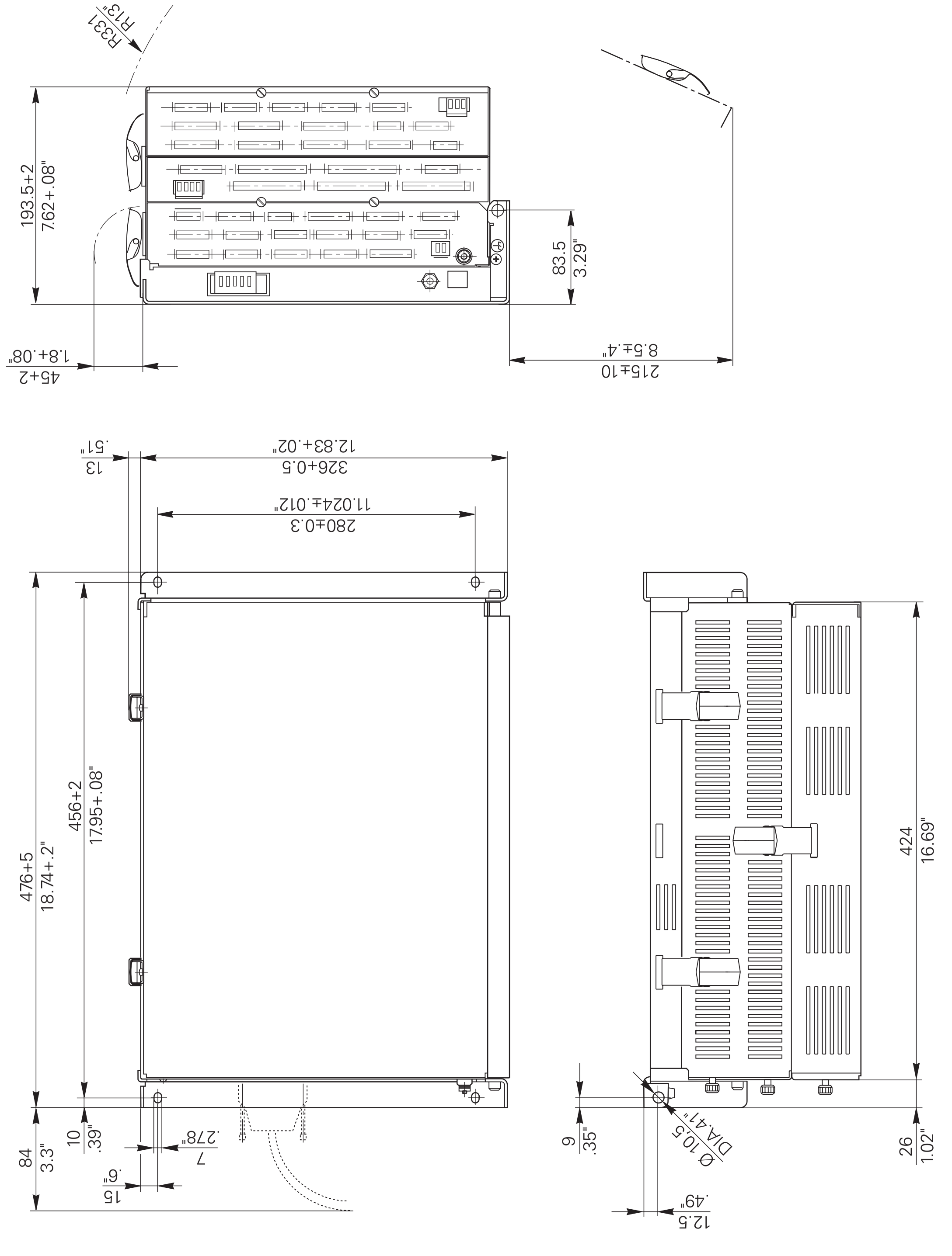
8.2.2 Powers of Two

n	**2^n**
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1 024
11	2 048
12	4 096
13	8 192
14	16 384
15	32 768
16	65 536
17	131 072
18	262 144
19	524 288
20	1 048 576

8.3 Dimensions

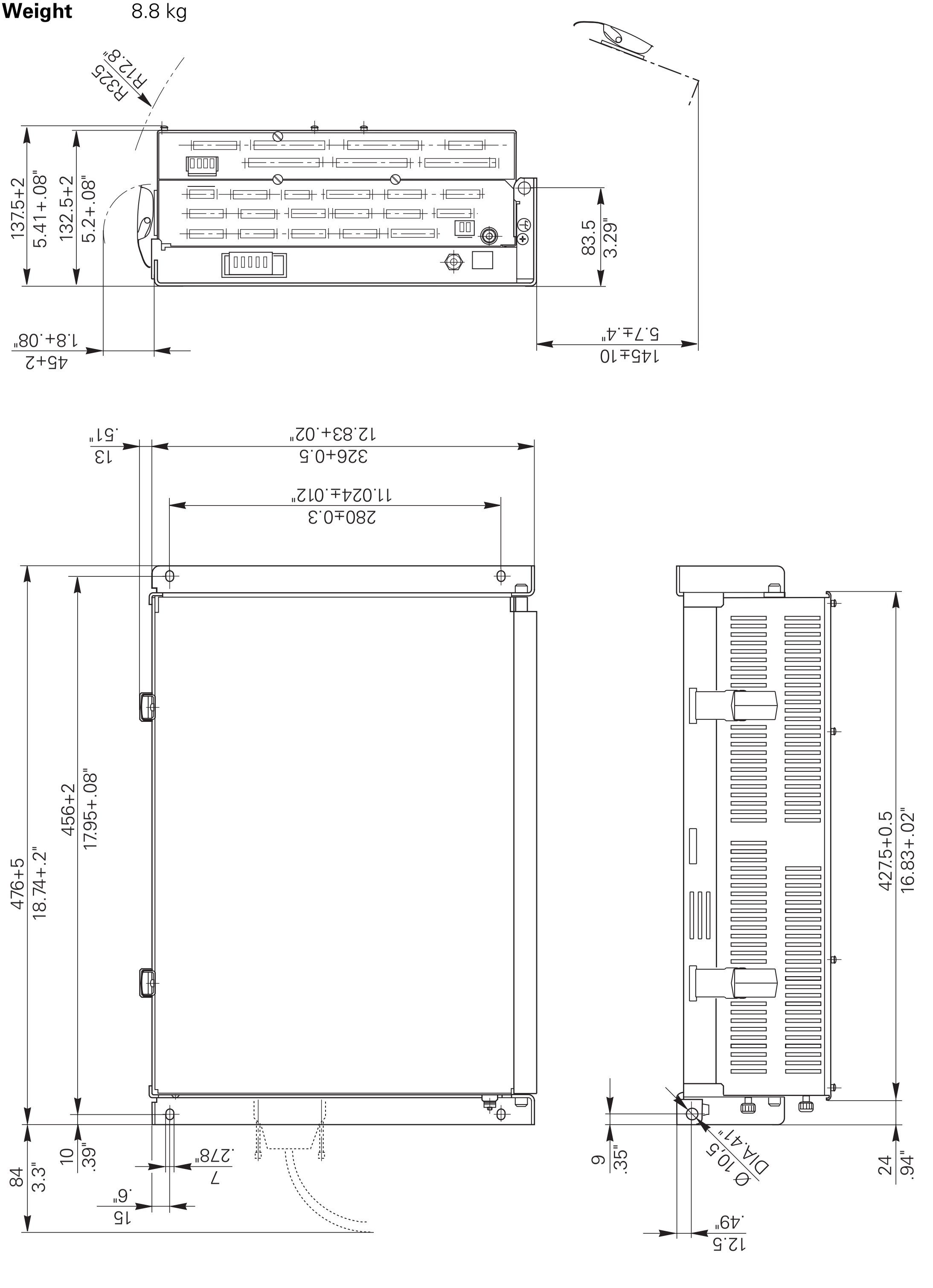
8.3.1 LE 426 PB, LE 430 PA

Weight 12 kg



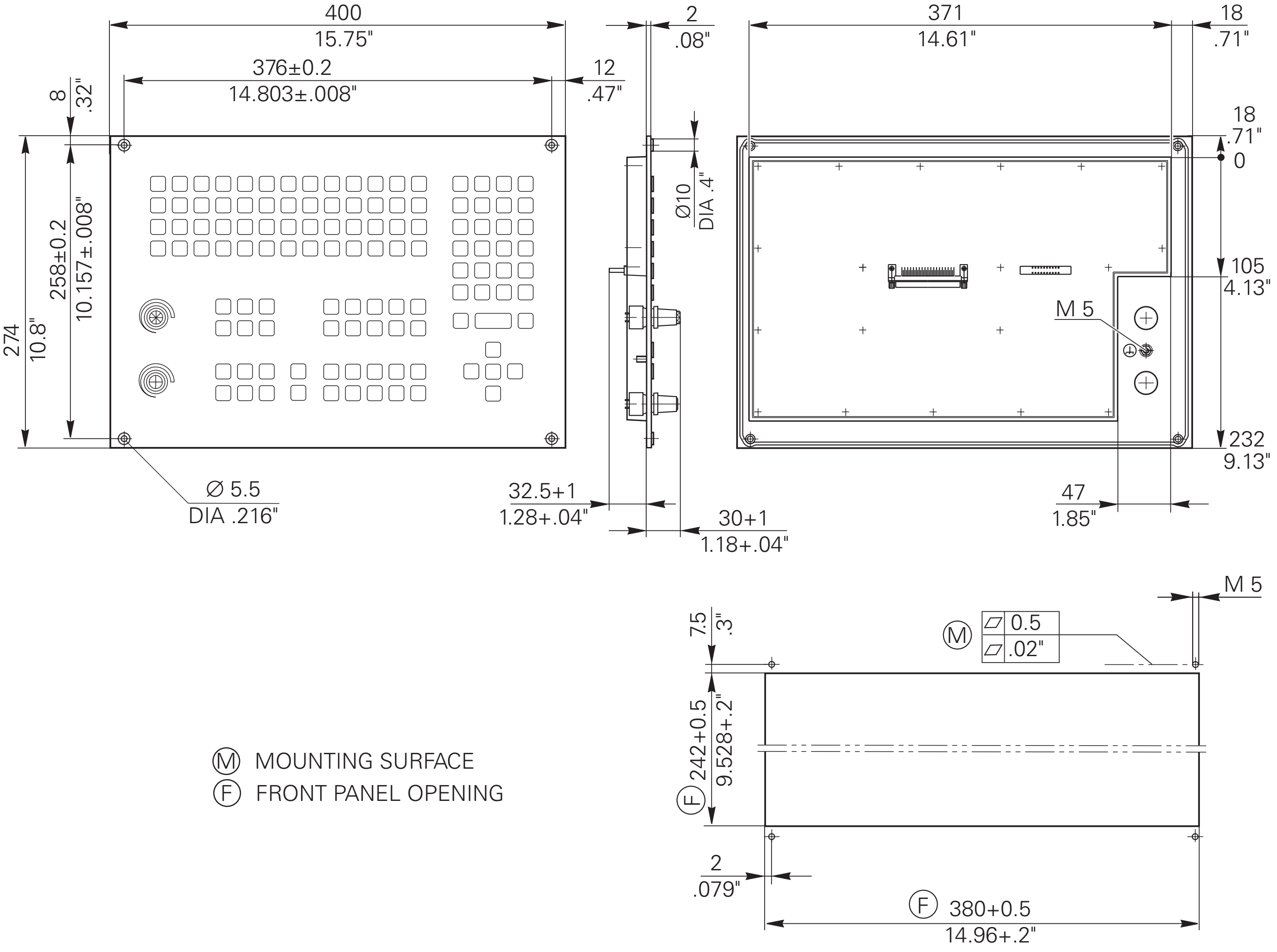
8.3.2 LE 426 CB, LE 430 PA

Weight 8.8 kg

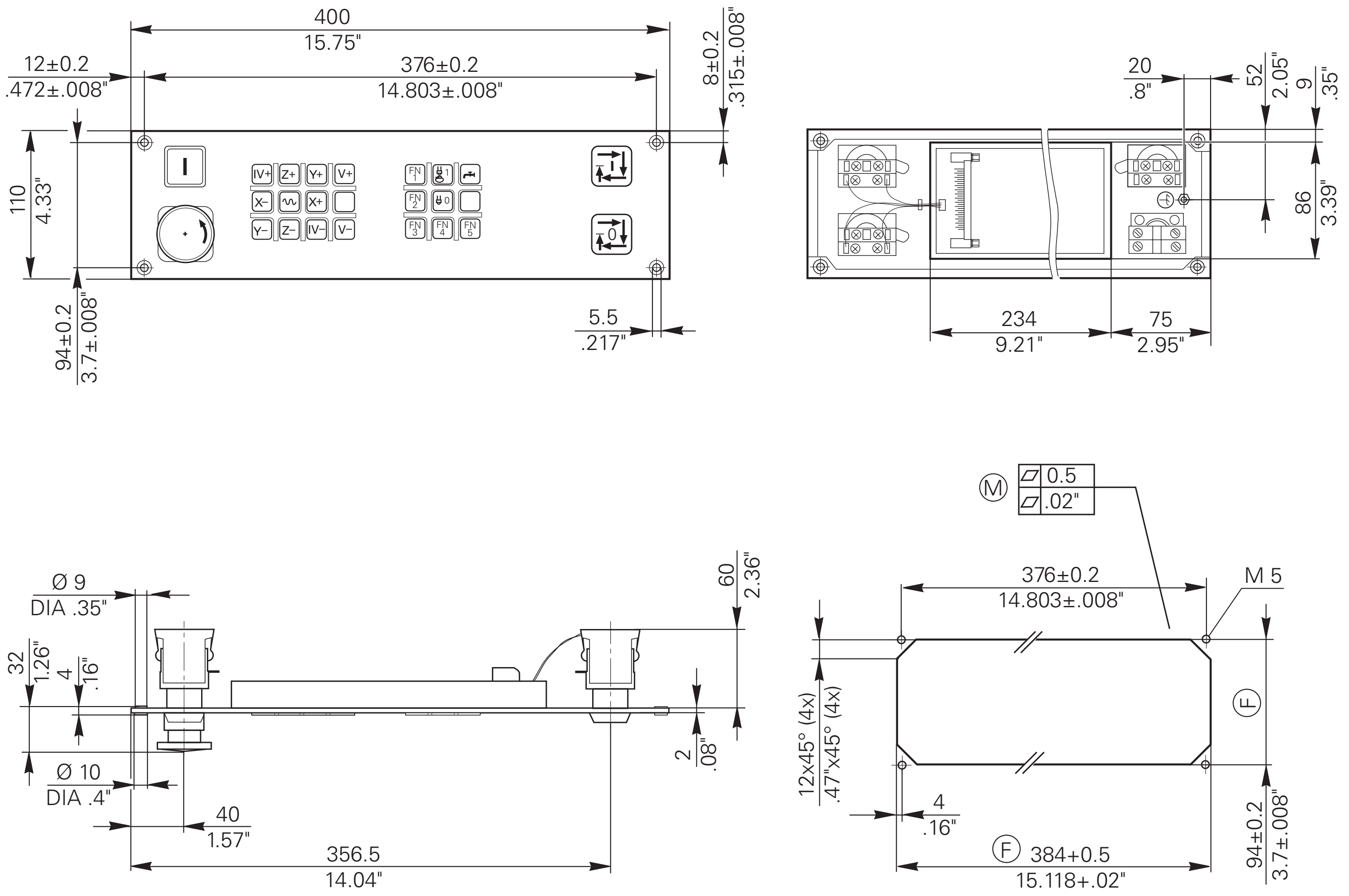


8.3.3 TE 420

Weight 2.4 kg



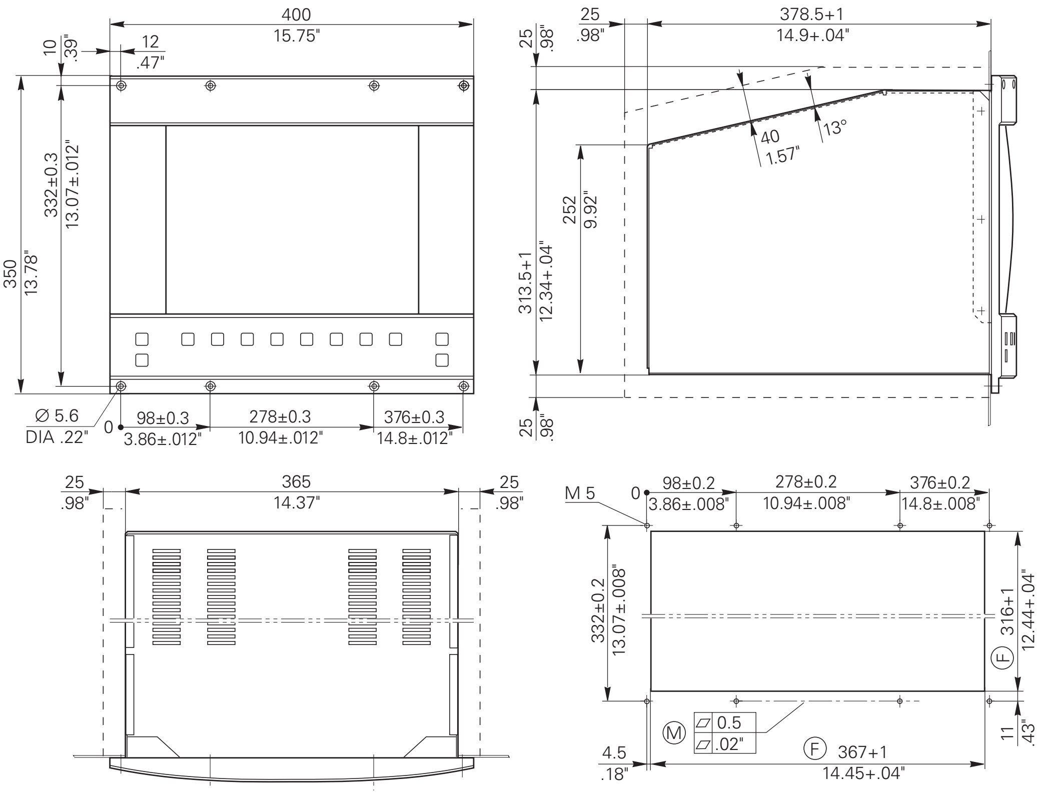
8.3.4 MB 420



400
15.75"
12±0.2
.472±.008"
376±0.2
14.803±.008"
8±0.2
.315±.008"
110
4.33"
94±0.2
3.7±.008"
5.5
.217"
I
IV+
Z+
Y+
V+
X–
X+
Y–
Z–
IV–
V–
FN 1
FN 2
FN 3
FN 4
FN 5
1
0
20
.8"
52
2.05"
9
.35"
86
3.39"
234
9.21"
75
2.95"
Ø 9
DIA .35"
32
1.26"
4
.16"
Ø 10
DIA .4"
40
1.57"
356.5
14.04"
60
2.36"
2
.08"
M
0.5
.02"
376±0.2
14.803±.008"
M 5
12x45° (4x)
.47"x45° (4x)
F
4
.16"
384+0.5
15.118+.02"
94±0.2
3.7±.008"

8.3.5 BC 120

Weight: Approx. 14 kg

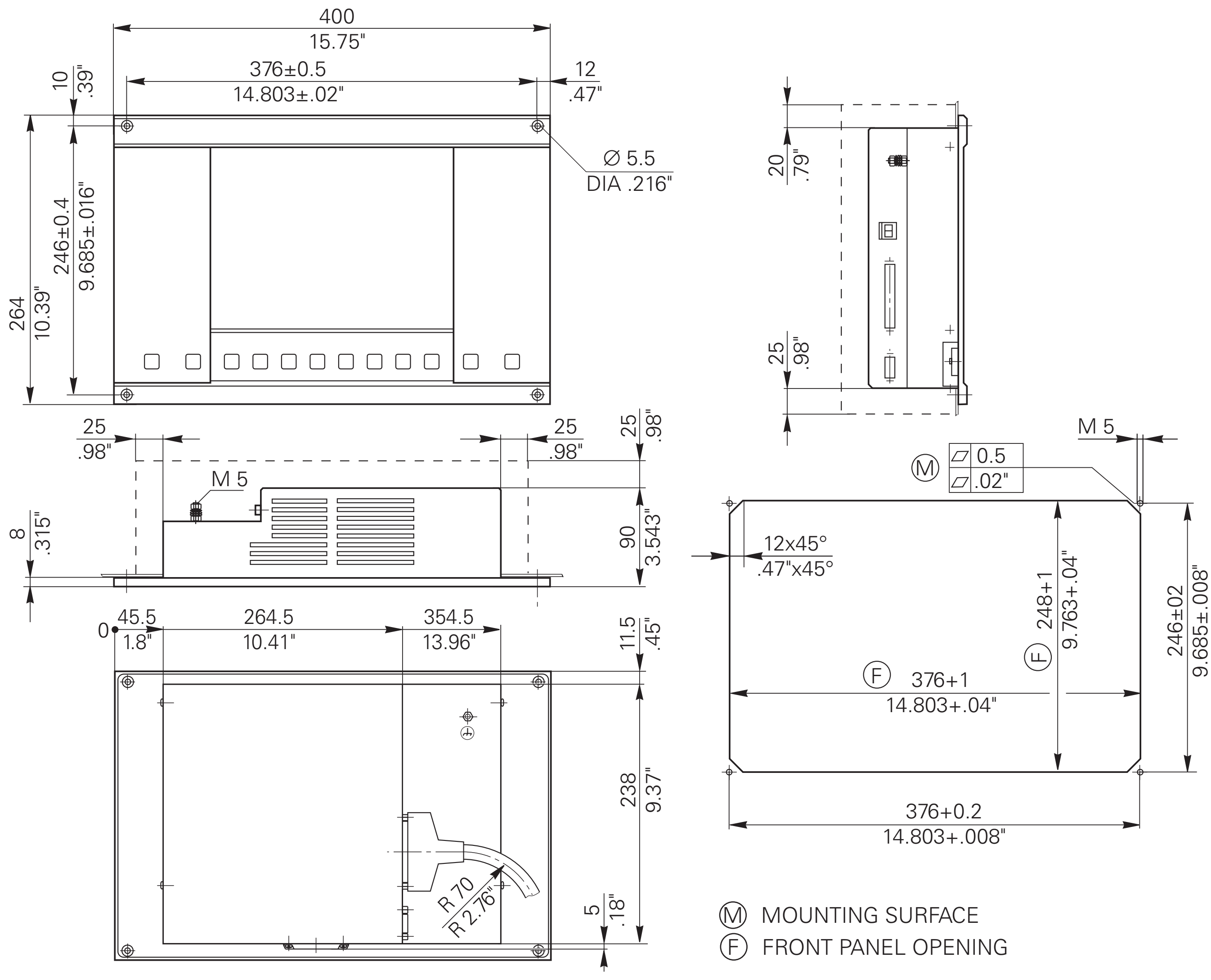


Ⓜ MOUNTING SURFACE

Ⓕ FRONT PANEL OPENING

8.3.6 BF 120

Weight: 3 kg

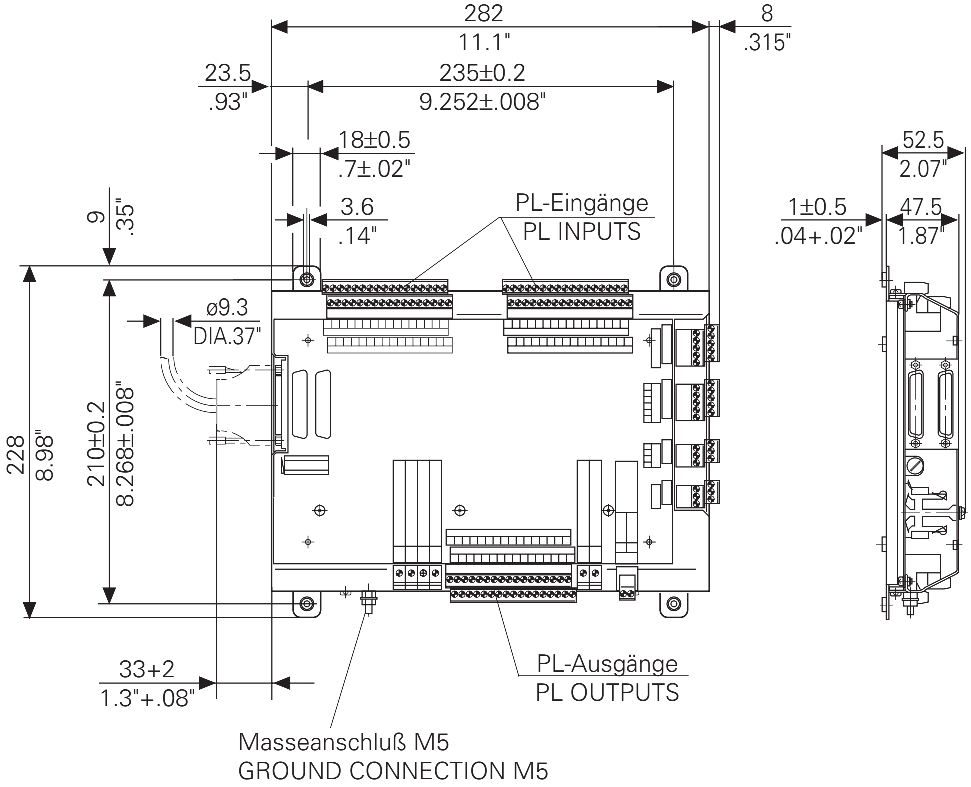


Ⓜ MOUNTING SURFACE

Ⓕ FRONT PANEL OPENING

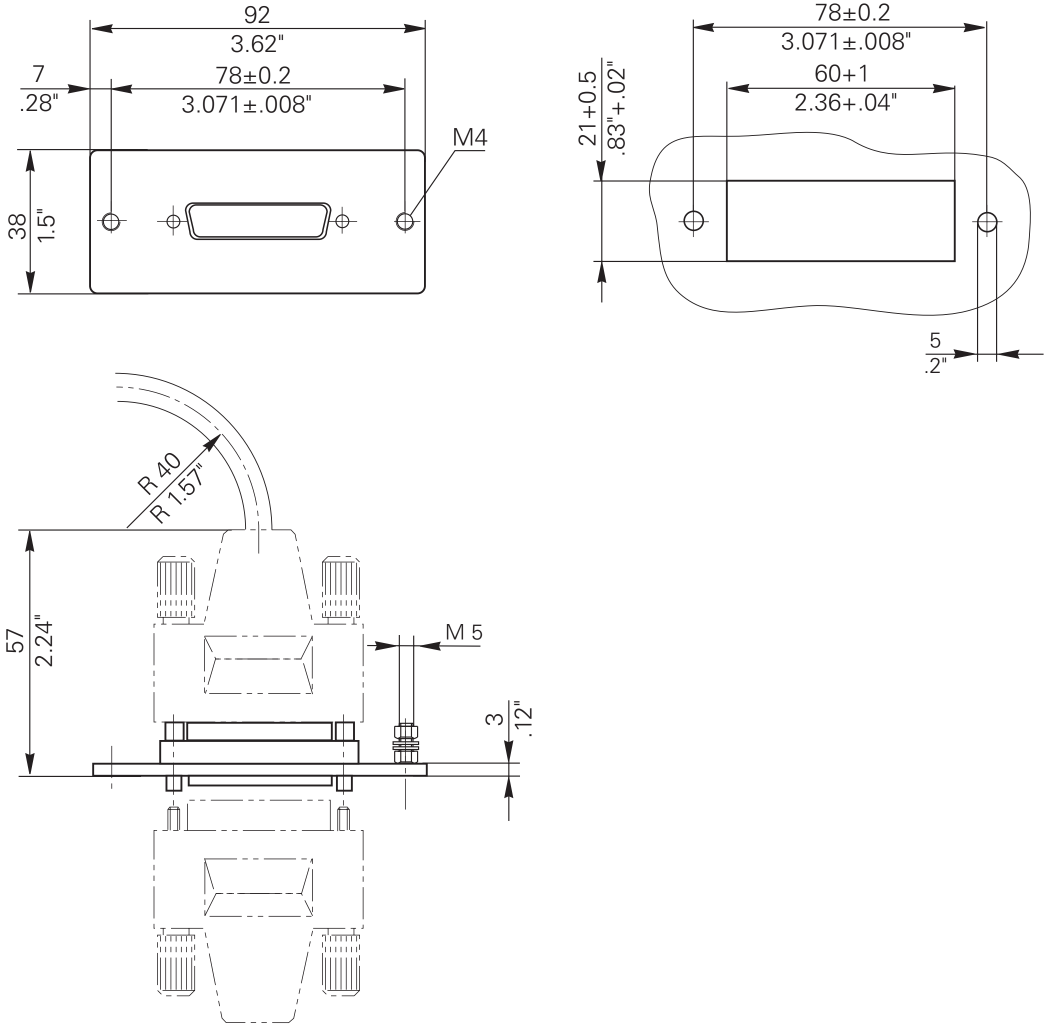
8.3.7 PL 410 B

Weight: 1.5 kg



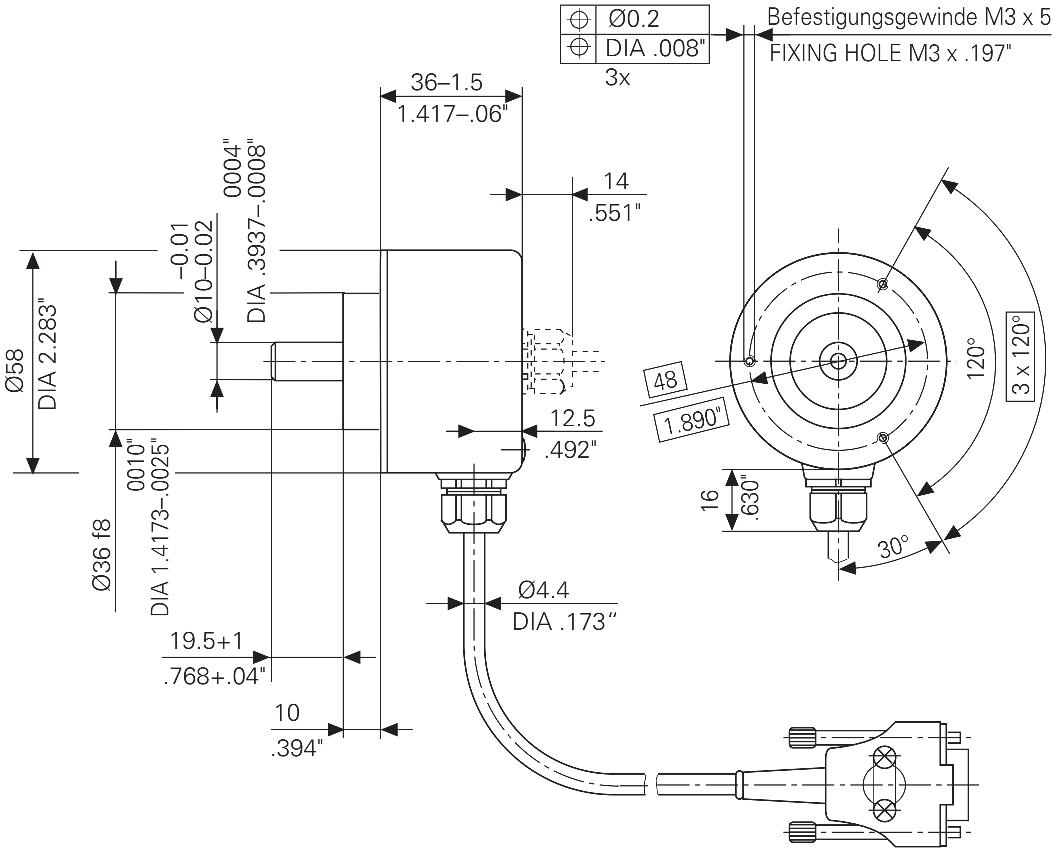
8.3.8 Adapter Block for Data Interface

RS-232-C/V.24 Adapter Block
RS-422/V.11 Adapter Block

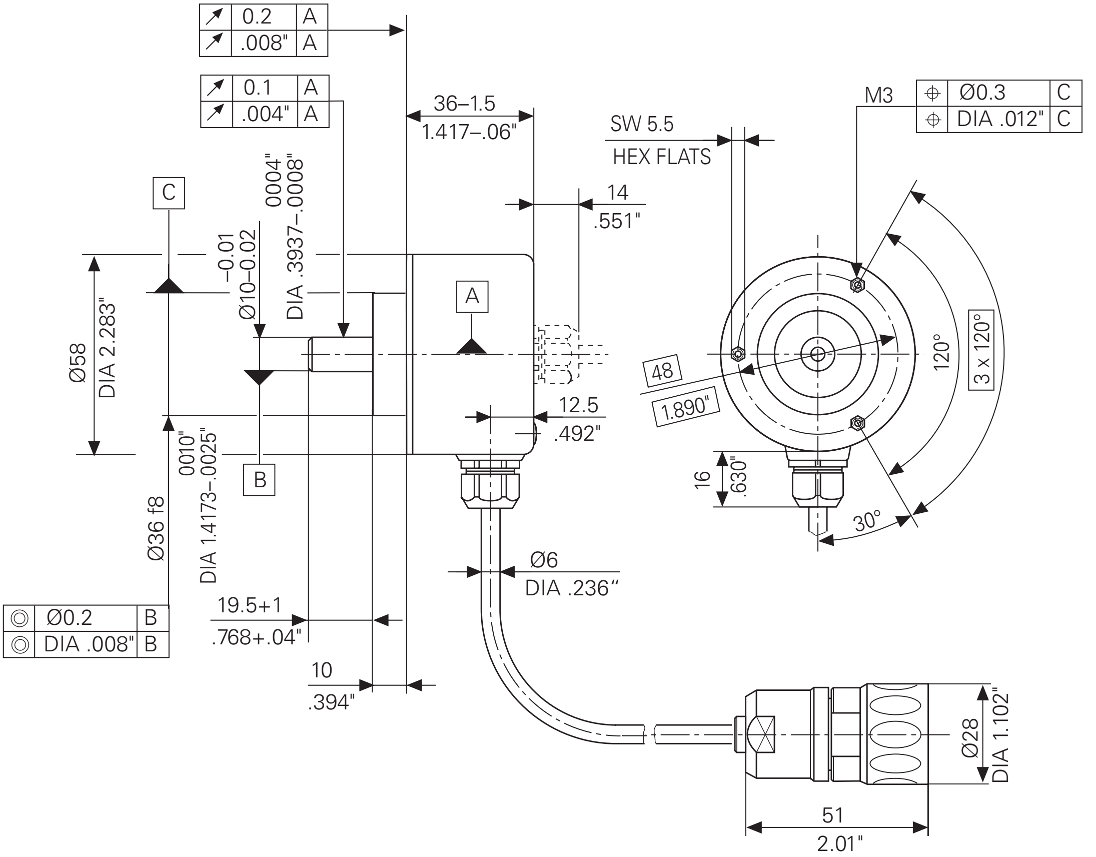


8.3.9 Electronic Handwheels

HR 130

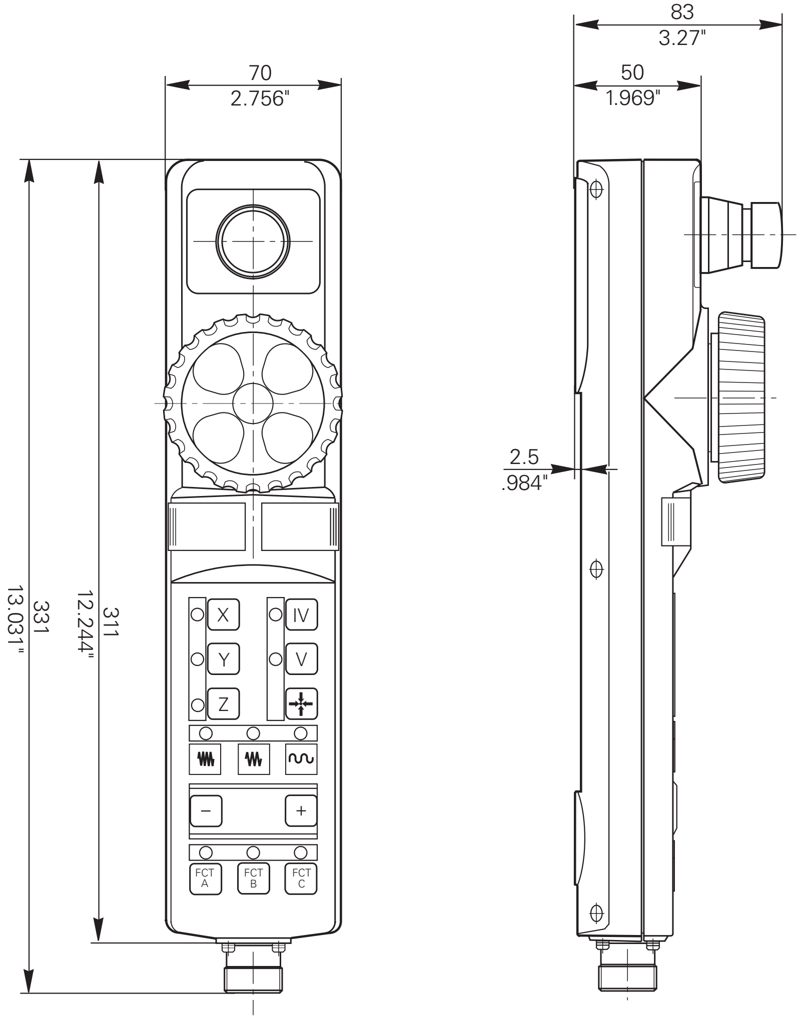


HR 150



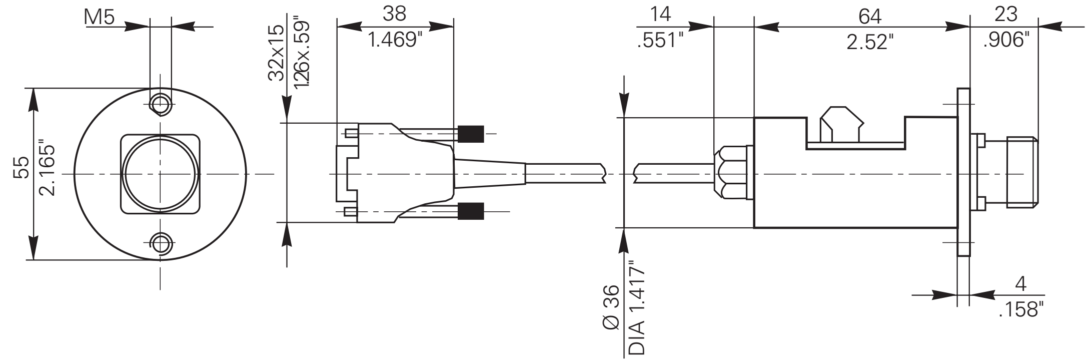
0.2 A
.008" A
0.1 A
.004" A
36–1.5
1.417–.06"
SW 5.5
HEX FLATS
M3
Ø0.3 C
DIA .012" C
14
.551"
C
A
–0.01
Ø10–0.02
.0004"
DIA .3937–.0008"
Ø58
DIA 2.283"
12.5
.492"
48
1.890"
120°
3 x 120°
16
.630"
30°
.0010"
DIA 1.4173–.0025"
B
Ø36 f8
Ø6
DIA .236"
Ø0.2 B
DIA .008" B
19.5+1
.768+.04"
10
.394"
Ø28
DIA 1.102"
51
2.01"

