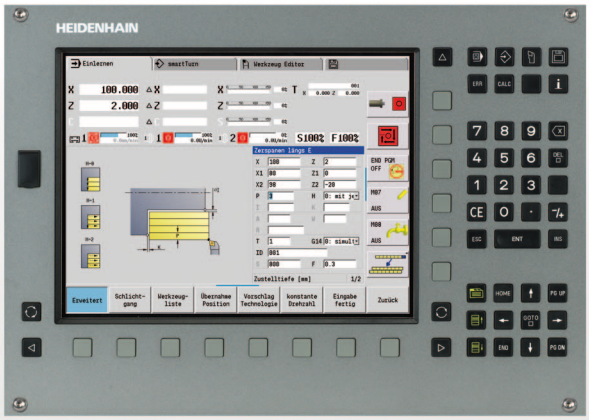
HEIDENHAIN



HSCI

Technical Manual

MANUALplus 620

NC Software
548 328-03

November 2010

1 Update Information No. 1

1 Update Information No. 2

1 Update Information No. 3

2 Introduction

3 Mounting and Electrical Installation

4 Machine Parameters

5 Modules and PLC Operands

6 Configuring the Axes and Spindle

7 Machine Interfacing

8 PLC Programming

9 Data Interfaces

10 Index

1 Update Information No. 1

1.1 Overview

1.2 NC Software 548 328-02

1.2.1 Important notes on updating software

Please remember the following important information when updating the software versions listed below:

- **MANUALplus 620: 548 328-01 to 548 328-02**

Note

If you are using linear encoders with EnDat interface or motor encoders with EnDat interface for position measurement on your machine, you must carry out the following step.

- **Move the EnDat axes to known positions before the update:**
  A function for monitoring the SRAM contents for consistency is introduced with the new NC software. After the software update, all EnDat axes will therefore display the error message **S-RAM contents of axis are invalid**. At the same time, the control will display a dialog box for confirmation, in which the current (the displayed) position of the axis is compared with the physical (switch-off) position.

▶ Move the axes to known positions before the update.

▶ Write down the switch-off positions.

▶ After the control has booted with the new software for the first time, confirm the positions of the EnDat axes.

Note

HEIDENHAIN recommends:

Making a backup of the control (e.g. with TNCbackup), before updating the NC software.

Saving your current machine configuration. The configuration editor (DATA BACKUP soft key) can be used for this purpose.

If you later want to undo the software update and return to the previous software version, you need the saved configuration data of the old version!

Note

Please perform the update of the NC software as described in the Technical Manual in Chapter 2 "NC Software Exchange".

Note

Be sure to remember the important information about the software update, which is provided on the following pages.

■ **Checking and saving new machine parameters:**
After having installed the new NC software and rebooted the control, you must check and confirm the new machine parameters. The code number dialog box appears on the screen:

- Enter the MP code number **95148** and confirm your entry with the ENT key.
- Press the **UPDATE RULES** soft key.
  Check the listed update rules. Each entry in the list stands for a new parameter that was added to the system by the update.
- Exit the **UPDATE RULES** with the **END** soft key.
- Press the **CONFIG DATA** soft key.
  Before the configuration editor opens, an informational window is displayed, reporting the removal of the CfgRestorePosition machine parameter. Press the **NEXT** soft key.
- All new machine parameters are marked with a red exclamation point in the configuration editor. The control indicates if certain machine parameters are faulty. Please ignore these messages for the time being.
- **Important step:** Press the **SAVE** soft key.
- The **Configuration data changed** dialog box opens. Press the **SAVE** soft key again. The new machine parameters are now automatically saved in the *.cfg files.
- Press the **END** soft key and exit the **Machine Parameter** operating mode by pressing the **END** soft key again.

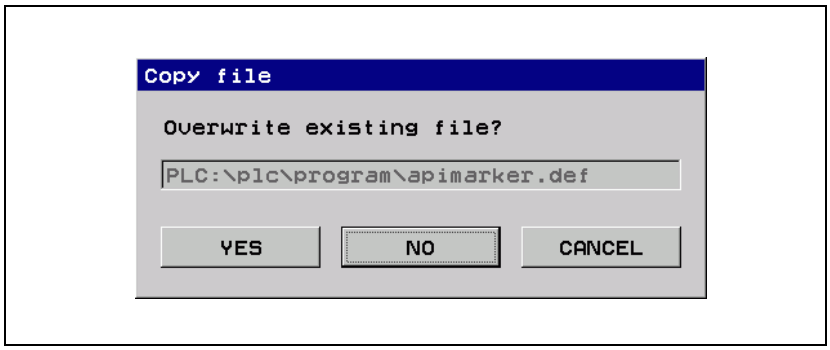
The control then continues booting.

Note

The definition file of the symbolic programming interface API 3.0 has been expanded. You must perform the step described below in order for the PLC program to be compiled successfully after the update.

■ **Replace the apimarker.def file:**
During the update of the NC software, a new version of the **apimarker.def** file was automatically copied to the PLC partition of the control. Proceed as follows:

- Switch to the **Organization** mode of operation.
- Enter the code number 95148 to call the **Machine Parameter** mode of operation.
- Press the **END** soft key and switch the soft-key row.
- Press the **PGMMGT** PGMMGT soft key to open the file manager.
- Switch to the PLC:\proto\plc directory.
- Copy the **apimarker.def** file to the program directory of your PLC program. Overwrite the existing **apimarker.def** file:



Note

Please also copy the **apimarker.def** file to your PC as well, and add it to the PLCdesignNT project. Otherwise, during the next transfer of PLC project files to the control, the file might be overwritten by the old version.

Note

After an update, please modify the previous file **oem_turning.mcg** as described below, and add it to the PLCdesignNT project. Otherwise, during the next transfer of PLC project files to the control, the file **oem_turning.mcg** might be overwritten by the old version, which leads to an error.

- **Modify the max. number of spindles in the file oem.mcg or oem_turning.mcg:**
  To modify the max. number of spindles in the PLC file **oem.mcg or oem_turning.mcg**, proceed as follows:

```
...

DEFINE SPINDLE_COUNT       = 6            ;  (old: =4)

...
```

Note

After an update, please modify the previous file **plc.cfg** as described below, and add it to the PLCdesignNT project. Otherwise, during the next transfer of PLC project files to the control, the file **plc.cfg** might be overwritten by the old version, which leads to an error (Fatal Error Syntax).

■ **Modify the plc.cfg file:**
The current plc.cfg file is located in the control in the directory PLC:\config\lathe\manplus\plc.cfg. You can use TNCremoNT to copy the file from the control to the PLC project, or you can use PLCdesignNT to modify the previous file in the PLC project.

▶ Make the following changes to the plc.cfg file:

```
CfgPlcOverrideDev (
  key:="PotentiometerF",
  source:=OVR1,
  mop:="MB",                ; This line must be added.
  mode:=LINEAR,
  values:=[]
)

CfgPlcOverrideDev (
  key:="PotentiometerS",
  source:=OVR2,
  mop:="MB",                ; This line must be added.
  mode:=LINEAR,
  values:=[]
)

CfgPlcStrobes (
...
sStrobe:=[                   ; Add an opening bracket
"S1"
],                           ; Add a closing bracket
...
)
```

```
CfgPlcOverrideS (
  key:="S1",
  minimal:=0.5,
  maximal:=1.5,
  source:= [                          ; Add an opening bracket
    "PotentiometerS"
  ]                                   ; Add a closing bracket
)
CfgPlcOverrideS (
  key:="S2",
  minimal:=1,
  maximal:=1,
  source:= [              ; Add an opening bracket
    "PotentiometerS"
  ]                       ; Add a closing bracket
)
CfgPlcOverrideF (
  key:="CH_NC1",
  minimal:=0,
  maximal:=1.5,
  source:= [              ; Add an opening bracket
    "PotentiometerF"
  ]                       ; Add a closing bracket
)

CfgPlcMop (                           ; Add this and all the following data
  key:="MB",
  type:=MB,
  primary:=FALSE,
  omg:=0,
  spindle:=0
)
CfgPlcMop (
  key:="HR",
  type:=HR,
  primary:=FALSE,
  omg:=0,
  spindle:=0
)
```

Note

Please observe the following note if you
are using the HEIDENHAIN PLC Basic Program!

- **It is essential that you check and modify the PLC program:**
  The behavior of the symbolic API marker **NN_ChnProgCancel** (NC program cancelation) has been changed: **NN_ChnProgCancel** will now be set every time the NC program is canceled. For a normal end of program, **NN_ChnProgEnd** will be set. The **NN_ChnProgCancel** marker remains set during the complete Cancel cycle and beyond the program end until the next NC program is started.

  When a program is canceled, the **NN_ChnProgEnd** marker will not be set. The end of program run, including the execution of a Cancel cycle, has been reached when **NN_ChnControlInOperation** is reset. **NN_ChnProgCancel** and **NN_ChnProgEnd** will be reset when **NN_ChnControlInOperation** is set again.

  If both **NN_ChnProgCancel** and **NN_ChnControlInOperation** are set, this indicates that the Cancel cycle is being executed.

  Module 9429 or 9320 can be used to inquire the reason for the program cancelation.

Please check the following lines in the PLC basic program and modify them if required:

German: Biblioth.src

```
;External/Internal STOP
L            ApiChn.NN_ChnProgCancel
AN           ML_Internal_STOP
=            MG_Impuls_Internal_STOP

L            ApiChn.NN_ChnProgCancel
=            ML_Internal_STOP
```

English: Library.src

```
;External/Internal STOP
L            ApiChn.NN_ChnProgCancel
AN           ML_Internal_STOP
=            MG_pulse_internal_stop

L            ApiChn.NN_ChnProgCancel
=            ML_Internal_STOP
```

1.2.2 Description of the new functions

New software options

You can enable the following new software options by entering a code number. HEIDENHAIN can give you the code number after having been informed of the SIK number:

Option	Description	ID
#10	**Tools and technology** ■ Tool database expanded to 999 entries ■ Technology database expanded to 62 workpiece-material/tool-material combinations ■ Support of multipoint tools ■ Tool life monitoring with exchange tools	632 228-01
#17	**Tool measurement** ■ Determining tool-setting dimensions with a touch probe ■ Determining tool-setting dimensions with an optical gauge	632 230-01
#41	**Additional Language** Enabling of additional conversational languages. The languages listed below can be ordered. Please contact HEIDENHAIN if you require additional conversational languages.	
	Slovenian	530 184-01
	Slovak	530 184-02
	Latvian	530 184-03
	Norwegian	530 184-04
	Korean	530 184-06
	Estonian	530 184-07
	Turkish	530 184-08
	Romanian	530 184-09
	Lithuanian	530 184-10
#42	**DXF import** ■ Loading of DXF contours	632 231-01
#70	**Y-axis machining** ■ Y-axis machining	661 881-01
#94	**W-axis machining** ■ W-axis support	679 676-01

Overview of the improvements

A summary of the improvements in NC software 548 328-02 is given below. For more detailed information, please refer to the Technical Manual sections indicated in the brief descriptions given below.

Machine Configuration

- **New: Search for the iTNC MP number in the configuration editor**
  For numerous machine parameters, the compatible iTNC MP number is stored in the help text in the configuration editor. Up to now, it has not been possible to search the configuration for these numbers. The search function of the configuration editor now enables you to search for the iTNC MP number (selection: MP number). See "Finding / Replacing" on page 257.
- **New: Separate parameter numbers for OEM parameters**
  In order for the OEM to group the parameters in the configuration editor according to his own needs, or to make them easier to find, there is now a separate number range for the OEM. Numbers 900000 to 999999 are reserved for the OEM. For those parameters for which an OEM number is defined, this number is shown instead of the HEIDENHAIN number. The numbers are to be defined in the **`%OEM%\config\layout\PlcUniqueNumbers.xml`** file. If the file is missing or empty, no OEM numbers will be displayed. For more information, please refer to "User Parameters" on page 277.
- **Expanded: OEM motor table (only digital control)**
  Until now, if the OEM-specific motor table (path: **`PLC:\table\motor_oem.mot`**) was missing, a warning was issued and the user had to create the table himself. Now no warning will be issued. The control itself creates a new blank table as soon as the **MP_motName** (401301) parameter is edited. If the OEM motor table exists but columns are missing, then the columns that exist in the SYS motor table will be transferred to the OEM motor table during copying. The control indicates which columns were not copied.
- **Expanded: Saving update rules**
  If the user tries to exit the configuration editor without saving the changes made by the update rules, a dialog window appears prompting the user to save the data. The configuration editor cannot be exited until the data are saved, see "Update rules" on page 265.

Note

The first restart of the control after the update cannot be continued without saving the configuration changes made by the update rules.

Expanded/changed machine parameters:

MP number	Config object	Parameter	Description of change
100201	**CfgMachineSimul**	**MP_simMode**	■ When the new value **Delivery** is set, during startup of the control all axes are set to the test mode, and a switch-on of the axes is prevented. The user should then be able to start the control, even with an incomplete or faulty axis configuration in order to put the axes into operation. After the configuration of all axes has been completed, the control can be switched to full operation (FullOperation). ■ If **Delivery**, **CcAndExt** or **CcOnly** is set, the control now no longer outputs any analog nominal values.
102902	**CfgFileType**	**MP_standardEditor**	"TEXT-EDITOR" can now be selected in order to assign the ASCII editor of the control to a file type.
103502	**CfgPlcTimer**	**MP_value**	The input range of the parameter was expanded from 1 000 to 1 000 000 seconds (corresponds to approx. 11.5 days).
104300 **104400**	**CfgPlcOverrideDev** **CfgPlcOverrideS**	–	The reaction of the parameters under CfgPlcOverrideDev, CfgPlcOverrideS, CfgPlcOverrideF and CfgPlcOverrideR was changed from NOTHING to RESET. The control must now be rebooted after a parameter change.
105201	**CfgSystemTime**	**MP_offsetToUTC**	Values with decimal places can now be entered for time differences to universal time (GMT). This is necessary for parts of Australia (+8.5 and +9.5 hours) and Kazakhstan (+3.5 hours), for example. Also, the maximum value was extended from +13 to +14 [hours].
400011	**CfgAxisHardware**	**MP_posEncoder Resistor**	The default value of the parameter was changed from **`without`** to **`120 ohm`**.

New machine parameters:

MP number	Config object	Parameter	Description
102907	**CfgFileType**	**MP_protect**	Disables filtering or editing of a file type, see page 1192.
104018	**CfgPlcSStrobe**	**MP_cuttingSpeed**	Optional parameter – If parts of the configuration indicate the symbolic name or number of a word marker to which the cutting speed is copied.
104304	**CfgPlcOverrideDev**	**MP_mop**	In **MP_mop**, enter the key name of the machine operating panel on which the override source is located, see page 986.
106501	**CfgConfigSettings**	**MP_undoListSize**	Defines the number of entries in the parameter change list, see page 257.
106502		**MP_suppressUsrMsg**	This parameter is used to suppress the warning **`Key is non-functional`**, see page 930.
106503		**MP_dispParam Numbers**	Specifies whether MP numbers or symbolic names are displayed in the parameter change list, see page 257.
106504		**MP_hideWrite Protected**	If the parameter is set to TRUE, write-protected parameters are hidden in the configuration editor, see page 264.
116103	**CfgPlcSymName**	**MP_dbLoadDisplay**	The parameter defines the variable name for the dashboard load display, see page 1036.
203804	**CfgChannelProperties**	**MP_kinManualMode**	Y axis as oblique axis: Activate the compensating motion in Manual mode as well, see page 1071.
300110	**CfgAxis**	**MP_deactivatedAtStart**	Deactivate the axis or spindle during start-up, see page 448.
300111		**MP_restoreModuloCntr**	Save modulo counter of the axis in SRAM, see page 403.
300205	**CfgAxisPropKin**	**MP_parAxComp**	Define the compensation for parallel minor axes, see page 1079.
401509	**CfgSpindle**	**MP_changeTurnDir**	Rotational direction reversal with M3 and M4, see page 435.
601801	**CfgGlobalProperties**	**MP_lifeTime**	Activate tool life monitoring for tool service age or workpiece quantity, see page 1038.
601806	**CfgGlobalProperties**	**MP_doProgAfterTCall**	Run subprogram after the tool change, see page 1072.

MP number	Config object	Parameter	Description
604601	**CfgToolMeasuring**	**MP_measuringType**	Type of tool measurement, see page 1018.
604602		**MP_feed**	Tool measurement: Measuring feed rate, see page 1018.
604603		**MP_distance**	Tool measurement: Measuring range, see page 1018.
604701	**CfgProbePosition**	**MP_positionProbePos**	Tool measurement: Position of the touch probe in positive axis direction, see page 1019.
604702		**MP_positionProbeNeg**	Tool measurement: Position of the touch probe in negative axis direction, see page 1019.
604703		**MP_maxMeasuringFeed**	Tool measurement: Maximum permissible measuring feed rate, see page 1019.
604801	**CfgGlbDispSettings**	**MP_plcSpindleSelect**	Selection of spindle number by PLC, see page 1039.

Operation and technology

- **Enhanced: Dashboard display of PLC signals**
  The attribute evaluation of the dashboard element "LoadDisplay" was enhanced so that PLC data can now optionally also be transferred to this element. Therefore, it is now also possible for analog control to realize a load display for axis and spindle drives directly in the dashboard, see "Load display for analog drives" on page 1036.
- **Enhanced: Update of NC software**
  The NC software can now also be updated while it is running. The new code number 231019 was therefore introduced, see "Start update while software is running on the control" on page 98.
- **Enhanced: Log**
  The display of the control's log was improved. The accumulated keystrokes are now stored simultaneously with the control events in the log and are displayed in table view.
  In order to be able to track machine operation or machine conditions systematically, detailed additional information is entered and stored simultaneously with all important log entries, such as keystrokes, errors, system errors or warnings, see "Error messages and log files" on page 907.
- **Enhanced: Display of configuration errors during start-up**
  If configuration errors occur during control start-up, the **`Error during start-up`** message appears instead of **`Power interrupted`**. Also, the error messages for the incorrect configuration data, which are triggered by the applications, are displayed individually.

Machine interfacing

- **New: Additional data types for table columns**
  The control supports additional data types for columns in NC tables. The FEED_CUT column data type applies to the cutting speed in units of m/min or feet/min. The FEED_ROT column data type applies to the feed rate per revolution in units of mm/rev or inch/rev. As usual, the new column data types are listed as enumeration values of parameter **MP_unit** (105602) of the configuration object **CfgColumnDescription**.
- **Enhanced: Integrated oscilloscope – Selection of symbolic operands**
  In the MIOTC dialog (dialog box for selecting markers, inputs, outputs, timers and counters) in the integrated oscilloscope, symbolic API operands can now also be selected and displayed conveniently in a list. See "Setup for digital signals" on page 806.
- **New: Integrated oscilloscope – Circular interpolation test**
  A circular interpolation test can now be performed with the integrated oscilloscope. See "Circular interpolation test with the integrated oscilloscope" on page 814.
- **New: Test of internal EMERGENCY STOP by code number**
  For test purposes, the behavior during an internal EMERGENCY STOP can now be simulated in order to inspect the correct wiring of the machine. The control-is-ready output is reset, and the NC and PLC are no longer operable. It is essential that you support hanging axes before the test in order to prevent damage to the machine in case of error. To start the test, press the **`CODE NUMBER`** soft key and enter the code number **`6871232`**. Enter the code number again to reset the control status to "ready for operation."

Configuring the Axes and Spindle

- **New: Software option #70 – Y-axis machining**
  With a Y axis you can drill and mill a workpiece on its front, back and lateral surfaces.
  During use of the Y-axis, two axes interpolate linearly or circularly in the given working plane, while the third axis interpolates linearly. This enables you to machine slots or pockets, for example, with plane floors and perpendicular edges. By defining the spindle angle, you can determine the position of the milling contour on the workpiece.
  If the Y axis is to be positioned at an angle not equal to 90° to the X or Z axis, the Y axis can also be configured as an oblique axis.
  For configuring the Y axis, please refer to "Configuring the Y axis" on page 1066. For information on programming the Y axis, please refer to the User's Manual for the control.
- **New: Software option #94 – W-axis support**
  The control can now offset the display of movements in the Z axis with those of its parallel secondary axis W. The W axis is already configured in the control kinematics and can be moved via the PLC.
  For more information on the configuration of the W axis, please refer to "Configuring the W axis" on page 1073.
- **New: Spindle change key**
  Starting immediately, a spindle change key can be supported by the PLC. It assigns the input (TSF dialog) to the selected spindle. The selected spindle is identified in the corresponding display element of the dashboard. See "Selection of spindle and channel by PLC" on page 1039.
- **Enhanced: Backlash compensation**
  If nonlinear axis-error compensation is active, **MP_backLash** can now be used to activate backlash compensation in addition to the compensation-value tables, see "Axis Error Compensation" on page 467.

PLC programming

- **Enhanced: Symbolic memory interface (API 3.0) – New operands introduced:**

PLC operand / Description	Model
NP_ChnProgSelected NC program selected This marker can be used to interrogate whether an NC program is selected in the Program Run modes of operation. The marker is not set if an NC program is selected from a pallet table.	M
NN_ChnFeedRapidTraverseActive 0: Rapid traverse is active (FMAX) 1: Rapid traverse is not active	M

- **Changed: Behavior of NN_ChnToolLifeExpired (tool life 1 expired)**
  The status of **NN_ChnToolLifeExpired** is now reset by the PLC runtime system after the end of an NC program.
- **New: PLC process monitor**
  In the PLC programming mode you can use the MONITOR and PROCESS MONITOR soft keys to open a status screen in which the control displays all parallel processes, as well as the process for the submit queue. See "Control of events" on page 1328.

■ **Enhanced: Compilation of PLC program**

- If the PLC program has already been compiled in the **PLC Programming** mode of operation before the **Power interrupted** message has been acknowledged, the PLC program will not be compiled again when the message is acknowledged. This change makes it possible to observe the PLC operands with the integrated oscilloscope during start-up of the PLC program:
  - Start the control, do not acknowledge the **Power interrupted** message.
  - Compile the PLC program in the PLC Programming mode of operation.
  - Activate the integrated oscilloscope in order to observe the desired PLC operands and start the measurement.
  - Acknowledge the Power interrupted message now for the control to start the PLC program.
- Now a PLC program is compiled even if the machine parameters refer to symbolic names of PLC operands that are not defined in the PLC program. The control issues an error message for every undefined symbolic name and then compiles the PLC program.

Attention

A PLC program with undefined symbolic PLC operands can lead to hazardous behavior of the machine! It is essential that you check whether the parameters for configuring the M functions (CfgPlcMStrobe) contain meaningful strobe definitions. The data of strobe definitions must be mapped onto defined PLC operands.

■ **Enhanced: Commands for string processing**
Symbolic operands (B/W/D operands) can now be used for indexed access to the string operands "S" or the PLC error and dialog files, see "Commands for String Processing" on page 1313.

■ **Enhanced: WATCH LIST and TRACE function**
New TYPE column: Type (M for marker, B for byte, W for word, etc.) of the PLC operand, see "The WATCH LIST function" on page 1100.
The **ADD TO WATCH LIST** soft key can be used to transfer the PLC operands of the currently highlighted line to the WATCH LIST, see "The TRACE function" on page 1104.

■ **Enhanced: EDIT function**
The features and the operation of the editor were changed. The editor is now a full-fledged ASCII editor. The cursor can be positioned in all directions and line breaks can be inserted, see "The EDIT function" on page 1107.

■ **New: Moving PLC axes with the handwheel**
It is now possible to assign a PLC axis to a handwheel via Module 9036. In the **El. Handwheel** mode of operation, the machine operator can use the axis keys to select a PLC axis and move it with the handwheel.

PLC modules modified/enhanced

- **Changed: Module 9145 (Actual-to-Nominal Value Transfer)**
  A call of PLC API Module 9145 for actual-to-nominal value transfer is now synchronized with other positioning commands. The transfer cannot be started while another positioning command is pending.
  A PLC positioning movement cannot be started while the transfer is running. During the transfer, NC program execution is not continued after a strobe.

- **Enhanced: Modules 9226 and 9418 (Define the Status of an Axis or Spindle):**
  Comprehensive possibilities for deactivating/activating an axis without rebooting have been created. The improvements are described in detail in "Writing axis information—activating and deactivating axes" on page 444.

- **Enhanced: Modules 9040, 9041 and 9049**
  The following additional axis information can be read:
  7: Actual values in the reference system with backlash
  8: Distance traversed in [mm] since the last lubricating pulse
  9: Temperature compensation.
  For the complete module documentation, see:
  - "Module 9040 Reading of axis coordinates by the PLC in the format 1/1000 (0.001) mm" on page 441
  - "Module 9041 Reading of axis coordinates by the PLC in the format 1/10000 (0.0001) mm" on page 442
  - "Module 9049 Read position value and speed value of an axis" on page 440.

- **Enhanced: Modules 9240, 9248, 9250, 9277, 9290, 9291, 9295 and 9343**
  (Modules for accessing files)
  The name of an OEM machine parameter from the CfgOemString configuration object can now be transferred instead of the path name. The transferred character string must begin with "**>OEM.**" and end with the key name from CfgOemString. The parameter value must contain the path name to the file. If no key name with the transferred path name is found, the Modules 9248, 9277 and 9295 return the error code 62. The other modules set the error marker to the value provided for an invalid path name.

- **Enhanced: Module 9247 (Searching for a Condition in a Table)**
  Module 9247 now accepts the SQL keyword "WHERE" in a string in order to transfer a search condition to the module, see page 1211.

- **Enhanced: Module 9434 (Selecting Parameter Block)**
  The PLC program can now also activate another parameter block while a PLC positioning movement is being executed. A PLC positioning movement can also be started while a new parameter set is being selected via Module 9434. In this case, the PLC program must ensure the safety of the machine. The PLC programmer must ensure that parameter blocks containing machine parameters that are not suitable for this drive are not selected. Unsuitable parameter blocks can cause incorrect positioning movements and lead to damage to the machine!
  New error code 5 added:
  The module was not executed, because the axis is deactivated.
  For a detailed description of the module, see page 314.

New PLC modules

- **Module 9066: Status of HEIDENHAIN Hardware**, see page 675.
- **Module 9067: Status of HEIDENHAIN Software**, see page 676.
- **Module 9128: Torque Limiting by the PLC**, see page 616.
- **Module 9129: Status of Torque Limiting by the PLC**, see page 617.
- **Module 9158: Maximum Torque**, see page 617.
- **Module 9146: Storing/Restoring Actual Position Values**, see page 720.
- **Module 9155: Axis Switchover from Closed Loop to Open Loop**, see page 451.
- **Module 9156: Axis Switchover from Open Loop to Closed Loop**, see page 452.

Note

Modules 9155 and 9156 were introduced to ensure compatibility with earlier HEIDENHAIN contouring controls. HEIDENHAIN recommends:

Using Modules 9226 and 9418 for activating and deactivating axes and spindle, if possible.

- **Module 9193: Setting the Operating Hours Counter**, see page 1011.
- **Module 9227: Position auxiliary axes and NC axes**, see page 464.

1 Update Information No. 2

1.1 Overview

1.2 NC Software 548 328-03

1.2.1 Important notes on updating software

Please remember the following important information when updating the software versions listed below:

- **MANUALplus 620: 548 328-02 to 548 328-03**
- **MANUALplus 620: 548 328-01 to 548 328-03**
  For an update from NC SW 548 328-01 to NC SW 548 328-03 it is absolutely necessary to comply with the information in the Update Information No.1 on the software update from NC SW 548 328-01 to NC SW 548 328-02, see "Important notes on updating software" on page 17.

Note

HEIDENHAIN recommends:

Making a backup of the control (e.g. with TNCbackup), before updating the NC software.

Saving your current machine configuration. The configuration editor (DATA BACKUP soft key) can be used for this purpose.

If you later want to undo the software update and return to the previous software version, you need the saved configuration data of the old version!

Note

Please perform the update of the NC software as described in the Technical Manual in Chapter 2 "NC Software Exchange".

Note

Be sure to remember the important information about the software update, which is provided on the following pages.

- **Checking and saving new machine parameters:**
  After having installed the new NC software and rebooted the control, you must check and confirm the new machine parameters. The code number dialog box appears on the screen:

- Enter the MP code number **95148** and confirm your entry with the ENT key.
- Press the **UPDATE RULES** soft key.
  Check the listed update rules. Each entry in the list stands for a new parameter that was added to the system by the update.
- Exit the **UPDATE RULES** with the **END** soft key.
- Press the **CONFIG DATA** soft key.
  Before the configuration editor opens, an informational window is displayed, reporting the removal of the CfgRestorePosition machine parameter. Press the **NEXT** soft key.
- All new machine parameters are marked with a red exclamation point in the configuration editor. The control indicates if certain machine parameters are faulty. Please ignore these messages for the time being.
- **Important step:** Press the **SAVE** soft key.
- The **Configuration data changed** dialog box opens. Press the **SAVE** soft key again. The new machine parameters are now automatically saved in the *.cfg files.
- Press the **END** soft key and exit the **Machine Parameter** operating mode by pressing the **END** soft key again.

The control then continues booting.

1.2.2 Description of the new functions

Software Options

The features of the following MANUALplus 620 options will change as of NC software version 548 328-03:

- **Software option 3—tools and technology (option 10)**
  This option is no longer required for the support of multi-edge tools (tools with multiple cutting edges or multiple reference points) in smart.Turn and DIN programs. The support is now included in the standard features of the NC software.
- **Touch probe functions (option 17)**
  This option has been expanded by automatic workpiece measurement with touch probes.

Overview of the improvements

A summary of the improvements in NC software 548 328-03 is given below. For more detailed information, please refer to the Technical Manual sections indicated in the brief descriptions given below.

Machine Configuration

- **New: HSCI/PROFIBUS diagnostics**
  After the OEM code word has been entered, the BUS DIAGNOSIS soft key will now be available in the Organization mode of operation after pressing the DIAGNOSIS soft key. The arrangement of all bus participants as well as the properties and conditions of each individual device are displayed graphically and separately for HSCI and Profibussee "Bus diagnosis" on page 1111.
- **Implementation of new kinematic model**
  The new kinematic model developed for NCK-based controls is now also available for the MANUALplus 620. The new kinematic model makes it possible to use the PC software **KinematicsDesign** to create and modify kinematic configurations, see "Machine kinematics on lathes (as of NC software 548328-03)" on page 487.
- **Expanded: OEM cycles with dialog texts and help graphics**
  The OEM can define his own cycles (G500 to G590) with dialog texts and help graphics. A prepared file in XML format is available as a template in the control under **PLC:\resource\formdlg\g_oem.fdxml** see "OEM cycles (G5xx)" on page 1184.
- **Expanded: PLC G functions with dialog texts and help graphics**
  Dialog texts with help graphics can now be saved in the control for G functions (G602 to G699) that are not executed in a subprogram, but by the PLC see "PLC-G functions (G6xx)" auf Seite 1184.
- **Expanded: creating subprograms**
  when subprograms are written, a separate help graphic can can now be defined and displayed for every input field in the dialog.
- **Expanded: update rules rules for OEM parameters**
  In the directory **PLC:\config\lathe\manplus**, release-specific files were provided for the update rules of the tool builder.
  In the control's shipping condition, the **UpdateOemRe100x.cfg** files (update rules for release x) are empty.

Machine parameters new/expanded/ changed

Expanded/changed machine parameters:

MP number	Config object	Parameter	Description of change
100402	**CfgFilter**	**MP_typeFilter1**	The input range for the maximum value of the filter order was increased from 31 to 63.
100404		**MP_typeFilter2**	
113102	**CfgDashboardElement /DB_LD_S1 /DB_LD_S2**	**MP_attribut**	For spindles, the utilization of the rotational speed limit can be displayed by setting bit1=1 in **MP_attribut**, see "Configuring dashboards" auf Seite 936.
113102	**CfgDashboardElement /DB_OVERRIDE**	**MP_attribut**	In the override display of the dashboard, the current rapid traverse reduction can be displayed by setting bit1=1 in **MP_attribut**, see "Configuring dashboards" auf Seite 936.

New machine parameters:

MP number	Config object	Parameter	Description
116104	**CfgPlcSymName**	**MP_readTsfData**	The data for feed rate and spindle speed can now be read out by the PLC from the "Set T, S, F" dialog, see "Transferring the spindle speed and feed rate data to the PLC" on page 1036.
116105		**MP_displayMode**	The active display mode (e.g. "manual control," "Automatic") can now be passed on to the PLC, see "Transfer display mode to PLC" auf Seite 1037.
116106		**MP_setToolPlace**	By entering a PLC operand, the PLC can command the turret position, which you would otherwise have to set manually; see "MP_setToolPlace" on page 1037.

MP number	Config object	Parameter	Description
202601	CfgKinSimpleTrans	MP_dir	As of NC software 548 328-03, a new kinematic model is available for the MANUALplus 620 as an alternative to the previous kinematic model. The new kinematic model, which is provided as a standard feature, makes it possible to use the PC software KinematicsDesign to create and modify kinematics for the control. For configuration of the new kinematic model, see "Machine kinematics on lathes (as of NC software 548328-03)" on page 487.
202602		MP_val	
202603		MP_realtimeComp	
202701	CfgKinSimpleAxis	MP_dir	
202702		MP_axisRef	
202801	CfgKinSimpleModel	MP_kinObjects	
202901	CfgKinComposModel	MP_subKinList	
202902		MP_activeSpindle	
202904		MP_tiltingAllowed	
203001	CfgKinList	MP_kinCompositeModels	
203701	CfgKinAnchor	MP_kindOfAnchor	
300112	CfgAxis	MP_advancedSettings	This machine parameter makes it possible to configure that the PLC movement of an individual axis is not canceled if the touch probe is deflected; see "Advanced settings for individual axes" on page 409.
400413	CfgReferencing	MP_externRefPulse	For referencing a single axis with an external reference signal, see "Referencing with external reference signal" auf Seite 709.
600418	CfgToolMount	MP_kinModelToModify	Assign axis mirroring to a tool holder: see "Axis mirroring on lathes (as of NC software 548328-03)" on page 501.
600419		MP_kinModel	
601807	CfgGlobalProperties	MP_threadHandWheelOn	Activation of the "handwheel in thread" function: see "Activate handwheel in the thread" on page 1044.
601808		MP_freezeVconst	Activate constant spindle speed for rapid traverse movements and active constant surface speed: see "Freeze spindle speed for rapid traverse" on page 1024.
604803	CfgGlbDispSettings	MP_axesDisplayMode	This machine parameter makes it possible to configure the type of axis display in the dashboard. You can choose between actual value, nominal value, following error or distance to go; see "Configuring the axis display" on page 937
604901	CfgMMISettings	MP_extManualMode	Activation of an extended menu structure in the **Machine** and **Program Run** operating modes:see "Expanded menu structure" on page 884.
604902		MP_extProgramMode	

Operation and technology

- **New: Dashboard display of unit quantities and time per unit**
  The dashboard element **QuantityInformationAndTimePerUnit** is now available in the machine display to show the defined quantity, current quantity, time per unit and total time of finished workpieces; see "Configuring dashboards" on page 936.

- **New: Configuring the axis display in the dashboard**
  Effective immediately, it is possible to use **MP_axesDisplayMode** to set whether the axis display in the dashboard should show the actual value, nominal value, following error or distance to go; see "Configuring dashboards" on page 936.

- **New: Expanded menu structure**
  In the **Machine** and **Program Run** operating mode it is now possible to use **MP_axesDisplayMode** to activate an expanded menu structure; see "Expanded menu structure" on page 884.

- **New: Workpiece measurement with TS touch probe**
  The control now also supports tool measurement with a touch probe; see "Touch Probe" on page 1015.

- **New: Activate handwheel in the thread**
  Effective immediately, the "handwheel in thread" function can be activated through a machine parameter. This function makes it possible to compensate position and angular error of the linear and spindle axes; see "Activate handwheel in the thread" on page 1044.

- **Expanded: Dashboard display for reed rate reduction**
  The attribute evaluation of the dashboard element "ChannelDisplay" was expanded so that now you can display the feed rate reduction in the dashboard in addition to the spindle and feed rate override; see "Configuring dashboards" on page 936.

- **Expanded: Speed display for spindles**
  For a C-axis with external spindle drive (e.g. S4 drives the main spindle S1 through a transmission), the spindle speed of S4 is now displayed in the S1-element in the C-axis mode; see "Configuring dashboards" on page 936.

- **Expanded: Utilization display for spindles**
  The attribute evaluation of the dashboard element "LoadDisplay" was expanded for spindles so that you can now display the speed limitation in the dashboard in addition to the utilization display for spindle drives; see "Configuring dashboards" on page 936.

Configuring the Axes and Spindle

- **New: Advanced settings for individual axes**
  MP_advancedSettings makes it possible to configure that the PLC movement of an individual axis is not canceled if the touch probe is deflected by another axis.
  A faster acceleration and filter calculation for PLC movements can be activated in addition to the configuration of a fast axis; see "Advanced settings for individual axes" on page 409.
- **New: Referencing with external reference signal**
  Effective immediately, an external reference signal can be used instead of the reference signal of the connected motor encoder to reference an individual axis of the entire system (e.g. spindle), see "Referencing with external reference signal" on page 709.
- **New: Axis mirroring with new kinematic model**
  In the new kinematic model, an axis can also be mirrored within a kinematic group without switching the kinematics; see "Axis mirroring on lathes (as of NC software 548328-03)" on page 501.
- **New: Keeping spindle speed for rapid traverse constant**
  With the new machine parameter **MP_freezeVconst** you can prevent the spindle from changing its speed during constant surface speed Vconst according to the current diameter if there are several rapid traverse movements. This can prevent unnecessary braking and acceleration of the spindle during several successive rapid traverse movements; see "Freeze spindle speed for rapid traverse" on page 1024.
- **Expanded: Configuration of the nominal position value filters**
  The possible maximum value for the filter order was increased from 31 to 63 in the machine parameters **MP_orderFilter1** and **MP_orderFilter2**, which are effective for all axes; see "Nominal position value filter" on page 538.

Analog hardware

- **MANUALplus 620 for retrofitting**
  The new MC 320T main computer also supports purely analog drive control. The axes are controlled exclusively through the analog nominal speed command interface. The compact MC 320T main computer is integrated behind the screen of the operating panel to save space.

PLC programming

- **New: Transferring the spindle speed and feed rate data to the PLC**
  With **MP_readTsfData** you can now configure whether the PLC will read the data for feed rate and spindle speed shown in the dialog **"Set T, S, F "** from the tables **ch_tsf.mch** and **sp_tsf.msp**; see "Transferring the spindle speed and feed rate data to the PLC" on page 1036.
- **New: Transferring the display mode of the machine display to the PLC**
  The active display mode (e.g. "manual control," and "automatic") of the dashboard can now be passed on to the PLC. This makes it possible to switch the view of the machine display; see "Transfer display mode to PLC" on page 1037.
- **New: Tool pocket preset by PLC**
  In the machine parameter **MP_setToolPlace** you can now define a symbol variable name under which the PLC can name the NC a tool pocket that is then used and displayed by the user interface; see "Tool pocket preset by the PLC" on page 1037.

New PLC modules

- **Module 9142: Reference value for a programmed axis**, see page 400.
- **Module 9250: Starting the editor for sections of a table**, see page 1213)
- **Module 9251: End the PLC table editor**, see page 1214.
- **Module 9252: Position the cursor in the PLC table editor**, see page 1215.
- **Module 9285: Disable operating modes**, see page 892.
- **Module 9480: Selection of channel display**, see page 1040.
- **Module 9481: Finding the channel display**, see page 1041.
- **Module 9482: Selection of spindle display**, see page 1041.
- **Module 9483: Finding the spindle display**, see page 1042.

1 Update Information No. 3

1.1 Hardware

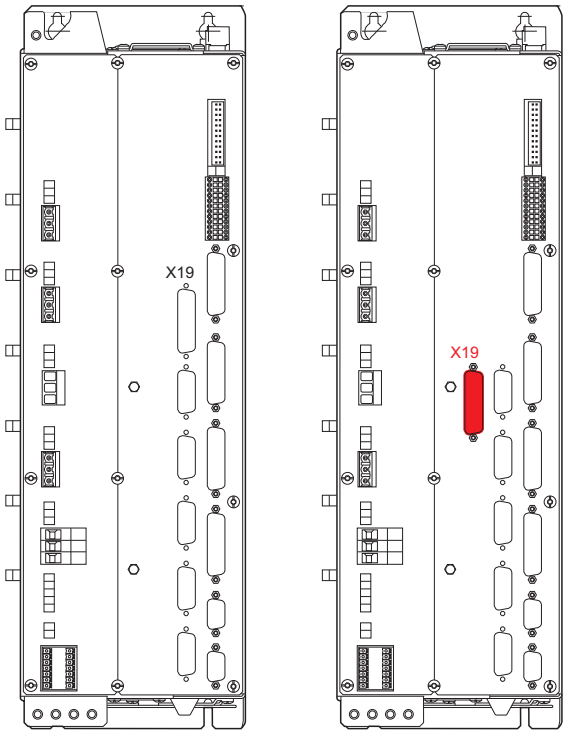
1.1.1 UEC 11x controller unit with inverter and PLC I/O

A new variant was released for the UEC 11x compact controller units.

Previously, the UEC 11x was shipped both with variant 01 and variant 02. The new variant 03 replaces both of the previous variants. The changed ID numbers are detailed in the table below:

Device	Old IDs	New ID
UEC 111 Without Functional Safety (FS) Max. 4 control loops 4 x speed and 4 x position inputs DC-link power rating: 14 kW 38 x PLC inputs, 23 x PLC outputs	625 777-01 625 777-02	625 777-03
UEC 112 Without Functional Safety (FS) Max. 5 control loops 5 x speed and 5 x position inputs DC-link power rating: 14 kW 38 x PLC inputs, 23 x PLC outputs	625 779-01 625 779-02	625 779-03

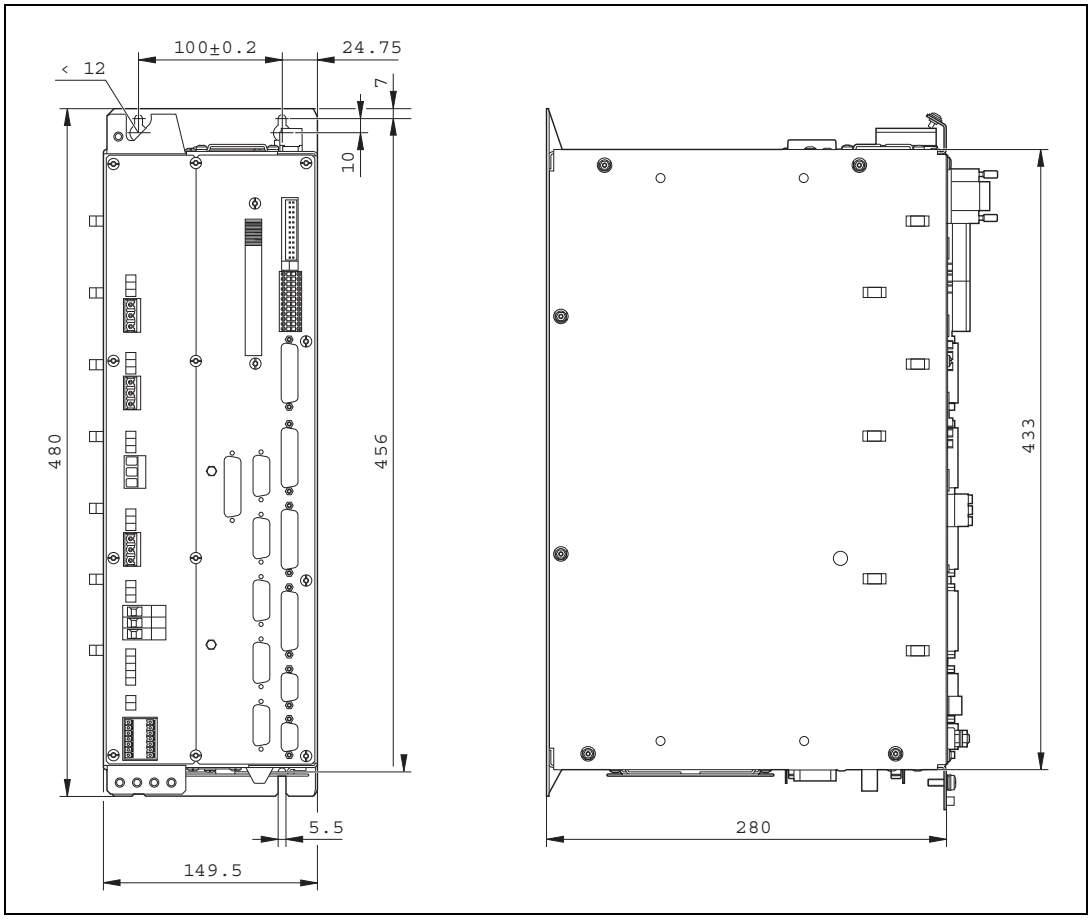
Overview of changes to the UEC 11x since its introduction:

UEC 111, UEC 112	
ID number	**Changes**
625 777-01 625 779-01	**Initial introduction**
625 777-02 625 779-02	**1st Improvement** ■ **Support of motor holding brakes** New X344 and X394 connections for control of motor holding brakes of axes 1 to 4. In variant 01 the motor holding brakes had to be controlled via PLC outputs. ■ **New terminals for the axis motors** Pluggable screw terminals for the axis motors connected to X81 to X84. In variant 01, the terminals for the motors are permanently integrated in the unit and are not pluggable.
625 777-03 625 779-03	**2nd Improvement** ■ **Optimized active cooling** A new arrangement of fans in the UEC 11x has improved the distribution of cooling air in the unit. ■ **Changed arrangement of connection X19** Connection X19 (speed encoder of the 4th axis, only UEC 112) was shifted slightly. See drawing below. **UEC 11x, old:** **UEC 11x, new:** 

Note

Operation of the new UEC 11x (variant 03) with the CNC PILOT 620 is supported starting with the initial release of NC software version 688 945-01.

Dimensions



2 Introduction

2.1 Meaning of the Symbols Used in this Manual

Danger

Failure to comply with this information could result in most serious or fatal injuries, and/or in substantial material damage.

Attention

Failure to comply with this information could result in injuries and interruptions of operation, including material damage.

Note

Tips and tricks for operation as well as important information, for example about standards and regulations as well as for better understanding of the document.

2.2 Proper Operation

The described components may only be installed and operated as described in this manual. Commissioning, maintenance, inspection and operation are only to be performed by trained personnel.

2.3 Trained Personnel

Trained personnel in the sense of this manual means persons who are familiar with the installation, mounting, commissioning, and operation of the HEIDENHAIN components. Furthermore, electrical engineering work on the system may be carried out only by trained electrical engineering technicians or persons trained specifically for the respective application.

Basically, persons who perform work on HEIDENHAIN components must meet the following requirements:

- They must have been trained or instructed in the standards of safety engineering.
- They must have appropriate safety equipment (clothing, measuring systems).
- They should be skilled in first-aid practice.

2.4 General Information

The HEIDENHAIN **MANUALplus 620** contouring control was conceived for standard CNC lathes. The control has an integrated digital drive control and drives the power modules through PWM signals.

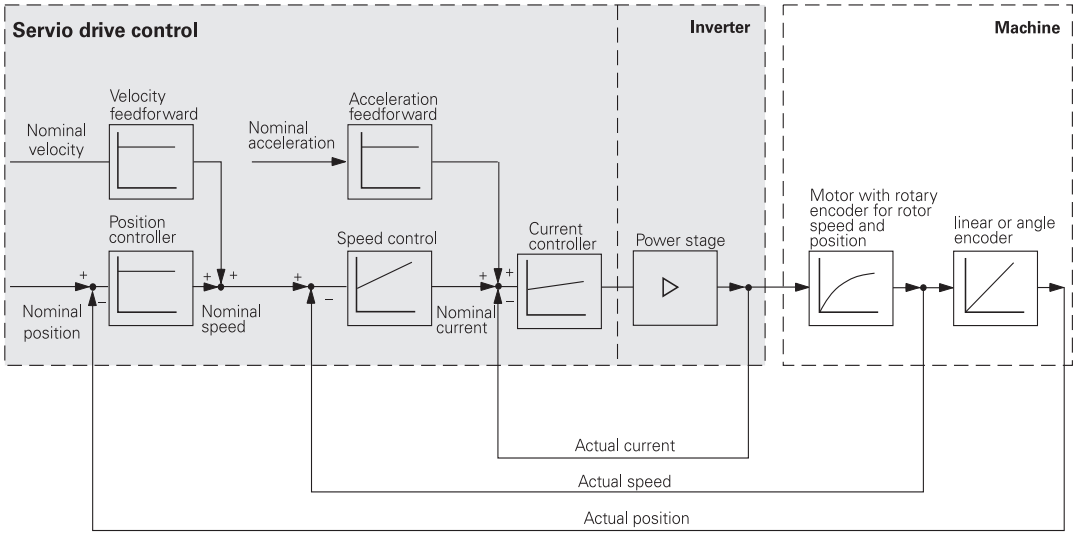
Integration of the drive controllers in the MANUALplus 620 offers the following advantages:

- All the software is contained centrally in the NC; this means that the individual components of the NC, such as feed axes, spindle, NC or PLC, are optimally matched.
- High control quality, because the position controller, speed controller and current controller are combined into one unit.
- The same functions are available for commissioning, optimizing and diagnosing feed drives as well as spindles.

The **MANUALplus 620** supports lathes up to the following level:

- 1 slide (NC channel)
- 4 axes (X/ Z/ Y and W axis)
- 2 spindles (main spindle, driven tool)
- 1 C axis (via main spindle drive or with separate drive)

The following diagram illustrates drive control with the use of a rotary encoder for measuring the shaft speed actual values and a linear encoder or angle encoder for measuring the position actual values.



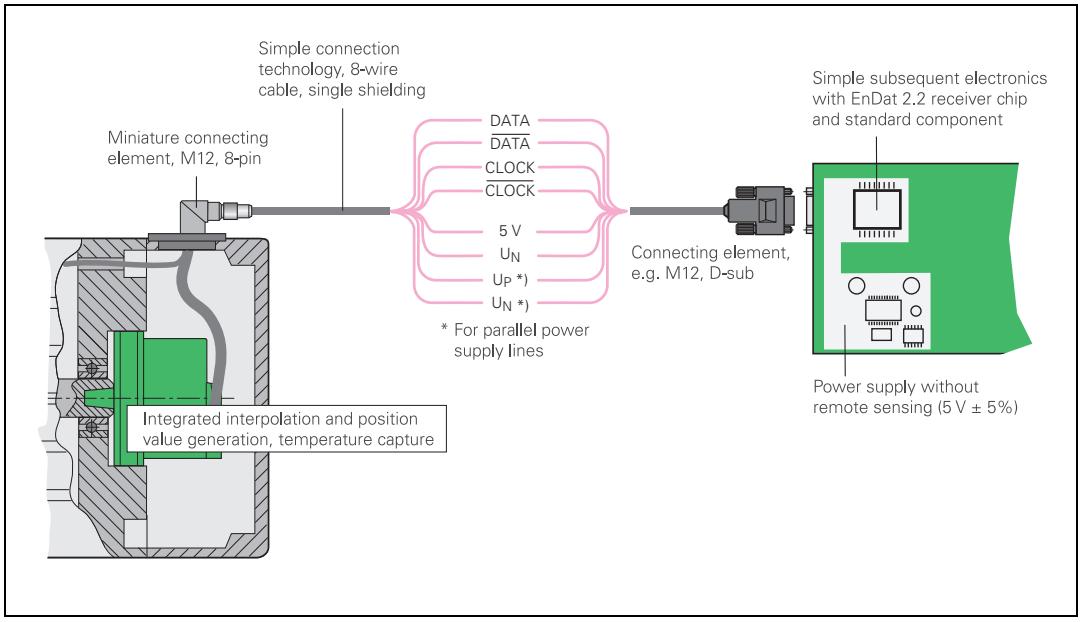
The **MANUALplus 620** is based on **HEIDENHAIN NCK**, the software platform for the HEIDENHAIN control family. Other controls with NCK are, for example, the TNC 620, TNC 320 or the CNC PILOT 620.

The **MC 6110T**, the compact main computer of the MANUALplus 620, takes up very little space, as it is housed in the operating console, directly behind the integrated 12.1-inch TFT display.

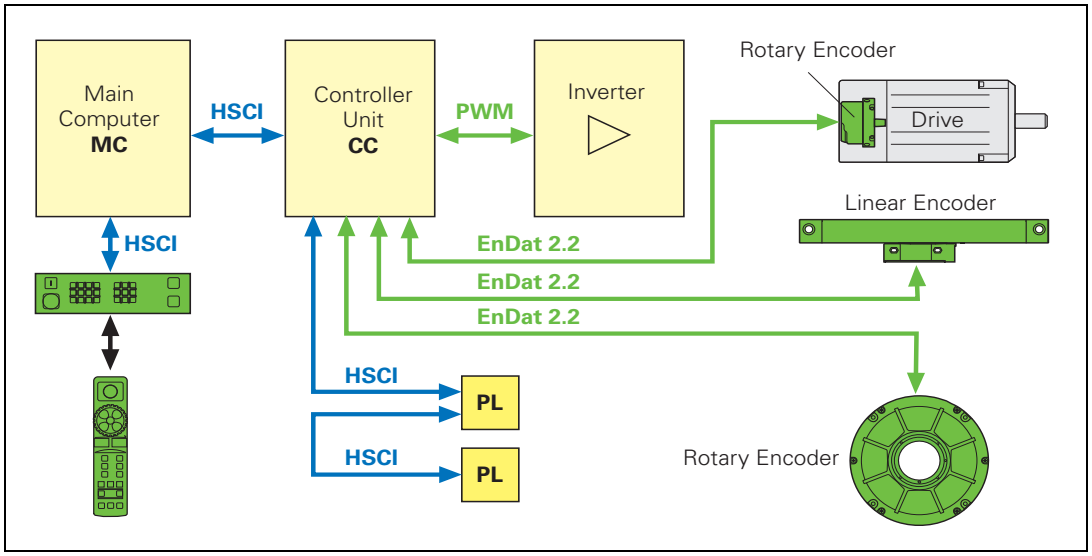
The MC is connected to the CC controller unit, the MB machine operating panel and the PL 6xxx PLC input/output systems via **HSCI** (HEIDENHAIN Serial Controller Interface). The connection of the various control components via HSCI offers numerous benefits, including:

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

The MANUALplus 620 is prepared for the connection of incremental and absolute position and shaft-speed encoders. EnDat 2.2, which is purely digital and compatible to version 2.1, makes it possible to very rapidly transmit highly resolved position values over long cable lengths. An overview of EnDat 2.2:



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



The **MANUALplus 620** is designed for connection of a compact or modular inverter system. A complete control package, including drives and HEIDENHAIN motors, can be delivered (see the "Inverter Systems and Motors" Technical Manual).

2.4.1 HSCI interface

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

The following features characterize the HSCI connection:

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

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

The following applies to the assignment of the HSCI address:

- The HSCI address (bus address) is the result of the device's position in the bus
- The master (MC) always has the HSCI address 0.
- The HSCI addresses of the slaves result from their position in the bus:
  - First device after the master (MC): Bus address 1
  - Second device after the master (MC): Bus address 2
  - etc.

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

More information on the connection of the HSCI components is provided under “HSCI" on page 122.

2.5 Component Overview of MANUALplus 620

2.5.1 MC main computer, CFR memory card and SIK

The MANUALplus 620 always includes at least the following components:

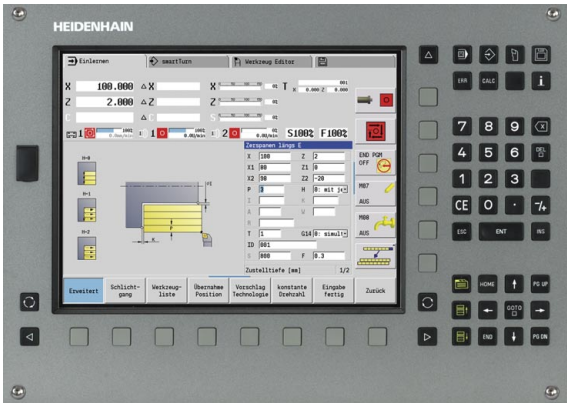
- MC 6110T main computer
  (MC = Main Computer)

and either:

- CC 61xx controller unit
  (CC = Controller Computer)
- PL 620x system PL
- Modular or compact HEIDENHAIN inverter system

or

- UEC 11x controller unit with integrated inverter and PLC

MC 6110T main computer	
Compact main computer for incorporation in the operating console, with integrated TFT flat-panel display and operating keys. Machine operating panel is optional. ■ Processor: Intel Celeron M 1.0 GHz ■ 1 GB RAM ■ HSCI interface ■ Ethernet interface 100BaseT ■ 3 x USB 2.0 (1 in the operating panel, 2 on the rear) ■ 1 x RS-232C MC 6110T **ID 731 604-xx**	
Additionally required: ■ **CFR** CompactFlash memory card • CompactFlash memory card, Type 1 • Contains the NC software • 1 GB memory capacity, of which 250 MB are for V:\ partition and 50 MB for O:\ partition. Remaining memory is used for system data.	