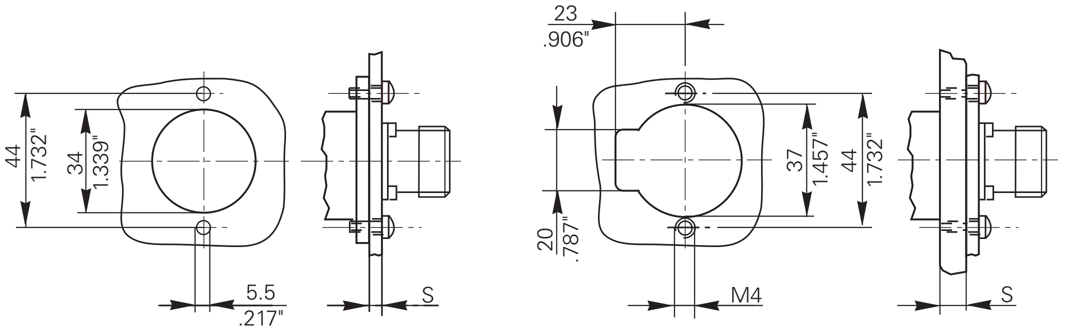
HR 410



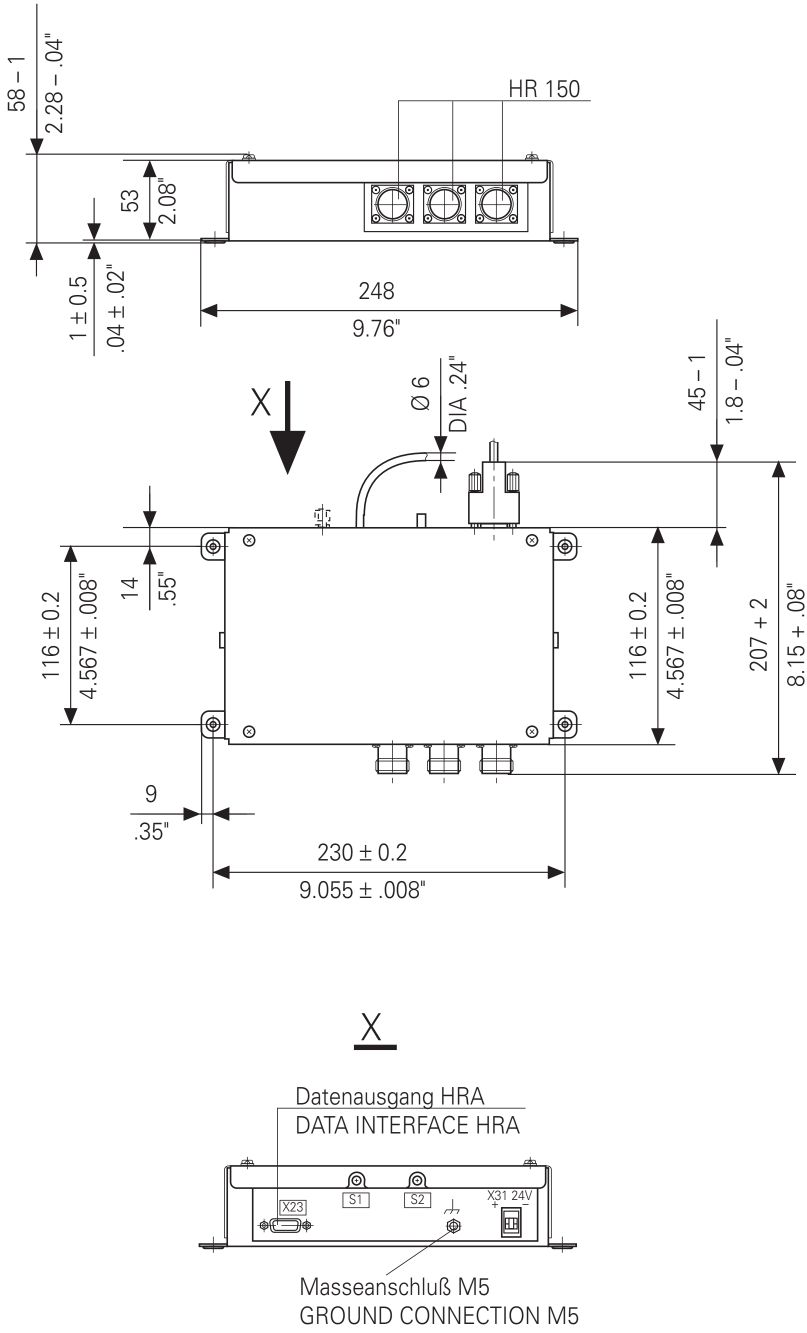
83
3.27"
70
2.756"
50
1.969"
2.5
.984"
331
13.031"
311
12.244"
X
Y
Z
IV
V
−
+
FCT A
FCT B
FCT C

Adapter Cable

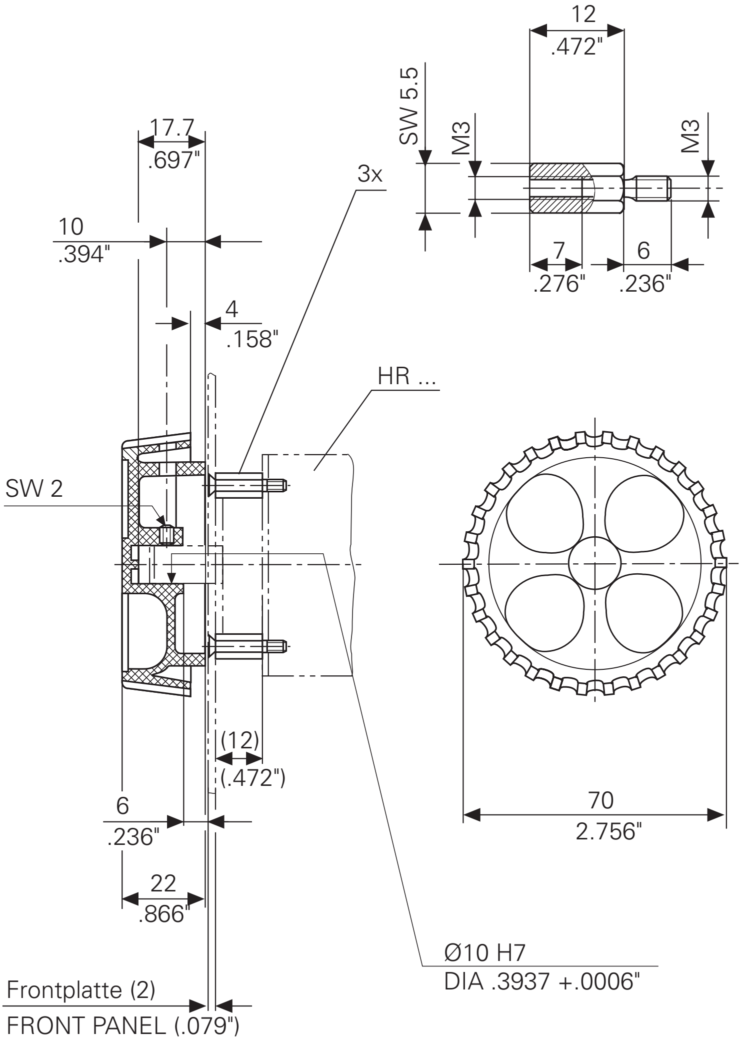




HRA 110

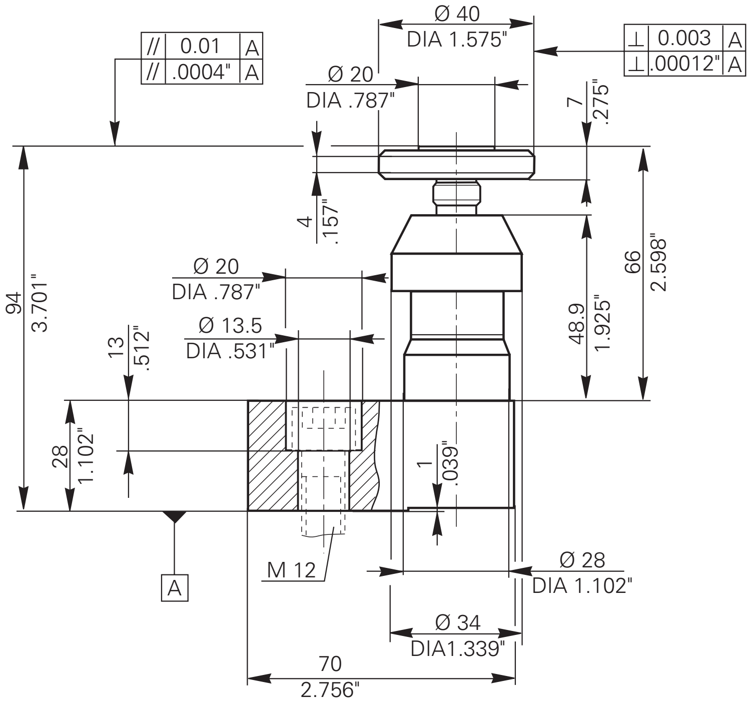

58 – 1
2.28 – .04"
HR 150
53
2.08"
1 ± 0.5
.04 ± .02"
248
9.76"
X
Ø 6
DIA .24"
45 – 1
1.8 – .04"
14
.55"
116 ± 0.2
4.567 ± .008"
116 ± 0.2
4.567 ± .008"
207 + 2
8.15 + .08"
9
.35"
230 ± 0.2
9.055 ± .008"
X
Datenausgang HRA
DATA INTERFACE HRA
X23
S1
S2
X31 24V
+
–
Masseanschluß M5
GROUND CONNECTION M5

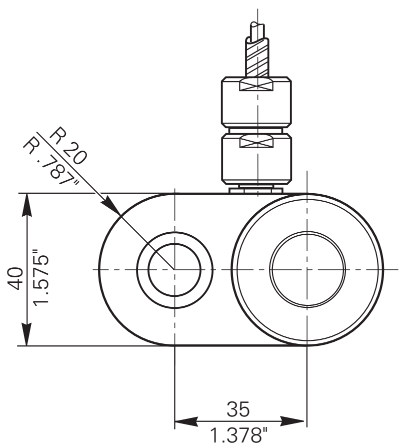
Wheel


12
.472"
SW 5.5
M3
M3
7
.276"
6
.236"
17.7
.697"
3x
10
.394"
4
.158"
HR ...
SW 2
(12)
(.472")
6
.236"
70
2.756"
22
.866"
Ø10 H7
DIA .3937 +.0006"
Frontplatte (2)
FRONT PANEL (.079")

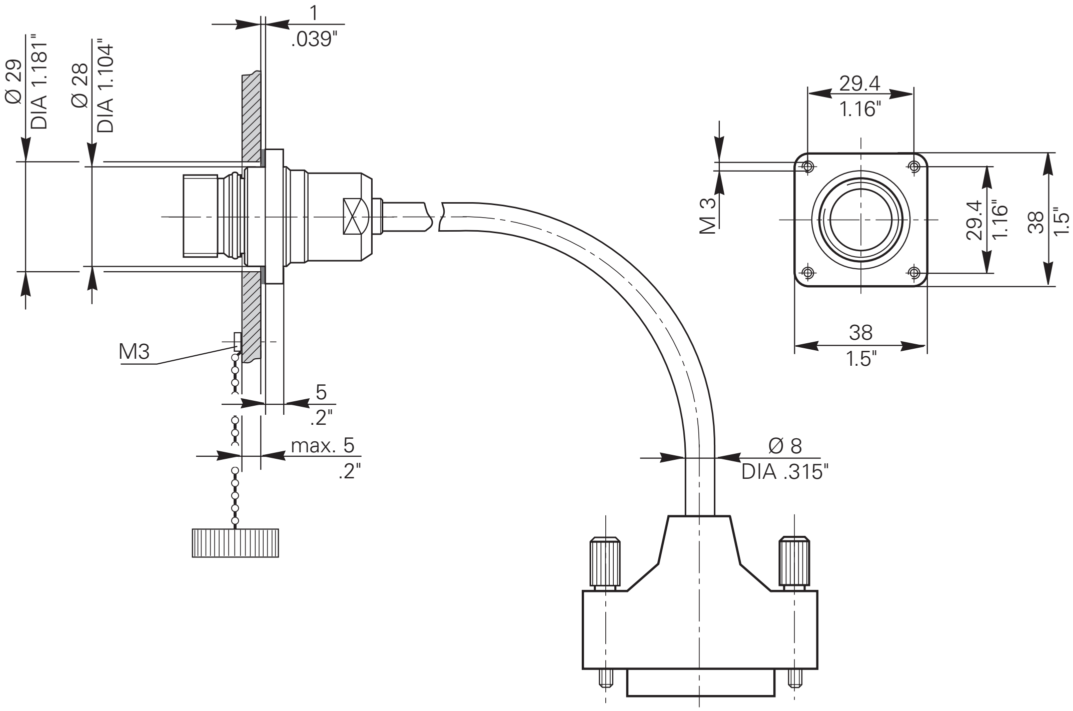
8.3.10 Touch Probe Systems

TT 120

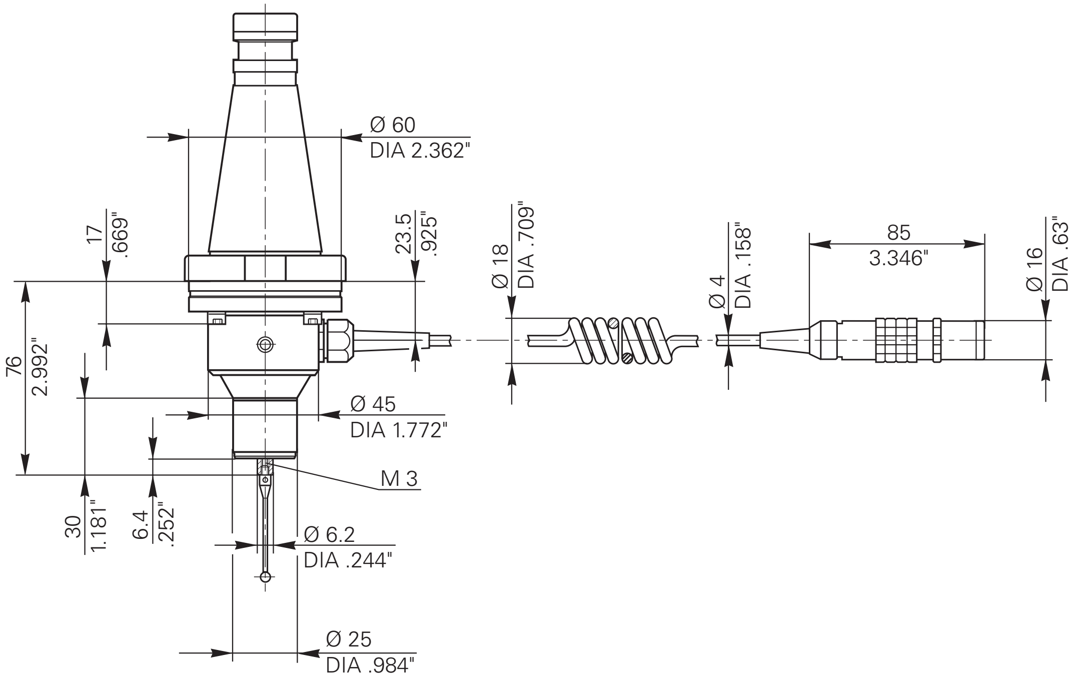




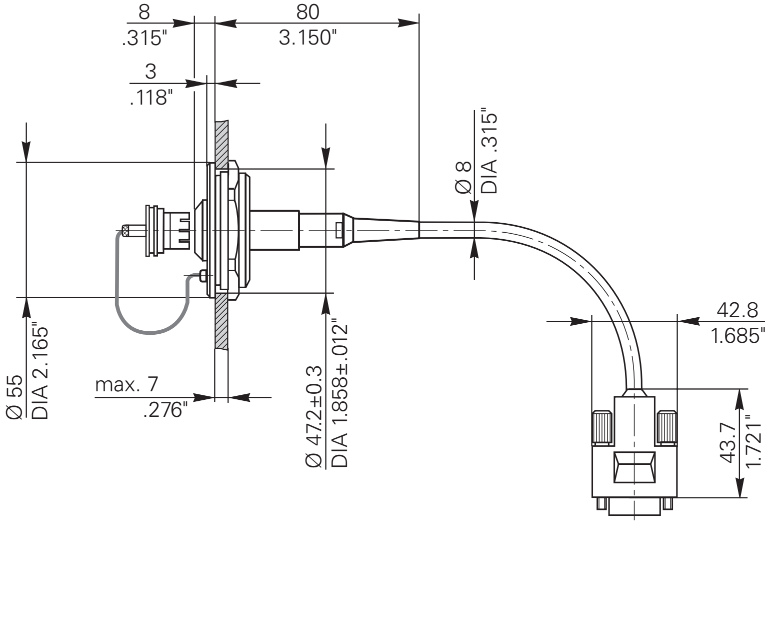
Adapter Cable for TT 120



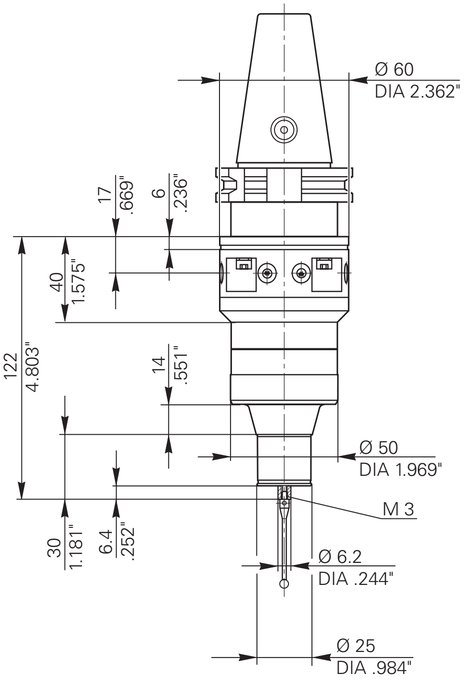
TS 220



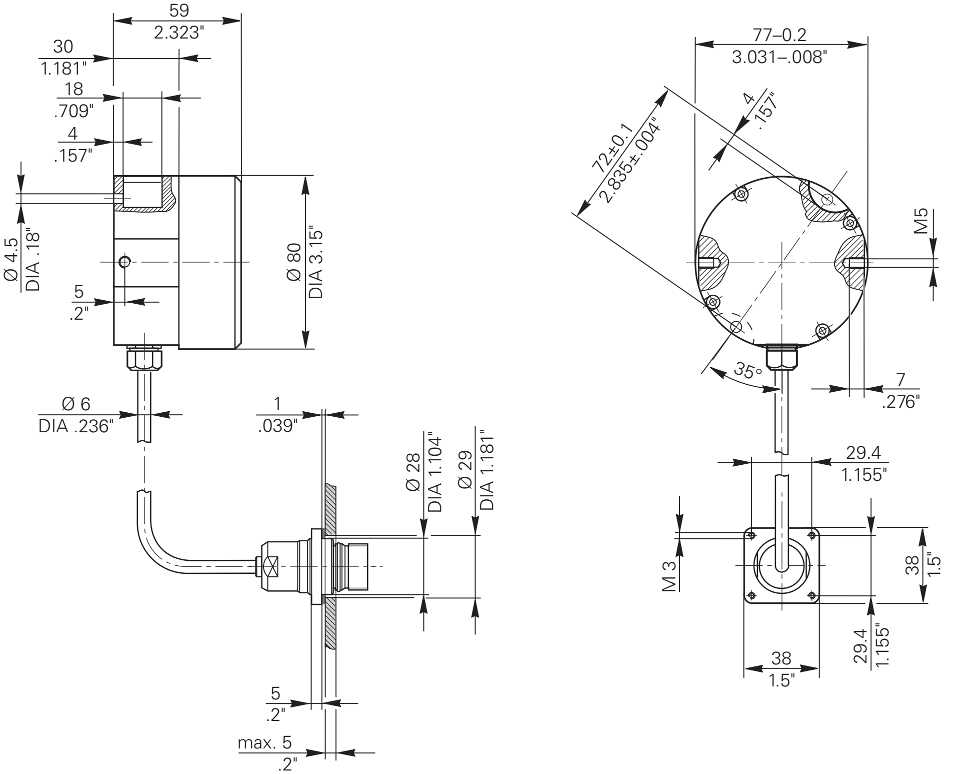
Adapter Cable for TS 120 / TS 220



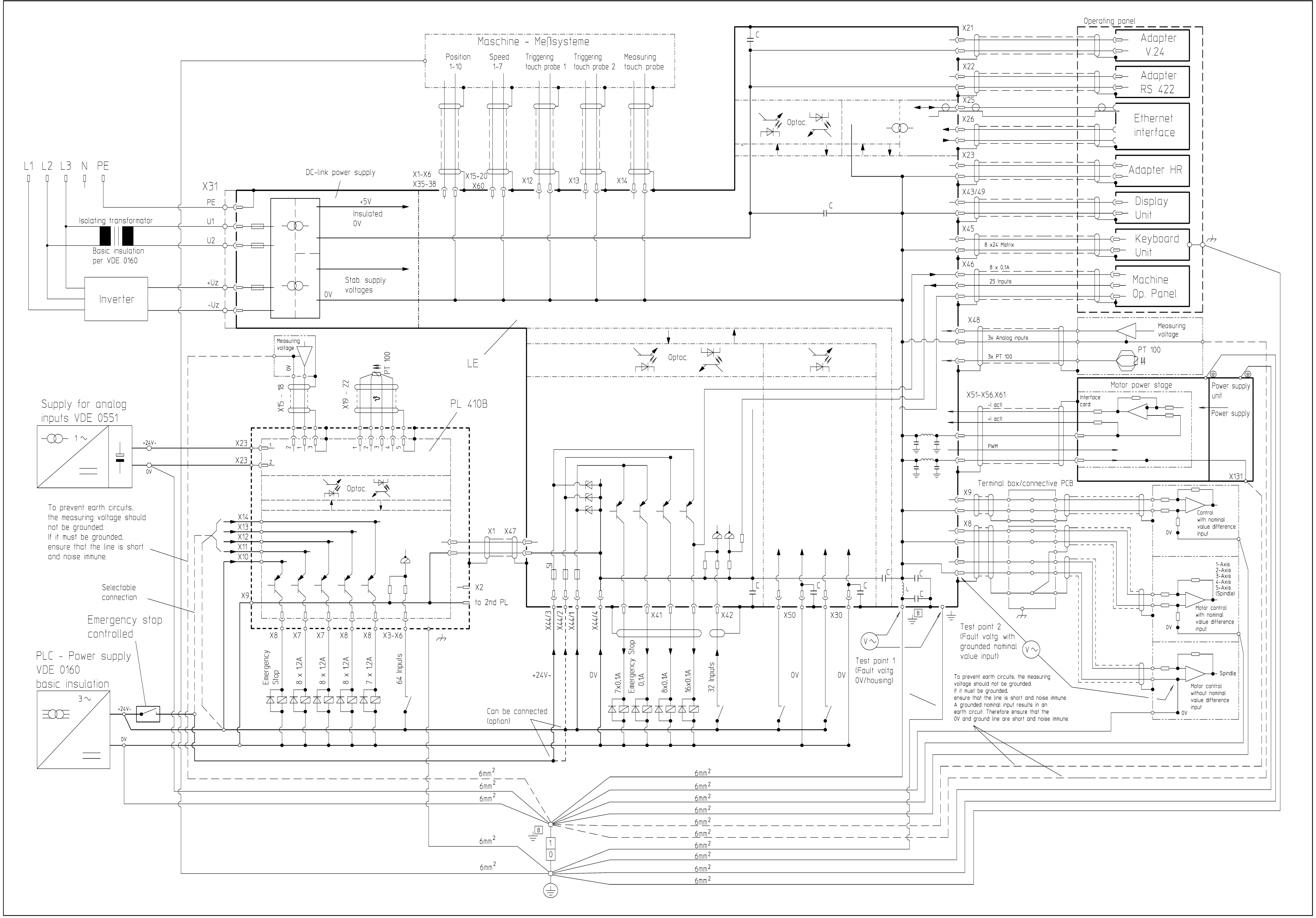
TS 630



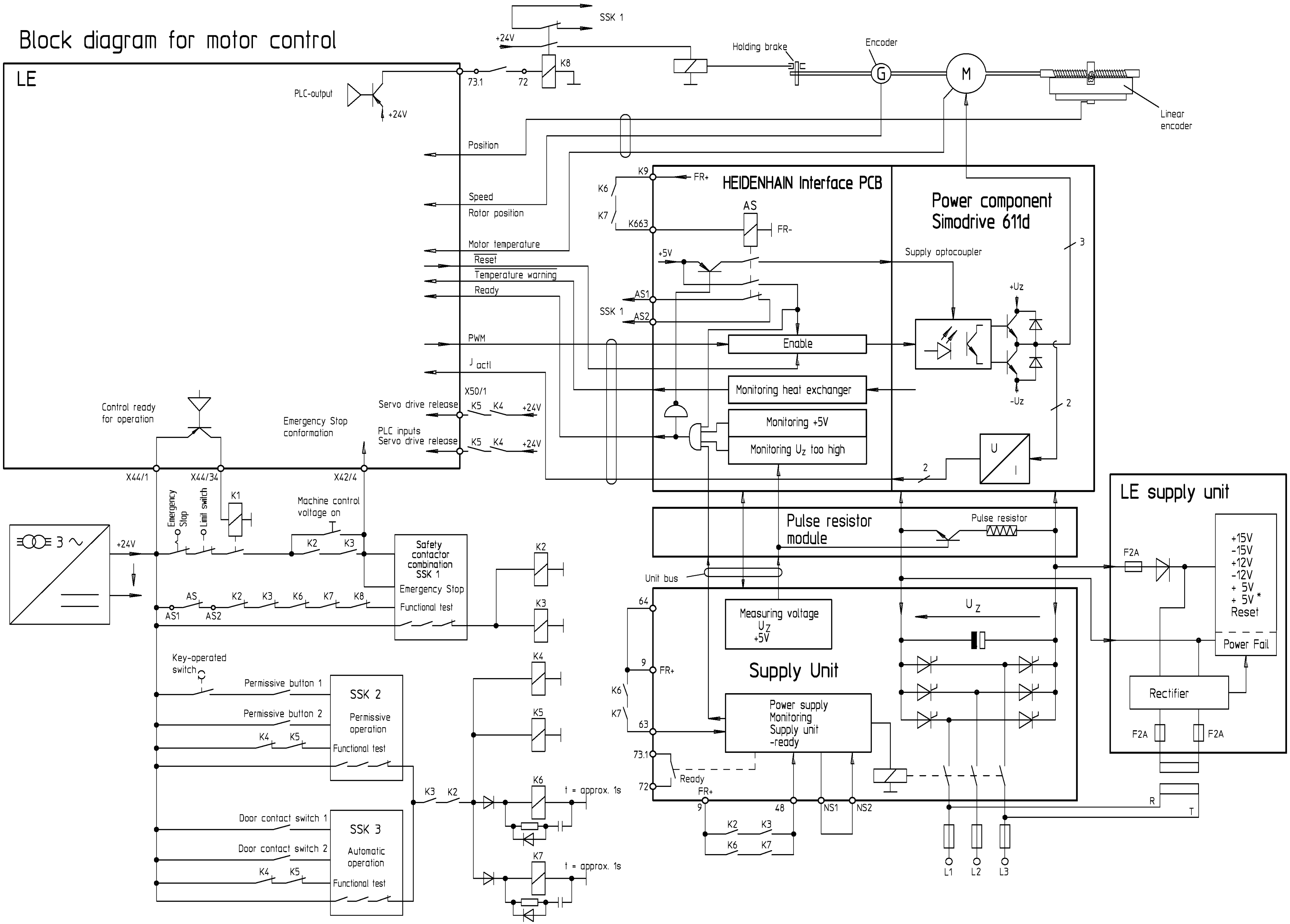
EA 550 Receiver Unit



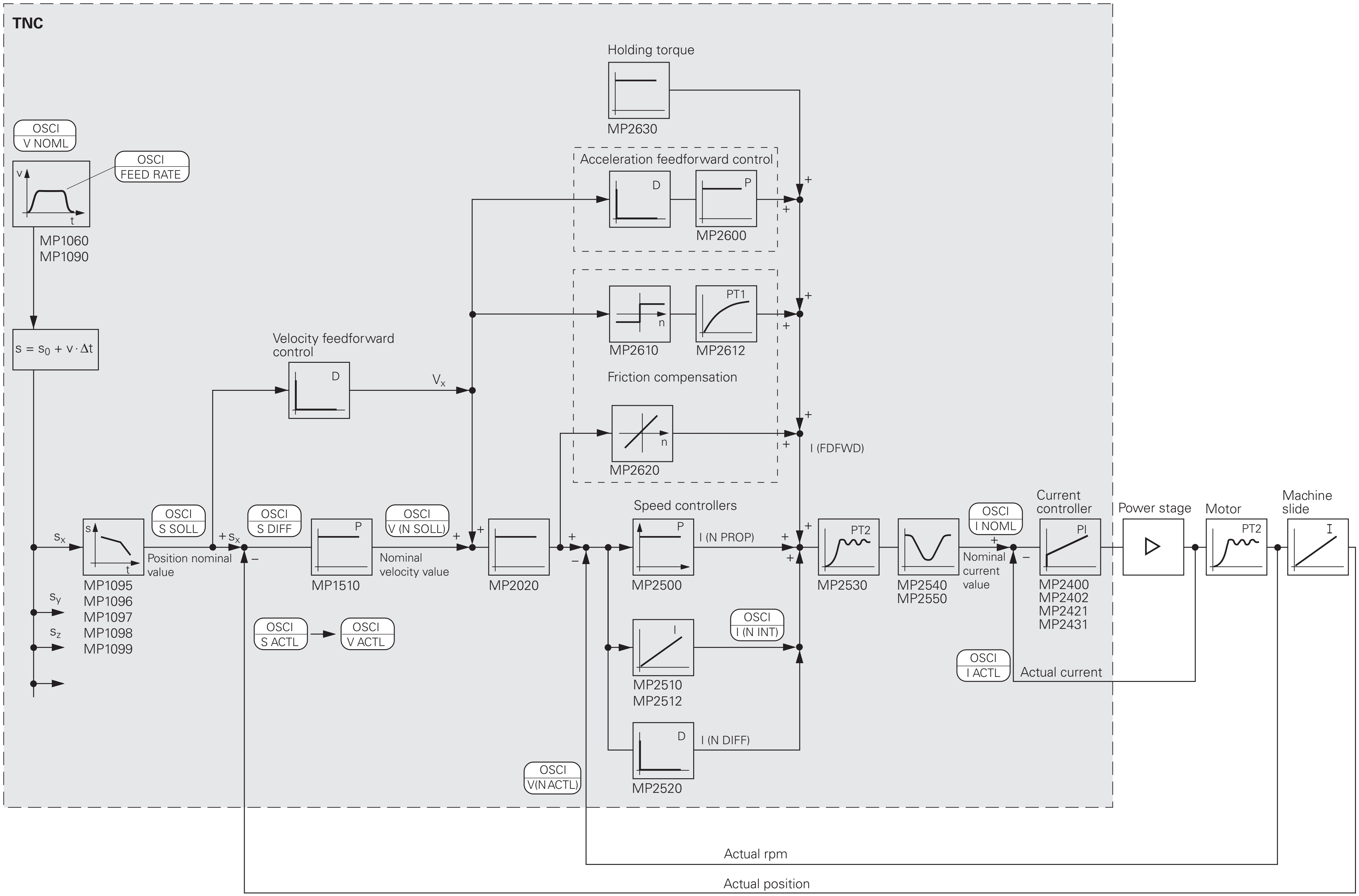
8.4 Grounding Diagram



8.5 Basic Circuit Diagram: Motor Control with TNC 426 PB, TNC 430 PA

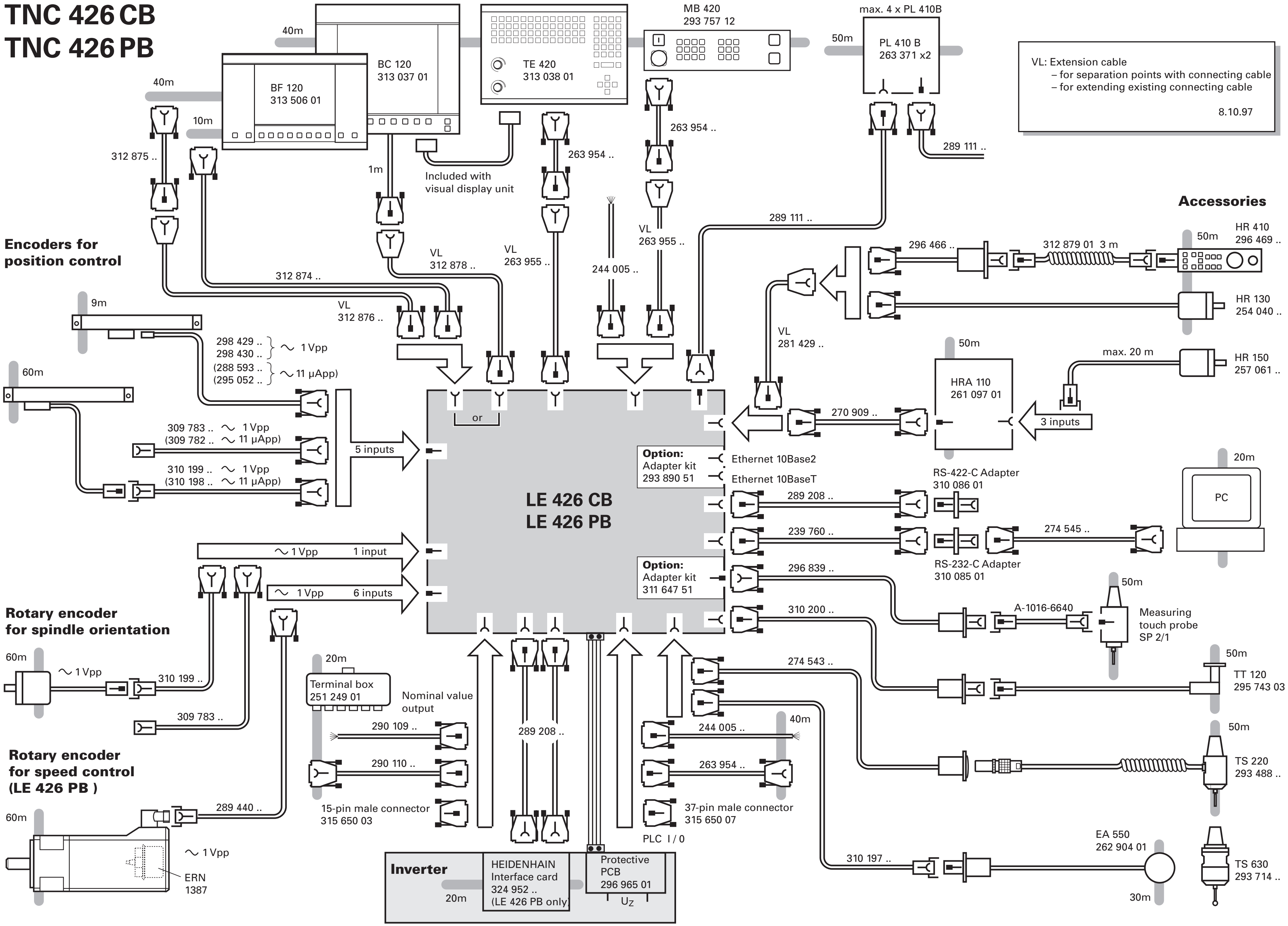


8.6 Block Diagram TNC 426 PB, TNC 430 PA



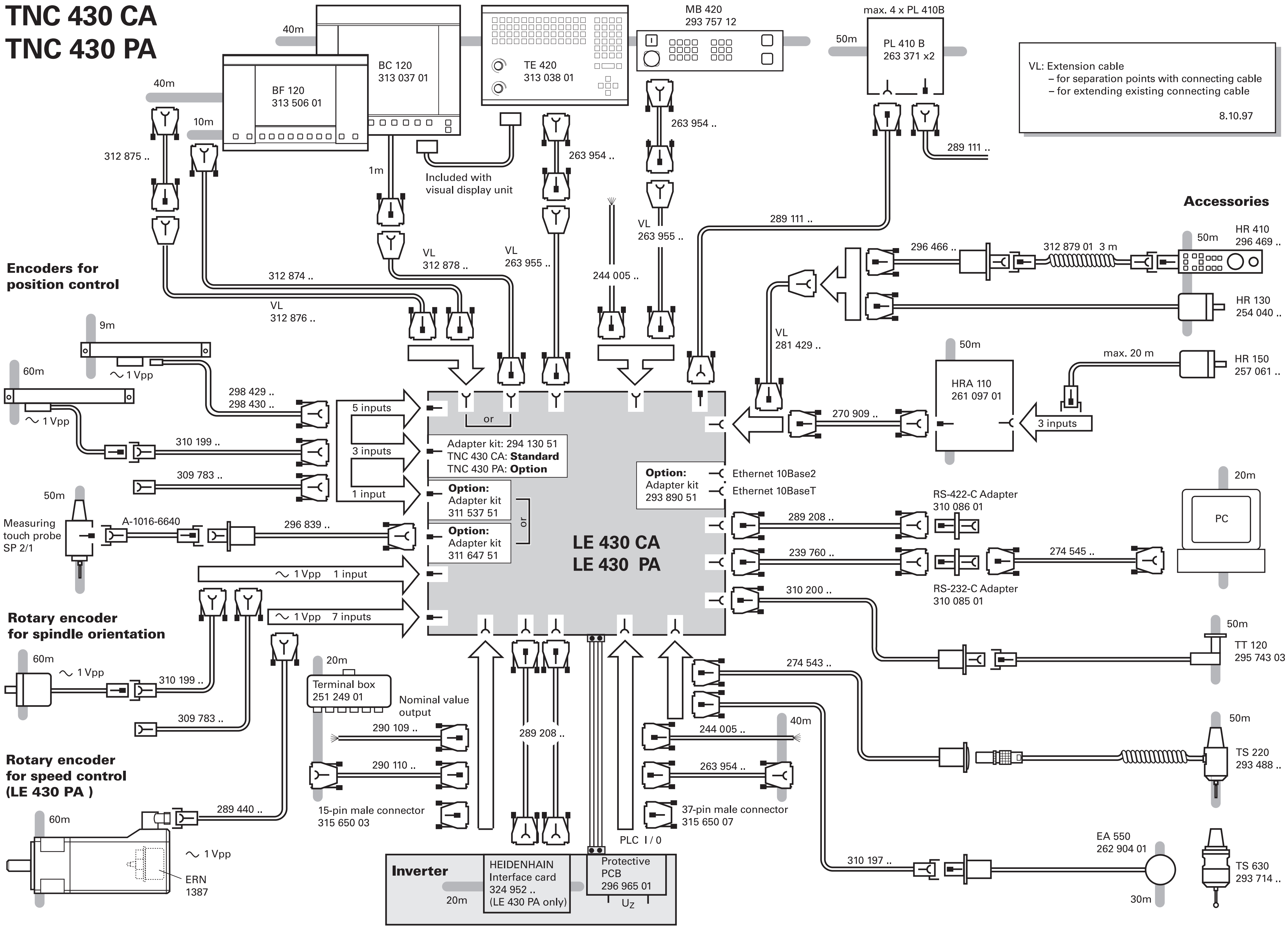
8.7 Cable Overview

8.7.1 TNC 426



8.7.2 TNC 430

TNC 430 CA
TNC 430 PA



9 Machine Parameters

9.1 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 so-called machine parameters.
In addition, machine parameters can be used to activated certain functions, which are possible with HEIDENHAIN contouring controls, but are required only on certain types of machines (e.g. automatic tool changing). The list of machine parameters is not numbered in sequence but is divided into groups according to function.

Machine parameter	Functional Group
10 to 999	Encoders and Machines
1000 to 1399	Positioning
1400 to 1699	Operation with Velocity Feedforward
1700 to 1999	Operation with Servo Lag
2000 to 2999	Integrated Closed-Loop 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 Touch Trigger Probe
7100 to 7199	Tapping
7200 to 7349	Display and Operation
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 provided with 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

Other machine parameters function as on/off switches for specific functions. These machine parameters are bit-encoded. Each bit is assigned either to an axis or a function.

9.2 Input and Output of Machine Parameters

If the machine parameters have not yet been entered in a HEIDENHAIN contouring control (e.g., during commissioning), the TNC presents the list of machine parameters after the memory test. Now you must enter the values either by hand on the keyboard or through the data interface.

9.2.1 Input Format

A number is entered for each machine parameter. This value can be, for example, the acceleration in mm/s^2 of an individual axis, or the analog voltage in volts. You can add a written comment to your entry by placing a semicolon ";" behind the numerical entry, followed by your comment. The input values can be entered in decimal, binary (%) or hexadecimal ($) format.

There are machine parameters with which individual functions are activated bit-coded. Binary entry (%) is recommended for these machine parameters. The hexadecimal format ($) may be advisable for other machine parameters.

Example:	Disabling soft keys for file types with MP7224.0.		
Bit 0	HEIDENHAIN programs	.H	0 = do not disable
Bit 1	ISO programs	.I	1 = disable
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.

Input value for MP7224.0 =	Binary	%00101000	
	Hexadecimal	$ 38	
	Decimal	40	($2^3 + 2^5$)

Special case: Entering a formula (as of NC software 280 472 01)
Currently only for MP2020 (Distance covered in one motor revolution)
Instead of a fixed value, you can enter a formula. Attention must be paid to small and capital letters. Functions are written in small letters and variables in capitals.

Functions:

+	addition	sin	sine
–	subtraction	cos	cosine
*	multiplication	tan	tangent
/	division	asin	arcsine
log	logarithm	acos	arccosine
log10	logarithm to the base 10	atan	arctangent
exp	exponent	sqrt	square root
( )	remove the brackets	sqr	square

Variable:

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

9.2.2 Activating the Machine Parameter Settings

After you have entered the values for the machine, 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	Entry value incorrect
4	MP doubly defined
6	MP can not be stored

If the control does not recognize any errors, it automatically exits the machine parameter editor and is ready for operation. If during commissioning you do not make any entries in the parameter list (MP NAME), the TNC will generate a standard machine parameter list when you press the END key and leave the machine parameter editor. In this list the TNC is defined as a programming station with the HEIDENHAIN standard colors. All other machine parameters assume the minimum value.

You can keep several machine parameter lists and load the desired list into the TNC when needed. The desired list can be selected in the machine parameter editor by pressing the PGM MGT key and the SELECT soft key. The parameter list that is active when you exit the machine parameter editor goes into effect.

9.2.3 Changing the Input Values

After you have created a machine parameter list, it can be changed either through the machine parameter editor or directly through the PLC.

The list of machine parameters included the following indicators showing how the value can be changed and how the TNC reacts after the change:

- CN123 The MP is also accessible through the code number 123
- PLC The MP can be changed through the PLC
- 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":

- Code number **95148**
  This code number give you access to the complete list of machine parameters.
- Code number **123**
  This code number gives you access to only some of the machine parameters. These are the machine parameters that the user is authorized to change (see User's Manual). In the following list, the machine parameters that can be changed through the code number 123 are indicated by „CN123.“

To exit the machine parameter editor, press END.

User parameters

With the USER PARAMETER MOD function you can easily access certain machine parameters without having to first enter a code number. In MP7330.x you can define up to 16 machine parameters, and in MP7340.x you define the associated dialog to be shown when the USER PARAMETER soft key is pressed. See also the chapter „Display and Operation."

Changing the input values through the PLC

The PLC can also change the machine parameters. You can use the following modules for this purpose:

- Overwrite machine parameters (Module 9031)
- Read machine parameters (Module 9032)
- Select machine parameter file (Module 9033)
- Load machine parameter partial file (Module 9034)

In the list below, the machine parameters that you can change with modules 9031 or 9034 are indicated with „PLC":

Overwrite Machine Parameter (Module 9031)

With Module 9031 you can overwrite the value of the given machine parameter with a new value. The input value must be a natural number including all possible decimal places.

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

Only the value in the run-time memory is modified, the value in the editable machine parameter list does not change. This means that the old value is valid again after editing and exiting from the machine parameter list.

Zero must be given as the index for non-indexed machine parameters. Once the NC program has started the module operates only during the output of M/S/T/Q strobes.

Call only from a submit job:

```
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/not modifiable/not modifiable once program has started
                2:      MP value out of range
                3:      Error when saving (fatal error)
                4:      Call was not from submit job
                5:      Call during running program without strobe
```

Read Machine Parameter (Module 9032)

With Module 9032 you can read the value of a machine parameter defined by its number and index from the current machine parameter list. The value is transferred as a natural number including all possible decimal places.

Only the value from the editable machine parameter list is read, not any value in the run-time memory modified by PLC Module 9031.

Zero must be given as the index for non-indexed machine parameters.

Call only from a submit job:

```
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:      No such MP number
                2:      No separator „:“
                3:      MP value out of range
                4:      MP not in file
                5:      No MP file found
                6:      Call was not from submit job
                7:      MP is of "string" type
```

Select Machine Parameter File (Module 9033)

With Module 9033 you select a new machine parameter file. If machine parameters were changed, triggering a reset, this is first carried out. The module disregards any safety problems when initiating the control reset (e.g. free run-out of axes and spindle).

The new MP 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 specified in a string that must contain the complete path and file name (1 to 16 characters). Additional characters (including blank characters) are not allowed. 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 only operates during the output of M/S/T/Q strobes.

Call only from submit job:

```
PS   B/W/D/K    <String number> 0 to 3
CM   9033       Note: Program execution ends here if  a new file is selected.
PL   B/W/D      <Error code>
                0:      No error, file was already selected
                1:      The specified string does not conform to the above conventions.
                2:      File found not
                3:      File is faulty
                4:      Incorrect string was transferred (out of range 0 to 3)
                5:      The module was not called from a submit job.
                6:      The module was called after the NC program started without a strobe marker
                        being active.
```

Loading a Machine Parameter Partial File (Module 9034)

With Module 9034 you load the contents of the given machine parameter file into the main memory. All parameters not contained in this file remained unchanged. The new MP file to be selected is checked; no faulty files are loaded. The MP file is not loaded if it contains parameters that require a system reset. The file name is transferred in a string that must contain the complete path with file name and extension. Additional characters (including blank characters) are not allowed. 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 only operates during the output of M/S/T/Q strobes.

Call only from a submit file:

```
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 have a valid file name, or name (including path) is too long
                2:     File found not
                3:     File is faulty / File contains reset parameters
                4:     Incorrect string number was transferred (out of range 0 to 3)
                5:     The module was not called from a submit job.
                6:     The module was called after the NC program started without a strobe marker
                       being active.
```

9.3 List of Machine Parameters

9.3.1 Encoders and Machine

Machine parameter	Function and input	Change via	Reaction	Page
MP10	Active axes Input: %xxxxxxxxx Bit 0 to 8 0 = not active Axis 1 to 9 1 = active		RESET	4–1
MP20.0	Monitoring the absolute position of the distance-coded reference marks Input: %xxxxxxxxx Bit 0 to 8 0 = not active Axis 1 to 9 1 = active	PLC	RUN	4–3
MP20.1	Monitoring the amplitude of the encoder signals Input: %xxxxxxxxx Bit 0 to 8 0 = not active Axis 1 to 9 1 = active	PLC	RUN	4–3
MP20.2	Monitoring the edge separation of the encoder signals Input: %xxxxxxxxx Bit 0 to 8 0 = not active Axis 1 to 9 1 = active	PLC	RUN	4–3
MP21.0	Monitoring of absolute position of the distance-coded reference marks of the spindle Input: %x Bit 0 0 = not active Spindle 1 = active	PLC	RUN	4–3
MP21.1	Monitoring of amplitude of the spindle position encoder signal Input: %x Bit 0 0 = not active Spindle 1 = active	PLC	RUN	4–3
MP21.2	Monitoring the edge separation of the spindle position encoder signal Input: %x Bit 0 0 = not active Spindle 1 = active	PLC	RUN	4–3
MP100 MP100.0-2	Assignment of axis characters to the axes Input: XYZABCUVWxyzabcuvw- (capital letters for NC axes, small letters or hyphen for PLC axes) Characters 1 to 9 (1 = right) Axes 1 to 9 Traverse range 1 to traverse range 3	PLC	RUN	4–5; 4–11

Machine parameter	Function and input	Change via	Reaction	Page
MP110	Assignment of position encoder inputs to axes Input: 0 = No position encoder 1 = Position encoder input X1 2 = Position encoder input X2 3 = Position encoder input X3 4 = Position encoder input X4 5 = Position encoder input X5 35 = Position encoder input X35 36 = Position encoder input X36 37 = Position encoder input X37 38 = Position encoder input X38		RESET	4–6
MP110.0-8	Axis 1 to axis 9			
MP111	Position encoder for the spindle Input: as for MP110			4–6
MP111.0-1	Spindle 1 and spindle 2 (only 280 474-xx)			
MP112	Assignment of the speed encoder inputs to the axes Input: 15 to 20 = Speed encoder input X15 to X20 62 to 64 = Speed encoder input X62 to X64			–
MP112.0-8	Axis 1 to axis 9			
MP113	Assignment of the speed encoder inputs to the spindles Input: 15 to 20 = Speed encoder input X15 to X20 60 = Speed encoder input X60 62 to 64 = Speed encoder input X62 to X64			–
MP113.0	First spindle			
MP113.1	Second spindle			

Machine parameter	Function and input	Change via	Reaction	Page
MP115.0	Switching the encoder inputs between 1 V_{PP} and 11 μA_{PP} Input: %xxxxxxxxxx Bit 0 to 5 X1 to X6 0 = 1 V_{PP} 1 = 11 μA_{PP} Bit 6 to 9 X35 to X38			–
MP115.1	Reserved Input: %0000000000			
MP115.2	Low or high input freq. (recommended input value for linear encoders: 50 kHz) Input: %xxxxxxxxxx Bit 0 to 5 X1 to X6 1 V_{PP}: 0 = 50 kHz 1 = 350 kHz Bit 6 to 9 X35 to X38 11 μA_{PP}: 0 = 50 kHz 1 = 150 kHz			
MP120	Assignment of nominal speed outputs to the axes Input: 0 = non-controlled axis A1 to A13 or 1 to 13 = Analog axis with analog output 1 to 13 (Analog outputs 1 to 6 at X8 connection Analog outputs 7 to 13 at X9 connection) D1 to D6 = digital axis 1 to 6 *Input (as of NC software 280 474-xx):* *0 = non-controlled axis* *1 = Analog nominal value at X8/1* *2 = Analog nominal value at X8/2* *3 = Analog nominal value at X8/3* *4 = Analog nominal value at X8/4* *5 = Analog nominal value at X8/5* *6 = Analog nominal value at X8/6* *7 = Analog nominal value at X9/7* *8 = Analog nominal value at X9/8* *9 = Analog nominal value at X9/9* *10 = Analog nominal value at X9/10* *11 = Analog nominal value at X9/11* *12 = Analog nominal value at X9/12* *13 = Analog nominal value at X9/13* *51 to 59, 61 = Digital nominal value at X51 to X59, or X61*		RESET	4–6
MP120.0-8	Axis 1 to axis 9 *NC software 280 470 ..: only 0 to 13 and definition of digital axes in MP2000*			

Machine parameter	Function and input	Change via	Reaction	Page
MP121 MP121.0-1	Assignment of nominal speed outputs to the spindle Input: 0 = non-controlled spindle A1 to A13 or 1 to 13 = analog spindle with analog output 1 to 13 (analog outputs 1 to 6 at X8 connection analog outputs 7 to 13 at X9 connection) S1 = digital spindle *Input (as of NC software 280 474-xx):* *0 = Non-controlled spindle* *1 = Analog nominal value at X8/1* *2 = Analog nominal value at X8/2* *3 = Analog nominal value at X8/3* *4 = Analog nominal value at X8/4* *5 = Analog nominal value at X8/5* *6 = Analog nominal value at X8/6* *7 = Analog nominal value at X9/7* *8 = Analog nominal value at X9/8* *9 = Analog nominal value at X9/9* *10 = Analog nominal value at X9/10* *11 = Analog nominal value at X9/11* *12 = Analog nominal value at X9/12* *13 = Analog nominal value at X9/13* *51 to 59, 61 = Digital nominal value at X51 to X59, or X61* Spindle 1 and spindle 2 *NC software 280 470 ..: only 0 to 13 and definition of digital spindles in MP2001*		RESET	4–6
MP210	Counting dir. of the position encoder signals Input: %xxxxxxxxx Bit 0 to 8 0 = Positive Axis 1 to 9 1 = Negative		RESET	4–2
MP331 MP331.0-8	Distance traveled for number of signal periods from MP332 Input: 0 to 99 999.9999 [mm or °] Axis 1 to axis 9	PLC	REF	4–2
MP332 MP332.0-8	Number of signal periods output in the distance entered in MP331 Input: 1 to 16 777 215 Axis 1 to axis 9	PLC	REF	4–2
MP334 MP334.0-8	Number of grating periods between zero pulses of distance-coded encoders Input: 1 to 65 535 0 = 1000 Axis 1 to axis 9	PLC		–

Machine parameter	Function and input	Change via	Reaction	Page
MP410 MP410.3 MP410.4	Axis designation for axis buttons IV and V Input: A, B, C, U, V, W, a, b, c, u, v, w Axis button IV Axis button V	PLC	RESET	4–5
MP420 MP420.0-8	Hirth coupling Input: 0 = no Hirth coupling 1 = Hirth coupling active Axis 1 to axis 9	PLC	RESET	4–243
MP430 MP430.0-8	Step size for Hirth coupling Input: 0.0000 to 30.0000 [°] Axis 1 to axis 9	PLC	RESET	4–243
MP710 MP710.0-8	Backlash compensation Input: –1.0000 to +1.0000 [mm or °] Axis 1 to axis 9	PLC		4–14
MP711 MP711.0-8	Height of the reversal spikes during circular traverse Input: –1.0000 to +1.0000 [mm] Axis 1 to axis 9	PLC	RUN	4–22
MP712 MP712.0-8	Compensation value per control-loop cycle time Input: 0.000000 to 99.999999 [mm] Axis 1 to axis 9	PLC	RUN	4–22
MP715 MP715.0-8	Height of the reversal peaks during circular traverse with M105 Input: –1.0000 to +1.0000 [mm] Axis 1 to axis 9	PLC	RUN	4–22
MP716 MP716.0-8	Compensation value per control-loop cycle time with M105 Input: 0.000000 to 99.999999 [mm] Axis 1 to axis 9	PLC	RUN	4–22
MP720 MP720.0-8	Linear axis-error compensation Input: –1.000 to +1.000 [mm/m] Axis 1 to axis 9	PLC		4–15
MP730	Selection of linear or non-linear axis-error compensation Input: %xxxxxxxxx 0 = linear axis error compensation 1 = nonlinear axis error compensation Bit 0 to 8 Axis 1 to 9	PLC		4–15, 4–19
MP750 MP750.0-8	Backlash Input: –1.0000 to +1.0000 [mm] Axis 1 to axis 9	PLC		4–15
MP752 MP752.0-8	Compensation time for value from MP750.x Input: 0 to 1000 [ms] Axis 1 to axis 9	PLC		4–15

Machine parameter	Function and input	Change via	Reaction	Page
MP810 MP810.0-8	Display mode for rotary and PLC axes Input: 0.0000 to 99 999.9999[°] 0 = no modulo; Software limit switch active > 0 = modulo value for display; Software limit switch inactive Axis 1 to axis 9	PLC	REF	4–124
MP850 MP850.0-8	Synchronized axes Input: 0 = main axis 1 = following axis for axis 1 to 9 = following axis for axis 9 Axis 1 to axis 9	PLC		4–39
MP855 MP855.0-8	Synchronization monitoring Input: 0 to 100.0000 [mm] 0 = monitoring not active Axis 1 to axis 9	PLC		4–40
MP860 MP860.0-8	Datum for synchronization control Input: 0 = Datum from position at switch-on 1 = Datum from reference marks (machine datum) 2 = Axis is torque slave axis Axis 1 to axis 9	PLC		4–39; 4–40
MP910 MP910.0-8	Traverse range 1; Default setting after switch-on; activated by PLC M4575 = 0, M4574 = 0 Input: –99 999.9999 to +99 999.9999 [mm] or [°] (Input values are referenced to the machine datum) Software limit axis 1+ to 9+	PLC		4–9
MP911 MP911.0-8	Traverse range 2; activation by PLC M4575 = 0, M4574 = 1 Input: –99 999.9999 to +99 999.9999 [mm] or [°] (Input values are referenced to the machine datum) Software limit axis 1+ to 9+	PLC		4–9
MP912 MP912.0-8	Traverse range 3; activation by PLC: M4575 = 1, M4574 = 0 Input: –99 999.9999 to +99 999.9999 [mm] or [°] (Input values are referenced to the machine datum) Software limit axis 1+ to 9+	PLC		4–9

Machine parameter	Function and input	Change via	Reaction	Page
MP920 MP920.0-8	Traverse range 1; Default setting after switch-on; activated by PLC M4575 = 0, M4574 = 0 Input: –99 999.9999 to +99 999.9999 [mm] or [°] (Input values are referenced to the machine datum) Software limit axis 1– to 9–	PLC		4–9
MP921 MP921.0-8	Traverse range 2; activation by PLC M4575 = 0, M4574 = 1 Input: –99 999.9999 to +99 999.9999 [mm] or [°] (Input values are referenced to the machine datum) Software limit axis 1– to 9–	PLC		4–10
MP922 MP922.0-8	Traverse range 3; activation by PLC: M4575 = 1, M4574 = 0 Input: –99 999.9999 to +99 999.9999 [mm] or [°] (Input values are referenced to the machine datum) Software limit axis 1– to 9–	PLC		4–10
MP950 MP950.0-8	Datum for positioning blocks with M92 Input: –99 999.9999 to +99 999.9999 [mm] or [°] (Values referenced to the machine datum) Axis 1 to axis 9	PLC	RUN	4–118
MP951 MP951.0-8	Simulated tool-change position for TOOL CALL in block scan Input: –99 999.9999 to +99 999.9999 [mm] or [°] Axis 1 to axis 9	PLC	RUN	4–167
MP960 MP960.0-8	Machine datum Input: –99 999.9999 to +99 999.9999 [mm] or [°] Values referenced to scale reference point Axis 1 to axis 9	PLC	REF	4–43; 4–118

9.3.2 Positioning

Machine parameter	Function and input	Change via	Reaction	Page
MP1010 MP1010.0-8	Rapid traverse Input: 10 to 300 000 [mm/min] Axis 1 to axis 9	PLC		4–60
MP1020 MP1020.0-8	Manual feed rate Input: 10 to 300 000 [mm/min] Axis 1 to axis 9	PLC		4–60
MP1030 MP1030.0-8	Positioning window Input: 0.0001 to 2.0000 [mm] Axis 1 to axis 9	PLC		4–82
MP1040	**Analog axes:** Polarity of the nominal value voltage for the positive traverse direction **Digital axes:** Polarity of rotational speed nominal value for positive traverse direction Input: %xxxxxxxxx Bit 0 to 8 0 = Positive Axis 1 to 9 1 = Negative			4–2
MP1050 MP1050.0-8	**Analog axes:** Analog voltage at rapid trav. Input: 1.000 to 9.000 [V] **Digital axes:** not applicable Input: 1 Axis 1 to axis 9	PLC		4–60
MP1060 MP1060.0-8	Acceleration Input: 0.001 to 20.000 [m/s^2] Axis 1 to axis 9	PLC		4–54
MP1070	Radial acceleration Input: 0.001 to 20.000 [m/s^2]	PLC	RUN	4–76
MP1080 MP1080.0-8	**Analog axes:** Integral factor Input: 0 to 65 535 **Digital axes:** not applicable Input: 0 Axis 1 to axis 9	PLC	RUN	4–76
MP1090 MP1090.0 MP1090.1	Limiting jerk Input: 0.1 to 1000 [m/s^3] Limiting jerk with machining feed rate Limiting jerk as of feed rate in MP1092	PLC	RUN	4–54
MP1092	Threshold from which MP1090.1 functions Input: 10 to 300 000 [mm/min]	PLC	RUN	4–54
MP1094	Cutoff frequency for HSC filter Input: 0 = no HSC filter 0 to 166.0 [Hz]			–
MP1095	Nominal position value filter Input: 0 = Single filter 1 = Double filter Suggested input value = 0	PLC	RUN	4–54

Machine parameter	Function and input	Change via	Reaction	Page
MP1096	Tolerance Input: 0 = no nom. position value filter 0.001 to 3.000 [mm] = permissible tolerance at contour transitions	PLC	RUN	4–54, 4-77
MP1097 MP10970-8	Axis-specific jerk for single filter (MP1095 = 0) Input: 0.1 to 1000 [m/s³] Axis 1 to axis 9	PLC	RUN	4–54
MP1098 MP10980-8	Axis-specific jerk for double filter (MP1095 = 1) Input: 0.1 to 1000 [m/s³] Suggested input value = 2 • MP1097.x Axis 1 to axis 9	PLC	RUN	4–54
MP1099 MP1099.0 MP1099.1	Minimum filtering order Input: 0 to 20 Minimum filtering order for single filter (MP1095 = 0) Suggested input value = 5 Minimum filtering order for double filter (MP1095 = 1) Suggested input value = 3	PLC	RUN	4–55
MP1110 MP1110.0-8	Standstill monitoring Input: 0.0010 to 30.0000 [mm] Axis 1 to axis 9	PLC		4–82
MP1140 MP1140.0-8	Threshold from which movement monitoring is effective Input: **Analog axes:** 0.030 to 10.000 [V] **Digital axes:** 0.030 to 10.000 [1000/min] Axis 1 to axis 9	PLC	RUN	4–81
MP1150 *MP1150.0[1]* *MP1150.1[1]* *MP1150.2[1]*	Delay time for switching off the residual voltage with the blinking error message Excessive servo lag <axis> Input: 0 to 65.535 [s] Recommended: 0 *Delay time for canceling the nominal speed value with above error message* *Input: 0 to 65.535 [s]* *Recommended: 0* *Time during which the monitoring functions should remain switched off* *Input: 0 to 65.535 [s]* *Recommended: 0.2 s to 0.5 s* *Min. time for which monitoring functions should remain active after MP1150.1 has elapsed* *Input: 0 to 65.535 [s]*	PLC	RUN	4–80

[1] As of NC software 280 476-xx

Machine parameter	Function and input	Change via	Reaction	Page
MP1220	(removed as of NC software 280 474-07) **Analog axes:** automatic cyclic drive offset compensation Input: 0 to 65 536 [s] 0 = no automatic compensation **Digital axes:** not applicable Input: 0	PLC	RUN	4–75
MP1320	Direction for traversing the reference marks Input: %xxxxxxxxx Bit 0 to 8 0 = Positive Axis 1 to 9 1 = Negative	PLC		4–43
MP1330 MP1330.0-8	Feed rate for traversing the reference marks Input: 80 to 300 000 [mm/min] Axis 1 to axis 9	PLC	RUN	4–43
MP1331 MP1331.0-8	Feed rate for leaving the reference end position switches (only for rotary encoders MP1350=2) Input: 80 to rapid traverse [mm/min] Axis 1 to axis 9	PLC	RUN	4–43
MP1340 MP1340.0-8	Axis sequence for traversing the reference marks Input: 0 = No evaluation of the reference mark 1 = Axis 1 to 9 = Axis 9	PLC	REF	4–44
MP1350 MP1350.0-8	Type of reference mark approach Input: 0 = Linear encoder with distance-coded reference marks (not recommended, see Technical Manual) 1 = Linear encoder with one ref. mark 2 = Special sequence (linear measurement with ROD) 3 = Linear encoder with distance-coded reference marks 4 = Same as 3, but with evaluation of 2 additional reference pulses 5 = Encoder with EnDat interface Axis 1 to axis 9	PLC	REF	4–44

Machine parameter	**Function and input**	**Change via**	**Reaction**	**Page**
MP1390	(removed as of NC software 280 474-07) Velocity feedforward control in the "Positioning with MDI,“ "Program run, single block" and "Program run, full sequence“ operating modes Input: 0 = Velocity feedforward control 1 = Control with servo lag	PLC		4–56
MP1391	Velocity feedforward control in the "Manual" and "Handwheel" operating modes Input: %xxxxxxxxx 0 = Servo lag control 1 = Velocity feedforward control Bit 0 to 8 Axis 1 to 9	PLC		4–24; 4–56
MP1392	Velocity feedforward control in the "Positioning with MDI", "Program Run, Single Block" and "Program Run, Full Sequence" operating modes Input: %xxxxxxxxx 0 = servo lag control 1 = velocity feedforward control Bit 0 to 8 Axis 1 to 9	PLC		–

9.3.3 Operation with Velocity Feedforward

Machine parameter	Function and input	Change via	Reaction	Page
MP1410 MP1410.0-8	Position monitoring for velocity feedforward control (erasable) Input: 0.0010 to 30.0000 [mm] Axis 1 to axis 9	PLC		4–80
MP1420 MP1420.0-8	Position monitoring for velocity feedforward control (EMERGENCY STOP) Input: 0.0010 to 30.0000 [mm] Axis 1 to axis 9	PLC		4–80
MP1510 MP1510.0-8	k_v factor for velocity feedforward Input: 0.100 to 20.000 $[\frac{m/min}{mm}]$ Axis 1 to axis 9	PLC	RUN	4–59
MP1511 MP1511.0-8	Factor for static friction compensation Input: 0 to 16 777 215 Axis 1 to axis 9	PLC	RUN	4–24
MP1512 MP1512.0-8	Limit to amount of static friction compensation Input: 0 to 16 777 215 [counting steps] Axis 1 to axis 9	PLC	RUN	4–24
MP1513 MP1513.0-8	Feed-rate limit for static friction compensation Input: 0 to 300 000 $[\frac{mm}{min}]$ Axis 1 to axis 9	PLC	RUN	4–24
MP1515 MP1515.0-8	k_v factor for velocity feedforward control (M105) Input: 0.100 to 20.000 $[\frac{m/min}{mm}]$ Axis 1 to axis 9	PLC	RUN	4–59
MP1521	Transient response Input: 1 to 255 [ms] 0 = function not active	PLC		–

9.3.4 Operation with Servo Lag

Machine parameter	Function and input	Change via	Reaction	Page
MP1710 MP1710.0-8	Position monitoring for control with servo lag (erasable) Input: 0.0000 to 300.0000 [mm] Recommended: 1.2 • servo lag Axis 1 to axis 9	PLC		4–80
MP1720 MP1720.0-8	Position monitoring for control with servo lag (EMERGENCY STOP) Input: 0.0000 to 300.0000 [mm] Axis 1 to axis 9	PLC		4–80
MP1810 MP1810.0-8	k_v factor for control with servo lag Input: 0.100 to 20.000 $[\frac{m/min}{mm}]$ Axis 1 to axis 9	PLC		4–57
MP1815 MP1815.0-8	k_v factor for control with servo lag effective after M105 Input: 0.100 to 20.000 $[\frac{m/min}{mm}]$ Axis 1 to axis 9	PLC		4–57
MP1820 MP1820.0-8	Multiplier for the k_V factor Input: 0.001 to 1.000 Axis 1 to axis 9	PLC		4–61
MP1830 MP1830.0-8	Kink point in characteristic curve Input: 0.000 to 100.000 [%] Axis 1 to axis 9	PLC		4–61

9.3.5 Integral Speed and Current Control (Digital Axes Only)

Machine parameter	Function and input	Change via	Reaction	Page
MP2000[1] *MP2000.0-5*	*Type of drive* *Input:* *0 = Output rotational speed nominal value (analog axis)* *1 = Output PWM pulses (digital axis)* *Axis 1 to axis 6*		*RESET*	*4–307*
MP2001[1]	*Spindle drive type* *Input:* *0 = Output rotational speed nominal value (analog spindle)* *1 = Output PWM pulses (digital spindle)*			*4–307*
MP2020 MP2020.0-5	Distance covered in one motor revolution Input: 0 to 100.000 mm or <formula> Axis 1 to axis 6			4–81
MP2100 MP2100.0-5	Type of power module for the axes Name of power module (listed in the TNC) Axis 1 to axis 6			4–309
MP2101	Type of power module for the spindle Name of power module (listed in the TNC)			4–309
MP2180	PWM frequency Input: 0 to 10 000 [Hz]			–
MP2190	DC link voltage Input: 0 to 10 000 [V]			4–310
MP2191	Decelerating the spindle at EMERGENCY STOP Input: %x 0 = deceleration with monitoring of max. braking current 1 = deceleration without monitoring of max. braking current			–
MP2200 MP2200.0-5	Motor type for the axes Name of motor (listed in the TNC) Axis 1 to axis 6		RESET	4–309
MP2201	Motor type for spindle Name of motor (listed in the TNC)			4–309
MP2221	Monitoring the spindle reference pulse Input: 0 = monitoring active 1 = monitoring not active	PLC		–

[1] NC software 280 470 ..

Machine parameter	Function and input	Change via	Reaction	Page
MP2302 MP2302.0-5	Reference value for I^{2}t monitoring of axis motors Input: 0 to 1 000.000 [factor] 0 = I^{2}t monitoring of axis motor inactive 1 = motor nominal current as reference value Axis 1 to axis 6			4–85
MP2303	Reference value for I^{2}t monitoring of the spindle motor Input: 0 to 1 000.000 [factor] 0 = I^{2}t monitoring of the spindle motor inactive 1 = motor rated current as reference value			4–85
MP2312.0-5	Reference value for utilization display of feed motors Input: 0 to 1 000.000 [A_s] 0 or 1 = motor rated current as reference value			4–88
MP2313	Reference value for utilization display of the spindle motor Input: 0 to 1 000.000 [Factor] 0 or 1 = motor rated current is used as reference value			4–88
MP2340 MP2340.0-5	Speed starting from which the field angle begins to shift on synchronous motors Input: 0 to 100 000 [rpm] 0 = No field angle displacement Axis 1 to axis 6			4–309
MP2350 MP2350.0-5	Maximum field angle shift on synchronous motors Input: 0 to 60 [°] Axis 1 to axis 6			4–309
MP2360 MP2360.0-8	Time constant for braking a second spindle during EMERGENCY STOP Input: 0.01 to 5.00 [s] 0 = function not active PWM output of second spindle X51 to X59			–
MP2361	Time constant for braking the spindle during EMERGENCY STOP Input: 0.01 to 5.00 [s]			–
MP2391 MP2391.0 MP2391.1	Limiting the braking power Input: 0.1 to 3000.0 [kW] 0 = do not limit braking power Wye Delta			–

Machine parameter	Function and input	Change via	Reaction	Page
MP2393[1)] *MP2393.0* *MP2393.1*	*Limiting the power of the spindle motor* *Input: 0.1 to 3000.0 [kW]* *0 = no limit* *Wye* *Delta*			–
MP2400 MP2400.0-5	Gain for axis current controllers at standstill Input: 0.00 to 9 999.99 [V/A] 0 = controller disabled Axis 1 to axis 6			4–74
MP2401	Gain for spindle current controller at standstill, only TNC 426 (9000) Input: 0.00 to 9 999.99 [V/A] 0 = controller disabled			4–114
MP2402 MP2402.0-5	Gain for axis current controller at maximum speed Input: 0.00 to 9 999.99 [V/A] 0 = value from MP2400.x Axis 1 to axis 6			4–74
MP2403	Gain for spindle current controller at maximum speed, only TNC 426 (9000) Input: 0.00 to 9 999.99 [V/A] 0 = value from MP2401.x			4–114
MP2421 MP2421.0 MP2421.1	Proportional factor of the spindle current controller, only TNC 430 and TNC 426 (15 000) Input: 0.00 to 9 999.99 [V/A] Wye Delta			4–115
MP2431 MP2431.0 MP2431.1	Integral factor of the spindle current controller, only TNC 430 and TNC 426 (15 000) Input: 0 to 9 999 999 [V/As] Wye Delta			4–115
MP2500 MP2501.0-5	Proportional factor of the rotational speed controller Input: 0 to 100.000 [As] Axis 1 to axis 6	PLC		4–68
MP2501 MP2501.0 MP2501.2	Proportional factor of the rotational speed controller for the spindle Input: 0 to 100.000 [As] Wye Delta, only TNC 430 and TNC 426 (15 000)	PLC		4–114
MP2510 MP2510.0-5	Integral factor of the speed controller Input: 0 to 30 000 [A] Axis 1 to axis 6	PLC		4–68

[1)] *As of NC software 270 476-xx*

Machine parameter	Function and input	Change via	Reaction	Page
MP2511 MP2511.0 MP2511.2	Integral factor of the rotational speed controller for the spindle Input: 0 to 30 000 [A] Wye Delta, only TNC 430 and TNC 426 (15 000)	PLC		4–114
MP2512 MP2512.0-5	Threshold for the integral factor of the speed controllers Input: 0.000 to 30.000 [s] 0 = inactive (recommended input value) Axis 1 to axis 6	PLC		4–25; 4–70
MP2520 MP2520.0-5	Differential factor of the speed controllers Input: 0 to 1.0000 [As²] Recommended: 0 Axis 1 to axis 6	PLC		4–68
MP2521 MP2521.0 MP2521.1	Differential factor of the spindle speed controller Input: 0 to 1.0000 [As²] Recommended: 0 Wye Delta, only TNC 430 and TNC 426 (15 000)	PLC		4–114
MP2530 MP2530.0-5	PT2 second-order time-delay element of the speed controllers Input: 0 to 1.0000 [s] Axis 1 to axis 6	PLC		4–69
MP2531 MP2531.0 MP2531.1	PT2 second-order time-delay element of the spindle controller Input: 0 to 1.0000 [s] Wye Delta, only TNC 430 and TNC 426 (15 000)	PLC		4–114
MP2540 MP2540.0-5	Damping for frequency filter (axes) Input: 0.0 to 18.9 [dB] Axis 1 to axis 6	PLC		4–69
MP2541	Damping for frequency filter (spindle) Input: 0.0 to 18.0 [dB]	PLC		4-114
MP2550 MP2550.0-5	Nominal frequency for freq. filter (axes) Input: 0.0 to 999.9 [Hz] Axis 1 to axis 6	PLC		4–69
MP2551	Nominal frequency for frequency filter (spindle) Input: 0.0 to 999.9 [Hz]	PLC		4-114
MP2560 MP2560.0-8	Low pass filter (axes) Input: 0 = No low pass filter 1 = Low pass filter 1st order 2 = Low pass filter 2nd order Axis 1 to 9	PLC		–