■ **SIK** System Identification Key • Contains the NC software license for enabling control loops and software options • The SIK number provides the control with a unique identification.	

Main computer components	**ID**
MC 6110T compact main computer	731 604-xx
CompactFlash memory card (CFR) with MANUALplus 620 software	733 606-51
SIK component, 3 control loops and the "Teach-in," "smart.Turn," "Thread recutting" and "C-axis machining" software options are enabled, see "Software options" on page 85.	733 604-53
Control loop expansions:	**ID**
Enabling for:	
Addition of a 4th control loop	354 540-01
Addition of a 5th control loop	353 904-01
Addition of a 6th control loop	353 905-01
Addition of a 7th control loop	367 867-01

Further control loops can be enabled in addition to the control loops of the respective SIK version. The maximum number is:

- UEC 111: 4 control loops
- UEC 112: 5 control loops
- CC 6106: 6 control loops
- CC 6108: 7 control loops

Note

Control loop for C axis: If the main spindle drive is used for the C axis, then one control loop for both the main spindle and the C axis suffices. If there is a separate drive for the C axis, then the main spindle and C axis each require their own control loop.

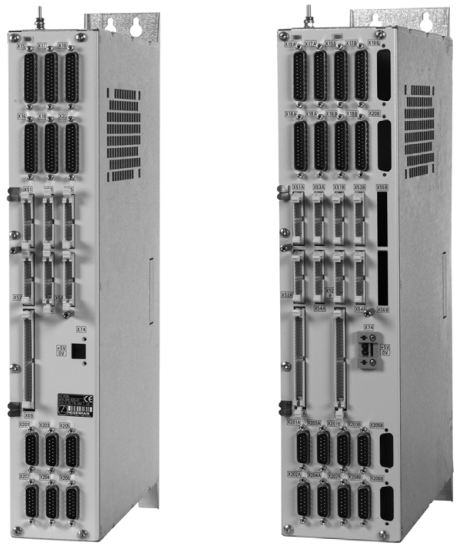
2.5.2 SIK (System Identification Key)

Each control is clearly identified by the SIK (System Identification Key).

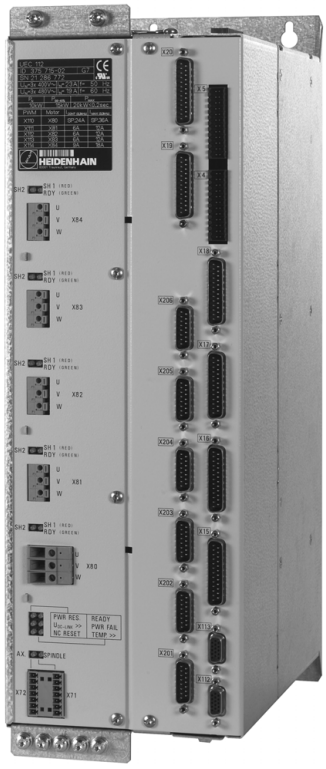
Note

If you replace the MANUALplus 620, you must also replace the SIK in order to ensure that the enabled options will also be enabled on the new hardware.

2.5.3 CC 6106 controller unit

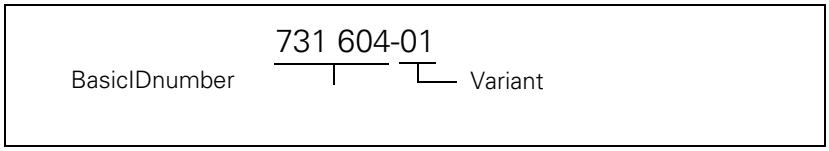
CC 610x	
Controller unit with HSCI interface for up to 6 or 8 control loops It is equipped with: ■ 6 or 8 PWM outputs ■ 6 or 8 speed encoder inputs ■ 6 or 8 position encoder inputs ■ 2 SPI expansion slots ■ Power supply through UV(R) power supply unit CC 6106 ID 598 928-xx CC 6108 ID 662 637-xx	

2.5.4 UEC 11x controller unit with integrated inverter and PLC

UEC 11x	
Controller unit with integrated inverter and PLC for up to 5 control loops. Compact unit for machines with limited number of axes and low power demands. It is equipped with: ■ HSCI interface ■ 4 (UEC 111) or 5 (UEC 112) speed encoder inputs ■ 4 (UEC 111) or 5 (UEC 112) position encoder inputs ■ Connection for 3 axes plus spindle (UEC 111) or Connection for 4 axes plus spindle (UEC 112) ■ Braking resistor ■ 38 PLC inputs, 23 PLC outputs (expandable via PL 61xx) ■ Integrated power supply unit 24 V NC / 3.5 A for supplying the HSCI components **UEC 111 with 4 control loops** ID 625 777-xx **UEC 112 with 5 control loops** ID 625 779-xx	

Designation of the MC 6110T, CC 610x and UEC 11x

ID of MC 6110T:



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

Variant	Changes to MC 6110T
xxx xxx-y1	Initial version

Variant	Changes to CC 6106
xxx xxx-y1	Prototype
xxx xxx-y2	Prototype
xxx xxx-y3	Initial version

Variant	Changes to CC 6108
xxx xxx-y1	Initial version

Variant	Changes to UEC 111
xxx xxx-y1	Prototype
xxx xxx-y2	Initial version

Variant	Changes to UEC 112
xxx xxx-y1	Prototype
xxx xxx-y2	Initial version

2.5.5 PLC input/output systems with HSCI interface

The PLC inputs and outputs of the MANUALplus 620 are available via the external modular PL 6xxx PLC input/output systems.

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

Note

The MC 6110T main computer of the MANUALplus 620 does **not** have integrated PLC inputs/outputs, and has **no** connections for TS or TT touch probes. In order to operate the control, at least the PL 62xx system PL (when using a CC 610x) or the UEC 11x controller unit with integrated inverter and PLC is necessary. (The system PL is integrated in the UEC.)

The PLC inputs/outputs components are configured with the PC software IOconfig.

System PL

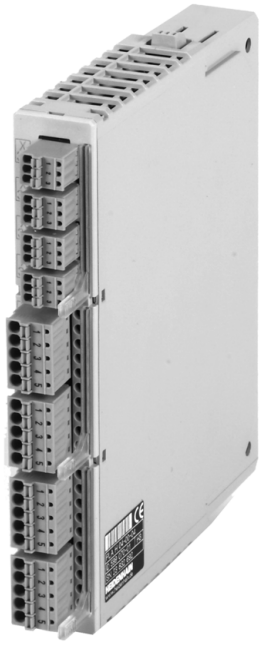
PL 62xx	
System PL, consisting of PLB 620x basic module and I/O modules. ■ One module must be in the HSCI system if no UEC 11x is used ■ Available with 4, 6 or 8 slots ■ HSCI interface ■ Connections for TS and TT touch probes ■ Safety-relevant PLC inputs/outputs For an overview of the available I/O modules, See "I/O modules" on page 59. They are mounted on standard NS 35 rails (DIN 46 227 or EN 50 022) **PLB 6204** **ID 591 832-xx** **PLB 6206** **ID 630 054-xx** **PLB 6208** **ID 630 055-xx**	

PL 61xx	
Expansion PL, consisting of PLB 620x basic module and I/O modules. ■ Available with 4, 6 or 8 slots ■ HSCI interface ■ Up to 7 PL 61xx can be present in the HSCI system For an overview of the available I/O modules, See "I/O modules" on page 59. They are mounted on standard NS 35 rails (DIN 46 227 or EN 50 022) **PLB 6104** **ID 591 828-xx** **PLB 6106** **ID 630 058-xx** **PLB 6108** **ID 630 059-xx**	

I/O modules

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

PLD-H xx–xx–xx	
Digital I/O module: ■ **PLD-H 16-08-00:** I/O module with 16 digital inputs and 8 digital outputs ■ **PLD-H 08-16-00:** I/O module with 8 digital inputs and 16 digital outputs ■ **PLD-H 08-04-00:** I/O module with 8 digital inputs and 4 digital outputs ■ **PLD-H 04-08-00:** I/O module with 4 digital inputs and 8 digital outputs **PLD-H 16-08-00** **ID 594 243-xx** **PLD-H 08-16-00** **ID 650 891-xx** **PLD-H 08-04-00** **ID 598 905-xx** **PLD-H 04-08-00** **ID 727 219-xx**	

PLA-H xx–xx–xx	
Analog I/O module: ■ **PLA-H 04-00-04:** Analog module with 4 analog inputs ±10 V, 0 analog outputs and 4 inputs for Pt 100 thermistors. ■ **PLA-H 08-04-04:** Analog module with 8 analog inputs ±10 V, 4 analog outputs ±10V and 4 inputs for Pt 100 thermistors. **PLA-H 04-00-04** **ID 599 070-xx** **PLA-H 08-04-04** **ID 675 572-xx**	
Empty housing	
...for partial assembly **ID 383 022-11**	

2.5.6 PSL 130 low-voltage power supply unit

PSL 130	
Power pack to supply HSCI components with +24 V. ■ The power is supplied via line voltage (L1, L2) and the DC-link voltage U_z. This is used to produce the +24-V NC and +24-V PLC output voltages. ■ Both output voltages are produced by two internally separated power supplies. The NC and PLC power supplies are galvanically isolated and fulfill the requirements of EN 61800-5-1 for "low voltage electrical separation." ■ Output voltages must be grounded according to EN 60204-1:2006 "protective extra-low voltage (PELV)" (see Grounding Diagram). ■ The two output voltages can be connected in parallel. This way the PSL 130 provides an output voltage of +24 V at a maximum output power of 750 W. Please observe the information and regulations for the power connection cited under "PSL 130 low-voltage power supply unit" on page 140 and in the "Inverter Systems and Motors" Technical Manual. **ID 575 047-xx**	

PSL 135

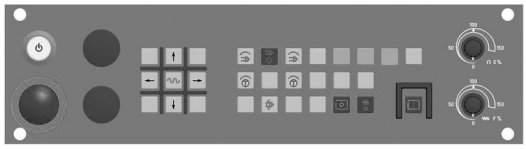
Power pack for supplying the HSCI components for use of a non-HEIDENHAIN inverter system.

- The power is supplied via line voltage (L1, L2) and the DC-link voltage U_z. This is used to produce the +24-V NC, +24-V PLC and +5-V output voltages.
- +24 V NC and +24 V PLC are produced by two internally separated power supplies. The NC and PLC power supplies are galvanically isolated and fulfill the requirements of EN 61800-5-1 for "low voltage electrical separation."
- Output voltages must be grounded according to EN 60204-1:2006 "protective extra-low voltage (PELV)" (see Grounding Diagram).
- +24 V NC and +24 V PLC can be connected in parallel. This way the PSL 135 provides an output voltage of +24 V at a maximum output power of 750 W.

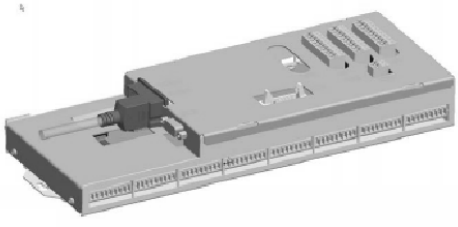
Please observe the information and regulations for the power connection cited under "PSL 130 low-voltage power supply unit" on page 140 and in the "Inverter Systems and Motors" Technical Manual.

ID 627 032-xx

2.5.7 MB 620T machine operating panel

MB 620T	
The MB 620T is equipped with: ■ HSCI interface ■ Handwheel connection, X23 ■ Spindle and feed rate override potentiometer ■ Snap-on (exchangeable) keys, see "Key symbols" on page 67. The key functions are freely definable via the PLC ■ 8 PLC inputs and 8 PLC outputs ■ Two holes for additional keys or keylock switches Controls and displays: ■ 9 axis keys ■ 17 function keys ■ NC start ■ NC stop ■ Feed rate stop ■ Spindle stop ■ EMERGENCY STOP button ■ Control voltage On[1] [1] Key is illuminated **ID 737 610-xx**	

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

PLB 6001	
The PLB 6001 is equipped with: ■ HSCI interface ■ Handwheel connection, X23 ■ 64 PLC inputs, 32 PLC outputs for keys / key illumination ■ Connection for spindle-speed and feed-rate override potentiometer ■ Screw fastening or top-hat-rail mounting ■ Weight: 1.2 kg **ID 668 792-xx**	

2.5.9 Handwheels

All handwheels are available with and without detent. For handwheels with detent, the cogging torque prevents movements of the handwheel due to motions or vibrations of the machine. On handwheels without detent, this is prevented by a defined holding torque.

Handwheels with detent feature 100 detent positions per revolution, i.e. every 3.6°. The machine manufacturer defines the increment via the machine configuration.

HR 410 handwheel

Portable electronic handwheel with snap-on (exchangeable) keys, see "Key symbols" on page 67.

- Five axis selection keys
- Keys for traverse direction
- Keys for preset feeds
- Actual-position-capture key
- Three keys for machine functions (definable via PLC)
  - Spindle right/left/stop
  - NC start/stop, spindle start; (for HEIDENHAIN basic PLC program)
- Two permissive buttons (24 V)
- Emergency stop button (24 V)
- Magnetic holding pads

The handwheel is available with or without detent.

See the following table for the possible handwheel assignments.

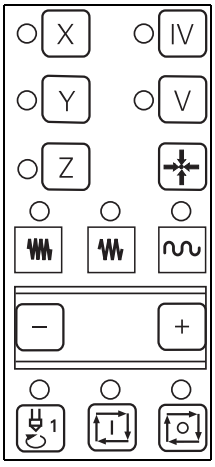
ID 312 879-01Connecting cable to cable adapter (spiral cable 3 m)
ID 296 467-xxConnecting cable to cable adapter (normal cable)
ID 296 687-xxConnecting cable to cable adapter (with metal armor)
ID 296 466-xxAdapter cable to control
ID 281 429-xxExtension to adapter cable
ID 271 958-03Dummy plug for emergency stop circuit

Key assignments **HR 410** handwheel *):

Without detent:
ID 296 469-55

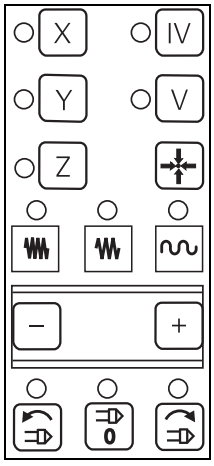
With detent:
ID 535 220-05

(for PLC basic program)



Without detent:
ID 296 469-54

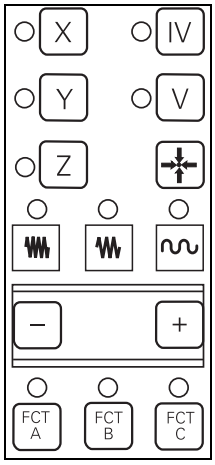
(special assignment)



Without detent:
ID 296 469-53

With detent:
ID 535 220-03

(standard)



*) For a list of the keys available for exchanging, see "Key symbols" on page 67.

HR 130 handwheel

Panel-mounted handwheel

ID 540 940-01HR 130 handwheel **with** detent, with ergonomic control knob, radial cable outlet

ID 540 940-03HR 130 handwheel **without** detent, with ergonomic control knob, radial cable outlet

HR 180 panel-mounted handwheel for connection to a position input.

ID 540 940-08 Handwheel with mechanical detent (100 stops per handwheel revolution), with ergonomic knob.

Output signal: 1 V_{PP}

HRA 110 handwheel adapter

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

ID 261 097-03	HRA 110
ID 540 940-06	HR 150 handwheel **without** detent, with ergonomic knob, radial cable outlet
ID 540 940-07	HR 150 handwheel **with** detent, with ergonomic knob, radial cable outlet
ID 270 908-xx	Selection switch

2.5.10 Key symbols

Key symbols for the spindle

Key	Description Print/Background ID		Key	Description Print/Background ID
	Spindle stop White/Red 330 816-08			Spindle start White/Green 330 816-09
	Spindle direction left Black/Gray 330 816-40			Spindle direction right Black/Gray 330 816-41
	Spindle stop White/Red 330 816-47			Spindle start White/Green 330 816-46
	Clamp the spindle Black/Gray 330 816-48			

Key symbols with axis designations

Key	Description Print/Background ID		Key	Description Print/Background ID
	X Black/Orange 330 816-24			Y Black/Orange 330 816-36
	Z Black/Orange 330 816-25			A Black/Orange 330 816-42
	B Black/Orange 330 816-26			C Black/Orange 330 816-23
	U Black/Orange 330 816-43			V Black/Orange 330 816-38
	W Black/Orange 330 816-45			IV Black/Orange 330 816-37

Key symbols for axis direction keys for the principal axes

Key	Description Print/Background ID		Key	Description Print/Background ID
	X– Black/Gray 330 816-63			X+ Black/Gray 330 816-64
	X– <– Black/Gray 330 816-18			X+ –> Black/Gray 330 816-17

Key	Description Print/Background ID		Key	Description Print/Background ID
	X'– –> Black/Gray 330 816-0W			X'+ <– Black/Gray 330 816-0V
	X– <– Black/Gray 330 816-0N			X+ –> Black/Gray 330 816-0M
	Y– Black/Gray 330 816-67			Y+ Black/Gray 330 816-68
	Y'– –> Black/Gray 330 816-21			Y'+ <– Black/Gray 330 816-20
	Y– <– Black/Gray 330 816-0P			Y+ –> Black/Gray 330 816-0R
	Y– –> Black/Gray 330 816-0D			Y+ <– Black/Gray 330 816-0E
	Z– Black/Gray 330 816-65			Z+ Black/Gray 330 816-66
	Z– <– Black/Gray 330 816-19			Z+ –> Black/Gray 330 816-16
	Z'– –> Black/Gray 330 816-0L			Z'+ <– Black/Gray 330 816-0K

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

Key	Description Print/Background ID		Key	Description Print/Background ID
	A– Black/Gray 330 816-95			A+ Black/Gray 330 816-96
	B– Black/Gray 330 816-97			B+ Black/Gray 330 816-98
	C– Black/Gray 330 816-99			C+ Black/Gray 330 816-0A
	U– Black/Gray 330 816-0B			U+ Black/Gray 330 816-0C
	V– Black/Gray 330 816-70			V+ Black/Gray 330 816-69

Key	Description Print/Background ID		Key	Description Print/Background ID
	W– Black/Gray 330 816-0G			W+ Black/Gray 330 816-0H
	IV– Black/Gray 330 816-71			IV+ Black/Gray 330 816-72

Key symbols for machine functions

Key	Description Print/Background ID		Key	Description Print/Background ID
	Special function Black/Gray 330 816-0X			Function A White/Black 330 816-30
	Function B White/Black 330 816-31			Function C White/Black 330 816-32
	Function 1 Black/Gray 330 816-73			Function 2 Black/Gray 330 816-74
	Function 3 Black/Gray 330 816-75			Function 4 Black/Gray 330 816-76
	Function 5 Black/Gray 330 816-77			Unlock door Black/Gray 330 816-78
	Unlock door Black/Gray 330 816-79			Coolant Black/Gray 330 816-80
	Coolant (internal) Black/Gray 330 816-0S			Coolant (external) Black/Gray 330 816-0T
	Rinse water jet Black/Gray 330 816-81			Spotlight Black/Gray 330 816-82
	Chip removal Black/Gray 330 816-83			Chip conveyor Black/Gray 330 816-84
	Tool change Black/Gray 330 816-89			Tool changer left Black/Gray 330 816-85
	Tool changer right Black/Gray 330 816-86			Unlock tool Black/Gray 330 816-87

Key	Description Print/Background ID		Key	Description Print/Background ID
	Unlock tool Black/Gray 330 816-88			Lock tool Black/Gray 330 816-94
	Lock tool Black/Gray 330 816-0U			Retract axis Black/Gray 330 816-91

Other key symbols

Key	Description Print/Background ID		Key	Description Print/Background ID
	No symbol –/Black 330 816-01			No symbol –/Gray 330 816-61
	NC start White/Green 330 816-11			NC stop White/Red 330 816-12
	NC start White/Green 330 816-49			NC stop White/Red 330 816-50
	Feed rate 1 Black/Gray 330 816-33			Feed rate 2 Black/Gray 330 816-34
	Rapid traverse Black/Gray 330 816-35			Permissive button White/Green 330 816-22
	Permissive button Black/Gray 330 816-90			Actual position capture White/Black 330 816-27
	– White/Black 330 816-28			+ White/Black 330 816-29
	Menu selection –> Black/Gray 330 816-92			Menu selection <– Black/Gray 330 816-93
	0 Black/Gray 330 816-0Y			

2.5.11 Touch probes

Touch probes for workpiece measurement are connected via the system PL 62xx or the UEC 11x. The touch probes generate a trigger signal that captures the current position value. For more information about touch probes, please request the "Touch Probes" brochure or CD-ROM from HEIDENHAIN.

Workpiece measurement

The TS touch trigger probe has a stylus with which it probes workpieces. The MANUALplus 620 provides standard routines for workpiece measurement (software option 17 required). The touch probes are available with various taper shanks. Assorted styli are available as accessories.

TS 220 touch probe

Touch trigger probe with cable connection for signal transmission for machines with manual tool change. For workpiece setup and measurement during machining.

ID 293 488-xx	TS 220
ID 633 613-xx	Adapter cable for connection to the system PL or the UEC

TS 740, TS 640, TS 444, TS 440 touch probes

Touch trigger probe with infrared transmission, for workpiece setup and measurement during machining. For machines with automatic tool changer.

- TS 440 with compact dimensions
- TS 444 with alternative battery-free power supply via compressed air through the spindle head
- TS 640 with wide-range infrared transmission and long operating time
- TS 740 with high probing accuracy and repeatability, and low probing forces

The infrared transmission is established between the TS touch probe and the SE transceiver unit. The following SE units can be combined with the TS touch probes:

- SE 640 for integration in the machine workspace
- SE 540 for integration in the spindle head

ID 573 757-xx	TS 740
ID 620 189-xx	TS 640
ID 620 046-xx	TS 440
ID 588 008-xx	TS 444
ID 631 225-xx	SE 640 transmitter-receiver unit
ID 626 001-xx	SE 540 transmitter/ receiver unit

TS 640, TS 740

TS 440, TS 444

SE 640

SE 540

TT 140 tool touch probe

Triggering touch probe with rated break point of the connection pin for the probe head and optical deflection display. An additional connection pin is delivered with the touch probe.

ID 527 797-03	TT 140
ID 676 497-01	Probe contact, cuboid
ID 559 758-01	Connection pin
ID 633 616-xx	Adapter cable for connection to the system PL or the UEC

2.5.12 Other accessories

Further components	ID
Adapters for encoder signals	
TTL (HEIDENHAIN layout)/1 V_{PP}	317 505-01
TTL (SIEMENS layout) / 1 V_{PP}	317 505-02

2.5.13 Documentation

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

The features of the control are described in the following manuals:

- **MANUALplus 620 User's Manual** (ID 634 864-xx)
- **MANUALplus 620 / CNC PILOT 620 User's Manual, smart.Turn and DIN Programming** (ID 685 556-xx)

The HEIDENHAIN inverters and motors are described in the

- **Technical Manual for Inverters and Motors** (ID 208 962-xx)

The **DataPilot MP/CP 620** is the new programming station for the MANUALplus 620 and CNC PILOT 620 lathe controls.

- **DataPilot MP/CP 620** CD-ROM
  - Demo software (ID 737 139-xx)
  - Single user license (ID 737 157-xx)
  - Network license for 14 training stations (ID 737 158-xx)
  - Network license for 20 training stations (ID 737 159-xx)

Other documentation

Available in the form of brochures:

- **MANUALplus 620** brochure (ID 634 865-xx)
- **MANUALplus 620 OEM** brochure (ID 634 867-xx)
- Touch Probes: **brochure** (ID 208 951-xx)
- **Inverter Systems** brochure (ID 622 420-xx)
- **Motors** brochure (ID 208 893-xx)
- **Remote Diagnosis with TeleService**product overview (ID 348 236-xx)
- **Touch Probes** CD-ROM (ID 344 353-xx)

2.6 Brief Description

2.6.1 Specifications of the MANUALplus 620

Specifications	MANUALplus 620
MC 6110T	
	Compact contouring control with integrated main computer, TFT color flat-panel display and operating keys ■ Intel Celeron M 1.0 GHz processor ■ 1 GB SDRAM main memory ■ 100 MHz bus frequency ■ HSCI interface ■ Three USB interfaces ■ Unique identification of MC 6110T through SIK (System Identification Key)
CC 610x	
	All position and speed encoder inputs 1 V_{PP} or EnDat
	■ HSCI interface ■ Max. 6 or 8 digital control loops ■ 6 or 8 position and 6 or 8 speed encoder inputs with 1 V_{PP} or EnDat 2.2 for axes and spindle (EnDat 2.2 is backward-compatible to EnDat 2.1) ■ 6 or 8 PWM outputs ■ Power supply via UV(R), UE or UR power supply unit
UEC 11x	
	Controller unit with integrated inverter and PLC, for machines with low power demands ■ HSCI interface ■ Controller unit with position, speed and current controller ■ UEC 111: Up to 4 digital control loops, connection for 3 axes and spindle ■ UEC 112: Up to 5 digital control loops, connection for 4 axes and spindle ■ Integrated braking resistor ■ 38 PLC inputs, 23 PLC outputs (expandable via PL 61xx) ■ Interfaces to the speed encoders ■ Interfaces to the position encoders ■ Interfaces for one TS and TT touch probe each ■ +24-V NC power supply with 2.5 A for MC and other control components

Specifications	MANUALplus 620
Axis feedback control	
	Velocity feedforward control / Operation with following error / Jerk limiting
	Connection of the CC controller unit via HSCI
Cycle time for path interpolation	3 ms
Options	
	Software options can be enabled by entering a code number.
Display	
	12.1-inch TFT color flat-panel display (integrated)
Program memory	
	250 MB on CFR memory card
Input resolution and display step	
Linear axes	X axis: 0.5 µm (diameter: 1 µm) Z Y and W axis: 1 µm
C axis	0,001°
Block processing time	
	3 ms
Interpolation	
Straight line	In 2 axes (max. ±100 m), optional in 3 principal axes
Circle	In 2 axes (radius max. 999 m), optional additional linear interpolation in the third axis
C axis	Interpolation of X and Z linear axes with the C axis (option)
Helix	Superimpositioning of circular and straight paths
Look-ahead	Precalculation of up to 5000 blocks for determining the contouring velocity profile

Specifications	MANUALplus 620
Regulation with CC 6xxx / UEC 1xx	
Position loop resolution	$\frac{\text{Signal period of position encoder}}{4096}$ or encoder resolution (EnDat 2.2 interpol.)
Path interpolation	3 ms
Cycle time of current controller	PWM frequency Cycle time: 3333 Hz 150 µs 4000 Hz 125 µs 5000 Hz 100 µs
Cycle time of speed controller	Speed controller cycle time = 2 · current controller cycle time
Cycle time of position controller	= Cycle time of speed controller
Feed rate	Maximum feed rate: $\frac{60000\ 1/\text{min}}{\text{No. of pole pairs}} \cdot$ Screw pitch in mm at f_{PWM} = 5000 Hz ■ Up to approx. 32.4 m/min (27 kHz) or approx. 480 m/min (400 kHz) for encoders with 20 µm grating period ■ Up to approx. 162 m/min (27 kHz) or approx. 2400 m/min (400 kHz) for encoders with 100 µm grating period ■ mm/min or mm/revolution ■ Constant surface speed ■ Feed rate with chip breaking
Shaft speed (spindle)	Maximum revolutions per minute: $n_{max} = \frac{f_{PWM} \cdot 60000\ \text{min}^{-1}}{p \cdot 5000\ \text{Hz}}$ n_{max}: Maximum spindle speed [min^{-1}] f_{PWM}: PWM frequency [Hz] p: Number of pole pairs The following PWM frequencies are available: 3333 Hz, 4000 Hz, 5000 Hz

Specifications	MANUALplus 620
Thread	
	■ Longitudinal thread ■ Transversal thread (as DIN cycle) ■ Tapered thread ■ API thread ■ Multiple thread ■ Discontinuous threads with slanted entry and exit ■ Variable pitch ■ Tapping ■ Thread milling (possible only with C axis)
Error compensation	
	■ Linear and nonlinear axis error ■ Backlash ■ Hysteresis ■ Reversal error during circular movements ■ Thermal expansion ■ Stick-slip friction ■ Sliding friction ■ Tool nose (cutting) radius ■ Milling tool radius
Monitoring functions	
	■ Amplitude of encoder signals ■ Edge separation of encoder signals ■ Absolute position for encoders with distance-coded reference marks ■ Following error ■ Movement monitoring ■ Standstill monitoring ■ Nominal speed value ■ Checksum of safety-related functions ■ Power supply ■ Buffer battery ■ Operating temperature ■ Running time of the PLC program ■ Motor current ■ Motor temperature ■ Temperature of power stage ■ dc-link voltage

Specifications		MANUALplus 620
Integrated PLC		
	PLC memory	50 MB on CFR memory card
	Program format	Statement list (STL)
	PLC process memory RAM	Dynamic, determined by the free main memory of the control
	PLC cycle time	9 ms to 30 ms (adjustable)
	PLC inputs, 24 V–	Via PL
	PLC outputs, 24 V–	Via PL
	Analog inputs, ±10 V	Via PL
	Analog outputs, ±10 V	Via PL
	Inputs for thermistors	Via PL

Machine interfacing	MANUALplus 620
Commissioning aids	
	■ Oscilloscope ■ Trace function ■ Table function ■ API DATA function ■ Watchlist function ■ Logic diagram (integrated in oscilloscope) ■ Log ■ OnLine monitor (OLM) ■ TNCopt PC software ■ TNCscopeNT recording software ■ TeleService
Interfaces	
	■ 100BaseT Fast Ethernet interface ■ 2 x HSCI ■ 3 x USB 2.0 ■ RS-232-C/V.24 with max. 115 Kbps ■ Expanded interface with LSV-2 protocol for data exchange and external operation of the control with HEIDENHAIN software TNCremoNT
Permissible temperature range	Incoming air in panel or electrical cabinet 0 to +50 °C Temperature range outside the panel: 0 °C to +45 °C Storage: -20 °C to +60 °C

2.6.2 User functions

User functions		MANUALplus 620
Operating modes		
	Manual operation	■ Manual slide movement through manual direction keys, intermediate switch or electronic handwheels ■ Graphic support for entering and running cycles without saving the machining steps in alternation with manual machine operation ■ Thread repair (thread reworking in a second workpiece setup)
	Teach-In	■ Sequential linking of fixed cycles, where each cycle is run immediately after input or is graphically simulated and subsequently saved.
	Program Run	■ Cycle programs, DIN PLUS or smart.Turn programs in single block or full sequence

User functions		MANUALplus 620
	Setup functions	■ Workpiece datum setting ■ Definition of tool-change position ■ Definition of protection zone ■ Tool measurement—alternatively: • By touch-off • With a touch probe (**Option 17**) • With measuring optics (**Option 17**) ■ Automatic workpiece measurement (option**17**)
Programming		
	Cycle programming	■ Area clearance cycles for simple and complex contours, as well as contours described with Interactive Contour Programming (ICP) ■ Contour-parallel turning cycles ■ Recessing cycles for simple and complex contours, as well as contours described with ICP ■ Repetitions with recessing cycles ■ Recess turning cycles for simple and complex contours, as well as contours described with ICP ■ Undercut and parting cycles ■ Threading cycles for single or multi-start longitudinal, taper or API threads ■ Cycles for axial and radial drilling, pecking and tapping operations with the C axis ■ Thread milling with the C axis ■ Axial and radial milling cycles for slots, figures, single surfaces and polygons as well as for complex contours defined with ICP for machining with the C axis ■ Helical slot milling with the C axis ■ Linear and circular patterns for drilling and milling operations with the C axis ■ Use of DIN macros in cycle programs ■ Transfer of cutting values from technology database ■ Context-sensitive help graphics ■ Conversion of cycle programs to smart.Turn programs
	Interactive contour programming(ICP)	■ Contour definition with linear and circular contour elements ■ Immediate display of entered contour elements ■ Calculation of missing coordinates, intersections, etc. ■ Graphic display of all solutions for selection by the user if more than one solution is possible ■ Chamfers, rounding arcs and undercuts available as form elements ■ Input of form elements immediately during contour creation or by superimposition later ■ Changes to existing contours can be programmed

User functions	MANUALplus 620
Subfunction of the ICP	C-axis machining on face and lateral surface: ■ Description of individual holes and hole patterns (only with smart.Turn) ■ Description of figures and figure patterns for milling operations (only with smart.Turn) ■ Creation of freely definable milling contours Y-axis machining (option 70) in the XY and ZY planes (only with smart.Turn): ■ Description of individual holes and hole patterns ■ Description of figures and figure patterns for milling operations ■ Creation of freely definable milling contours DXF import (option 42): Import of contours for lathe and milling operations
smart.Turn programming	■ Program blocks (UNITS) for the complete description of a machining block (geometry, technology and cycle data) ■ Dialog boxes divided into overview and detail forms ■ Fast navigation between the fillable forms and input groups via the "smart" keys ■ Context-sensitive help graphics ■ Start unit with global settings ■ Transfer of global values from the start unit ■ Transfer of cutting values from technology database ■ Units for all turning and recessing operations for simple contours and ICP contours ■ Units for boring, drilling and milling operations with the C and Y axes for simple holes, milling contours and drilling and milling patterns or those programmed with ICP ■ Special units for activating/deactivating the C axis, subprograms and section repeats ■ Verification graphics for blank and finished part and for C and Y axis contours ■ Turret assignment and other setup information in the smart.Turn program ■ Parallel programming ■ Parallel simulation

User functions	MANUALplus 620
DIN PLUS - Programming	■ NC programming as per DIN 66025 (ISO 6983) ■ Extended command format (IF...THEN...ELSE...) ■ Simple geometry programming (calculation of missing data) ■ Powerful fixed cycles for area clearance, recessing, recess turning and thread machining ■ Powerful machining cycles for boring, drilling and milling with the C axis ■ Powerful fixed cycles for drilling and milling with the Y axis (option 70) ■ Subprograms ■ Programming with variables ■ Contour description with ICP ■ Program verification graphics for workpiece blank and finished part ■ Turret assignment and other setup information in the DIN PLUS program ■ Conversion of smart.Turn units into DIN PLUS command sequences ■ Parallel programming ■ Parallel simulation
Program verification graphics	Graphic simulation of cycle execution for cycle programs, smart.Turn or DIN PLUS programs: ■ Display of the tool paths as wire-frame or cutting-path graphics, special identification of the rapid-traverse paths ■ Machining simulation (2-D material-removal graphic) ■ Side or face view, or 2-D view of cylindrical surface for verification of C-axis machining ■ Display of programmed contours ■ View of face and YZ plane for verification of Y-axis machining ■ Workpiece blank definition ■ Three-dimensional graphic display of the workpiece blank and finished part ■ Shifting and magnifying functions
Machining time analysis	■ Calculation of machining times and idle times ■ Consideration of switching commands triggered by the CNC ■ Individual times per cycle or tool change

User functions	MANUALplus 620
Tool database	■ Database for 250 tools with tool description ■ **Option 10:** 999 tools ■ Tool description can be entered for every tool ■ Automatic inspection of tool-tip position with respect to the contour ■ Compensation of tool-tip position in the X/Y/Z plane ■ High-precision correction via handwheel, capturing compensation values in the tool table ■ Automatic tool-tip and cutter radius compensation ■ Management of multipoint tools (multiple inserts on one tool holder) ■ Tool monitoring after rated life of insert or number of workpieces ■ **Option 10:** Tool monitoring with automatic tool change after tool insert wear
Tool compensation	■ Compensation of tool-tip position in the X/Y/Z plane ■ Automatic tool point position detection (left, right, inward, outward) ■ High-precision adjustment via handwheel, capturing compensation values in the tool table ■ Automatic tool-tip and cutter radius compensation
Technology database	■ Access to cutting data after definition of workpiece material, cutting material and machining mode (reduced number of database entries). The MANUALplus distinguishes between 16machining modes. Each workpiece-material/tool-material combination includes the cutting speed, the main and secondary feed rates, and the infeed for 16machining modes. ■ Automatic determination of the machining modes from the cycle or the machining unit ■ The cutting data are entered in the cycle or in the unit as default values. ■ 9 workpiece-material/tool-material combinations (144 entries) ■ **Option 10:** 62 workpiece-material/tool-material combinations (992 entries)
Conversational languages	Chinese (simplified), Chinese (traditional), Czech, Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Polish, Portuguese, Russian, Spanish, Swedish **Only with option 41** (ID 530 184-xx): Estonian, Korean, Latvian, Lithuanian, Norwegian, Romanian, Slovak, Slovenian, Turkish

2.6.3 Software options

Option number	Option	ID	Comment
0	Additional axis	354 540-01	**Additional control loops 1, 2 and 3 (additional control loop 3 available on with MC 420)**
1		353 904-01	
2		353 905-01	
8	Software option 1 Teach-in	632 226-01	**Cycle programming** ■ Contour description with ICP ■ Cycle programming ■ Technology database with 9 workpiece-material/ tool-material combinations
9	Software option 2 smart.Turn	632 227-01	**smart.Turn** ■ Contour description with ICP ■ Programming with smart.Turn ■ Technology database with 9 workpiece-material/ tool-material combinations
10	Software option 3 Tools and technology	632 228-01	**Tools and technology** ■ Tool database expanded to 999 entries ■ Technology database expanded to 62 workpiece-material/tool-material combinations ■ Tool life monitoring with exchange tools
11	Software option 4 Thread recutting	632 229-01	**Thread** ■ Thread recutting ■ Handwheel superimposition during thread cutting
17	Touch probe functions	632 230-01	**Tool and workpiece measurement** ■ Determining tool-setting dimensions with a touch probe ■ Determining tool-setting dimensions with an optical gauge ■ Measuring the workpiece with a touch probe
41	Additional languages	530 184-xx	**Additional conversational language** ■ Estonian, Korean, Latvian, Norwegian, Romanian, Slovak, Slovenian, Turkish, Lithuanian
42	DXF import	632 231-01	**DXF import** ■ Loading of DXF contours
55	C-axis machining	633 944-01	**C-axis machining**
70	Y-axis machining	661 881-01	**Y-axis machining**
94	W-axis machining	679 676-01	**W-axis support**

Accessories	MANUALplus 620
PL 6xxx PLC input/output systems with HSCI	Up to eight PL 6xxx can be connected
	■ PL 620x (system PL) • Necessary once for each control system (except with UEC) • Has connections for TS and TT touch probes • Safety-relevant inputs/outputs • Available for 4, 6 or 8 I/O modules ■ PL 610x (expansion PL) • As addition to the system PL for increasing the number of PLC inputs/outputs • Available for 4, 6 or 8 I/O modules ■ I/O modules • PLD-H 16-08-00 I/O module with 16 digital inputs and 8 digital outputs • PLD-H 04-00-04 I/O module with 4 analog inputs for PT 100 thermistors and 4 analog inputs for ± 10 V • PLD-H 00-12-00 I/O module with 12 analog outputs for ± 10 V
Power supply for HSCI components	■ PSL 13x 24-V power pack for supplying the HSCI components. • Outputs: NC: 24 V– (double isolation) PLC: 24 V– (basic isolation) Per output: max. 21 A/ 500 W Total: Max. 32 A / 750 W • Outputs can be connected in parallel
Electronic handwheels	■ One **HR 130** panel-mounted handwheel or an **HR 410** portable handwheel at the X23 serial input. ■ Up to three **HR 150** at the X23 serial input via **HRA 110**. ■ **HR 180** panel-mounted handwheels at position inputs. The number is limited by the number of vacant position inputs. You can additionally connect an HR 410 serial handwheel, an HR 130 or up to three HR150 (via HRA 110) to X23.
Workpiece touch probe	■ **TS220** 3-D touch trigger probe with cable connection or ■ **TS 440, TS 444, TS 640 and TS 740** triggering 3-D touch probe with infrared transmission
Tool touch probe	■ **TT 140** with a cuboid probe contact

Accessories	MANUALplus 620
Software	■ **PLCdesignNT**[a] PLC software developing environment ■ **IOconfig**[a] Software for configuring PLC I/O and PROFIBUS-DP components ■ **TNCremoNT** Data transfer software ■ **TNCremoPlus** Data transfer software with "live" screen ■ **TNCscopeNT**[a] Software for recording data online or evaluating oscilloscope measurement series ■ **DriveDiag**[a] Software for diagnosis of digital control loops ■ **TNCopt**[a] Software for putting digital control loops into service ■ **KinematicsDesign**[a] Software for configuring the machine kinematics ■ **TeleService** Software for remote diagnostics, monitoring, and operation

a. For registered customers, these software products are available for downloading over the Internet.

Attention

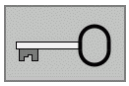
Do not make any changes to the operating system, the operating system settings, or to the software supplied by HEIDENHAIN. Non-HEIDENHAIN applications may be used only with the permission of HEIDENHAIN.

2.7.1 Designation of the software

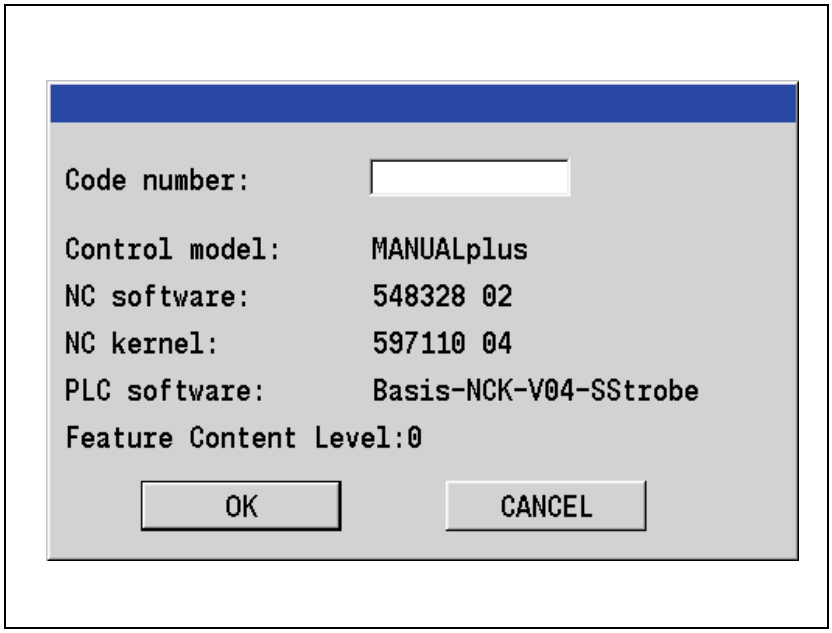
The control features a separate software for the NC and the PLC. The NC software is identified with an eight-digit ID number.

To show the software version:

- Switch to the **Organization** mode of operation.



- Press the soft key. The MANUALplus 620 shows the control model and the versions of the NC and PLC software. An installed service pack is shown by **SPx** after the ID number of the NC software.



Model

The MANUALplus 620 is approved for export to all countries. No export license is required for the NC software of the control.

HEIDENHAIN may release a new NC software type when it introduces extensive new functions.

2.7.2 PLC software

The PLC software is stored on the hard disk of the MANUALplus 620. HEIDENHAIN offers a PLC basic program you can order directly from HEIDENHAIN. With the PLC development software **PLCdesignNT,** the PLC basic program can very easily be adapted to the requirements of the machine.

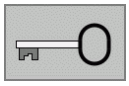
2.7.3 Enabling additional control loops or software options

In the standard version of the MANUALplus 620 with SIK (ID 530 005-53), three control loops (2 axes + 1 spindle), the software options with option numbers #08 "Cycle programming", #09 "smart.Turn", #11 "Threads" and #55 "C axis" are enabled. You can enable up to three further control loops (#00, #01, #02) as well as the software options with option numbers #10 Tools and Technology, #17 Tool Measurement, #41 Additional Conversational Languages, #42 DXF Import, #70 Y-axis Machining and #94 W-axis Support by entering a code number.

If you wish to enable an additional control loop or software options, please contact HEIDENHAIN for the code number. HEIDENHAIN can give you the code number after having been informed of the SIK number.

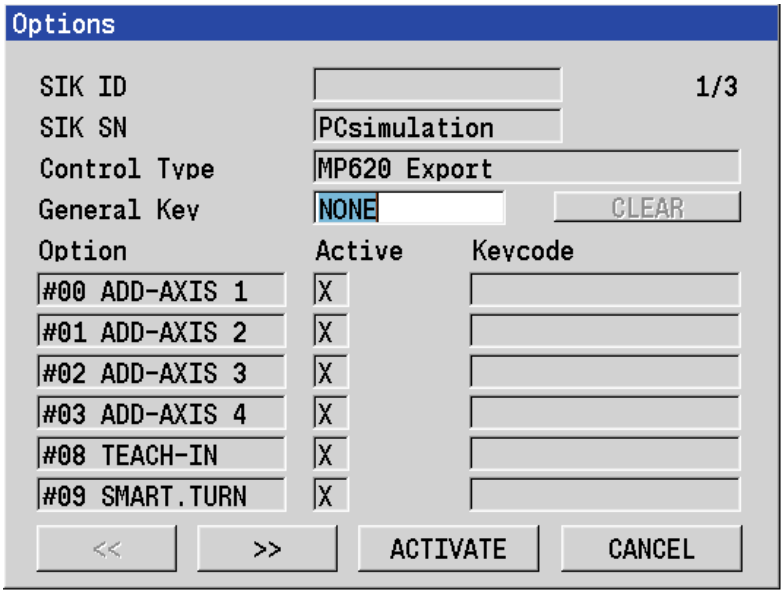
To enable options, proceed as follows:

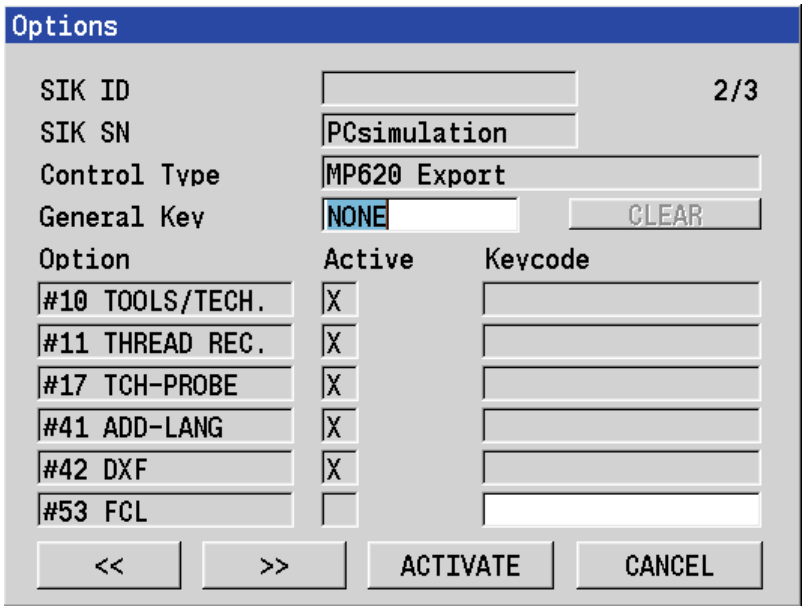
- Switch to the **Organization** mode of operation.

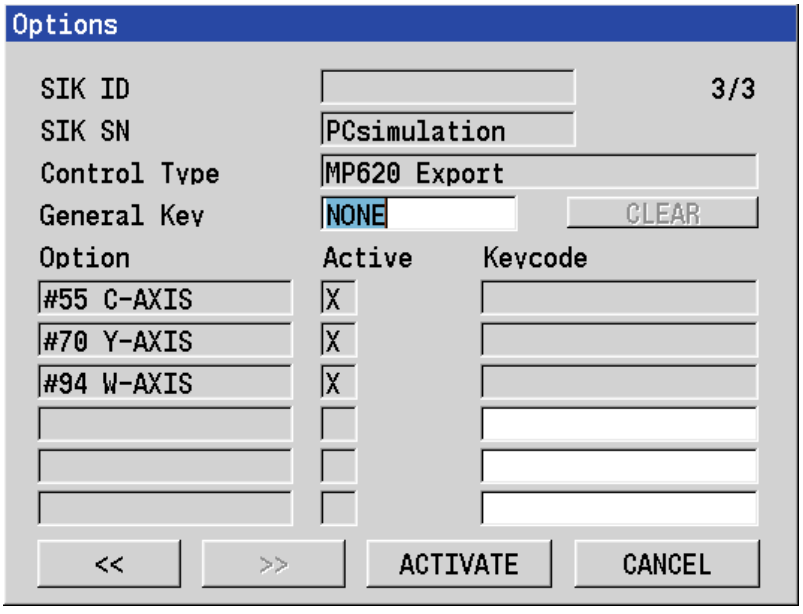


- Press the soft key. The MANUALplus 620 shows the control model and the software versions.
- Enter the code number SIK and confirm your entry with the ENT key.

The following SIK dialog for the enabling of options appears:




Options
SIK ID
2/3
SIK SN
PCsimulation
Control Type
MP620 Export
General Key
NONE
CLEAR
Option
Active
Keycode
#10 TOOLS/TECH.
X
#11 THREAD REC.
X
#17 TCH-PROBE
X
#41 ADD-LANG
X
#42 DXF
X
#53 FCL
<<
>>
ACTIVATE
CANCEL


Options
SIK ID
3/3
SIK SN
PCsimulation
Control Type
MP620 Export
General Key
NONE
CLEAR
Option
Active
Keycode
#55 C-AXIS
X
#70 Y-AXIS
X
#94 W-AXIS
X
<<
>>
ACTIVATE
CANCEL

Display	Meaning
SIK ID	SIK number
SIK SN	SIK serial number
Control Type	Control model
General Key	Enter the master code number 65535 to enable all options for the duration of two weeks. **NONE:** Master code number has not been entered yet. **dd.mm.yyyy:** Date up to which all options will be enabled. It is not possible to enable the options again by entering the master code number. **EXPIRED:** The two weeks since the master code number was entered have expired.
Option column	Display of option numbers and brief description of options that can be enabled.
Active column	**X:** Option is enabled
Keycode column	Input field for entering the key code of the option to be enabled. This field is gray for options that have already been enabled.

- The **Option** column shows all options available for the MANUALplus 620.
- Enter under **Keycode** the code number for enabling the option.
- Confirm your entry with the **ACTIVATE** soft key or button. HEIDENHAIN can give you the code number after having been informed of the SIK number.

The message **Option <number> has been set** appears.

If the code number is correct, the enabled option is identified by the entry **X** in the **Active** column.

Note

Should you have ordered more than one option, and received more than one key code from HEIDENHAIN:

You must enable each option individually. It is possible to enter more than one option in the **Keycode** column, but then these options cannot be enabled. In this case the **Incorrect password** error message always appears.
It is not possible to enable more than one option simultaneously. Enter the first key code and then press the **ACTIVATE** soft key. Then enter the second key code and press the **ACTIVATE** soft key again, and so on.

- Press the **CANCEL** soft key or button. To be able to use the option, you first have to restart the control.

After entry of the master code number (65535), all options are enabled for 14 days. After these 14 days have expired, an error message appears, the program currently running is aborted, and it is not possible to restart the program. In order to avoid this error message (and the associated unintentional program abort), the General Key must be cleared with the **CLEAR** soft key. Only the options actually enabled are then available, and the General Key cannot be entered again.

Soft keys and buttons in the SIK menu

Soft key or button	Function
CLEAR	The General Key, which enables all options for two weeks, is disabled.
<<	Jumps back one page in the option list.
>>	Jumps forward one page in the option list.
ACTIVATE	Activates an option on the control, if the key code has been entered under **Keycode.**
CANCEL	Exits the SIK menu. If a software option has been enabled, the user also has to restart the control.

Overview of the options

See "Software options" on page 85.

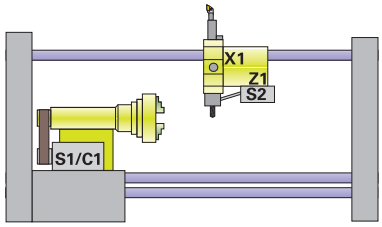
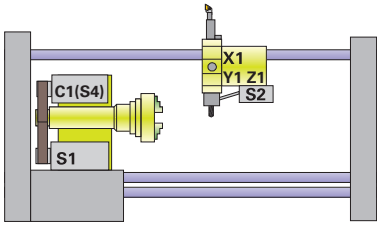
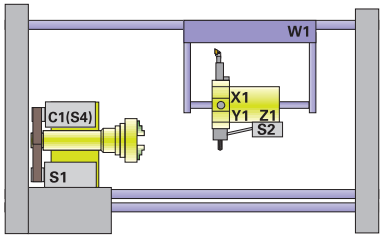
Interrogate options with PLC module

PLC Module 9067 can be used to interrogate the software options set in the SIK (see "Module 9067 Status of software settings" on page 676).

2.7.4 Configurations

The configuration of the MANUALplus 620 is designed for one slide (with X, Z, Y and W axes), spindle, C axis and driven tool. Use parameters to hide components not present on the machine. You also set the type of C-axis drive (separate C-axis drive or driven with spindle motor) via parameter. If there is a separate drive for the C axis, an additional control loop is required.

Please contact HEIDENHAIN if you require a different configuration.

Configuration of the MANUALplus 620	
Machine setup	**Axes and spindles**
	■ 2 spindles (spindle and driven tool) ■ 1 slide ■ 2 linear axes (X and Z axes) ■ 1 C axis (drive with spindle motor)
	Maximum expansion (6 control loops) ■ 2 spindles (spindle and driven tool) ■ 1 slide ■ 3 linear axes (X, Z and Y axes) ■ 1 C axis (drive with separate motor)
	Maximum expansion (7 control loops) ■ 2 spindles (spindle and driven tool) ■ 1 slide ■ 4 linear axes (X, Z, Y and W axes) ■ 1 C axis (drive with separate motor)

2.7.5 Coordinate system of the lathe

You specify the coordinate system of the lathe in the kinematics description.

The parameter MP_CoordSystem (in System/DisplaySettings/CfgCoordSystem) is relevant for the simulation, graphic representation in ICP and for the help graphics. The following settings are possible:

- +X / +Z: Horizontal lathe—turning behind the center
- –X / +Z: Horizontal lathe—turning in front of center
- +Z / +X: Vertical lathe—turning to the right of center

Note

- The NC software must be exchanged only by trained personnel.
- To enable the user to exchange the NC software, HEIDENHAIN provides a packed file with the NC software. For intermediate storage, the packed file is stored on a USB memory stick or a USB hard disk. Then it is installed on the control from the intermediate storage medium.
- HEIDENHAIN recommends making a backup of the control, for example with TNCbackup (included in TNCremoNT), before updating the NC software, See "Data backup" on page 106.

General information

- The **setup.zip** file is required for the software exchange. The installation program of the control automatically detects the file, which must be stored in the root directory of the data medium from which the update is run.

Note

HEIDENHAIN recommends:

Use a USB memory stick (1 GB or larger) to exchange the software. Do not use any memory stick with a smaller storage capacity.

- A backup of the control's most recent NC software is automatically created, while the software is being updated. The backup file is packed and stored on the memory stick. If the update is not completed successfully, your previous NC software version will automatically be restored. In addition, the complete update archive is automatically unpacked to the memory stick before installation. The memory stick must have enough free space for the backup and for unpacking the setup files. This is ensured by using a USB memory stick with a storage capacity of 1 GB.

Note

As a registered customer, you will receive the **setup.zip** file necessary for the update directly from HEIDENHAIN.
To do so, please write by e-mail to filebaseteam@heidenhain.de.

- Software updates and installation from service packs are loaded in the same manner.
- The NC software has been prepared in such a manner that when an update is performed or a service pack loaded, the PLC program, the machine configuration or data on the PLC partition can be updated as well, according to the OEM's specific requirements. The OEM uses the HEIDENHAIN PC software **PLCdesignNT** to add all necessary files to the **setup.zip** archive. These files are copied to the appropriate locations on the control during an update.

Note

For detailed information on how to add OEM specific files to the setup packet, refer to the online help for PLCdesignNT under "Generate Machine Setup." The support necessary for this from PLCdesignNT will be available starting from version 2.5.