Machine parameter	Function and input	Change via	Reaction	Page
MP2561	Low pass filter (spindle) Input: 0 = No low pass filter 1 = Low pass filter 1st order 2 = Low pass filter 2nd order	PLC		–
MP2600 MP2600.0-5	Acceleration feedforward Input: 0.0000 to 3.0000 [As²)/U] Axis 1 to axis 6	PLC		4–70
MP2610 MP2610.0-5	Friction compensation at low motor speed Input: 0 to 30.0000 [A] 0 = No friction compensation Axis 1 to axis 6	PLC		4–25
MP2612 MP2612.0-5	Delay of friction compensation Input: 0.0000 to 1.0000 [s] Axis 1 to axis 6	PLC		4–25
MP2620 MP2620.0-5	Friction compensation at rated speed Input: 0 to 30.0000 [A] 0 = No friction compensation Axis 1 to axis 6	PLC		4–25
MP2630 MP2630.0-5	Holding current Input: –30.000 to +30.000 [A] Axis 1 to axis 6	PLC		4–70
MP2800 MP2800.0-5	Monitoring for position and speed Input: 0 to 99 999.9999 [mm] 0 = No monitoring Axis 1 to axis 6	PLC		4–81
MP2900 MP2900.0-8	Torque bias between master and slave in master-slave torque control Input: –100.00 to +100.00 [Nm] Axis 1 to axis 9	PLC		–
MP2910 MP2910.0-8	Gain of the torque controller in master-slave torque control Input: 0.00 to 999.99 [1/(Nm*min)] Axis 1 to axis 9	PLC		–
MP2920 MP2920.0-8	Factor for variable torque distribution in master-slave torque control Input: 0.000 to 100.000 Axis 1 to axis 9	PLC		–
MP2930 MP2930.0-8	Speed rating factor between torque master and torque slave in master-slave torque control Input: –100.00 to +100.00 [%] Axis 1 to axis 9	PLC		–

9.3.6 Spindle

Machine parameter	Function and input	Change via	Reaction	Page
MP3010	Output of rotational speed, gear range Input: 0 = No output of spindle speed 1 = Coded output of spindle speed, only if the speed changes 2 = Coded output of spindle speed on each TOOL CALL 3 = Analog output of spindle speed, but gear change signal only if the speed changes 4 = Analog output of spindle speed, but gear change signal on each TOOL CALL 5 = Analog output of spindle speed, but no gear change signal 6 = Same as input value 3 but with controlled spindle for orientation 7 = Same as input value 4 but with controlled spindle for orientation 8 = Same as input value 5 but with controlled spindle for orientation	PLC	RESET	4–94
MP3011	Function of S-analog output if MP3010 < 3 Input: 0 = No special function 1 = Voltage proportional to actual feed rate (depending on MP3012) 2 = Voltage defined via PLC module 9130 3 = Voltage defined via M functions (M200 to M204)		RESET	4–297
MP3012	Feed rate for output of an analog voltage of 10 V (MP3011 = 1) Input: 0 to 300 000 [mm/min]			4–297
MP3013.0-11	Characteristic curve kink point for output of analog voltage with M202 Input: 10 to 300 000 [mm/min]	PLC	RUN	4–298
MP3014.0-11	Characteristic curve kink point for output of analog voltage with M202 Input: 0.000 to 9.999 [V]	PLC	RUN	4–298
MP3020	Define spindle speed range Input: 0 to 99 999	PLC		4–102
MP3030	Axis halt on a TOOL CALL for which only spindle speed is output Input: 0 = Axis halt on TOOL CALL 1 = No axis halt on TOOL CALL	PLC		4–300
MP3120	Spindle speed 0 permitted Input: 0: S = 0 permitted 1: S = 0 not permitted	PLC		4–98

Machine parameter	Function and input	Change via	Reaction	Page
MP3130	Polarity of the S-analog voltage Input: 0 = M03 positive; M04 negative 1 = M03 negative; M04 positive 2 = M03 and M04 positive 3 = M03 and M04 negative	PLC	RUN	4–97
MP3140	Counting direction for the spindle encoder signals Input: 0 = Positive counting direction with M03 1 = Negative counting direction with M03	PLC	RUN	4–97
MP3142	Line count of the position encoder on the spindle Input: 100 to 9 999 [lines]	PLC	RUN	4–95
MP3143	Mounting mode of spindle position encoder 0 = Position encoder directly on the spindle 1 = Position encoder via gear ratio defined in MP3450.x and MP3451.x Input at X30 pin 1 = reference pulse 2 = Position encoder via gear ratio defined in MP3450.x and MP3451.x Input at X30 pin 1 = reference pulse release 3 = As input value 1, however TNC waits to evaluate second reference pulse	PLC	RUN	4–95
MP3210 MP3210.0-7	**Analog axes:** S-analog voltage at rated speed Input: 0 to 100.000 [V] **Digital axes:** Motor revolutions at rated speed Input: 0 to 100.000 [1000/min] Gear range 1 to 8	PLC	RUN	4–98
MP3240.1	**Analog axes:** minimum S-analog voltage output Input: 0 to 9.999 [V] **Digital axes:** minimum motor speed output Input: 0 to 9.999 [1000/min]	PLC	RUN	4–98
MP3240.2	**Analog axes:** jog voltage for gear change Input: 0 to 9.999 [V] **Digital axes:** motor speed for gear change Input: 0 to 9.999 [1000/min]	PLC	RUN	4–100

Machine parameter	Function and input	Change via	Reaction	Page
MP3310.0-1	Limits for spindle override Input: 0 to 150 [%]	PLC	RUN	4–101
MP3310.0	Upper limit			
MP3310.1	Lower limit			
MP3411	Ramp gradient of the spindle for M03 and M04 in 8 gear ranges Input: **Analog spindle:** 0.000 to 1.999 [V/ms] **Digital spindle:** 0.000 to 1.999 [(1000/min)/ms]	PLC	RUN	4–97
MP3411.0-7	Gear range 1 to gear range 8			
MP3412	Multiplier for MP3411.x Input: 0.000 to 1.999	PLC	RUN	
MP3412.0	for M05			4–97
MP3412.1	for spindle orientation			4–106
MP3412.2	for tapping			4–110
MP3412.3	for rigid tapping			4–113
MP3415	Transient response of the spindle Input: 0 to 1 000 [ms]	PLC	RUN	
MP3415.0	for M03, M04 and M05			4–97
MP3415.1	for spindle orientation			4–106
MP3415.2	for tapping			4–110
MP3415.3	for rigid tapping			4–113
MP3420	Positioning window for the spindle Input: 0 to 360.0000 [°]	PLC	RUN	4–106
MP3430	Deviation of the reference mark from the desired position (spindle preset) Input: 0 to 360 [°]	PLC	RUN	4–106
MP3440	k_v factor for spindle orientation Input: 0.1 to 10 [$\frac{1000°/\text{min}}{°}$]	PLC	RUN	4–106
MP3440.0-7	Gear range 1 to gear range 8			
MP3450	Number of spindle motor revolutions Input: 0 to 65 535 0 = no ratio	PLC	RUN	4–95
MP3450.0-7	Gear range 1 to gear range 8			
MP3451	Number of spindle revolutions Input: 0 to 65 535 0 = no ratio	PLC	RUN	4–95
MP3451.0-7	Gear range 1 to gear range 8			
MP3510	Nominal speed for gear ranges Input: 0 to 99 999.999 [rpm]	PLC		4–98
MP3510.0-7	Gear range 1 to gear range 8			
MP3515	Maximum spindle speed for gear range Input: 0 to 99 999.999 [rpm]	PLC		4–101
MP3515.0-7	Gear range 1 to gear range 8			

Machine parameter	Function and input	Change via	Reaction	Page
MP3520.0	Spindle speed activated by marker M4011 Input: 0 to 99 999.999 [rpm]	PLC	RUN	4–107
MP3520.1	Spindle speed for oriented stop Input: 0 to 99 999.999 [rpm]	PLC	RUN	4–106

9.3.7 Integral PLC

Machine parameter	Function and input	Change via	Reaction	Page
MP4020	PLC compatibility with TNC 415 / TNC 425 Input: %xxxxx Bit 0 = Convert "axis" word W1024 and following to markers Bit 1 = Convert new markers to old markers Bit 2 = Convert configuration bits from MP4310 to markers (M2192 to M2239 and M3200 to M3263) Bit 3 = Error message markers remain available Bit 4 = Non-volatile markers remain in range M1000 to M1999 Bit 5 = 0: one-spindle operation 1: two-spindle operation Bit 6= reserved Bit 7= 0: transfer Pt100-input values with 1 K/s. 1: transfer Pt100-input values immediately		RESET	5–36
MP4030[1)] *MP4030.0-3*	*Assignment physical PL / logical PL* *Input: 0 = first logical PL* *1 = second logical PL* *2 = third logical PL* *3 = fourth logical PL* *First to fourth physical PL*	*PLC*		–
MP4060 MP4060.0-8	Path-dependent lubrication Input: 0 to 99 999.9999 [mm] Axis 1 to axis 9	PLC	RUN	4–10
MP4070	Compensation per PLC cycle for lag-tracking error compensation Input: 0.0001 to 0.005 [mm]	PLC	RUN	4–20
MP4110.0-47	Time for timers T0 to T47 Input: 0 to 65 535 [PLC cycles]	PLC	RUN	5–26
MP4120.0-31	Preset value for counter C0 to C31 Input: 0 to 65 535 [PLC cycles]	PLC	RUN	5–28

[1)] As of NC software 280 476-xx

Machine parameter	Function and input	Change via	Reaction	Page
MP4130	Numerical designation of fast PLC inputs Input: 0 to 255 [No. of the PLC input]			
MP4130.0	Fast PLC input for suppressing the monitoring functions			4–79
MP4130.1	Without function			
MP4130.2	Fast PLC input sets M4590 (signal duration > 4 ms)			5–29
MP4130.3	Fast PLC input sets M4591 (signal duration > 4 ms)			5–29
MP4130.4	Fast PLC input sets M4592 (signal duration > 4 ms)			5–29
MP4130.5	Fast PLC input sets M4593 (signal duration > 4 ms)			5–29
MP4131	Condition for activating the fast PLC input from MP4130 Input: 0 = activation at low level 1 = activation at high level			
MP4131.0	Fast PLC input for suppressing the monitoring functions			4–79
MP4131.1	Without function			
MP4131.2	Fast PLC input sets M4590 (signal duration > 4 ms)			5–29
MP4131.3	Fast PLC input sets M4591 (signal duration > 4 ms)			5–29
MP4131.4	Fast PLC input sets M4592 (signal duration > 4 ms)			5–29
MP4131.5	Fast PLC input sets M4593 (signal duration > 4 ms)			5–29
MP4210.0-47	Set a number in the PLC (D768 to D956) Input: –99 999.9999 to +99 999.9999			5–16
MP4220.0-4	Set a number in the PLC (word range W960 to W968) Input: 10 to 30 000			5–16
MP4230.0-31	Set a number in the PLC (module 9032) Input: –99 999.9999 to +99 999.9999			5–16
MP4231.0-31	Set a number in the PLC (module 9032) Input: –99 999.9999 to +99 999.9999			5–16
MP4310.0-6	Set a number in the PLC (W976 to W988) Input: 0 to 65 535			5–16

9.3.8 Configuration of Data Interfaces

Machine parameter	Function and input	Change via	Reaction	Page
MP5000	Disable data interface Input: 0 = No interface disabled 1 = RS-232-C/V.24 interface disabled 2 = RS-422/ V.11 interface disabled	PLC	RUN	6–14
MP5020.0 MP5020.1 MP5020.2	Operating mode EXT1 Operating mode EXT2 Operating mode EXT3 (PLC) Input: %xxxxxxx Bit 0 7 or 8 data bits 0 = 7 data bits 1 = 8 data bits Bit 1 Block Check Character 0 = Any block check character 1 = BCC not control character Bit 2 Transfer stop by RTS 0 = Not active 1 = Active Bit 3 Transmission stop by DC3 0 = Not active 1 = Active Bit 4 Character parity 0 = Even 1 = Odd Bit 5 Character parity 0 = Not required 1 = Required Bit 6/7 Stop bits Bit 6 Bit 7 1½ Stop bits 0 0 2 Stop bits 1 0 1 Stop bits 0 1 1 1	PLC CN123	RUN	6–17
MP5030.0 MP5030.1 MP5030.2	Operating mode EXT1 Operating mode EXT2 Operating mode EXT3 (PLC) Input: 0 = Standard data transfer 1 = Blockwise transfer 2 = Without protocol (only MP5030.2)	PLC CN123	RUN	6–17
MP5040	Data transfer rate in operating mode EXT3 (data transfer via PLC) Input: 0 = 110 baud 6 = 4800 baud 1 = 150 baud 7 = 9600 baud 2 = 300 baud 8 = 19 200 baud 3 = 600 baud 9 = 38 400 baud 4 = 1200 baud 10 = 57 600 baud 5 = 2400 baud 11 = 115 200 baud	PLC	RUN	6–31

9.3.9 3-D Touch Probe

Machine parameter	Function and input	Change via	Reaction	Page
MP6010	Selection of touch probe Input: 0 = Touch probe with cable transmission 1 = Touch probe with infrared transmission	PLC CN123		4–206
MP6120	Probing feed rate (touch trigger probe) Input: 1 to 3 000 [mm/min]	PLC CN123	RUN	4–206
MP6130	Maximum measuring range Input: 0.001 to 99 999.9999 [mm]	PLC CN123	RUN	4–206
MP6140	Safety clearance above measurement point Input: 0.001 to 99 999.9999 [mm]	PLC CN123	RUN	4–206
MP6150	Rapid traverse in probe cycle (touch trigger probe) Input: 10 to 20 000 [mm/min]	PLC CN123	RUN	4–206
MP6160	M function for probing from opposite orientations Input: –1 = Spindle orientation directly via NC 0 = Function inactive 1 to 88 = Number of the M function for spindle orientation via PLC	PLC CN123	RUN	4–206
MP6161	M-function to orient infrared probe before each measurement Input: –1 = Spindle orientation directly via NC 0 = Function inactive 1 to 88 = Number of the M function for the spindle orientation	PLC SZ123	RUN	–
MP6162	Orientation angle for infrared probe Input: 0 to 359.9999 [°]	PLC SZ123	RUN	–
MP6163	Difference between current spindle angle and value in MP6162 from which a spindle orientation is to be carried out. Input: 0 to 3.0000 [°]	PLC SZ123	RUN	–
MP6170	Number of measurements in a programmed touch-probe block Input: 1 to 3	PLC CN123	RUN	4–207
MP6171	Confidence range in a programmed measurement (MP6170 > 1) Input: 0.002 to 0.999 [mm]	PLC CN123	RUN	4–207
MP6180	Coordinates of the calibration ring for Cycle TCH PROBE 2 (calibrate TS) referenced to the machine datum (traverse range 1) Input: –99 999.9999 to +99 999.9999 [mm]	PLC SZ123		–
MP6180.0-2	Axis X to axis Z			

Machine parameter	Function and input	Change via	Reaction	Page
MP6181 MP6181.0-2	Coordinates of the calibration ring for Cycle TCH PROBE 2 (calibrate TS) referenced to the machine datum (traverse range 2) Input: –99 999.9999 to +99 999.9999 [mm] Axis X to axis Z	PLC SZ123		–
MP6182 MP6182.0-2	Coordinates of the calibration ring for Cycle TCH PROBE 2 (calibrate TS) referenced to the machine datum (traverse range 3) Input: –99 999.9999 to +99 999.9999 [mm] Axis X to axis Z	PLC SZ123		–
MP6185	Distance below the ring edge, which is to be probed during calibration. Input: 0.001 to 99 999.9999 [mm]	PLC SZ123		–

9.3.10 Digitizing with TS (Available Only with Digitizing Option)

Machine parameter	Function and input	Change via	Reaction	Page
MP6200	Selection of touch trigger or measuring probe (only with 'Digitizing with measuring probe' option) Input: 0 = Touch trigger probe(e.g. TS 120) 1 = Measuring touch probe	PLC CN123		4–206; 4–220
MP6210	Number of oscillations per second in normal direction Input: 0 to 65.535 [1/s]	PLC CN123	RUN	4–213
MP6220	Traverse Distance for lubricating the touch probe axis at end of line Input: 0.000 to 999.999 [mm]	PLC CN123	RUN	4–214
MP6221	Time after which the probe axis must be lubricated Input: 0 to 65 535 [min]	PLC CN123	RUN	4–214
MP6230	Feed rate in normal direction Input: 0 to 1 000 [mm/min]	PLC CN123	RUN	4–213
MP6240	Maximum stylus deflection Input: 0 to 10.000 [mm]	PLC CN123	RUN	4–213
MP6260	Output of M90 for NC blocks with digitized data Input: 0 = No output of M90 1 = Output of M90 In every NC block	PLC CN123	RUN	4–213
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 0.000 1-mm steps (0.1 µm)	PLC CN123	RUN	4–213

9.3.11 Digitizing with Measuring Touch Probe (Available Only with Digitizing Option)

Machine parameter	Function and input	Change via	Reaction	Page
MP6310	Stylus deflection depth (measuring touch probe) Input: 0.1000 to 2.0000 [mm]	PLC CN123		4–220
MP6320	Counting direction of the measuring system signals (measuring touch probe) Input: %xxx Bit 0 to 2 0 = Positive Axis 1 to 3 1 = Negative	CN123		4–220
MP6321	Measuring the center offset in calibration of the measuring touch probe Input: 0 = Calibrate and measure center offset 1 = Calibrate without measuring center offset	CN123		4–206; 4–220
MP6322	Assignment of measuring probe axes to machine axes Input: 0 = Probe axis X 1 = Probe axis Y 2 = Probe axis Z	CN123		4–220
MP6322.0	Machine axis 1			
MP6322.1	Machine axis 2			
MP6322.2	Machine axis 3			
MP6330	Maximum stylus deflection (measuring touch probe) Input: 0.100 to 4.000 [mm]	CN123		4–220
MP6350	Feed rate for positioning to the MIN point and contour approach (measuring touch probe) Input: 1 to 3 000 [mm/min]	CN123		4–220
MP6360	Scanning feed rate (measuring touch probe) Input: 1 to 3 000 [mm/min]	CN123		4–206; 4–220
MP6361	Rapid traverse in scanning cycle (measuring touch probe) Input: 10 to 20 000 [mm/min]	CN123		4–206; 4–221
MP6362	Feed rate reduction if stylus of measuring touch probe is deflected sideways Input: 0 = Feed rate reduction not active 1 = Feed rate reduction active	PLC CN123		4–221

Machine parameter	Function and input	Change via	Reaction	Page
MP6370	Radial acceleration during digitizing with measuring touch probe Input: 0.001 to 3.000 [m/s²] Recommended: 0.1	PLC CN123	RUN	4–221
MP6390	Target window for contour line Input: 0.1000 to 4.0000 [mm]	PLC CN123		4–221

9.3.12 Tool Measurement with TT

Machine parameter	Function and input	Change via	Reaction	Page
MP6500	Tool measurement with TT Input: %xxxxxxxxxxxxx Bit 0: 0 = Cycles for tool measurement disabled 1 = Cycles for tool measurement not disabled Bit 1: 0 = Tool radius measurement permitted; tool length measurement with rotating spindle 1 = Tool radius measurement and individual edge measurement disabled Bit 2: 0 = Tool length measurement with stationary spindle (bit 1 = 1) 1 = Tool length measurement with rotating spindle. The tool length is then calibrated with the spindle rotating only if a tool radius offset (TT:R-OFFS) is entered in the tool table. Bit 3: 0 = Tool measurement with oriented spindle stop 1 = Tool measurement without spindle orientation; individual edge measurement not possible, the tool radius measurement may be incorrect in some circumstances Bit 4: 0 = Measuring speed is limited to maximum 1000 rpm 1 = Measuring speed is not limited	PLC		4–227

Machine parameter	Function and input	Change via	Reaction	Page
	Bit 5: NC stop during tool check 0 = The NC program is not stopped when the breakage tolerance is exceeded 1 = If the breakage tolerance is exceeded, the NC program is stopped and the error message "Tool broken" is output 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 "Tool broken" is output Bit 7: *Reserved* Bit 8: Probing routine 0 = The probe contact is approached from several directions 1 = The probe contact is approached from only one direction Bit 9: Automatic determination of the basic rotation for the probe contact (bit 8 = 1) 0 = The basic rotation is not determined 1 = The basic rotation for the probe contact is determined automatically Bit 10: Probing routine (bit 8 = 1) 0 = The starting point is pre-positioned in all three axes 1 = The starting point is only pre-positioned in the tool axis and the axis for probing direction (MP6505) (bit 9 = 0) Bit 11: TOOL check and editing the tool table 0 = The tool table is edited after tool check 1 = The tool table is not edited after tool check Bit 12: 0 = Do not include the PLC datum shift 1 = Include the PLC datum shift Bit 13: 0 = Tool is measured in the same tilted system as the TT was calibrated 1 = Tool is measured in a different tilted system			4–227

Machine parameter	Function and input	Change via	Reaction	Page
MP6505 MP6505.0-2	Probing direction for tool measurement Input: 0 = Positive probing direction in the angle reference axis (0° axis) 1 = Positive probing direction in +90° axis 2 = Negative probing direction in the angle reference axis (0° axis) 3 = Negative probing direction in the +90° axis Traverse range 1 to traverse range 3	PLC CN123		4–228
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		4–228
MP6510	Max. permissible measuring error for tool measurement with rotating tool Input: 0.002 to 0.999 [mm]	PLC CN123		4–228
MP6520	Probing feed rate for tool measurement with non-rotating tool Input: 1 to 3 000 [mm/min]	PLC CN123	RUN	4–228
MP6530 MP6530.0-2	Distance from tool lower edge to probe contract upper edge for tool radius measurement Input: 0.001 to 99.9999 [mm] Traverse range 1 to traverse range 3	PLC CN123		4–228
MP6531 MP6531.0-2	Diameter or edge length of the TT 120 probe contact Input: 0.001 to 99 999.9999 [mm] Traverse range 1 to traverse range 3	PLC		4–228
MP6540 MP6540.0 MP6540.1	Safety zone around the TT 120 probe contact for pre-positioning Input: 0.001 to 99 999.9999 [mm] Safety clearance in tool axis direction Safety clearance in the plane perpendicular to the tool axis	PLC CN123		4–228
MP6550	Rapid traverse in probing cycle for TT 120 Input: 10 to 20 000 [mm/min]	PLC CN123	RUN	4–228
MP6560	M function for spindle orientation with individual-tooth calibration Input: –1 = Spindle orientation directly via NC 0 = Function inactive 1 to 88 = Number of the M function for spindle orientation via PLC	PLC CN123	RUN	4–228

Machine parameter	Function and input	Change via	Reaction	Page
MP6570	Max. permissible surface cutting speed at the tool edge Input: 1.0000 to 120.0000 [m/min]	PLC CN123		4–228
MP6580 MP6580.0-2	Coordinates of the TT 120 probe center in relation to the machine datum (traverse range 1) Input: –99 999.9999 to +99 999.9999 [mm] Axis X to axis Z	PLC CN123		4–228
MP6581 MP6581.0-2	Coordinates of the TT 120 probe center in relation to the machine datum (traverse range 2) Input: –99 999.9999 to +99 999.9999 [mm] Axis X to axis Z	PLC CN123		4–229
MP6582 MP6582.0-2	Coordinates of the TT 120 probe center in relation to the machine datum (traverse range 3) Input: –99 999.9999 to +99 999.9999 [mm] Axis X to axis Z	PLC CN123		4–229
MP6585[1]	*Monitoring the positions of the rotary and additional linear axes during the tool measurement cycles* *Input: %xxxxxx* *0 = axis is not monitored* *1 = axis is monitored* *Bit0 A-axis* *Bit1 B-axis* *Bit2 C-axis* *Bit3 U-axis* *Bit4 V-axis* *Bit5 W-axis*	*PLC* *SZ123*		–
MP6586[1] *MP6586.0-5*	*Ref-coordinate for the monitoring of the rotary and additional linear axis positions during the tool measurement cycles* *Input: –99 999.9999 to 99 999.9999 [mm]* *A-axis to W-axis*	*PLC* *SZ123*		–

[1] *As of NC software 280 476-xx*

9.3.13 Tapping

Machine parameter	Function and input	Change via	Reaction	Page
MP7110.0	Minimum feed rate override in tapping Input: 0 to 150 [%]	PLC	RUN	4–110
MP7110.1	Maximum feed rate override in tapping Input: 0 to 150 [%]	PLC	RUN	4–110
MP7120.0	Dwell time for reversal of spindle rotation direction Input: 0 to 65.535 [s]	PLC	RUN	4–110
MP7120.1	Advanced switching time of the spindle for tapping with coded output Input: 0 to 65.535 [s]	PLC	RUN	4–111
MP7120.2	Spindle slow-down time after reaching the total hole depth Input: 0 to 65.535 [s]	PLC	RUN	4–110
MP7130	Deceleration behavior of spindle for rigid tapping Input: 0.001 to 10 [°/min]	PLC		4–113
MP7150	Positioning window of tool axis Input: 0.0001 to 2.0000 [mm]	PLC		4–113
MP7160	Input: %xxx Bit 0 0 = Spindle orientation before execution of Cycle 17 1 = No spindle orientation before execution of Cycle 17 Bit 1 0 = Spindle speed is not limited 1 = Spindle speed is limited for small thread depths Bit 2 Rigid tapping 0 = No position control for spindle 2 = position control for spindle	PLC CN123	RUN	4–113

9.3.14 Display and Operation

Machine parameter	Function and input	Change via	Reaction	Page
MP7210	Programming station Input: 0 = controlling and programming 1 = programming station with PLC active 2 = programming station with PLC inactive	CN123		4–188
MP7212	POWER INTERRUPTED message Input: 0 = POWER INTERRUPTED message must be acknowledged with the CE key 1 = POWER INTERRUPTED message does not appear	PLC CN123	RUN	4–191
MP7220	Block number increment for ISO programs Input: 0 to 250	PLC CN123	RUN	4–171
MP7224.0	Disable soft keys for file types Input: %xxx xxxxx Bit 0 HEIDENHAIN programs Bit 1 ISO programs Bit 2 Tool tables Bit 3 Datum tables *Bit 4 Pallet tables (NC software 280 470 ..)* Bit 5 Text files Bit 6 Help files Bit 7 Point tables 0 = do not disable 1 = disable	PLC CN123	RUN	4–171
MP7224.1	Protect file types Input: %xxxxxxx Bit 0 HEIDENHAIN programs Bit 1 ISO programs Bit 2 Tool tables Bit 3 Datum tables *Bit 4 Pallet tables (NC software 280 470 ..)* Bit 5 Text files Bit 6 Help files Bit 7 Point tables 0 = Not protected 1 = Protected	PLC CN123	RUN	4–171
MP7226.0	*Size of pallet tables (NC software 280 470 ..) Input: 0 to 255 [lines]*	*PLC CN123*	*RUN*	*4–176*
MP7226.1	Size of datum tables Input: 0 to 255 [lines]	PLC CN123	RUN	4–172

Machine parameter	Function and input	Change via	Reaction	Page
MP7229.0	Program length for program verification Input: 100 to 9999	PLC CN123	RUN	–
MP7229.1	Program length up to which FK blocks are permitted Input: 100 to 9999	PLC CN123	RUN	–
MP7230	Dialog language selection 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	PLC CN123	RUN	4–189
MP7230.0	NC dialog language			
MP7230.1	PLC dialog language (OEM cycles, users parameters), soft keys for OEM cycles PLC error messages			
MP7230.2	Help files			
MP7230.3				
MP7235	Difference from Universal Time (Greenwich Mean Time) Input: –23 to +23 [hours] 0 = Universal Time –5 = Eastern Standard Time (EST) –4 = EST during Daylight Savings Time	PLC CN123	RUN	4–194
MP7237	PLC counter display and resetting Input: %xxxxxxxx	PLC	RUN	4–191
MP7237.0	Display the PLC counter			
MP7237.1	Enable reset of the PLC counter with code number 857 282			
MP7237.0-1	Bit 0 to 7 PLC counter 1 to 8 0 = no reset; 1 = reset			
MP7237.2	Enable reset of the NC counter with code number 857282 Bit 1: MACHINE ON timer Bit 2: PROGRAM RUN timer 0 = no reset; 1 = reset	PLC	RUN	4–191
MP7238	Dialogs for PLC counter Input: 0 to 4 095 [Dialogue no. from file defined by PLCDIALOG= (OEM.SYS)]	PLC	RUN	4–192
MP7238.0-7	Dialog for PLC counters 1 to 8			

Machine parameter	**Function and input**	**Change via**	**Reaction**	**Page**
MP7245	Auxiliary cycles (cycle 18, cycle 33) Input: 0 = auxiliary cycles disabled 1 = auxiliary cycles permitted	PLC	RUN	–
MP7246	Disable paraxial positioning blocks Input: 0 = Enable paraxial positioning blocks 1 = Disable paraxial positioning blocks	PLC	RUN	4–190
MP7251	Number of global Q-parameters transferred from the OEM cycle to the calling program Input: 0 to 40	PLC	RUN	7–2
MP7260	Number of the tools in the tool table Input: 0 to 30 000	CN123		4–248
MP7261	Number of pockets in the tool magazine Input: 0 to 32 767	CN123		4–248
MP7262	Maximum index number for indexed tools Input: 0 to 9	CN123		–
MP7263	Show "Pocket table" soft key Input:: 0 = Show "Pocket table" soft key 1 = Do not show "Pocket table" soft key	CN123		–
MP7266	Elements of the tool table Input: 0 = No display 1 to 99 = Position in the tool table	CN123		4–248
MP7266.0	Tool name (NAME)			
MP7266.1	Tool length (L)			
MP7266.2	Tool radius (R)			
MP7266.3	Tool radius 2 (R2)			
MP7266.4	Oversize for tool length (DL)			
MP7266.5	Oversize for tool radius (DR)			
MP7266.6	Oversize for tool radius 2 (DR2)			
MP7266.7	Tool locked? (TL)			
MP7266.8	Replacement tool (RT)			
MP7266.9	TIME 1			
MP7266.10	TIME 2			
MP7266.11	CURRENT TIME			
MP7266.12	Comment on the tool (DOC)			
MP7266.13	Number of cutting edges (CUT)			
MP7266.14	Wear tolerance for length (LTOL)			
MP7266.15	Wear tolerance for radius (RTOL)			
MP7266.16	Cutting direction of tool (DIRECT)			
MP7266.17	PLC status (PLC)			
MP7266.18	Tool offset for length (TT: L-OFFS)			
MP7266.19	Tool offset for radius (TT: R-OFFS)			

Machine parameter	**Function and input**	**Change via**	**Reaction**	**Page**
MP7266.20 MP7266.21 MP7266.22 MP7266.23 MP7266.24 MP7266.25 MP7266.26 MP7266.27 *MP7266.28*[1)] *MP7266.29*[1)] *MP7266.30*[1)]	Break tolerance for length (LBREAK) Break tolerance for radius (RBREAK) Length of cutting edge (LCUTS) Plunge angle (ANGLE) Tool type (TYP) Tool material (TMAT) Cutting data table (CDT) PLC value (PLC-VAL) *Probe center misalignment in main axis (CAL-OF1)* *Probe center misalignment in secondary axis (CAL-OF2)* *Spindle angle during calibration (CAL-ANG)*			
MP7267 MP7267.0 MP7267.1 MP7267.2 MP7267.3 MP7267.4 MP7267.5	Elements of the pocket table Input: 0 = No display 1 to 99 = Position in the pocket table Tool number (T) Special tool (ST) Fixed pocket (F) Locked pocket (L) PLC status (PLC) Tool name (TNAME)	CN123		4–248
MP7270	Display of feed rate in manual modes (manual operation, electronic handwheel) Input: 0 = Display of the axis feed rate only when an axis-direction button is pressed (axis-specific feed from MP1020.X) 1 = Display axis feed rate even before axis-direction button is pressed (smallest value from MP1020.x for all axes)	PLC CN123	RUN	4–127
MP7280	Decimal character Input: 0 = Decimal comma 1 = Decimal point	PLC CN123	RUN	4–190
MP7285	Offset tool length in the position display of the tool axis Input: 0 = Tool length is not offset 1 = Tool length is offset	PLC CN123	RUN	4–124
MP7289	Position display step of the spindle Input: see MP7290	PLC SZ123	RUN	–

[1)] as of NC software 280 476-xx

Machine parameter	Function and input	Change via	Reaction	Page
MP7290 MP7290.0-8	Position display step 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° Axis 1 to axis 9	PLC CN123	RUN	4–124
MP7291 MP7291.0-2	Displayed axes Input: SWVUCBAZYXwvucbazyx Capital letters for NC axes Small letters for PLC axes Hyphen for inactive axes S = spindle Characters 1 to 9 (1 = right) Lines 1 to 9 Character 10 = spindle (S) (display always in line 9) Traverse range 1 to traverse range 3			4–1
MP7295	Disable the datum-setting function Input: %xxxxx Bit 0 to 8 0 = Enabled Axis 1 to 9 1 = Disabled	PLC CN123	RUN	4–118
MP7296	Datum setting via axis keys Input: 0 = Datum can be set via axis keys or soft key 1 = Datum can only be set via soft key	PLC CN123	RUN	4–118

Machine parameter	Function and input	Change via	Reaction	Page
MP7300	Cancel status display and Q parameters Input: 0 = Cancel the status display, Q parameters and tool data when a program is selected 1 = Cancel the status display, Q parameters and tool data with M02, M30, END PGM and when a program is selected 2 = Cancel the status display and tool data when a program is selected 3 = Cancel the status display and tool data with M02, M30, END PGM, and when a program is selected 4 = Cancel the status display and Q parameters when a program is selected 5 = Cancel the status display and Q parameters when a program is selected and with M02, M30, END PGM 6 = Cancel the status display when a program is selected 7 = Cancel the status display when a program is selected and with M02, M30, END PGM	PLC CN123	RUN	4–129
MP7310	Graphic representation Input: %xxxxx Bit 0: Change view in 3 planes 0 = Projection preferred in Germany 1 = Projection preferred in America Bit 1: Rotating the coordinate system in the working plane by +90° 0 = No rotation 1 = Coordinate system rotated by +90° Bit 2: BLK form after a datum shift 0 = BLK form is not shifted 1 = BLK form is shifted Bit 3: Display of cursor position during view in three planes 0 = No display 1 = Display of the cursor position	PLC CN123	RUN	4–123
MP7315	Tool radius for graphic display without TOOL CALL Input: 0.0000 to 99 999.9999 [mm]	PLC CN123	RUN	4–299
MP7316	Tool penetration depth Input: 0.0000 to 99 999.9999 [mm]	PLC CN123	RUN	4–299

Machine parameter	Function and input	Change via	Reaction	Page
MP7317.0	M function to start graphic display Input: 0 to 88	PLC CN123	RUN	4–299
MP7317.1	M function to interrupt graphic display Input: 0 to 88	PLC CN123	RUN	4–299
MP7330.0-15	Definition of parameters as user parameters Input: 0 to 9999.00 (no. of the desired machine parameters)	PLC	RUN	4–187
MP7340.0-15	Dialogs for user parameters Input: 0 to 4 095 (line number of the PLC dialog)	PLC	RUN	4–187

9.3.15 Colors

Machine parameter	Function and input		Change via	Reaction	Page
MP7350	Window frame	$030200C	PLC	RUN	4–119
MP7351	Error messages	$03F3F0F	PLC	RUN	
MP7352	"Machine" operating mode display		PLC	RUN	
MP7352.0	Background	$0000000			
MP7352.1	Text for operating mode	$0342008			
MP7352.2	Dialog	$03F3828			
MP7353	"Programming" operating mode display		PLC	RUN	
MP7353.0	Background	$0000000			
MP7353.1	Text for operating mode	$0342008			
MP7353.2	Dialog	$03F3828			
MP7354	"Machine" program-text display		PLC	RUN	
MP7354.0	Background	$0080400			
MP7354.1	General program text	$038240C			
MP7354.2	Current block	$038341C			
MP7354.3	Background not current window	$00C0800			
MP7355	"Programming" program-text display		PLC	RUN	
MP7355.0	Background	$0080400			
MP7355.1	General program text	$038240C			
MP7355.2	Current block	$038341C			
MP7355.3	Background not current window	$00C0800			
MP7356	Status and PLC windows		PLC	RUN	
MP7356.0	Background	$00C0800			
MP7356.1	Axis positions in the status display	$03F2C18			
MP7356.2	Status display except for axis positions	$03F280C			
MP7357	"Machine" soft-key display		PLC	RUN	
MP7357.0	Background	$0000000			
MP7357.1	Symbols	$03F3828			
MP7358	"Programming" soft-key display		PLC	RUN	
MP7358.0	Background	$0000000			
MP7358.1	Symbols	$03F3828			
MP7360	Graphics: 3-D view		PLC	RUN	
MP7360.0	Background	$0000000			
MP7360.1	Top surface	$0203038			
MP7360.2	Front face	$00C1820			
MP7360.3	Text displays in the graphic window	$03F3F3F			
MP7360.4	Side surface	$0102028			

Machine parameter	Function and input	Change via	Reaction	Page
MP7361	Graphics: view in three planes	PLC	RUN	4–119
MP7361.0	Background $0000000			
MP7361.1	Top view $0203038			
MP7361.2	Front and side view $0203038			
MP7361.3	Axis cross and text in display $03F3F3F			
MP7361.4	Cursor $03F0000			
MP7362	Additional status display in the graphic window and pocket calculator	PLC	RUN	
MP7362.0	Background graphic window and pocket calculator $0080400			
MP7362.1	Background status display and keys of the pocket calculator $00C0800			
MP7362.2	Status symbols and symbols of the pocket calculator (c in cos) $038240C			
MP7362.3	Status values and texts of the pocket calculator (os in cos) $03F2C18			
MP7363	Programming graphics	PLC	RUN	
MP7363.0	Background $0000000			
MP7363.1	Resolved contour $03F3F3F			
MP7363.2	Subprograms and frame for zooming $0003F00			
MP7363.3	Alternative solutions $0003F00			
MP7363.4	Non-resolved contours $03F0000			
MP7364	Colors of the help illustrations for cycles	PLC	RUN	
MP7364.0-6	Colors 1 to 7 of the graphic program used $0000000			
MP7364.7	Line colors (color 8 of the graphic program used) $038240C			
MP7364.8	Color for highlight if defined			
MP7364.9	in help illustration $038341C			
	Background $0000000			
MP7365	Oscilloscope	PLC	RUN	
MP7365.0	Background $0000000			
MP7365.1	Channel 1 $0203038			
MP7365.2	Channel 2 $0003F00			
MP7365.3	Channel 3 $03F3F00			
MP7365.4	Channel 4 $03F003F			
MP7365.5	Selected channel $03F0000			
MP7365.6	Grid $030200C			
MP7365.7	Cursor and Text $03F3F3F			
MP7366	Superimposed window (HELP key, pop-up menus, etc.)	PLC	RUN	
MP7366.0	Background $0333333			
MP7366.1	Text or foreground $0281408			
MP7366.2	Current line$0140A04			
MP7366.3	Headline $02F2818			
MP7366.4	Scroll bar field $0100C08			
MP7366.5	Scroll bar $02F2818			
MP7366.6-14	*Reserved* Gray shades			