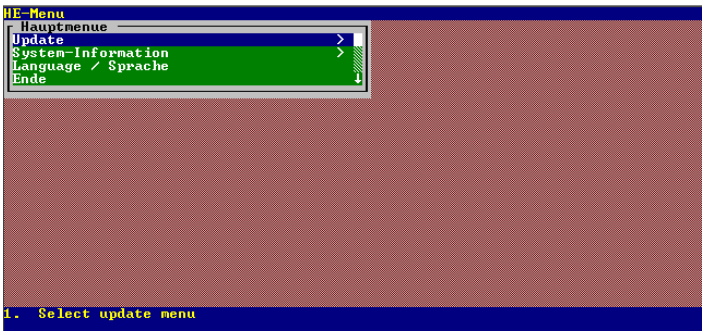
Procedure for exchanging the NC software

To install an NC software update, proceed as follows:

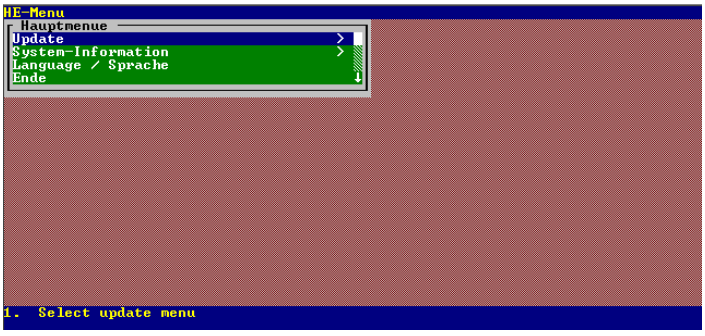
- If the machine is running, shut down the control by pressing the **OFF** soft key.
- Switch off the machine.
- If the new NC software is stored on a USB memory stick or a USB hard disk, connect the storage medium to a free USB socket (X141 or X142) on the MC 420.
- Switch the machine back on again.
- When the screen turns blue while the control is starting up, press the **DEL** key on the operating panel repeatedly. You must not hold down the DEL key. The control interrupts the boot process and a login message of the HEIDENHAIN operating system will be displayed:

- Enter **049866931** or **1** for "User name." If a USB keyboard is connected to the control, you can also enter **update** as an alternative.
- Press the **ENT** key or the RETURN key on the external USB keyboard.

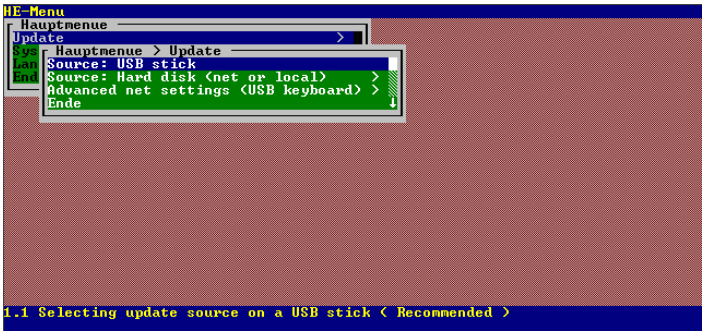
▶ The control starts the HE menu. The following window appears on the screen:



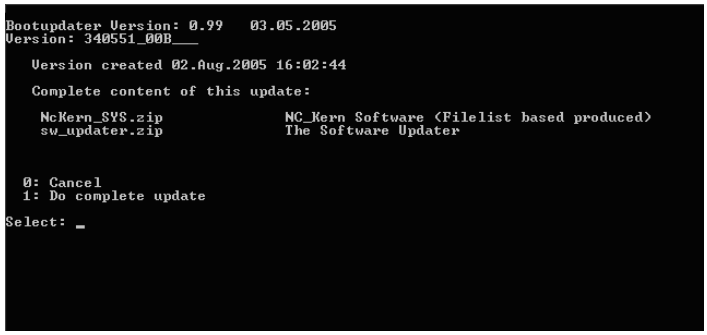
▶ Select the **Update** menu item and confirm your entry with the **ENT** key.



▶ Select the source on which the packed files for the software update are stored. If the setup files are stored on a USB stick, select **Source: USB stick**. Press the **ENT** key to confirm your selection.



- The control then starts the boot updater. A new window opens, which displays the contents of the update packet:



- Select **1** and confirm your selection with the ENT key.
- The control automatically performs the NC software update (approx. 20 to 30 minutes).
- After the update has been completed, the control requests you to remove the USB memory stick and then press **ENT**:

Remove the USB stick and then press ENT or Return!

- Remove the USB device (memory stick or hard disk) from the control.
- Press the **ENT** key or the RETURN key on the optional USB keyboard.
- The control is shut down and then restarts automatically.
- The NC software update is complete.

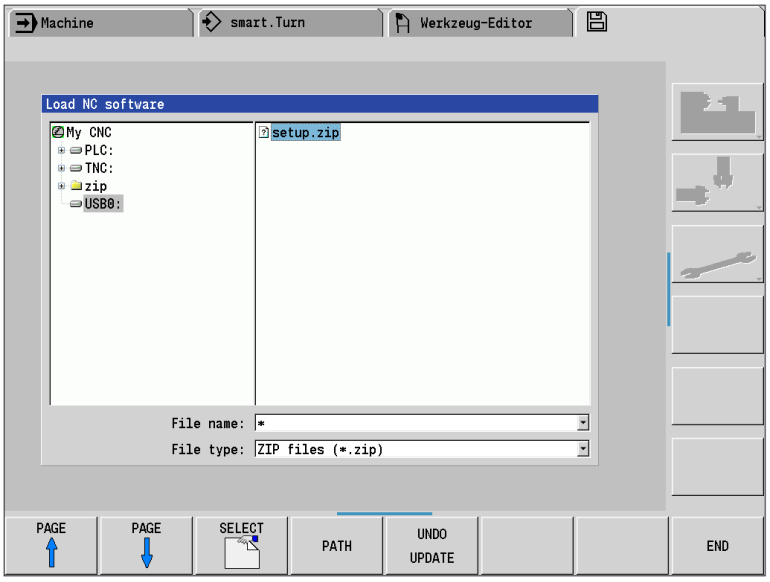
Start update while software is running on the control

As of SW version 548 328-02 you can update the NC software even while it is running on the control.

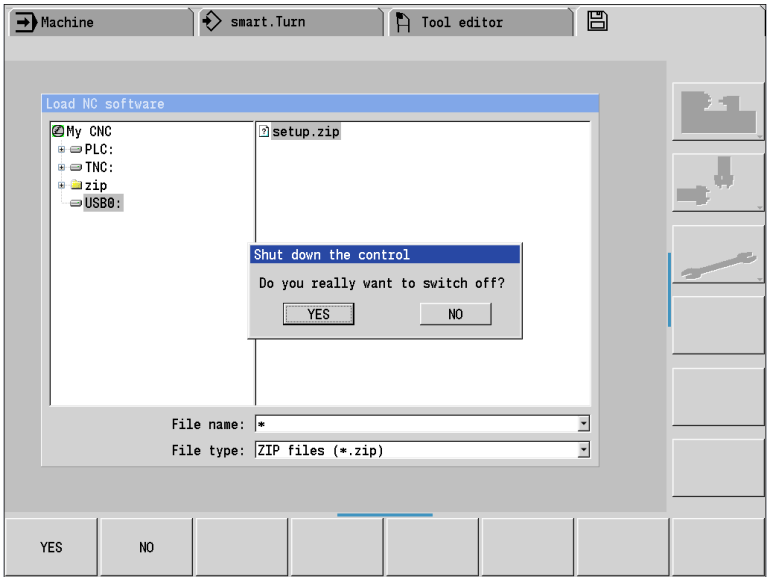
Note

Please note that the space on the CFR memory card is not sufficient for the MANUALplus 620 to open the Setup.zip file from the "TNC:" or "PLC:" drive. Always save the file on a USB memory stick with at least 512 MB free memory.
Plug the memory stick in a free USB socket if the control and start the update as described below.

- Switch to the **Organization** operating mode
- Press the  soft key.
- Enter the code number 231019.
- Press the **UPDATE DATA**  soft key.
- Then press the **LOAD ZIP**  soft key
- Press the **PATH** soft key to select the directory, in which the setup file is located, in the left window.



- Then press the **FILES** soft key to place the cursor in the right window on file level. Use the **SELECT** soft key to select the "setup.zip" file. The MANUALplus checks whether the selected setup file can be used for the current software version of the control.



- Confirm the confirmation request "Do you really want to switch off?". The NC software is now automatically shut down and then the actual update program is started.
- Select a dialog language.
- The next dialog field shows the old and new software. Confirm the confirmation request "Do you really want to install the software now?". Now the software update is started. The bar diagram displays the current progress of installation.
- After the software update is complete you will be prompted to restart the control. If you have used a USB stick, remove it before restarting the control.
- While the control is running up the dialog window for entering the code number appears. Enter here 95148.
- In the next step you are prompted to check the update rules. Press the **UPDATE RULES** soft key.
- To leave the update rules, press the **END** soft key.
- In the next step, check the configuration data by pressing the **CONFIG DATA** soft key.
- If parameters were changed, added or removed in the configuration data by the software update, this is marked by a red exclamation mark. Check all changed passages of the configuration data and press the **SAVE** soft key.
- Exit the software update by twice pressing the **ENDE** soft key, and after booting the control, conduct a restart.
- To finally conclude the software update, you have to confirm in a dialog window any changes of the firmware/hardware of the control; See "Monitoring hardware changes" on page 106.

Attention

When needed, HEIDENHAIN prepares service packs for the various versions of the NC software. To do so, please write as registered customer by e-mail to filebaseteam@heidenhain.de. Installation of a service pack in addition to the already installed NC software implements important error fixes. Please ensure that the NC software always contains the latest service pack before you ship the machine. Perform all tests required of the machine or the NC software again after having installed the service pack.

The latest service pack always includes all changes from earlier service packs.

When the control is started, a note regarding the installed service pack is shown.

HEIDENHAIN recommends always installing the latest released service pack!

Attention

If a service pack has already been installed, it will not be possible to install a service pack with a lower index. This will be checked during the installation of a service pack and a message will be displayed if an error is found.

Note

- As a registered customer, you will receive the file necessary for the service pack directly from HEIDENHAIN. The file name consists of the NC software number and the number of the service pack, e.g. **54832801sp1.zip**. To do so, please write by e-mail to filebaseteam@heidenhain.de.
- A service pack must be installed only by trained personnel.

The service pack consists of a packed file **(setup.zip).** For intermediate storage, the packed file is stored on a USB memory stick, for example. Then it is installed on the control from the intermediate storage medium.

Installing a service pack

A service pack is installed in the same manner as the NC software update. For instructions, please refer to "Procedure for exchanging the NC software" on page 95.

2.7.8 Reversing a software update

Under certain circumstances it may be necessary to reverse an NC software update. The prerequisite is that you still have the USB stick or the USB block device from which the update was carried out.

In addition to the **setup.zip** update file the control automatically saves several backup ZIP files of the last NC software on the USB drive during the software update. Together with the **setup.zip** file you can thus return to the previous software version.

Proceed as follows:

- Connect the USB stick or the USB drive with a vacant USB port on the control.
- Switch on the control.
- Interrupt the boot process by pressing the DEL key several times while the blue boot screen is shown; See "Procedure for exchanging the NC software" on page 95.
- The login message for the HEIDENHAIN operating system appears. Enter the code number **049866931** or **1**, or the keyword **update**. The control opens the HE menu.
- Open the **Update** > **Special features** > **Restore** > **RESTORE of setup.zip on USB stick** menu items, concluding each selection with the ENT key.
- If the control detects that the backup files for reversing the update exist, then an overview of the setup files stored on the USB stick appears. Otherwise an error message is displayed.
- Press the "1" key on the operating panel to start the reversal of the update.

Note

HEIDENHAIN recommends:

After you have updated your software, keep the USB stick used for this in a safe place. Then you can return to a previous software version if needed.

Note

Back up the NC programs or parameters you have created or changed after the installation of the last software update, before returning to the previous software version!
During this return to the previous software version, the data on the hard disk is replaced without a confirmation prompt! The exact state of the previous software is restored. Any changes or data added after the update was performed are lost.

Undo the update while the software is running on the control.

As of SW version 548 328-02 you can undo an NC software update even while the software is running on the control.

- Switch to the **Organization** operating mode
- Press the  soft key.
- Enter the code number 231019.
- Press the **UPDATE DATA**  soft key.
- Then press the **LOAD ZIP**  soft key
- Press the **PATH** soft key to select the directory, in which the setup file is located, in the left window, e.g. USB0:
- Then press the **FILES** soft key to place the cursor in the right window on file level.
- Press the **UNDO UPDATE** soft key.
- Use the **SELECT** soft key to select the "setup.zip" file. The MANUALplus checks whether it can use the selected setup file together with the REDO files saved in the same directory for restoring the previous software level.
- Confirm the confirmation request "Do you really want to switch off?". The NC software is now automatically shut down and then the actual update program is started.
- A dialog field prompts you to select a language.
- Confirm the confirmation request "Do you want to continue?".
- The next dialog box shows the previous and the currently installed software version and once again requests a confirmation for uninstalling the software. After your confirmation, a progress bar appears for the restoration of the previous software version.
- Then confirm the status message announcing the successful restoration of the software level and remove the USB stick. The control now automatically reboots.

2.7.9 Special features of the software

Firmware

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

Note

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

If the NC software detects that a firmware update is necessary, you will be prompted by a dialog box to start the firmware update. The firmware updates are included in the NC software update package. Please read the notes on firmware updates under„Firmware update on HSCI devices" on page 104. When the firmware versions of all HSCI participants have been updated, the control boots until the "Power interrupted" message appears.

Configuration

During startup, a control in an HSCI system expects the complete configuration of the HSCI system in the form of an IOC file. The IOconfig software for PCs is used to configure the system, the PLC input/output systems (PL6xxx, UEC11x, MB machine operating panel with a connected HR handwheel) and the PROFIBUS components.

With IOconfig, the components of an HSCI/PROFIBUS system are specified in a project and their arrangement is described. On the basis of this data, IOconfig generates the IOC file (*.IOC) that contains all relevant configuration data for the HSCI system. You then transfer the IOC file to the control. With the **MP_iocProject** machine parameter (103402) your enter the path and name of the project file (e.g. **%OEM%\IOC\Structur.ioc**). When the control is next booted, the file is read in and evaluated by the control.

Note

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

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

In addition to the configuration, the serial numbers of the individual HSCI components are also saved. During start-up they are compared with those of the detected hardware. Any difference must be checked and accepted by the user. (See "Monitoring hardware changes" on page106.)

The control also starts up without the IOC file or entry in the **MP_iocProject** (103402) parameter. The control then automatically detects all components connected to the HSCI chain and the current configuration—without the PLC input/output systems, however. Press the **Accept** button in the dialog box to confirm the detected configuration and continue start-up of the control. However, further configuration of the control is not possible. In the **Programming** mode you can start the BUS DIAGNOSIS by pressing the MOD key and the DIAGNOSIS soft keys. The control will display the structure of the detected HSCI system. If required, you can use this information for the configuration with IOconfig.

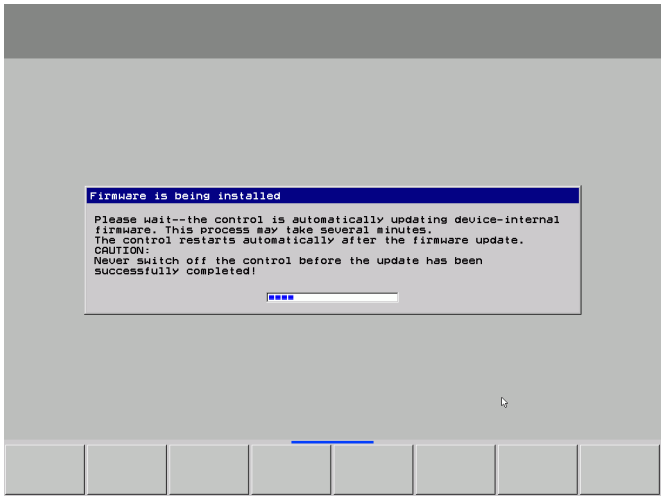
2.7.10 Firmware update on HSCI devices

During a software update or after installing a service pack, an update of the internal firmware of HSCI components may be required.

In this case, the control displays the following error message after start-up:

▶ Acknowledge the error message by pressing the **DELETE ALL** soft key.

The control now starts to install the new firmware on the HSCI devices. This procedure is displayed with the following dialog window:



Attention

Never shut down the control while a firmware update is running - otherwise the control will require servicing!

After the update is finished, the control displays another message on the screen. After you have acknowledged the message, the control will shut down automatically. Switch off the machines and restart the control.

2.7.11 Monitoring hardware changes

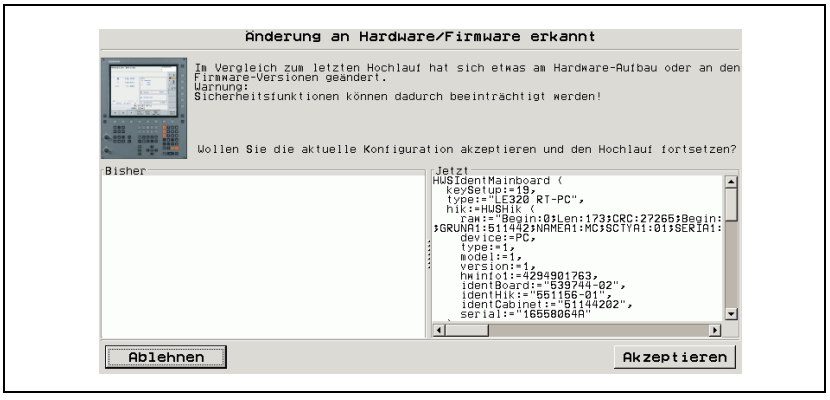
The MANUALplus 620 automatically monitors and checks whether hardware changes to control components took place. If the control detects a hardware change, it displays a dialog window after start-up. The operator must confirm the hardware change(s).

To accept a control hardware change:

- Press the right arrow key to navigate to the **Accept** button. Then press the ENT key:

To reject a control hardware change:

- Press the **Reject** button. The control will continue booting and start in the Programming Station mode. With the next start-up, the dialog window will be displayed again.



2.7.12 Data backup

For data backup, HEIDENHAIN offers the PC software **TNCbackup** free of charge. TNCbackup provides convenient functions for backing up and restoring data. TNCbackup is part of the TNCremoNT software package and can be downloaded from the HEIDENHAIN file base on the Internet. (http://filebase.heidenhain.de)

HEIDENHAIN recommends that the machine manufacturer use the TNCbackup software to save all his **machine-specific data** and to supply them with the machine (e.g. on CD-ROM). Do do so, use the PC software TNCremoNT to set up a connection with the PLC partition (drive O:) of the control and, in the "Extras/Backup" menu, start the TNCbackup software. In TNCbackup, under "File/Scan file list," select the "Scan directory tree" function to display and mark all directories and subdirectories of the PLC partition. With "Run/Backup" you can back up all marked files in the current file list. Before starting, TNCbackup asks you for the name of the backup file to be saved. In addition to the actual backup file *.BCK, the file list *.LST is saved. For a backup to be restored, the *.BCK file and the associated *.LST file must be located in the same folder.

The customer, too, should save his data before exchanging the control. It is also advisable that the customer save all of the files and programs created on the control at regular intervals on a PC. The CD-ROM supplied with the machine should therefore also contain the current version of TNCremoNT.

2.8 Software Releases

2.8.1 NC software 548 328-xx

NC software 548 328-01	Release: 05/2008 Initial version
NC software 548 328-02	Release: 07/2009
NC software 548 328-03	Release: 08/2010

3 Mounting and Electrical Installation

3.1 General Information

Attention

Keep the following in mind during mounting and electrical installation:

- National regulations for power installations
- Interference and noise immunity
- Conditions of operation
- Mounting attitude

3.1.1 Safety precautions

Danger

Ensure that the main switch of the control or machine is switched off when you engage or disengage connecting elements or connection clamps.

Danger

Ensure that the grounding conductor is connected. Interruptions in the equipment grounding conductor may cause damage to persons or property.

Danger

Incorrect or non-optimized input values can lead to faulty machine performance and therefore to serious injury to persons and damage to equipment. Modifications of the machine configuration should be done with caution and uncontrolled axis motions should be taken into account.

Attention

In order to be able to judge the behavior of an NC controlled machine, you need to have fundamental knowledge about drives, inverters, controls and encoders. Inappropriate use may cause considerable damage to persons or property.
HEIDENHAIN does not accept any responsibility for direct or indirect damage caused to persons or property through incorrect use or operation of the machine.

Danger

The interfaces for the PLC inputs/outputs, machine operating panel and PL connection comply with the requirements for basic insulation in accordance with **IEC 742 EN 50 178**.
Only units that comply with the requirements of **IEC 742 EN 50 178** for basic insulation may be connected, otherwise damage to persons or property may be caused. The maximum dc voltage mean value of the PLC inputs is 31 V.

3.1.2 Degrees of protection

The following components fulfill the requirements for IP54 (dust and splash-proof protection).

- Visual display unit (when properly installed)
- Keyboard unit (when properly installed)
- Machine operating panel (when properly installed)
- Handwheel

3.1.3 Electromagnetic compatibility

This unit fulfills the requirements for Class A according to EN 55022 and is intended for operation in industrially zoned areas.

Protect your equipment from interference by observing the following rules and recommendations.

Likely sources of interference

Interference is mainly produced by capacitive and inductive coupling from electrical conductors or from device inputs/outputs, such as:

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

Protective measures

- Keep a minimum distance of 20 cm from the control and its leads to interfering equipment.
- A minimum distance of 10 cm from the control and its leads to cables that carry interference signals. For cables in metallic ducting, adequate decoupling can be achieved by using a grounded separation shield.
- Shielding according to EN 50 178.
- Use potential compensating lines with a cross section of 6 mm^2
- Use only genuine HEIDENHAIN cables, connectors and couplings.

3.1.4 ESD protection

Always assume that all electronic components and assemblies are endangered by electrostatic discharge (ESD).

To ensure protection from ESD, follow the precautionary measures described in IEC 61340-5-1, IEC 61340-5-2 and IEC 61340-4-1.

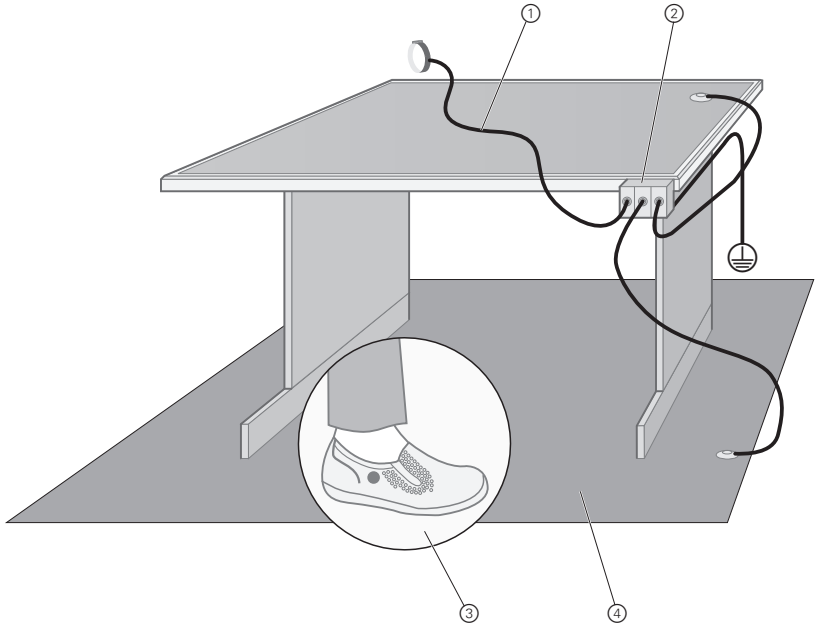
Note

Improper handling can result in damage to the components or assemblies due to ESD!

The following are some points covered in the above mentioned standards:

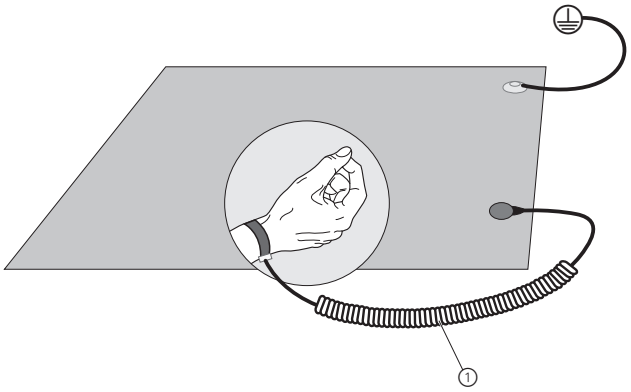
- When handling electrostatically endangered components or assemblies (e.g. exchange, installation, shipping), always comply with the precautionary measures described in IEC 61340-5-1, IEC 61340-5-2 and IEC 61340-4-1.
- Store and transport ESD-sensitive components in ESD protective containers.
- Ensure during handling the proper grounding of the working area (e.g. tool, workbench, packaging) and the person.
- Inspect the ESD protection system regularly.

The following figure shows how a suitable working area could look in accordance with IEC 61340.



- 1: Wristband with 1 MOhm grounding cable for grounding the person
- 2: Grounded connection for wristbands, floor mats, table mats etc. for equipotential bonding
- 3: Dissipative shoes
- 4: Dissipative flooring or floor mat

An important part of the working area is a suitable working surface with a wristband with 1 MOhm grounding resistance for personal grounding:



3.2 Environmental Conditions

3.2.1 Storage and operating temperatures

Limit values

Device	Air approaching the device in the panel / electrical cabinet	Temperature range outside the panel / electrical cabinet
MC 6110T	0°C to +50°C	0°C to +45°C (no direct exposure to sunlight)
MB 6xx, TE 6xx	0°C to +50°C	0°C to +45°C
HR 4xx		0°C to +45°C
PLB 6xxx, PLD-H, PLA-H	+5°C to +40°C	
CompactFlash card	0°C to +70°C	0°C to +45°C (no direct exposure to sunlight)

Limit value for temperature inside the panel

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

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

Attention

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

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

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

Humidity during operation

Attention

Condensation on the electronics is not permitted!

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

In tropical areas it is recommended that the control not be switched off, so that condensation is avoided on the circuit boards.

Storage temperatures

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

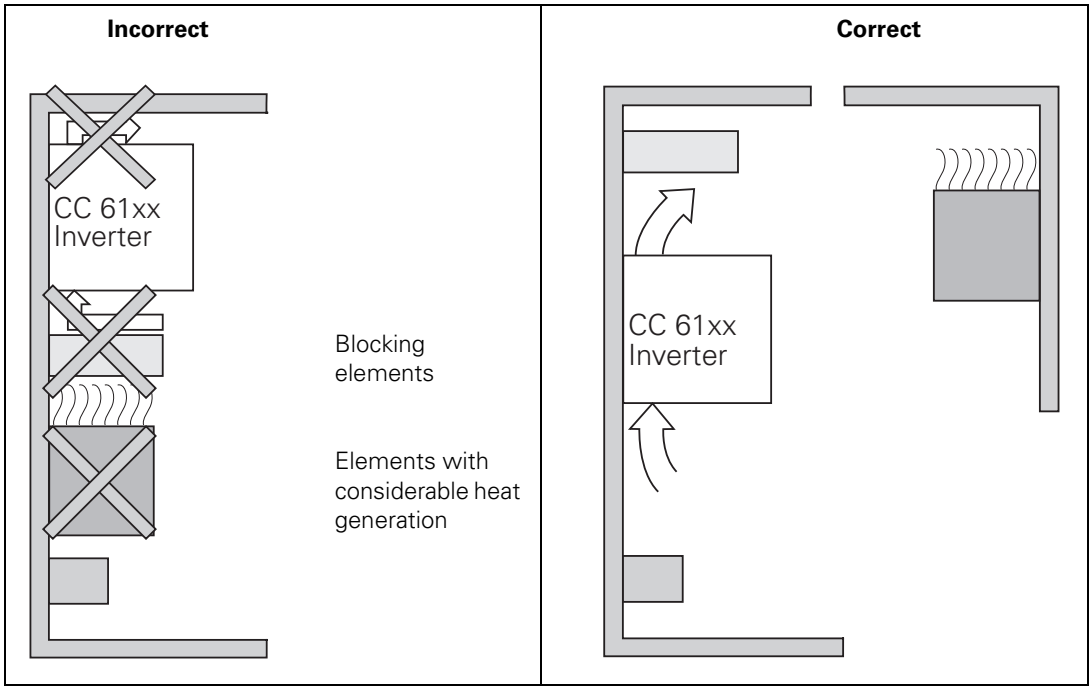
3.2.2 Heat generation and cooling

A heat exchanger or a cooling unit is preferable for controlling the internal temperature of the electrical cabinet.

If filtered air is blown into the electrical cabinet for cooling purposes, the standard IEC 61800-5-1 applies, which permits contamination level 2.

Danger

Be sure to take the measures required for preventing dust from entering the electrical cabinet.
Dust depositing inside electrical devices may cause them to fail and impair the safety of the system.



3.2.3 Limit values for ambient conditions

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

Furthermore, the following limit values apply:

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

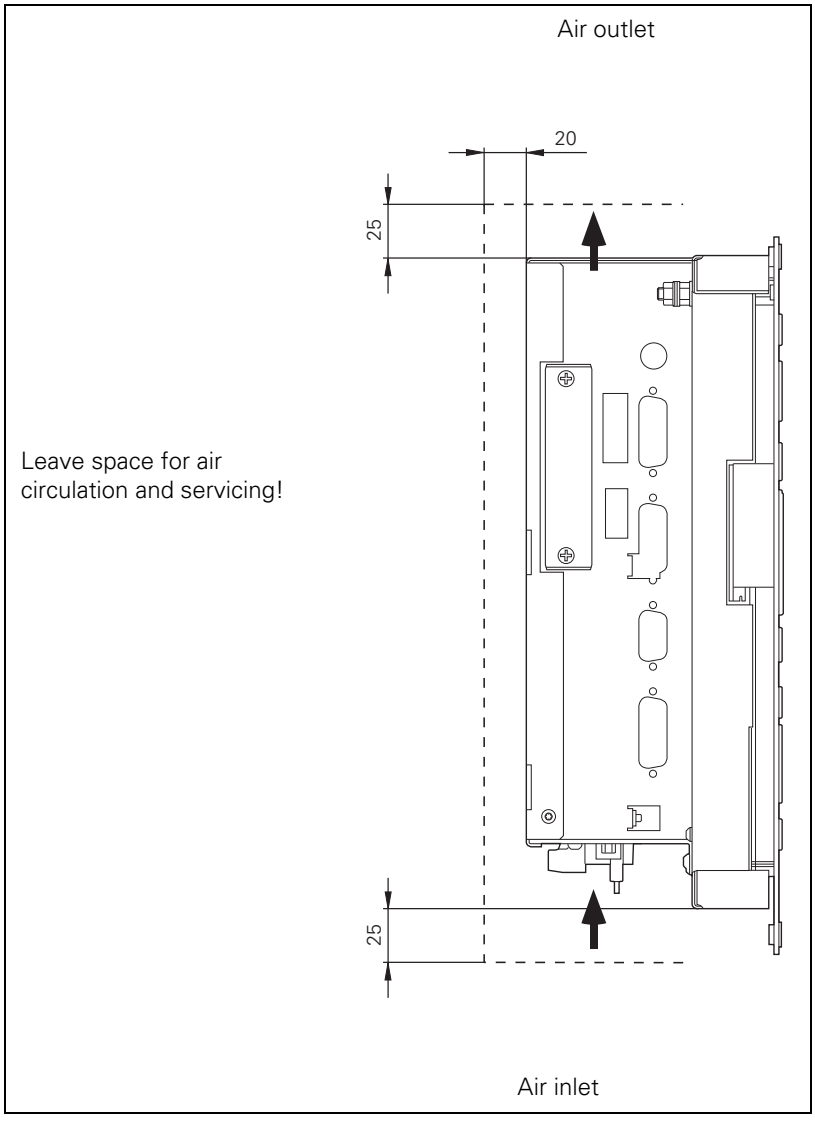
3.2.4 Installation elevation

The maximum installation height is 3000 m above sea level.

3.2.5 MC6110T mounting position

Attention

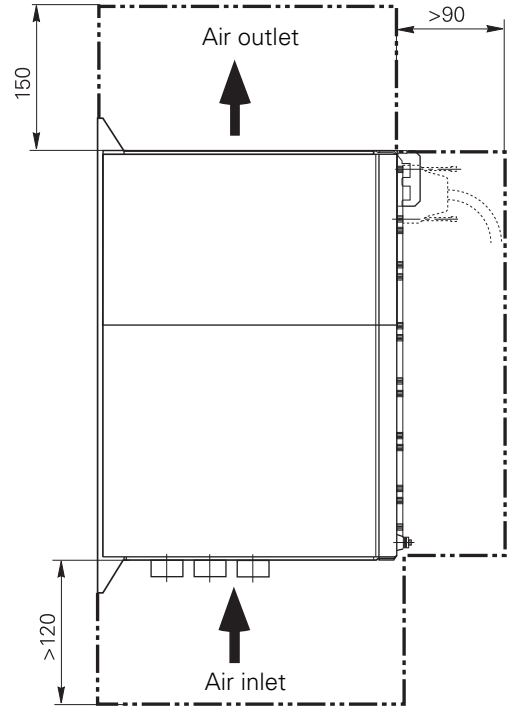
When mounting, please observe proper minimum clearance, space requirements, length and position of the connecting cables.



3.2.6 Mounting attitude of CC 61xx, UV xxx, UM xxx, UE 2xx B

Attention

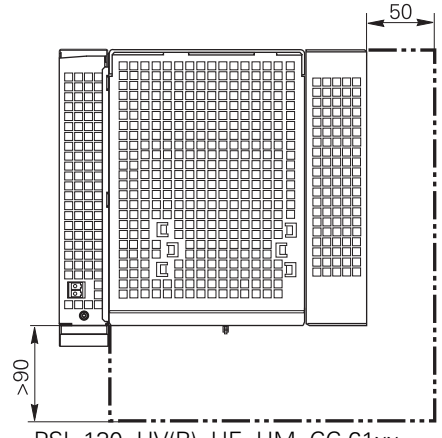
When mounting, please observe proper minimum clearance, space requirements, length and position of the connecting cables.



Leave space for air circulation! Temperatures of > 150 °C are possible with UE 21xB with integral braking resistor; Do not mount any temperature-sensitive components!

Leave space for servicing!

Leave space for air circulation and servicing!



PSL 130, UV(R), UE, UM, CC 61xx

Leave space for servicing and connecting cables!

3.3 Overview of Components

Hardware components		ID
MC 6110T	Main computer (compact) 1 GHz	731 604-xx
CFR MANUALplus 620	Memory card for MC 6110T with MANUALplus 620 software	733 606-51
SIK MANUALplus 620	NC software license for MC 6110T, 3 control loops	733 604-53
CC 6106	Controller unit for HSCI for max. 6 control loops	662 636-xx
CC 6108	Controller unit for HSCI for max. 8 control loops	662 637-xx
UEC 111	Controller unit with inverter and PLC, 4 control loops	625 777-xx
UEC 112	Controller unit with inverter and PLC, 5 control loops	625 779-xx
PSL 130	Low-voltage power supply unit, 750 W for +24 V NC and +24 V PLC	575 047-xx
PSL 135	Low-voltage power supply unit, 750 W, for +24 V NC and +24 V PLC, also suitable for non-HEIDENHAIN systems	627 032-xx
MB 620	Machine operating panel for HSCI connection	617 973-xx
MB 620T	Machine operating panel for HSCI connection with spindle and feed rate override potentiometers	737 610-xx
PLB 6001	HSCI adapter for OEM-specific machine operating panel, 64 digital inputs, 32 digital outputs	668 792-xx
PLB 6104	PLB for HSCI, 4 slots	591 828-xx
PLB 6106	PLB for HSCI, 6 slots	630 058-xx
PLB 6108	PLB for HSCI, 8 slots	630 059-xx
PLB 6204	PLB for HSCI, 4 slots, with system module	591 832-xx
PLB 6206	PLB for HSCI, 6 slots, with system module	630 054-xx
PLB 6208	PLB for HSCI, 8 slots, with system module	630 055-xx
PLD-H 16-08-00	PL for PLB 6xxx: 16 digital inputs, 8 digital outputs	594 243-xx
PLD-H 08-16-00	PL for PLB 6xxx: 8 digital inputs, 16 digital outputs	650 891-xx
PLA-H 04-00-04	PL for PLB 6xxx, 4 ± 10 V inputs, 0 analog outputs, 4 PT 100 inputs	599 070-xx
PLA-H 08-04-04	PL for PLB 6xxx, 8 x ±10 V inputs, 4 x ±10 V analog outputs, 4 x PT 100 inputs	675 572-xx
PL empty housing	Empty housings for slots of a PL 6xxx	383 022-11
HSCI cable	HSCI connecting cable	618 893-xx

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

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

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

Device designation	**Device ID for systems with integrated FS**	**Device ID for systems without integrated FS**
Inverter modules		
UM 116D	667954-01	542998-01
UM 116DW	667946-01	369629-01 index B
UM 115D	671566-01	387852-01 index E
UM 114D	671288-01	510509-01 index E
UM 113D	730435-01	518703-01 index B
UM 112D	731984-01	519971-01 index C
UM 122D	667633-01	519972-01 index C
UM 121BD	667942-01	513037-01 index C
UM 111BD	671968-01	513035-01 index E
UM 121D	667838-01	392319-01 index F
UM 111D	667945-01	392318-01 index F
Power supply modules		
UVR 120D	728252-01	390188-01 index K
UV 130D	728250-01	389311-01 index E
UVR 130D	728248-01	377639-01 index K
UVR 140D	728253-01	390281-01 index N
UVR 150D	728255-01	390421-01 index P
UVR 160D	728257-01	530341-01 index G
UVR 160DW	728258-01	560106-01 index G
Non-regenerative compact inverters		
UE 210D	733 421-01	558302-01 index C
UE 211D	733 423-01	558303-01 index C
UE 212D	733 424-01	558304-01 index C
UE 230D	733 425-01	558305-01
UE 240D	733 426-01	558306-01
UE 241D	733 427-01	558307-01
UE 242D	733 428-01	558308-01
UEC 111(FS)	665 630-xx	625 777-xx

Device designation	Device ID for systems with integrated FS	Device ID for systems without integrated FS
UEC 112(FS)	665 629-xx	625 779-xx
UE 110	Not yet available	375713-02 index B
UE 111	Not yet available	375714-02 index B
UE 112	Not yet available	375715-02 index B
Regenerative compact inverters		
UR 242D	Not yet available	536565-01 index A
UR 230D	Not yet available	536561-01 index A
UR 240D	Not yet available	536564-01 index B

With the following HSCI control components, you must also make a distinction between those that are required in a system with functional safety and those that can be used in a system without functional safety.

Device designation	Device ID for systems with integrated FS	Device ID for systems without integrated FS
Machine operating panels and keyboard units In FS systems, you must use a machine operating panel for FS applications. In these operating panels, all keys have twin channels. A movement can therefore be executed without additional permissive button.		
MB 620 (FS)	660 090-xx	617 973-xx
TE 635Q (FS)	662 255-xx	617 975-xx
PLB basic modules In systems with FS, mixed use of PLC basic modules with and without FS is possible. However, at least one PLB 62xxFS must be used in systems with FS.		
PLB 6104 (FS)	590 479-xx	591 828-xx
PLB 6204 (FS)	586 789-xx	591 832-xx
PLB 6206 (FS)	622 721-xx	630 054-xx
PLB 6208 (FS)	620 927-xx	630 055-xx
PLB 6001 (FS)	Not yet available	668 792-xx
PLD-H I/O modules In FS systems, the mixed use of PLD-H modules with and without FS is possible in PLB basic modules with FS. However, PLD-H modules with FS must not be inserted in PLB basic modules without FS. Furthermore, the modules with Functional Safety must always be inserted into the PLB with Functional Safety starting from the left.		
PLD-H 16-08-00, PLD-H 08-04-00 FS	598 905-xx	594 243
PLD-H 08-16-00, PLD-H 04-08-00 FS	727 219-xx-xx	650 891-xx

The other control components MC 6xxx, CC 6xxx, PSL 13x, MS 1xx, BF 2xx can be used regardless of whether the system is a system with or without Functional Safety.

3.4 HSCI

3.4.1 Introduction

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

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

The logic area of the control system is supplied with power via two separate power loops:

- The MC, MB, PLs and BF are supplied with power via the PSL 130 power supply unit. The voltages of the PSL 130 correspond to the PELV system as per EN 50178.
- The logic voltages of the CC are voltages derived from the dc-link voltage of the inverters, and are transmitted via connector X69 "NC supply voltages and control signals." These voltages also correspond to the PELV as per EN 50178.

The following table shows the maximum permissible number of individual HSCI participants:

HSCI component	Maximum number
MC (HSCI master)	1 in the system
CC (HSCI slave)	4 controller basic boards, (distributed to the CCs 61xx as desired)
PLB 62xx (FS) or UxC 11x (HSCI slave with PLC system module)	1 in the system
PLB 61xx (FS) (HSCI slave)	7 in the system
PLD-H xx-xx FS (in PLB 6xxx FS)	8 in the system
PLD-H xx-xx (in PLB 6xxx (FS))	64 in the system

The order of the HSCI participants at X500 of the main computer is freely selectable. Each HSCI participant is assigned its HSCI address based on its position in the HSCI chain:

- 1. Second device after the master (main computer): Bus address 1
- 2. Second device after the master (main computer): Bus address 2, etc.

The main computer selects a master controller unit, based on the position of the controller units in the HSCI network. The first controller unit in the system (nearest the main computer) becomes the master controller unit.

3.4.2 Topology

The HSCI slaves are connected—in series—to connector X500 of the main computer. Connector X502 is always the HSCI input to the HSCI slaves and X500 the HSCI output to the next HSCI slave.

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

Terminating resistors are not required in the HSCI system.

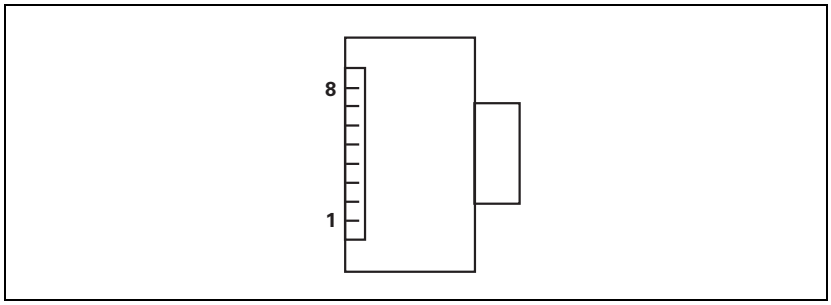
The HSCI network is to be structured as an "open" ring (line structure). Connector X500 on the main computer is the beginning of the HSCI network. The HSCI connection is then led to X502 of the next HSCI participant (CC, UxC, PL or MB), where X500 again serves as the output for continuing the HSCI network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

Note

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

3.4.3 HSCI interface

Face of the connector:



X500
X501
X502

Pin layout:

The order of the HSCI participants at X500 of the MC is freely selectable. Each HSCI participant is assigned its HSCI address based on its position in the HSCI chain, see page 123.

Pin layout of the HSCI cable:

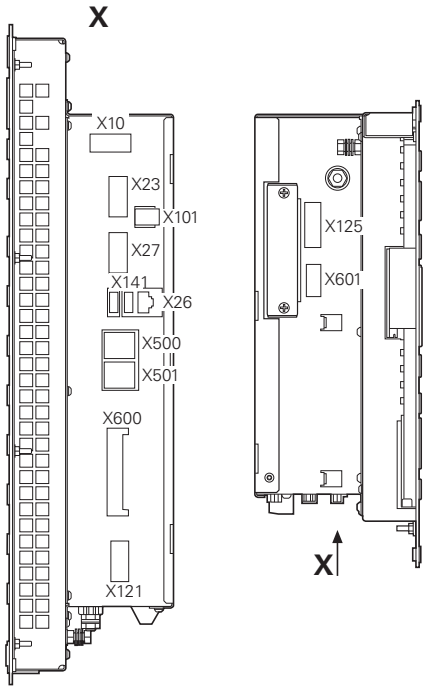
ID 618893-xx			
Female	**Color**	**Pin layout**	**Female**
1	White/Green	Data	1
2	Green	Data	2
3	White/Orange	Data	3
4	Vacant	Vacant	4
5	Vacant	Vacant	5
6	Orange	Data	6
7	Vacant	Vacant	7
8	Vacant	Vacant	8

3.5 Connection OverviewMANUALplus 620

3.5.1 MC 6110T main computer

MC 6110T, compact main computer, integrated screen and keyboard, HSCI interface

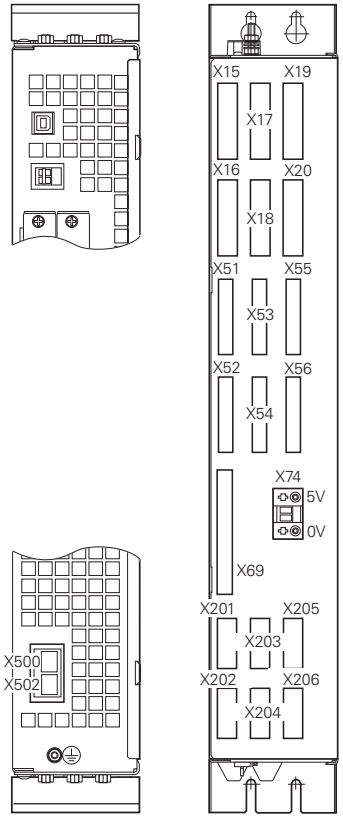
Connection overview



Connector	Function	Page
X3	Screen soft keys	–
X10	Feed rate/spindle override potentiometer	199
X26	Ethernet data interface	191
X27	RS-232-C/V.24 data interface	193
X29	Reserved, do not assign	–
X101	Power supply for NC, 24 V–	136
X116	Reserved, do not assign	–
X121	Profibus (only on MC 6120 with ID 680 391-xx)	–
X125	SIK (System Identification Key)	52
X141 X142	USB interface	194
X500	HSCI output 1 (synchronized)	124
X501	HSCI output 2	124
X600	CompactFlash Removable CFR	52
X601	Reserved, do not assign	–
⏚	Protective ground	–

Attention

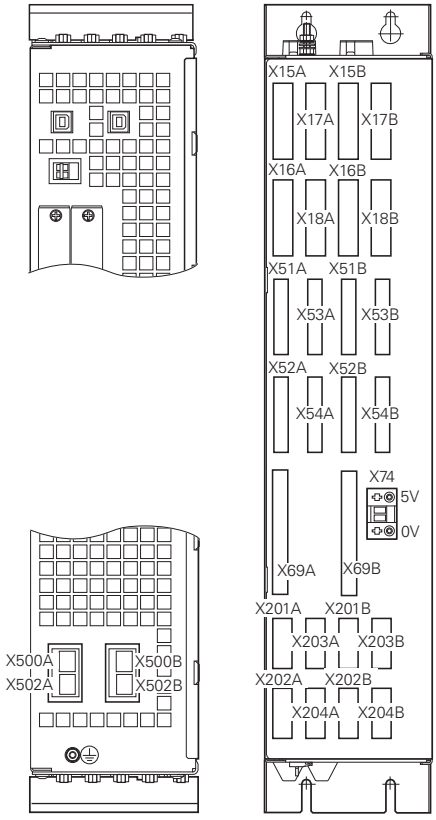
Do not engage or disengage any connecting elements while the unit is under power!

CC 6106, controller unit with 6 control loops and HSCI interface			
Pin layout	**Connector**	**Function**	**Page**
	X15 to X20	Speed encoder	178
	X51 to X56	PWM output	187
	X69	Supply bus	138
	X201 to X206	Position encoder	175
	X500	HSCI output	124
	X502	HSCI input	124
	–	SPI slot 1 (on bottom, reserved for expansion modules)	–
	–	SPI slot 2 (on bottom, reserved for expansion modules)	–
	X74	+ 5 V supply	139
	X7	Bridge for signal ground (= functional ground) (on bottom)	139
	⏚	Protective ground	–

Attention

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

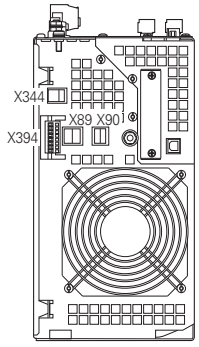
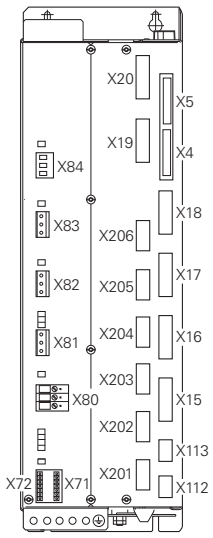
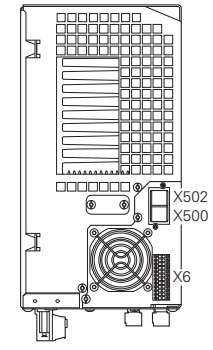
CC 6108 controller unit with 8 control loops and HSCI interface

Pin layout	Connector	Function	Page
	X15A - X18A	Speed encoder Drive-control main board A	178
	X15B - X18B	Speed encoder Drive-control main board B	178
	X51A - X54A	PWM output Drive-control main board A	187
	X51B - X54B	PWM output Drive-control main board B	187
	X69A	Supply bus Drive-control main board A	138
	X69B	Supply bus Drive-control main board B	138
	X201A - X204A	Position encoder Drive-control main board A	175
	X201B - X204B	Position encoder Drive-control main board B	175
	X500A	HSCI output Drive-control main board A	124
	X502A	HSCI input Drive-control main board A	124
	X500B	HSCI output Drive-control main board B	124
	X502B	HSCI input Drive-control main board B	124
	X74	+ 5 V supply	139
	–	SPI slot 1 (on bottom, reserved for expansion modules)	–
	–	SPI slot 2 (on bottom, reserved for expansion modules)	–
	X7	Bridge for signal ground (= functional ground) (on bottom)	139
	⏚	Protective ground	–

Attention

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

UEC 11x: Compact controller unit with integrated inverter and PLC I/Os
(**without** Functional Safety (FS)

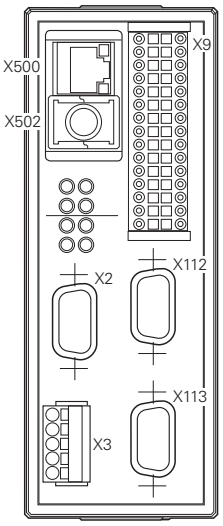
Pin layout	Connector	Function	Page
	X4, X5	PLC inputs	164
	X6	PLC outputs	167
	X15 to X19	Speed encoder	178
	X31	Supply voltage for UEC 11x (3 x 400 V ± 10%)	150
	X71	Spindle safety relay (pulse inhibitor for spindle)	152
	X72	Axes safety relay (pulse inhibitor for axes)	152
	X80	Motor connection for spindle (24 A rated current at 3.3 kHz)	152
	X81	Motor connection axis 1 (6 A rated current at 3.3 kHz)	152
	X82	Motor connection axis 2 (6 A rated current at 3.3 kHz)	152
	X83	Motor connection axis 3 (9 A rated current at 3.3 kHz)	152
	X84	Motor connection axis 4 (6 A rated current at 3.3 kHz)	152
	X89	Braking resistor	152
	X90	24 V NC output / 3.5 A	152
	X112	TS touch-trigger probe	188
	X113	TT touch-trigger probe	188
	X201 to X205	Position encoder	175
	X344	24 V supply for motor holding brake	153
	X394	Motor holding brake 1 to 4	153
	X500	HSCI output	124
	X502	HSCI input	124
	⏚	Protective ground M5	–

Attention

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

3.5.5 PLB 62xx

System module

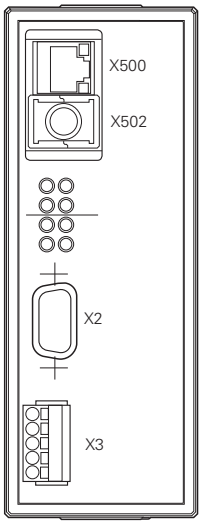
Pin layout	Connector	Function	Page
	X500	HSCI output	124
	X502	HSCI input	124
	X9	Safety-related PLC inputs/outputs	160
	X2	Reserved	–
	X3	+ 24 V NC, +24 V PLC power supply	157
	X112	TS or TT touch trigger probe	188
	X113	TS or TT touch trigger probe	188

Diagnosis (meanings of the LEDs):

- Green (LEDs to the right): status of PL reports OK
- Yellow (LEDs to the left): group message with error to PL
- Alternating flashing of green and yellow LEDs in top row indicates a faulty HSCI connection

3.5.6 PLB 61xx

Expansion module

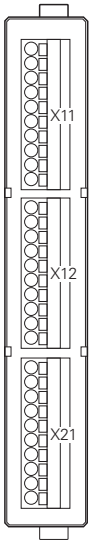
Pin layout	Connector	Function	Page
	X500	HSCI output	124
	X502	HSCI input	124
	X2	Reserved	–
	X3	+ 24 V NC, +24 V PLC power supply	157

Diagnosis (meanings of the LEDs):

- Green (LEDs to the right): status of PL reports OK
- Yellow (LEDs to the left): group message with error to PL
- Alternating flashing of green and yellow LEDs in top row indicates a faulty HSCI connection

3.5.7 I/O modules PLD-H and PLA-H

PLD-H 16-08-00

Pin layout	Connector	Function	Page
	X11	PLC inputs, channel A	162
	X12	PLC inputs, channel A	162
	X21	PLC outputs, channel A	163

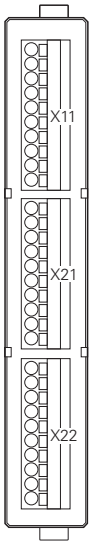
Diagnosis (meanings of the LEDs):

- Red (X11/pin 1) status LED
  - Flashes: status of I/O module OK
  - Permanently on or off: error on I/O module
- Yellow (per output): Status of the output

Error recognition:

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

PLD-H 08-16-00

Pin layout	Connector	Function	Page
	X11	PLC inputs, channel A	162
	X21	PLC outputs, channel A	163
	X22	PLC outputs, channel A	163

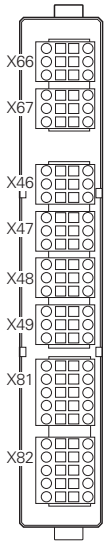
Diagnosis (meanings of the LEDs):

- Red (X11/pin 1) status LED
  - Flashes: status of I/O module OK
  - Permanently on or off: error on I/O module
- Yellow (per output): Status of the output

Error recognition:

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

PLA-H 08-04-04

Pin layout	Connector	Function	Page
	X66 to X67	±10 V analog outputs	170
	X46 to X49	±10 V analog inputs	169
	X81 to X82	PT 100 analog inputs	170

3.6 Supply Voltages in the HSCI System

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

HEIDENHAIN recommends using the dc-link buffered PSL 13x for supplying power to the HSCI components, see "PSL 130 low-voltage power supply unit" auf Seite 140.

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

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

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

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

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

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

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

- PLB 6xxx input/output module
- PLB6xxx input/output module
- PLD-H digital plug-in module for PLB
- PLA-H analog plug-in module for PLB
- Motor holding brakes, further components in the PLC circuit

Note

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

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

Danger

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

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

Note

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

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

Note

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

3.6.1 X90: +24 V NC output of the UxC 11x (FS)

Pin layout

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

Load capacity

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

Current consumption of the HSCI components

Device	Current consumption of the 24 V NC supply
MC 6110T	1.5 A
MB 620T	1.0 A
PL 62xx (incl. TS and TT)	0.5 A
PL 61xx	0.2 A

Example: MANUALplus620 configuration with UEC 11x

Device	Current consumption 24 V NC
MC 6110T	1.5 A
MB 620T	1.0 A
PL 61xx	0.2 A
Total	2.7 A < 3.5 A

A PSL 130 unit is not needed for this application. The +24 V NC supply of the UEC 11x (X90) suffices for the connected components.

3.6.2 X101: NC power supply

The MC main computer is supplied with +24 V NC (control voltage) of the machine, for example by the PSL 130, See "PSL 130 low-voltage power supply unit" on page 140 or by the integrated 24 V power supply unit of the UEC 11x (FS).

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

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

Pin layout:

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

Attention

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

Power consumption:

Device	Power consumption
MC 6110T	35 W

Efficiency: 85%

Note

If USB components that are connected to X141/X142 require more than 0.5 A, a separate power supply becomes necessary for these components. One possibility is the USB hub from HEIDENHAIN (582 884-02).