Machine parameter	Function and input	Change via	Reaction	Page
MP7367 MP7367.0 MP7367.1 MP7367.2 MP7367.3 MP7367.4 MP7367.5 MP7367.6-14	Large PLC window Background $0333333 Color 1 $0281408 Color 2 $0140A04 Color 3 $02F2818 Color 4 $0100C08 Color 5 $02F2818 Color 6 to color 14 Gray shades	PLC	RUN	–
MP7392	Screen saver Input: 0 to 99 [min] 0 = no screen saver 1 to 99 = time after which the screen saver switches on	PLC CN123	RUN	–

9.3.16 Machining and Program Run

Machine parameter	Function and input	Change via	Reaction	Page
MP7410	Scaling factor cycle in two or three axes Input: 0 = Scaling factor cycle effective in all three primary axes 1 = Scaling factor cycle effective only in the working plane	PLC CN123	RUN	4–165
MP7411	Tool data in touch probe block Input: Bit 0 0 = The touch probe calibrated data is used 1 = The current tool data from the last TOOL CALL is used *Bit1[1)] 0 = Only one set of probe calibration data* *1 = More than one set of calibration data can be managed in the tool table*	PLC CN123	RUN	4–207

Machine parameter	Function and input	Change via	Reaction	Page
MP7420	Cycles for milling pockets with combined contours Input: %xxxx Bit 0: Traverse direction for channels 0 = Pocket counterclockwise, island clockwise 1 = Pocket clockwise, island counterclockwise Bit 1: Sequence for clearing and channel-milling 0 = First mill the channel, then clear the pocket 1 = First clear the pocket, then mill the channel Bit 2: Merging of listed contours 0 = Contours are combined only if the tool center paths intersect 1 = Contours are combined if the programmed contours intersect Bit 3: Clearing and pocket-milling to pocket depth or for each pecking depth 0 = Each process uninterrupted to pocket depth 1 = Both processes for each pecking depth before proceeding to the next depth Bit 4: Position after machining the cycle 0 = Tool moves to the same position as before the cycle was called 1 = Tool moves only to the "clearance height"	PLC CN123	RUN	4–165
MP7430	Overlap factor for pocket milling Input: 0.1 to 1.414	PLC CN123	RUN	4–163
MP7431	Arc end-point tolerance Input: 0.0001 to 0.016 [mm]	PLC CN123	RUN	4–190

Machine parameter	Function and input	Change via	Reaction	Page
MP7440	Output of M functions Input: %xxxxx	PLC CN123	RUN	
	Bit 0: Program stop with M06 0 = Program stop with M06 1 = No program stop with M06			4–158
	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			4–158
	Bit 2 Program stop with M functions 0 = Program stop until acknowledgment of M function 1 = No program stop (do not wait for acknowledgment)			4–158
	Bit 3 Select k_V factors with M105/M106 0 = Function not effective 1 = Function effective			4–22 4–57 4–59
	Bit 4 Reduced feed rate in the tool axis with M103 0 = Function not effective 1 = Function effective			4–158
	Bit 5: Reserved Bit 6: Automatically activate M134 when selecting an NC program 0 = Function not effective 1 = Function effective			
MP7441	Error message "Spindle?" when a fixed cycle is called without M3 or M4 Input: 0 = Do not suppress error message 1 = Suppress error message	PLC CN123	RUN	–
MP7450	Calculate the tool change position from MP951 in block scan Input: %xxxxx Bit 0 to 8 0 = do not calculate Axis 1 to 9 1 = calculate	PLC	RUN	4–167
MP7451	Feed rate for returning to the contour Input: 10 to 300 000 [mm/min]	PLC		4–167
MP7451.0-8	Axis 1 to axis 9			
MP7460	(280 470-xx) Constant contouring speed at corners Input: 0.0001 to 179.9999 [°]	PLC SZ123	RUN	4–78

Machine parameter	Function and input	Change via	Reaction	Page
MP7470	Maximum feed rate at 100% override Input: 0 to 300 000 [mm/min] 0 = no limit	PLC CN123	RUN	–
MP7471	Maximum feed rate of linear axes during compensation movement by positioning the angular axes with M128 Input: 0 to 300 000 [mm/min]	PLC CN123	RUN	–
MP7475	Reference in datum table Input: 0 = Reference is the workpiece datum 1 = Reference is the machine datum	PLC CN123	RUN	4–172
MP7480.0	Output of tool or pocket number with TOOL CALL block Input: 0 = No output 1 = Tool number output only when tool number changes (W264) 2 = Tool number output with every TOOL CALL block (W264) 3 = Output of pocket number (W262) and tool number (W264) only when tool number changes 4 = Output of pocket number (W262) and tool number (W264) with every TOOL CALL block 5 = Output of pocket number (W262) and tool number (W264) only when tool number changes. Pocket table does not change. 6 = Output of pocket number (W262) and tool number (W264) with every TOOL CALL block. Pocket table does not change.	PLC	RUN	4–268

Machine parameter	Function and input	Change via	Reaction	Page
MP7480.1	Output of tool or pocket number with TOOL DEF block Input: 0 = No output 1 = Tool number output only when tool number changes (W264) 2 = Tool number output with every TOOL DEF block (W264) 3 = Output of pocket number (W262) and tool number (W264) only when tool number changes 4 = Output of pocket number (W262) and tool number (W264) with every TOOL DEF block	PLC	RUN	4–268
MP7490	Functions for traverse ranges Input: %xxxx Bit 0: Traverse ranges displayed via MOD 0 = One traverse range 1 = Three traverse ranges Bit 1: Number of reference points 0 = One reference point for each individual traverse range 1 = One reference point for all traverse ranges Bit 2: Calibration data: touch probe for workpiece measurement 0 = One set of calibration data for all traverse ranges 1 = One set of calibration data for each individual traverse range Bit 3: Calibration data: touch probe for workpiece measurement 0 = One set of calibration data for all traverse ranges 1 = One set of calibration data for each individual traverse range	PLC	RUN	4–10 4–207 4–228

Machine parameter	Function and input	Change via	Reaction	Page
MP7500	Tilt Working Plane function Input: %xxxxxx Bit 0: Turn on Tilt Working Plane function 0 = off 1 = on Bit 1: 0 = Angle corresponds to the position of the tilting axes of the head / table 1 = Angle corresponds to the angles in space (the TNC calculates the position of the tilt 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 = <set datum> = <machine datum> also valid for rotary tables 1 = With rotary tables, Tilt Working Plane cycle is referenced to new datum after the workpiece has been aligned Bit 4: 1 = In the default setting, the offset that results from exchanging the spindle head is compensated through PLC datum shift. During call-up of M128, M114 or the "tilt working plane" cycle, this offset is not compensated again. 0 = In the default setting this offset is not compensated until M128, M114 or the "tilt working plane" cycle is called. Bit 5: Position of Tilted axis during "datum setting" [1] 0 = Current position of tilted axis referenced to the machine datum 1 = Position of tilted axis as it is entered with the 3D ROT soft key *Bit6[1]: Cycle 19 with spatial angle C ≠ 0* *0 = Spatial angle C is achieved by rotating the coordinate system* *1 = Spatial angle C is achieved by rotating the table*	PLC		4–31

[1] As of NC software 280 476-xx

Machine parameter	Function and input	Change via	Reaction	Page
MP7510 MP7510.0-14	Transformed axis Input: %xxxxxx 0 = end of the transformation sequence Bit 5 4 3 2 1 0 Axis C B A Z Y X 1st transformation to 15th transformation	PLC	RUN	4–31
MP7520 MP7520.0-14	Additional code for transformation Input: %xx Bit 0: Swivel axis 0 = Swivel head 1 = Tilting table Bit 1: Type of dimension in MP7530 0 = Incremental dimension (for swivel head) 1 = Absolute dimension referenced to the machine datum (for tilting table) 1st transformation to 15th transformation	PLC	RUN	4–31
MP7530 MP7530.0-14	Type of dimension for transformation Input: –99 999.9999 to +99 999.9999 0 = free swivel axis 1st transformation to 15th transformation	PLC	RUN	4–31
MP7550 MP7550.0 MP7550.1 MP7550.2	Default setting of tilting head Input: –99 999.9999 to +99 999.9999 A axis B axis C axis	PLC	RUN	–

9.3.17 Hardware

Machine parameter	Function and input	Change via	Reaction	Page
MP7600.0 MP7600.1	Position controller cycle time = MP7600.0 * 0.6 ms Input: 5 to 10 PLC cycle time = MP7600.1 * Position controller cycle time Input: 1 to 20			–
MP7620	Feed-rate override and spindle override Input: %xxxx Bit 0: Feed rate override if the rapid traverse key is pressed in the "program run" operating mode 0 = Override not in effect 1 = Override in effect Bit 2: Feed rate override if the rapid traverse key and machine direction button are pressed in the "manual" op. mode 0 = Override not in effect 1 = Override in effect Bit 3: Feed-rate override and spindle override in 1% steps or by a nonlinear curve 0 = 1% steps 1 = Nonlinear curve	PLC	RUN	4–101; 4–127
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	PLC CN123	RUN	4–233 4–234 4–236 4–235
MP7641	Entry of interpolation factor Input: 0 = Via TNC operating panel 1 = Via PLC Module 9036	PLC CN123	RUN	4–233
MP7645.0-7	Initializing parameters for handwheel Input: 0 to 255	PLC CN123	RUN	4–235; 4–236
MP7650 MP7650	Counting direction for handwheel Input: 0 = Negative counting direction 1 = Positive counting direction Count direction for handwheel axis-specific Input: %xxxxxxxxx Bit 0 to 8 0= negative counting direction Axis 1 to 9 1= positive counting direction	PLC	RUN	4–233
MP7660	Threshold sensitivity for elec. handwheel Input: 0 to 65 535 [increments]	PLC	RUN	4–233

Machine parameter	Function and input	Change via	Reaction	Page
MP7670 MP7670.0 MP7670.1 MP7670.2	Interpolation factor for handwheel Input: 0 to 10 Interpolation factor for low speed Interpolation factor for medium speed (only HR 410) Interpolation factor for high speed (only HR 410)	PLC	RUN	4–233; 4–235
MP7671 MP7671.0 MP7671.1 MP7671.2	Manual feed rate in handwheel mode with HR 410 Input: 0 to 1000 [% of MP1020] Low speed Medium speed High speed	PLC	RUN	4–235
MP7680	Machine parameter with multiple functions Input: %xxxxxxxxxxx Bit 0: Memory function for axis direction buttons 0 = Not stored 1 = Stored 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 with STOP or M06 0 = Interruption 1= No interruption Bit 4: Allowance for programmed dwell time during block scan 0 = Allow for dwell time 1 = Do not allow for dwell time Bit 5: Start of block scan 0 = Start from block with cursor 1 = Start from beginning of program Bit 6: Tool length for blocks with surface-normal vectors 0 = Without DR2 from the tool table 1 = With DR2 from the tool table Bit 10: Cutter-compensated outside corners 0 = Inserting a circular arc 1 = Inserting a spline Suggested input = 1	PLC	RUN	4–201 4–168 4–253 4-77

Machine parameter	Function and input	Change via	Reaction	Page
MP7680	*NC software 280 470-xx* *Bit 7 Insertion of the radius or spline defined in M112* *0 = Always insert* *1 = Insert only if the acceleration from MP1060.x or MP1070 would be exceeded* *Bit 8 Insertion of a circular arc or cubic spline* *0 = Insert a circular arc with M112* *1 = Insert a cubic spline with M112* *Bit 9 M112: Constant jerk on spline (bit 8 = 1)* *0 = Jerk not constant* *1 = Jerk constant*			4–78
MP7681	M/S/T/Q transfer to the PLC during block scan Input: %xxxx Bit 0: 0 = Transfer M functions during block scan 1 = Collect M functions and transfer them after a block scan Bit 1: 0 = Transfer T code during a block scan 1 = Transfer last T code after a block scan Bit 2: 0 = Transfer S or G code during a block scan 1 = Transfer last S or G code after a block scan Bit 3: 0 = Transfer FN19 outputs during a block scan 1 = Transfer last FN19 outputs after a block scan	PLC		4–170

Machine parameter	Function and input	Change via	Reaction	Page
MP7682	Machine parameter with multiple functions Input: %xxx	PLC	RUN	
	Bit 0: Calculating tool length 0 = Tool length is calculated in an I block that follows a TOOL CALL 1 = Tool length is not calculated in an I block that follows a TOOL CALL			–
	Bit 1: Reference value for calculating the presets in "datum setting" 0 = Actual value is used 1 = Nominal value is used			4–29
	Bit 2: Traverse path of rotary axes with modulo display 0 = Positioning without crossing over zero 1 = Shorter path positioning			4–124
MP7683	Executing pallet tables Input: %xxx Bit 0: Operating mode "Program run/Single block 0 = One line of the NC program is run at every start. The pallet change macro is completely executed. 1 = A complete NC program is run at every start. Bit 1: Operating mode "Program run/Full sequence 0 = A complete NC program is run at every start 1 =All of the NC programs up to the next pallet are run at every start Bit 2: Operating mode "Program run/Full sequence 0 = As defined in Bit 1 1 = All NC programs and pallets to the end of the table are run Bit 3: When the end of the pallet file is reached the program begins again with the first line. 0 = Function not effective 1 = Function effective (bit 2 must also be set) *Bit 4:* *0 = Current pallet table cannot be edited.* *1 = In the "Program run, full sequence" and "Program run, single block" modes the current pallet table can be edited.*	PLC	RUN	4–175

[1] as of NC software 280 474-xx

Machine parameter	Function and input	Change via	Reaction	Page
MP7690	Memory test at switch-on Input: %xxx Bit 0: Test RAM Bit 1: Test EPROM Bit 2: Test hard disk 0 = Memory test on switch-on 1 = No memory test on switch-on			4–190

9.3.18 Second Spindle

Machine parameter	Function and input	Change via	Reaction	Page
MP13010 to MP13520	Machine parameter block for the second spindle; identical with MP3010 to MP3520 Input: Same as MP3010 to MP3520			–

10 List of Markers and Words

10.1 List of Markers

Marker		S	R	Page
M1900 to M1999	Decoded M function if M4571 is set	NC	NC	4–157
	Spindle			
M4000	Spindle in Position	NC	NC	4–106
M4001	Nominal speed command signal of the spindle not in the ramp	NC	NC	4–97
M4002	Nominal speed command signal of the spindle = zero	NC	NC	4–97
M4003	Nominal speed output analog or digital	NC	NC	4–96
M4004	Illegal rotational speed	NC	NC	4–98
M4005	Status display and nominal speed value output for M03	PLC	PLC	4–97; 4–128
M4006	Status display and nominal speed value output for M04	PLC	PLC	4–97; 4–128
M4007	Status display for M05 and spindle stop	PLC	PLC	4–97; 4–128
M4008	Disable speed output for spindle	PLC	PLC	4–97; 4–128
M4009	Spindle rotation counterclockwise (for gear change)	PLC	PLC	4–99
M4010	Spindle rotation clockwise (for gear change)	PLC	PLC	4–99
M4011	Activate rotational speed MP3520.0 and direction of rotation from M4013	PLC	PLC	4–107
M4012	Open the spindle control loop	PLC	PLC	4–106
M4013	Direction for spindle orientation 0 = M03 1 = M04	PLC	PLC	4–107
M4014	Reverse the direction of spindle rotation	PLC	PLC	4–97
M4015	Renewed evaluation of the spindle reference mark	PLC	NC	4–106
M4016	Cycle 13 is executed	NC	PLC	4–107
M4017	Servo-controlled spindle in motion	NC	NC	4–106
M4018	Reference mark for spindle not yet traversed	NC	NC	4–106
M4019	Reverse the count direction of the spindle position encoder	PLC	PLC	4–97
	Thread Cutting			
M4030	Cycle 2 or Cycle 17 active	NC	NC	4–110; 4–113
M4031	Cycle 17 or Cycle 18 active	NC	NC	4–113

Marker		S	R	Page
	Coolant Status			
M4040	Status display M07, M08, M09 highlighted	PLC	PLC	4–128
M4041	Status display M07, M08, M09 and MK	PLC	PLC	4–128
M4042	Status display M07, M08, M09 and MK	PLC	PLC	4–128
	Touch Probe			
M4050	Touch probe not ready (ready signal is missing)	NC	NC	4–207
M4051	Stylus deflected before start of probing cycle	NC	NC	4–207
M4052	Stylus deflected (probing process not executed)	NC	PLC	4–207
M4053	Probing sequence ended or interrupted	NC	NC	4–207
M4054	Battery voltage too low (battery warning at touch probe connection); evaluated only during the probing process	NC	NC	4–207
M4055	Enabling the probing process	NC	PLC	4–207
M4056	NC stop in all operating modes if stylus is deflected	PLC	PLC	4–207
M4060	Cycle for tool measurement started	NC	NC	4–229
M4061	0 = tool measurement 1 = tool inspection	NC	NC PLC	4–229
M4062	0 = wear tolerance not exceeded 1 = wear tolerance exceeded	NC	NC PLC	4–229
M4063	0 = breakage tolerance not exceeded 1 = breakage tolerance exceeded	NC	NC	4–229
M4065	All dimensions of the workpiece are OK	NC	PLC	–
M4066	The workpiece needs rework	NC	PLC	–
M4067	The workpiece is scrap	NC	PLC	–
	Strobe Signals from the NC to the PLC			
M4070	Strobe signal for gear code	NC	NC	4–99
M4071	Strobe signal for S code	NC	NC	4–102
M4072	Strobe signal for M function	NC	NC	4–157
M4073	Strobe signal T code (P code) with TOOL CALL	NC	NC	4–269
M4074	Strobe signal T code (P code) with TOOL DEF	NC	NC	4–269
M4075	Transfer with FN19 active	NC	NC	5–12
	Acknowledgment of NC Strobe Signals			
M4090	Acknowledgment "gear change completed"	PLC	PLC	4–99
M4091	Acknowledgment of S code	PLC	PLC	4–102
M4092	Acknowledgment of M code	PLC	PLC	4–157
M4093	Acknowledgment of T code (P code) with TOOL CALL	PLC	PLC	4–269
M4094	Acknowledgment of T code (P code) with TOOL DEF	PLC	PLC	4–269
M4095	Acknowledgment of transfer with FN19	PLC	PLC	5–12

Marker		S	R	Page
	Strobe Signals from the PLC to the NC			
M4120	PLC positioning axis 1 active	PLC	NC; PLC	4–28
M4121	PLC positioning axis 2 active	PLC	NC; PLC	4–28
M4122	PLC positioning axis 3 active	PLC	NC; PLC	4–28
M4123	PLC positioning axis 4 active	PLC	NC; PLC	4–28
M4124	PLC positioning axis 5 active	PLC	NC; PLC	4–28
M4125	PLC positioning axis 6 active	PLC	NC; PLC	4–28
M4126	PLC positioning axis 7 active	PLC	NC; PLC	4–28
M4127	PLC positioning axis 8 active	PLC	NC; PLC	4–28
M4128	PLC positioning axis 5 active	PLC	NC; PLC	4–28
M4130	Activation of PLC positioning for spindle orientation	PLC	NC	4–107
M4131	Activation of transfer of the value from D528 to the Q parameter defined in W516	PLC	NC	5–24
M4132	Activate datum shift from D528 to D544	PLC	NC	4–244
M4133	Starting and stopping the free-rotation function	PLC	NC	4–126
M4134	Activation of a gear range and speed through the PLC	PLC	NC	4–99
M4135	Activation of the selected traverse range (M2816/M2817)	PLC	NC	4–10
	NC Operating Modes and Status			
M4150	Operating mode: Manual operation	NC	NC	–
M4151	Operating mode: Electronic handwheel	NC	NC	–
M4152	Operating mode: Positioning with manual data input	NC	NC	–
M4153	Operating mode: Program run, single block	NC	NC	–
M4154	Operating mode: Program run, full sequence	NC	NC	–
M4155	Operating mode: Traversing the reference marks	NC	NC	–
M4156	MANUAL OPERATION soft key was pressed	NC	NC	4–168
M4157	Returning-to-contour function (RESTORE POSITION) active	NC	NC	4–168
M4158	Block scan is active	NC	NC	4–168
M4159	PLC editor: Press the END button or END soft key	NC	NC, PLC	4–181
M4160	Pallet table is selected	NC	NC	4–175
M4161	M/S/T/Q transfer after mid-program startup	NC	NC	4–168
M4170	END PGM, M02 or M30 was executed	NC	NC	4–175

Marker		S	R	Page
M4172	First PLC scan after power on	NC	NC	–
M4173	First PLC scan after interruption of the PLC program	NC	NC	–
M4174	First PLC scan after editing the MPs (MP edit was exited and the MPs were altered)	NC	NC	–
M4175	Program interruption (control-in-operation symbol flashes)	NC	NC	4–129
M4176	Control is in operation (control-in-operation symbol is on or is blinking)	NC	NC	4–129
M4177	Erasable error message is displayed	NC	NC	4–89
M4178	Error message EMERGENCY STOP is displayed	NC	NC	4–89
M4180	Rapid traverse programmed (FMAX)	NC	NC	4–127
M4181	*NC program selected*[1]	*NC*	*PLC*	–
	Arithmetic or Module Error in the PLC			
M4200	Overflow during multiplication	NC	PLC	5–107; 5–153
M4201	Division by 0	NC	PLC	5–108; 5–153
M4202	MODULO incorrectly executed	NC	PLC	5–109; 5–153
M4203	Is set or reset during module calls	NC	NC PLC	5–153; 5–176
M4204	Reserved for errors that the PLC programmer would like to intercept	PLC	PLC	5–153
	Markers That Can Be Changed by Machine Parameter			
M4300 to M4347	Markers that can be changed by MP4310.0, MP4310.1 and MP 4310.2	NC	NC	–
M4348 to M4411	Values from MP4310.3 to MP4310.6			–
	Tool Change			
M4520	Another T code (P code) follows with TOOL CALL	NC	NC	4–270
M4521	Tool no. zero programmed	NC	NC	4–269
M4522	Tool programmed with pocket number	NC	NC	4–270
M4523	Tool programmed without pocket number	NC	NC	4–270
M4524	Call special tool (TOOL CALL)	NC	NC	4–270
M4525	TOOL CALL programmed or after expiration of tool life	NC	NC	4–270

[1] *as of NC software 280 476-xx*

Marker		S	R	Page
M4526	Axis 1 is tool axis	NC	NC	4–8
M4527	Axis 2 is tool axis	NC	NC	4–8
M4528	Axis 3 is tool axis	NC	NC	4–8
M4529	Axis 4 is tool axis	NC	NC	4–8
M4530	Axis 5 is tool axis	NC	NC	4–8
M4531	Axis 6 is tool axis	NC	NC	4–8
M4532	Axis 7 is tool axis	NC	NC	4–8
M4533	Axis 8 is tool axis	NC	NC	4–8
M4534	Axis 9 is tool axis	NC	NC	4–8
M4538	Geometry of the tool from W264	PLC	NC	4–269
M4539	T highlighted in status display	PLC	PLC	–
M4540	Sequence for tool change from special tool to normal tool	PLC	PLC	4–270
M4541	Special tool to original pocket despite variable pocket coding	PLC	PLC	4–252; 4–270
M4542	Do not update pocket number in pocket table	PLC	PLC	4–168; 4–270
M4543	Tool life expired (TIME1 in the tool table)	NC	NC; PLC	4–253
	Additional Keys			
M4560	NC stop ("0" corresponds to stop)	PLC	PLC	4–201
M4561	Rapid traverse	PLC	PLC	4–201
M4562	Memory function for axis direction keys	PLC	PLC	4–201
M4563	Feed rate release for all axes	PLC	PLC	4–63
M4564	NC start (edge evaluation)	PLC	PLC	4–201
	General Functions			
M4570	Unit of measure for transfer with FN19 0 = mm; 1 = inch	NC	NC	5–11
M4571	Enabling of decoded M-code transfer in markers M1900 to M1999	PLC	PLC	4–157
M4572	Enabling of incremental jog positioning	PLC	PLC	4–242
M4574	Selecting the traverse range	PLC	PLC	4–8; 4–10
M4575	Selecting the traverse range	PLC	PLC	4–8; 4–10
M4576	Suppression of handwheel pulses	PLC	PLC	4–233
M4577	Disabled key was pressed	NC	PLC	4–196
M4579	INCREMENT ON/OFF soft key to on	NC	NC	4–242

Marker		S	R	Page
M4580	Suppress EMERGENCY STOP, open all position control loops, NC stop	PLC	PLC	4–64; 4–89
M4581	Open all position control loops, NC stop	PLC	PLC	4–62
M4590	Triggering signal of the PLC input defined in MP4130.2	NC	PLC	5–29
M4591	Triggering signal of the PLC input defined in MP4130.3	NC	PLC	5–29
M4592	Triggering signal of the PLC input defined in MP4130.4	NC	PLC	5–29
M4593	Triggering signal of the PLC input defined in MP4130.5	NC	PLC	5–29
M4800 to M4899	Error messages to which markers in the PET tables are assigned	PLC	NC; PLC	4–159

10.2 List of Words

Word	Function	Page
W256	G code for S analog	4–99
W258	S code	4–102
W260	Code for M function	4–157
W262	Tool pocket number	4–269
W264	Tool number	4–269
W266	Index number of a programmed indexed tool	–
W270	Help-file line number –1 = no help file selected –2 = no valid numerical value 0 to 9999 = line number	4–152
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 = Traversing the reference point	4–44
W274	Code of the activated key	4–196
D276	Code of the code number last entered via MOD	4–188
D280	First numerical value from FN19	5–11
D284	Second numerical value from FN19	5–11
W302	PLC soft-key number of the activated PLC soft key	4–151
W320	Nominal rotational speed	4–96
W322	Actual rotational speed	4–96
D356	Programmed rotational speed [0.001 rpm]	4–96, 4–99
D360	Programmed feed rate [mm/min] (NC → PLC)	4–60
D364	Nominal speed	–
D368	Actual speed	–
D388	Current feed rate [mm/min]	4–60
W480	Analog input 1 on X48 [0.1 V]	–
W482	Analog input 2 on X48 [0.1 V]	–
W484	Analog input 3 on X48 [0.1 V]	–
W486	Temperature input 1 on X48 [0.5°]	–
W488	Temperature input 2 on X48 [0.5°]	–
W490	Temperature input 3 on X48 [0.5°]	–
W492	Percentage factor for spindle override (NC → PLC)	4–101
W494	Percentage factor for feed-rate override (NC → PLC)	4–127
W516	Number of the Q parameter to be overwritten (Q100 to Q107 = 0 to 7)	5–24

[1] as of NC software 280 474-xx

Word	Function	Page
B518	0 = cancel the free-rotation function 8 = free-rotation function for axis 4 16 = free-rotation function for axis 5	4–126
B519	Definitions of the traversing direction	4–126
B520	Axis-specific feed rate enable Bit 0 +1 Feed-rate enable in axis 1 +0 No feed-rate enable in axis 1 Bit 1 +2 Feed-rate enable in axis 2 +0 No feed-rate enable in axis 2 Bit 2 +4 Feed-rate enable in axis 3 +0 No feed-rate enable in axis 3 Bit 3 +8 Feed-rate enable in axis 4 +0 No feed-rate enable in axis 4 Bit 4 +16 Feed-rate enable in axis 5 +0 No feed-rate enable in axis 5 Bit 5 +32 Feed-rate enable in axis 6 +0 No feed-rate enable in axis 6 Bit 6 +64 Feed-rate enable in axis 7 +0 No feed-rate enable in axis 7 Bit 7 +128 Feed-rate enable in axis 8 +0 No feed-rate enable in axis 8 Bit 8 +256 Feed-rate enable in axis 9 +0 No feed-rate enable in axis 9	–
W522	Activate the fast PLC inputs Bit 0 Fast PLC input defined in MP4130.0 Bit 1 Without function 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	 4–79 5–29 5–29 5–29 5–29

D528	Value to be transferred to the Q parameters
	Datum shift for axis 1
	Position of axis 1 [1/10 000 mm]
		4–28;
		4–244;
		5–24

D532	Datum shift for axis 2 Position of axis 2	4–244
D536	Datum shift for axis 3 Position of axis 3	4–244
D540	Datum shift for axis 4 Position of axis 4	4–244
D544	Datum shift for axis 5 Position of axis 5	4–244

Word	Function	Page
W560	Feed rate in axis 1 [mm/min]	4–28
W562	Feed rate in axis 2	4–28
W564	Feed rate in axis 3	4–28
W566	Feed rate in axis 4	4–126
W568	Feed rate in axis 5	4–126
	Lag-tracking error compensation (compensation speed from MP4070) Input range: + 32 767 to – 32 768 [1/10 µm]	4–20
W576	Axis 1	
W578	Axis 2	
W580	Axis 3	
W582	Axis 4	
W584	Axis 5	
D592	Nominal position for oriented spindle stop	4–107
D596	Max. feed rate from PLC [mm/min]	4–60
D604	Maximum possible spindle speed	–
W754	Percentage of feed-rate override for free rotation (0 to 300%)	4–126
D756	Preset speed from the PLC; programmed speed	4–99
D760	Offset for tilting axes (touch probe center misalignment) [1/10 000°]	4–206
W764	% factor for spindle override (PLC → NC)	4–101
W766	% factor for feed-rate override (PLC → NC)	4–127
D768 to D956	Input values from MP4210.0 to MP4210.47	5–15
W960 to W968	Input values from MP4220.0 to MP4220.4	5–16
W976 to W988	Input values from MP4310.0 to MP4310.6	5–16
W1008	S code for minimum speed	4–102
W1010	Rotational speed increment for S code	–
W1022	Error status of the last called module	–
W1024	Axis releases, bit-coded (bits 0 to 8 = axis 1 to 9)	4–62
W1026	Axes in position, bit-coded (bits 0 to 8 = axis 1 to 9)	4–82
W1028	Axes in motion, bit-coded (bits 0 to 8 = axis 1 to 9)	4–83
W1030	Traverse direction, bit-coded (bits 0 to 8 = axis 1 to 9) 0 = positive; 1 = negative	4–2
W1032	Reference marks not yet traversed, bit-coded (bits 0 to 8 = axis 1 to 9)	4–44
W1034	Positive limit switch was traversed, bit-coded (bits 0 to 8 = axis 1 to 9)	4–10
W1036	Negative limit switch was traversed, bit-coded (bits 0 to 8 = axis 1 to 9)	4–10
W1038	Preparing to open the position loop, bit-coded (bits 0 to 8 = axis 1 to 9)	4–62

[1] as of NC software 280 474-xx

Word	Function	Page
W1040	Opening the control loop, bit-coded (bits 0 to 8 = axis 1 to 9)	4–62
W1042	No monitoring, bit-coded (bits 0 to 8 = axis 1 to 9)	4–79
W1044	Actual position capture, bit-coded (bits 0 to 8 = axis 1 to 9)	4–63
W1046	Manual traversing with + direction button, bit-coded (bits 0 to 8 = axis 1 to 9)	4–201
W1048	Manual traversing with – direction button, bit-coded (bits 0 to 8 = axis 1 to 9)	4–201
W1050	Incremental jog positioning +, bit-coded (bits 0 to 8 = axis 1 to 9)	4–242
W1052	Incremental jog positioning –, bit-coded (bits 0 to 8 = axis 1 to 9)	4–242
W1054	Reference end position, bit-coded (bits 0 to 8 = axis 1 to 9)	4–44
W1056	Lubrication pulse. Value from MP4060.x exceeded, bit-coded (bits 0 to 8 = axis 1 to 9)	4–10
W1058	Reset the accumulated distance for lubrication, bit-coded (bits 0 to 8 = axis 1 to 9)	4–10
W1060	Enable axis-specific feed rate (bits 0 to 8 = axis 1 to 9)	4–63
W1062	Disable axis-specific handwheel pulses	–

11 List of Modules

Module	Function	Page
9000	Copy marker block	5–176
9001	Copy word block	5–176
9002	Read inputs of a PL	4–238
9003	Read analog input of a PL	4–240
9004	Read edges of PLC inputs	4–239
9005	Update outputs of a PL 410	4–239
9006	Starting the PLC timer	5–27
9008	Read certain input of a PL 410	–
9009	Set certain outputs of a PL 410	–
9010	Read indexed byte	5–177
9011	Read indexed word	5–177
9012	Read indexed doubleword	5–177
9019	Check program stack	5–34
9020	Write indexed byte	5–178
9021	Write indexed word	5–178
9022	Write indexed doubleword	5–178
9031	Write machine parameter	9–4
9032	Read machine parameters from MP list	9–5
9033	Select a machine parameter file	9–5
9034	Load an incomplete machine parameter file	9–6
9035	Read status information	4–19, 4–132
9036	Write status information	4–234, 4–242
9038	Read axis information	4–7
9040	Read coordinate values (format 1/1000 mm)	4–125
9041	Read coordinate values (format 1/10 000 mm)	4–125
9042	Read spindle coordinates	4–95
9050	Number conversion binary ASCII	5–179
9051	Number conversion binary-ASCII formatted	5–180
9052	Number conversion ASCII-binary	5–181
9053	Conversion binary-ASCII/hexadecimal	5–181
9054	Conversion ASCII/Hexadecimal-binary	5–182
9055	Conversion binary time =>ASCII	4–194
9070	Copy a number from a string	5–152
9071	Determine string length	5–152
9080	Delete PLC window	4–133
9081	Interrogate status of PLC window	4–134

Module	Function	Page
9082	Display string	4–135
9083	Display bar chart	4–136
9085	Display PLC error message	4–160
9086	Delete PLC error message	4–160
9087	Interrogate status of PLC error message	4–161
9088	Show M functions	–
9089	Display control-in-operation symbol (∗)	4–128
9090	Select a line in the pallet table	4–175
9091	Find line number of a tool from the tool table	–
9092	Search in .T/.D/.TCH tables	4–250
9093	Read from .T/.D/.TCH tables	4–250
9094	Write in .T/.D/.TCH tables	4–251
9095	Select non-linear compensation by PLC	4–19
9096	Delete a line in the tool table	4–251
9100	Assign RS-232-C interface	6–32
9101	Release RS-232-C interface	6–32
9102	Read interface status	6–33
9103	Send string via RS-232-C	6–33
9104	Receive string from RS-232-C	6–34
9105	Transmit binary data via RS-232-C	6–34
9106	Receive binary data from RS-232-C	6–34
9107	Read binary data from reception buffer	6–35
9110	Send a message via LSV2	6–35
9111	Receive a message via LSV2	6–36
9112	Send ASCII characters via RS-232-C	6–36
9113	Receive ASCII characters via RS-232-C	6–37
9120	Start a PLC axis	4–12
9121	Stop a PLC axis	4–12
9122	Interrogate status of PLC axis	4–12
9123	Traverse the reference point of a PLC axis	4–13
9124	Set override value for PLC axis	4–13
9130	Output analog voltages	4–241
9135	Switch-on or retrigger infrared probe	–
9145	Automatic actual and nominal value transfer for certain axes	–
9147	Assigns a new REF value to the current position of an axis	–
9151	Select traverse range and axis designation	4–9
9152[1]	*Select traverse range, axis display and axis designation*	–

[1] *as of NC software 280 476-xx*

Module	**Function**	**Page**
9153	Define probe axis for manual measurement	–
9155	Switch axis from controlled to noncontrolled condition	–
9156	Switch axis from noncontrolled to controlled condition	–
9158[1]	*Maximum torque of an axis*	–
9159	Interrogation of the drives to be switched off in 200 ms	–
9160	Read error status of the integrated current controller	4–86
9161	Enable integral drive control loops	4–71
9162	Read status of motor controller	4–87
9163	Wye/delta connection	4–115
9164	Read actual rotation speed of drive motors	4–68
9165	Read temperature of drive motors	4–84
9166	Read actual utilization of drive motors	4–88
9167	Enable/disable power supply monitoring	4–84
9168	Status of the current controller	4–311
9169	Definition of the axes for which I32 does not cause switch off of the drives	–
9171	Oriented spindle stop	4–106
9175	Activate spindle	–
9180	Key simulation	4–196
9181	Disable individual NC keys	4–197
9182	Re-enable NC keys	4–197
9183	Disable groups of NC keys	4–197
9184	Re-enable groups of NC keys	4–198
9186	Call soft-key function	4–198
9187	Status of a soft-key function call	4–198
9190	Start run-time counter	4–192
9191	Stop run-time counter	4–192
9192	Output of run-time counter	4–192
9193	Set run-time counter	4–193
9194	Alarm in case of excessive run-time	4–193
9195	Output of real-time clock	4–193
9200	Create/delete PLC soft-key row	4–150
9201	Create/delete PLC soft key	4–151
9202	Select/deselect PLC soft keys and PLC windows	4–151
9210	Open/delete mask file for PLC window	4–145
9211	PLC window status	4–146
9215	Superimpose PLC window	4–155

[1] *as of NC software 280 476-xx*

Module	Function	Page
9220	Cross over reference points	4–43
9221	PLC positioning	4–27
9222	Status for PLC positioning	4–27
9223	Free rotation	4–126
9225	Compensation value for zero pulse	–
9230	Datum shift	4–244
9231	Compensation of thermal expansion	4–20
9240	Open a file	4–184
9241	Close a file	4–184
9242	Position within a file	4–185
9243	Read line-by-line out of a file	4–185
9244	Write line-by-line to a file	4–186
9245	Read a field out of a table (string)	4–178
9246	Write a field in a table (string)	4–179
9247	Search for a condition in a table	4–180
9250	Start PLC editor for tables	4–181
9251	End PLC editor for tables	4–182
9252	Position cursor in PLC editor	4–182
9255	Read a field out of a table (whole number)	4–178
9256	Write a field in a table (whole number)	4–179
9260	Receive and wait for events	5–158
9261	Send events	5–159
9262	Context change between SPAWN processes	5–159
9263	Interrupt SPAWN process for defined period of time	5–160
9270	Read in OEM.SYS	5–32
9271	Write to OEM.SYS	5–32
9275[1]	*Write ASCII data in the log book*	–
9276[1]	*Write operand contents in the log book*	–
9280	Start NC macro (run pallet entry)	–
9281[1]	*Place the cursor on any line of the selected pallet table*	–
9290	Select NC program	–

[1] as of NC software 280 476-xx

12 Subject Index

1 Update Information No. 12

The following NC software versions have been released:

NC software **280 470-12** and **280 471-12** February 99

NC software **280 472-12** and **280 473-12** December 98

NC software **280 474-11** and **280 475-11** December 98
NC software **280 474-12** and **280 475-12** March 99

NC software **280 476-01** and **280 477-01**

1.1 Error in positioning after mid-program startup

The tool change can be executed via an OEM-specific NC macro. The following PLC functions may not be activated in this NC macro:

- Datum shift
- Traverse range switching
- Spindle switching (module 9175)

If one of these functions is programmed in the NC macro, and this NC macro is started at the end of the mid-program startup, (NCMACRO=TC in MGROUPS.SYS), a positioning error occurs after "mid-program startup" and "restore position".