3.6.3 Power supply of the CC61xx

The CC 61xx controller unit is supplied with a power of **+5 V** by the power supply units via supply bus X69 and connector X74, see page 138.

The control monitors the 5 V supply voltage. If it falls below 4.75 V, the error message **`5 V power supply too low`** appears. If it rises above 5.4 V, **`5 V power supply too high`** is indicated.

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

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

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

a. For cable lengths > 10 m between the logic unit and the encoders with EnDat interfaces, a line drop compensator is required (efficiency = 75%).

X69: CC- supply voltage and control signals

Pin layout:

50-pin ribbon connector	Pin layout	50-pin ribbon connector	Pin layout
1a to 5b	+5 V	16b	GND
6a to 7b	+12 V	17a	RDY.PS
8a	+5 V (low-voltage separation)	17b	GND
8b	0 V (low-voltage separation)	18a	$\overline{\text{ERR.ILEAK}}$
9a	+15 V	18b	GND
9b	–15 V	19a	$\overline{\text{PF.PS.AC}}$ (only UV 120, UV 140, UV 150, UR 2xx)
10a	UZAN	19b	GND
10b	0 V	20a	Do not assign
11a	IZAN	20b	GND
11b	0 V	21a	Do not assign
12a	$\overline{\text{RES.PS}}$	21b	GND
12b	0 V	22a	Do not assign
13a	$\overline{\text{PF.PS.ZK}}$	22b	GND
13b	GND	23a	Reserved (SDA)
14a	$\overline{\text{ERR.UZ.GR}}$	23b	GND
14b	GND	24a	Reserved (SLC)
15a	$\overline{\text{ERR.IZ.GR}}$	24b	GND
15b	GND	25a	RES.LE
16a	$\overline{\text{ERR.TMP}}$	25b	GND

X74: +5 V power supply for CC

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

Attention

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

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

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

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

X7: Bridge for signal ground (= functional ground)

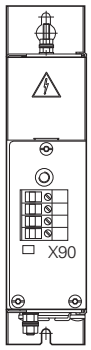
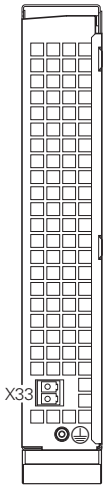
Connecting terminal X7	Pin layout
1	Connection for signal ground (= functional ground)
2	Connection on housing

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

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

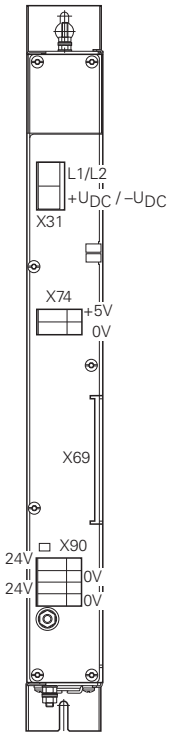
If there are two or more CC 61xx units powered over more than one UV power module and are therefore connected with different supply bus systems (X69), then the external signal-ground bridge is to be connected with only one each CC 61xx of the respective supply bus. In order to prevent ground loops, disengage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

3.6.4 PSL 130 low-voltage power supply unit

PSL 130 pin layout	Connector	Function
	Conductor bar	Connection of DC-link voltage Uz
	B – Signal ground (= functional ground)	Signal ground (0 V signal of the +24 V NC signal connected internally to protective ground)
	X90	Output for control voltages: ■ Terminal 1: +24 V– NC ■ Terminal 2: 0 V NC (ground +24 V– NC) ■ Terminal 3 + 24 V– PLC ■ Terminal 4: 0 V PLC (ground +24 V– PLC)
	X33	Input voltages L1, L2
	⏚	Protective ground

Attention

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

PSL 135 pin layout	Connector	Function
	X31	Input voltages L1, L2 and connection of the DC link voltage Uz
	B – Signal ground (= functional ground)	Signal ground (0 V signal of the +24 V NC signal connected internally to protective ground)
	X74	Output for control voltages: ■ Terminal 1: +5 V ■ Terminal 2: 0 V
	X69	Power supply and control signals for CC 61xx (for X69 on CC)
	X90	Output for control voltages: ■ Terminal 1: +24 V– NC ■ Terminal 2: 0 V NC (ground +24 V– NC) ■ Terminal 3 + 24 V– PLC ■ Terminal 4: 0 V PLC (ground +24 V– PLC)
	⏚	Protective ground

Attention

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

General information

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

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

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

The +24 V PLC voltage is also electrically separated by the power supply unit. However, because of the connection to the PLC part of the HSCI system this is a supply voltage with basic insulation. This voltage must not be linked with other voltages.

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

Danger

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

Specifications

Specifications	PSL 130	PSL 135
Power supplies: at X33 (L1, L2)	400 V~ ± 10% 50 Hz	
Power supplies: DC-link power bar or X31	400 V– to 750 V–	
Protection	Page 145	
Output voltages: Accuracy of the +24 V NC Accuracy of the +24 V PLC Accuracy of the +5 V NC	 +24 V NC: +/- 5 % +24 V PLC: Variations depending on the load, between 20 V and 28 V - - -	 +24 V NC: +/- 5 % +24 V PLC: Variations depending on the load, between 20 V and 28 V +5 V NC: +/– 5% (power supply unit is adjusted to 5.2 V)
Output power	24 V NC output: max. 500 W 24 V PLC output: max. 500 W Total power output: max. 750 W	24 V NC output: max. 350 W 24 V PLC output: max. 500 W 5 V NC output: max. 100 W +/–15 V at X69: max. 50 W Total power output: max. 750 W
24 V NC output current	Max. 20 A	Max. 14.5 A
24 V PLC output current	Max. 20 A	Max. 20 A
5 V NC output current	- - -	Max. 20 A
24 V output current total	Continuous load: max. 31 A Max. load time 1 s: max. 41 A Max. load time 0.1 s: max. 46 A The outputs are short-circuit proof and switch off automatically when overloaded.	
5 V output current	- - -	Max. 20 A The outputs are short-circuit proof and switch off automatically when overloaded.
Power consumption	Max. 1000 W	
Degree of protection	IP 20	
Module width	50 mm	
Weight	2.1 kg	
ID	575 047-xx	627 032-xx

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

Danger

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

U_Z DC-link current

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

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

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

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

Length	**For connection to**	**ID (conductor bar set)**
150 mm	UVR 120D/130D	687 669-01
200 mm	UVR 140D/150D	687 670-01
250 mm	UVR 160D/160DW	687 671-01

Note

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

X33: Input voltage of the PSL 130

Supply voltage: 400 V ± 10%

Connection:

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

Through the connection to the 400 V AC line voltage (L1, L2) via X33, the output voltages of the PSL 130 are available as soon as the machine's main switch has been turned on. The connection of L1 and L2 is absolutely essential for booting the control.

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

Note

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

Operation of the PSL 130 is possible with connection to the U_z line voltage, e.g as PLC supply in combination with the UEC 11x.

However, be aware that operation with the U_z dc-link voltage but without the power connection (L1 and L2) at X33 is not possible. The line voltage is necessary for the PSL 130 to start correctly.

X31: Input voltage of the PSL 135

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

Connection:

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

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

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

Note

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

X90: Output voltage of the PSL 13x

Output voltages: + 24 V (2 x)

Connection:

Note

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

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

X74: Output voltage of the PSL 135

Output voltages: + 5 V

Connection:

Note

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

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

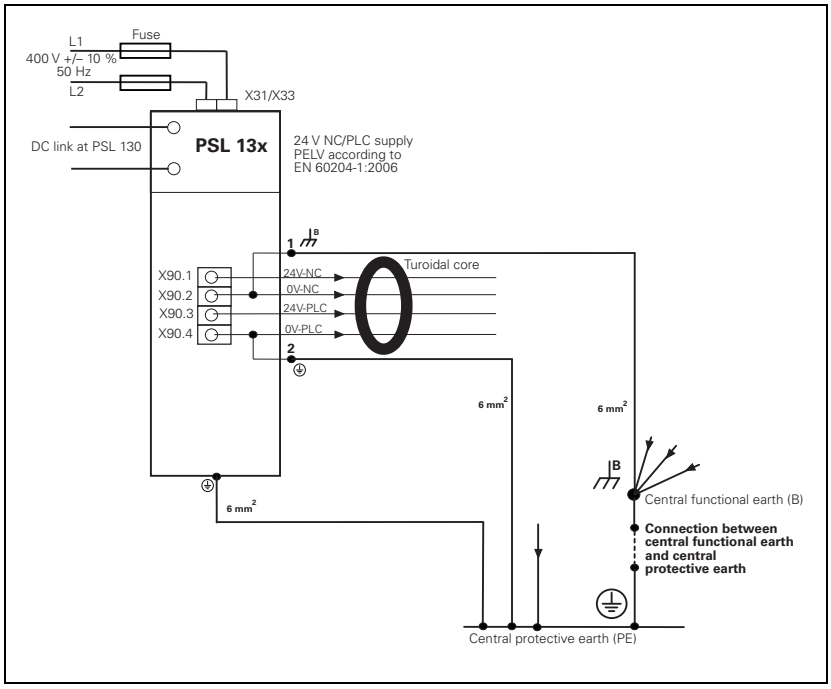
Signal ground (= functional ground)

Connections for signal ground, which are connected internally in the PSL 130 to the 0 V NC and 0 V PLC signals

Connection:

Note

Conductor cross section of at least 6 mm2 for connecting to signal ground (= central functional ground).



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

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

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

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

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

3.7 UxC 11x (FS): Power Supply and Motor Connection

3.7.1 UEC 11x (FS)

General information

Number of available control loops

- UEC 111 (FS): 4 control loops
- UEC 112 (FS): 5 control loops

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

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

Note

Under some circumstances, certain high-speed spindles cannot be satisfactorily controlled with 3.3 kHz PWM frequency. In this case, increase the PWM frequency of the spindle. The spindle (X80) and the 1st axis (X81) share a controller group. You therefore have to set the same PWM frequency for X80 and X81. (Note the D rating!)

X31: UEC power supply

Danger

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

Note

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

Note

If the power supply is other than 400 V, an autotransformer is required. It must comply at least with the connection specifications of the UEC 11x.

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

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

Connecting terminals	**UEC 111, UEC 112**
Operation on 400 V~	
L1	400 V~ ± 10% 50 Hz to 60 Hz
L2	
L3	
	Cable / single conductor (HT wire): 6 mm^2 (AWG 10) Single conductor H07 V2-K: 4 mm^2 (AWG 10) Line fuse: 25 A (gR) Siemens Sitor type Grounding terminal: ≥ 10 mm^2 (AWG 6)
	Tightening torque for connecting terminals: 0.7 Nm (6.5 to 7 lb-in)
Operation on 480 V~	
L1	480 V~ ± 10% 50 Hz to 60 Hz
L2	
L3	
	Cable / single conductor (HT wire): 6 mm^2 (AWG 10) Single conductor H07 V2-K: 4 mm^2 (AWG 10) Line fuse: 25 A (gR) Siemens Sitor type Grounding terminal: ≥ 10 mm^2 (AWG 6)
	Tightening torque for connecting terminals: 0.7 Nm (6.5 to 7 lb-in)

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

Connection:

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

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

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

For information on the wiring and function, see the Basic Circuit Diagram for your control. Registered customers can download the Basic Circuit Diagram from the HEIDENHAIN FileBase on the Internet.

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

a. Max. 125 V

Attention

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

X89: Braking resistor

Connection at the UEC 11x:

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

X90: 24 V output

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

X344: 24 V supply for motor holding brake

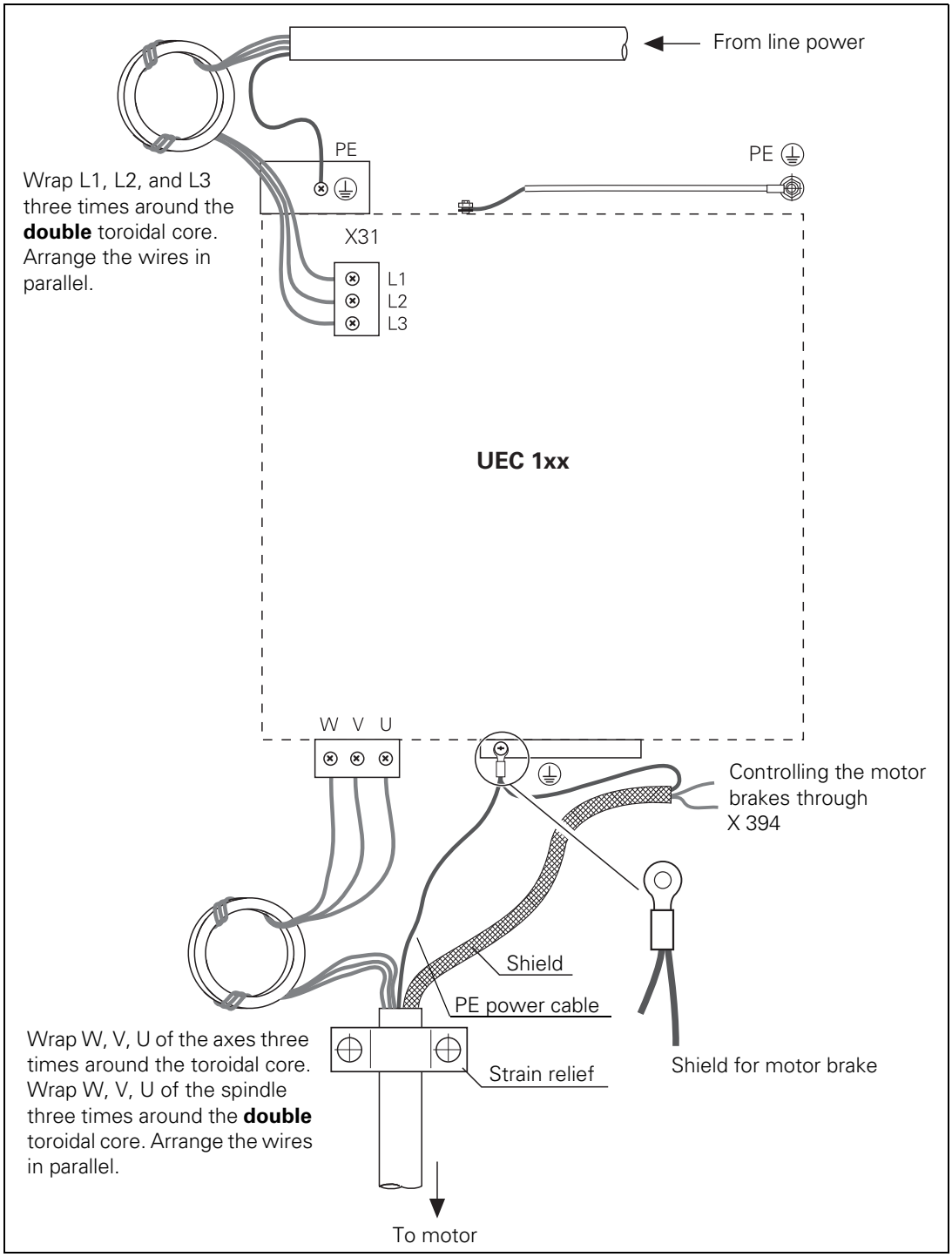
Connecting terminals	Pin layout
1	+24 V PLC
2	0 V PLC

X394: Motor holding brake

Connecting terminals	Pin layout
1	Holding brake 1
2	0 V PLC
3	Holding brake 2
4	0 V PLC
5	Holding brake 3
6	0 V PLC
7	Holding brake 4
8	0 V PLC

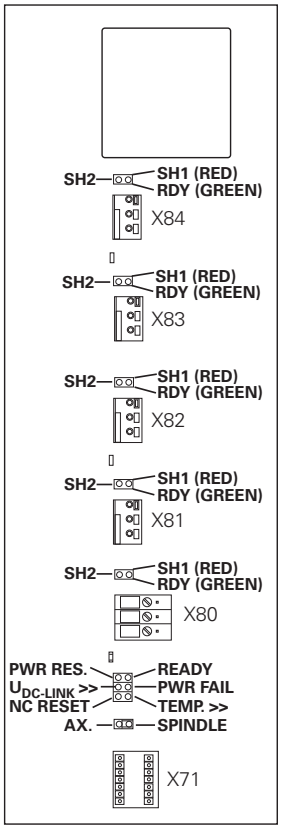
Mounting the toroidal cores

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



3.8 UxC 11x (FS): Meaning of the LEDs

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

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

3.9 Power supply for PLC outputs

The PLC outputs of the PLB 62xx are powered by the 24 V control voltage of the machine (in accordance with VDE 0551).

The control voltage must be smoothed with a minimum 1000 µF at a rated current capacity of 150 µF/A. At a current load of 15 A, for example, this corresponds to a capacity of 2250 µF. If the PSL 130 is used as 24 V– supply unit, this additional smoothing is not necessary.

Note

HEIDENHAIN recommends the PSL 130 (575 047-01) as 24 V– power supply unit (See "PSL 130 low-voltage power supply unit" on page 140).

EN 61 131-2:1994 permits:

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

Attention

Use only original replacement fuses.

Power consumption

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

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

Power output

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

Rated operating current per output

UxC 11x: 0.150 A
PLD-H xx-xx-xx: 2 A
Simultaneity with a supply voltage of 25.4 V:
2 outputs with 4 A each
4 outputs with 2 A each
8 outputs with 1 A each
Total current:
Out0 to Out7: ≤8 A
Out0 to Out3: ≤4 A
Out4 to Out7: ≤4 A

3.10 Power Supply for PLB 6xxx (FS)

Note

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

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

Pin layout of X3:
Supply voltage for logic and PLC outputs

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

3.11 Power supply for control-is-ready signal

X9: Power supply for control-is-ready signal

The control-is-ready signal output is powered by 24 V– provided by the UE 2xxB inverter or the UV 1xx power supply unit. The voltage is connected with terminal X9 of the PLB 620x.

Pin layout:

Connecting terminal X8	Pin layout	Connection when using a HEIDENHAIN inverter
1a	+24 V	X72/1
2b	0 V	X72/2

3.12 Drive Controller Enable

A drive controller can be enabled by the NC software only if the controller is enabled with 24 V at terminal X9, pin 7b of the PLB 620x.

X9/7b: Global drive controller enable

Pin layout:

Connection X9 on the PLB	Pin layout
..	..
7b	+24 V– (drive controller enable)
..	..

Drive controller enabling for axis groups

A CC 6106 or a UEC 11x does not have an X150 connector for axis-specific or axis-group-specific drive controller enabling. But in the **MP_driveOffGroupInput** machine parameter, you can enter up to six numbers of those PLC inputs that simulate the inputs of X150.

With the axis-specific parameter **MP_driveOffGroup** you have to define the axes to be switched off if the 24 V are no longer available at the specified PLC input.

For more information, see „HSCI: switching drives on and off, enabling the drive controller" on page 596

MP_driveOffGroupInput

PLC inputs belonging to the switch-off groups
Available from NCK software version: 597 110-03.

Format: Array
Input: **Group1...Group8**
You can specify up to 8 PLC inputs for the switch-off groups. Depending on the number of switch-off groups you want to realize, you have to enter the parameters here and define the PLC inputs.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_driveOffGroup

Assignment of the axis to the switch-off group
Available from NCK software version: 597 110-03.

Format: Array [0...7]
Input: You can assign the axis to a maximum of 8 switch-off groups. To do so, insert one parameter each under MP_driveOffGroup and select the desired switch-off group in the selection menu.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

3.13 Digital PLC Inputs/Outputs

Input signals and addresses

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

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

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

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

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

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

Output signals and addresses

The switching outputs are transistor outputs with current limitation.

Please note:

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

Output signals:

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

Note

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

Attention

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

X9: Safety-related PLC inputs/outputs

Pin layout of PLB 620x FS:

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

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

Terminal	Signal design. NEW	OLD connector/signal design. (MC 42xC)	Assignment / Function
1a	24 V.A	X44	24 V supply of the outputs MC.RDY, O.0 to O.1
2a	-STOS.A.G[a]	–SHS1A (safe stop of spindle) X41.32	24 V output: (spindle safe torque off)
3a	–STO.A.G[a]	–SH1A (safe stop) X41.34 / O33	24 V output: (safe torque off) Control-is-ready signal
4a	TEST.A	T.2 / X165.2	24 V output for emergency stop chain
5a	O.A.0		24 V outputs (high-side driver)
6a	O.A.1		
7a	–ES.A	–NE1 / X42.4 / I3 Acknowledgment: "Control is ready"	24 V input Emergency Stop input 1
8a	I.A.0		24 V inputs (PLC)
9a	I.A.1		
10a	I.A.2		
11a	I.A.3		
12a	I.A.4		
13a	I.A.5		
14a	–PF.PS.AC	–PF.PS.AC (signal on X69)	24 V outputs for powerfail
15a	–PF.PS.DC	–PF.PS.ZK (signal on X69)	
1b	24 V.B	X44	24 V supply of the outputs O.0 to O.1
2b	0 V		0 V PLC for all I/Os
3b	Do not assign		24 V outputs (high-side driver)
4b	TEST.B	T.1 / X165.1	24 V output for emergency stop chain
5b	O.B.0		24 V outputs (high-side driver)
6b	O.B.1		
7b	–ES.B	–NE2 / X42.33 / I32 "Drive enabling"	24 V input Emergency Stop input 2
8b	I.B.0		24 V inputs (PLC)
9b	I.B.1		
10b	I.B.2		
11b	I.B.3		
12b	I.B.4		
13b	I.B.5		
14b	–SP.REF+	X30	Optocoupler input, Spindle ref.
15b	–SP.REF–	X30	

a. 2-A outputs

X9: Safety-related PLC inputs/outputs

Pin layout of PLB 620x:

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

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

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

a. 2-A outputs

PLC inputs on the PLD-H

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

Note

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

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

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

X14: PLC inputs channel B										
Pin layout	**Terminal**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PL 6xxx Slot 1	0 V PLC	0 V PLC	I0.B	I1.B	I2.B	I3.B	I4.B	I5.B	I6.B	I7.B

X17: PLC inputs, channel A/B										
Pin layout	**Terminal**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PL 6xxx Slot 1	0 V PLC	0 V PLC	I0.A	I1.A	I2.A	I3.A	I0.B	I1.B	I2.B	I3.B

Please note that a system with Functional Safety (FS) permits up to 100 safe, dual-channel inputs (channels A and B). An MB 6xx FS from HEIDENHAIN requires 56 of the 100 safe inputs, and connector X9 of a system PL requires eight additional safe inputs. This means that max. 36 safe inputs can be distributed to the PLD-H xx-xx-xx FS.

Fast PLC inputs

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

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

PLC outputs on the PLD-H

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

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

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

X24: PLC outputs, channel B										
Pin layout	**Terminal**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PL 6xxx Slot 1	O0.B	O1.B	O2.B	O3.B	O4.B	O5.B	O6.B	O7.B	24 V PLC for O0 to O3	24 V PLC for O4 to O15

X27: PLC outputs, channel A / channel B										
Pin layout	**Terminal**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
PL 6xxx Slot 1	O0.A	O1.A	O2.A	O3.A	O0.B	O1.B	O2.B	O3.B	24 V PLC for O0.A to O3.A	24 V PLC for O0.B to O3.B

Please note that a system with Functional Safety (FS) permits up to 64 safe, dual-channel outputs (channels A and B). Connector X9 of a system PL requires three of the outputs. This means that max. 61 safe outputs can be distributed to the PLD-H xx-xx-xx FS.

3.13.1 UxC 11x (FS): Digital PLC inputs/outputs

The following digital inputs and outputs are available on the UEC 111 (FS), UEC 112 (FS) or UMC 111 FS.

X4: Single-channel PLC inputs

Connection on the front of the UxC 11x (FS):

Terminal	Signal designation	Assignment / Function
1a	+24 V PLC.01	24 V supply of the outputs MC.RDY, O16 to O22
2a	+24 V PLC.02	24 V supply of the outputs O8 to O15
3a	+24 V PLC.03	24 V supply of the outputs O0 to O7
4a	0 V PLC	0 V for all I/Os
5a	–REF.SP	Reserved, do not assign
6a	0 V PLC	0 V for all I/Os
7a	I12	24 V inputs
8a	I13	
9a	I14	
10a	I15	
11a	I16	
12a	I17	
1b	I0	24 V inputs
2b	I1	
3b	I2	
4b	I3	
5b	I4	
6b	I5	
7b	I6	
8b	I7	
9b	I8	
10b	I9	
11b	I10	
12b	I11	

X5: Single-channel PLC inputs

Connection on the front of the UMC 111 (FS):

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

Note

If the integrated PLC inputs do not suffice for your application, you can connect up to 7 additional external PL 61xx expansion PLs to the UxC 111FS by the HSCI interface.

X104: Single/ double channel PLC inputs

Connection on the front of the UxC 11x FS:

Terminal	Signal designation	Assignment / Function
1a	+24 V.A	24 V supply of the OA0 to OA7 outputs
2a	+24 V.B	24 V supply of the OB0 to OB7 outputs
3a	+24 V.C	24 V supply of the OA8 to OA15 outputs
4a	Do not assign	
5a	IB4	24 V inputs
6a	IB5	
7a	IB6	
8a	IB7	
9a	IA4	
10a	IA5	
11a	IA6	
12a	IA7	
11a	IA6	
12a	IA7	
1b	Do not assign	
2b	Do not assign	
3b	Do not assign	
4b	Do not assign	
5b	IB0	24 V inputs
6b	IB1	
7b	IB2	
8b	IB3	
9b	IA0	
10b	IA1	
11b	IA2	
12b	IA3	

Note

If the integrated PLC inputs do not suffice for your application, you can connect up to 7 additional external PL 61xx expansion PLs to the UxC 111FS by the HSCI interface.

X6: Single-channel PLC outputs

Connection on the top of the UxC 11x (FS):

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

Note

If the integrated PLC outputs do not suffice for your application, you can connect up to 7 additional external PL 61xx expansion PLs to the UxC 111FS by the HSCI interface.

X106: Single/ double channel PLC outputs

Connection on the front of the UxC 11x FS:

Terminal	Signal designation	Assignment / Function
1a	OB4	24 V outputs, can be switched off via terminal X104.2a (+24 V.B)
2a	OB5	
3a	OB6	
4a	OB7	
5a	OA4	24 V outputs, can be switched off via terminal X104.1a (+24 V.A)
6a	OA5	
7a	OA6	
8a	OA7	
9a	OA12	24 V outputs, can be switched off via terminal X104.3a (+24 V.C)
10a	OA13	
11a	OA14	
12a	OA15	
1b	OB0	24 V outputs, can be switched off via terminal X104.2a (+24 V.B)
2b	OB1	
3b	OB2	
4b	OB3	
5b	OA0	24 V outputs, can be switched off via terminal X104.1a (+24 V.A)
6b	OA1	
7b	OA2	
8b	OA3	
9b	OA8	24 V outputs, can be switched off via terminal X104.3a (+24 V.C)
10b	OA9	
11b	OA10	
12b	OA11	

Note

If the integrated PLC outputs do not suffice for your application, you can connect up to 7 additional external PL 61xx expansion PLs to the UxC 111FS by the HSCI interface.

3.14 Analog PLC inputs/outputs

Note

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

Specifications:

Analog inputs

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

Analog outputs

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

Inputs for Pt 100 thermistors

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

X31 to X34: Analog inputs

Pin layout

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

X46 to X49: Analog inputs

Pin layout

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

X51 to X54: Analog output

Pin layout

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

X66 to X67: Analog output

Pin layout

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

X71 to X74: Connection for Pt 100

Pin layout:

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

X81 to X82: Connection for Pt 100

Pin layout:

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

3.15 PROFIBUS Connection

Pin layout on X121 of the MC or IPC main computer and on X1 of the PLB 550 PROFIBUS slave

Main computer X121		**Connecting cable ID 515 845-01**			**PLB 550 X1**	
D-sub connctn. (female) 9-pin	**Pin layout**	**D-sub cnnctr. (male) 9-pin**		**D-sub cnnctr. (male) 9-pin**	**X1 D-sub cnnctn. (female) 9-pin**	**Pin layout**
1	Do not assign	1	–	1	1	Do not assign
2	Do not assign	2	–	2	2	Do not assign
3	B line	3	B line	3	3	B line
4	RTS (Signal type: TTL)	4	–	4	4	RTS (Signal type: TTL)
5	GND	5	–	5	5	GND
6	+5 V	6	–	6	6	+5 V
7	Do not assign	7	–	7	7	Do not assign
8	A line	8	A line	8	8	A line
9	Do not assign	9	–	9	9	Do not assign
Housing	Ext. shield	Housing	Ext. shield	Housing	Housing	Ext. shield

All signals on the PL 550 are electrically isolated.

The +5 V and GND pins supply the terminating resistor in the connector.

Note

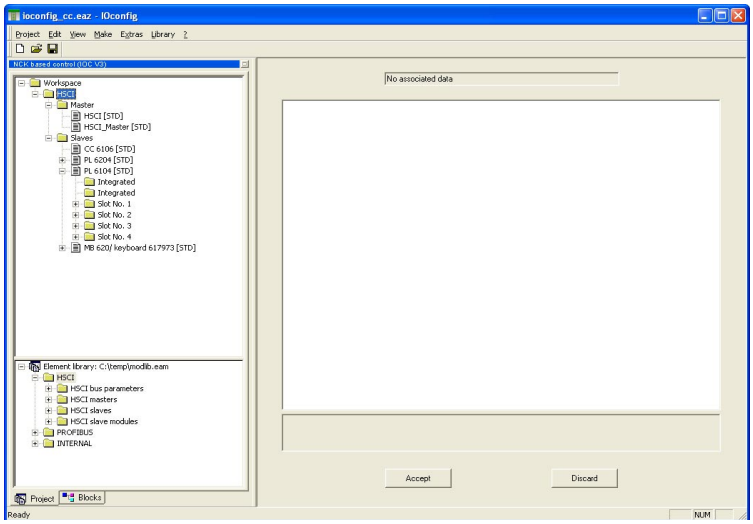
For more information about PROFIBUS on HEIDENHAIN controls, refer to the documentation about the IOconfig software for PCs.

3.16 Configuring the PLC Inputs/Outputs with IOconfig

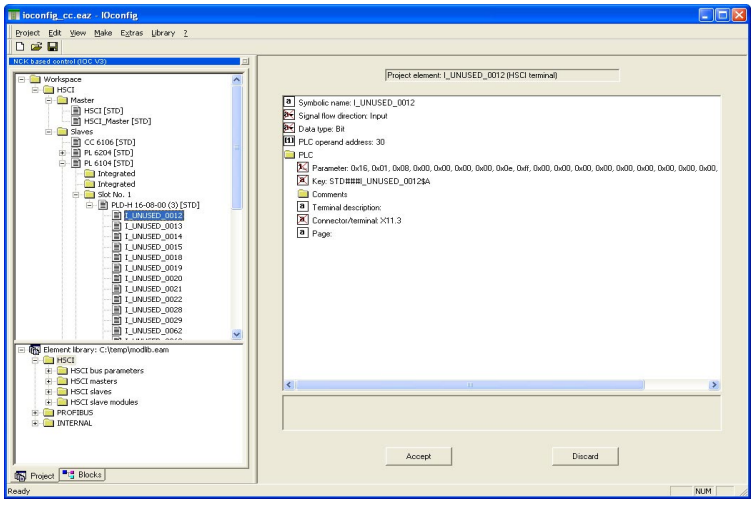
The PC program IOconfig is used to configure the PLB 6xxx input/output modules connected via the HSCI network. The program has previously been used to describe a PROFIBUS system, and now it also includes the HSCI system. If desired, IOconfig creates the configuration file (IOC file) and symbol definition file (DEF file) for the control or the PLC program.

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

- Position of all input/output modules in the HSCI system (PL 6xxx with associated PLD-H and PLA-H modules, MB 6xx and UEC 11x)



- Definition of all I/Os freely available to the OEM on PLD-Hs, PLA-Hs and MB6xx and UEC 11x.
- Assignment of symbolic names, PLC operand addresses and other properties of the individual terminals (I/Os)



For more information about using IOconfig and configuring the HSCI I/Os, please refer to the program's online help.

3.17 Buffer battery

Note

Make a data backup before changing the buffer battery.

Danger

When exchanging the buffer battery, remember:

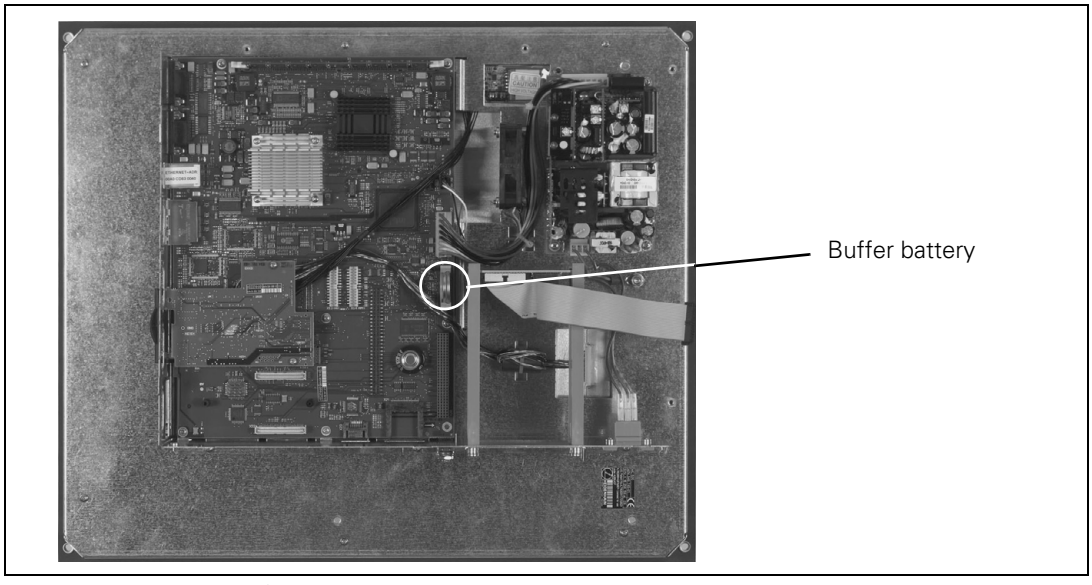
- Switch off the machine and the control.
- The buffer battery may be exchanged only by trained personnel.

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

If the voltage of the buffer battery falls below 2.6 V, the error message **Exchange buffer battery** appears. If the voltage does not rise above 2.6 V again, the error message is reactivated after 30 minutes. You can determine the voltage of the buffer battery with the OLM, see "Diagnosis with the Online Monitor (OLM)" auf Seite 819.

To exchange the battery:

- The buffer battery is on the main board of the MC.
- Loosen the screws securing the rear housing cover of the MC.
- Remove the cover.
- The buffer battery is at the border of the PCB:



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

3.18 Encoder connections

3.18.1 General information

HEIDENHAIN contouring controls are designed for use with incremental or absolute linear and angular encoders as measuring systems. The encoder signals are subdivided 1024-fold.
Encoders with one reference mark or distance-coded reference marks and with EnDat interface are permissible.
HEIDENHAIN recommends the use of absolute encoders with EnDat interface or the use of encoders with distance-coded reference marks because they greatly reduce the traverse distance required to establish the absolute position.

Note

On controller units with HSCI interface (e.g. CC 61xx, UEC 11x or UMC 11x) you can no longer directly connected encoders with 11 µA signals. You need an adapter ID 313 119-xx to adjust the signals; See "Adapters for Encoder Signals" on page 183.

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

	Position encoder	**Speed encoder**
Signal amplitude	EnDat, 1 V_{PP} (to be defined in **MP_posEncoderSignal**)	EnDat, 1 V_{PP}
Input frequency 1 V_{PP}	CC 61xx: 27 kHz/400 kHz (to be defined in **MP_posEncoderFreq**)	CC 61xx: 400 kHz

Note

Keep in mind the line count of the speed encoders when choosing the motors:

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

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

Example:
f = 350 kHz; n = 10 000 min^{-1}

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

3.18.2 Position encoder input

X201 to X206:
Position encoder
1 V_{PP}

Pin layout:

CC 6106		Adapter cable ID 309 783-xx Adapter cable 310 199-xx			Encoder	
Male	**Pin layout**	**Female**	**Color**	**Female**	**Male**	**Color**
1	+5 V (U_P)	1	Brown/Green	12	12	Brown/Green
2	0 V (U_N)	2	White/Green	10	10	White/Green
3	A+	3	Brown	5	5	Brown
4	A–	4	Green	6	6	Green
5	Do not assign	5				
6	B+	6	Gray	8	8	Gray
7	B–	7	Pink	1	1	Pink
8	Do not assign	8				
9	+5 V (sensor)	9	Blue	2	2	Blue
10	R+	10	Red	3	3	Red
11	0 V (sensor)	11	White	11	11	White
12	R–	12	Black	4	4	Black
13	0 V	13				
14	Do not assign	14	Violet	7	7	Violet
15	Do not assign	15				
Hsg.	Ext. shield	Hsg.	Ext. shield	Hsg.	Hsg.	Ext. shield

Note

The interface complies with the requirements of EN 50178 for "low voltage electrical separation."

X201 to X206: Position encoder with EnDat interface

Pin layout:

MC 6106		Adapter cable 332 115-xx			Connecting cable 323 897-xx				Adapter cable 313 791-xx		
Male	**Pin layout**	**Female**	**Color**	**Female**	**Male**	**Color**	**Fem.**		**Male**	**Color**	**Fem.**
1	+5 V (U_P)	1	Brown/ Green	7	7	Brown/ Green	7	Line drop compensator ID 336 697-02, if required	7	Brown/ Green	5b
2	0 V (U_N)	2	White/ Green	10	10	White/ Green	10		10	White/ Green	6a
3	A+	3	Green/ Black	15	15	Green/ Black	15		15	Green/ Black	2a
4	A–	4	Yellow/ Black	16	16	Yellow/ Black	16		16	Yellow/ Black	2b
5	Data	5	Gray	14	14	Gray	14		14	Gray	3b
6	B+	6	Blue/ Black	12	12	Blue/ Black	12		12	Blue/ Black	1a
7	B–	7	Red/ Black	13	13	Red/ Black	13		13	Red/ Black	1b
8	$\overline{\text{Data}}$	8	Pink	17	17	Pink	17		17	Pink	3a
9	+5 V (sensor)	9	Blue	1	1	Blue	1		1	Blue	5a
10	Vacant	10		3	3	Red	3		3		
11	0 V (sensor)	11	White	4	4	White	4		4	White	6b
12	Vacant	12		2	2	Black	2		2		
13	Internal shield	13	Internal shield	11	11	Internal shield	11		11	Internal shield	
14	Clock	14	Violet	8	8	Violet	8		8	Violet	4a
15	$\overline{\text{Clock}}$	15	Yellow	9	9	Yellow	9		9	Yellow	4b
Hsg.	Housing	Hsg.	Ext. shield	Hsg.		Ext. shield			Hsg.	Ext. shield	

Note

The interface complies with the requirements of EN 50178 for "low voltage electrical separation."

Note

For cable lengths > 10 m between the CC 61xx or UxC 11x and the encoders with EnDat interfaces (EnDat 2.1), a line-drop compensator (336 697-xx) is required (efficiency = 75 %).

3.18.3 Input of speed encoder

Attention

If you connect angle or linear encoders from HEIDENHAIN to the speed encoders (such as for torque motors), you must pay attention to the different connector layouts!
HEIDENHAIN offers special cables and line-drop compensators for such applications. More information is in the Cable Overviews.

X15 to X20: Shaft speed encoder 1 V_{PP}

Pin layout:

CC 61xx		Adapter cable 289 440-xx				Connecting cable 336 847-xx		
Male	**Pin layout**	**Female**	**Color**	**Female**		**Male**	**Color**	**Female**
1	+5 V (U_P)	1	Brown/Green	10	Line drop compensator ID 370 226-01, if required	10	Brown/Green	10
2	0 V (U_N)	2	White/Green	7		7	White/Green	7
3	A+	3	Green/Black	1		1	Green/Black	1
4	A–	4	Yellow/Black	2		2	Yellow/Black	2
5	0 V							
6	B+	6	Blue/Black	11		11	Blue/Black	11
7	B–	7	Red/Black	12		12	Red/Black	12
8	0 V	8	Internal shield	17		17	Internal shield	17
9	Do not assign							
10	Do not assign							
11	Do not assign							
12	Do not assign							
13	Temperature +	13	Yellow	8		8	Yellow	8
14	+5 V (sensor)	14	Blue	16		16	Blue	16
15	Do not assign							
16	0 V (sensor)	16	White	15		15	White	15
17	R+	17	Red	3		3	Red	3
18	R–	18	Black	13		13	Black	13
19	C+	19	Green	5		5	Green	5
20	C–	20	Brown	6		6	Brown	6
21	D+	21	Gray	14		14	Gray	14
22	D–	22	Pink	4		4	Pink	4
23	Do not assign							
24	0 V							
25	Temperature –	25	Violet	9		9	Violet	9
Hsg.	Housing	Hsg.	Ext. shield	Hsg.		Hsg.	Ext. shield	Hsg.

Note

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

X15 to X20: Position encoder with EnDat interface

Pin layout:

CC 61xx		Adapter cable 336 376-xx				Connecting cable 340 302-xx		
Male	**Pin layout**	**Female**	**Color**	**Female**		**Male**	**Color**	**Female**
1	+5 V (U_P)	1	Brown/Green	10	Line drop compensator ID 370 224-01, if required	10	Brown/Green	10
2	0 V (U_N)	2	White/Green	7		7	White/Green	7
3	A+	3	Green/Black	1		1	Green/Black	1
4	A–	4	Yellow/Black	2		2	Yellow/Black	2
5	0 V							
6	B+	6	Blue/Black	11		11	Blue/Black	11
7	B–	7	Red/Black	12		12	Red/Black	12
8	0 V	8	Internal shield	17		17	Internal shield	17
9	Do not assign							
10	Clock	10	Green	5		5	Green	5
11	Do not assign							
12	$\overline{\text{Clock}}$	12	Brown	14		14	Brown	14
13	Temperature +	13	Yellow	8		8	Yellow	8
14	+5 V (sensor)	14	Blue	16		16	Blue	16
15	Data	15	Red	3		3	Red	3
16	0 V (sensor)	16	White	15		15	White	15
17	Do not assign							
18	Do not assign							
19	Do not assign							
20	Do not assign							
21	Do not assign							
22	Do not assign							
23	$\overline{\text{Data}}$	23	Black	13		13	Black	13
24	0 V							
25	Temperature –	25	Violet	9		9	Violet	9
Hsg.	Housing	Hsg.	Ext. shield	Hsg.		Hsg.	Ext. shield	Hsg.

Note

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

Danger

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

Pin layout (for the LC or RCN):

CC 61xx		Adapter cable 336 376-xx				Adapter cable 369 124-xx Adapter cable ID 369 129-xx	
Male	**Pin layout**	**Female**	**Color**	**Female**		**Male**	**Color**
1	+5 V (U_P)	1	Brown/Green	10	Line drop compensator ID 368 210-02, if required	7	Brown/Green
2	0 V (U_N)	2	White/Green	7		10	White/Green
3	A+	3	Green/Black	1		15	Green/Black
4	A–	4	Yellow/Black	2		16	Yellow/Black
5	0 V						
6	B+	6	Blue/Black	11		12	Blue/Black
7	B–	7	Red/Black	12		13	Red/Black
8	0 V	8	Internal shield	17		11	Internal shield
9	Do not assign						
10	Clock	10	Green	5		8	Violet
11	Do not assign						
12	$\overline{\text{Clock}}$	12	Brown	14		9	Yellow
13	Temperature +	13	Yellow	8			
14	+5 V (sensor)	14	Blue	16		1	Blue
15	Data	15	Red	3		14	Gray
16	0 V (sensor)	16	White	15		4	White
17	Do not assign						
18	Do not assign						
19	Do not assign						
20	Do not assign						
21	Do not assign						
22	Do not assign						
23	$\overline{\text{Data}}$	23	Black	13		17	Pink
24	0 V						
25	Temperature –	25	Violet	9			
Hsg.	Housing	Hsg.	Ext. shield	Hsg.		Hsg.	Ext. shield

1
2 temperature+
3 temperature–
4

Note

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

Pin layout (for the LC or RCN):

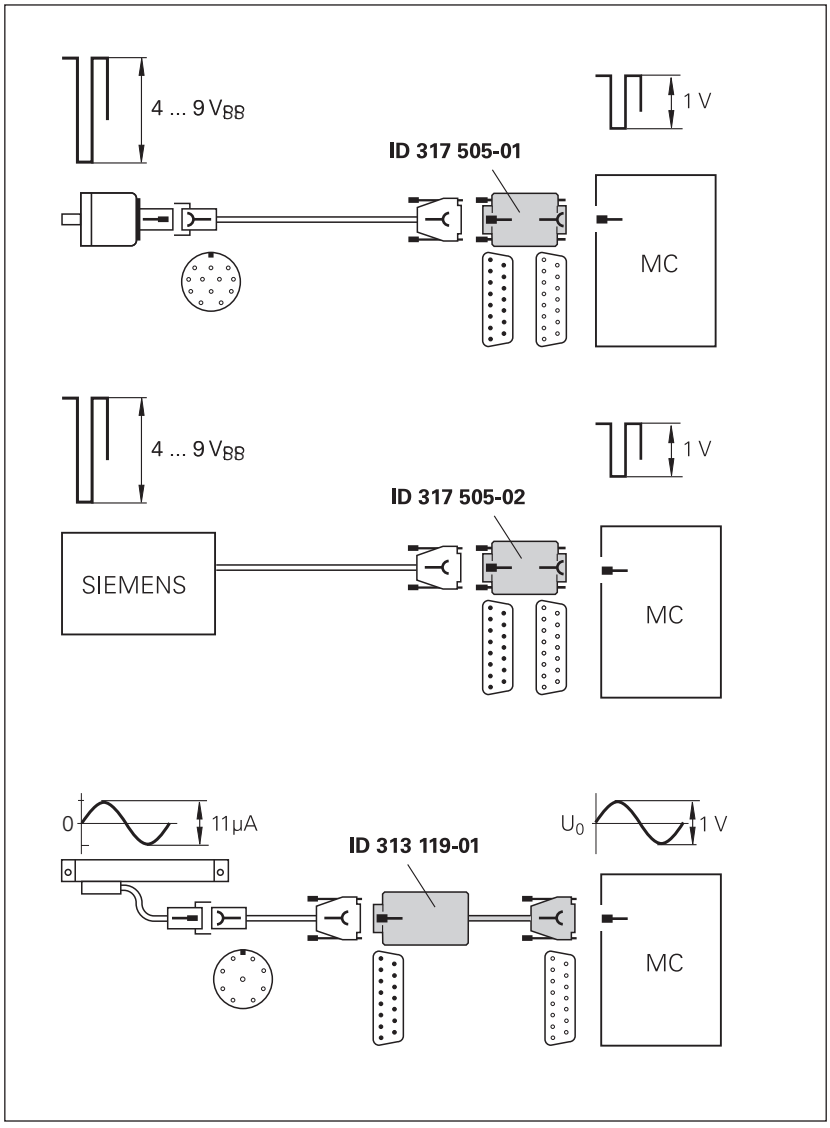
CC 61xx		Adapter cable 509 667-xx			Adapter cable 369 124-xx Adapter cable ID 369 129-xx or RCN	
Male	**Pin layout**	**Female**	**Color**	**Female**	**Male**	**Color**
1	+5 V (U_P)	1	Brown/ Green	7	7	Brown/ Green
2	0 V (U_N)	2	White/ Green	10	10	White/ Green
3	A+	3	Green/ Black	15	15	Green/Black
4	A–	4	Yellow/ Black	16	16	Yellow/ Black
5	0 V					
6	B+	6	Blue/ Black	12	12	Blue/Black
7	B–	7	Red/Black	13	13	Red/Black
8	0 V	8	Internal shield	11	11	Internal shield
9	Do not assign					
10	Clock	10	Green	8	8	Violet
11	Do not assign					
12	$\overline{\text{Clock}}$	12	Brown	9	9	Yellow
13	Temperature +	13	Yellow	5		
14	+5 V (sensor)	14	Blue	1	1	Blue
15	Data	15	Red	14	14	Gray
16	0 V (sensor)	16	White	4	4	White
17	Do not assign					
18	Do not assign					
19	Do not assign					
20	Do not assign					
21	Do not assign					
22	Do not assign					
23	$\overline{\text{Data}}$	23	Black	17	17	Pink
24	0 V					
25	Temperature –	25	Violet	6		
Hsg.	Housing	Hsg.	Ext. shield	Hsg.	Hsg.	Ext. shield

Note

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

3.19 Adapters for Encoder Signals

Encoder signals with 11 μA_{PP} or TTL levels can be adapted to the 1 V_{PP} interface with HEIDENHAIN adapter connectors.



Note

Please note:

- The adapters adjust only the levels, not the signal shape.
- The contamination signal of the square-wave encoder cannot be evaluated.
- A square-wave signal can be subdivided no more than 4-fold.

Adapter connector TTL (HEIDENHAIN)/ 1 V_{PP}

Pin layout of D-sub connector (female) and D-sub connector (male):

D-sub connector (female) 15-pin	Pin layout	D-sub connection (male) 15-pin	Pin layout
1	+5 V (U_P)	1	+5 V (U_P)
2	0 V (U_N)	2	0 V (U_N)
3	A+	3	U_{a1}
4	A–	4	$-U_{a1}$
5	Not assigned	5	Not assigned
6	B+	6	U_{a2}
7	B–	7	$-U_{a2}$
8	Not assigned	8	Not assigned
9	+5 V	9	+5 V
10	R+	10	U_{a0}
11	0 V	11	0 V
12	R–	12	$-U_{a0}$
13	Not assigned	13	Not assigned
14	**Do not assign**[a]	14	**Do not assign**[a]
15	Not assigned	15	Not assigned

a. The control assigns the EnDat clock to pin 14. Therefore, you must not assign any signals to this pin when using the TTL adapter connector.

Adapter connector TTL (SIEMENS) / 1 V_{PP}

Pin layout of D-sub connector (female) and D-sub connector (male):

D-sub connector (female) 15-pin	Pin layout	D-sub connection (male) 15-pin	Pin layout
1	Not assigned	1	Not assigned
2	0 V	2	0 V
3	A+	3	U_{a1}
4	A–	4	$-U_{a1}$
5	Not assigned	5	Not assigned
6	B+	6	U_{a2}
7	B–	7	$-U_{a2}$
8	Not assigned	8	Not assigned
9	Not assigned	9	Not assigned
10	R+	10	Not assigned
11	Not assigned	11	Not assigned
12	R–	12	U_{a0}
13	Not assigned	13	$-U_{a0}$
14	Not assigned	14	Not assigned
15	Not assigned	15	Not assigned

Adapter connector 11 μA_{PP} / 1 V_{PP}

Pin layout of D-sub connector (female) and D-sub connector (male):

D-sub connector (female) 15-pin	Pin layout	D-sub connection (male) 15-pin	Pin layout
1	+5 V (U_P)	1	+5 V (U_P)
2	0 V (U_N)	2	0 V (U_N)
3	A+	3	0°+
4	A–	4	0°–
5	0 V	5	0 V
6	B+	6	90°+
7	B–	7	90°–
8	0 V	8	0 V
9	+5 V	9	+5 V
10	R+	10	R+
11	0 V	11	0 V
12	R–	12	R–
13	0 V	13	0 V
14	Not assigned	14	Not assigned
15	Not assigned	15	Not assigned

3.20 Connecting the Motor Power Stages (Only CC 61xx)

The MANUALplus 620 is connected with HEIDENHAIN or non-HEIDENHAIN inverters through a PWM interface.

For a description of the HEIDENHAIN inverter systems, refer to the Technical Manual "Inverter Systems and Motors." The components required for operation of the MANUALplus 620 with non-HEIDENHAIN inverter systems are described in the manual "Technical Information for the Operation of SIMODRIVE and POWER DRIVE Inverter Systems."

The individual PWM outputs of the CC 61xx are assigned to different controller groups.

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

Logic level:	5 V
Analog signals I_{ACTL}:	±7.5 V
PWM frequency:	Can be set it at 3333 Hz, 4166 Hz, 5000 Hz, 6666 Hz, 8333 Hz and 10000 Hz

X51 to X56: PWM output

Pin layout:

Ribbon connector, 20-pin	Pin layout
1a	PWM U1
1b	0 V U1
2a	PWM U2
2b	0 V U2
3a	PWM U3
3b	0 V U3
4a	$\overline{SH2}$
4b	0 V ($\overline{SH2}$)
5a	$\overline{SH1}$
5b	0 V ($\overline{SH1}$)
6a	+IIST 1
6b	–IIST 1
7a	0 V (analog)
7b	+IIST 2
8a	–IIST 2
8b	0 V (analog)
9a	$\overline{BRK}$
9b	Do not assign
10a	$\overline{ERR}$
10b	RDY

Note

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

3.21 Touch Probe Systems

The following touch probes can be connected to the MANUALplus 620:

- TS 220, touch-trigger probe with cable connection for workpiece setup and measurement during machining
- TS 440, TS 444, TS 640, TS 740, touch-trigger probes with infrared transmission for workpiece setup and measurement during machining
- TT 130, TT 140, touch probes for workpiece measurement

Note

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

Note

For the PLB 62xx up to variant -02, please note:

To connect a TT touch probe to the PLB 62xx, you have to connect the touch probe adapter, ID 667 674-01, to X113 See "Adapter for connection of a TT touch probe to X113" on page 190. The adapter is supplied with the PL.

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

X112/X113: Triggering touch probe

Note

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

Note

For the PLB 62xx up to variant -02, please note:

To connect a TT touch probe to the MANUALplus 620, you have to connect the touch probe adapter, ID 667 674-01, to X113.

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

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

Note

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

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

a. Stylus at rest means logic level HIGH.

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

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

Adapter for connection of a TT touch probe to X113

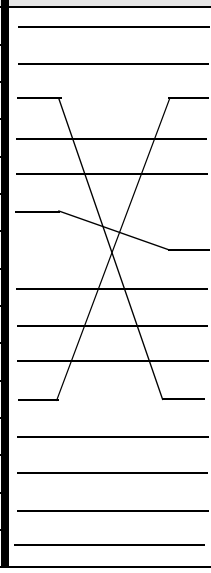
Only variant -02 of PLB 62xx:

The items supplied with the PLB 62xx include a cable adapter with the ID 667 674-01.

If you want to connect a TT touch probe to X113 of the PLB 62xx, you need the cable adapter. It makes the Start and Ready signals of the TT touch probes available on the correct pins of X113.

Pin layout of adapter ID 667 674-01:

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

Female	Assignment of X113 (TT) up to variant -02	Adapters ID667 674-01	TT adapter cables ID 633 616-xx
1	Trigger signal		Trigger signal
2	$\overline{\text{Trigger signal}}$a		$\overline{\text{Trigger signal}}$a
3	TS ready		Do not assign
4	$\overline{\text{Battery warning}}$		$\overline{\text{Battery warning}}$
5	+ 5 V NC (+/– 5%)		+ 5 V NC (+/– 5%)
6	Start		Do not assign
7	0 V NC		Start TT
8	0 V NC		0 V NC
9	0 V NC		0 V NC
10	+ 24 V NC		+ 24 V NC
11	TT ready		TT ready
12	Do not assign		Do not assign
13	Do not assign		Do not assign
14	Do not assign		Do not assign
15	Do not assign		Do not assign

X26, X166: Ethernet interface RJ45-port

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

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

Danger

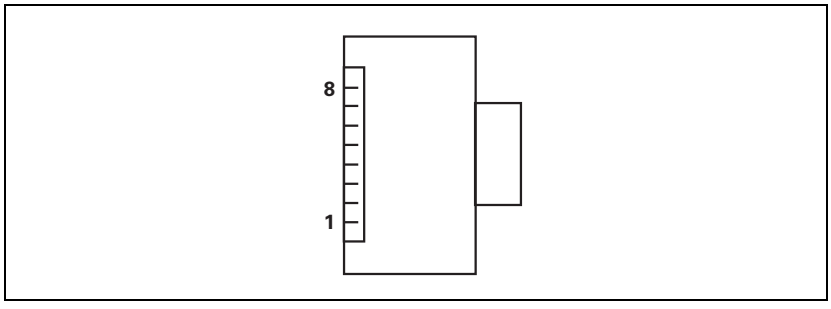
The Ethernet interfaces of the MC 6xxx comply with the requirements of PELV ("low voltage electrical separation") according to EN 61800-5-1 and are powered internally by 24 V NC. All devices connected to these Ethernet interfaces must comply with the requirements of SELV or PELV according to EN 61800-5-1.