The above functions must also be transferred to the PLC and executed via the PLC sequential program during the mid-program startup (do not list the functions in MGROUPS.SYS, but enter REMAIN=OUTPUT).

To ensure that positioning errors do not occur on your machine, please check your PLC program for the functioning described above.

HEIDENHAIN will make the following changes with the next software release:

- The error message "PLC function not permitted" will be generated if a datum shift or a traverse range switch is executed by the PLC during the function "Restore machine status".
- If the spindle is switched during "Restore machine status", module 9175 will acknowledge with an error code.

1.2 NC software 280 470-xx

NC software	Release		
280 470-12	02/99	Export version:	280 473-12

PLC module

- Module 9175 (Activate spindle) was expanded by error code 6:
  W1022 = 6: Marker M4157 = 1 (RESTORE POSITION active)

Other improvements

- With the machining cycles 212 (POCKET FINISHING) and 214 (CIRCULAR POCKET FINISHING), the contour is always approached at the machining feed.
- With the machining cycles 210 (SLOT) and 211 (CIRCULAR SLOT), the starting point for the finishing cycle is approached at the machining feed.
- The default values for Q208 in the machining cycles 201 (REAMING), 202 (BORING) and 203 (UNIVERSAL DRILLING) were increased from 500 mm/min or 20 inch/min to 30000 mm/min or 1200 inch /min.

1.3 NC software 280 472-xx

NC software	Release		
280 472-12	12/98	Export version:	280 473-12

Other improvements

- With the touch-probe cycles 421 and 422, the probing direction depends on whether a basic rotation or a rotation via Cycle 10 is active.
- The default values for Q208 in the machining cycles 201 (REAMING), 202 (BORING) and 203 (UNIVERSAL DRILLING) were increased from 500 mm/min or 20 inch/min to 30000 mm/min or 1200 inch /min.
- With the machining cycles 210 (SLOT) and 211 (CIRCULAR SLOT), the starting point for the finishing cycle is approached at the machining feed.
- With the machining cycles 212 (POCKET FINISHING) and 214 (CIRCULAR POCKET FINISHING), the contour is always approached at the machining feed.

1.4 NC software 280 474-xx

NC software	**Setup disks**	**Release**
280 474-11	286 195-11	12/98
Export version:		
280 475-11	286 195-11	12/98

Machine parameter

- MP1090 was expanded:
  The input range was expanded from 1 to 1000 $[m/s^3]$ to 0.1 to 1000.0 $[m/s^3]$

Other improvements

- With the touch-probe cycles 421 and 422, the probing direction depends on whether a basic rotation or a rotation via Cycle 10 is active.

NC software	**Setup disks**	**Release**
280 474-11	286 195-12	02/99
Export version:		
280 475-11	286 195-12	02/99

NC software	**Setup disks**	**Release**
280 474-12	286 195-13	03/99
Export version:		
280 475-12	286 195-13	03/99

Machine parameter

- MP2221 is new:
  Monitoring the reference pulse of the spindle motor encoder was introduced. If you get a DSP error message (in particular in connection with gear encoders), you can switch off this monitoring with MP2221.
  Input: 0 = monitoring active
  1 = monitoring not active

Machine adjusting

- M4181 is new:

	Set	Reset
NC program selected	NC	PLC

- The max. input value for the rated frequency F-N in the motor table for asynchronous motors was increased from 999.9 Hz to 2000.0 Hz.

PLC modules

- Module 9175 (Activate spindle) was expanded by error code 6:
  W1022 = 6: Marker M4157 = 1 (RESTORE POSITION active)
- Module 9281 is new:
  With module 9281, the cursor can be placed on a particular line in the pallet table selected in one of the operating modes Single Block or Full Sequence. Programmed parameters such as datum shift or datum set can be executed immediately.

  Call:
  ```
  PS          B/W/D/K       <line number in the pallet table>
  PS          B/W/D/K       <mode>
                            0 = datum shift / do not execute datum set
                            1 = datum shift / execute datum set
  CM          9281
  PL          B/W/D         <Error code>
                            0: Cursor was placed in the line
                            1: Module was not called in a SPAWN or submit job
                            2: Module was called when NC program already started
                            3: No pallet table selected as main program
                            4: Line does not exist
                            5: Error during datum set, or error in datum or pallet table
  Error code:               M4203 = 0:   Cursor was placed in the line
                            M4203 = 1:   Error according to error code
  ```

Other improvements

- The new error number 1068 (datum table?) was introduced for the function FN14: ERROR.
- The maximum input value for Q239 in cycle 207 (RIGID TAPPING NEW) and for the pitch in cycles 17 (RIGID TAPPING) and 18 (THREAD CUTTING) was increased to 99.9999 mm.

1.5 Advance information for NC software 280 476-xx

NC software	**Setup disks**
280 476-01	286 197-01
Export version:	
280 477-01	286 197-01

Machine parameters

- MP1097 and MP1098 were expanded:
  Input range was expanded from 1 to 1000 [m/s^3] to 0.1 to 1000.0 [m/s^3]
- MP1150 was expanded:
  MP1150 was changed to MP1150.0 with the same meaning, MP1150.1 and MP1150.2 were added new
  MP1150.0 Delay time for canceling the nominal speed value with the blinking error message Excessive servo lag <axis>
  Input: 0 to 65.535 [s]
  Recommended input value: 0
  MP1150.1 Time during which the monitoring functions should remain switched off after the fast PLC input defined in MP4130.0 is set. Once this time has elapsed, the monitoring functions once again become active.
  Input: 0 to 65.535 [s]
  Recommended input range: 0.2 s to 0.5 s
  MP1150.2 Minimum time for which the monitoring functions should remain active after the time set in MP1150.1 has elapsed (e.g. if the input changes quickly).
  Input: 0 to 65.535 [s]
- MP2221 is new:
  Monitoring of the reference pulse of the spindle motor encoder was introduced. If you get a DSP error message (in particular in connection with gear encoders), you can switch off this monitoring with MP2221.
  Input: 0 = monitoring active
  1 = monitoring not active
- MP2360, MP2361 and MP2391 are new:
  In the event of an emergency stop, the spindle must be braked to a stop as quickly as possible. Increasing the recovered power shortens the braking process. However, if the recovered power is too high, the braking energy is not dissipated quickly enough: the inverter may under certain circumstances switch off and the spindle will coast to a stop.

  For spindles without a dedicated spindle DSP, you can enter a time constant in MP2360.x or MP2361 by which the recovered power is reduced. The larger the time constant, the less the mean recovered power during braking, and the longer the braking process.
  To find the optimum value, begin with a large time constant. Then make an emergency stop to brake the spindle from maximum speed. Reduce the time constant until you find the shortest possible braking time at which the inverter does not switch off. Since the temperature of the braking resistor and the mass of the tool in the spindle affect the braking power, this time constant must be increased somewhat to provide a safety margin, or it must be found with the heaviest tool and a hot braking resistor.

MP2360.x Time constant for braking a second digital spindle during EMERGENCY STOP

Effective MP2360.x	**PWM output of the second spindle**
MP2360.0 to MP2360.8	X51 to X59

MP2361 Time constant for braking the spindle during EMERGENCY STOP
Input: 0.01 to 5.00 [s]
0 = function not active (braking is only dependent on MP2191)

For spindles with dedicated spindle DSP the braking power can be limited in MP2391.

Input value $\text{MP2391.x} = 0{,}9 \cdot \frac{U_{Zmin}^2}{R \cdot 1000}$

U_{Zmin} = minimum dc-link voltage [V]
R = braking resistance [Ω]
MP2391.0 Wye connection
MP2391.1 Delta connection
Input: 0.1 to 3000.0 [kW]
0 = braking power is not limited
(preferable for inverters with dc-link power recovery)

- MP2393 is new:
  To achieve a "wide range" performance from a spindle motor, the power of the motor must be limited. With MP2393 the power for spindles with their own spindle DSP can be limited for wye and delta connection.
  MP2393.0 Wye connection
  MP2393.1 Delta connection
  Input: 0.1 to 3000.0 [kW]
  0 = no power limit
- MP4030 is new:
  A max. 4 PL 4xx B PLC I/O units can be connected to the LE 426 and LE 430. The first PL is connected to the LE and the other PLs to the first PL. The number of the physical PL corresponds to the number of the logical PL. With MP4030.x the logical number of a PL can be assigned to a different physical PL.
  MP4030.0 to MP4030.3 First to fourth physical PL
  Input: 0 = first logical PL
  1 = second logical PL
  2 = third logical PL
  3 = fourth logical PL

- MP6585 and MP6586 are new:
  Before the start of a tool measurement cycle, MP6585 checks the position of the rotary and additional linear axes. MP6586.x indicates the reference coordinate at which the axis must be positioned for the measurement to take place.
  If monitoring is active for an axis and if the current reference coordinate does not correspond with the reference coordinate in MP6586, an error message is output.

  MP6585 Monitoring the positions of the rotary and additional linear axes during the tool measurement cycles

Bit0	A-axis	0 = axis is not monitored
Bit1	B-axis	1 = axis is monitored
Bit2	C-axis	
Bit3	U-axis	
Bit4	V-axis	
Bit5	W-axis	

  MP6586 Ref-coordinate for the monitoring of the rotary and additional linear axis positions during the tool measurement cycles

Input:	–99 999.9999 to + 99 999.9999 [mm]
MP6586.0	A-axis
MP6586.1	B-axis
MP6586.2	C-axis
MP6586.3	U-axis
MP6586.4	V-axis
MP6586.5	W-axis

- MP7245 is new:
  With MP7245 the auxiliary cycles 18 (THREAD CUTTING) and cycle 33 (TAPER THREAD BORING) can be disabled.
  Input: 0 = auxiliary cycles are inhibited
  1 = auxiliary cycles are permitted
- MP7266 was expanded:
  MP7266.28 Probe center misalignment main axis (CAL-OF1)
  MP7266.29 Probe center misalignment secondary axis (CAL-OF2)
  MP7266.30 Spindle angle during calibration (CAL-ANG)
  See also section on Machine integration
- MP7411 was expanded:
  Bit1 = 0: Only one set of probe calibration data
  = 1: More than one set of probe calibration data in the tool table
  See also section on Machine integration
- MP7500 was expanded:
  Bit6 Cycle 19 (Tilt working plane) with spatial angle C ≠ 0 (for A = 0 and B = 0) on machines with C rotary table and tool axis Z
  = 0: Spatial angle C is achieved by rotating the coordinate system.
  = 1: Spatial angle C is achieved by rotating the table (and storing the angle in Q122).
- MP7530 was expanded:
  In MP7530.x (dimensions for transformation) a formula can be entered for temperature compensation for swivel heads and tilting tables with M128 and for "Tilt working plane". The formulas may not have more than 31 characters and must contain PLC words as variables.
  As soon as M128 or the "Tilt working plane" function are activated in the Automatic modes, the dimensions of the swivel head or tilting table are recalculated.

Machine integration

- HSC filter was expanded:
  With the HSC filter up until now, only the jerk at corners that was entered in MP1098.x was limited. Now an additional permissible jerk for curve changes (e.g. tangential transitions of a linear movement in a circular path) can be entered in MP1097.x. Recommended input value: MP1097.x = MP1098.x
- If the fast PLC input defined in MP4130.0 is set, the drive is no longer switched off automatically (see also machine parameter MP1150.x).
- The position indications in the Tilt working plane function (MP7500 Bit1 = 1) have been expanded by the following tilting axis combinations:
  Universal swivel head: Axis sequence A fixed / B –90° / A variable / B +90° / A fixed / C variable (tool axis Z)
  Universal swivel head: Axis sequence B fixed / A variable / B fixed / C variable (tool axis X)
- M4181 is new:

	Set	Reset
NC program selected	NC	PLC

- The max. input value for the rated frequency F-N in the table for asynchronous motors was increased from 999.9 Hz to 2000.0 Hz
- Several sets of probe calibration data (only trigger probe) can be managed in the tool table. This function must be activated with MP7411 bit1. The tool table was expanded by the columns CAL-OF1 (probe center misalignment in main axis), CAL-OF2 (probe center misalignment in secondary axis) and CAL-ANG (spindle angle during calibration). These columns are not visible as default, can however be made visible with MP7266.28, MP7266.29 and MP7266.30. During manual measurement, the current probe calibration data can be viewed and changed in the calibration menu.
- In the Manual operating modes, the highest axis feed for all axes is stored in D388 (current machining feed rate).

PLC programming

- FN17: SYSWRITE was expanded:
  The software limit switches can be overwritten.
  ID230
  NR4
  IDX xxx = yyy

  xxx = number of axes whose limit switches are to be overwritten
  yyy = number of the first of several successive Q parameters
  1. Q parameter: negative limit switch of the 1st axis
  2. Q parameter: positive limit switch of the 1st axis
  3. Q parameter: negative limit switch of the 2nd axis
  4. Q parameter: positive limit switch of the 2nd axis
  5. Q parameter: negative limit switch of the 3rd axis
  etc.

- FN18: SYSREAD was expanded:
  The number of axes that are programmed in the selected datum table is indicated in the return Q parameter.
  ID990
  NR3
  ID xxx

  xxx = number of the first of 9 successive Q parameters for the axes X, Y, Z, A, B, C, U, V, W
  Q xxx = +1.0: axis exists
  Q xxx = –1.0: axis does not exist

PLC modules

- Module 9122 was expanded:
  With bit 6 of the status word, you can interrogate whether the PLC axis has reached the target position (0 = target position not yet reached, 1 = target position reached)
- Module 9151 was expanded:
  The axis designations defined with Module 9151 remain unchanged, even if other machine parameters are edited.
- Module 9152 is new:
  With module 9152 you can select a freely-definable axis display or designation and one of the three traverse ranges. The same conditions apply as for traverse range selection with M4135. When the module is called, M4135 is set and it is reset after the change has been made. When inputting the string for axis designation and display, you can use the same syntax as for input in MP100.x or MP7291.
  MP100.x or MP7291 cannot be activated again until the string for axis display or axis assignment has been transferred to Module 9151 or 9152 –1, a traverse range has been activated with M4135, MP100.x or MP7291 have been edited or a control reset has been carried out.

Call:		
PS	B/W/D/K/S	<string with axis display> –1 = use axis display from MP7291
PS	B/W/D/K/S	<string with axis assignment> –1 = use axis assignment from MP100.x
PS	B/W/D/K	<traverse range> 0 to 2 = traverse range –1 = do not change traverse range
CM 9152		
Error code:		M4203 = 0: traverse range, axis designation or display has been switched M4203 = 1: Error code in W1022 W1022 = 2: Value not valid for the traverse range, or string for the display configuration too long W1022 = 3: Neither a string nor –1 was transferred for the axis assignment or display W1021 = 21: The module was called when the program was already started, or without a M/S/T/T2/Q strobe

- Module 9158 is new:
  Module 9158 can limit the max. torque for an axis. After the drive is switched off, the original torque is restored again.

  ```
  Call:
  PS      B/W/D/K/S   <axis number>
  PS      B/W/D/K/S   <torque in 1/1000 Nm>
                  –1 = torque from motor data
  CM      9158
  Error code:         M4203 = 0:   Torque indicator active
                  M4203 = 1:   Error code in W1022
                  W1022 = 1:   Transfer torque value 0
                  W1022 = 2:   Invalid axis number
                  W1022 = 24:  Module was called in a SPAWN or submit job
  ```
- Module 9169 is new:
  With Module 9169 you can define axes that are not switched off by the input I32 (X42/33, drive enable, acknowledgment for control-is-ready signal) . This module works only in conjunction with the TNC 426 M / TNC 430 M controls.

  ```
  Call:
  PS      B/W/D/K     <bit-coded axes>
  CM      9169
  Error code:         M4203 = 0:no errors possible
  ```
- Module 9175 (Activate spindle) was expanded by error code 6:
  W1022 = 6: Marker M4157 = 1 (APPROACH POSITION active)
- Module 9275 is new:
  With module 9275, ASCII data can be written to the log book. Each entry can be given an identification for future processing.
  Module 9275 should not be used in the general PLC program, but rather only for debug purposes, because the processing time may be negatively influenced, unnecessary space used on the hard disk and the log book used for purposes not intended for its function (namely recording keys and error messages).

  ```
  Call:
  PS      B/W/D/K/S   <log book entry>
                      –1 = no entry
  PS      B/W/D/K/S   <log book identification>
                      –1 = no entry
  PS      B/W/D/K     <priority>
                      0 = information
                      1 = warning
                      2 = error
  CM      9275
  Error code:         M4203 = 0:   Entry was written in book
                      M4203 = 1:   Error code in W1022
                      W1022 = 1:   Invalid priority
                      W1022 = 2:   Invalid string or invalid immediate string
                      W1022 = 12:  No string end identifier
                      W1022 = 20:  Module was not called in SPAWN or submit job
  ```

- Module 9276 is new:
  With module 9276, the contents of operands (inputs, outputs, markers, bytes, words, doublewords, timers, counters) are written to the log book. Each entry can be given an identification for future processing.
  Module 9276 should not be used in the general PLC program, but rather only for debug purposes, because the processing time may be negatively influenced, unnecessary space used on the hard disk and the log book used for purposes not intended for its function (namely recording keys and error messages).

  Call:

```
PS          B/W/D/K        <identification for operand designation>
                           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 book identification>
                           –1 = no entry
PS          B/W/D/K        <priority>
                           0 = information
                           1 = warning
                           2 = error
CM          9276
Error code:                M4203 = 0:   Entry was written in log book
                           M4203 = 1:   Error code in W1022
                           W1022 = 1:   Invalid priority
                           W1022 = 2:   Invalid identification for operand designation
                           W1022 = 3:   Invalid first operand address
                           W1022 = 4:   Sum from the first operand address and number of
                                     operands invalid
                           W1022 = 5:   Address is not a word/doubleword address
                           W1022 = 20:  Module was not called in a SPAWN or submit job
                           W1022 = 36:  Entry in log book was shortened to 210 characters
```

- Module 9281 is new:
  With module 9281, the cursor can be placed on a particular line in the pallet table selected in one of the operating modes Single Block or Full Sequence. Programmed parameters such as datum shift or datum set can be executed immediately.

```
Call:
PS      B/W/D/K     <line number in the pallet table>
PS      B/W/D/K     <mode>
                    0 = datum shift / do not execute datum set
                    1 = datum shift / execute datum set
CM      9281
PL      B/W/D       <Error code>
                    0: Cursor was placed in the line
                    1: Module was not called in a SPAWN or submit job
                    2: Module was called when NC program already started
                    3: No pallet table selected as main program
                    4: Line does not exist
                    5: Error during datum set, or error in datum or pallet table
Error code:         M4203 = 0:   Cursor was placed in the line
                    M4203 = 1:   Error according to error code
```

Other improvements

- Cycle 33 (TAPER THREAD BORING) is new:
  With auxiliary cycle 33, a thread can be turned on a taper with a Davis head. For this cycle, in addition to the tool axis, a further axis is tracked to the turning spindle for the radial movement.
- The values from the offset adjusting via code number 75368 are displayed in a superimposed window.
- With the machine parameters MP910.x, MP911.x, MP912.x, MP920.x, MP921.x, MP922.x, MP950.x and MP951.x, the values related to the machine datum, and with MP960.x related to the scale datum, are transferred with the "Actual position capture" key. With synchronized axes both values are transferred automatically.
- M136 was changed:
  Feed rate has been changed from mm/min to mm/spindle revolution (280 474-xx: µm/spindle revolution).
- In the NC program the feed rate can be indicated with three decimal places.
- In the OEM cycles, the number of possible transfer parameters has been increased to 32.
- With the soft keys "F" and "F max", the last value input is suggested as the default value.
- With M91 and M92, linear movements can be executed in the machine coordinate system with a tilted working plane.
- With TNCremoNT, data access to the LSV2 protocol is also possible via Ethernet.
- The maximum input value for Q239 in cycle 207 (RIGID TAPPING NEW) and for the pitch in cycles 17 (RIGID TAPPING) and 18 (THREAD CUTTING) was increased to 99.9999 mm.
- The new error number 1068 (datum table?) was introduced for the function FN14: ERROR.

- Traversing the reference-mark function has a new screen layout. On the left side of the screen, the actual positions are shown, on the right side, the designations for the axes to be traversed.
- With cycle 247 (DATUM SHIFT), a datum can be set during program run, whose values are selected via a number in the datum table.
- The cycles for tool measurement are available in ISO under G480 to G483.
- With the new soft key "Optional stop", an NC program can be stopped at the position where M01 is programmed.
- The datum table can be edited with the new soft key "Edit Datum-Table".

1.6 Connecting encoders with EnDat interface

Encoders with EnDat interface can be connected to certain inputs on the LE 426 M and LE 430 M logic units as of hardware version xxx xxx-3x. The evaluation of the encoder signals is supported from NC software 280 474-07.

Numerical control	Speed encoder input	Linear encoder input
LE 426 M / 12 000 rpm	X19, X20	X5, X6
LE 426 M / 24 000 rpm	X19	X5, X6
LE 430 M / 6 axes	X19, X20	X5, X6
LE 430 M / 9 axes	X15 to X20, X62 to X64	X1 to X6

Note:
A maximum of two encoders with EnDat interface can be connected per numerical control!

Cable for connecting encoders with EnDat interface

<table>
<tr><th colspan="4">Position encoder up to a distance of 10 m</th></tr>
<tr><td><i>Adapter cable</i></td><td><i>Extension cable</i></td><td colspan="2"><i>Cable length LC 181</i></td></tr>
<tr><td>332 115-xx</td><td>323 897-xx</td><td colspan="2">Max. 1 m</td></tr>
<tr><th colspan="4">Position encoder up to a distance of 60 m</th></tr>
<tr><td>Adapter cable</td><td><i>Extension cable</i></td><td><i>Voltage controller 5 V</i></td><td><i>Cable length LC 181</i></td></tr>
<tr><td>332 115-xx</td><td>323 897-xx</td><td>336 697-02</td><td>Max. 9 m</td></tr>
<tr><th colspan="4">Speed encoder up to a distance of 15 m</th></tr>
<tr><td colspan="4">Adapter cable</td></tr>
<tr><td colspan="4">336 376-xx</td></tr>
<tr><th colspan="4">Speed encoder up to a distance of 30 m</th></tr>
<tr><td>Adapter cable</td><td><i>Voltage controller 5 V</i></td><td colspan="2"><i>Extension cable</i></td></tr>
<tr><td>336 376-xx (max. 15 m)</td><td>336 697-01</td><td colspan="2">340 320-xx (max. 15 m)</td></tr>
</table>

Definition of the machine parameters for encoders with EnDat interface:

Encoder with EnDat interface at the	Machine parameter setting

Speed encoder input	MP1350.x = 5
	MP110.x = 0

Position encoder input	MP1350.x = 5
	MP110.x = yy

x = index of the axis, yy = designation of the position encoder input

1.7 Hardware

- The **MB 420** machine operating panel (Id. Nr. 293 757-13) supports the inputs I149 to I152 and the outputs O0 to O7. In addition, there are two terminal strips on the underside of the MB 420 with the designations X3 and X4.

X3 Inputs

Terminal	Assignment
1	I 151
2	I 152
3	+24 V

X4 Outputs

Terminal	Assignment
1	O 0
2	O 1
3	O 2
4	O 3
5	O 4
6	O 5
7	O 6
8	O 7
9	0 V

- The new hardware variant xxx xxx-3x of the LE 426 M, LE 430 M has been released. This hardware allows the connection of encoders with EnDat interface to certain encoder inputs.

1 Update Information No. 11

The following NC software has been released:

NC software **280 472-09** and **280 473-09** October 98
NC software **280 472-10** and **280 473-10** November 98
NC software **280 472-11** and **280 473-11** November 98
NC software **280 472-12** and **280 473-12** December 98

NC software **280 474-09** and **280 475-09** October 98
NC software **280 474-10** and **280 475-10** December 98

1.1 Software Error Corrected

The new software corrects an error that could result in collision under certain circumstances:

- In MP7682, bit 0 must be set = 1 (Tool length is not offset in an I block following a TOOL CALL).
- If the above bit = 1, the tool length of the replacement tool is not compensated during an automatic TOOL CALL (M101 = automatic tool change with replacement tool).
- If the replacement tool is longer than the previous tool, a collision may result. If it is shorter, its depth of cut may be insufficient.

Error existed as of	Error corrected as of	Release
280 470-11	280 470-12	January 99
280 472-08	280 472-12	December 98
280 474-04	280 474-10	December 98

1.2 NC Software 280 472-xx

NC software	**Release**		
280 472-09	10/98	Export version:	280 473-09

Machine parameters

- MP6500 has been expanded:
  Bit 13 Coordinate system in which the tool is to be measured. If the measurement is not made in the same coordinate system in which the tool touch probe was calibrated, it must be ensured that the tool is perpendicular to the touch probe's contact plate.
  = 0: Tool is measured in the coordinate system in which the tool touch probe was calibrated.
  = 1: Tool is measured in another coordinate system.

PLC modules

- Module 9210 has been expanded:
  The path and file name of the screen mask can be transferred to Module 9210 (open/delete screen mask for PLC window) also as an immediate string.

Other improvements

- Cycle 420 (angle measuring) can also find an angle in the touch probe axis.
- Cycle 427 (coordinate measuring) can automatically compensate tool length and radius.

NC software	**Release**		
280 472-10	11/98	Export version:	280 473-10

Other improvements

- Pressing the MOD key now also shows the version of the setup diskettes.

NC software	**Release**		
280 472-11	11/98	Export version:	280 473-11

Machine parameters

- MP334 has been added:
  With MP334 you can define the number of grating periods between the datum pulses on encoders with distance-coded reference marks.
  Input: 1 to 65535
  0 = 1000
  MP334.0 Axis 1
  to
  MP334.8 Axis 9

NC software	**Release**		
280 472-12	12/98	Export version:	280 473-12

1.3 NC Software 280 474-xx

NC software	**Release**		
280 474-09	10/98	**Export version**:	280 475-09

Machine parameters

- MP334 has been added:
  With MP334 you can define the number of grating periods between the datum pulses for encoders with distance-coded reference marks.
  Input: 1 to 65535
  0 = 1000
  MP334.0 Axis 1
  to
  MP334.8 Axis 9
- MP7289 has been added:
  MP7289 can define the display step of the spindle position.
  Input: 0 = 0.1 [°]
  1 = 0.05 [°]
  2 = 0.01 [°]
  3 = 0.005 [°]
  4 = 0.001 [°]
  5 = 0.0005 [°]
  6 = 0.0001 [°]

PLC programming

- FN18: SYSREAD has been expanded:
  The function FN18: SYSREAD Qxxx = ID200 NR1 provides the current tool radius with algebraic sign.
- FN17: SYSWRITE has been expanded:
  FN17 can find the current line of a pallet table.
  ID510
  NR3
- FN17: SYSWRITE has been expanded:
  FN17 can find the last line of the NC program for the current pallet.
  ID510
  NR4

PLC Modules

- Module 9210 has been expanded:
  The path and file name of the screen mask can be transferred to Module 9210 (open/delete screen mask for PLC window) also as an immediate string.
- Module 9247 has been expanded:
  With Module 9247 (searching for a condition in a table) you can search in all tables for a field content.

- Module 9290 has been added:
  With Module 9290 you can select an NC program if the Single Block or Full Sequence mode of operation is selected. The module must be called in a SPAWN job or submit job.

  Call:
  PS B/W/D/K/S <String number or file name>
  CM 9290
  Error code: M4203 = 0: No error
  M4203 = 1: Error code in W1022
  W1022 = 2: Invalid string transferred
  W1022 = 8: Control is not in the Program Run, Single Block or Program Run, Full Sequence operating mode
  W1022 = 20: Module was not called in a SPAWN job or Submit job
  W1022 = 29: Selected file is invalid or does not exist

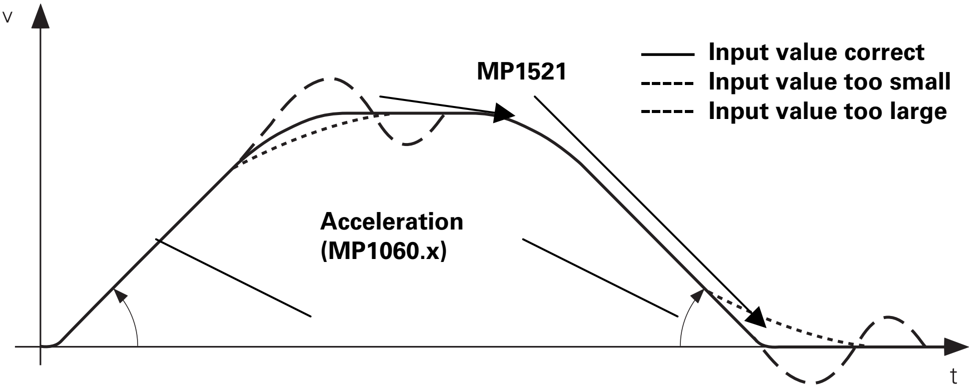
Other improvements

- Cycle 405 (ROT IN C AXIS) has been added:
  The misalignment of a workpiece can be corrected with the C axis.
- Cycle 420 (angle measuring) can also find an angle in the touch probe axis.

NC software	**Release**		
280 474-10	12/98	**Export version**:	280 475-10

Machine parameters

- MP1521 has been added:
  With MP1521 you can influence the overshoot behavior during acceleration and braking.



Input: 0 = Function inactive
1 to 255 [ms]

- MP2360, MP2361 and MP2391 have been added:
  In the event of an emergency stop, the spindle must be braked to a stop as quickly as possible. Increasing the recovered power shortens the braking process. However, if the recovered power is too high, the braking energy is not dissipated quickly enough: the inverter may under certain circumstances switch off and the spindle will coast to a stop.

  For **spindles without a dedicated spindle DSP,** you can enter a time constant in MP2360.x or MP2361 by which the recovered power is reduced. The larger the time constant, the less the mean recovered power during braking, and the longer the braking process.

  To find the optimum value, begin with a large time constant. Then make an emergency stop to brake the spindle from maximum speed. Reduce the time constant until you find the shortest possible braking time at which the inverter does not switch off. Since the temperature of the braking resistor and the mass of the tool in the spindle affect the braking power, this time constant must be increased somewhat to provide a safety margin, or it must be found with the heaviest tool and a hot braking resistor.

  MP2360.x Time constant for braking a second digital spindle through an emergency stop

Effective MP2360.x	PWM output of the second spindle
MP2360.0	X51
to	to
MP2360.8	X59

MP2361 Time constant for braking the spindle through an emergency stop
Input: 0.00 to 5.00 [s]
0 = Function inactive (braking depends only on MP2191)

For **spindles with dedicated spindle DSP** the braking power can be limited in MP2391.

Input value $MP2391.x = 0,9 \cdot \frac{U_{Zmin}^2}{R}$

U_{Zmin} = minimum dc-link voltage [V]
R = Braking resistor [Ω]
MP2391.0 Wye connection
MP2391.1 Delta connection
Input: 0.0 to 3000.0 [kW]
0 = Braking power is not limited
(primarily for inverters with energy recovery)

- MP6500 has been expanded:
  Bit 13 Coordinate system in which the tool is to be measured. If the measurement is not made in the same coordinate system in which the tool touch probe was calibrated, it must be ensured that the tool is perpendicular to the touch probe's contact plate.
  = 0: Tool is measured in the coordinate system in which the tool touch probe was calibrated.
  = 1: Tool is measured in another coordinate system.
- MP7160 has been expanded:
  With MP7160 bit 2 you can define whether the spindle is position controlled during rigid tapping. If the spindle is position controlled it shows a better speed curve.
  Input: %xxx
  Bit2 = 0: Spindle is not position controlled
  = 1: Spindle is position controlled

- MP7245 has been added:
  With MP7245 you can disable the auxiliary cycles, e.g. Cycle 18 (thread cutting).
  Input: 0 = Auxiliary cycles permitted
  1 = Auxiliary cycles disabled

Machine Interfacing

- The log has been expanded:
  The control model and NC software is entered in every log. In addition, the selected files and the condition of the control-in-operation symbol (symbol on, symbol blinks, symbol off) is entered.
- Format instruction for the large PLC window was expanded:
  The special command TABREAD (line, column) can be used in conjunction with the switch /c to display cells in a table and updated them cyclically.
- The drive is no longer switched off automatically if the fast PLC input defined in MP4130.0 is set. It must be switched on through the PLC program or the wiring.

PLC programming

- The maximum size of the PLC program was increased to 512 KB.
- FN18: SYSREAD has been expanded:
  The index of the active tool can be found.
  ID20
  NR11
- FN18: SYSREAD has been expanded:
  With FN18 you can find out whether the run is in block scan or in the automatic operating modes. The following values are reported:
  ID990
  NR2
  IDXx 10: Execution in block scan?
  16: Execution in the automatic operating modes?

Reported value	Meaning
0.0	Block scan/Automatic mode not active
1.0	Block scan/Automatic mode active
–1.0	Invalid index

- FN18: SYSREAD has been expanded:
  You can ask whether M128 is active (–1 = M128 active, 0 = M128 not active).
  ID280
  NR1
- FN18: SYSREAD has been expanded:
  FN18 provides the feed rate that was programmed with M128. The feed rate is the maximum speed of compensation movements in the linear axes.
  ID280
  NR2

PLC modules

- Module 9035 has been expanded:
  Module 9035 was expanded by the transfer values 22 (M128 active) and 1022 (pallet machining condition).

Transferred number:		Values read:
22	M128 active	0: M128 not active 1: M128 active
1002	Pallet machining status	–1: Main program is not a pallet table 0: machining has not started 1: NC program selected, but not started 2: NC program started 3: Pallet change macro started 4: Macro started from the PALEPILOG entry in NCMAKRO.SYS 5: Pallet change macro started by PLC (Module 9280)

- Module 9159 has been added:
  With this module you can ask which drives will be switched off in 200 ms.

  Call:

```
CM      9159
PL      W/D         <Drives bit-coded that will be switched on in 200 ms>
                    Bit     15                 0
                    Axis    Sxxxxxx987654321
```

  Error code: M4203 = 0: No error

- Module 9169 has been added:
  With Module 9169 you can define axes that are not switched off by the input I32 (X42/33, drive enable, acknowledgment for control-is-ready signal) . This module works only in conjunction with the TNC 426 M / TNC 430 M controls.

  Call:

```
PS      B/W/D/K     <Axes bit-coded>
CM      9169
```

  Error code: M4203 = 0: No error possible

- If in Modules 9120, 9123, 9221 or 9223 a noncontrolled axis is programmed, an error is reported.

Module	Reported error code
9120, 9123, 9221, 9223	5
9220	8
9122	Bit 5 = 1

Other improvements

- The setup diskettes contain the current *.CDF and CONSTCYC.CDC files.
- Two new soft keys appear during use of the manual measuring cycles in connection with datum tables. You can select whether the result of measurement is transferred to the datum table or as a datum to the control.
- Reentry in a radius-compensated NC program with block scan is necessary only if a transitional spline was interrupted.
- The default value of Q208 in the fixed cycles 201 (REAMING), 202 (BORING) and 203 (UNIVERSAL DRILLING) was increased from 500 mm/min or 20 ipm to 30 000 mm/min or 1200 ipm.
- The starting point for the finishing process in the fixed cycles 210 (SLOT) and 211 (CIRCULAR SLOT) is approached with the programmed machining feed rate.
- With Cycle 427 (MEASURE COORDINATES) you can automatically compensate tool length and radius.
- The MOD key now also calls the version of the setup diskettes.
- With the manual measuring cycles the datum must be transferred with a soft key. In addition, you can enter delta values (distances between measured points and datum).

1.4 Hardware

1.4.1 Securing the Hard Disk of the LE 426 M/LE 430 M for Shipping

The hard disks of the LE 426 M and LE 430 M can be secured for shipping. Before the LE is transported, the hard disk is fastened to the housing with two screws. There are two vacant threaded holes in the housing to hold the screws after they have been removed from the hard disk. A sticker on the LE describes how to secure the hard disks and release them again.

- Before commissioning the LE, the bracing screws must be removed from the hard disk and driven into the vacant holes.
- If the LE is to be shipped already installed in a machine tool, it is not necessary to brace the hard disks.
- If service becomes necessary (i.e., the LE is shipped alone) the hard disks must be secured with the bracing screws.

1.4.2 New UV 111A/UV 111B Power Module

With the new **UV 111x** power module it is possible to configure the **LE 426 M/LE 430 M** in two rows together with the SIMODRIVE 611 D inverter system.

The UV 111x power module supplies the power for the LE, conducts the PWM signals from ribbon cable to D-sub connections, and is arranged to the left of the LE.

No additional ribbon cables are required. The cover for the UV 111x is supplied together with the UV 111x.

The old expansion cards of the LE 426 B/LE 430 and die PWM cable with D-sub connectors are required for the SIMODRIVE 611 D inverter system.

Product	**ID Number**
UV 111A (6 axes + spindle)	317 559-22
UV 111B (9 axes + spindle)	317 559-32
Expansion card with D-sub ports	324 952-10
PWM cable with C-sub connectors	289 208-xx

1 Update Information No. 10

1.1 NC software 280 472-08

NC software	Release		
280 472-08	07/98	Export version:	280 473-08

Machine parameters

- MP2510 and MP2511 expanded:
  Maximum input value reduced to 30 000 [A]
- MP2191 new:
  Deceleration of the spindle during Emergency Stop
  The spindle must be decelerated as quickly as possible during Emergency Stop. If the braking energy cannot be led away quickly enough, the dc-link voltage will rise significantly. In the worst case, the inverter will switch off and the spindle will then turn freely. With MP2191 you can set whether you wish the max. braking current to be monitored and reduced if necessary. Monitoring increases the duration of the braking function somewhat.
  Input: %x
  Bit 0 = 0: Braking with monitoring of the max. braking current
  (preferably for inverters with braking resistance)
  = 1: Braking without monitoring of the max. braking current
  (preferably for inverters with energy recovery)
- MP6161, MP6162 and MP6163 new:
  MP6161 defines an M function for orienting the infrared sensor before each measurement. The orientation angle is set in MP6162. Spindle orientation is then carried out when the difference between the current spindle angle and the orientation angle from MP6162 is smaller than the tolerance value in MP6163.
  MP6161 M-function to orient infrared sensor before each measurement
  Input: –1 = Spindle orientation direct via NC
  0 = Function inactive
  1 to 88 = Number of the M-function for spindle orientation

  MP6162 Orientation angle for infrared sensor
  Input: 0 to 359.9999 [°]

  MP6163 Difference between current spindle angle and angle in MP6162 from which a spindle orientation is to be carried out.
  Input: 0 to 3.0000 [°]

PLC programming

- The number of strings was increased from 4 to 8 (now: S0 to S7).
- FN17: SYSWRITE expanded:
  Data of certain tools can be overwritten in the tool table.
  ID50
  NRxx System datum
  IDXxxxx Tool number (0 or NO ENT = current tool)

- FN17: SYSWRITE expanded:
  Cycles 7 (DATUM SHIFT), 8 (MIRROR IMAGE), 10 (ROTATION), 11 (SCALING FACTOR), 26 (AXIS-SPECIFIC SCALING) and 19 (WORKING PLANE) only have local effect in an OEM cycle. With the new function FN17: SYSWRITE ID420 = 0 these cycles become effective globally, i.e. those cycles that are programmed before the FN17:SYSWRITE function also affect the calling program.
  ID420 = 0

PLC modules

- Module 9032 expanded:
  Error code 8: no system buffer
- Modules 9071, 9082 and 9210 expanded:
  An Immediate string can be transferred to these modules. An Immediate string is a string which is entered directly in the PLC program. It can have up to 37 characters and must be within quotation marks.

Others

- "Shut Down" soft key is new:

  To prevent data loss on the hard disk, the soft key OFF was introduced in the Manual Operation mode to bring down the control. After confirming your intentions with the YES soft key, the control is brought down. Now the control can be switched off or brought up again with the END key.
- DIN/ISO: With spindle orientation with G36, a Q parameter can also be programmed as the angular value.
- Touch probe cycles 421 to 426 allow automatic tool compensation in the tool table and breakage tolerance is monitored.

1.2 NC software 280 474-xx

NC software	**Release**		
280 474-06	07/98	Export version:	280 475-06

NC software	**Release**		
280 474-07	08/98	Export version:	280 475-07

Machine parameters

- MP1094 new:
  For controls with their own spindle DSP, an HSC filter can be activated instead of the single or double filter. A cutoff frequency must be entered in MP1094. The cutoff frequency should be approx. 20% to 30% below the resonance frequency of the machine. The axis-specific jerk from MP1098 is valid. MP1095, MP1097 and MP1099 have no effect. With input value 0, the old filters are active.
  Input: 0 = No HSC filter active
  0 to 166.0 [Hz] Cutoff frequency for an HSC filter
- MP1220 was removed as it had no function.
- MP1390 was removed and MP1392 was added instead.
- MP1392 is new:
  With MP1390 operation with servo lag or velocity feedforward control could only be selected for all axes. With MP1392 operation with servo lag or feedforward can be set on an axis-specific basis. M90 (operation with servo lag: constant speed at corners) is only effective if operation with servo lag is selected for all axes.
  Input: %xxxxxxxxx
  Bit 0 to 8 0 = Operation with servo lag
  Axis 1 to 9 1 = Operation with velocity feedforward control
- MP2560 and MP2561 new:
  With machine parameters MP2560.x and MP2561 a 1st or 2nd order low pass filter can be activated for the axes and the spindle. This filter is suitable for dampening disruptive high-frequency vibrations (> 600 Hz). If you have problems with such vibrations, first try this filter. Use the first order filter for vibrations of < approx. 700 Hz and the second order filter for vibrations of > approx. 700 Hz. If these filters do not give satisfactory results, then try using the PT_2 time-delay element (MP2530).
  MP2560.0 to MP2560.8 low pass filter for the axes 1 to 9
  MP2561 low pass filter for the spindle
  Input: 0 = No low pass filter
  1 = low pass filter 1st order
  2 = low pass filter 2nd order
- MP6161, MP6162 and MP6163 new:
  see improvements of NC software 280 472-08
- MP7267 expanded:
  MP7267.5 Tool name (TNAME)
- See also the section "Machine Integration."

- MP4020 expanded:
  The values of the Pt100 inputs are transferred with a change speed of 1 K/s. The disadvantage is that with larger change speeds, it can take quite a long time to transfer the correct temperature. For example it would take 30 s to transfer a change of 30 K. The advantage of this change speed of 1K/s is the low influence of disturbance, i.e. a temperature display will not "jump" between values. With bit 7 you can switch between a change speed of 1 K/s and immediate transfer of the Pt100 inputs.

Input:	%xxxxxxxx
Bit6	Reserved
Bit7	= 0: Transfer values of the Pt100 inputs at a change speed of 1 K/s.
	= 1: Transfer values of the Pt100 inputs immediately.

- MP7367 new:
  With MP7367 you can set the colors for the large PLC window.
  Input: see 4.13.2 Setting the colors

MP7367.0:	background $0333333	
MP7367.1:	color 1	$0281408
MP7367.2:	color 2	$0140A04
MP7367.3:	color 3	$02F2818
MP7367.4:	color 4	$0100C08
MP7367.5:	color 5	$02F2818
MP7367.6	color 6	
to	to	
MP7367.14	color 14	Gray shades

- MP7600 expanded:
  With MP7600, both position-controller and PLC cycle time can be set. A factor is entered in MP7600.0 which when multiplied by 0.6 ms, sets the position controller cycle time.
  The factor from MP7600.1, multiplied by the position-controller cycle time, sets the PLC cycle time.

  With input value 5 in MP7600.0 the TNC will have a position-controller cycle time of 3 ms, as before. Sometimes, with processing-intensive applications (e.g. M128) or several (5 to 9) axes, the error PROCESSOR CHECK ERROR B will result. In this case increase the position controller cycle time to 3.6 ms, by entering the factor 6 in MP7600.0.

  This increase will also cause an increase in the PLC cycle time. To restore the original PLC cycle time, enter the factor 6 in MP7600.1; the PLC cycle time is now 21.6 ms.

  MP7600.0 position controller cycle time = MP7600.0 * 0.6 ms
  Input: 5 to 10 (suggested input value: 5)

  MP7600.1 PLC cycle time = MP7600.1 * position controller cycle time
  = MP7600.1 * MP7600.0 * 0.6 ms

  For applications leading to a PLC cycle time < 20 ms, the PLC cycle time is limited to 20 ms.
  Input: 1 to 20 (suggested input value: 7)

- MP7650 expanded:
  The machine parameter MP7650 is bit-coded axis-specifically; thus it is possible to set the count direction individually for several handwheels via the handwheel adapter HRA . If only one handwheel is being used, bit 0 is always effective.

Input:	%xxxxxxxxx	
Bit	0 to 8	0 = Negative count direction
Axis	1 to9	1 = Positive count direction

- MP7500 expanded:

Bit5	Tilting axis position during datum set
	= 0: The current tilting axis position is used, referenced to the machine datum.
	= 1: The tilting axis position is used, which is set with the 3D ROT soft key.

Machine integration

- The gear range from W256 (G code with S-analog) is output with spindle speed = 0.
- The pocket table was expanded by the column TNAME. The tool name from the tool table is entered in this column and therefore cannot be edited. With indexed tools, the name of the tool is entered with the index 0.
- The column LOCK can be added to a pallet table. Lines containing any entry in this column will not be processed. If several programs or pallets are being run, the TNC will move to the next unlocked line. Likewise unlocked lines in a locked pallet will be skipped.

PLC programming

- FN17: SYSWRITE ID420 = 0 new:
  see improvements for NC software 280 472-08
- FN17: SYSWRITE expanded:
  With the new touch probe cycle 3 it is possible to switch between inputs X12 and X13. If the value 1.0 is assigned (= input X13), marker M4060 is set. If 0.0 is assigned (= input X12), marker M4060 is reset (if it had been set).
  ID990
  NR6 = <input>
  0.0 = input X12
  1.0 = input X13
- FN26: TABOPEN, FN 27: TABWRITE and FN28: TABREAD new:
  These FN functions open numeric cells for freely definable tables which can then be written to or read from. Only one table can be open at any one time. A new TABOPEN closes an open table and opens a new one.
  FN26: TABOPEN <path> path of the table to be opened (*.TAB)
  FN27: TABWRITE <line number> / <"column name"> = Qxxx
  The content of the Q parameter (Q0 to Q399) is written to the cell which is defined by the line number and the column name. Up to 8 column names can be defined, divided by commas. Only the first Q parameter needs to be indicated. The following Q parameters are assigned accordingly.
  FN28: TABREAD Qxxx = <line number> / <"column name">
  The content of the cell, defined by the line number and the column name, is stored in the Q parameter (Q0 to Q399). Up to 8 column names can be indicated, divided by commas. Only the first Q parameter needs to be indicated. The following Q parameters are assigned accordingly.

- Formatting options expanded for the large PLC window:
  Each line can now contain more than one input field. Selection is made via the cursor keys.
  The switches for variables have been expanded:

  With the switches /e and /i, you can indicate an identifier (/e = n, /i = n). It is then possible to use this identifier to determine, using the expanded module 9211, in which field the cursor is located.

  With the new switch /s = n a field is generated in which no input can be made. You can use the identifier however to determine whether the cursor is located in this field.
- New code words for OEM.SYS:

MPPASSWORD	Code word for calling the machine parameter editor (instead of 95148). If a code word is entered under MPPASSWORD, the machine parameters can no longer be edited with the standard code word 95148.
MPLOCKFILE	The path of a machine parameter subfile can be entered. If differences exist with the current machine parameter file (e.g. after read in), a window is superimposed with the message to change the MP to the values suggested in the subfile.
- New code word for NCMAKRO.SYS:

PALEPILOG	An NC macro (with complete path) can be defined. This macro will be called at the end of every NC program which was started from the pallet table.

Markers and words

- W1062 new:
  W1062 inhibits handwheel pulses on an axis-specific basis if several handwheels are being used via the HRA 110 handwheel adapter. If marker M4576 (Suppression of handwheel pulses) is set, then all axes are inhibited. When marker M4576 is reset, W1062 is valid. If only one handwheel is being used, it can be inhibited with W1062.

PLC modules

- Module 9153 new:
  With module 9153 you can define a new probe axis (axis 0, 1 or 2) for manual measurement. A new probe axis can only be defined if bit 2 from MP7490 is set. The module must be called in a SPAWN or Submit job

  Call:

  ```
  PS   B/W/D/K      <axis number 0...2>
  CM   9153
  Error code:       M4203 = 0:   No error
                    M4203 = 1:   Error code in W1022
                    W1022 = 2:   Invalid axis number
                    W1022 = 20:  Module was not called in the SPAWN or submit job.
  ```

- Module 9211 expanded:
  Identifiers can be assigned to the fields of the large PLC window. The new transfer value 4 indicates the identifier of the field in which the cursor may be found, see also PLC programming.

  ```
  Call:
  PS   B/W/D/K     <number of the status information>
                   0 = Status
                   1 = Horizontal length
                   2 = Vertical length
                   3 = Displayed page
                   4 = Identifier
  CM 9211
  PL   B/W/D       <status information>
                   –1 = error
  Error code:      M4203 = 0:   No error
                   M4203 = 1:   Error code in W1022
                   W1022 = 1:   Invalid number of the status information
  ```

- Module 9280 new:
  Module 9280 starts the NC macro which is called when running a pallet entry (PAL). The NC macro must be defined in the file PLC:\NCMAKRO.SYS under the entry PALETT.

  The NC macro can only be activated if the control is in the Full Sequence or Single Block mode and only if a pallet table is selected and no NC macro or NC program is being run at that time. The NC macro can read the transferred values with the function FN18: SYSREAD Qxxx = ID510 NR1 or NR2.

  ```
  Call:
  PS   B/W/D/K     <pallet number>
  PS   B/W/D/K     <line number>
  CM   9280
  Error code:      M4203 = 0:   No error
                   M4203 = 1:   Error code in W1022
                   W1022 = 7:   The file entered under PALETT in NCMAKRO.SYS does not exist.
                   W1022 = 8:   The control is not in Full Sequence or Single Block mode.
                   W1022 = 20:  Module was not called in SPAWN or Submit job.
                   W1022 = 28:  An NC program or an NC macro is already being run.
                   W1022 = 29:  No pallet table is selected.
                   W1022 = 30:  There is no entry PALETT in NCMAKRO.SYS.
                   W1022 = 36:  NCMAKRO.SYS does not exist.
  ```

Other improvements

- A customer-specific company logo can be displayed while the control is starting. To convert a picture of this logo to a format that can be read by the TNC (16-color bitmap format, pixel width divisible by 8), you need the conversion tool Bmp2Logo 1.0. This tool includes two files. The *.SYS file must be copied to the main directory PLC:\ as LOGO.SYS and the *.VEC file must be copied to the directory PLC:\LOGO as LOGO.VEC.
- Soft key "Shut down" new:
  See improvements for NC software 280 472-08
- The cycle structure can be composed of up to 9 cycle trees. This function is only supported with CycleDesign as of version 2.2.
- M117 new:
  M117 can deactivate module M116 (feed rate in mm/min for angular axes).
  M116 is also automatically deactivated when a tool change macro or a manufacturing cycle is being executed.
- M136 and M137 new:
  M136 switches the programming of the contouring feed rate from mm/min to µm/spindle/revolution. The position of the spindle override is taken into consideration. M137 switches back to mm/min.
- M138 new:
  Initially all of the tilting axes defined in MP7510 are affected by the functions M114, M128 and "Tilt the working plane." With M138 you can define axes which are affected by these functions, but not by M116 (feed rate in mm/min for angular axes). M138 with no axis indication cancels the previous selection made.
- The new touch probe cycle 3 allows probing without retraction. The cycle can be effective at input X12 or input X13. You can switch between the two inputs with FN17: SYSWRITE ID990 NR6 = <input>.
- The block number for the mid-program startup or the block number up to which the test run is to be carried out, is superimposed in a window.

NC software	**Release**		
280 474-08	09/98	Export version:	280 475-08

Other improvements

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

1.3 Hardware

- The **BTS 1x0** has been added to the standard product program. Thus it is now possible to connect 2 keyboards and 2 monitors to one LE. Both monitors are always active. You can switch between keyboard units via a PLC input on the BTS 1x0.

	Connections	ID number
BTS 110	2 x BC 120 2 x TE 420 1 x PLC input for switching keyboards 1 x keyboard connection from the LE 1 x monitor connection from the LE	317 292-01
BTS 120	2 x BF 120 2 x TE 420 1 x PLC input for switching keyboards 1 x keyboard connection from the LE 1 x monitor connection from the LE 1 x power supply unit	329 965-01

- The mounting foot of the **TT 120** table touch probe was updated. The TT can now be secured either with 2 fixing clamps or via an adapter with screw.
  (ID number: 295 743-xx)
- The PW 120 braking resistor ((Id. Nr. 333 000-01) was added to the standard product program for the new modular controls **TNC 426 M / TNC 430 M.** It should be used from a spindle power output of 20 kW.
  Continuous power: 4 kW
  Peak power: 36 kW (2% cyclic duration factor at a cycle duration of 120 seconds)
- If the **LE 426 M / LE 430 M** is being run with the HEIDENHAIN compact inverter or with the SIMODRIVE 611 D, either the UV 102 or the UV 101 is necessary for power supply to the LE. Voltage must be supplied to terminals U1 and U2 at X31 of the **UV 10x** via an isolating transformer (250 VA). A simple insulation is sufficient for the isolating transformer (basic insulation in accordance with EN 50178 or VDE 0550).

1 Update Information No. 9

1.1 Hardware

The PLC input/output unit PL 405 B (Id. Nr. 263 371-22) has been added to the standard product program. The PL 405 B has 32 switching inputs (24 Vdc) and 15 switching outputs (24 Vdc). No further PL can be connected to the PL 405 B.

	Inputs on the PL 405 B	
	X3	**X4**
Connection of the PL 405 B	Terminal 1 to 16	Terminal 1 to 16
As the only PL	I64 to I79	I80 to I95
To a PL 410 B	I192 to I207	I208 to I223
To the 2nd PL 410 B	I256 to I271	I272 to I287
To the 3rd PL 410 B	I320 to I335	I336 to I351

	Outputs on the PL 405 B	
	X8	**X8**
Connection of the PL 405 B	Terminal 1 to 15	Terminal 16
As the only PL	O48 to O62	"control-is -ready"
To a PL 410 B	O80 to O94	
To the 2nd PL 410 B	O144 to O158	
To the 3rd PL 410 B	O176 to O190	

1.2 NC software

The following NC software versions have been released:

NC software	**Release**		
280 470-11	06/98	Export version:	280 471-11

Improvements:

- MP2510 and MP2511 were expanded:
  Maximum input value reduced to 30 000 [A].
- MP6500 was expanded:
  Bit12 = 0: A PLC datum shift is not considered during tool measurement.
  = 1: A PLC datum shift is considered during tool measurement..
- MP7500 was expanded:
  The TNC displaces the datum automatically if the table is turned and the function "Tilt the working plane" is active.
  Bit3 = 0: The workpiece must be aligned in the 0° position (ref. value) of the rotary table. The displacement value is calculated from the REF coordinate in datum set and the REF coordinate of the tilted axis after tilting.
  = 1: The workpiece is aligned in a rotation of the rotary table and is perhaps no longer in the 0° position (ref. value). The displacement can no longer be calculated from the difference in the REF coordinates. Instead the REF value of the tilted axis after tilting is used directly.

NC software	**Release**		
280 474-04	05/98	Export version:	280 475-04

Improvements:

- MP860 was expanded:
  With input value 2, an axis is defined as a torque-slave axis.
- MP2900 is new:
  Torque bias between master and slave in master/slave torque control. The torque bias must be entered at the torque-slave axis.
  Input: –100.00 to +100.00 [Nm]
  MP2900.0 Axis 1
  to
  MP2900.8 Axis 9
- MP2910 is new:
  Gain (proportional gain) of the torque controller in master/slave torque control. The gain must be entered at the torque-slave axis.
  Input: 0.00 to 999.99 [1/(min*Nm)]
  MP2910.0 Axis 1
  to
  MP2910.8 Axis 9
- MP2920 is new:
  Factor for variable torque distribution in master/slave torque control. The factor must be input at the torque-slave axis. If identical motors are used for both torque-master and torque-slave axes, the value 1 must be input.
  Input: 0.000 to 100.000
  MP2920.0 Axis 1
  to
  MP2920.8 Axis 9
- MP2930 is new:
  Speed compensation ratio between torque-master and torque-slave axes in master/slave torque control. The speed compensation ratio must be entered at the torque-slave axis.
  Input: –100.00 to +100.00 [%]
  MP2930.0 Axis 1
  to
  MP2930.8 Axis 9
- MP7160 was expanded:
  Limiting the spindle speed in rigid tapping
  Bit 1 = 0: Spindle speed is not limited.
  = 1: With small thread depths, the spindle speed is limited so that the spindle will turn at a constant speed for approx. 1/3 of the time.
- MP7263 is new:
  Display/do not display "Pocket Table" soft key.
  Input: %x
  Bit 0 = 0: Display "Pocket Table" soft key
  = 1: Do not display "Pocket Table" soft key

- MP7440 was expanded:
  Bit6 = 0: M134 (exact stop at non-tangential contour transitions during positioning with rotary axes) is activated as always in the NC program.
  = 1: M134 is activated automatically when an NC program is selected.
- MP7441 is new:
  With MP7441, the error message "Spindle must be turning" is suppressed when a machining cycle without M3 or M4 is called.
  Input: %x
  Bit0: = 0: Do not suppress error message
  = 1: Suppress error message
- MP 7683 was expanded:
  Bit4 = 0: Current pallet table cannot be edited.
  = 1: In the operating modes "Program Run/Full Sequence" and "Program Run/Single Block" the soft key "EDIT PALLET" appears to allow you to edit the current pallet table.
- FN17: SYSWRITE was expanded:
  Data of certain tools can be overwritten in the tool table.
  ID50
  NRxx System date
  IDXxxxx Line number of the tool table (0 = current tool)
- Modules 9092 and 9093 were expanded:
  The Modules 9092 and 9093 were expanded by the element numbers 24 (tool number) and 25 (tool index).
- Module 9147 is new:
  With Module 9147, a new ref value is assigned to the current position of an axis.
  This is necessary if an axis is mechanically fixed after positioning and then the encoder is moved.
  If the axis is then positioned again, the original ref value is reassigned to the current position, which has not changed due to the mechanical fixing.

  Call:

```
PS   B/W/D/K     <axis number>
PS   B/W/D/K     <new ref value in 1/10 µm>
CM   9147
Error code:      M4203 = 0:    No error
                 M4203 = 1:    Error code in W1022
                 W1022 = 2:    Invalid axis number
                 W1022 = 21:   Missing strobe or M4176 = 1
                 W1022 = 24: Module was called in SPAWN or SUBMIT job
```

- Module 9155 is new:
  With module 9155 you can switch axes from controlled to non-controlled status (e.g. to position them manually).

  ```
  Call:
  PS   B/W/D/K    <axes bit-coded>
  CM   9155
  Error code:     M4203 = 0:   No error
                  M4203 = 1:   Error code in W1022
                  W1022 = 2:   Invalid axis number
                  W1022 = 21:  W1022 = 21:  Missing strobe or M4176 = 1
                  W1022 = 24: Module was called in SPAWN or SUBMIT job
  ```

- Module 9156 is new:
  With 9156 you can switch axes that you have already switched from controlled to non-controlled status with 9155, back to controlled status.

  ```
  Call:
  PS   B/W/D/K    <axes bit-coded>
  CM   9156
  Error code:     M4203 = 0:   No error
                  M4203 = 1:   Error code in W1022
                  W1022 = 2:   Invalid axis number
                  W1022 = 21:  W1022 = 21:  Missing strobe or M4176 = 1
                  W1022 = 24: Module was called in SPAWN or SUBMIT job
  ```

- Module 9220 was expanded:
  Module 9220 (traverse over the reference marks a second time) also has effect for PLC axes.
- Module 9225 was expanded:
  With module 9225 a compensation value can be assigned to the zero pulse before reference-mark traverse. It is then possible to use encoders with more than one reference mark. The compensation value must be the distance between the reference mark being traversed and the reference mark whose distance to the machine datum is entered in MP960.x. Both NC and PLC axes can be selected.

  ```
  Call:
  PS   B/W/D/K    <axis>
                  0 to 8 = axis 1 to 9
                  15 = spindle
  PS   B/W/D/K    <compensation value>
                  0 = starting value for zero pulse
  CM   9225
  PL   B/W/D      <error code>
                  1 = axis does not exist
  Error code:     M4203 = 0:   No error
                  M4203 = 0:   Axis does not exist
  ```

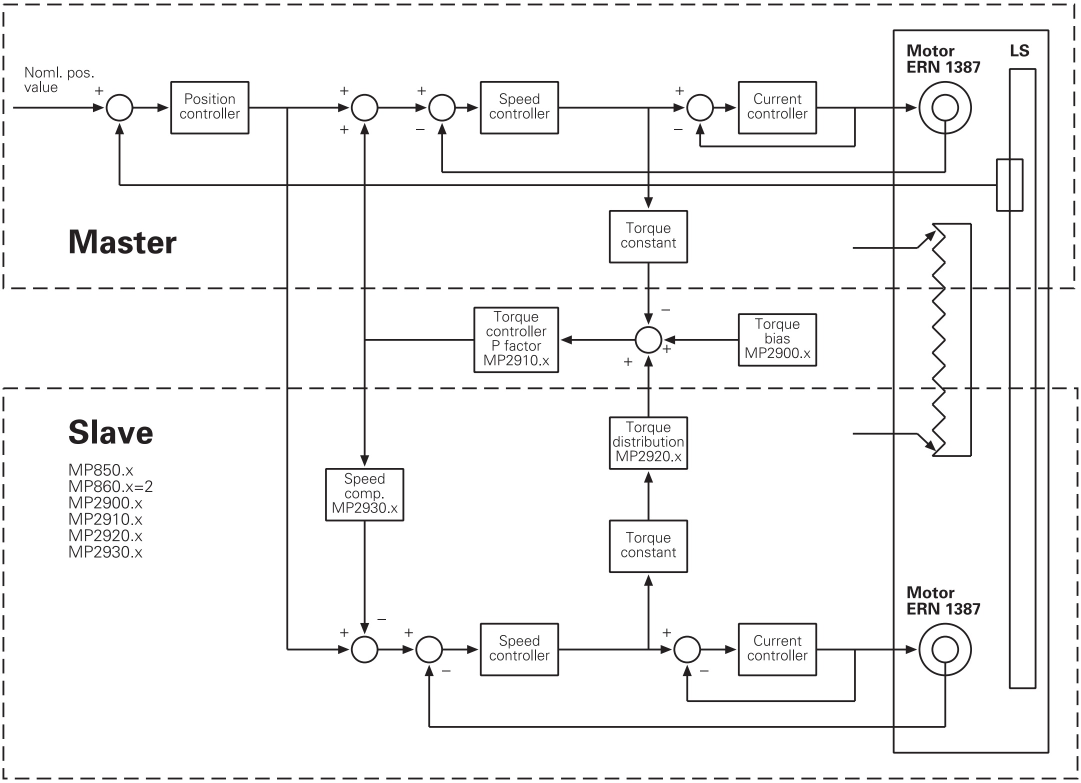
NC software	**Release**		
280 474-05	06/98	Export version:	280 475-05

Improvements:

- MP6150, MP6361 and MP6550 were expanded:
  Maximum input value increased to 20 000 mm/min.
- MP7266 was expanded:
  MP7266.27 PLC value (PLC-VAL)
- MP7600 is new:
  MP7600.0 reserved; input: 0
  MP7600.1 PLC cycle time = MP7600.1 * 3 ms
  Input: 1 to 20 (suggested input value: 7)
  For input values that would result in a PLC cycle time < 20 ms, the PLC cycle time is limited to 20 ms.
- The position of the working plane (MP7500 bit1 = 1) was expanded by the following tilting-axis combinations:
  Rotary and tilting table: axis sequence C variable / A variable (tool axis Y)
  Swivel head 45° and rotary table: axis sequence A fixed / C variable / A fixed / B variable (tool axis Y)
- It is now possible to drive an axis and a spindle with the same motor in so-called C-axis operation (see also 1.4 C-axis operation).
- In the program modes Program Run/Single Block, Program Run/Full Sequence and Positioning with MDI, if the position control loop is opened at standstill or at machine stop by the setting of M4581 (e.g. by opening the doors), and then the axes are moved, the resetting of M4581 (e.g. by closing the doors) will activate the function Approach Position.
- PLC axes can be operated with feedforward control. Axis-specific jerk is taken into consideration (MP1097.x and MP1098.x).
- The tool table was expanded by the column PLC-VAL. In this column, certain values can be exchanged between the PLC and the tool table. This column has a width of 11 characters (input range: –99999.9999 to +99999.9999).
- FN17: SYSWRITE was expanded:
  Data can be written to the new column, PLC-VAL, of the tool table.
  ID50
  NR23
  IDXxxxx Tool number
- FN18: SYSREAD was expanded:
  Data can be read from the new column, PLC-VAL, of the tool table.
  ID50
  NR23
  IDXxxxx Tool number
- FN18: SYSREAD was expanded:
  The last reference point of a manual touch probe cycle or the last touch point of the touch probe cycle 0 (reference level) can be read.
  ID360
  NRx 1 = workpiece coordinate system
  2 = machine coordinate system
  IDXx axis number (1 to 9 X/Y/Z/A/B/C/U/V/W)
- The modules 9092, 9093, 9094 were expanded by the element number 26 (PLC-VAL).
- Module 9145 was expanded:
  If the module is only used for programming PLC axes, no strobe or M4176 = 1 is necessary.
  If NC axes are also being programmed, a strobe or M4176 = 1 is necessary.

1.3 Master-slave torque control

Two motors which are coupled mechanically constitute one master-slave torque pair.
The position control of the slave is deactivated. The nominal speed of the master axis is the same as the nominal speed of the slave axis. The speed controllers for both axes are active - independent of each other. The variables of the speed controllers, the nominal torque currents, weighted with the torque constants of the motors are compared with each other. A torque bias (MP2900.x) can be added to the comparison at this point. To enable a distribution of the drive torques, the nominal torque of the slave axis can be multiplied by a weighting factor (MP2920.x). The result at this comparison point is input into a torque compensating controller (MP2910.x) which amplifies it proportionally. The variable of this torque compensating controller is a speed compensation value, which is divided and added to the current nominal speed values of the master and slave axes, usually in equal proportions. Sometimes however it may be useful to distribute the speed compensation value in an unequal ratio between master and slave axis. Thus a factor for variable speed compensation (MP2930.x) can be entered.



1.4 C-axis operation

With NC software 280 474-05, it is possible to move an axis and a spindle with the same servo drive. This is called C-axis operation as the spindle motor of the tool axis Z is often used also to drive a rotary axis. The rotary axis corresponding to the tool axis Z is the C axis. It is however possible to drive any other axis with the spindle drive. The spindle and the axis can each have their own position encoder. As the speed encoder is built into the motor, both spindle and axis will have the same speed encoder.
Enter the encoder inputs in the corresponding machine parameters, e.g.:
MP110.3: 4 (X4: position encoder for the C-axis)
MP111.0: 6 (X6: position encoder for the spindle)
MP112.3: 0 (speed encoder for the C axis: the speed encoder for the spindle is used from MP113.0)
MP113.0: 60 (X60: speed encoder for the spindle)

The spindle and the axis can only be operated digitally. It is not important whether the digital spindle has its own spindle DSP or not.
As the same drive is used for the axis and the spindle, the same output is entered in the machine parameter for nominal speed value outputs, e.g.:
MP120.3: 61 (X61: PWM output for the C-axis)
MP121.0: 61 (X61: PWM output for the spindle)

Commissioning of the current and speed controller only takes place for the spindle; the value zero is entered in the machine parameter for the current and speed controller of the axis (MP2xxx.x).
Commissioning of the position encoders is separate for the spindle and for the axis.

Use bit 15 (spindle) from module 9161 to enable the integrated drive controller (current and speed controller) for the axis and the spindle.

If the position controller for the axis is closed (corresponding bit set in W1024), the output of a spindle speed has no effect. For a spindle speed to be output, the position controller for the axis needs to be open (corresponding bit not set in W1024). An axis can be switched from controlled to noncontrolled status with module 9155. With module 9156 controlled status is regained and an automatic actual and nominal position capture is carried out.

1.5 User functions in NC software 280 474-xx

The following improvements to the user functions have been made to date in NC software version 280 474-xx:

Improvement	From NC software
In the creation of OEM cycles, a separate standard value entry for INCH programs is supported.	280 474-01
F AUTO can be programmed in OEM cycles.	
New touch-probe cycles for workpiece measurement, with datum setting and sensor calibration.	
With the touch-probe cycles 6, 7, 8, 16, 17 and 18, it is now also possible to digitize just one line only.	
The Q parameters Q150 to Q167 and Q180 to Q182 are used for the measurement results of the touch-probe cycles.	
M128 with radius compensation: The radius compensation RR/RL is effective in the plane perpendicular to the tool axis programmed in TOOL CALL.	
New soft key: F MAX In the Program Run/Single Block and Program Run/Full Sequence modes, the rapid traverse speed can be reduced.	
New soft key: STATUS OF M FUNCT. The active M functions are displayed in their own status window.	280 474-03
M118 positioning can be shown in the status display.	
Indexed tools can be entered.	
In the NC editor it is possible to mark program sections and then copy or delete them.	
ISO: When the tool numbers are programmed, finishing allowances can be entered.	
With M128 you can enter a feed rate for the max. speed of the linear-axis compensation movements.	
Cycle 205: Universal pecking	
Cycle 206: Tapping with floating tap holder	
Cycle 207: Rigid tapping	
Cycle 208: Helical finish milling	
Cycle 200 (Drilling) expanded by input parameters for the dwell time at the bottom of the hole.	
Cycle 203 (Universal drilling) expanded by input parameters for the retraction path during chip breaking.	
Cycle 220 (circle pattern) and Cycle 221 (line pattern) expanded by input parameter as to whether or not the safety clearance should be approached between the machinings in the pattern.	
The cycles 210, 211, 212, 213, 214, 215, 220, 221 are now prepositioned with a positioning logic.	
Touch-probe cycles 410 to 418 expanded by input parameter allowing the datum be set unequal to zero.	
Cycle 28: Cylinder surface slot	
ISO: With G36 (spindle orientation), a Q parameter can be programmed as the angle.	
ISO: In Cycle G80 (Tilt the working plane), a feed rate and a setup clearance can be programmed as an option .	
Cycle 202 (Boring) and Cycle 204 (Back boring) were expanded by input parameter for angle of spindle orientation.	280 474-04

More than one pocket table can exist.	
M104: the datum set manually for all axes is reactivated.	
The ID number for the DSP software is displayed in the info system.	
With the touch probe cycles 421 to 426, an automatic tool compensation can be carried out in the tool table and the breakage tolerance (RBREAK) is monitored.	
New soft key: EDIT PALLET In the "Program Run/Single Block" and "Program Run/Full Sequence" modes, the current pallet table may be edited in pallet operation.	
Face milling: With M128 used in conjunction with LN blocks, the tool is held perpendicular to the contour. A tool direction vector can be entered in addition in the LN block.	
Peripheral milling: With M128 used in conjunction with RR/RL, the negative radius offset (DR) is compensated perpendicular to the movement and tool directions. With M107, errors caused by a positive radius offset (DR) are suppressed.	280 474-04
In the pallet editor, the "Actual position capture" key can be used to transfer actual or reference values or the values from the preceding touch probe cycle for the current or for all axes.	
ISO: Cycle G128 (cylinder surface slot)	280 474-05

1.6 DSP error messages

With Update Information No. 7 you received a list of the DSP error messages. This contained only a list of the DSP error messages for the NC software version 280 472-07. We are now sending you a new list with all of the DSP error messages for the NC software versions 280 470-xx, 280 472-xx and 280 474-xx. With NC software version 280 474-04 the error codes with some of the DSP error messages were changed and some new DSP messages were added.

1.6.1 DSP error messages with error code

Non axis-specific DSP error messages
The error codes are valid for the NC software versions 280 470-xx, 280 472-xx and 280 474-01 to 280 474-03. The error codes in brackets are valid for NC software from version 280 474-04.

Error message		Explanation	Classification
DSP ERROR	**xxxx**	**xxxx: Error code**	
	FF01 (C001)	Undefined error, no causal connection	Data processing error
	FF02 (C002)	Host command not recognized, not valid	Data processing error
	FF03 (C003)	Host / DSP watchdog do not coincide	Data processing error
	FF04 (C004)	Undefined interrupt	Data processing error
	FF05 (C005)	Hardware identification not recognized	Hardware and software do not match
	FF06 (C006)	No V_NOML value received from host	Data processing error

Error message		Explanation	Classification
DSP ERROR	**xxxx**	**xxxx: Error code**	
	FF07 (C007)	AC fail	Power stage error
	FF09 (C009)	Stack overflow	Data processing error
	FF0A (C00A)	Delta signal pulse width modulation	Hardware error or incorrect value in MP2180
	FF0B (C00B)	Error in save command	Data processing error
	FF0C (C00C)	No speed control interrupt	Data processing error
	FF0D (C00D)	Error in check sum (code)	Data processing error
	FF0E (C00E)	Time exceeded in speed interrupt	Data processing error
	FF0F (C00F)	Error initializing a software timer	Data processing error
	FF10 (C010)	Error in LSV2 transfer	Data processing error
	FF11 (C011)	Drive start without previous synchronization	Data processing error
	(C012)	No TL and sync-source initialization	Data processing error

Error message		Explanation	Classification
DSP ERROR	**xxxx y**	**xxxx: Error code** **y: 0 = axes DSP (axes 1 to 6)** **1 = spindle DSP** **2 = axes DSP (axes 7 to 9)**	
	1000 y	Timeout during a command	Data processing error
	1001 y	Incorrect acknowledgement of a command	Data processing error
	1002 y	New command is sent before the previous command is acknowledged	Data processing error
	1003 y	Synchronization error between DSP and NC	Data processing error
	1004 y	Incorrect message DSP → NC	Data processing error
	1005 y	Too many commands NC → DSP	Data processing error
	1100 y	Check sum error	Data processing error
	1101 y	Timeout during word transfer command (load DSP code)	Data processing error
	1102 y	Timeout during check sum	Data processing error
	1103 y	Timeout during GO command	Data processing error
	1104 y	File not found	Data processing error

Axis-specific DSP error messages

The error codes are valid for the NC software versions 280 470-xx, 280 472-xx and 280 474-01 to 280 474-03. The error codes in brackets are valid for NC software from version 280 474-04.