Maximum data transfer rate:

- For integration into the company network via NFS or SMB protocol: 10 or 100 Mbps
- For **LSV2 protocol** (in conjunction with TNCremoNT or RemoTools): 2 to 5 Mbps (depending on file type and network utilization)

RJ45 connection (female) 8-pin	Pin layout
1	TX+
2	TX-
3	REC+
4	Do not assign
5	Do not assign
6	REC-
7	Do not assign
8	Do not assign
Housing	Ext. shield

Face of the connector:



Meanings of the LEDs on the Ethernet data interface X26:

LED	Status	Meaning
Green	Blinking	Interface active
	Off	Interface inactive
Yellow	On	100 Mb network
	Off	10 Mb network

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

- Maximum cable length with RS-232-C/V.24 is 20 meters

Pin layout:

Note

The interface complies with the requirements of EN 50178 for "low voltage electrical separation."

25-pin adapter block:

MC 6xxx		Connecting cable 365 725-xx			Adapter block 310 085-01		Connecting cable 274 545-xx		
Male	**Pin layout**	**Female**	**Color**	**Female**	**Male**	**Female**	**Male**	**Color**	**Female**
1	Do not assign	1		1	1	1	1	White/ Brown	1
2	RXD	2	Yellow	3	3	3	3	Yellow	2
3	TXD	3	Green	2	2	2	2	Green	3
4	DTR	4	Brown	20	20	20	20	Brown	8
5	Signal GND	5	Red	7	7	7	7	Red	7
6	DSR	6	Blue	6	6	6	6		6
7	RTS	7	Gray	4	4	4	4	Gray	5
8	CTS	8	Pink	5	5	5	5	Pink	4
9	Do not assign	9					8	Violet	20
Hsg.	Ext. shield	Hsg.	Ext. shield	Hsg.	Hsg.	Hsg.	Hsg.	Ext. shield	Hsg.

9-pin adapter block:

MC 6xxx		Connecting cable 355 484-xx			Adapter block 363 987-02		Connecting cable 366 964-xx		
Male	**Pin layout**	**Female**	**Color**	**Male**	**Female**	**Male**	**Female**	**Color**	**Female**
1	Do not assign	1	Red	1	1	1	1	Red	1
2	RXD	2	Yellow	2	2	2	2	Yellow	3
3	TXD	3	White	3	3	3	3	White	2
4	DTR	4	Brown	4	4	4	4	Brown	6
5	Signal GND	5	Black	5	5	5	5	Black	5
6	DSR	6	Violet	6	6	6	6	Violet	4
7	RTS	7	Gray	7	7	7	7	Gray	8
8	CTS	8	White/ Green	8	8	8	8	White/ Green	7
9	Do not assign	9	Green	9	9	9	9	Green	9
Hsg.	Ext. shield	Hsg.	Ext. shield	Hsg.	Hsg.	Hsg.	Hsg.	Ext. shield	Hsg.

3.22.1 USB interface (USB 2.0)

X141, X142, X143, X144

Pin layout for USB connection (Type A):

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

Note

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

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

USB hub

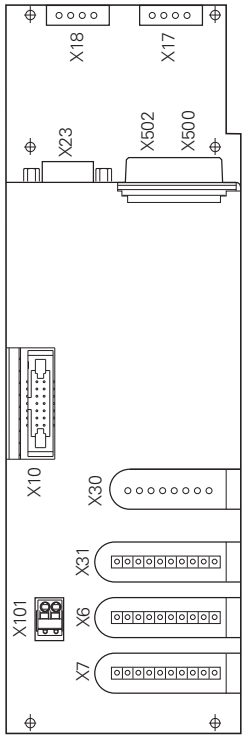
The power supply for the USB hub must comply with EN 50 178, 5.88 requirements for "low voltage electrical separation."

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

3.23 MB 620T Machine Operating Panel

For lathes, HEIDENHAIN offers the MB 620T machine operating panel with HSCI interface. It is based on the MB 620 and contains in addition two override potentiometers for adjusting the spindle speed and the feed rate.

On the underside of the machine operating panel are terminal strips bearing the PLC inputs as well as the PLC outputs. Also, connection X23 (203) for HR serial handwheels is on the underside of the MB 620T.

Pin layout for MB 620T (FS)	Connector	Function	Page
	X17	Emergency stop (MB)	196
	X18	Emergency stop (MB)	196
	X500	HSCI output	124
	X502	HSCI input	124
	X6	PLC inputs	197
	X7	PLC outputs	196
	X10	Interface to keyboard and potentiometers	199
	X23	Handwheel connection	203
	X30	Connection for handwheel adapter	200
	X31	Permissive button, NC start, NC stop	198
	X101	24 V NC power supply	199
	⏚	Protective ground	–

X17/X18: EMERGENCY STOP on MB

Connectors X17 and X18 are electrically parallel.

With the MB 620T without FS, the EMERGENCY STOP must be wired externally in the EMERGENCY STOP chain as before. In the MB 620T without FS, the connectors X17 and X18 do not support dual channel evaluation of the EMERGENCY STOP button. These inputs must therefore not be used for evaluating EMERGENCY STOP!

With the MB 620T FS, the EMERGENCY STOP is evaluated by the control via HSCI and has HEIDENHAIN standard wiring.

Pin layout X18 on MB 620 FS (X17 without function):

Connecting terminals	Pin layout
1	Power supply for emergency stop channel A (-TEST.A)
2	Emergency stop channel A
3	Power supply for emergency stop channel B (-TEST.B)
4	Emergency stop channel B

X7: PLC outputs

Pin layout:

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

a. With standard wiring

Ampacity of the outputs: Maximum 150 mA per output

Danger

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

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

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

X6: PLC inputs

Pin layout of MB 620T without FS:

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

a. With standard wiring

Pin layout of MB 620T FS:

Connecting terminals	Pin layout
1	I0 channel A
2	I1 channel A
3	I2 channel A (control voltage ON, CVO.A)[a]
4	I3 channel A
5	I0 channel B
6	I1 channel B
7	I2 channel B (control voltage ON, CVO.B)a
8	I3 channel B
9	Power supply for channel A (-TEST.A)
10	Power supply for channel B (-TEST.B)

a. With standard wiring

Danger

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

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

X31: MB 620T, permissive button / NC start / NC stop

Standard wiring of connection X31 for MB 620T without FS:

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

a. With standard wiring

Pin layout of MB 620T FS:

Connecting terminals	Pin layout
1	Optional: Permissive button channel A / terminal 1 (+24 V)
2	Optional: Permissive button channel A / terminal 2
3	Optional: Permissive button channel B / terminal 1 (+24 V)
4	Optional: Permissive button channel B / terminal 2
5	NC Start channel A / terminal 1 (a)
6	NC Start channel B / terminal 2 a
7	NC start shared terminal (+24 V NC) (a)
8	NC Start channel A / terminal 1 a
9	NC Start channel B / terminal 2 (a)
10	NC stop shared terminal (+24 V NC)[a]

a. With standard wiring

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

Danger

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

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

X10: Interface to keyboard and potentiometers

Pin layout:

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

X101: Power supply

Pin layout:

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

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

Power consumption of the MB 620T:4.0 W

Power consumption of the PLB 6001:5.0 W

X30: Handwheel connection, permissive button / emergency stop

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

Pin layout of MB 620T FS:

Connecting terminals	Pin layout
1	Permissive button channel A / terminal 1 (+24 V NC)
2	Permissive button channel A / terminal 2
3	Permissive button channel B / terminal 1 (+24 V NC)
4	Permissive button channel B / terminal 2
5	Emergency stop channel A / supply for channel A (-TEST.A)
6	Emergency stop channel A / terminal 2
7	Emergency stop channel B / supply for channel B (-TEST.B)
8	Emergency stop channel B / terminal 2

The emergency stop inputs (pin 6 and pin 8) are active even if the HR handwheel has been deactivated via machine parameter MP_type (100601). It is therefore always necessary to wire the Emergency Stop inputs correctly.

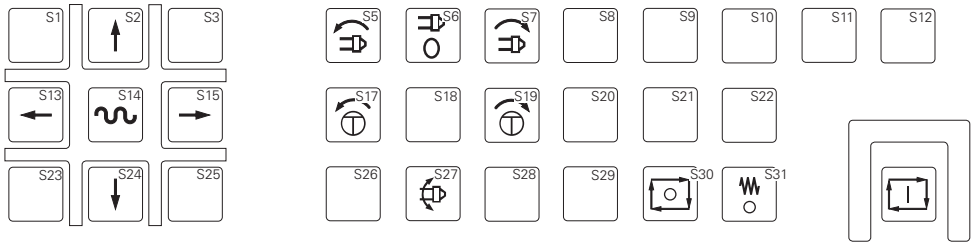
Danger

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

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

Connection X30 not used on MB 620T without FS.

Machine operating panel: Key assignment



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

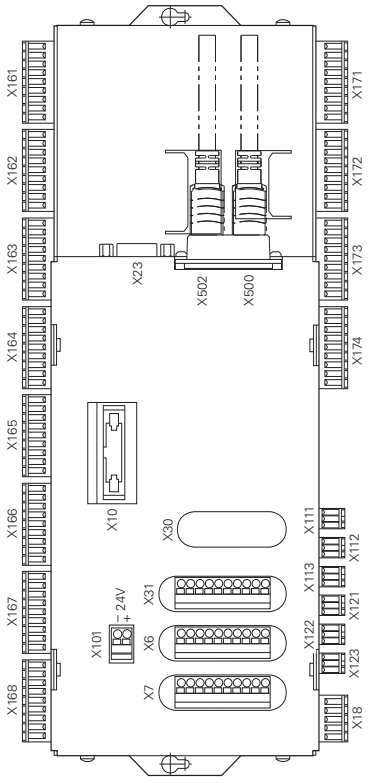
HEIDENHAIN offers the PLB 6001 HSCI adapter with HSCI interface for connecting an OEM-specific machine operating panel.

On the underside of the machine operating panel are terminal strips bearing the PLC inputs as well as the PLC outputs. Also, connection X23 for HR handwheels is on the underside of the PLB 6001 (See "Handwheel Input" on page 206).

Danger

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

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

PLB 6001 pin layout	Connector	Function	Page
	X500	HSCI output	124
	X502	HSCI input	124
	X6	PLC inputs	197
	X7	PLC outputs	196
	X10	Interface to keyboard and potentiometers	199
	X18	Reserved	–
	X23	Handwheel connection	206
	X30	Reserved	–
	X31	Permissive button, NC start, NC stop	198
	X101	24 V NC power supply	199
	X111	Potentiometer connection 1	205
	X112	Potentiometer connection 2	205
	X113	Potentiometer connection 3	205
	X121	Potentiometer connection 4	205
	X122	Potentiometer connection 5	205
	X123	Potentiometer connection 6	205
	X161	PLC inputs I0 to I7	203
	X162	PLC inputs I8 to I15	203
	X163	PLC inputs I16 to I23	203
	X164	PLC inputs I24 to I31	203
	X165	PLC inputs I32 to I39	203
	X166	PLC inputs I40 to I47	203
	X167	PLC inputs I48 to I55	203
	X168	PLC inputs I56 to I63	203
	X171	PLC outputs O0 to O7	205
	X172	PLC outputs O8 to O15	205
	X173	PLC outputs O16 to O23	205
	X174	PLC outputs O24 to O31	205
	⏚	Protective ground	–

Attention

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

X23: Handwheel input

Pin layout:

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

Note

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

Danger

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

X161 to X168: PLC inputs

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

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

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

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

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

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

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

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

X171 to X174: PLC outputs

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

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

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

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

X111 to X123: Potentiometer connection

Pin layout:

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

3.25 Handwheel Input

The following handwheels can be used with the MANUALplus 620.

- HR 130 panel-mounted handwheel
- HR 180 panel-mounted handwheel for connection to the position input at CC 61xx
- Up to three HR 150 panel-mounted handwheels via HRA 110 handwheel adapter
- HR 410 portable handwheel

The handwheel is connected to the MANUALplus 620 via X23 on the MB 620T machine operating panel.

3.25.1 HR 410 portable handwheel

The HR 410 is a portable electronic handwheel.

For the assignment of the keys of the HR 410 to the PLC inputs and outputs, see "HR 410 portable handwheel" auf Seite 981.

Pin layout

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

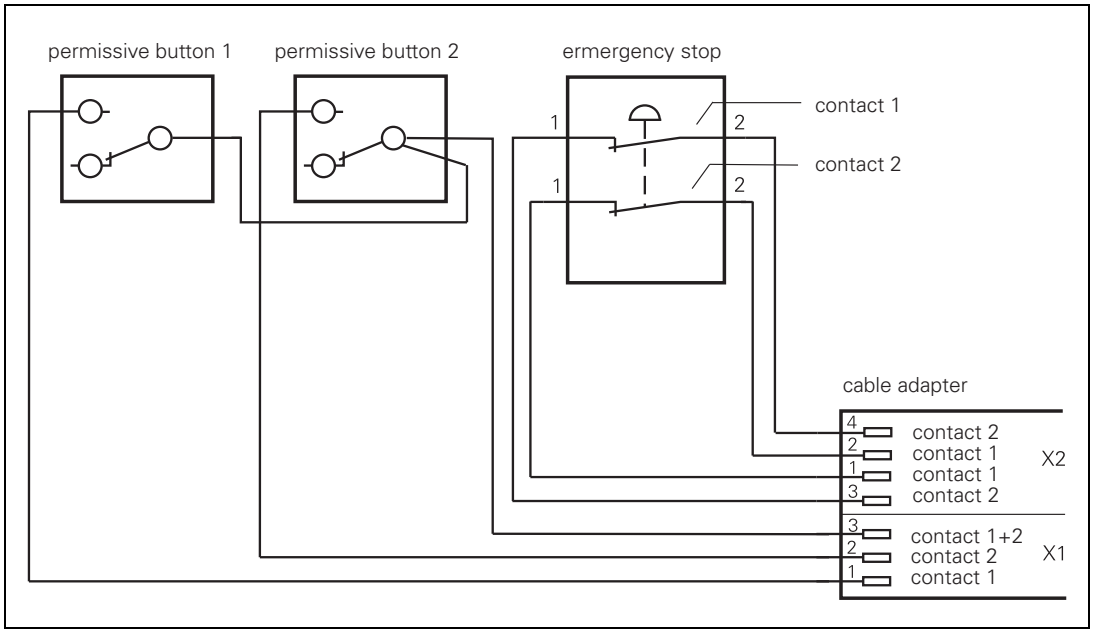
Extension cable, ID 281 429-xx			**Adapter cable ID 296 466-xx**			**Connecting cable ID: see "Introduction" chapter**			**HR 4xx**	
D-sub connector (male) 9-pin		**D-sub cnnctr. (female) 9-pin**	**D-sub connector (male) 9-pin**		**Cplng. on mntg. base (female) (5+7)-pin**	**Cnnctr. (male) (5+7)-pin**		**Cnnctr. (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	

Extension cable, ID 281 429-xx		Adapter cable ID 296 466-xx		Connecting cable ID: see "Introduction" chapter			HR 4xx	
D-sub connector (male) 9-pin	D-sub cnnctr. (female) 9-pin	D-sub connector (male) 9-pin	Cplng. on mntg. base (female) (5+7)-pin	Cnnctr. (male) (5+7)-pin		Cnnctr. (female) (5+7)-pin	Connector (male) (5+7)-pin	
			6	6	BK	6	6	
			7	7	RD/BL	7	7	
			5	5	Red	5	5	
			4	4	Blue	4	4	
			2	2	WH/ GN	2	2	
			3	3	BN/ GN	3	3	
			1	1	GY/PK	1	1	
		WH/ BN	3	Contacts 1 + 2				
		WH/ YL	2	Contact 2 (left) permissive button				
		WH/ GN	1	Contact 1 (right)				
		WH/ BL	1	Contact 1				
		WH/ RD	2	Contact 1 EMERGENCY STOP				
		YL/BK	3	Contact 2				
		WH/ BK	4	Contact 2				

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

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

Internal wiring of the contacts for the EMERGENCY STOP and permissive buttons:



Additional components		ID
Dummy plug for EMERGENCY STOP circuit		271 958-03
Connecting cables		
	Spiral cable	312 879-01
	Normal cable	296 467-xx
	With metal armor	296 687-xx
Plug-in terminal strips for advance ordering		
	3-pin terminal block	266 364-06
	4-pin terminal block	266 364-12

3.25.2 HR 130 panel-mounted handwheel

The standard cable length for the HR 130 is 1 meter.

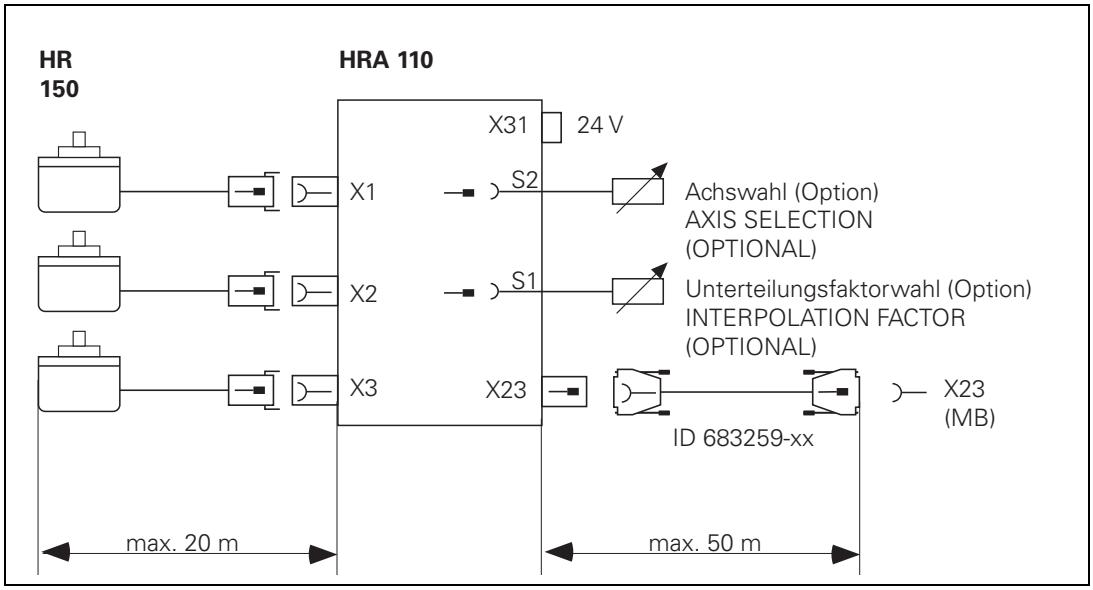
Pin layout for extension cable and handwheel:

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

3.25.3 HRA 110 handwheel adapter

With the handwheel adapter you can connect two or three HR 150 panel-mounted handwheels to the MANUALplus 620.

The first and second handwheels are assigned to the X and Y axes. The third handwheel can be assigned through a selection switch (option).



An additional switch enables you to select, for example, the subdivision factor for the handwheel. In the PLC you must evaluate the current position of the handwheel selection switch and activate the corresponding interpolation factor with Module 9036.

X1 to X3: Inputs on the HRA 110 for the HR 150

Pin layout:

HRA 110	
Connection (female) 9-pin	**Pin layout**
1	I_1 +
2	I_1 –
5	I_2 +
6	I_2 –
7	I_0 –
8	I_0 +
3	+5 V
4	0 V
9	Internal shield
Housing	External shield

X23: Connection to MB 620 / MB 620T

Pin layout on the HRA 110:

HRA 110	
D-sub connection (female) 9-pin	**Pin layout**
1	RTS
2	0 V
3	CTS
4	+12 V +0.6 V (U_V)
5	Do not assign
6	DSR
7	RxD
8	TxD
9	DTR
Housing	External shield

X31: HRA 110 supply voltage

Pin layout on the HRA 110:

Attention

The power supply of the PLC must not be used simultaneously for the HRA 110, otherwise the metallic isolation of the PLC inputs/outputs would be bridged.

HRA 110	
Connecting terminal	**Pin layout**
1	+24 V– as per IEC 742 (VDE 551)
2	0 V

Maximum current consumption 200 mA.

HR 180 panel-mounted handwheel

The **HEIDENHAIN HR 180 panel-mounted handwheels** supply 1 V_{PP} signals. They are connected with the connecting cable ID 310 199-xx to the position inputs X201..X20x of the CC61xx or UEC11x (See "Position encoder input" on page 175).

3.26 CML 110 Capacitor Module

Danger

Before service or maintenance work, you must ensure that the CML 110 has been completely discharged.

The CML 110 (Capacitor Module Low Voltage) can be used to realize the LIFTOFF function if line power fails.

The LIFTOFF function can protect workpieces and tools from damage. When a power failure occurs and the LIFT OFF function is active, the MANUALplus 620 tries to lift the tool off of the contour, using the residual energy of the DC-link. In this case, the various enablings for operating the control system must be maintained during the LIFTOFF. The CML 110 ensures the 24-V supply for this.

Connection

The CML 110 capacitor module is connected via X1 parallel to the 24 V power supply (++/- -).

The two ++ and - - terminals of the CML 110 are each connected to each other internally. This makes it possible to connect several CMLs 110 in parallel without needing to use additional external terminals.

Utilizability

Calculating the utilizability of the CML 110:

A successful LIFTOFF mainly depends on sufficient energy being available in the dc-link of the inverter system. Generally it suffices if the energy in the dc-link is available for the duration of one second. The 24 V supply must also be ensured for precisely this time. The following formula can be used to check this:

$$t = R_L \times C \times \ln (U_O/U_C)$$

where:

t = time until U_C is reached
R_L = ohmic load of the consumers
C = capacitance of the CML (for CML 110 = 8.3 F)
ln = natural logarithm
U_O = output load of the power supply unit with which the CML is operated
U_C = lowest voltage at which the consumers still fulfill their function

Example:
During operation at 24 V, a total current of 10 A is required for switching the control components on. This corresponds to an ohmic load of 2.4 ohms. In addition, the voltage for the 24 V components may not sink below 18 V (e.g. switching voltage of the contactors), for example.
This means:
t = 2.4 Ohm x 8.3 F x ln(24 V/18 V)
t = **5.73 s**
If the line voltage fails, then in the best case the voltage will not fall below 18 V until 5.73 seconds have passed. This is significantly longer than 1 second, and so the CML 110 is suitable for LIFTOFF here.

If the capacitance of the CML 110 should not suffice, then you can also switch more than one CML 110 in parallel. However, here you must note that a maximum charging current of 2.4 A per CML 110 is to be expected at switch-on. The full power of the 24 V power supply unit can only be used once all CMLs have finished charging.

3.27 Connecting Cable: Specifications

Device	ID	Max. bend radius (rigid configuration)	Max. bend radius (frequent flexing)	Cable diameter
HSCI	618 893-xx	≥ 40 mm	≥ 100 mm	ø 6.8 mm
Position 1 V_{PP}	298 429-xx, 298 430-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
Position 1 V_{PP}	310 199-xx, 309 783-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Position EnDat	332 115-xx, 323 897-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Position EnDat	313 791-xx, 332 790-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
Speed 1 V_{PP}	289 440-xx, 336 376-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Speed EnDat	336 376-xx, 340 302-xx, 369 502-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
Analog output	290 110-xx, 290 109-xx	≥ 40 mm	[a]	ø 7.3 mm
TS 220	274 543-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
SE 640, SE 540	310 197-xx, 517 518-xx	≥ 40 mm, ≥ 10 mm	≥ 100 mm, ≥ 50 mm	ø 8 mm ø 4.5 mm
HR 130, HR 410 (extension)	281 429-xx	≥ 20 mm	[a]	ø 5.6 mm
HR 410 (VL)	296 466-xx	≥ 20 mm	[a]	ø 5.6 mm
HR 410	296 687-xx	≥ 40 mm	≥ 100 mm	ø 8 mm
RS232, 9-pin	355 484-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
RS232, 9-pin (extension)	366 964-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
RS232, 25-pin	365 725-xx	≥ 40 mm	≥ 100 mm	ø 7.1 mm
RS232, 25-pin (extension)	274 545-xx	≥ 20 mm	≥ 75 mm	ø 6 mm
USB[b]	354 770-xx	≥ 20 mm	≥ 75 mm	ø 4.5 mm
USB (with hub)[c]	624 775-xx	Cable like 354 770-xx, hub: ø ~ 20 mm, length ~ 115 mm		

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

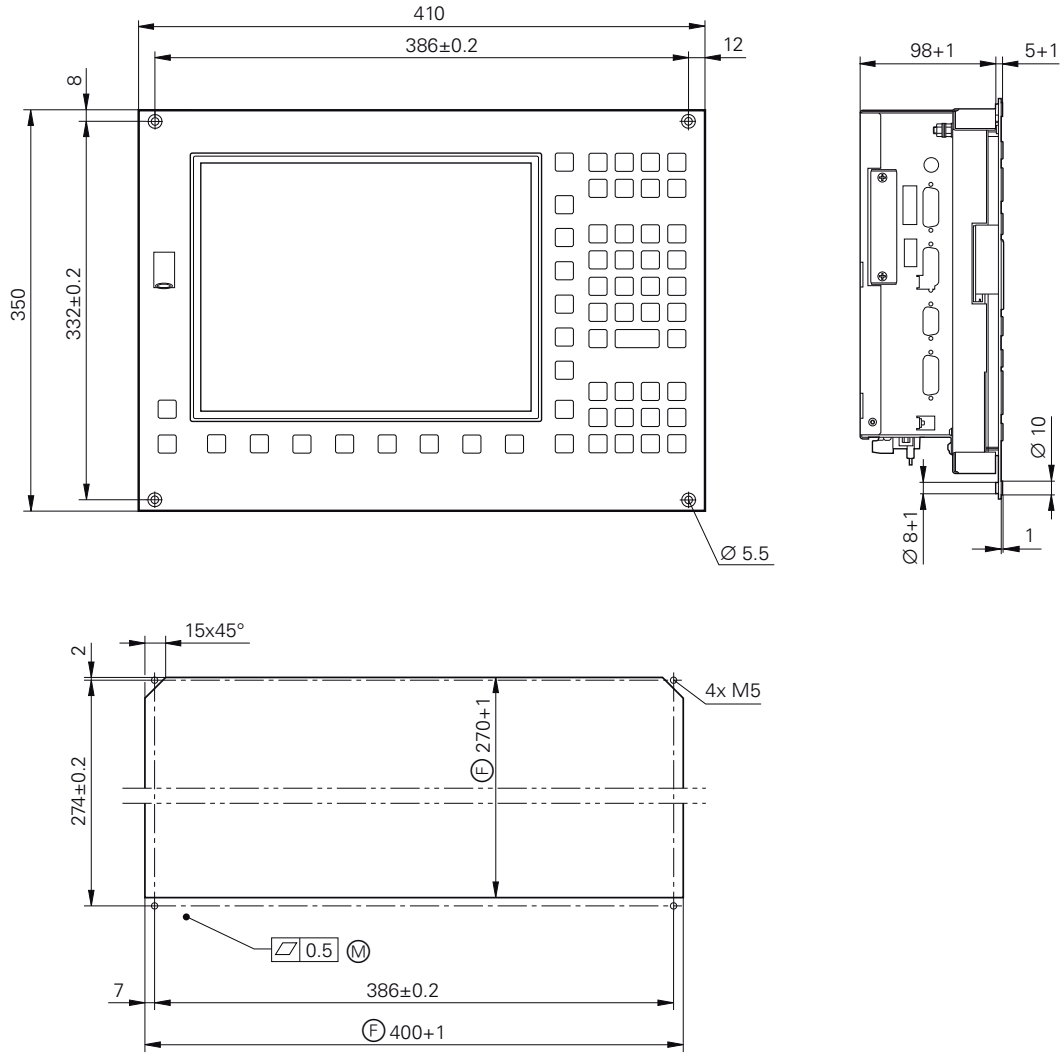
3.28 Dimensions

Note

All dimensions are in millimeters [mm].

3.28.1 MC 6110T

Weight: 5.6 kg



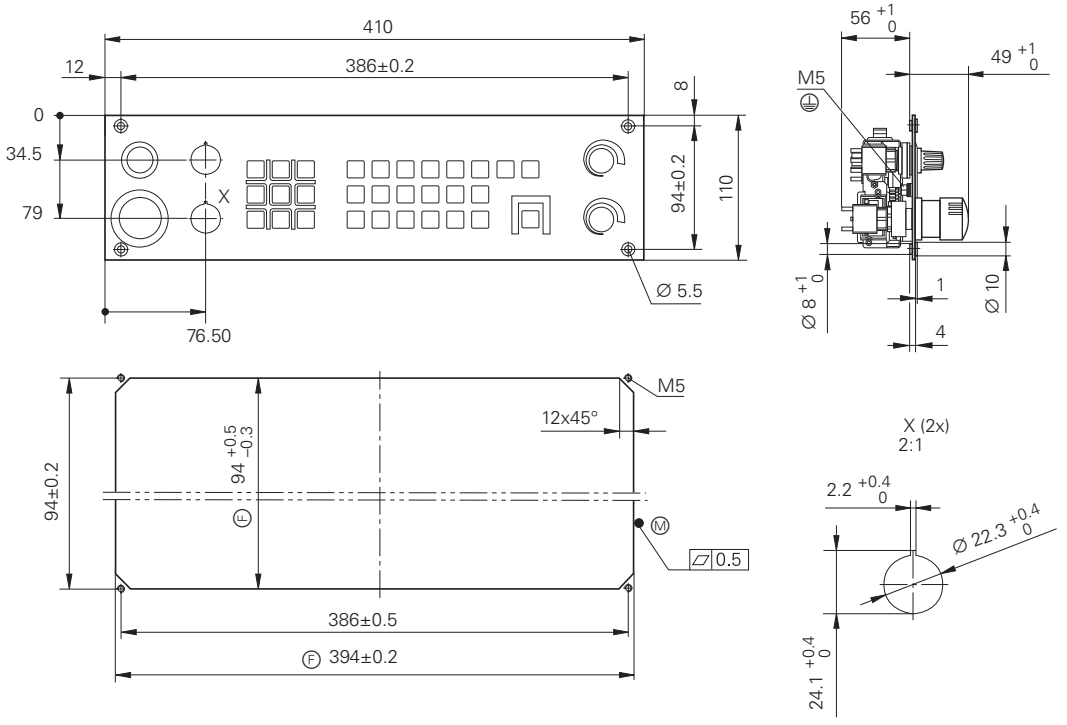
M: Mounting surface

F: Front panel opening

B: Space for air circulation

3.28.2 MB 620T

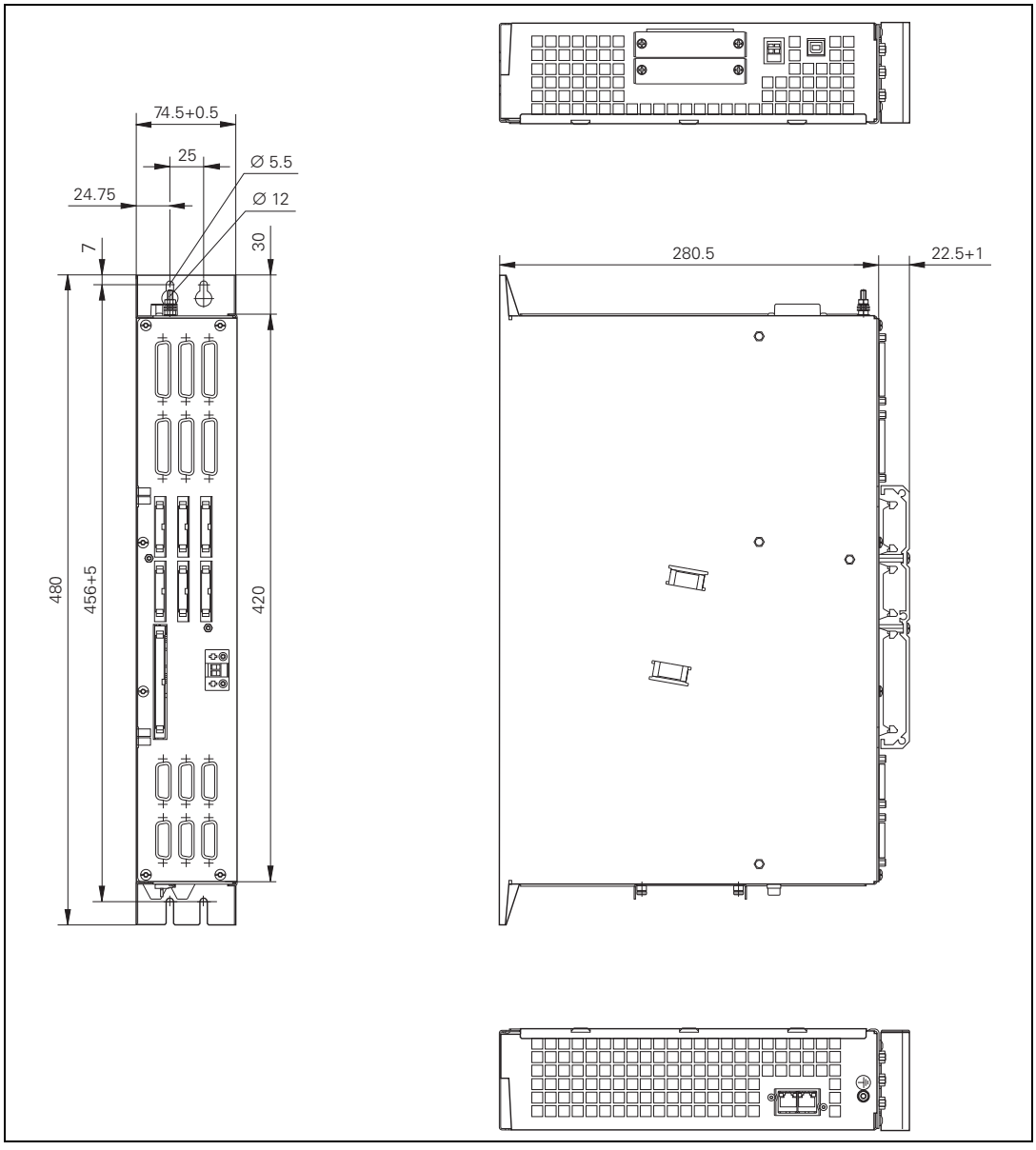
Weight: 0.9 kg

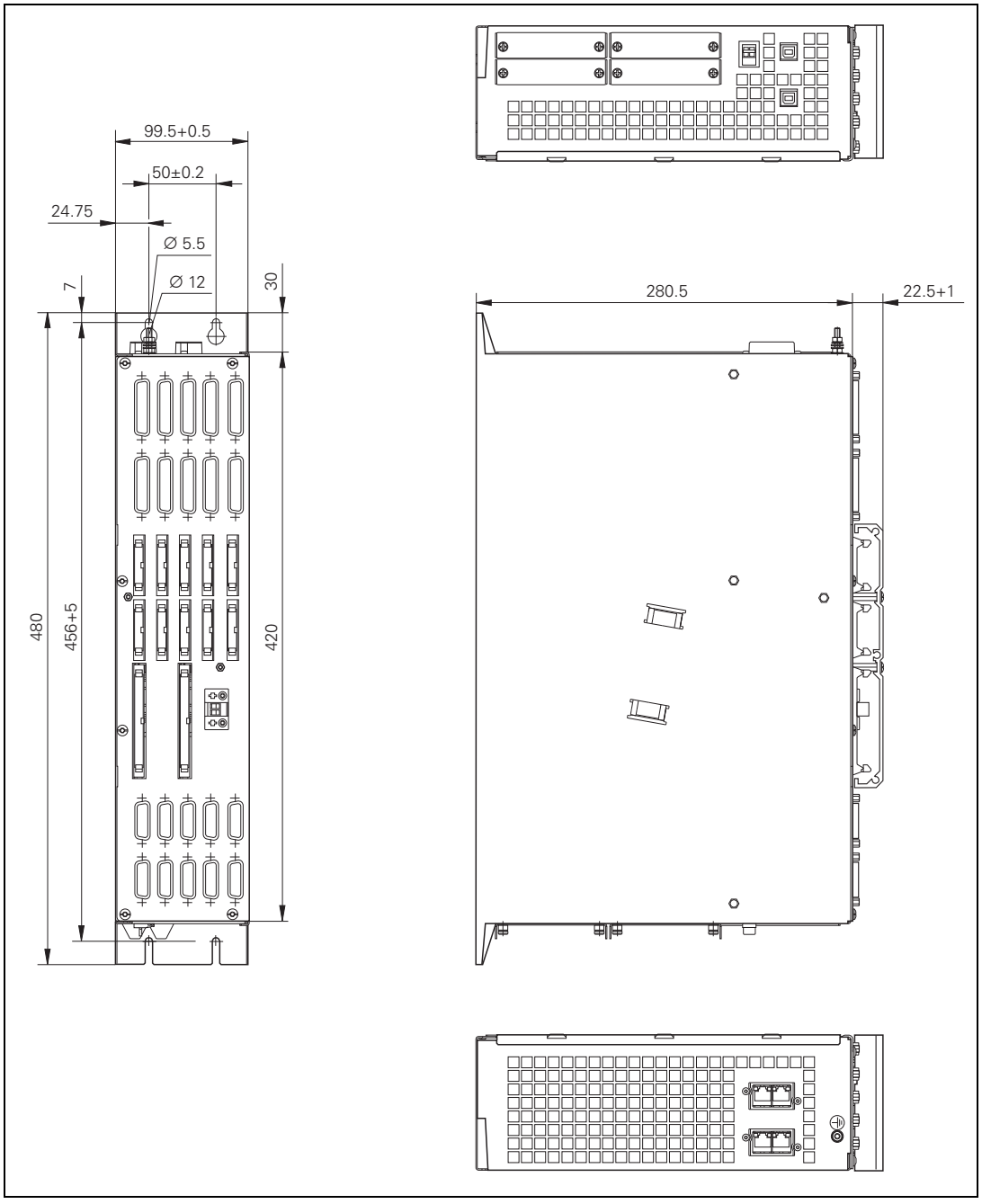


F: Front panel opening

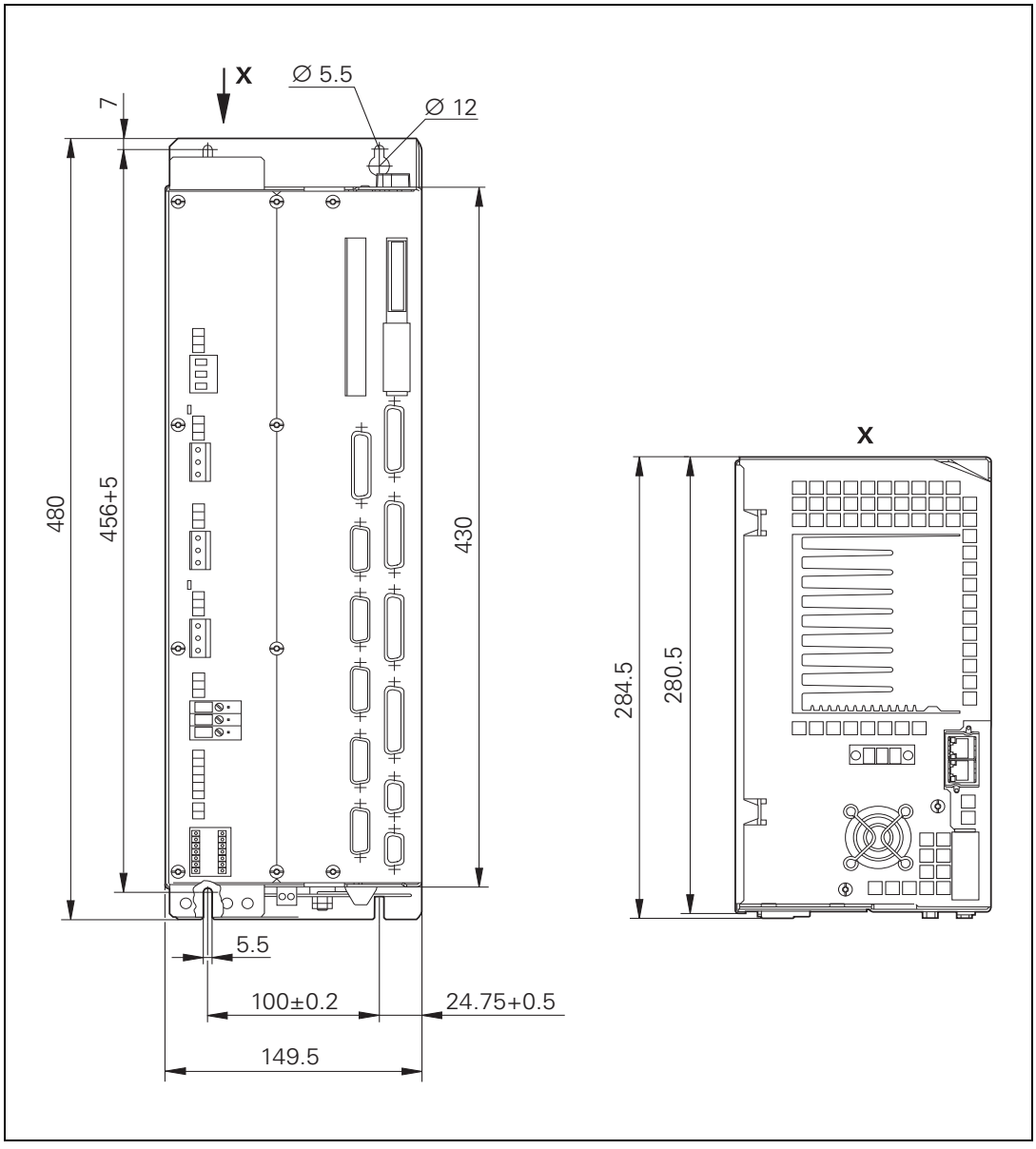
M: Mounting surface

3.28.3 CC 6106



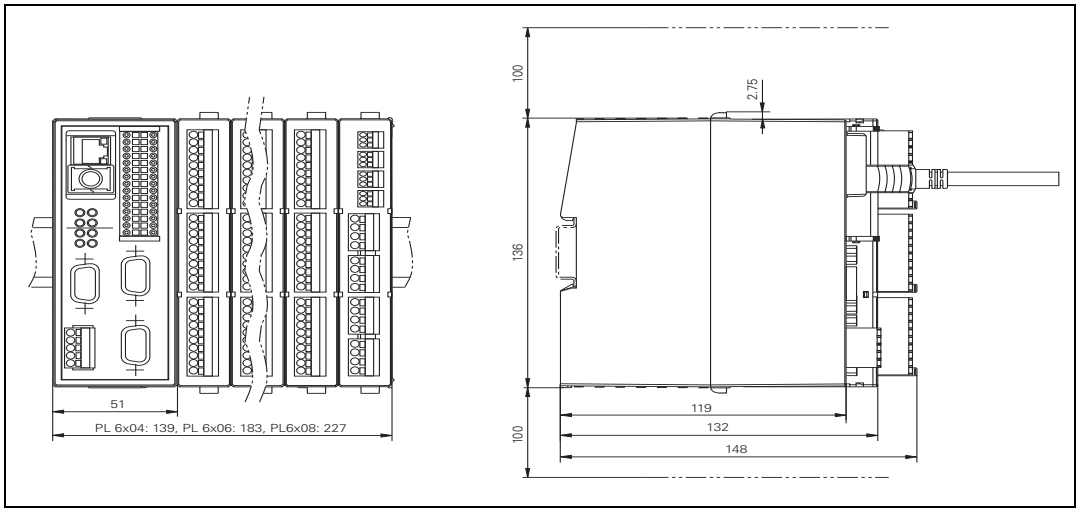

99.5+0.5
50±0.2
24.75
Ø 5.5
Ø 12
7
30
480
456+5
420
280.5
22.5+1

3.28.5 UEC 11x (FS)



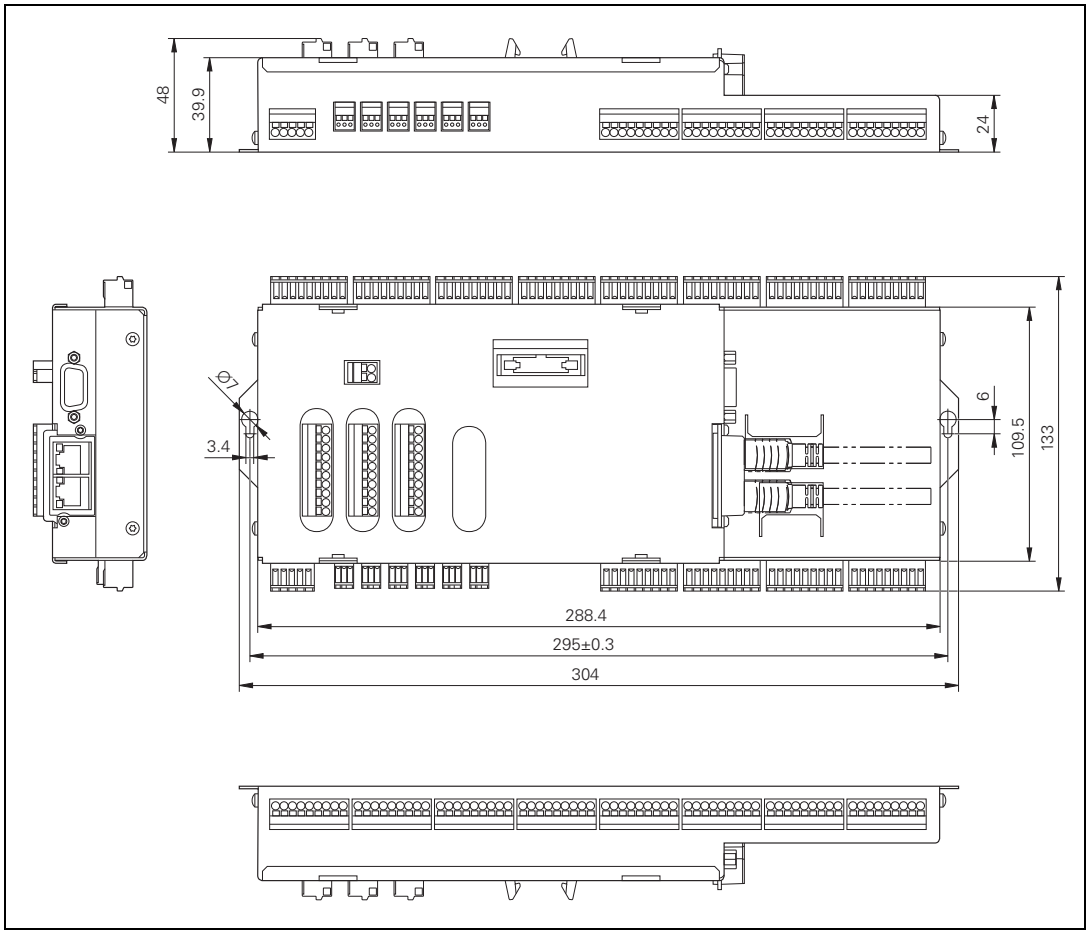
3.28.6 PL 6xxx (FS)

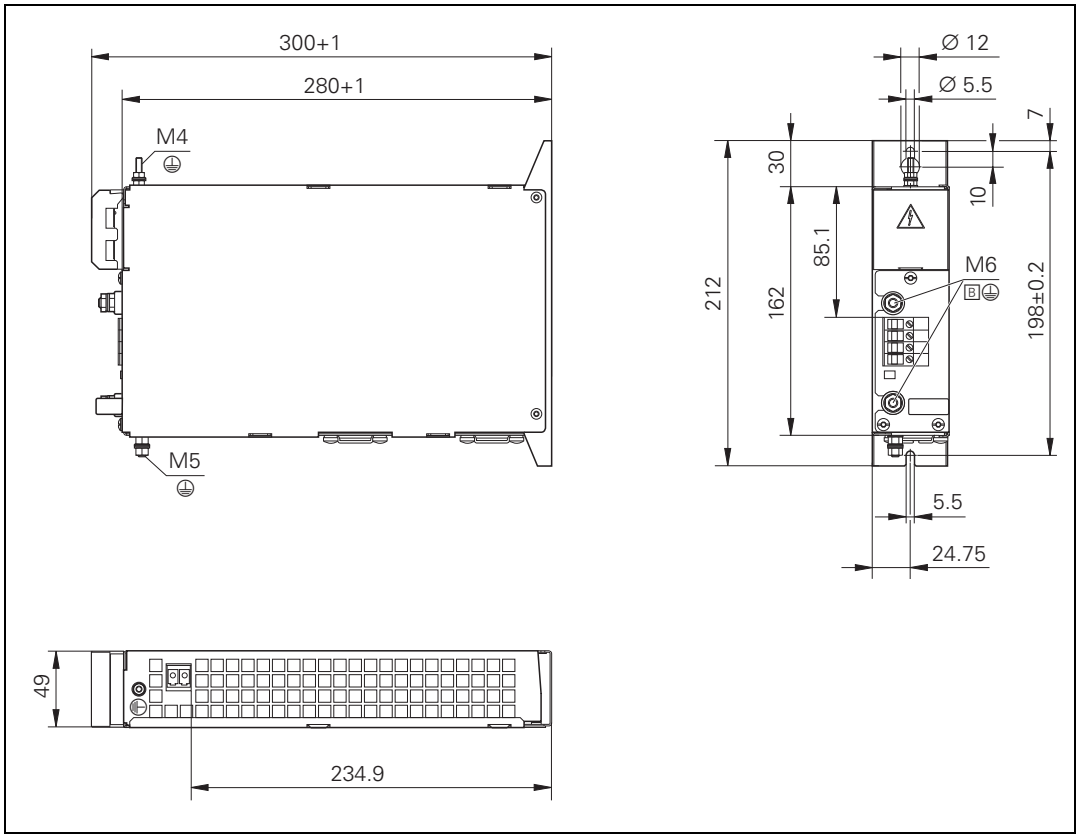
Weight: 1.0 kg



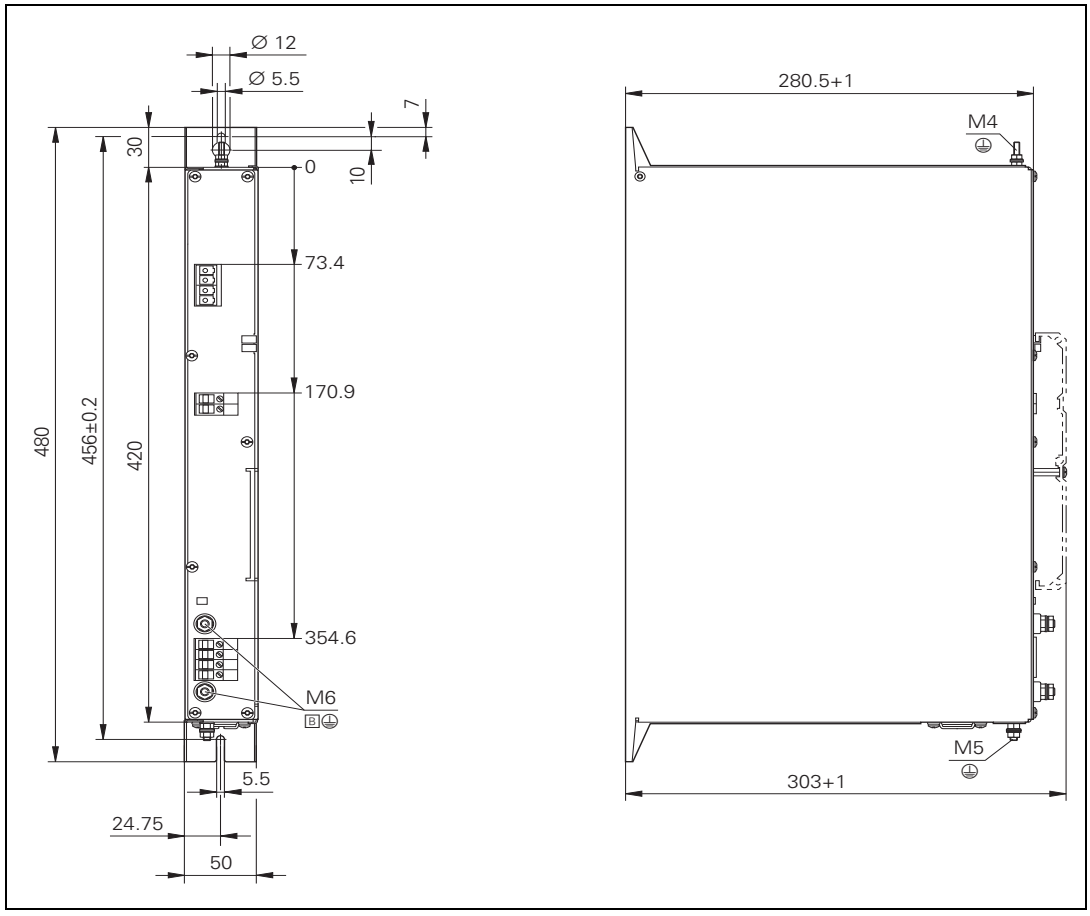
Dashed line: Space for air circulation

3.28.7 PLB 6001 (FS)



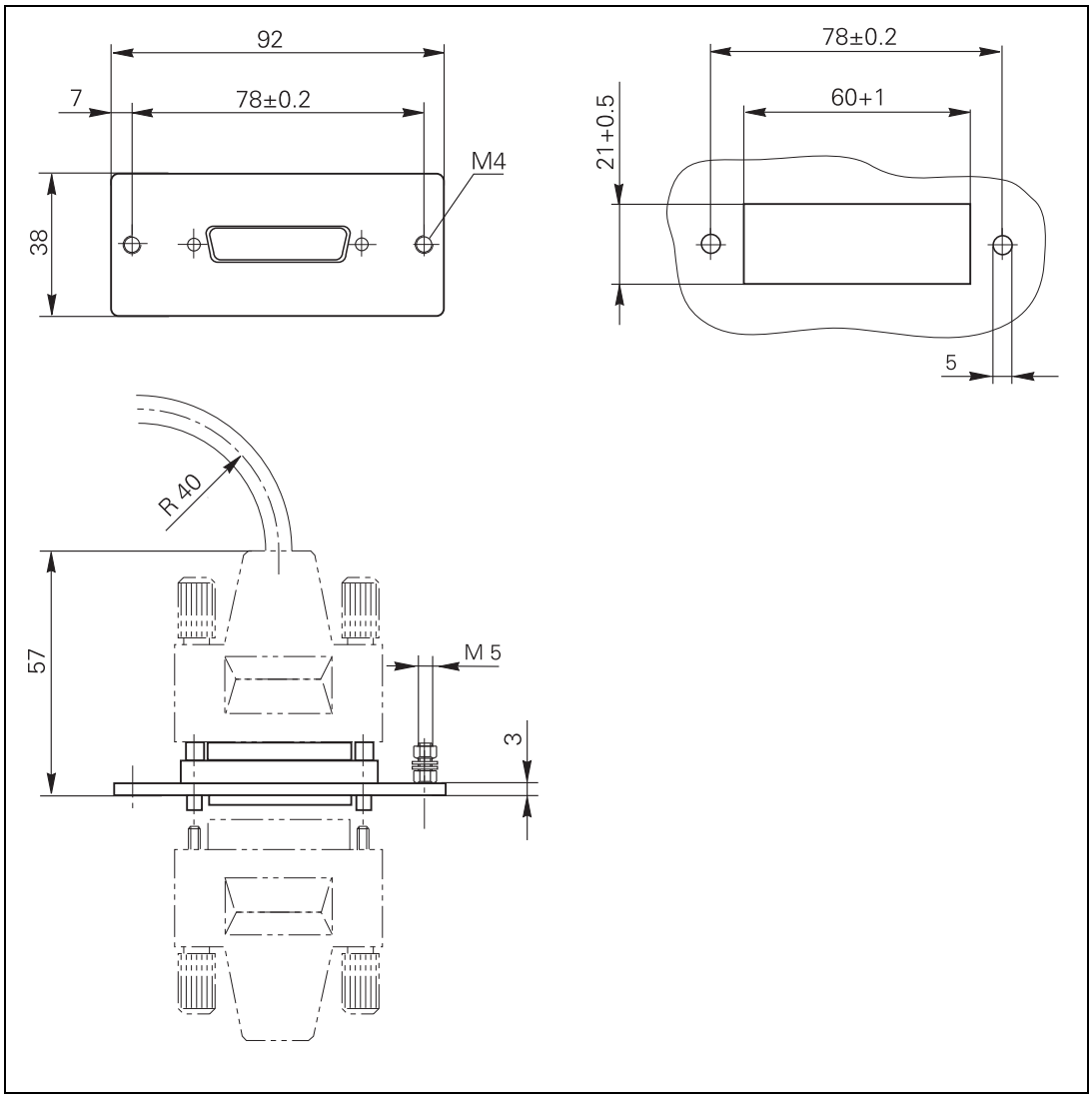

300+1
280+1
M4
M5
Ø 12
Ø 5.5
7
30
10
85.1
212
162
M6
198±0.2
5.5
24.75
49
234.9

3.28.9 PSL 135

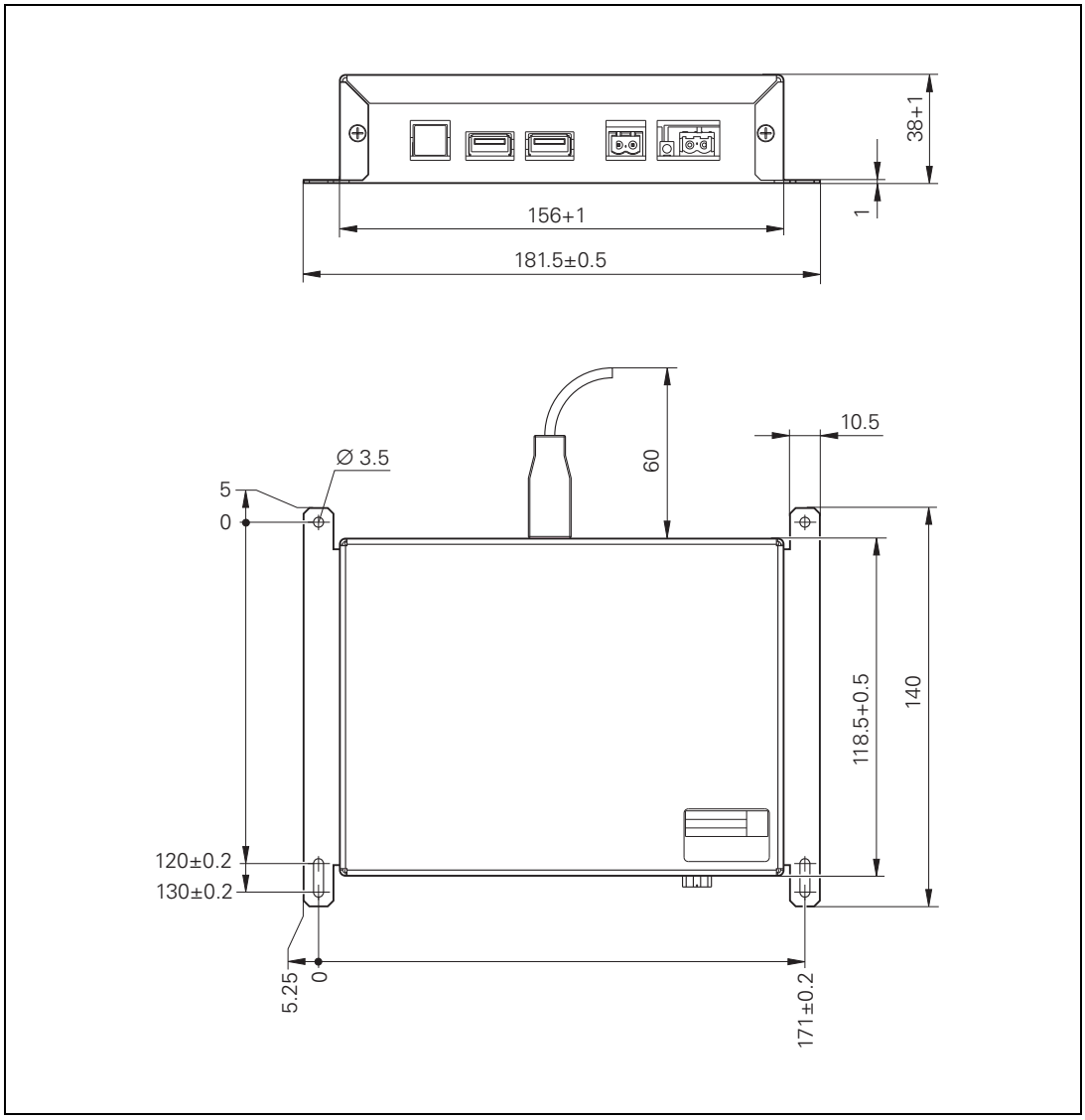

Ø 12
Ø 5.5
7
30
10
0
73.4
170.9
354.6
480
456±0.2
420
M6
5.5
24.75
50
280.5+1
M4
M5
303+1

3.28.10 Adapter block for the data interface

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

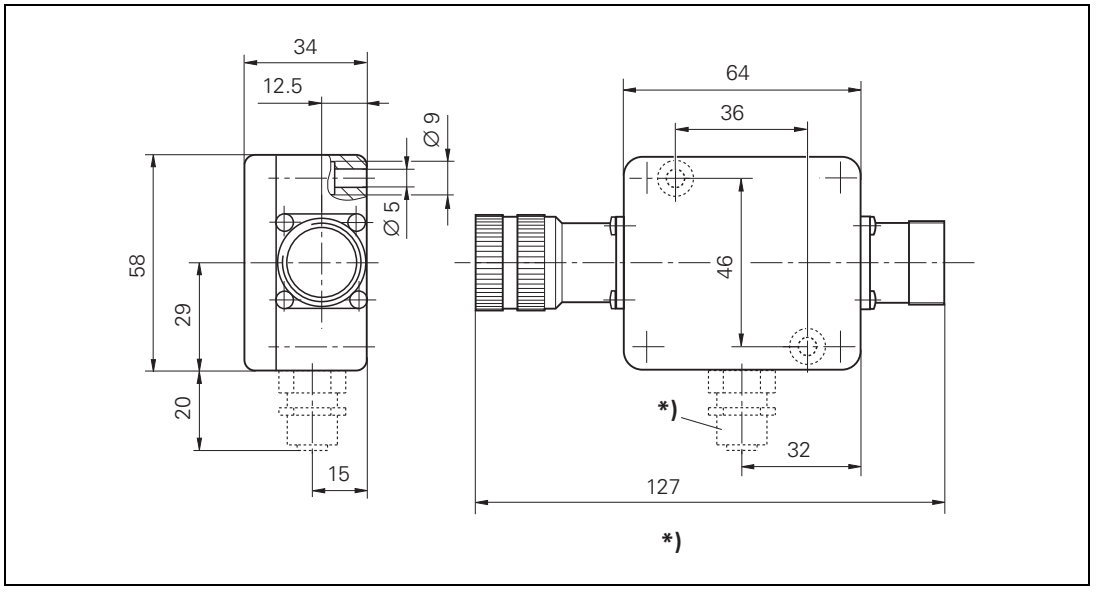


3.28.11 USB hub



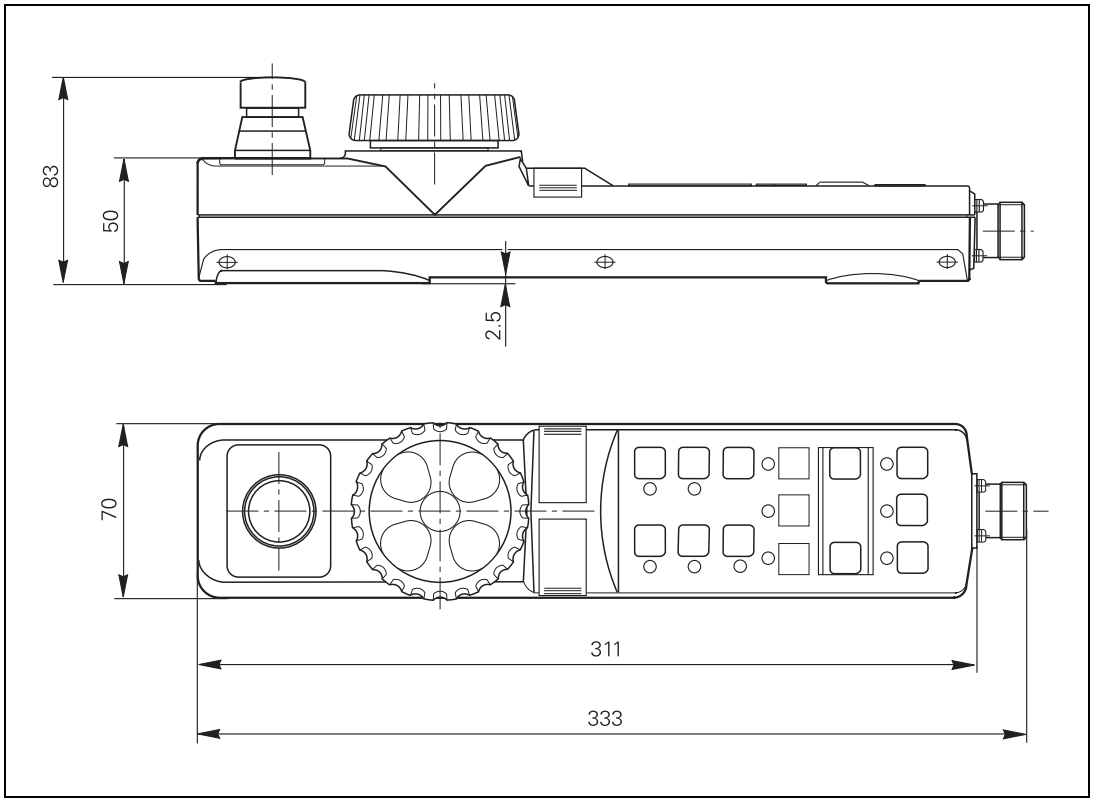
3.28.12 Line-drop compensator

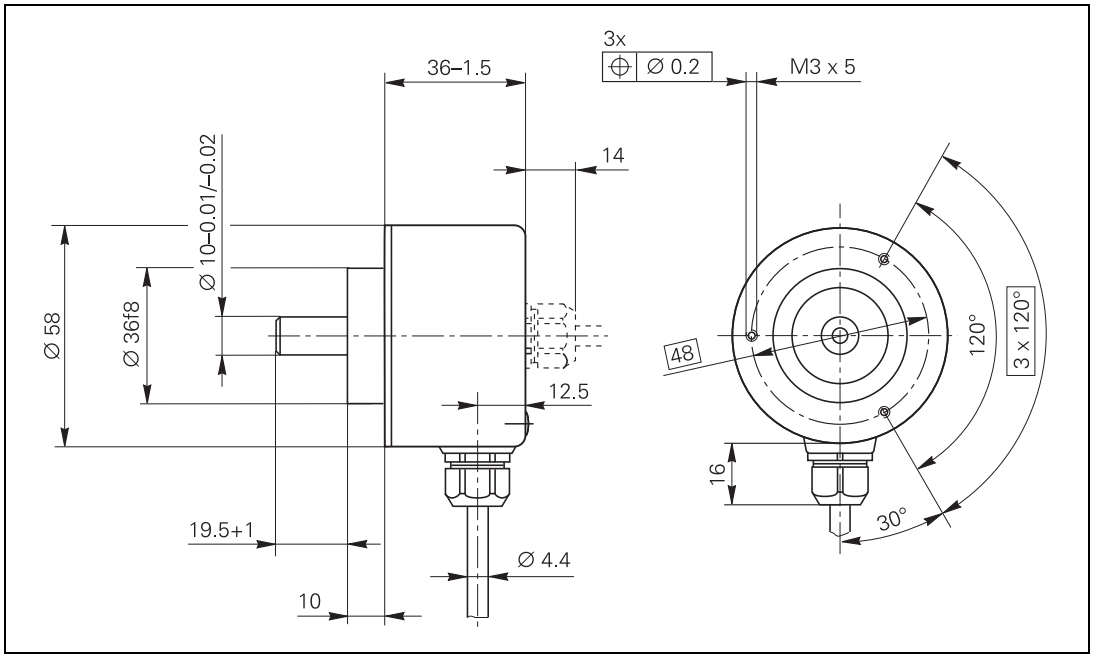
Line drop compensator for encoders with EnDat interface



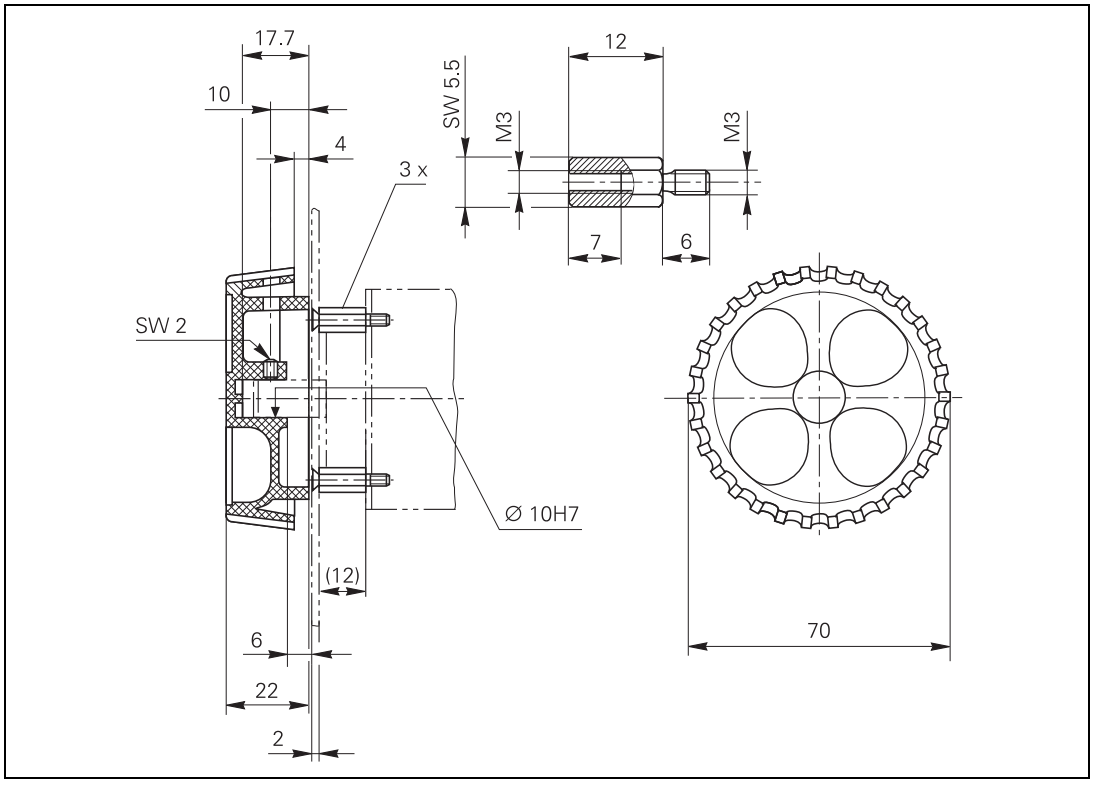
3.28.13 Handwheels

HR 410

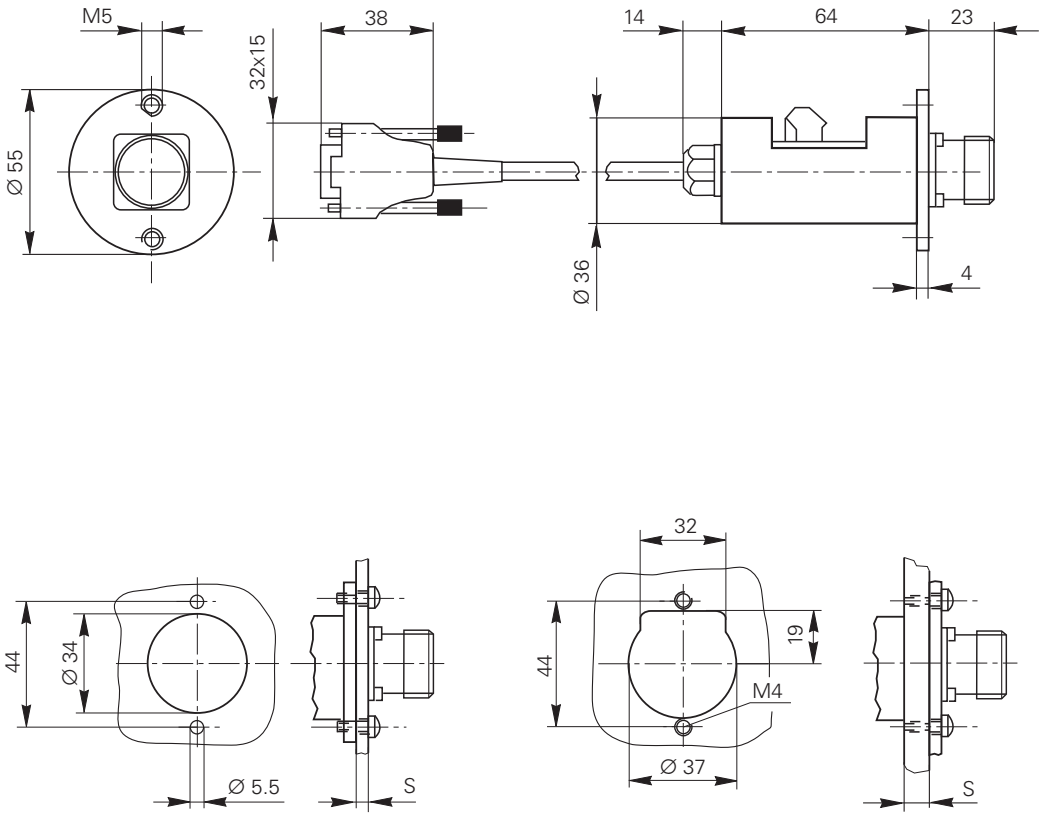




Control knob for HR 130



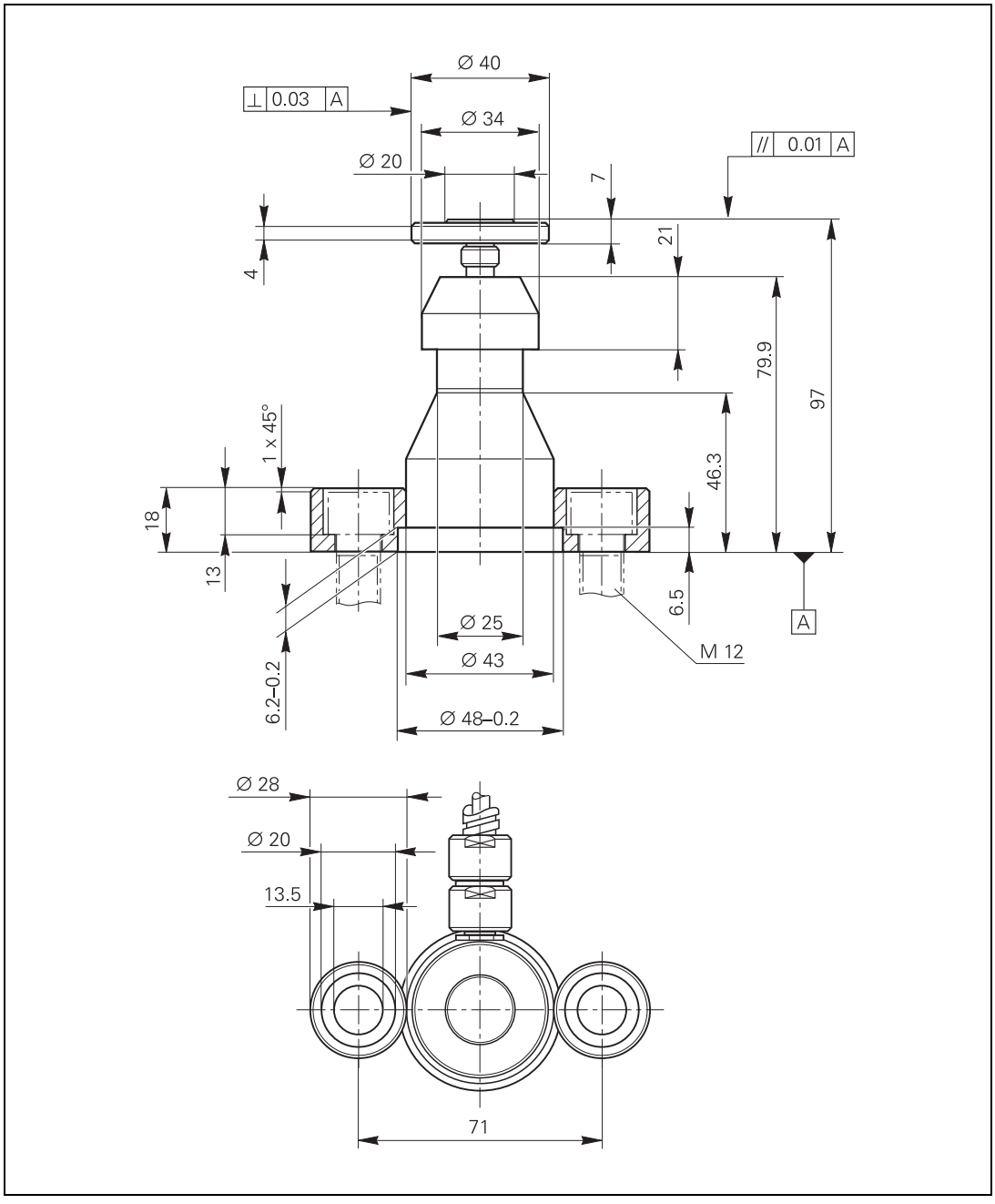
Adapter cable



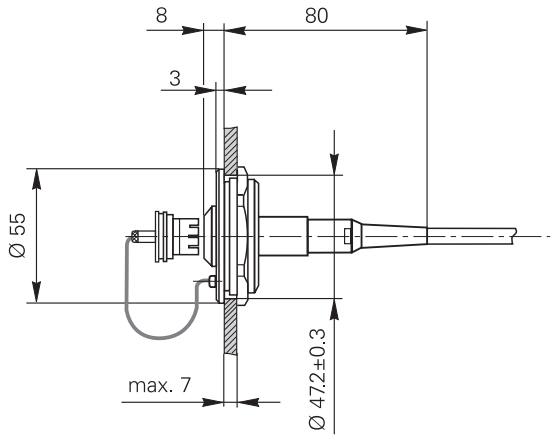
Mounting opening for wall thickness S ≥ 4

Mounting opening for wall thickness S > 4

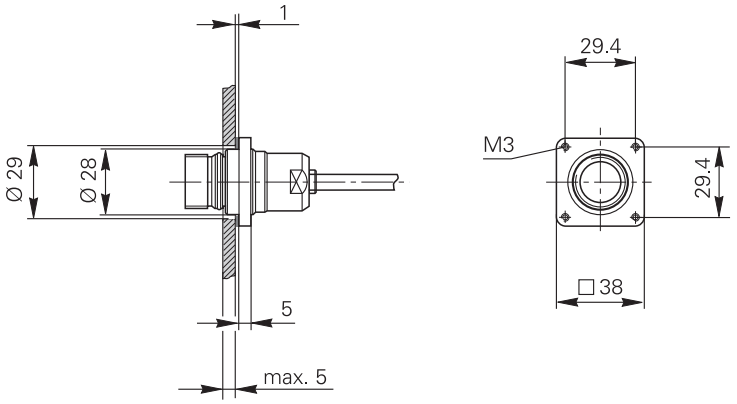
TT 130 / TT 140

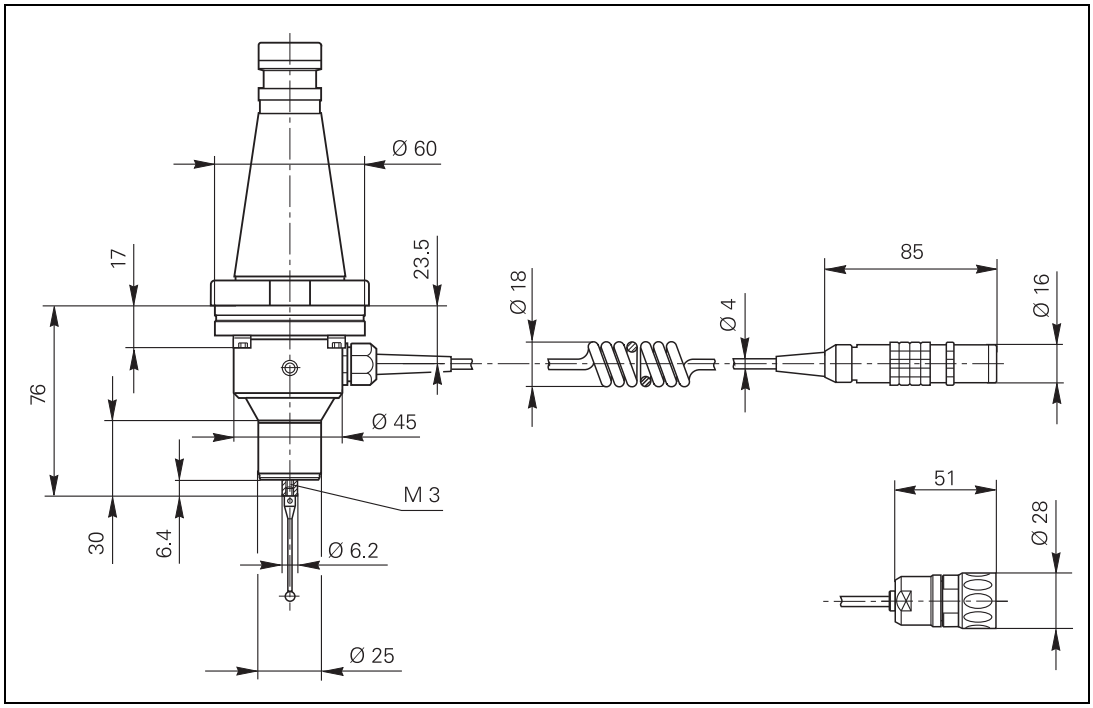

Ø 40
⊥ 0.03 A
Ø 34
Ø 20
// 0.01 A
7
21
4
79.9
97
1 x 45°
46.3
18
13
6.5
A
Ø 25
M 12
Ø 43
6.2–0.2
Ø 48–0.2
Ø 28
Ø 20
13.5
71

Mounting coupling for quick connection

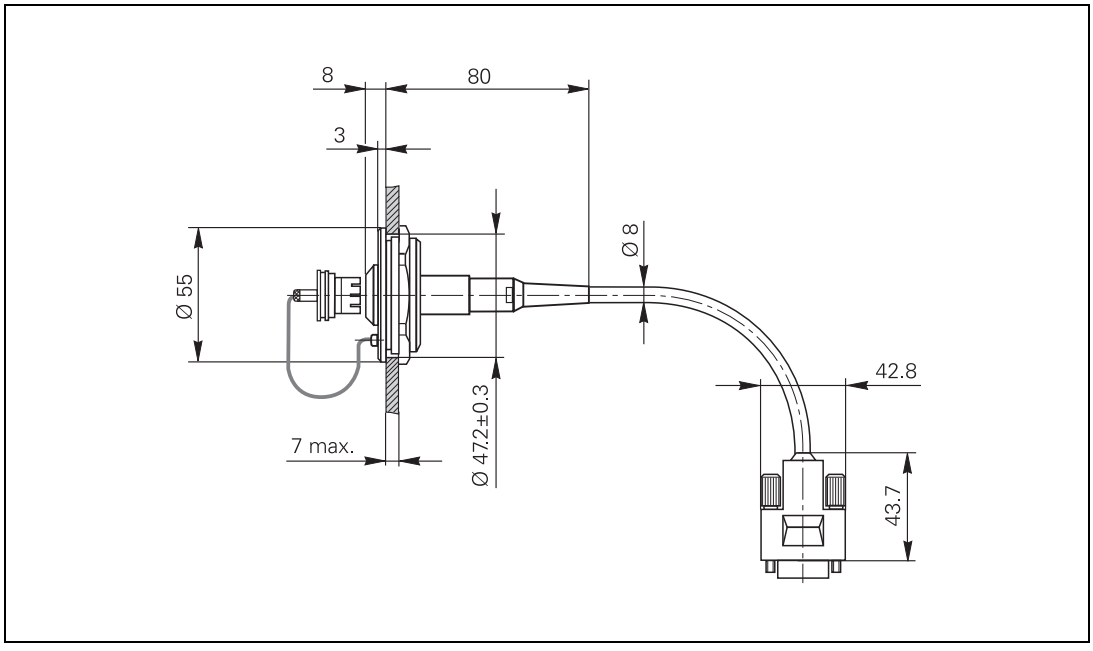


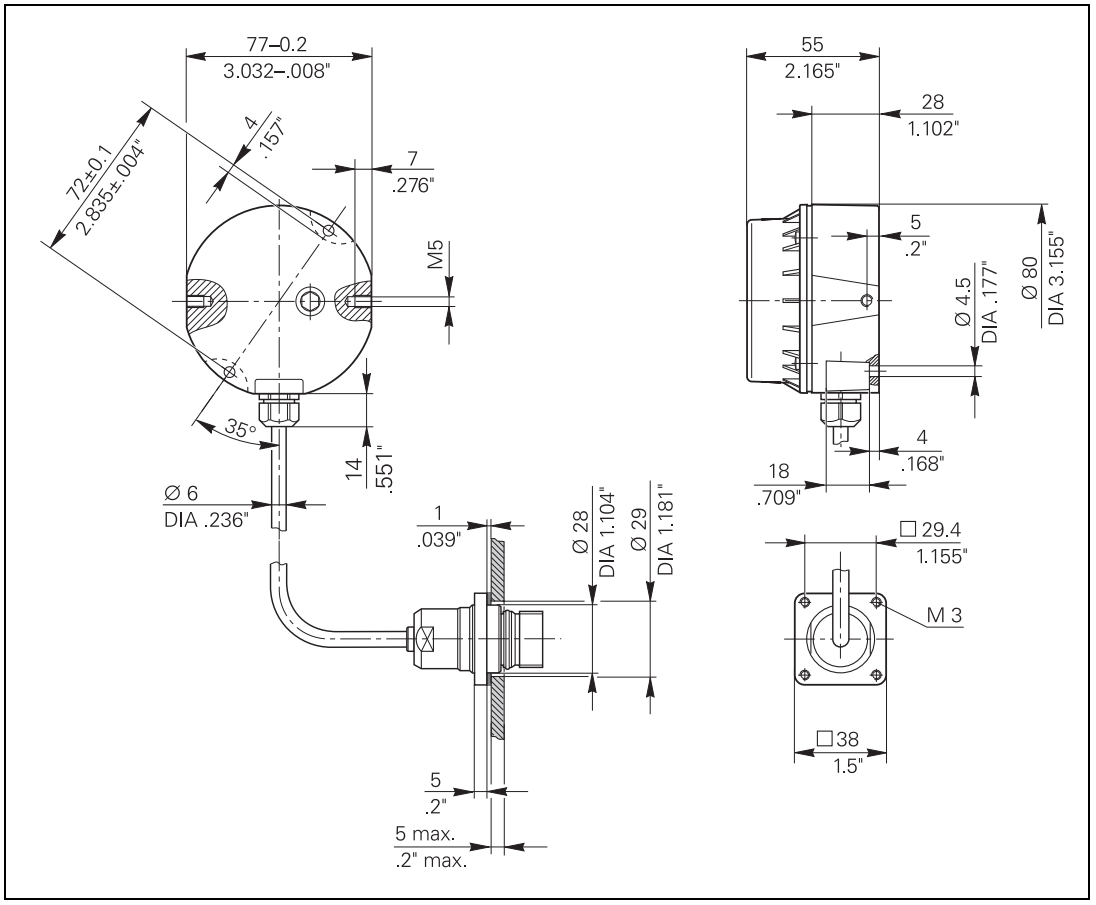
Mounting coupling for HEIDENHAIN standard connector

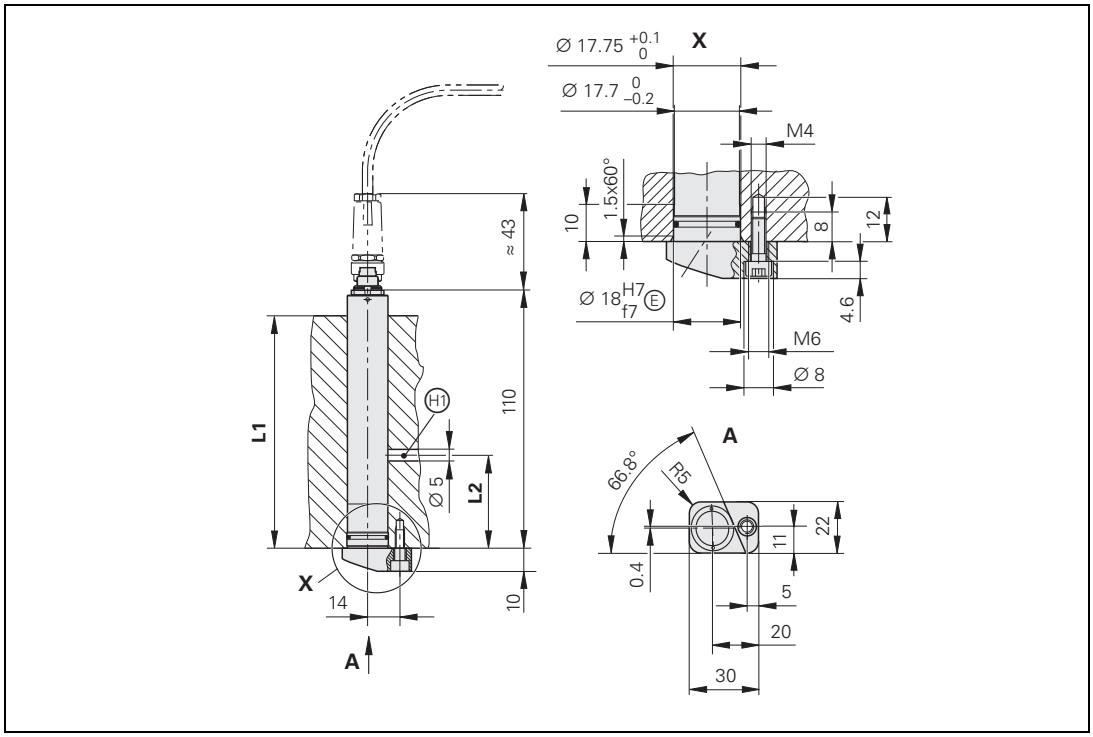


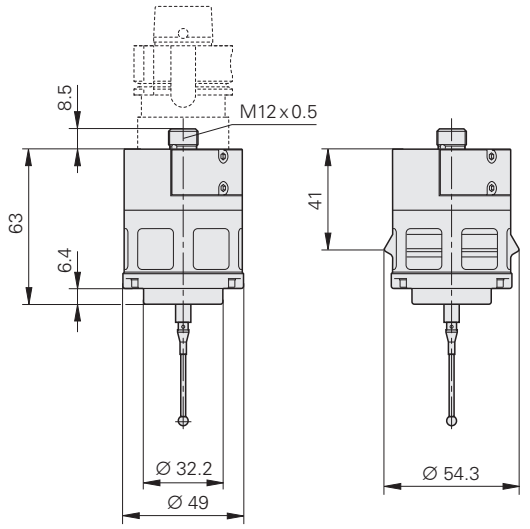


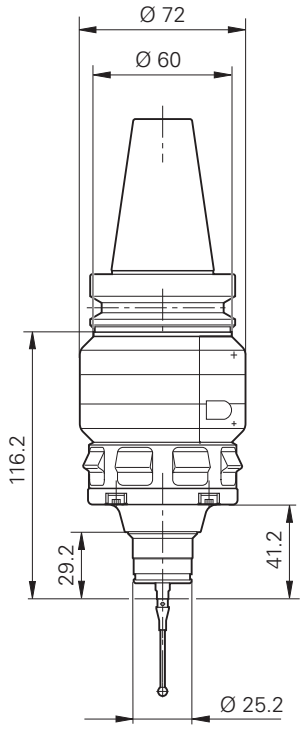
Adapter cable for
TS 120/TS 220



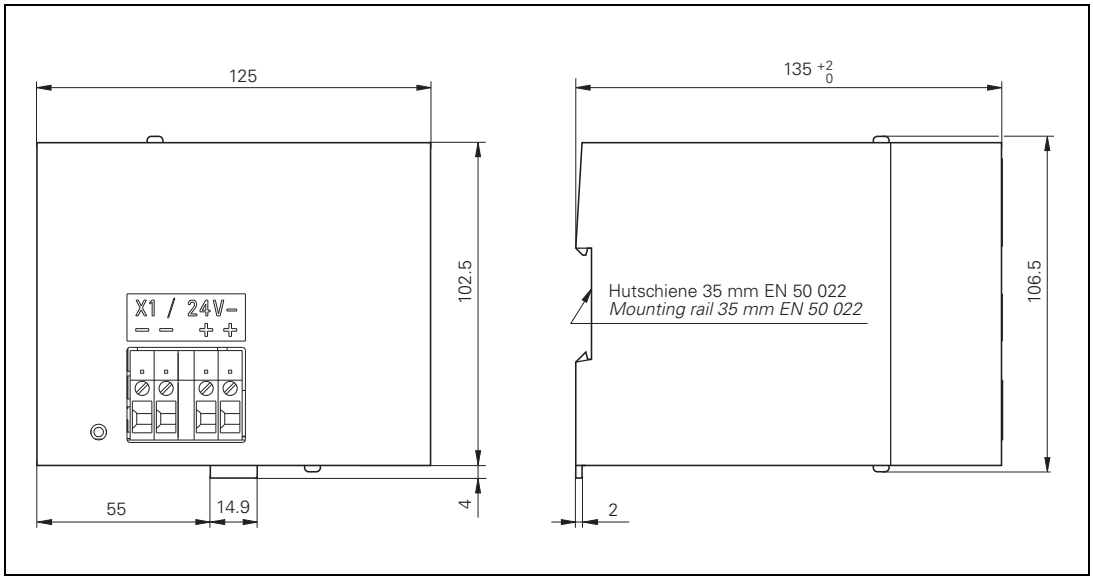

77–0.2
3.032–.008"
72±0.1
2.835±.004"
4
.157"
7
.276"
M5
35°
14
.551"
Ø 6
DIA .236"
1
.039"
Ø 28
DIA 1.104"
Ø 29
DIA 1.181"
5
.2"
5 max.
.2" max.
55
2.165"
28
1.102"
5
.2"
Ø 4.5
DIA .177"
Ø 80
DIA 3.155"
4
.168"
18
.709"
□ 29.4
1.155"
M 3
□38
1.5"


Ø 17.75 +0.1 0
X
Ø 17.7 0 −0.2
M4
1.5x60°
10
8
12
Ø 18 H7 f7 E
4.6
M6
Ø 8
≈ 43
110
L1
H1
Ø 5
L2
X
14
10
A
A
66.8°
R5
22
11
0.4
5
20
30

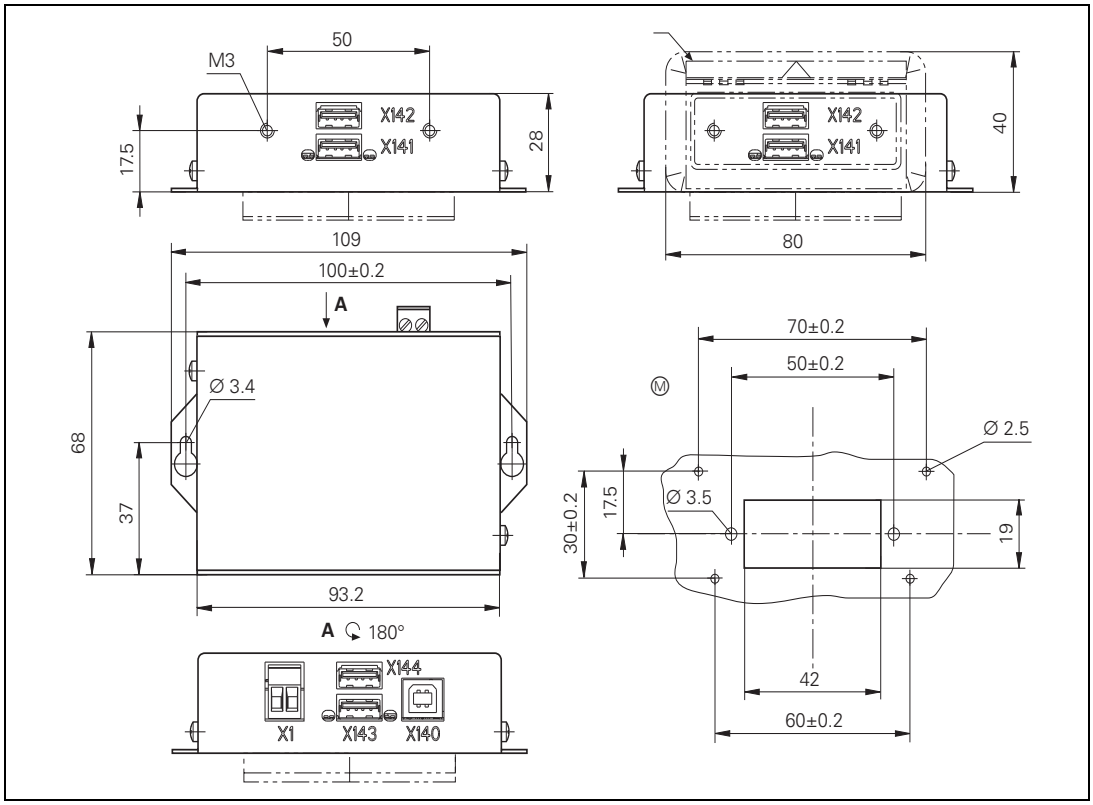

8.5
M12 x 0.5
63
6.4
41
Ø 32.2
Ø 49
Ø 54.3


Ø 72
Ø 60
116.2
29.2
41.2
Ø 25.2

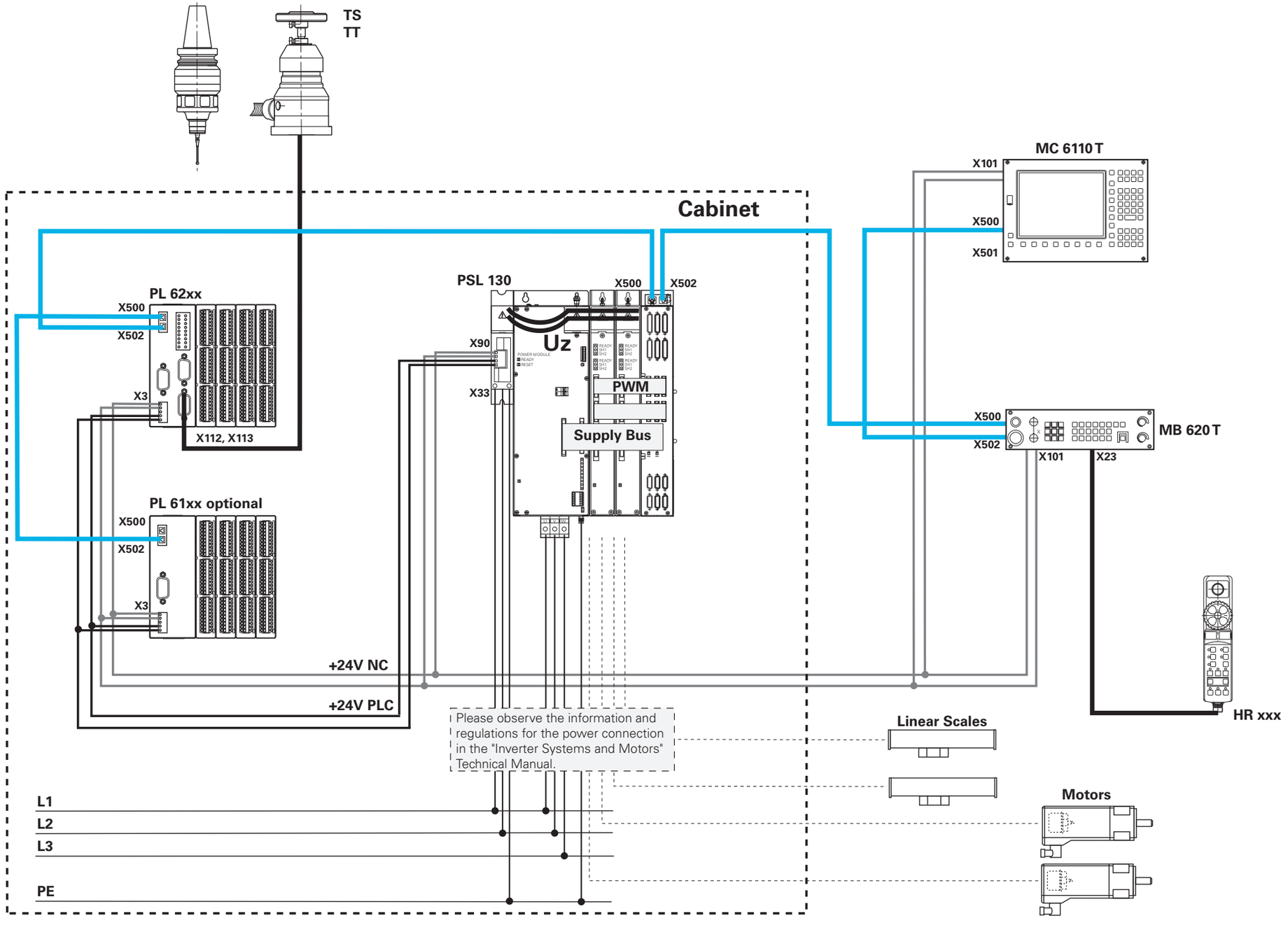
3.28.15 CML 110



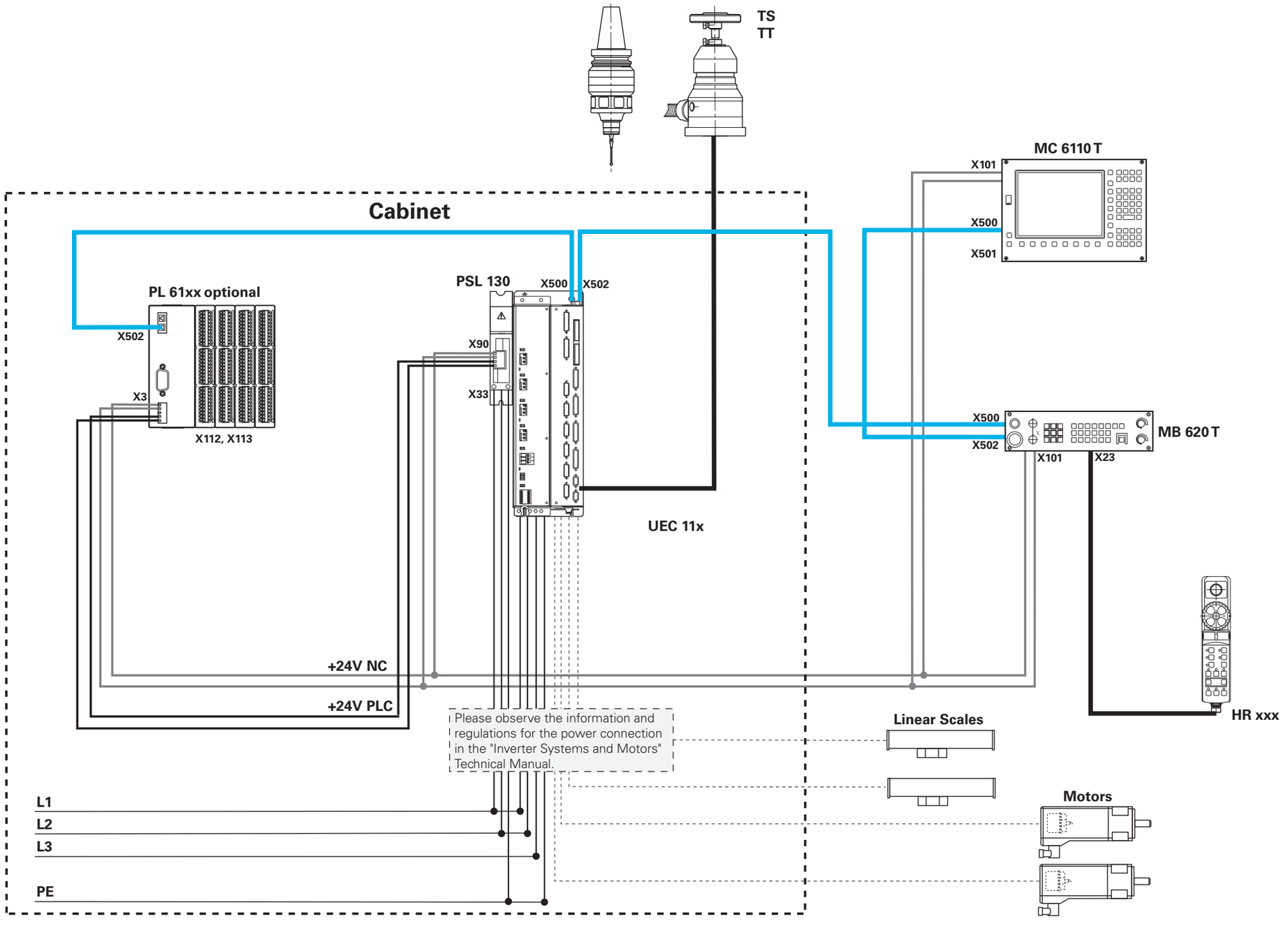
3.28.16 USB hub for operating panel

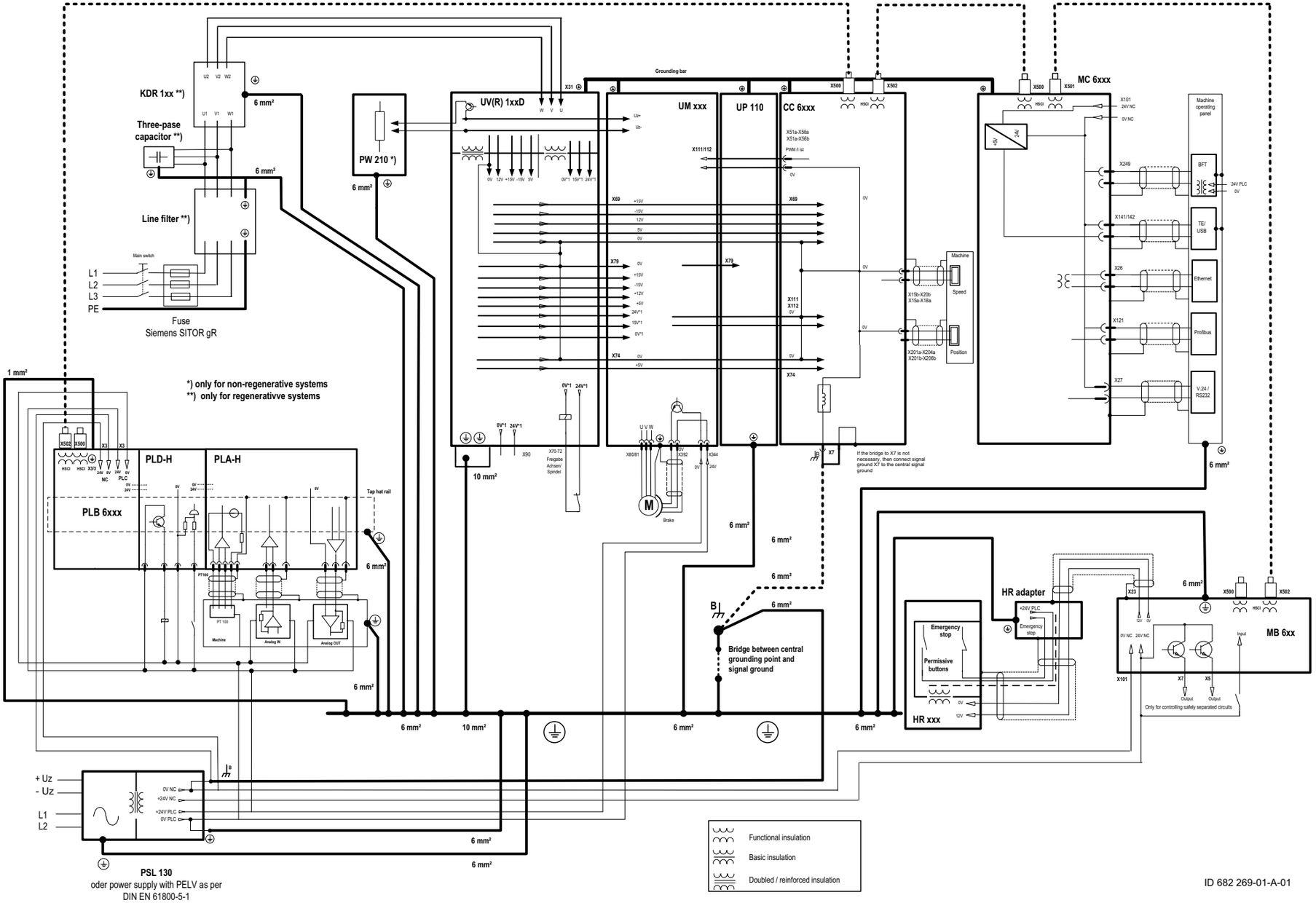


3.29 HSCI Connection Overview of the MANUALplus 620 with CC 61xx



3.30 HSCI Connection Overview of the MANUALplus 620 with UEC 11x

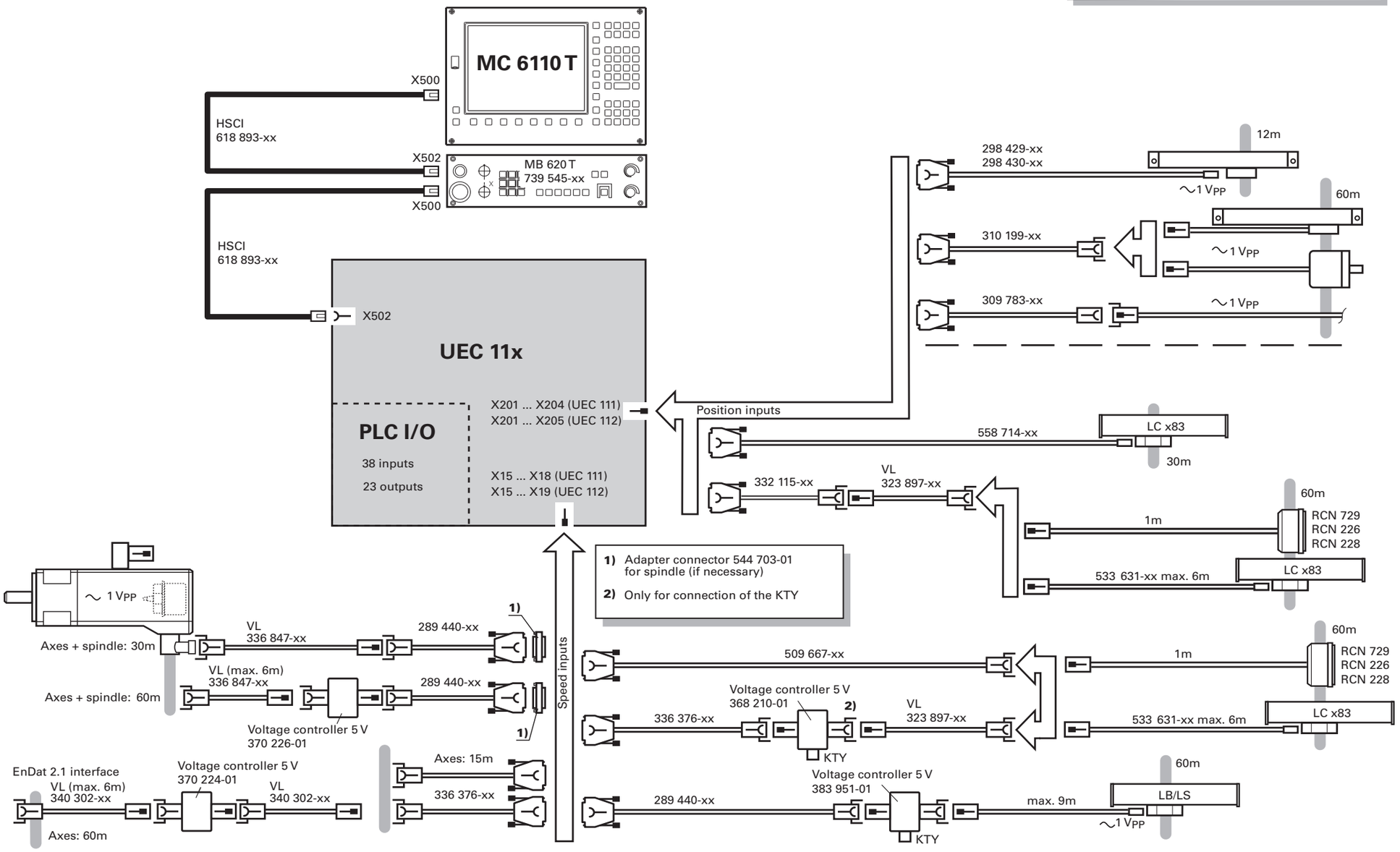



KDR 1xx **)
Three-pase capacitor **)
Line filter **)
Main switch
L1
L2
L3
PE
Fuse
Siemens SITOR gR
PW 210 *)
UV(R) 1xxD
UM xxx
UP 110
CC 6xxx
MC 6xxx
Grounding bar
*) only for non-regenerativve systems
**) only for regenerativve systems
PLD-H
PLA-H
PLB 6xxx
Tap hat rail
Brake
Bridge between central grounding point and signal ground
If the bridge to X7 is not necessary, then connect signal ground X7 to the central signal ground
HR adapter
HR xxx
Emergency stop
Permissive buttons
MB 6xx
Only for controlling safely separated circuits
+ Uz
- Uz
L1
L2
PSL 130
oder power supply with PELV as per DIN EN 61800-5-1
Functional insulation
Basic insulation
Doubled / reinforced insulation
ID 682 269-01-A-01

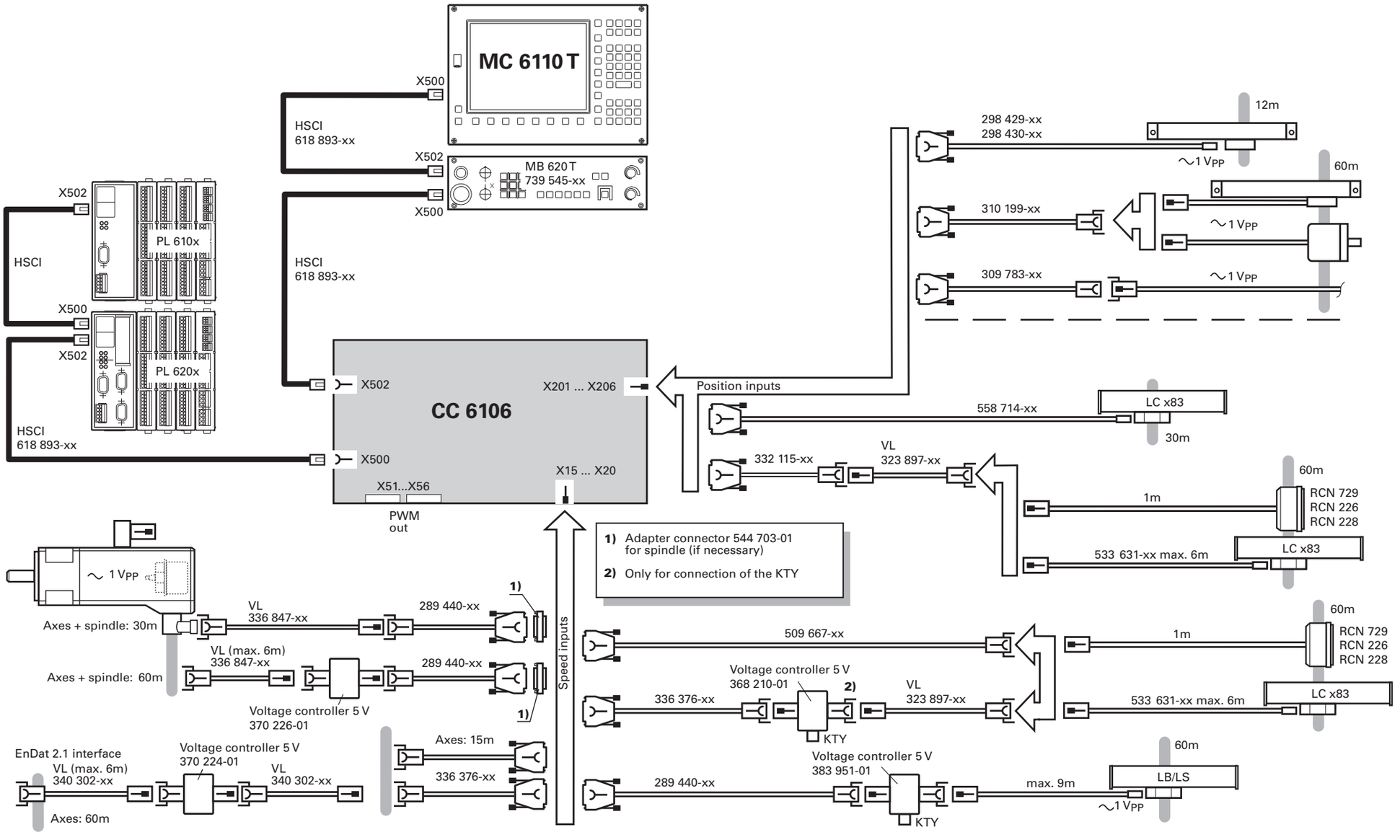
3.32 Basic Circuit Diagram for MANUALplus 620

You can find the current basic circuit diagrams in the download area of the HEIDENHAIN FileBase on the Internet at http://filebase.heidenhain.de.
For this area you need access rights that you can request via e-mail.