Error message		Explanation	Classification
DSP ERROR	**xxxx y**	**xxxx: Error code** **y: 0 = axis 1** **to** **8 = axis 9** **9 or F = spindle**	
	F010 y (C110 y)	Type of motor unknown (MP2200)	Error in motor table or MP2200
	F020 y (C120 y)	*Reserved*	
	F030 y (C130 y)	*Reserved*	
	F040 y (C140 y)	No. pole pairs too high	Error in motor table or MP2230
	F050 y (C150 y)	ASM: field-determining current	Error in motor table
	F060 y (C160 y)	Grating period of speed encoder	Error in motor table
	F070 y (C170 y)	ASM: rotor time constant	Error in motor table
	F080 y (C180 y)	Kink / rated speed	Error in motor table
	F090 y (C190 y)	Type of drive unknown (MP2000)	
	F0A0 y (F1A0 y)	*Reserved*	
	F0B0 y (C1B0 y)	*Reserved*	
	F0C0 y (C1C0 y)	*Reserved*	
	F0D0 y (C1D0)	Current sensor voltage	Error in power stage table
	F0E0 y (C1E0 y)	Peak current in power stage	Error in power stage table
	F0F0 y (C1F0 y)	Current controller proportional factor too high	
	F100 y (C200 y)	Current controller integral factor too high	
	F110 y (C210 y)	Motor temperature	
	F120 y (C220 y)	*Reserved*	
	F130 y (C230 y)	Osci parameter is incorrect (for test)	Data processing error
	F140 y (C240 y)	Rated current of power stage	Error in power stage table
	F150 y (C250 y)	Rated current of motor	Error in motor table

Error message		**Explanation**	**Classification**
DSP ERROR	**xxxx y**	**xxxx: Error code** **y: 0 = axis 1** **to** **8 = axis 9** **9 or F = spindle**	
	F160 y (C260 y)	Peak current of motor	Error in motor table
	F170 y (C270 y)	Motor maximum speed	Error in motor table
	F180 y (C280 y)	SM: incorrect angle compensation values (MP2340/MP2350)	
	F190 y (C290 y)	Power stage for dc-link voltage incorrect (MP2190)	
	F1A0 y (C2A0 y)	Incorrect speed input selected	
	F1B0 y (C2B0 y)	Invalid PWM output	
	F1C0 y (C2C0 y)	Band filter parameter incorrect (MP2540/MP2550)	
	F200 y (C300 y)	Contamination on encoder Zn-track (amplitude too low)	
	F210 y (C310 y)	Contamination on encoder Z1-track (amplitude too low)	
	F220 y (C320 y)	*Reserved*	
	F230 y (C330 y)	Motor temperature too high	
	F240 y (C340 y)	Unrecognized counter IC type at speed input	Hardware error
	F250 y (C350 y)	Power stage switches off during operation	External operation error
	F260 y (C360 y)	*Reserved*	
	F270 y (C370 y)	Angle deviation too large during alignment; Zn/Z1-tracks do not coincide	Encoder error
	F280 y (C380 y)	Motor cannot be controlled (at Imax no expected rotation)	Drive error
	F290 y (C390 y)	Error in 3-D touch probe / evaluation, it was not latched with L1 input (G19/G26)	Hardware error
	F2A0 y (C3A0 y)	Incorrect ref position	Hardware error
	F2B0 y (C3B0 y)	Standstill identification	Drive error
	F2C0 y (C3C0 y)	Actual motor current above limit value	Drive error
	F2D0 y (C3D0 y)	Status error in PWM component	Hardware error
	F2E0 y (C3E0 y)	Incorrect motor rated current	Error in motor table

1.6.2 DSP error messages with text

Error message	Explanation
Power stage in axis <axis> too weak	Power stage for the displayed axis is too weak.
Motor enc. <axis> line count too high	Line count of the motor encoder for the displayed axis is too high.
Motor <axis>: Xh; X2; f-n; R2 incorrect	One datum of the following listed motor data for the displayed axis is erroneous: Xh: Magnetizing reactance X2: Rotor leakage reactance f-n: Rated frequency R2: Rotor resistance cold
Motor <axis>: n-n; f-n incorrect	One datum of the following listed motor data is erroneous: n-n: Rated speed f-n: Rated frequency
Power stage <axis>: U Imax incorrect	U Imax of the power stage for the displayed axis is incorrect. (U Imax = voltage of the current sensor)
Power stage <axis>: I-max incorrect	Imax of the power stage for the displayed axis is incorrect. (Imax = peak current)
Motor <axis>: t-max incorrect	T-max of the motor for the displayed axis is incorrect.
Motor <axis>: I-n incorrect	I-N of the motor for the displayed axis is incorrect.
Motor <axis>: I-max incorrect	I-max of the motor for the displayed axis is incorrect.
Motor <axis>: n-max incorrect	n-max of the motor for the displayed axis is incorrect.
Axis <axis>: MP2340/MP2350 incorrect	MP2340/MP2350 (field-angle displacement) for the displayed axis is erroneous.
Axis <axis>: MP2190 incorrect	MP2190 (dc-link voltage) for the displayed axis is erroneous.
Axis <axis>: MP120/MP121 incorrect	MP120/MP121 (assignment of the nominal speed value outputs) for the displayed axis is erroneous.
Axis <axis>: MP2540/MP2550 incorrect	MP2540/MP2550 (band-rejection filter damping) for the displayed axis is erroneous.
Motor enc. <axis> zn ampl. too small	Zn amplitude of the motor encoder (ERN 1381) for the displayed axis is too low
Motor enc. <axis> z1 ampl. too small	Z1 amplitude of the motor encoder (ERN 1381) for the displayed axis is too low
Motor enc. <axis>: temperature too high	The temperature of the motor for the displayed axis is too high.
Motor encoder <axis> defective	The motor encoder of the displayed axis is defective.
Motor <axis>: speed is not equal to Imax	The current speed of the motor does not correspond with the expected speed at I-max. Perhaps the direction of rotation is incorrect.
Motor enc. <axis> frequency too high	The maximum permissible input frequency at the motor encoder was exceeded.
Motor <axis>: is not turning	The motor for the displayed axis is not turning.
Power stage <axis> not ready	The ready signal for the power stage was turned off during operation.
Axis <axis>: MP112/MP113 incorrect	MP112/MP113 (assignment of the speed encoder inputs) for the displayed axis is erroneous.

1 Update Information No. 8

1.1 NC Software

The following NC software has been released:

NC Software	Release		
280 474 03	04/98	Export version:	280 475 03

There is a new User's Manual for the NC software 280 474 xx.

Improvements:

- MP111 changed to MP111.0 and MP111.1:
  (see also section 1.2 ”New Functions for a Second Spindle”)
  MP111.0 Position encoder for the first spindle
  MP111.1 Position encoder for the second spindle
  Input: 0 = no position encoder
  1 to 6 = position encoder inputs X1 to X6
  35 to 38 = position encoder inputs X35 (only LE 430 M/9 axes) to X38
- MP112 is new:
  Assignment of speed encoder inputs to the axes
  Input: Connector designations (see table)
  MP112.0 to MP112.8 = axis 1 to axis 9

- Input of connector designations for MP110.x, MP112.x and MP120.x
  Only the following assignments of connector designations to axes are possible:

	Axis	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
LE 426 M	MP110.x	1 to 6					–	–	–	–
	MP112.x	15 to 19					–	–	–	–
	MP120.x	51 to 55					–	–	–	–
LE 430 M / 6 axes	MP110.x	1 to 6					–	–	–	–
	MP112.x	15 to 20						–	–	–
	MP120.x	51 to 56						–	–	–
LE 430 M / 9 axes	MP110.x	1 to 6 and 35 to 38								
	MP112.x	15 to 20						62 to 64		
	MP120.x	51 to 56						57 to 59		

- MP113.0 and MP113.1 are new:
  (see also section 1.2 ”New Functions for a Second Spindle”)
  MP113.0 Speed encoder input for the first spindle
  Input: 0 = no speed encoder
  20 = speed encoder input X20 (LE 426 M without spindle DSP)
  60 = speed encoder input X60 (LE 426 M with spindle DSP, LE 430 M)

  MP113.1 Speed encoder input for the second spindle
  Input: 0 = no speed encoder
  15 to 19 = speed encoder inputs X15 to X19
  20 = speed encoder inputs 20 (only le 430 m)
  62 = speed encoder inputs 62 (only le 430 m/9 axes)
  63 = speed encoder inputs 63 (only le 430 m/9 axes)
  64 = speed encoder inputs 64 (only LE 430 M/9 axes)
- MP121 changed in MP121.0 and MP121.1:
  Nominal speed output for the spindle
  (see also 1.2 ”New Functions for a Second Spindle”)
  MP121.0 Nominal speed output for the first spindle
  Input: 0 = no controlled spindle
  Analog signal at connector X8
  1 to 6 = Outputs 1 to 6
  Analog signal at connector X9
  7 to 13 = Outputs 7 to 13
  Digital nominal speed value
  56 = digital nominal speed value at X56 (LE 426 M without spindle DSP)
  61 = digital nominal speed value at X61 (LE 426 M with spindle DSP, LE 430 M)

  MP121.1 Nominal speed output for the second spindle
  Input: 0 = no controlled spindle
  Analog nominal speed signal on connector X8
  1 Outputs 1 to 6
  Analog nominal speed signal on connector X9
  7 Output 7
  Digital nominal speed signal
  51 to 55 = digital nominal values at X51 to X55
  56 = digital nominal value at X56 (only LE 430 M)
  57 = digital nominal value at X57 (only LE 430 M/9 axes)
  58 = digital nominal value at X58 (only LE 430 M/9 axes)
  59 = digital nominal value at X59 (only LE 430 M/9 axes)

- MP2180 is new:
  PWM frequency for the axes and the spindle (if without spindle DSP). If entry is 0 the standard PWM frequency of 5000 Hz is used.
  Input: 0 = PWM frequency 5000 [Hz]; standard setting for HEIDENHAIN inverters
  3000 to 7000 [Hz]
  The values between 1 Hz and 2999 Hz, and between 7001 Hz and 10 000 Hz cause a DSP error message (DSP-ERROR FF0A) after confirmation of the message POWER INTERRUPTED.
- MP2191 is new:
  Decelerating the spindle after an emergency stop
  In the event of an emergency stop command the spindle must be brought to a standstill as quickly as possible. If the braking energy cannot be dispelled quickly enough, the dc-link voltage increases sharply. In the worst case the inverter switches off and stops braking the spindle. With MP2191 you can define whether the maximum braking current is monitored during braking and, if required, should be reduced. The braking time is delayed somewhat by the monitoring.
  Input: %x
  Bit 0 = 0: Braking with monitoring of the maximum braking current (primarily for inverters with braking resistor)
  = 1: Braking with monitoring of the maximum braking current (primarily for inverters with regenerative braking)
- MP2510 and MP2511 have been expanded:
  Maximum input value has been reduced to 30 000 [A].
- MP2900, MP2910, MP2920, MP2930 have been added:
  reserved; input: 0
- MP4020 has been expanded:
  (see also 1.2 "New Functions for a Second Spindle")
  Bit 5 = 0: One-spindle mode
  = 1: Two-spindle mode
- MP6120, MP6350, MP6360, MP6520 have been expanded:
  Minimum inputs values have been reduced to 1 mm/min
- MP6540 has been expanded:
  Safety clearance to the stylus of the TT 120 during tool measurement
  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
- MP7262 has been added:
  Maximum index number for indexed tools (e.g. shoulder mills)
  Input: 0 to 9
- MP7683 has been expanded:
  Bit 3: If bit 3 and bit 2 are set, when the control reaches the end of a pallet table it begins again with the first line.
- MP13010 to MP13520 have been added:
  These machine parameters are for the second spindle. They are analog to the machine parameters MP3010 to MP3520 for the first spindle (see also section 1.2 "New Functions for a Second Spindle").
- FN18: SYSREAD has been expanded:
  It is now possible to transfer data on certain tools from the tool table.
  ID50
  NRxx System data
  IDXxxxx Tool number (0 = current tool)

- FN18: SYSREAD has been expanded:
  The current PLC datum shift can be read out.
  ID220
  NR4
  IDXx   Axis number (1 to 9 = X/Y/Z/A/B/C/U/V/W)
- Three markers for tool measurement have been added:

	Set	Reset
M4065 — All workpiece dimensions are OK.	NC	PLC
M4066 — Workpiece must be reworked.	NC	PLC
M4067 — Workpiece must be scrapped.	NC	PLC

- The number of strings was increased from 4 to 8 (now: S0 to S7).
- Timers T144 to T303 have been added:
  The new timers can be started only via Module 9006. The time is set immediately after module call and after expiration of the run time.
- W266 has been added:

	Set	Reset
Index number of a programmed indexed tool	NC	NC

- D604 has been added:
  Maximum possible spindle speed
  To ensure compatibility, D604 is preassigned with 99 999 999 after switching on the control or after an interruption of the PLC run.
- Module 9008 has been added:
  The current states of certain PLC I/O-board inputs are read into PLC addresses (see ”Mounting and Electrical Installation”). The addressed remain unchanged until they call either this module or Module 9002. The control recognizes whether a PLC I/O board is connected.

```
Call:
PS   B/W/D/K          <Number of the PL>
                      0:    First PLC input/output board
                      1:    Second PLC input/output board
                      2:    Third PLC input/output board
                      3:    Fourth PLC input/output board
PS   D/K              <Bit 0...31 = PL input 0...31>
PS   D/K              <Bit 0...31 = PL input 32...63>
CM 9008
Error recognition:    M4203 = 0:    Inputs were read
                      M4203 = 1:    Error code in W1022
                      W1022 = 2:    Invalid PL number or PL not connected
                      W1022 = 20:   Module was called from a SPAWN job or submit job.
```

- Module 9009 has been added:
  Certain outputs of a PLC input/output board can be set. The outputs are set or reset immediately at the time of module processing. The outputs remain in their states until they are reset by this module or Module 9005. The control recognizes whether a PLC input/output board is connected.

```
    Call:
PS   B/W/D/K          <number of the PL>
                      0:    First PLC input/output board
                      1:    Second PLC input/output board
                      2:    Third PLC input/output board
                      3:    Fourth PLC input/output board
PS   D/K              <Bit 0...31 = PL output 0...31>
CM 9009
Error flag:           M4203 = 0:    Outputs were set
                      M4203 = 1:    Error code in W1022
                      W1022 = 2:    Invalid PL number or PL not connected
                      W1022 = 20:   Module was called from a SPAWN job or submit job.
```

- Module 9088 has been added:
  M functions are shown in their own status window, which is called with a new soft key. The upper part of the window shows the M functions of the NC. The lower part shows the M functions that are activated by the PLC. These M functions are shown by means of Module 9088.

```
Call:
PS   B/W/D/K          <Number of the M function>
PS   B/WD/K           <Mode>
                      -1:   Delete all M functions
                      0:    Delete M function
                      1:    Display M function
CM 9088
Error recognition:    M4203 = 0:    M function displayed or deleted
                      M4203 = 1:    Error code in W1022
                      W1022 = 1:    Invalid M function number
                      W1022 = 2:    Invalid mode number
```

- Module 9091 has been added:
  Module 9091 can find the line number of a tool in the tool table. This is required for the use of indexed tools, because Modules 9092, 9093 and 9094 need the line number.

```
Call:
PS   B/W/D/K          <Tool number>
PS   B/WD/K           <Index number>
CM 9091
PL   B/W/D            <Line number>
Error recognition:    M4203 = 0:    line number were shown
                      M4203 = 1:    Error code in W1022
                      W1022 = 2:    Invalid value for tool number or index number
                      W1022 = 20:   Module was called from a SPAWN job or submit job.
                      W1022 = 29:   Tool table (TOOL.T) missing
                      W1022 = 30:   Tool number not found
                      W1022 = 32:   Index number not found
```

- Module 9145 has been added:
  It is now possible to automatically transfer the actual and nominal values for certain axes in any operating mode.

  Call:

PS B/W/D/K	<axes bit coded>		
CM 9145			
Error recognition:	M4203 = 0:	Automatic actual and nominal value transfer	
	M4203 = 1:	Error code in W1022	
	W1022 = 2:	Invalid axis number	
	W1022 = 24:	Missing M/S/T/T2/G strobe or M4176 = 1	

- Module 9175 has been added:
  With Module 9175 the first or second spindle can be activated (see also section 1.2 "New Functions for a Second Spindle").

  Call:

PS B/W/D/K	<Spindle number>	0 = first spindle
		1 = second spindle
CM 9175		
Error recognition:	M4203 = 0:	Given spindle was activated
	M4203 = 1:	Error code in W1022
	W1022 = 2:	Invalid spindle number
	W1022 = 20:	Module was called from a SUBMIT job.
	W1022 = 21:	Missing strobe or M4176 = 1

- Module 9202 has been expanded:
  With the new transfer value 3, Module 9202 can now activate a PLC window while a table editor is active.

  Call:

PS B/W/D/K	<display mode>	
	0: PLC soft key and window deselected	
	1: PLC soft key and window (small) selected	
	2: PLC soft key and window (large) selected	
	3: PLC soft key and window (large) selected while the table editor is active	
CM 9202		
Error recognition:	M4203 = 0:	No error
	M4203 = 1:	Transfer parameter is incorrect

1.2 New Functions for a Second Spindle

NC software 280 474 03 makes it possible to control a second spindle. Control of the second spindle can be digital (not on an LE 426 M without spindle DSP) or analog. The second spindle is controlled in place of an axis, which means that one available axis must be sacrificed to control a second spindle.

LE	Number of remaining axes
LE 426 M	4
LE 430 M / 6 axes	5 (four axes with position and speed encoders, one axis with speed encoder for position measurement)
LE 430 M / 9 axes	8

Electrical connection of spindles to the modular HEIDENHAIN inverter

The first spindle is connected as usual. Since the second spindle is operated instead of an axis, it must use the nominal speed command output or the speed and position encoder input of an axis.

If modular HEIDENHAIN inverters are used, they must be arranged in the order of decreasing rated current values. This means that the power module for the second spindle must be arranged next to the power module for the first spindle. This arrangement also ensures that the required PWM output for the second spindle is at the proper height, since the lower PWM connection is always used on the UM xxx modules for the spindle.

PWM output for the first spindle:	X61
PWM output for the second spindle:	X55 (LE 426 M with spindle DSP)
	X56 (LE 430 M/6 axes)
	X59 (LE 430 M/9 axes)

Installation configuration of modules:

UV 130	UM xxx for the first spindle	UM xxx for the second spindle	UM xxx	UM xxx	LE 426 M, LE 430 M

Electrical connection of the second spindle to the SIMODRIVE 611D

Because the spindle module of the SIMODRIVE 611D for the second spindle is equipped with a one-axis expansion card, the possibility exists of taking a PWM output from the upper row of the PWM connection on the LE. Due to the arrangement of the SIMODRIVE modules, only the right PWM output of the LE comes into question:

X53 (LE 426 M with spindle DSP, LE 430 M/6 axes)
X55 (LE 430 M/9 axes)
X55 (LE 426 M with spindle DSP)
X56 (LE 430 M/6 axes)
X59 (LE 430 M/9 axes)

Parameters for spindle operation

Only one spindle can be active at any given time. MP4020 bit 5 defines whether the control operates with one spindle (Bit 5 = 0) or two spindles (Bit 5 = 1). Module 9175 activates the desired spindle. Machine parameters MP111, MP113 and MP121 arrange the inputs and outputs for the spindles, with index 0 for the first spindle and index 1 for the second. The connector designations must be entered.

The parameters for the second spindle are similar to those for the first spindle (see Technical Manual). Since the second spindle is used instead of an axis, one axis must be defined as inactive in MP10 (=0). In the corresponding machine parameters MP110.x, MP112.x and MP120.x must also be set to zero. For a digital second spindle, the choice of which axis is to be deactivated depends on the PWM output being used. For an analog second spindle, any axis can be chosen. For the second axis, the machine parameters MP 13012 to MP 13520 are in effect. If the second axis is digital, the machine parameters MP2020.x to MP2800.x also are in effect , whereby the index *x* indicates the PWM output used for the second spindle:

Machine Parameter	Axis	MP10 Bit no.	PWM output of the second spindle	Position encoder input	Speed encoder input
MP2020.0 to MP2800.0	1	0	X51	X1 to X6 and X35 to X38	X15 to X19 (LE 426 M) X20 (LE 430 M)
MP2020.1 to MP2800.1	2	1	X52		
MP2020.2 to MP2800.2	3	2	X53		
MP2020.3 to MP2800.3	4	3	X54		
MP2020.4 to MP2800.4	5	4	X55		
MP2020.5 to MP2800.5	6	5	X56		
MP2020.6 to MP2800.6	7	6	X57		X62 to X64
MP2020.7 to MP2800.7	8	7	X58		
MP2020.8 to MP2800.8	9	8	X59		

Note: The PWM output in gray can be used for the second spindle if a modular HEIDENHAIN inverter is used.

The digital second spindle should be operated in place of the axis shown in the above table, because otherwise the same machine parameters set in MP2020.x to MP2800.x may apply both for the axis and the spindle. Axis-specific machine parameters that are not needed for the second spindle are set to zero. Because the second spindle is described by axis parameters, it may not have all the machine parameters that are available for the second spindle.

MP111.0 Position encoder for the first spindle

MP111.1 Position encoder for the second spindle
Input: 0 = no position encoder
1 to 6= position encoder inputs X1 to X6
35 to 38 = position encoder inputs X35 to X38 (only LE 430 M/9 axes)

MP113.0 Speed encoder for the first spindle
Input: 0 = no speed encoder
20 = speed encoder input X20 (LE 426 M without spindle DSP)
60 = speed encoder input X60 (LE 426 M with spindle DSP, LE 430 M)

MP113.1 Speed encoder input for the second spindle
Input: 0 = no speed encoder
15 to 19 = speed encoder inputs X15 to X19
20 = speed encoder input X20 (only LE 430 M)
62 = speed encoder input X62 (only LE 430 M/9 axes)
63 = speed encoder input X63 (only LE 430 M/9 axes)
64 = speed encoder input X64 (only LE 430 M/9 axes)

MP121.0 Nominal speed command output for the first spindle
Input: 0 = no controlled spindle
56 = digital nominal speed value at X56 (LE 426 M without spindle DSP)
61 = digital nominal speed value at X61 (LE 430 M and LE 426 M with spindle DSP)

MP121.1 Nominal speed command output for the second spindle
Input: 0 = no controlled spindle
51 to 55 = digital nominal speed value at X51 to X55
56 = digital nominal speed value at X56 (only LE 430 M)
57 = digital nominal speed value at X57 (only LE 430 M/9 axes)
58 = digital nominal speed value at X58 (only LE 430 M/9 axes)
59 = digital nominal speed value at X59 (only LE 430 M/9 axes)

MP4020 PLC compatibility
Input: %xxxxxx
Bit 5 = 0: One-spindle operation
1: two-spindle operation

Spindle activation (Module 9175)

Module 9175 activates the first or second spindle. For switching over by means of an M strobe, MP7440 bit 2 (program run stop for M functions) must not be set. For switching over by means of an S/G strobe, MP3030 (axis standstill during TOOL CALL in which only one spindle speed is output) must not be set.

Call:
PS B/W/D/K <Spindle number>
0 = first spindle
1 = second spindle
CM 9175
Error recognition: M4203 = 0: Indicated spindle is activated
M4203 = 1: Error code in W1022
W1022 = 2: Invalid spindle number
W1022 = 20: Module was called from a SUBMIT job.
W1022 = 21: Missing strobe or M4176 = 1

1 Update Information No. 7

1.1 NC software

The following NC software version has been released:

NC software	Release		
280 472 07	02/98	Export version:	280 473 07

Improvements:

- FN18: SYSREAD expanded:
  Certain tool data can be read from the tool tables.
  ID50
  NRxx   System datum
  IDXxxxx   Tool number (0 = current tool)
- FN18: SYSREAD expanded:
  The current datum shift can be read.
  ID220
  NR2
  IDXx   Axis number (1 to 9 = X/Y/Z/A/B/C/U/V/W)
- FN18: SYSREAD expanded:
  The current PLC datum shift can be read.
  ID220
  NR4
  IDXx   Axis number (1 to 9 = X/Y/Z/A/B/C/U/V/W)
- Module 9008 new:
  The current conditions of certain inputs on the PLC I/O unit are read to PLC addresses (see Mounting and Electrical Installation). These addresses remain unchanged until this module or module 9002 is called. It is not known however whether a PLC I/O unit is actually connected; with NC software 280 474 xx, this is known.

  Call:

```
PS   B/W/D/K        <Number of the PL>
                    0:    First PLC I/O unit
                    1:    Second PLC I/O unit
                    2:    Third PLC I/O unit
                    3:    Fourth PLC I/O unit
PS   D/K            <Bit 0...31 = PL input 0...31>
PS   D/K            <Bit 0...31 = PL input 32...63>
CM 9008
Error code:         M4203 = 0:    Inputs were read
                    M4203 = 1:    Error code in W1022
                    W1022 = 2:    PL number not valid
                    (280 474 xx: PL not connected or invalid PL number)
                    W1022 = 20:   Module was called in SPAWN or submit job.
```

- Module 9009 new:
  Certain outputs of a PLC I/O unit can be set. The outputs are set or reset at the same time as the module is being executed. The outputs remain in this state until they are set again with this module or module 9005. It is not known however whether a PLC I/O unit is actually connected; with NC software 280 474 xx, this is known.

```
Call:
PS   B/W/D/K       <Number of the PL>
                   0:    First PLC I/O unit
                   1:    First PLC I/O unit
                   2:    First PLC I/O unit
                   3:    First PLC I/O unit
PS   D/K           <Bit 0...31 = PL output 0...31>
CM 9009
Error code:        M4203 = 0:    Outputs were set
                   M4203 = 1:    Error code in W1022
                   W1022 = 2:    PL number not valid
                   (280 474 xx: PL not connected or invalid PL number)
                   W1022 = 20:   Module was called in SPAWN or submit job.
```

- Module 9145 new:
  An automatic actual and nominal value transfer can be carried out for certain axes in every operating mode.

```
Call:
PS   B/W/D/K       <Bit-coded axes>
CM 9145
Error code:        M4203 = 0:    Automatic actual and nominal value transfer
                   M4203 = 1:    Error code in W1022
                   W1022 = 2:    Axis number not valid
                   W1022 = 24:   M/S/T/T2/G strobe missing or M4176 = 1
```

- MP6120, MP6350, MP6360, MP6520 expanded:
  Minimum input value reduced to 1 mm/min
- MP7260 expanded:
  Max. input value reduced to 30 000
- MP7683 expanded:
  Bit 3: If bit 3 and bit 2 are set, machining begins again at the first line when the end of a pallet table is reached.
- MP6500 expanded:
  Bit12: 0 = a PLC datum shift is not taken into consideration during tool measurement.
  1 = a PLC datum shift is taken into consideration during tool measurement.
- MP2180 new:
  PWM frequency for the axes and the spindle (without spindle DSP). If 0 is input, the standard PWM frequency of 5 000 Hz will be used. The values between 1 Hz and
  2 999 Hz, and between 7 001 Hz and 10 000 Hz will lead to a DSP error message (DSP ERROR FF0A) after the message POWER INTERRUPTION has been acknowledged.
  See also 1.2 Current reduction with SIMODRIVE power stages at a high PWM frequency
  Input: 0. 3 000 to 10 000 [Hz]

The software **280 474..** has been released for the new modular control **TNC 426 M / TNC 430 M.** A separate Technical Manual will be printed for this control. Here is a preliminary list of the new machine parameters.

NC software	**First release**		
280 474 01	12/97	Export version:	280 475 01

- MP7500 expanded:
  An offset results is the spindle head is exchanged in the basic position.
  Bit 4: 0 = this offset is not compensated until M128, M114 or "Tilt working plane" is called.
  1 = this offset can be compensated via PLC datum shift.
  It is not compensated again if M128, M114 or "Tilt working plane" is called.
- MP7550.x new:
  Angular coordinates of the basic position of the tilting unit in the machine coordinate system. With input value 0 in MP7550.x, the description of the tilting unit from MP7510, MP7520, MP7530 will be based on the position 0 in the machine coordinate system.
  Input: –99 999,9999 to +99 999,9999
  MP7550.1 A-axis
  MP7550.2 B-axis
  MP7550.3 C-axes
- MP120.x, MP121 input values changed:
  Assignment of the nominal speed value outputs to the axes through indication of the connector and/or pin number.
  0 = no controlled axis/spindle
  1 = nominal analog value at X8/1
  2 = nominal analog value at X8/2
  3 = nominal analog value at X8/3
  4 = nominal analog value at X8/4
  5 = nominal analog value at X8/5
  6 = nominal analog value at X8/6
  7 = nominal analog value at X9/7
  8 = nominal analog value at X9/8
  9 = nominal analog value at X9/9
  10 = nominal analog value at X9/10
  11 = nominal analog value at X9/11
  12 = nominal analog value at X9/12
  13 = nominal analog value at X9/13
  51 = nominal digital value at X51
  52 = nominal digital value at X52
  53 = nominal digital value at X53
  54 = nominal digital value at X54
  55 = nominal digital value at X55
  56 = nominal digital value at X56
  57 = nominal digital value at X57
  58 = nominal digital value at X58
  59 = nominal digital value at X59
  61 = nominal digital value at X61

- MP115 new:

MP115.0: Encoder input 1 V_{PP} or 11 µA

Input: %xxxxxxxxxx

Bit	0	Input	X1	0 = 1 V_{PP}
	1		X2	1 = 11 µA
	2		X3	
	3		X4	
	4		X5	
	5		X6	
	6		X35	
	7		X36	
	8		X37	
	9		X38	

MP115.1: *Reserved*

Enter %0000000000

MP115.2: Low or high input frequency

Recommended input value for linear encoders: 50 kHz

Input: %xxxxxxxxxx

Bit	0	Input	X1	1 V_{PP}:	0 = 50 kHz
	1		X2		1 = 350 kHz
	2		X3	11 µA:	0 = 50 kHz
	3		X4		1 = 150 kHz
	4		X5		
	5		X6		
	6		X35		
	7		X36		
	8		X37		
	9		X38		

NC software	**Release**		
280 474 02	01/98	Export version:	280 475 02

No improvements

1.2 Current reduction in SIMODRIVE power stages at high PWM frequency

The HEIDENHAIN controls TNC 426 PB / TNC 430 work with a PWM frequency of 5kHz.

SIEMENS power stages are usually operated at a PWM frequency of 3.2kHz (spindle) and 4kHz (axes). The rated currents I_N are defined for these frequencies.
If these power stages are operated at a higher PWM frequency (5kHz), the modules can, in certain cases, overheat (in particular 6SN1123-1AA00-0CA0 and 6SN1123-1AB00-0CA0 as axis modules). In machines where the modules are not used to their full capacity, there is no danger of overheating.

There are two ways of preventing this undesired heating:
– Reducing the PWM frequency
– Reducing the factor for I^2t monitoring or reducing the rated current I_N of the power stages
For spindles with their own DSP, the only possibility is to reduce the factor for I^2t monitoring or to reduce the rated current.

Reducing the PWM frequency
Using the new machine parameter MP2180, it is possible to set the PWM frequency for the axes and the spindle together.
The MP2180 has no effect on a spindle with its own DSP.

Input: 0 and 3 000 to 7 000 [Hz]
0 = PWM frequency 5000 [Hz]
3000 [Hz] to 7000 [Hz] PWM frequency for SIMODRIVE power stages
The values between 1 Hz and 2 999 Hz, and between 7 001 Hz and 10 000 Hz will lead to a DSP error message (DSP ERROR FF0A) after the message POWER INTERRUPTION has been acknowledged.

Note:
Reduction of the PWM frequency has no effect on the max. speed, but does require the axes and the spindle to be recommissioned!
HEIDENHAIN recommends setting a suitable PWM frequency for axis modules (usually 4kHz, see Siemens documentation) when commissioning new machines. If the spindle module continues to show signs of excess heating despite the reduction in the PWM frequency from 5 kHz to 4 kHz, then you need to reduce the reference value for I^2t monitoring (MP2302.x, MP2303) or the rated current instead.

Reducing the reference value for I^2t monitoring or reducing the rated current

The reduction of the rated current of the power stages as well as the reference value for I^2t monitoring can be calculated using two values (X1, X2), which can be taken from the SIEMENS documentation.

X1 = current reduction factor in % at 8 kHz PWM frequency;
X2 = PWM frequency in kHz as of which the current reduction becomes valid;
f_{PWM} = with PWM frequency in kHz as set in MP2180;

The percentage reduction of the rated current can be calculated using the following formula:

$$X_R\,[\%] = 100 - \left(\frac{(100 - X1) * (8kHz - f_{PWM})}{8\ kHz - X2} + X1\right)$$

From this we get the following reference value for I^2t monitoring:

$$X_B = 1 - \frac{X_R[\%]}{100};$$

Note:
The reduction of the rated current of the power stage can cause a reduction in the rated torque and the rated power of the motor, if the rated current of the power stage and the rated current of the motor had been set the same.

Examples for LT module 50A:

Axis LT module 50A, PWM frequency 5kHz, X1=40%, X2=4kHz

$$X_R\,[\%] = 100 - \left(\frac{(100 - 40) * (8kHz - 5kHz)}{8\ kHz - 4kHz} + 40\right) = 15\% \qquad X_B=0.85$$

Spindle LT module 50A, PWM frequency 5kHz, X1=40%, X2=3.2kHz

$$X_R\,[\%] = 100 - \left(\frac{(100 - 40) * (8kHz - 5kHz)}{8\ kHz - 3.2kHz} + 40\right) = 22.5\% \qquad X_B=0.78$$

Axis LT module 50A, PWM frequency 4kHz, X1=40%, X2=4kHz

$$X_R\,[\%] = 100 - \left(\frac{(100 - 40) * (8kHz - 4kHz)}{8\ kHz - 4kHz} + 40\right) = 0\% \qquad X_B=1.00$$

Spindle LT module 50A, PWM frequency 4kHz, X1=40%, X2=3.2kHz

$$X_R\,[\%] = 100 - \left(\frac{(100 - 40) * (8kHz - 4kHz)}{8\ kHz - 3.2kHz} + 40\right) = 10\% \qquad X_B=0.90$$

If you wish you can reduce the rated currents I_N of your power stages in the power stage table motor.amp, using the calculated reduction values, or lower the reference value for I^2t monitoring. The edited motor.amp table will automatically be stored in the PLC partition.

1.3 DSP error messages

DSP error messages with error code

Error message		Explanation	Classification
DSP ERROR	**XXXX**		
	1000 0	Timeout during a command	Data processing error
	1001 0	Incorrect acknowledgement of a command	Data processing error
	1002 0	Command is sent before preceding command is acknowledged	Data processing error
	1003 0	Synchronization error between DSP and NC	Data processing error
	1004 0	Incorrect message DSP Ω NC	Data processing error
	1005 0	Too many commands NC Ω DSP	Data processing error
	1100 0	Error determining check sum	Data processing error
	1101 0	Timeout during Word transfer command (load DSP code)	Data processing error
	1102 0	Timeout during check sum	Data processing error
	F010	Motor type unknown (MP2200)	Error in motor table or in MP2200
	F020	*Reserved*	
	F030	*Reserved*	
	F040	No. pole pairs too high	Error in motor table or in MP2230
	F090	Drive type unknown (MP2000)	
	F0A0	*Reserved*	
	F0B0	*Reserved*	
	F0C0	*Reserved*	
	F0F0	Current controller proportional factor too high	
	F100	Current controller integral factor too high	
	F110	Motor temperature	
	F120	*Reserved*	
	F130	Osci parameter is incorrect (for test)	Data processing error
	F140	Rated current power stage	Error in power stage table
	F220	*Reserved*	
	F240	Unknown counter-IC type at speed input	Hardware error
	F260	*Reserved*	
	F290	Error 3-D touch probe/evaluation, there was no latch with L1 input (G19/G26)	Hardware error
	F2C0	Actual motor current above limit value	Drive error
	FF01	Undefined error, no casual connection	Data processing error
	FF02	Host command not recognized, not valid	Data processing error
	FF03	Host-/DSP watchdogs do not agree	Data processing error

Error message		Explanation	Classification
DSP ERROR	**XXXX**		
	FF04	Undefined interrupt	Data processing error
	FF05	Hardware not recognized	Hardware and software do not match
	FF06	No V_NOML value received from the host	Data processing error
	FF07	AC-fail	Power stage error
	FF08	Emergency stop fail	
	FF09	Stack overflow	Data processing error
	FF0A	Pulse width modulation delta signal	Hardware error or incorrect value in MP2180
	FF0B	Error in store request	Data processing error
	FF0C	No speed control interrupt	Data processing error
	FF0D	Error in sum check (code)	Data processing error
	FF0E	Time exceeded in the speed interrupt	Data processing error
	FF0F	Error initializing a software timer	Data processing error
	FF10	Error in LSV2 transfer	Data processing error

DSP error messages with text

Error message	Explanation
Power stage in axis <axis> too weak	Power stage is too weak for the axis displayed
<Axis> motor enc. line count too high	Line count of motor encoder is too high for the axis displayed
Motor <axis>: Xh; X2; f-n; R2 incorrect	One of the datums in the following motor data for the axis displayed is incorrect: Xh: Magnetizing reactance X2: Rotor leakage reactance f-n: Rated frequency R2: Rotor resistance
Motor <axis>: n-n; f-n incorrect	One of the datums in the following motor data is incorrect: n-n: Rated speed f-n: Rated frequency
Power stage <axis>: U-Imax incorrect	U-Imax of the power stage for the displayed axis is erroneous (U-Imax = voltage of the current sensor)
Power stage <axis>: I-max incorrect	Imax of the power stage for the displayed axis is erroneous (Imax = peak current)
Motor <axis>: t-max incorrect	T-max of the motor for the displayed axis is erroneous
Motor <axis>: I-n incorrect	I-N of the motor for the displayed axis is erroneous
Motor <axis>: I-max incorrect	I-max of the motor for the displayed axis is erroneous
Motor <axis>: n-max incorrect	n-max of the motor for the displayed axis is erroneous
Axis <axis>: MP2340/MP2350 incorrect	MP2340/MP2350 (field angle displacement for the displayed axis) incorrect
Axis <axis>: MP2190 incorrect	MP2190 (dc-link voltage) for displayed axis incorrect

Error message	Explanation
Axis <axis>: MP120/MP121 incorrect	MP120/MP121 (assignment of the nominal speed value outputs) for the displayed axis is erroneous
Axis <axis>: MP2540/MP2550 incorrect	MP2540/MP2550 (band-rejection filter dampening) for the displayed axis is erroneous
<Axis> motor enc. zn ampl. too small	Zn amplitude of the motor encoder (ERN 1381) for the displayed axis is too small
<Axis> motor enc. z1 ampl. too small	Z1 amplitude of the motor encoder (ERN 1381) for the displayed axis is too small
Motor encoder <axis>: temperature too high	The temperature of the motor for the displayed axis is too high
<Axis> motor encoder defective	The motor encoder for the displayed axis is defective.
Motor <axis>: speed not equal to Imax	The current speed of the motor does not correspond with the expected I-max speed. Perhaps direction of rotation incorrect.
<Axis> motor encoder freq. too high	On the motor encoder the max. permissible input frequency was exceeded.
Motor <axis>: is not turning	The motor for the displayed axis is not turning.
Power supply unit <axis> not ready	The standby signal for the power stage was switched off during operation.