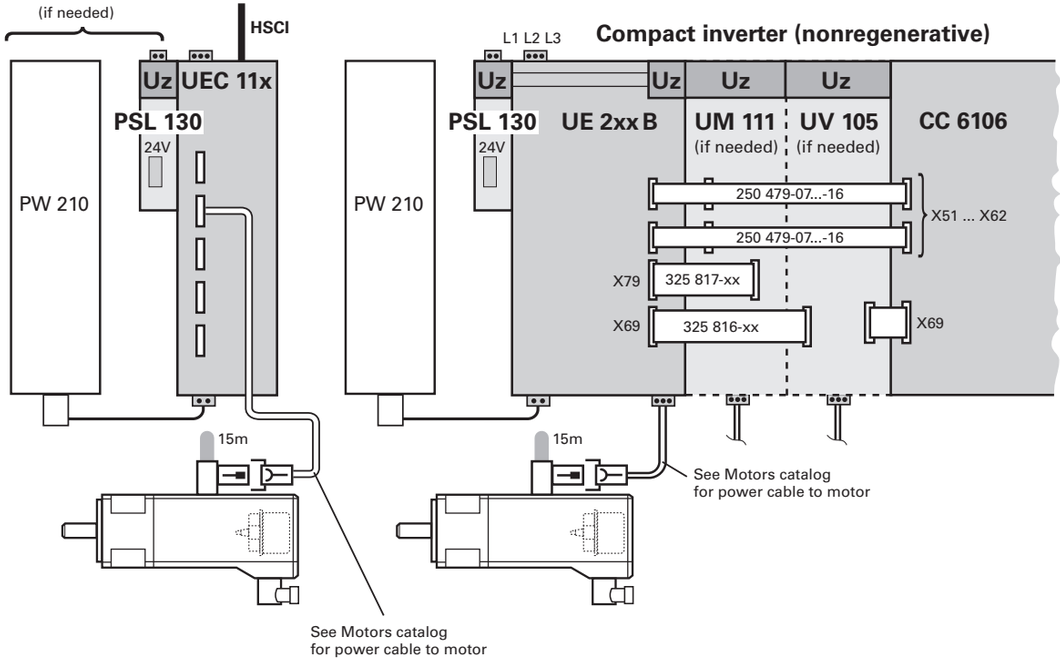
Basic configuration
with UEC 11x

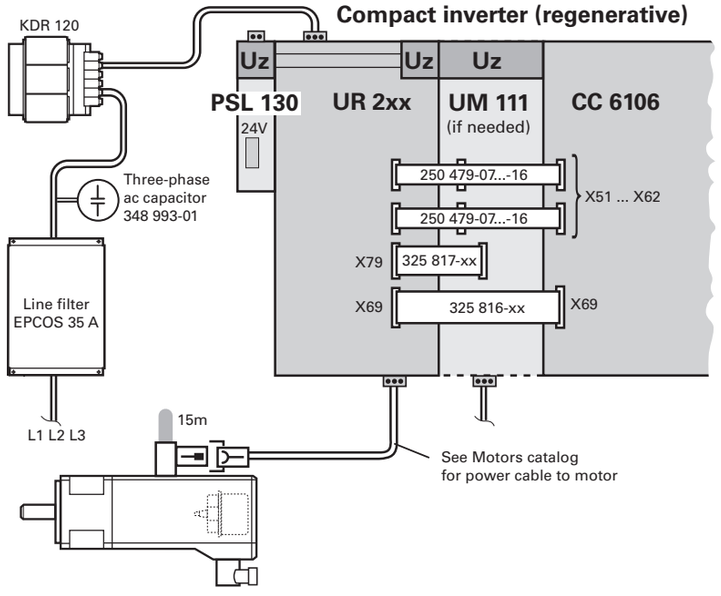


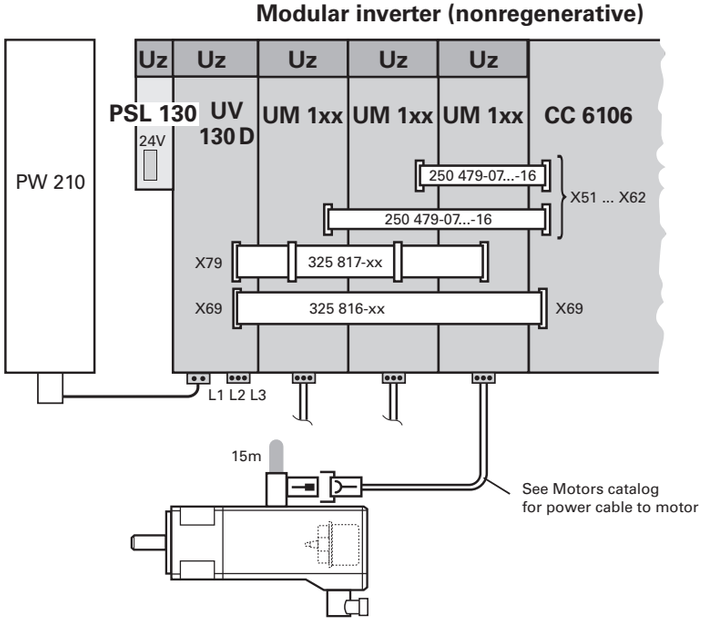
Basic configuration
with CC 6106

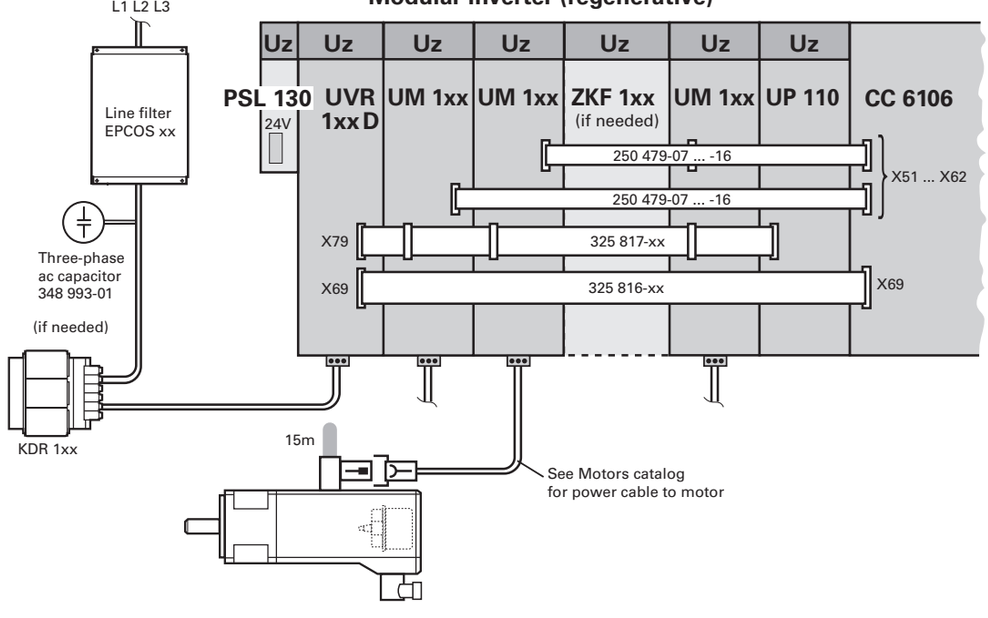


3.35 Cable Overview for HEIDENHAIN Inverter System

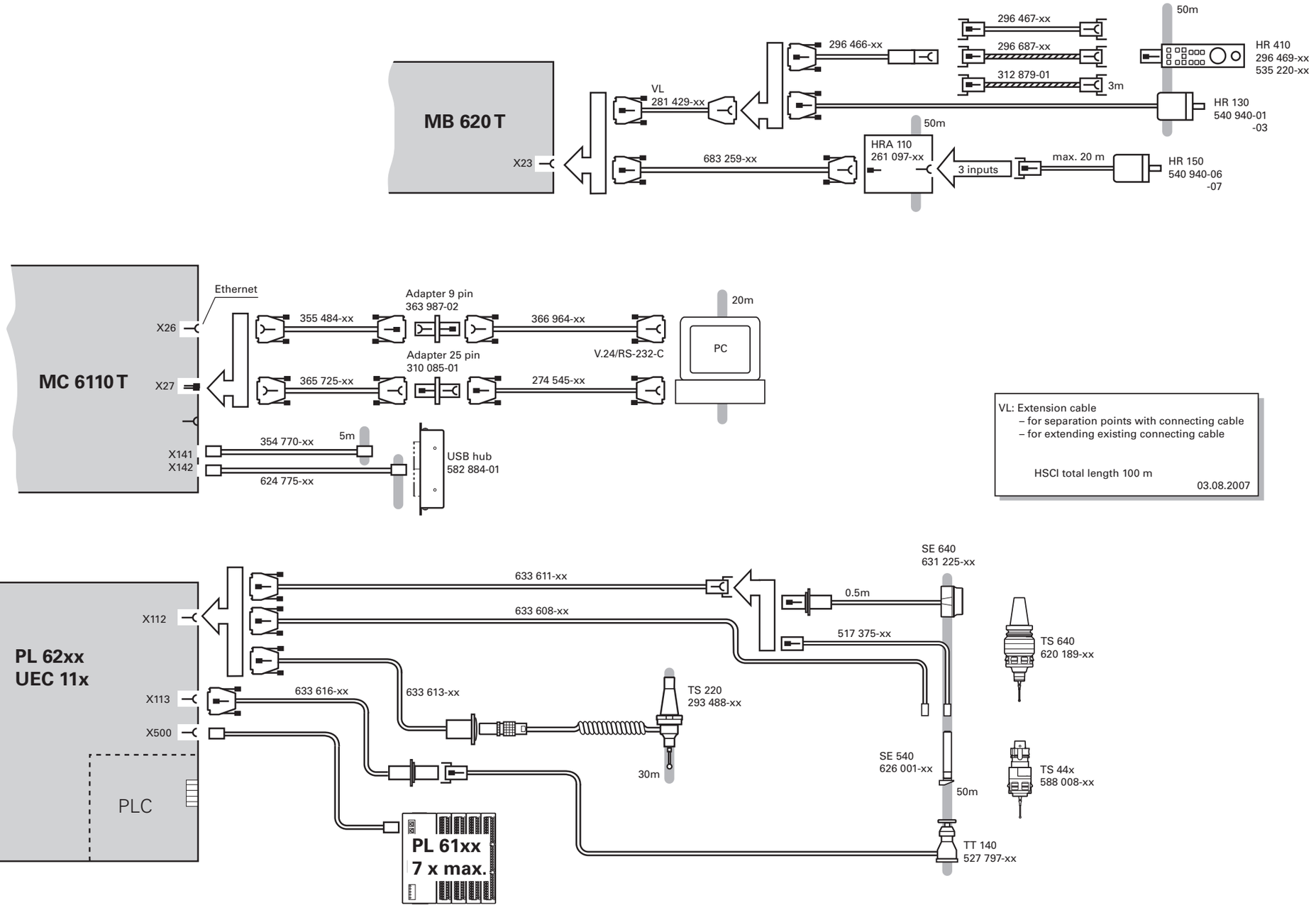








Accessories



4 Machine Parameters

4.1 General Information

A control must have access to specific machine data (e.g. traverse distances, acceleration, speeds) before it can execute its programmed instructions. You define these data in machine parameters. Each machine has its own set of machine parameters.

The parameter values are entered in the **configuration editor.**

The machine parameters are grouped as parameter objects in a tree structure in the configuration editor. As an alternative you can use unique parameter numbers to access the desired machine parameters directly (see "Accessing machine parameters via MP numbers" on page 259).

The machine parameters are saved in **parameter files** with the extension **.cfg** on the **TNC:\, PLC:** and **SYS:** drives.
These drives should always be addressed with the system variables %OEM% (PLC:\ ), %SYS% (SYS:\ ) and %USR% (TNC:\ ).

Note

Only use the configuration editor to make your changes to the machine configuration!
Only in exceptional cases should the *.cfg files be edited directly. This could accidentally lead to faulty syntax, which would prevent the control from starting up.

The parameter objects appear as folders in the configuration editor. Each parameter object (also referred to as entity, data object or object) has a name (beginning with **Cfg...**) that gives information about the machine parameters it contains.

Depending on the function, the parameters are differentiated into system-specific, channel-specific and axis-specific types. Each object has a **key** for unique identification. The key name can have a maximum of 18 characters.

The following applies:

- **System data** (parameters that are valid for the entire system) only occur once. The configuration editor does not require a key name for these parameters, nor is one entered. These objects are identified with an "empty" key in the *.cfg files.
- **Parameter objects that apply to axes** occur more than once.
  A unique key name is assigned to each axis. All objects that apply to a certain axis must be identified with this key. You can choose and specify the individual key names yourself, or you can use the predefined key names supplied by HEIDENHAIN.
  Example:
  - The key name "X" or "X axis" for all objects that belong to the X axis
  - The key name "S" or "S axis" for all objects that belong to the spindle

- **Parameter objects that apply to channels** occur more than once.
  A unique key name is assigned to each channel. All objects that apply to a certain channel must be identified with this key.
  Example:
  - Key name "CH_NC" for all objects that belong to the channel for executing the NC program
  - Key name "CH_SIM" for all objects that belong to the channel for simulating the NC program

Note

Key names should be short and clear, and refer to the function.

4.2 The “Machine Parameter” Mode of Operation

4.2.1 Calling the configuration editor

- Switch to the **Organization** mode of operation.
- Press the  soft key.
- Enter the code number **95148** or press the **CONFIG EDIT** soft key if the code number was already entered
- Press the **END** soft key to exit the **Machine Parameter** mode of operation.

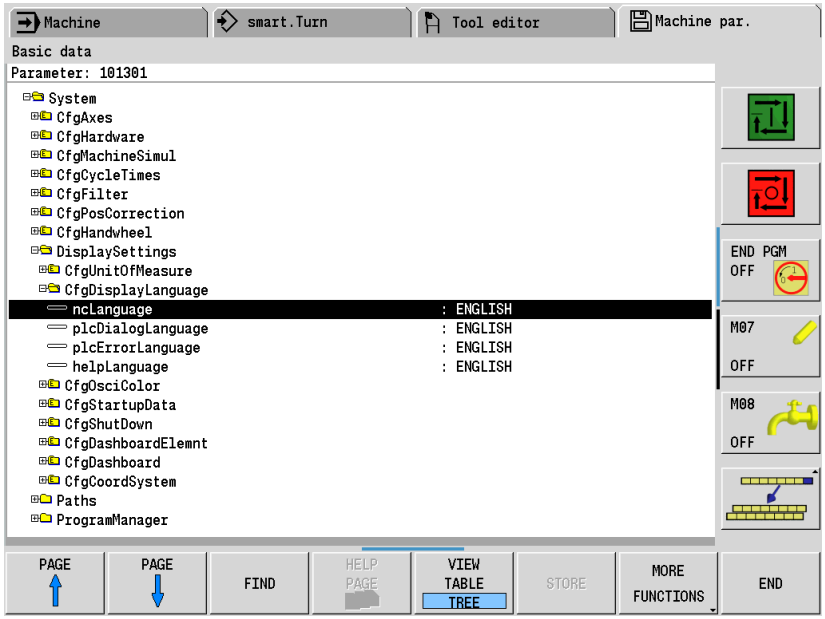
Soft keys

The following functions can be called through the soft-key row of the **Machine Parameter** mode of operation's opening screen:

Soft key	Function
SORT FILE CONTENT	Sorts the contents of the *.cfg files
DATA BACKUP	Backs up the machine parameters
LOAD MP SUBFILE	Activates MP subfiles
SELECT DATA RECORD	Selects various data records
CONFIG FILE LISTS	Path information for the ***.cfg** files of the machine configuration
CONFIG DATA	Opens the configuration editor for editing the machine parameters in tree or table view
MORE FUNCTIONS	Calls additional functions
OFF	Shuts down/restarts the control
REMOVE SYNTAX ERROR	Grayed out during normal operation. Only active after a software update or with faulty configuration data. Used for finding and fixing errors in the machine configuration.
UPDATE RULES	Displays and edits rules for the software exchange
RESET VERSION	Resets the update version
ACCESS SECURITY/ OPTIONS	Protects data from unauthorized access
ATTRIBUTE INFOS	Displays access rights, selection lists, limit values and units of measurement
	Returns to previous menu
END	Exits the **Machine Parameter** mode of operation

4.2.2 Entering and changing machine parameters

After pressing the **CONFIG DATA** soft key, the object tree for the machine parameters is displayed.



The **Parameter:** line above the object tree shows the unique MP number of the currently selected machine parameter or object folder (see "Accessing machine parameters via MP numbers" on page 259). If the help function is open or the table view is active, you can see the current input value next to the MP number, as well as the parameter's unit of measure, if there is one.

The actual machine parameters with their values are located on the lowest level of the tree.

The cursor is positioned within the tree either with the arrow keys or with the optionally connected USB mouse.

To open a branch:

- Press the **+** key or **ENT** key or the right arrow key, or click the folder symbol with the left mouse button

To close a branch:

- Press the **–** key or **ENT** key or the left arrow key, or click the folder symbol with the left mouse button

Double-click the parameter symbol with the left mouse button or press the **ENT** key to open the editing window.

Soft keys and screen buttons can also be clicked with the mouse.

Icons in the object tree

A symbol (icon) is displayed at the beginning of each line in the parameter tree. The icons have the following meanings:

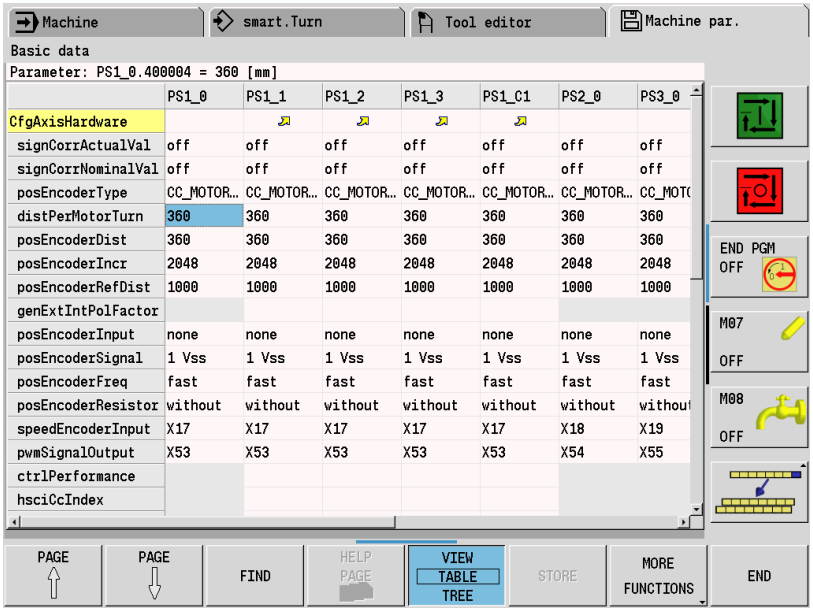
Button	Function
	Branch is closed
	Branch is open
	Empty parameter object, cannot be opened
	Initialized machine parameter
	Uninitialized (optional) machine parameter
	Machine parameter was changed but not saved yet
	Machine parameter or parameter object can be read but not changed
	Machine parameter or parameter object cannot be read
	Machine parameter with comment
	Derived parameter block

The type of the configuration object is identified by its folder symbol:

Button	Function
	Entity (object)
	Array (list)
	Key (key name)

Table view

You can activate a table view in the configuration editor. This is especially useful for the configuration of parameter blocks, since now the parameters of all axes are visible at a glance:



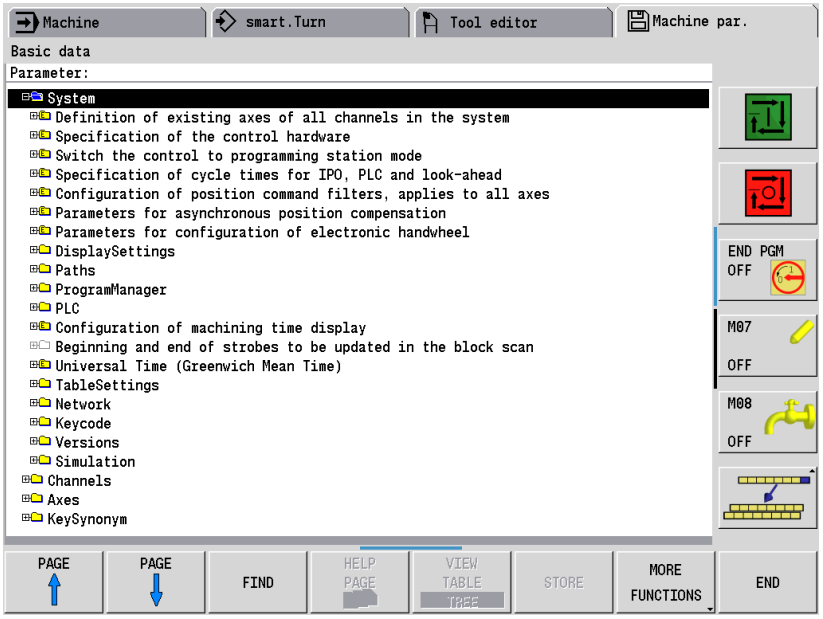
	PS1_0	PS1_1	PS1_2	PS1_3	PS1_C1	PS2_0	PS3_0
CfgAxisHardware							
signCorrActualVal	off	off	off	off	off	off	off
signCorrNominalVal	off	off	off	off	off	off	off
posEncoderType	CC_MOTOR...	CC_MOTOR...	CC_MOTOR...	CC_MOTOR...	CC_MOTOR...	CC_MOTOR...	CC_MOT
distPerMotorTurn	360	360	360	360	360	360	360
posEncoderDist	360	360	360	360	360	360	360
posEncoderIncr	2048	2048	2048	2048	2048	2048	2048
posEncoderRefDist	1000	1000	1000	1000	1000	1000	1000
genExtIntPolFactor							
posEncoderInput	none	none	none	none	none	none	none
posEncoderSignal	1 Vss	1 Vss	1 Vss	1 Vss	1 Vss	1 Vss	1 Vss
posEncoderFreq	fast	fast	fast	fast	fast	fast	fast
posEncoderResistor	without	without	without	without	without	without	without
speedEncoderInput	X17	X17	X17	X17	X17	X18	X19
pwmSignalOutput	X53	X53	X53	X53	X53	X54	X55
ctrlPerformance							
hsciCcIndex							

All editing functions available in the tree view are also available in the table view. Changed table columns are highlighted dark blue. The **Info** key of the operating panel can also be used in the table view to call the help function.

Detailed text view

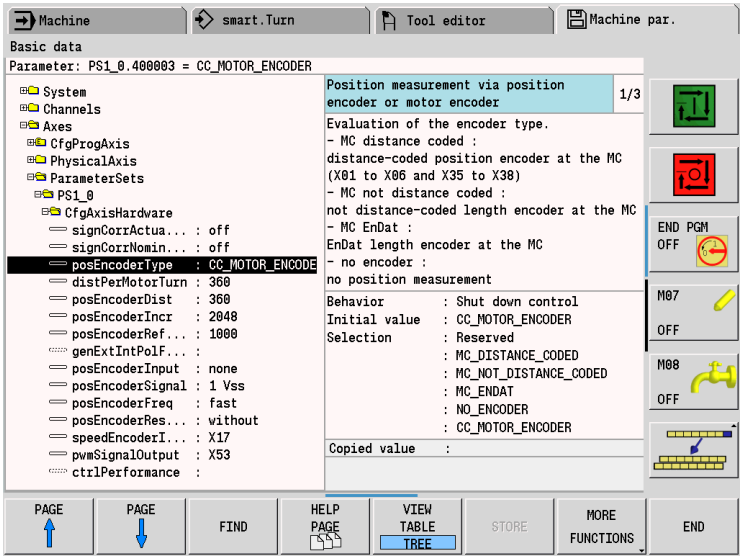
Press the **SHOW SYSTEM NAME** soft key to have the name of the parameter tree be shown in detail, or as the system short-form text:

- Press the “split screen layout” key to the left of the control’s TFT flat panel screen.
- Press the **SHOW SYSTEM NAME** soft key: the control now shows the folder and parameter names in detail.
- Press the **SHOW SYSTEM NAME** soft key again to switch the display back to the system short-form name.



Displaying help texts

The **Info** key enables you to call a help text for each parameter object or attribute.



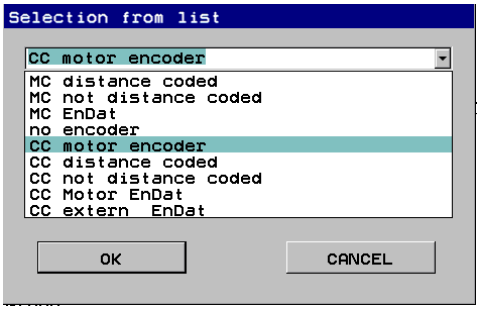
If the help text does not fit on one page (1/2 is then displayed at the upper right, for example), press the **HELP PAGE** soft key to scroll to the second page.

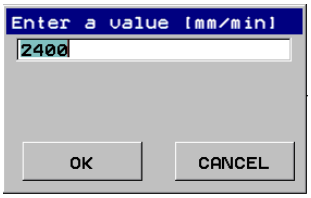
Additional information, such as the unit of measure, the initial value, or a selection list, is also displayed. If the selected machine parameter matches a parameter in the iTNC 530, the corresponding compatible MP number is shown.

To exit the help text, press the **Info** key again.

Entering and changing parameters

In order to change machine parameters, you must open an input or selection field by pressing the right arrow key, the **ENT** key or the **CHANGE VALUE** soft key, or by double-clicking the selected parameter:





Open and close selection lists (pull-down menus) by pressing the GOTO key or clicking the arrow symbol. Use the arrow keys (up and down) or the mouse to navigate through the pull-down menu. Press the **ENT** key to select the desired value from the list.

Units of measure can be defined for numeric machine parameters. The unit of measure assigned to this parameter is displayed. Enter a value appropriate to this unit.

Limit values

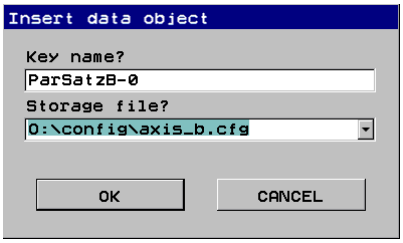
Limit values are displayed for numeric machine parameters. If you attempt to enter a value outside of these limits, a message is issued and the entry not accepted.

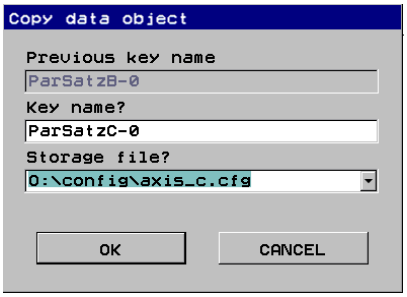
Deleting objects

Press the **DELETE** soft key to delete objects or parameters from a list (an axis, for example).

Inserting and copying objects

Press the **MORE FUNCTIONS** and **INSERT** or **COPY** soft keys to insert or copy objects or items in lists (an axis, for example). Items in lists (arrays) are inserted after the cursor.
When inserting an object, the object name (key name) and memory file must be given. The memory file is the *.cfg file in which the inserted object is to be saved. Press the up or down arrow keys and the **ENT** key to select the file. Open the pull-down menu by pressing the GOTO key.





The icons of empty objects, lists and parameters appear dimmed. They can be activated with the **INSERT** soft key.

Copying and inserting values

When a dialog box for entering a value is open, you can use the **COPY FIELD** soft key to copy the content of the current parameter to the clipboard. The value you have copied can be inserted with the **PASTE COPIED VALUE** soft key at any other location in the configuration if an entry dialog is open.

Changing key names

Press the **CHANGE KEY NAME** soft key to change the key name of an object, for example, from Kinem1 to Kinem_XYZ.

Saving input values

The input values are buffered with the **OK** soft key. The **CANCEL** soft key closes the dialog box without buffering the value. All changes that have been made but not yet saved are marked with a symbol on the left side of the screen.

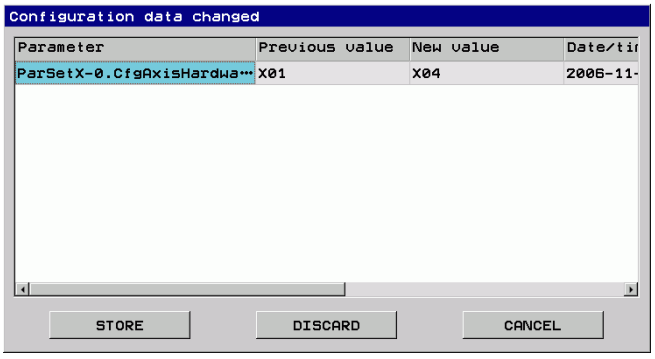
Press the **SAVE** soft key to actually save the changes you have made.

Certain data cannot be stored while an NC program is running. The message **Cannot change parameter during program run** appears. In this case the program must first be stopped and exited. Then the data can be saved.

Some data take effect as soon as they have been saved.
Others require that the axes be referenced again, or that the system be restarted. This is indicated in a corresponding message.

Change list

A machine-parameter change list is displayed after pressing the **SAVE** or **END** soft keys. The **Configuration data changed** window gives you an overview of all changed parameters. You can save, discard or cancel the changes:



The control also saves a list of the last 20 changes to the configuration data. In this list you can see all changes performed, and can undo any of them. The change list is maintained upon power-off of the control. The change list is reached in the configuration editor via the **MORE FUNCTIONS** and **DISCARD CHANGES** soft keys.

Settings in the configuration editor	MP number
System	
CfgConfigSettings	
undoListSize	106501
dispParamNumbers	106503

With the **MP_undoListSize** parameter you can specify the number of entries in the change list:

MP_undoListSize

Defines the number of entries in the change list
Available from NCK software version: 597 110-04.

Format: Numerical value
Default: 20
Access: LEVEL3
Reaction: NOTHING

With the **MP_dispParamNumber** parameter you specify whether the number of machine parameters or the symbolic name is displayed in the change list.

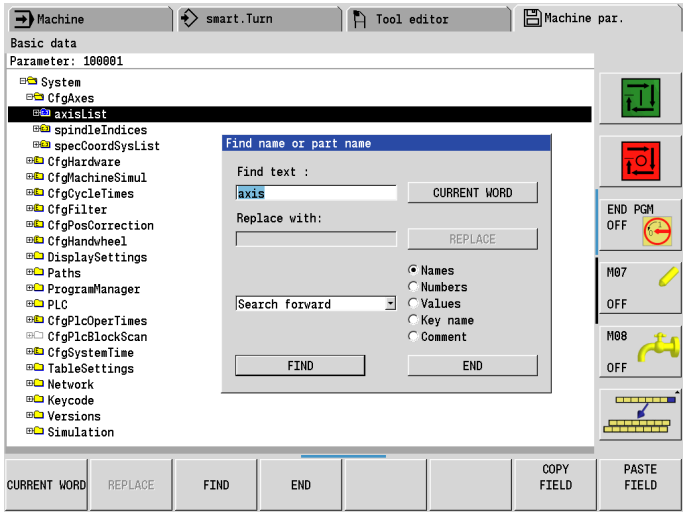
MP_dispParamNumbers

Display the symbolic names or the number of machine parameters in the change list
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: TRUE
Number of the machine parameter is displayed
FALSE
Symbolic name of the machine parameter is displayed.
Default: No value, parameter optional (= FALSE)
Access: LEVEL3
Reaction: NOTHING

Finding / Replacing

You can search for and replace objects and parameters within the configuration editor using a dialog box.
Open the dialog box with the **FIND** soft key.



Use the selection dialog on the right side of the window to specify the type of search. The following possibilities for searching are available:

Selection	Function
Names	Search for object and parameter names. You can also enter just a part of the text to be searched for as search criteria in the **Find text:** field. The **CURRENT WORD** button and soft key load the term marked with the cursor into the search field. The search term can be written in large or small letters.
Numbers	Search for an MP number. You can also enter just part of the number to be found.
MP number	Search for the compatible iTNC MP number. In many parameters the parameter number of the iTNC 530 contouring control is displayed in the help text. After this function is selected and an iTNC parameter number is entered, the configuration is searched for equivalents.
Values	Search for certain parameter values. As soon as you have selected the **Values** function, the **Replace with:** input field becomes active. By entering a value in the "Replace with" field, you can change the string found by pressing the **REPLACE** soft key. This replacement only affects the current search hit.
Key name	Search for a certain key name. As soon as you have selected the **Key name** function, the **Replace with:** input field becomes active. By entering a value in the "Replace with" field, you can change the string found by pressing the **REPLACE** soft key. This replacement only affects the current search hit.
Comment	Search for certain user comments; (see "User comment" on page 258.)

User comment

You can comment objects. A dialog box for entering a comment is opened after pressing the **MORE FUNCTIONS** and **COMMENT** soft keys. A maximum of four comment lines can be entered.

Objects with comments are displayed on the right side of the parameter tree and are identified with the letter "i." The complete text, including the help text for the object, appears after you press the **COMMENT** soft key.

The current value of a parameter can be buffered together with the comment and can, for example, be reactivated later.

Finish editing

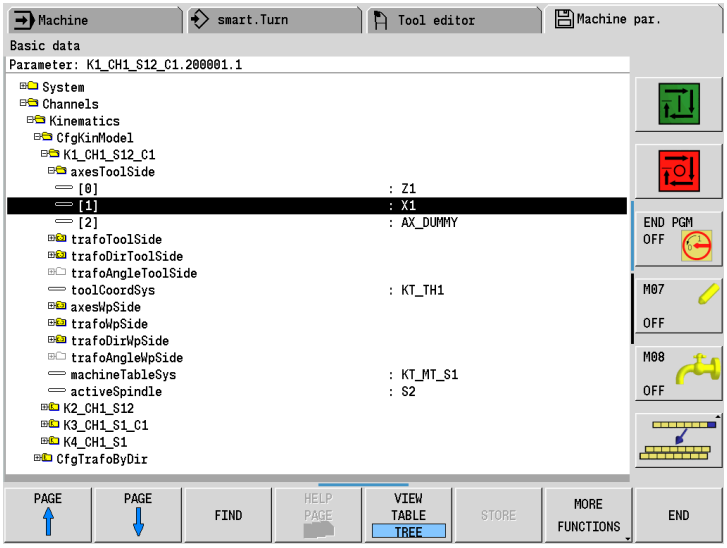
Press the **END** soft key to return to the main menu of the **Machine Parameter** mode of operation.

If any changes have been made, the control displays the **Configuration data changed** window (see "Saving input values" on page 256).

4.2.3 Accessing machine parameters via MP numbers

In order to reach a certain point in the machine configuration as quickly and directly as possible, a unique MP number has been assigned to each machine parameter.

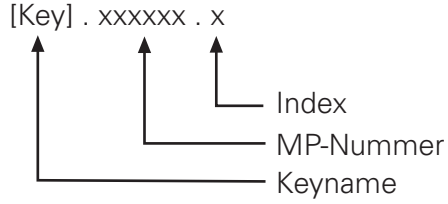
The MP number is shown above the object tree or the table view in the **Parameter:** line. The current input value, as well as the parameter's unit of measure, if there is one, are shown after the MP number:



The MP number consists of a 6-digit number. The key name precedes the 6-digit MP number, and the index number (e.g. for list parameters) is appended to the MP number. The three parts are each separated by a period.

The MP numbers are permanently assigned to a specific configuration object or machine parameter, and can occur more than once in the system, for example in the axis-parameter blocks or in the kinematics configuration. The prefixed key names then serve to distinguish between the numbers.

Structure of an MP number:



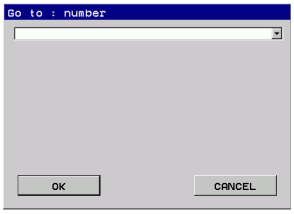
Parameter groups

Similar to the parameter tree-structure of the control, the MP numbers are collected in groups:

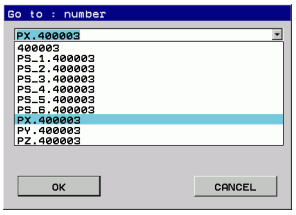
Machine parameters	Group
100000 to 199999	System configuration + miscellaneous
200000 to 299999	Channel-dependent settings
300000 to 399999	Axis configuration
400000 to 499999	Parameter blocks
500000 to 599999	Reserved
600000 to 699999	Range of parameters for CNC lathe machining
700000 to 799999	Reserved
800000 to 899999	Reserved
900000 to 999999	Parameter range for the OEM

GOTO key

Press the GOTO key on the control's operating panel to jump directly to any MP number. If the GOTO key is pressed while in the configuration parameter, the **Go to** dialog box appears:



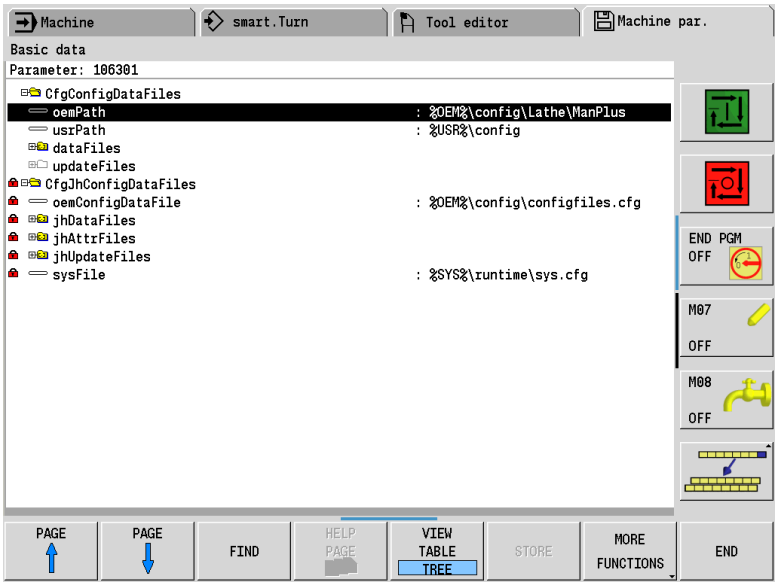
If an MP number exists more than once in the configuration (e.g. axis parameters), the control displays the available parameters as soon as you have entered the entire number. You can then select the desired parameter and jump to it by pressing the **ENT** key.



4.2.4 Managing configuration files

The configuration data is saved in several files with the extension **.cfg.** This enables different types of machines to establish the correct configuration by selecting the appropriate files from the paths entered.

There are two types of configuration file lists: HEIDENHAIN files and OEM files.
The HEIDENHAIN files are permanently defined and cannot be changed (e.g. **CfgJhConfigDataFiles**).



The paths and names of OEM files can be changed with the **CONFIG FILE LISTS** soft key. New configuration files can also be added (for a new axis, for example). The paths are saved in the **configfiles.cfg** file.

The paths and names of the configuration files are stored in the **dataFiles** list (see "Allocation of Configuration Data" on page 291). The control searches for the parameter objects and their parameters in these *.cfg files.

The paths of these files can be changed. Use the right arrow key to open an input dialog for entering the new path or file name.

4.2.5 Sort file content

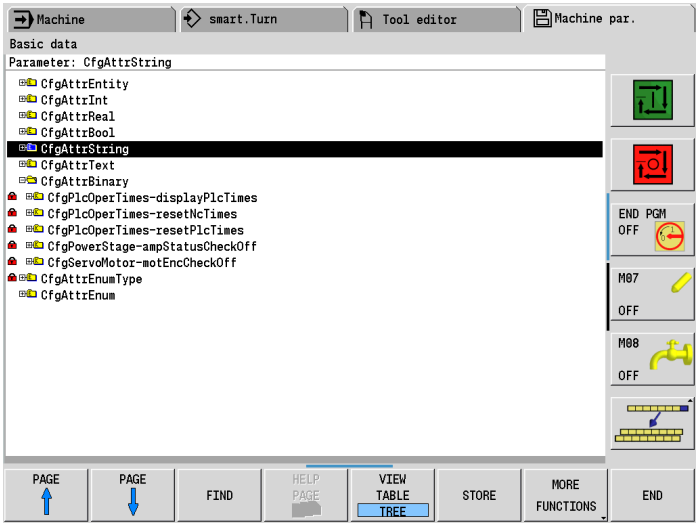
Pressing the **SORT FILE CONTENT** soft key in the main menu of the **Machine Parameter** mode of operation sorts the contents of the *.cfg configuration files so that the objects are in the same order as they are listed in the configuration editor.

However, since the data in the configuration editor comes from multiple files, there is no direct correlation between the display in the configuration editor and the contents of each file.
This sorting makes it easier to compare the contents of similar files (such as *.cfg for axes), since the entries are now in the same order.

4.2.6 Attribute information

Use the **ATTRIBUTE INFO** soft key to display write-accesses, reactions, selection lists, limit values, units of measure, and format instructions.

These data are permanently defined by HEIDENHAIN. The OEM cannot change them.



Access rights

Entering a code number also grants access rights to the machine parameters. There is a difference between the four levels, from Level1 to Level4. Level1 grants few rights, whereas Level4 grants all rights.

LEVEL1 access rights
Machine parameters on LEVEL1 can be reached and changed without the need to enter a code number.

LEVEL2 access rights
Machine parameters on LEVEL2 can be reached and changed with **code number 123**. Enter the code number in the **Organization** mode using the soft key.

LEVEL3 access rights
Machine parameters on LEVEL3 can be reached and changed with **code number 95148.**
The end user must not be told of the code number 95148.

LEVEL4 access rights
Machine parameters on LEVEL4 can only be accessed by **HEIDENHAIN.** The machine manufacturer can only read them.

Reaction to change

The following reactions can occur when machine parameters are changed:

- NOTHING
- RUN
- RESET
- REF

The detailed description for each parameter in this manual includes the reactions that occur for each machine parameter.

Reaction NOTHING
Data with this reaction can be changed at any time, including during program run.

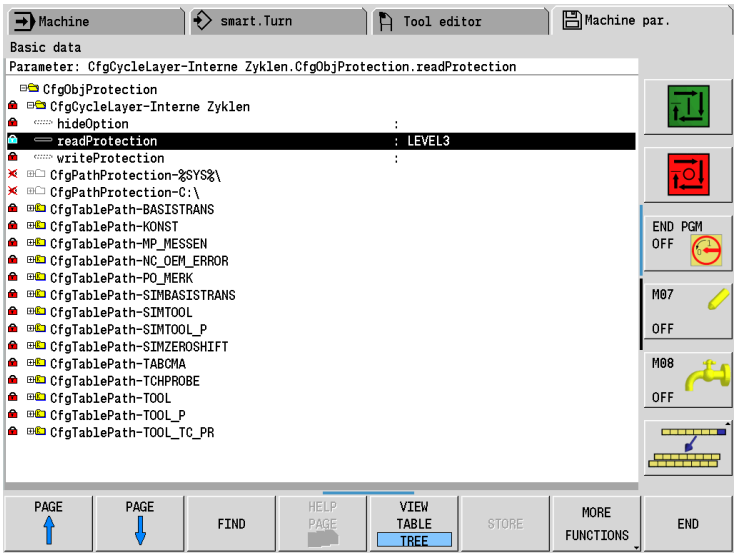
Reaction RUN
Changes are only possible during a PLC strobe or NC stop.

Reaction RESET
After a machine parameter to which the RESET reaction is assigned has been changed, the error message **`Machine parameters were changed. Shut down and restart the control`** is displayed.
This message cannot be cleared. The machine must be restarted. If you want to make more changes in the configuration editor, you can also perform the restart later.
Data objects with this reaction must not be changed during program run.

Reaction REF
After a machine parameter connected to the REF reaction has been changed, the affected axis is set to unreferenced.
A new reference run must be made for this axis.
Data objects with this reaction must not be changed during program run.

4.2.7 Access protection / options

Press the **ACCESS PROTECTION / OPTIONS** soft key to get to the configuration options for protecting data from unauthorized access.



Display of write-protected parameters

Settings in the configuration editor	MP number
System CfgConfigSettings **hideWriteProtected**	106504

With **MP_hideWriteProtected** you can specify whether write-protected parameters are to be displayed or hidden in the configuration editor. This applies especially to all data saved on the SYS partition.

MP_hideWriteProtected

Hiding write-protected parameters

Format: Selection menu

Selection: **TRUE**
Hide write-protected config objects
FALSE
Display all write-protected config objects

Default: FALSE

Access: LEVEL3

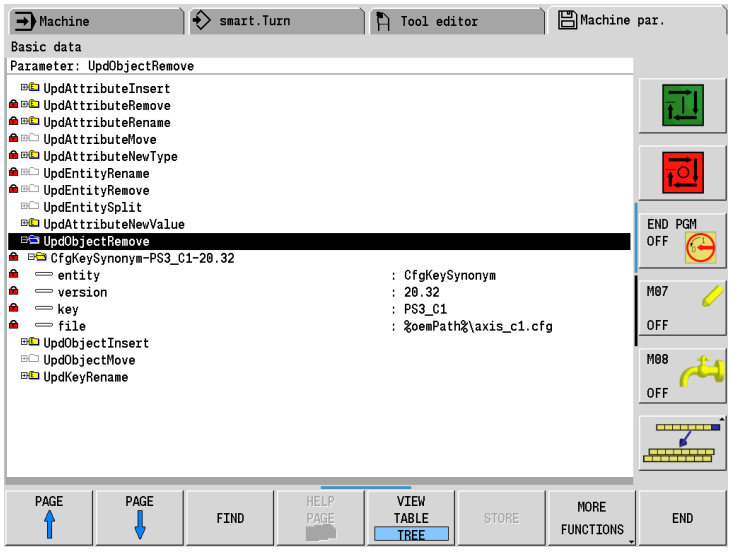
Reaction: NOTHING

4.2.8 Update rules

Introduction

You can use update rules to insert, delete, rename and move OEM machine parameters. You can also change parameter types and the sizes of list fields.

The update rules define rules that are required for automatically updating the machine parameters during a software update. To get to the update rules, press the **MORE FUNCTIONS** soft key in the main menu of the **Machine Parameter** mode of operation, and then the **UPDATE RULES** soft key.



If you want to transfer a new PLC program to the control, for example as part of an NC-software update via USB stick, then often it is also necessary to change or expand the OEM machine parameters. You can use the update rules to have these changes performed automatically.

These update rules are also a tool for keeping track of the versions. You can assign version numbers when changing machine parameters. During a software update, the control automatically detects an increase in the version number, and the update rules are applied.

The update rules can only be used to change or create new OEM machine parameters. You can also change input values for parameters accessible via the MP code number **95148.**

During the first restart of the control after the update, the service technician who updated the NC software must check and confirm the changes made by the update rules to the configuration.

If the user attempts to leave the configuration editor without saving the changes, a dialog window appears prompting the user to save them. The configuration editor cannot be exited until the data are saved.

Note

The first restart of the control after the update cannot be continued without saving the configuration changes made by the update rules.

If configuration objects are to be added or deleted during a software update, then you must define a command for this action in the **update rules.** You can call the **update rules** function to view all previous changes to the configuration made with update rules.

These rules no longer need to be followed once the software has been exchanged (see "NC software exchange on the MANUALplus 620" on page 94).

Using update rules

You must save update rules in a file with the extension **.cfg.**

- Ensure that a file with the name **PLC:\config\[update rule].cfg** exists on the control.
- Enter the Machine Parameter mode of operation, and use the **CONFIG FILE LISTS** soft key to enter the path to the [update rule].cfg file as described below:

Settings under CONFIG FILE LISTS	
CfgConfigDataFiles updateFiles **[0]**:	%oemPath%\[update rule].cfg

- Edit the **[update rule].cfg** file with an ASCII editor, such as Notepad or UltraEdit32. The following pages describe the available commands.

The following software version-dependent files for the machine manufacturers' update rules are already available in the directory PLC:\config\lathe\Manplus:

- UpdateOemRel002.cfg Update rules for software version -02
- UpdateOemRel003.cfg Update rules for software version -03

The machine manufacturer is responsible for the update rules specified in these files. When the control is shipped, the files are empty.

The current OEM update version can be reset with the **RESET VERSION** soft key in the configuration editor, in the same way as the HEIDENHAIN update rules.

Always save an **[update rule].cfg** file in the final configuration of your machine, and include this file with **CONFIG FILE LISTS** as described above. This way you can have this file be overwritten with a new version at any time as part of a software update, and have the update rules be followed.

UpdObjectInsert

The indicated configuration object is inserted. The procedure is performed if...

- the control determines during startup that the indicated object does not exist
- the current OEM version number of the configuration is less than the version number indicated in the update rule.

There are various possibilities for inserting parameters into the new configuration object:

- The name of the new configuration object is entered directly as a string. Only parameters that are to be set to specific input values must be entered.

Syntax:

UpdObjectInsert (

entity	:= STRING,	; Name of the configuration object
version	:= REAL,	; Version number
key	:= STRING,	; Key name of the configuration object
file	:= STRING,	; Path/file name of where the new object is to be saved
object	:= STRING (OPTIONAL)	; Name of the new configuration object (identifier)

)

Example:

```
UpdObjectInsert  (
      entity:="CfgOemBool",
      version:=1.00,
      key:="MG_OemBool",
      file:="%OEM%\\config\\plc_oem.cfg",
      object:="CfgOemBool (value:= [FALSE, TRUE])"
)
```

If the version code of the OEM configuration is < 1.00, this update rule creates the symbolic operand MG_OemBool in the configuration object CfgOemBool. Possible values are TRUE and FALSE.

UpdObjectRemove

The indicated configuration object is removed from the current machine configuration. If the name of a *.cfg file is entered, the configuration object is removed only from the entered file. This procedure is performed during startup of the control if the version number of the OEM configuration on the control is older than the version number of the update rule.

The object to be removed is marked in the configuration editor. The user must confirm the final removal by pressing the **SAVE** soft key in the configuration editor.

Syntax:

```
UpdObjectRemove (
        entity      := STRING,      ; Name of the configuration object
        version     := REAL,        ; Version number
        key         := STRING,      ; Key name of the configuration
                                      object
        file        := STRING       ; Configuration object is removed
                       (OPTIONAL)     only from the entered file
)
```

Example:

```
UpdObjectRemove  (
      entity:="CfgOemBool",
      version:=1.01,
      key:="MG_OemBool",
      file:="%OEM%\\config\\plc_oem.cfg
)
```

The indicated configuration object is removed from the plc_oem.cfg file. Other *.cfg files of the machine configuration in which the object also exists are not affected.

UpdObjectMove

The indicated configuration object is moved to another *.cfg file. If the indicated configuration object already exists in the file entered (*.cfg), no action is performed.

This procedure is performed during startup of the control if the version number of the OEM configuration on the control is older than the version number of the update rule.

Syntax:

```
UpdObjectMove (
        entity      := STRING,      ; Name of the configuration object
        version     := REAL,        ; Version number
        key         := STRING,      ; Key name of the configuration
                                      object
        file        := STRING       ; Move configuration object to
                                      indicated file
)
```

Example:

```
UpdObjectMove  (
      entity:="CfgOemBool",
      version:=1.01,
      key:="MG_OemBool",
      file:="%OEM%\\config\\plc.cfg"
)
```

The indicated configuration object is moved from the current file to the file plc.cfg.

UpdKeyRename

The key name of the indicated configuration object is changed. The file is not changed.

This procedure is performed during startup of the control if the version number of the OEM configuration on the control is older than the version number of the update rule.

Syntax:

```
UpdKeyRename (
        entity      := STRING,      ; Name of the configuration object
        version     := REAL,        ; Version number
        key         := STRING,      ; Key name of the configuration
                                      object
        keyNew      := STRING       ; New key name
)
```

Example:

```
UpdKeyRename  (
      entity:="CfgOemBool",
      version:=1.01,
      key:="MG_OemBool",
      keyNew:="MG_Boolean"
)
```

The symbolic marker MG_OemBool is renamed as MG_Boolean.

UpdAttributeInsert The indicated machine parameter is inserted in the machine configuration. The procedure is performed if...

- the control determines that the parameter is currently hidden or does not exist
- the current OEM version number of the configuration is less than the version number indicated in the update rule.

If you would like to insert a value for an optional machine parameter, you must set **insertOptional:=TRUE.**

If you do not enter a value, the control automatically uses the default parameter value when following the update rule.

Syntax:

UpdAttributeInsert (

entity	:= STRING,	; Name of the configuration object
version	:= REAL,	; Version number
key	:= STRING, (OPTIONAL)	; Key name of the configuration object
attrName	:= STRING,	; Name of the machine parameter to be inserted
index	:= UNSIGNED, (OPTIONAL)	; Index for list parameters (arrays). If the parameter is a list, then it is inserted at the given location
insertOptional	:= BOOLEAN, (OPTIONAL)	; Insert optional parameter
value	:= STRING (OPTIONAL)	; Parameter value. Default value is used if nothing is entered.

)

Example:

```
UpdAttributeInsert  (
      entity:="CfgOemBool",
      version:=1.02,
      key:="MG_OemBool",
      attrName:="value"
      index:=2,
      insertOptional:=TRUE,
      value:="TRUE"
)
```

A new index with the value TRUE is inserted for the marker MG_OemBool.

UpdAttribute Remove

The indicated machine parameter is removed from all configuration objects of the machine configuration.

This update rule is needed in case a machine parameter is not supported by a newer version of the NC software. This way you can automatically correct your configurations in the field.

Syntax:

```
UpdAttributeRemove (
        entity          := STRING,      ; Name of the configuration
                                           object
        version         := REAL,        ; Version number
        attrName        := STRING,      ; Name of the machine
                                           parameter to be removed
)
```

Example:

```
UpdAttributeRemove  (
      entity:="CfgSimulation",
      version:=1.02,
      attrName:="cc424"
)
```

The **MP_cc424** parameter is removed from all configuration objects with the name CfgSimulation.

UpdAttributeNew Value

The input value of the indicated machine parameter is checked. If the input value corresponds to the value entered in the update rule or is within the given value range...

- either the input value is changed or
- the input value is multiplied with the given factor and then changed. The conversion is only effective for numeric parameter values.

If no value range is defined (minValue and maxValue are not set), the new parameter value is always inserted.

If a single value is entered (minValue or maxValue is set), the new value is inserted if the old value exactly equals the given value.

This procedure is performed during startup of the control if the version number of the OEM configuration on the control is older than the version number of the update rule.

Syntax:

```
UpdAttributeNewValue (
    entity      := STRING,              ; Name of the configuration object
    version     := REAL,                ; Version number
    key         := STRING,              ; Key name, if the value is to be changed in only one specific configuration object. Otherwise do not enter anything.
    attrName    := STRING,              ; Name of the machine parameter
    index       := UNSIGNED, (OPTIONAL) ; Index for list parameters (arrays). Is only checked if the parameter is an index
    minValue    := STRING, (OPTIONAL)   ; Minimum value
    maxValue    := STRING, (OPTIONAL)   ; Maximum value
    value       := STRING (OPTIONAL)    ; New input value
    factor      := REAL (OPTIONAL)      ; Multiplication factor
)
```

Example:

```
UpdAttributeNewValue  (
      entity:="CfgOemBool",
      version:=1.01,
      key:="MG_OemBool",
      attrName:="value",
      value:="TRUE"
)
```

The MG_OemBool marker is changed to the input value TRUE in the configuration object CfgOemBool.

4.2.9 Remove syntax error

The **REMOVE SYNTAX ERROR** soft key becomes selectable when the configuration data in the *.cfg files is being changed manually, or when faulty or incomplete update rules are being used during a software update.
Pressing it opens the faulty file as well as a text editor so that the file can be corrected manually.

Since this soft key can only be selected in one of the above cases, and these cases do not occur during normal operation, the soft key cannot be selected during normal operation.

The start-up of the control is interrupted if a faulty file is detected. The window for entering a code number appears. You must enter the OEM or HEIDENHAIN code number for the configuration editor. The main menu of the configuration editor appears. The **REMOVE SYNTAX ERROR** and **END** soft keys can be selected. Pressing the **END** soft key continues start-up. However, this will lead to many error messages, since only faulty or no configuration data is available.

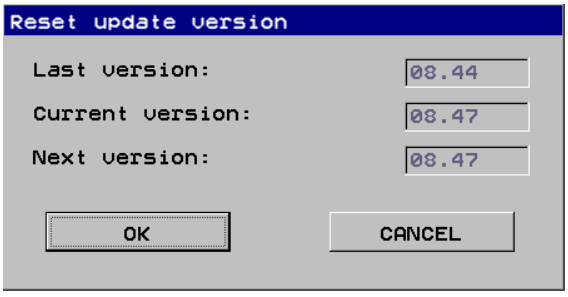
The **END** soft key saves and reloads the file. This can take a moment. If there are still errors, the soft key remains active.

Otherwise the **CONFIG DATA** soft key becomes selectable. You can use it for further corrections in the configuration editor. If the data is now correct, the **END** soft key in the main menu of the configuration editor will continue start-up.

If any other errors are reported, they must be fixed with the configuration editor.

4.2.10 Resets the update version

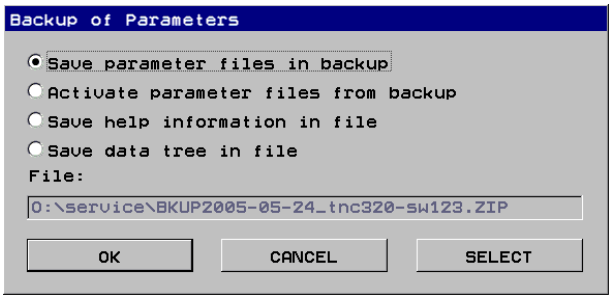
The **RESET VERSION** soft key enables you to return to the previous software version of machine parameters (configuration data).



If, after a software update, configuration errors occur while the control is starting up, the previous executable version can be reactivated. Then find and correct the error by using the update rules.

4.2.11 Backup of parameters

The **DATA BACKUP** soft key enables you to save and restore configuration data as well as to create text files with the current machine parameters:



The following functions are available:

- **Save parameter files in backup**
  The following file name is suggested:
  %OEM%\service\BKUPyear-month-day_.ZIP
  Append meaningful information to this name, for example, the control model, software version, etc.
  All active *.cfg configuration files from %OEM% and %USR% are saved in the selected backup file, e.g. BKUP2005-04-05_tnc320-sw123.ZIP; see %OEM%\config\Configfiles.cfg:
  - All files from %OEM% in the "_Oem_Config_Files_.zip" file
  - All files from %USR% in the "_Usr_Config_Files_.zip" file
  The update files listed in Configfiles.cfg under updateFiles:= are not saved in the backup file.
- **Activate parameter files from backup**
  The *.cfg configuration files from %OEM% and %USR% are retrieved from the selected backup file and activated.
- **Save help information in file**
  The following file name (to be amended) is suggested:
  %OEM%\service\HELPyear-month-day_.TXT
  The created text file with the selected name contains the help information about all parameter objects and attributes. If a parameter exists more than once, for example for several axes, only the information about the first parameter is saved.
- **Save data tree in file**
  The following file name (to be amended) is suggested:
  %OEM%\service\TREEyear-month-day_.TXT
  The created text file with the selected name contains the current values of all parameter objects and attributes.

Note

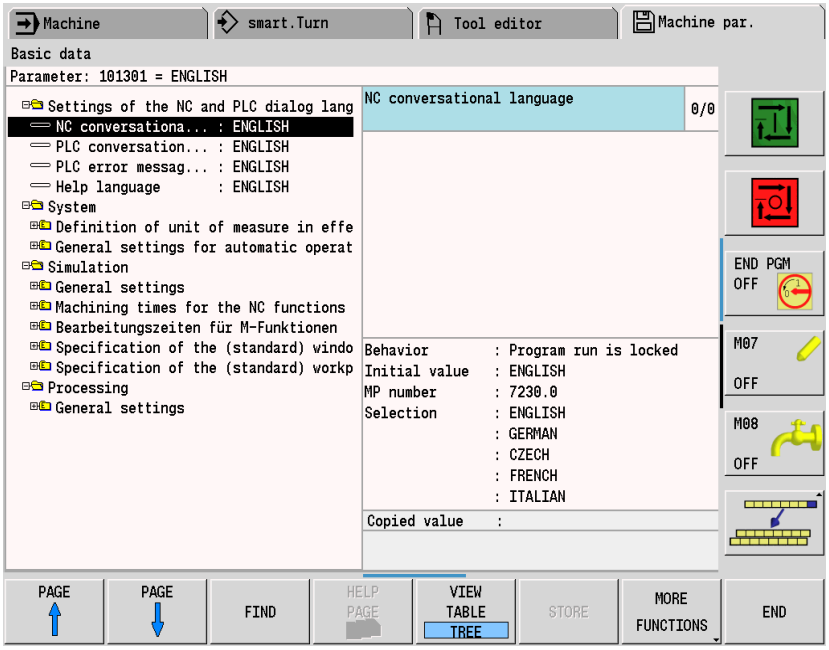
You can also use the PC software TNCbackup for backing up your data in an easy and convenient way. TNCbackup is part of TNCremoNT and is available free of charge from HEIDENHAIN, for example from the file base on the Internet (filebase.heidenhain.de).

Note

A backup should be performed after commissioning and every time the machine parameters have been edited.

4.3 User Parameters

You can provide the machine tool operator with easy access to parameters known as user parameters. These parameters can, for example, refer to language settings or data evaluated by the PLC. You can display a freely definable help text for every user parameter on the right half of the screen. Press the HELP key to show the help text. You define the help text in a language-sensitive *.CSV file.



Freely definable code numbers

Press the **USER PARAMETER** soft key in the **Organization** operating mode to view the predefined user parameters. Enter the following code numbers (default settings) for advanced views of the user parameters:

- Basic view via the **USER PARAMETER** soft key
- Advanced view of **USER PARAMETERS** with possibility of setting the conversational language: Enter code number 123
- **PLC PARAMETER** view with grouped parameters of the PLC basic program: Enter code number 5555

You can assign separate soft keys to the parameters. The control displays these soft keys on the third soft-key row level.

You can configure the code numbers for the user parameters any way you wish (see "Definition of code numbers" on page 283).

Layout file in XML format

You define the layout of the user parameters in layout files. These layout files are in XML file format. The layout of the user parameters consists of a general part prescribed by HEIDENHAIN (userparam.xml, user123.xml) and an OEM-specific part (userparamOem.xml, user123Oem.xml). The OEM-specific files are empty when shipped, and can be used by the OEM to display further parameters.

Extensible Markup Language (XML)) is a standard language for data exchange. It uses tags, allowing for exact description of the data and the structure. You will find a list of all XML commands supported by the control in "XML commands for creating the layout files" on page 286.

On page 284 a simple example illustrates how you create a layout file for your user parameters and separate help texts for the parameters.

4.3.1 Configuration of the user parameters

Settings in the configuration editor	MP number
System	
Key code	
CfgOemPassword	106900
[Key name]	
funcList	106901
CfgModOemSoftkey	107100
[Key name of the parameter view]	
activation	107101
skPos	107102
buttonText	107103
dialogRes	
text	
buttonImage	107104
funcKey	107105
helpId	107106
CfgCfgEditActivate	107200
[Key name of the parameter view]	
layoutFile	107201
dispLangText	107202
System	
Paths	
CfgOemPath	102000
dialogTextfile	102002

- **CfgOemPassword**
  [Key name]
  Define a key name for your own parameter view under **MP_funcList**, for example: **`MP_funcList[0]: CFGEDIT-OEM`**.

Note

If you also want to grant your machine operators access to LEVEL3 (see "Access rights" on page 262) machine parameters, you have to enter to following value additionally under **MP_funcList**
`MP_funcList[1]: CONFIG-LEVEL3`

- **CfgModOemSoftkey**:
  After entering an OEM code number, you can display a soft key to call the user parameters. For configuration, enter the same key name as in CfgOemPassword/funcList, e.g. CONFIGEDIT-OEM. You can define an image, a language-neutral or a language-sensitive text for the soft key. The individual parameters of CfgModOemSoftkey are described more in detail in the following.
- **CfgCfgEditActivate**
  Defines the XML layout file for the depiction of the user parameters. You have to enter the same name as in CfgOemPassword/funcList as key name. The views for the code number "5555" (CONFIGEDIT-PLC_Parameter), code number "123" (CONFIGEDIT-USER123) and the view without any code number (CONFIGEDIT-USERPARAM) are already defined.
  - **MP_layoutFile: path to the XML layout file**:
    You define the layout for the tree structure of the user parameters in the XML file. You define the path to the layout file in **MP_layoutFile**. HEIDENHAIN recommends saving the XML layout files under **%OEM%:\config\layout**.
    Three views are already defined by HEIDENHAIN. (CONFIGEDIT-PLC_Parameter, CFGEDIT-USER123 and CFGEDIT-USERPARAM). The configuration possibilities for parameters relevant to the PLC basic program (code number 5555) are located under CONFIGEDIT-PLC_Parameter. The settings in CONFIGEDIT-USER123 apply to the user parameters, which are displayed after you have entered the code number 123.
    CONFIGEDIT-USERPARAM refers to the user parameters you call by pressing the **USER PARAM** soft key. You can adapt these views to your requirements or remove them from the configuration. The associated XML layout files are available at:
    %SYS%\config\layout\userparam.xml and
    %SYS%\config\layout\user123.xml
    %OEM%\config\layout\PLC_Parameter.xml
  - **MP_dispLangText – show language-sensitive parameter name**
    Set the machine parameter **MP_dispLangText** to TRUE if you want to display by default your own parameter names from the *.CSV file (e.g. "conversational language") for the user parameters instead of the system names (e.g. CfgDisplayLanguage).
- **MP_dialogTextfile – language-sensitive help texts and parameter names:**
  You can configure the parameter names and the parameter help texts (to be shown in the help window) in multiple languages. In **MP_dialogTextfile,** enter the file name of the *.CSV file containing the parameter texts. The path is permanently defined: %OEM%\plc\language\en (or another language abbreviation). Therefore, enter only the file name without path. The language abbreviation defined in MP_System/DisplaySettings/CfgDisplayLanguage/ncLanguage is used. If the file is not available in the selected language, the control attempts to open the English file (directory: **%OEM%\plc\language\en**).

Machine parameters

Machine parameters in the **CfgOemPassword** config object:

MP_funcList

List of function names (= key names) that are called by entering the password:
Available from NCK software version: 597 110-01.

Format: Array 0...200

Input: Key name with max. 18 characters
Name of the functions that are called by entering the password.
Enter these names as key names in CfgModOemSoftkey and CfgCfgEditActivate.
The key name of CfgOemPassword defines the password.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

Machine parameters in the **CfgModOemSoftkey** config object:

MP_activation

Specifies whether the defined function is a foreground application.
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **TRUE**
The defined function is a foreground application, e.g. config editor. Set this value if the soft key opens the config editor to display user parameters.
FALSE
Function performed in the background

Default: FALSE

Access: LEVEL3

Reaction: RESET

MP_skPos

Position of the soft key in the 3rd menu bar
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: 0 to 7
0: first soft key from the left
No entry: the soft key is not displayed
Menu bars 1 and 2 are reserved for HEIDENHAIN.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

The **buttonText** folder contains two machine parameters used to define the labeling of the soft key. You can either define a reference to the text in a language-sensitive *.CSV file or a language-neutral text. Leave the **buttonText** empty if you want to define an image for the soft key with **MP_buttonImage**.

MP_dialogRes

Reference to a language-sensitive text in a *.CSV file.
Available from NCK software version: 597 110-01.

Format: String

Input: Max. 40 characters
Example:
OEMTXT_001_ncLanguage
The designator must be available in a text resource file (*.CSV). The text file must be defined under System/Paths/CfgOemPath/dialogTextfile.
Leave the parameter empty if you do not want the soft key labeling to be language-sensitive. Enter the text directly under**MP_text** instead.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

MP_text

Soft key designator.
Available from NCK software version: 597 110-01.

Format: String

Input: Max. 60 characters
Enter a language-neutral text. If a language-sensitive text was defined with **MP_dialogRes** you must not change the displayed string; the control displays the entry from a *.CSV file.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

MP_buttonImage

Image for soft keys
Available from NCK software version: 597 110-01.

Format: String

Input: Path/file name to an image for a soft key
Leave the parameter empty if you have defined a text for the soft key under **buttonText**.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

MP_funcKey

Key name of the function being activated when the soft key is pressed
Available from NCK software version: 597 110-01.

Format: String

Input: Max. 18 characters
The key name must be entered only if the name of the function does not correspond to the key name.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

MP_helpId

Symbolic name of the context information for online help (*.CHM)
Available from NCK software version: 597 110-04.

Format: String
Input: Max. 80 characters
The entered symbolic name can belong to a HEIDENHAIN or OEM manual.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

Machine parameters in the **CfgCfgEditActivate** config object:

MP_layoutFile

Path/name of the XML layout file for user parameters
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 80 characters
Default: Example for user parameters (code number 123)
%OEM%\config\layout\user123.xml
Access: LEVEL3
Reaction: NOTHING

MP_dialogTextfile

Name of the text file for OEM texts
Available from NCK software version: 597 110-01.

Format: Array [0...9]
Input: Enter the name (without the path!) for OEM text files. You can enter any files with the file extension *.CSV, in which your manufacturer-specific texts are defined.
The path %OEM%\plc\language\<language> is fixed, whereby <language> is formed from the configured conversational language, e.g. “en” for English.
Default: UserParam.CSV
Access: LEVEL3
Reaction: RESET

MP_readOnly

Open the config editor with read access only.
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: TRUE
The config editor is opened with read access only; the parameter values cannot be changed
FALSE
The config editor is opened with read and write access
Default: No value, parameter optional (= read and write access)
Access: LEVEL3
Reaction: RESET

Machine parameters in the **CfgOemPath** config object:

MP_dispLangText

Display language-sensitive names in the configuration editor?
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Language-sensitive names are displayed.
FALSE
System names are displayed.
Default: TRUE
Access: LEVEL3
Reaction: NOTHING

Note

All files can be managed with PLCdesignNT, as well as transmitted to the control. PLCdesignNT is available from HEIDENHAIN.

Definition of file types

Settings in the configuration editor	MP number
CfgConfigDataFiles **dataFiles**	106303

The file types and default settings of the PLC user parameters are defined in the file **plc_user.cfg.** The file can be named anything you want. The path of the file is to be entered in the **dataFiles** list of the **CfgConfigDataFiles** entity, which appears after pressing the **CONFIG FILE LISTS** soft key. The file types **CfgOemBool** and **CfgOemInt** can be used in the parameter layout for parameters of protection class 1 (code number 1234).

Definition of code numbers

In order to change the code number for PLC parameters (5555), the file **plc_attr.cfg** must be modified.

Example:
In order to display the PLC parameters, the code number 123456 is to be valid in addition to the code number 5555.

Add the following entity to the **plc_attr.cfg** file:

CfgPassword (

key:="123456",

funcList:= [

„CONFIGEDIT-PLC_Parameter",

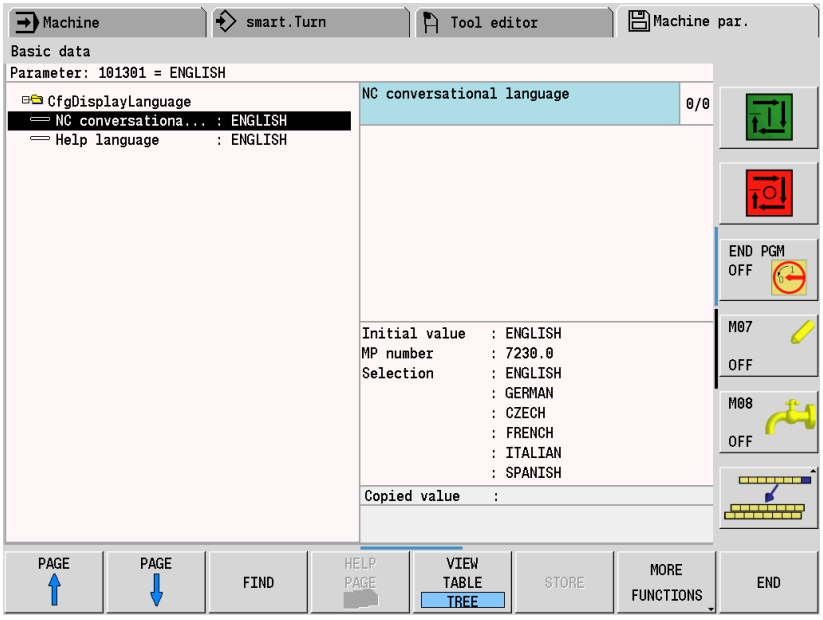
„CONFIG-LEVEL1"

])

4.3.2 Example:

Creating a layout for the user parameters. The example illustrates how you enable the machine operator to access a parameter from your machine configuration as a user parameter.

Goal:
Creation of a selection menu for setting the language. The menu is to appear on the screen after entering the code number 123.



The best solution for handling XML files is to use an XML editor. These editors are available from software vendors, or as freeware from the Internet for example. As an alternative, the XML layout file can also be edited with a text editor, such as Notepad. However, XML editors will help you a great deal in working with structured XML files.

- First step: Create a new empty XML file with the file name “user123.xml”. Example:

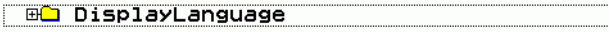
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- user123.xml from HEIDENHAIN 19.07.2005 -->
```

- Then specify the name of the XML schema file (XSD file). The XML editor uses the XML schema file to validate the created XML file. The XML editor verifies the file formally each time it is saved. This formal verification is based on the information in the *.XSD file. You will find the “configtreelayout.xsd“ file on the control under %OEM%\config\layout\. You can use this file to validate your new user parameter layout. For this purpose, you must copy the file into the project directory of your XML editor on your PC by using TNCremoNT.

▶ The XML file will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!– user123.xml from HEIDENHAIN 19.07.2005 –>
<ConfigtreeLayout xmlns="http://www.hng.ch/ConfigEditLayout/1"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.hng.ch/ConfigEditLayout/1
 configtreelayout.xsd">
```

▶ Then the “conversational language” folder is created:



The following entry is therefore added to the XML file:

```
<Node Name="CfgDisplayLanguage"
 DialogRes="OEMTXT_CfgDisplayLanguage">
```

Use the command **DialogRes=** to assign a language-sensitive name (here: “conversational language”) from the OEM text file (.*CSV) to the **CfgDisplayLanguage** folder. If you want to display this text as folder name by default, you must set **MP_dispLangText** to TRUE.

▶ In the previously created “conversational language” folder, two machine parameters for setting the language of the NC dialog texts and the help texts are to be defined:

For this, the following entries are required in the XML files:

```
<SimpleAttribute Keyfilter="*" Name="ncLanguage"
Entity="CfgDisplayLanguage" DialogRes="OEMTXT_ncLanguage">
</SimpleAttribute>
<SimpleAttribute Keyfilter="*" Name="helpLanguage"
Entity="CfgDisplayLanguage" DialogRes="OEMTXT_helpLanguage">
</SimpleAttribute>
```

▶ Then you conclude the “conversational language” folder and the XML schema:

```
</Node>
</ConfigtreeLayout>
```

▶ The file is now complete and can be transferred to the control by using TNCremoNT.

▶ Then you still need to make the entries for the OEM texts in the *.CSV file and to adjust the machine configuration accordingly.

Note

You will find a preconfigured version of the “user123.xml” file on the control under %OEM%\config\layout\. This file is only meant as a suggestion by HEIDENHAIN and you can adjust it to your specific requirements.

4.3.3 XML commands for creating the layout files

Overview of XML commands for defining the layout of the user parameters:

XML command	Description
Include Example: **<Include File="%OEM%\config\layout\axes.xml"/>**	
	■ Inserts the layout definition of the specified file in the layout. ■ **File:** File name and path of the file to be inserted.
Node Example: **<Node Name="DisplaySettings" DialogRes="OEMTXT_DisplaySettings">**</Node>	
	■ Defines a directory (branch, node) in the layout. ■ Permissible elements are: Node, Keylist, Object, UniqueObject, SimpleAttribute, Group, Key, Array ■ **Name:** Language-independent designation. Can be chosen as desired. ■ **DialogRes:** Text name from the OEM text file for displaying a language-sensitive designation.
Keylist Example: **<Keylist Name="ChannelSettings" Keyfilter="*"**	
	■ List of key names. The key names are displayed as directories. ■ Permissible element: Entity ■ **Keyfilter:** Filter for the key name. Permissible control characters for the filter are '*', '?' (wildcards) and '.'. ■ **DialogRes:** Text name from the OEM text file for displaying a language-sensitive designation.
Entity Example: **<Entity Name="CfgNcErrorReaction"/>**	
	■ Data object as element of a key list. The object name is displayed as directory. ■ Permissible element: Attributes ■ **Name:** Name of the data object ■ **DataOfAttribute:** Name of an attribute whose data are displayed here. The name of the attribute is not displayed. ■ **DialogRes:** Text name from the OEM text file for displaying a language-sensitive designation.

XML command	Description
Object Example: `<Object Name="CfgPosDisplayPace" Keyfilter="*"/>`	■ List with objects. The object name is displayed as the first directory, and the key name as the second. The key name can be filtered. ■ Permissible element: Attributes ■ **Name:** Name of the data object ■ **Keyfilter:** Filter for the key name. For permissible control character, see above. ■ **DataOfAttribute:** Name of an attribute whose data are displayed here. The name of the attribute is not displayed. ■ **DialogRes:** Text name from the OEM text file for displaying a language-sensitive designation.
UniqueObject Example: `<UniqueObject Name="CfgDisplayLanguage"/>`	■ Data object with an empty key name. These are data objects that occur only once. Only the object name is displayed. ■ Permissible element: Attributes ■ **Name:** Name of the data object ■ **DataOfAttribute:** Name of an attribute whose data are displayed here. The name of the attribute is not displayed. ■ **DialogRes:** Text name from the OEM text file for displaying a language-sensitive designation
Attributes	■ Use this command if only certain attributes are to be displayed. If this element is not specified, all attributes of a data object will be displayed (standard behavior). If the attribute is a field or a list, an element of a list can be defined by specifying its index in square brackets. Example: **[3]** ■ **Name:** Name of the attribute ■ **DialogRes:** Text name from the OEM text file for displaying a language-sensitive designation

XML command	Description
SimpleAttribute Example: **<SimpleAttribute Keyfilter="*" Name="ncLanguage" Entity="CfgDisplayLanguage" DialogRes="OEMTXT_ncLanguage"></SimpleAttribute>**	
	■ Attribute without display of object name and key name. If the attribute is used as a subelement of a node, an existing key name must be specified (i.e. the key filter must not contain any filter characters). ■ **Entity:** Object name ■ **Keyfilter:** Filter for the key name. ■ **Name:** Name of the attribute. If the attribute is a field or a list, an element of a list can be defined by specifying its index in square brackets. Example: **[3]** ■ **DialogRes:** Text name from the OEM text file for displaying a language-sensitive designation

Settings in the configuration editor	MP number
KeySynonym	
CfgKeySynonym	
[Key name for the new parameter set]	
relatedTo	109501
excludeList	109502

KeySynonym is a very helpful function for creating a new parameter set. You use KeySynonym to create parameter sets that refer to existing parameter sets and reuse the data contained in them. New parameter sets can be linked to existing ones in **KeySynonym/CfgKeySynonym**. Then you only have to define the parameters that differ from the ones of the parameter set to which you have related the present parameter set.

Proceed as follows if you want to create a new parameter set using the KeySynonym function:

- Open the configuration editor.
- You will find the KeySynonym folder at the bottom of the configuration editor. Move the cursor to the folder and open it.
- Move the cursor to the **CfgKeySynonym** configuration object.
- Press the **INSERT** soft key.
- The control prompts you to enter a new key name. Now enter a name for the new parameter set, such as ParSetS-1 for an additional set of parameters for the spindle. Also specify via the pull-down menu the memory file in which the data of the new parameter set are to be stored.
- The parameter-set key you just created contains the machine parameter **MP_relatedTo.** There you enter the original parameter set, i.e. the one to which the new parameter set is to be related to, such as ParSetS-0. This assigns (links) the values from your 0-set for the spindle to all parameters of the new set. Changes to the 0-set are automatically loaded by the linked parameter set.
- Save your changes with the **SAVE** soft key.
- A red message appears, stating that the control must be rebooted. Restart the control now.
- Open the configuration editor when the control has rebooted.
- If you look in the ParameterSets configuration folder now, you will see a new ParSetS-0 parameter set there. The yellow arrow in front of the parameter set and entities indicates that the parameters are being mirrored by the KeySynonym function.

- Now you can undo the linking of entities, as required. This means you enter values that deviate from the original set of parameters. Select the desired entity and press the **MORE FUNCTIONS** soft key and then the **INSERT** soft key. The yellow arrow then disappears.
- Select the memory file via the pull-down menu.
- Save your changes with the **SAVE** soft key.
- The parameter set has now been created with the KeySynonym function, and can be used.

MP_relatedTo

	Link to key name Available from NCK software version: 597 110-01.
Format:	String
Input:	Max. 18 characters The synonym name refers to the key name specified here. The data of the parameter object with the indicated key name are used for the parameter object with the synonym name. Example: **CH_SIM** uses the same data as **CH_NC**
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

MP_excludeList

	List with configuration objects to be excluded Available from NCK software version: 597 110-01.
Format:	Array [0...49]
Input:	Name of configuration objects; example: CfgPositionLimits Cancel the reference at CfgKeySynonym/relatedTo for the entered parameter objects.
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

4.5 Allocation of Configuration Data

The configuration data is saved in several files with the extension .cfg (see "Managing configuration files" on page 261). Paths saved in the file **configfiles.cfg** refer to these files.
This allocation has already been specified by HEIDENHAIN when the software is delivered, but it can be adapted by the OEM to his requirements.

Allocation of Configuration Data

O:\config contains the following data:

■ configfiles.cfg	Paths to the configuration data
■ version.cfg	Version number of the update rules (JH, OEM)

O:\config\lathe\manplus contains the following data:

■ configVersion.cfg	Directory path of the config. data
■ KeyList.cfg	Configuration of the system key names
■ ch1.cfg	Data for the machining channel
■ plc.cfg	Config. data for the PLC run-time system
■ plc_oem.cfg e.g.	PLC parameters that are relevant for the OEM, **MP_stoppingAngle** and **MP_maxSpeedSpindle**
■ plc_user.cfg	PLC user parameters
■ kin_ch1_s1.cfg	Old kinematics description
■ kinem*.cfg	New kinematics description
■ th1.cfg	Configuration of the 1st tool holder
■ axis_*.cfg	Each drive has its own configuration file. This way you can change the drive easily.
■ aggregate.cfg	Data for the aggregates
■ proc_data.cfg	Data for the machining mode of operation
■ oem.cfg	General data for systems and channels
■ simu.cfg	Simulation settings
■ oemtable.cfg	OEM-specific settings for tables
■ table.cfg	General settings for tables
■ table_mplus.cfg	MANUALplus 620 specific table settings
■ dashboard.cfg	Dashboard settings
■ oempassword.cfg	OEM-specific password settings, e.g. for file and directory release or special config data views.
■ runtime.cfg	General data for the run-time system of the control (reserved for HEIDENHAIN)
■ UpdateOemRel*.cfg	OEM-specific update rules for PLC parameters

V:\config contains the following data:

■ mptransfer.cfg	Configuration of network settings
■ user.cfg	General user parameters
■ user_ch1.cfg	User parameters for the machining channel

4.6 Structure of a Parameter File

The individual machine parameters are collected into parameter objects in the *.cfg parameter files. A parameter object has a name, of which the first three letters are always "Cfg." The name is followed by an open parenthesis and a "key" for identifying the parameter object. This is followed by the individual machine parameters. A parameter object must be surrounded by parentheses.

If there are several input values for a parameter (such as separate parameter settings for each axis), then the corresponding parameter objects are addressed via the key, and therefore occur more than once.
The parameter objects in the system files have an "empty" key.

Note

HEIDENHAIN recommends changing the parameter values directly in the *.cfg files only in exceptional cases.

Rules for entries

If changes are to be made directly in a text editor, the following rules must be observed:

- "Key": Each parameter object has a key at the beginning, which generally represents the name of the axis or channel, but in certain cases can also be empty. The control assigns this parameter object to the object addressed by the key, for example to the axis "X axis".
- The characters := must come between the parameter name and the value.
- Individual parameters must be separated by commas. No comma may follow the last parameter.
- Individual components, such as strings in a list or the components in the array must be separated by commas. No comma may follow the last component.
- The different levels in path entries must always be separated by "\\", for example, "%SYS%\\CONFIG\\AXIS\\...".
- A list must always be in brackets [ ].
- If data objects with the same names and same identifications (keys) are present, the error message "Data object already exists in file" is displayed.
- Comments are text that is ignored during transfer. You can enter two types of comments:
  - Comment in one line: After a double hyphen "–" the text until the end of the line is ignored.
  - Comments that are on more than one line must be surrounded like this: (*comment*).
- Comments in files that are overwritten by the control (such as files with axis-setting parameters or oscilloscope parameters) are deleted. For this reason you should only add comments to files that are not written to by the control.

Example of a parameter file *.cfg:

Parameter object	Description
CfgChannelAxes(	Name of the parameter object with open parenthesis. You cannot change this name.
Key:= "Channel1",	Identification of the parameter object with a string, such as the name of the NC channel or an axis.
progAxis:= [	Data variable from a list.
"X axis", "Y axis", "Z axis", “A axis”, “B axis”], ...	The individual elements of a list are separated by commas. No comma may follow the last element in a list. A list must always be in brackets [ ].
[.....], ...	More parameters follow…
)	No comma may precede the closing parenthesis. Conclusion of the parameter object
-- Comment to the end of the line	The text in the line after “--” (double hyphen) is ignored
(* Comment distributed over several lines *)	Characters for comment beginning Everything between the comment beginning and end is ignored Characters for comment end

Parameter object	Description
CfgAxis(	Name of the parameter object with open parenthesis. You cannot change this name.
Key:= "S1",	Identification of the parameter object with a string, such as the name of the NC channel or an axis.
isAng:= TRUE,	Boolean type data variable
.....,	More parameters follow…
parList:= [	Data variable from a list.

Parameter object	Description
"PS1_0", "PS1_1", "PS1_2", "PS1_3", "PS1_C1"],	The individual elements of a list are separated by commas. No comma may follow the last element in a list. A list must always be in brackets [ ].
noActToNomAtEmSt:=FALSE	Another parameter follows…
)	No comma may precede the closing parenthesis. Conclusion of the parameter object
-- Comment to the end of the line	The text in the line after "--" (double hyphen) is ignored
(* Comment distributed over several lines *)	Characters for comment beginning Everything between the comment beginning and end is ignored Characters for comment end

4.7 Machine-Parameter Subfiles

Individual data from the configuration files can be taken into the MP subfiles. These subfiles can be superimposed on the machine parameters during run time.
In principle, all files that do not require a system restart can be superimposed. The MP subfiles are usually activated by the PLC, but they can also be activated manually by using the configuration editor.

4.7.1 Syntax of machine parameter subfile

The syntax of an MP subfile corresponds to that of a basic file.
Subfiles differ from basic files in that only the entities or attributes to be changed must be described.
In basic files a data object (entity) must be described completely. This means that the basic file must contain the "key" and all "attributes" of the entity.

```
entity(
  key:= Key4711,
  attribute1:= value1,
  attribute2:= value2,
  attribute3:= value3,
  attribute4:= value4,
)
```

In subfiles, only the required data need to be indicated. Entity and key, however, must always be indicated.
Please note: MP subfiles must not contain any reset parameters.

Example of MP subfile with a new value for attribute 3:

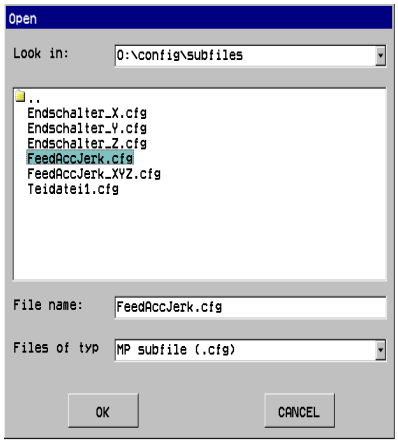
```
entity(
  key:= Key4711,
  attribute3:= valuex
)
```

4.7.2 Activating the machine parameter subfile

MP subfiles can be activated in the configuration editor or by the PLC.

Activation in the configuration editor

Use the **LOAD MP SUBFILE** soft key to activate individual subfiles.



A file will be loaded and activated immediately upon selection.

The PLC marker **NN_GenCycleAfterReConfig** is set upon activation if data relevant to the PLC have changed. Irrelevant data is indicated in the %SYS%\\config\\jh.cfg file as follows:

```
CfgNoNotification (
  key:="plc.QM4174",
  objectNames:=[
    "CfgOsci",
    "CfgOsciFile",
    "CfgOsciColor",
    "CfgOsciSetUp",
    "CfgOsciChannel",
    "CfgOsciTrigger",
    "CfgOsciDisplay",
    "CfgSelectFile",
    "CfgRecentFileList",
    "CfgDisplayData",
    "CfgPosDisplayPace",
    "CfgJogIncrement",
    "CfgInterpretOption",
    "CfgHandWheelFactor",
    "CfgAutoStartData",
    "CfgFeedRate",
    "CfgLayoutData",
    "CfgTablePath",
    "CfgEditor",
    "CfgGeoRotWorkPlane",
    "CfgUserPath",
    "CfgUnitOfMeasure",
    "CfgProgramMode",
    "CfgPassword",
    "CfgFunctionProtection",
    "CfgActualProtection",
    "CfgJhProtection",
    "CfgModSkText"
    ]
)
```

Activation by the PLC

Subfiles are activated by Module 9034. In this case the symbolic PLC operand **NN_GenCycleAfterReConfig** (M4174) will not be set.

Module 9034 Load a machine parameter subfile

With this module you load the contents of the given configuration file into the main memory. All configuration data not listed in this file remain unchanged.

The configuration file to be selected is checked. A faulty file is not loaded. If the file contains parameters that require a system reset, the file is not loaded.

The file name is transferred in a string that must contain the complete path, name and file extension. Further characters, even space characters, are not permitted.

If the PLC program is created externally, ensure that lower-case letters are not used for the file name!

Once the NC program has started, the module operates only during the output of an M/S/T/Q strobe.

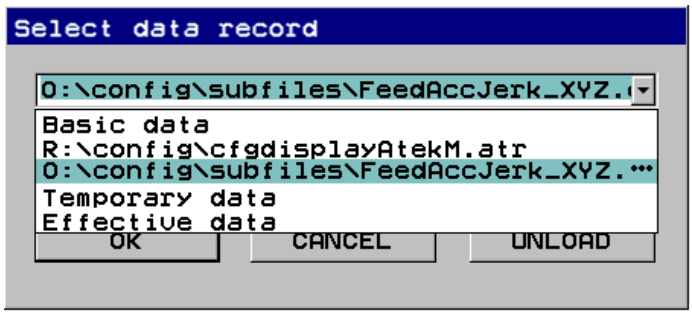
Call only in a submit job.

Call:

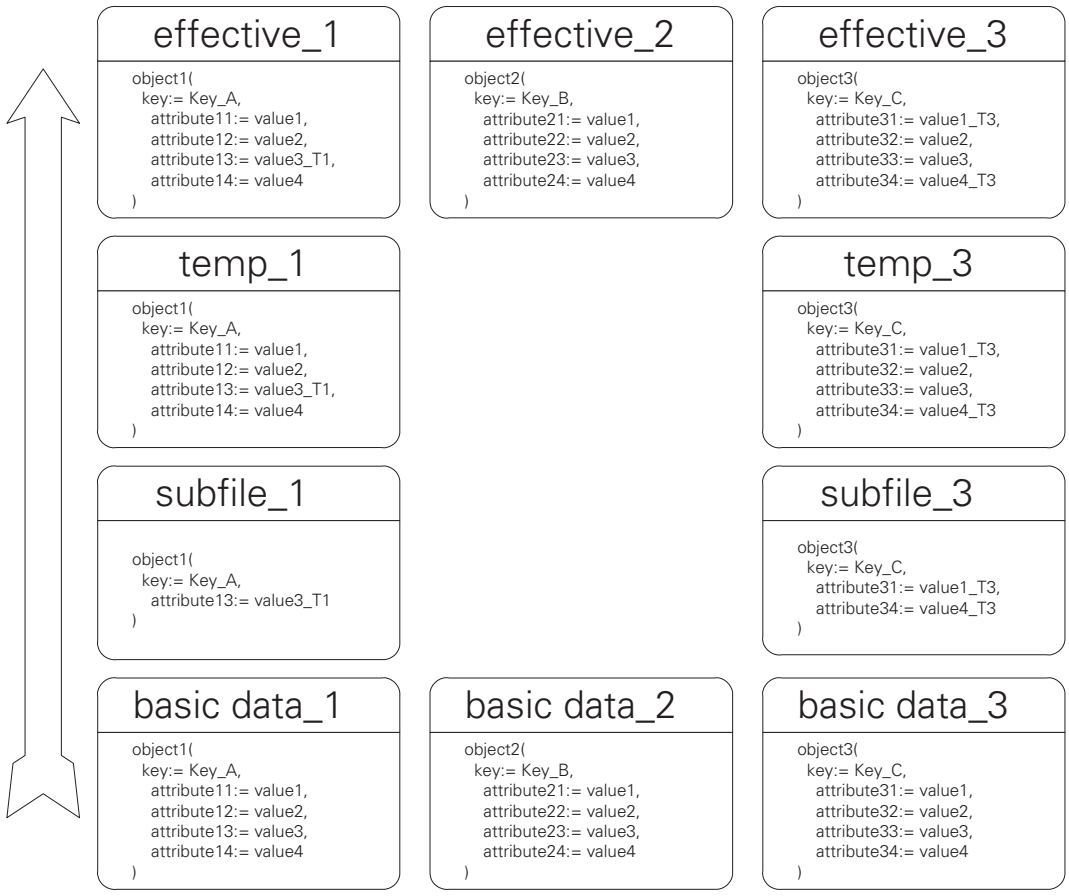
PS	B/W/D/K	<>String number> 0 to 99
CM	9034	
PL	B/W/D	<>Error code> 0: No error 1: String does not contain a valid file name, or the name (including the path) is too long. 2: File not found 3: File is faulty / contains reset parameters 4: Incorrect string number was transferred (0 to 3). 5: Call was not in a submit job 6: Call during running program without strobe.

4.7.3 Displaying/editing data records in the configuration editor

Use the **SELECT DATA RECORD** soft key to choose between the following views:



The following overview shows which values of an object (object1, 2, 3) are displayed in the individual views:



Basic data

This view shows the data imported during system start-up. Any changes will be rewritten to the respective basic files.

Loaded subfile(s)

You must have loaded a subfile during system start-up (with "CfgPortionFiles") or by soft key for the subfile to be shown. If you select a subfile, only the data of the subfile are displayed in the configuration editor. The attributes contained in the subfile can be edited and rewritten to the subfile while saving.
The subfile must be reloaded for the edited data to become effective.

Temporary data

The complete data objects of all loaded subfiles are shown. They can be edited, but they are not rewritten to the file. If a subfile from the PLC is loaded, the data is saved only as temporary data.

Effective data

The "effective data" view does not permit data to be edited. The displayed data refer to the current data loaded by the PLC or by selecting the data record of basic data or subfiles, and therefore show the data effective in the system.

Unload subfile

The **UNLOAD** soft key can be used to unload the selected subfile.
This automatically activates the basic data and the subfiles that are still loaded. The result is also shown in "Temporary data" and "Effective data."

4.8 Read or Change Machine Parameters via a PLC Module

Machine parameters can be read and overwritten via the PLC.

Module 9430 Change the numeric value of a machine parameter
Use this module to enter a numeric value in the machine parameter given. The value of the machine parameter is overwritten in the run-time memory. The machine parameter in the .cfg file is not overwritten. The overwritten parameters are only in effect until the next control start-up.

The NC software prescribes when a machine parameter can be changed. This information is included in the description of each parameter in the Technical Manual under "Reaction" or in the help display (**Info**key) of the configuration editor under "Behavior."

Constraints:

- The module can only be called in a spawn or submit job.
- The module can only be called while the PLC has control over the machine.

Call:

PS	B/W/D/K/S	<>Name of the configuration object> e.g. CfgAxisHardware
PS	B/W/D/K/S	<>Key name> Key name of the configuration object
PS	B/W/D/K/S	<>Name of the machine parameter>
PS	B/W/D/K	<>Index> Index of the parameter within an array (list field) 0: Parameter without array (list field)
PS	B/W/D/K	<>Factor> Conversion factor for real to integer, and vice-versa
PS	B/W/D/K	<>Value> Value of the parameter
CM	9430	
PL	B/W/D	<>Error> See "Return codes of PLC modules 9430-9433 (error stack)" on page 305.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	No error, data was changed
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Parameter out of value range
	2	Parameter invalid or does not exist
	6	Changing this value not allowed
	7	File faulty / could not be read
	13	No connection
	20	Module was not started from a spawn or submit job
	21	Module was called during program run without strobe

Entries for PS/PL	Description of the module entries
<Name of the configuration object>	Name of the object in which the parameter (attribute name) to be changed is located; e.g.: "PlcCfgValue"
<Key name>	Key where the object is located; e.g.: "Channel1"
<Name of the machine parameter>	Name of the machine parameter (attribute) that is to be changed; e.g.: MP_swLimitSwitchPos
<Index>	Index within an array (list field); 0= for parameters without array
<Factor>	Conversion factor for real to integer, and vice-versa
<Value>	Value of the parameter, e.g.: 123456
<String>	String number (0 to 15)
<Frror>	See module description

Module 9431 Read the numeric value of a machine parameter
Use this module to read the value of the given machine parameter from the run-time memory.

Constraints:

- The module can only be called in a spawn or submit job.

Call:

PS B/W/D/K/S<>Name of the configuration object>
e.g. CfgAxisHardware
PS B/W/D/K/S<>Key name>
Key name of the configuration object
PS B/W/D/K/S<>Name of the machine parameter>
PS B/W/D/K <>Index>
Index of the parameter within an array (list field)
0: Parameter without array (list field)
PS B/W/D/K <>Factor>
Conversion factor for real to integer, and vice-versa
PS B/W/D/K <>Value>
Value of the parameter
CM 9431
PL B/W/D <>Error>
See "Return codes of PLC modules 9430-9433 (error stack)" on page 305.

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	No error, data was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Parameter out of value range
	2	Parameter invalid or does not exist
	13	No connection
	20	Module was not started from a spawn or submit job

Module 9432 Change the string value of a machine parameter
Use this module to enter a string in the machine parameter given. The value of the machine parameter is overwritten in the run-time memory. The machine parameter in the .cfg file is not overwritten. The overwritten parameters are only in effect until the next control start-up.

Call only in a submit job.

Call:
PS B/W/D/K/S<>Name of the configuration object>
e.g. CfgAxisHardware
PS B/W/D/K/S<>Key name>
Key name of the configuration object
PS B/W/D/K/S<>Name of the machine parameter>
PS B/W/D/K <>Index>
Index of the parameter within an array (list field)
0: Parameter without array (list field)
PS B/W/D/K/S<>New string>
CM 9432
PL B/W/D <>Error>
See "Return codes of PLC modules 9430-9433 (error stack)" on page 305.

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	No error, data was changed
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Parameter out of value range
	2	Parameter invalid or does not exist
	13	No connection
	20	Module was not started from a spawn or submit job
	21	Module was called during program run without strobe

Module 9433 Read the string value of a machine parameter
Use this module to read the value of the given machine parameter from the run-time memory.

Call only in a submit job.

Call:
PS B/W/D/K/S<>Name of the configuration object>
e.g. CfgAxisHardware
PS B/W/D/K/S<>Key name>
Key name of the configuration object
PS B/W/D/K/S<>Name of the machine parameter>
PS B/W/D/K <>Index>
Index of the parameter within an array (list field)
0: Parameter without array (list field)
PS B/W/D/K <>String number 0...15>
CM 9433
PL B/W/D <>Error>
See "Return codes of PLC modules 9430-9433 (error stack)" on page 305.

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	No error, data was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Parameter out of value range
	2	Parameter invalid or does not exist
	13	No connection
	20	Module was not started from a spawn or submit job

Return codes of PLC modules 9430-9433 (error stack)

The following errors can occur when running Module 9430 to 9433, and are returned to the error stack:

Value (error stack)	Meaning
1	Parameter does not exist, cannot be changed, or cannot be changed during program run
3	Fatal error (no connection to config server, etc.)
5	Call during program run without a strobe
6	Module was not called in a submit/spawn job
7	Call parameter call invalid, for example parameter is not numeric or parameter is not a string
8	Entity/attribute unknown
9	Configuration data does not exist
10	Configuration data already exists
11	Wrong type
12	Value invalid
13	Attribute invalid
14	Server error
15	File error
16	Internal error

4.9 Switching Parameter Sets

You can store multiple parameter sets for an axis or spindle in the configuration editor.

Switching of parameter sets can be necessary, for example,

- if other parameters are necessary for a motor for different load or shaft speed ranges.
- if control is to be dependent on the load during C-axis operation.
- if a wye/delta connection switchover is carried out for the spindle motor.

If there is more than one parameter set for one axis, use Module 9434 to select the desired parameter set and Module 9435 to check the currently active parameter set.

Danger

Ensure that switching the machine parameters does not result in the machine entering an uncontrolled state.

HEIDENHAIN recommends:

When switching the parameters, the affected drive should be switched off, or at least brought to a standstill! Controller and filter parameters should only be switched while a drive is at a standstill, and not during machining, since the switchover could have an affect on the workpiece surface.

Incorrect or non-optimized parameter input values can lead to faulty machine performance!

HEIDENHAIN does not accept any responsibility for direct or indirect damage or injury caused to property or persons through improper use or incorrect parameterization of the machine.

Conditions

Conditions of the axis:

In principle, you are free to change any machine parameters of an axis parameter set. However, when some parameters are changed, the configuration editor triggers a reboot (RESET) of the control. These parameters therefore cannot be switched.

If when activating or deactivating axes over the PLC you also switch parameter sets (e.g. when exchanging milling heads), you have to observe particular conditions, see "Writing axis information—activating and deactivating axes" on page 444. The control checks for individual changed parameters immediately after the change to another activation condition. However, only those parameters are checked that are relevant for the respective activation condition. An axis in the activation condition "manually operated axis," for example, does not require controller parameters, and an axis in the "deactivated axis" activation condition does not need any parameters at all.

If the activation condition is switched over Module 9226 or 9418, the control then checks

- whether all required machine parameters are available, and
- whether the machine parameters have changed since the last use of this activation condition.

In this way, for example in a milling head exchange, an axis can be switched to the "deactivated axis" activation condition, the parameter set can be switched over (to other encoder configurations) and then the axis can be reactivated.

With some parameters you have to comply with the following conditions with regard to the switchover:

- **DRIVE OFF:**
  The motor of the affected axis must be switched off. If this is not the case, the control releases an error message with the reaction EMERGENCY STOP. This causes the drives to be switched off. The control conducts a parameter-set switchover.
- **AXIS DEACTIVATED:**
  The axis must be deactivated (e.g. over Module 9226).
  If this is not the case, the control releases the error message **`For a parameter change the drive has to be deactivated`** (reaction: EMERGENCY STOP). This switches the drives are off. The control **does not** conduct a parameter-set switchover. If you have changed a parameter requiring the condition AXIS DEACTIVATED via the config editor, the change will only become effective after the next control start-up.
- **REF:**
  The axis concerned must be homed again after the parameter set switchover.

The following table shows the parameters for which you must fulfill these conditions

Synchronization with NC-program calculation

Before the parameter set can be switched, some machine parameters require a synchronization with the look-ahead capability of the NC program. Perform this synchronization with a strobe in which the value SYNC_CALC is configured for **MP_sync.** The strobe with SYNC_CALC interrupts the interpretation of the NC program, and the geometry chain is processed.

In addition to the conditions for switching the parameter set, the following table presents an overview of all drive parameters which require either a strobe with SYNC_CALC or that the drive be switched off before the switchover.

Attention

The directives of this table must be followed in order to avoid uncontrolled behavior of the machine!

Machine parameters	RESET	Strobe with SYNC_CALC	DRIVE OFF	AXIS DISABLED	REF
CfgHardware					
hardwareType	X	–	–	–	–
l32stopsMonitoring	–	–	–	–	–
i32driveEnabling	–	–	–	–	–
currentControlAdjust	X	–	–	–	–
maxTouchFeed	–	–	–	–	–
driveOffGroupInput	–	–	–	–	–
CfgCycleTimes					
ipoCycle	X	–	–	–	–
plcCount	X	–	–	–	–
watchdogTime	–	–	–	–	–
CfgFilter					
typeFilter1	–	–	–	–	–
orderFilter1	–	–	–	–	–
typeFilter2	–	–	–	–	–
orderFilter2	–	–	–	–	–
CfgHandwheel					
All parameters	–	–	–	–	–
CfgPosCorrection					
enable	–	–	–	–	–
feed	–	–	–	–	–
CfgPlcFastInput					
All parameters	–	–	–	–	–
CfgChannelAxes					
All parameters	–	–	–	–	–

Machine parameters	RESET	Strobe with SYNC_CALC	DRIVE OFF	AXIS DISABLED	REF
CfgLaPath					
minPathFeed	–	–	–	–	–
minCornerFeed	–	–	–	–	–
maxG1Feed	–	–	–	–	–
maxPathJerk	–	–	–	–	–
maxPathJerkHi	–	–	–	–	–
pathTolerance	–	–	–	–	–
pathToleranceHi	–	–	–	–	–
maxPathYank	–	–	–	–	–
CfgAxis					
isAng	X	–	–	–	–
isModulo	–	X	X	–	–
restoreModuloCntr	–	–	–	–	–
isHirth	–	–	–	–	–
axisHw	–	X	X	–	–
axisMode	X	–	–	–	–
testMode	X	–	–	–	–
parList	X	–	–	–	–
realAxis	X	–	–	–	–
noActToNomAtEmSt	–	–	–	–	–
deactivatedAtStart	–	–	–	–	–
advancedSettings	–	–	–	–	–
CfgAxisHardware					
signCorrActualVal		X	X		X
signCorrNominalVal		X	X		X
posEncoderType	–	X	–	X	X
distPerMotorTurn	–	–	–	–	–
posEncoderDist	–	–	–	–	X
posEncoderIncr	–	–	–	–	X
posEncoderSignal	–	–	–	–	–
posEncoderFreq	–	–	–	–	–
posEncoderResistor	–	–	–	–	–
posEncoderRefDist	–	–	–	–	X
genExtIntPolFactor	–	–	–	–	X
posEncoderInput	–	X	X	–	X
speedEncoderInput	–	X	–	X	X
hsciCcIndex	–	X	–	X	X
pwmSignalOutput	–	X	–	X	X
ctrlPerformance	–	X	X	–	–
driveOffGroup	–	–	–	–	–
checkPhiFieldRef	X	–	–	–	–

Machine parameters	RESET	Strobe with SYNC_CALC	DRIVE OFF	AXIS DISABLED	REF
CfgAxisAnalog					
analogOutput	–	–	–	X	–
analogOffset	–	–	–	–	–
kvFactor2	–	–	–	–	–
kvSpeedLimit	–	–	–	–	–
maxFeedAt9V	–	–	–	–	–
accForwardFactor	–	–	–	–	–
compStrength	–	–	–	–	–
compWidth	–	–	–	–	–
compTimeOffset	–	–	–	–	–
compFFAdjust	–	–	–	–	–
compRefAcc	–	–	–	–	–
compLimitFactor	–	–	–	–	–
noOffsetAdjust	–	–	–	–	–
unipolar	–	–	–	–	–
CfgFeedLimits					
minFeed	–	X	–	–	–
maxFeed	–	X	–	–	–
rapidFeed	–	X	–	–	–
manualFeed	–	X	–	–	–
maxAcceleration	–	X	–	–	–
maxAccSpeedCtrl	–	X	–	–	–
maxDecSpeedCtrl	–	X	–	–	–
M19MaxSpeed	–	X	–	–	–
M19NcSpeed	–	X	–	–	–
nominalSpeed	–	X	–	–	–
restoreFeed	–	X	–	–	–
CfgReferencing					
refType	–	X	–	–	X
endatSerial	–	X	–	X	X
refPosition	–	X	–	–	X
refSwitchActive	–	X	–	–	X
externRefPulse	–	–	–	–	X
endatDiff	–	X	–	–	–
refFeedLow	–	X	–	–	–
refFeedHigh	–	X	–	–	–
refDirection	–	X	–	–	X
moveAfterRef	–	X	–	–	–
moveAfterRefType	–	X	–	–	–
moveAfterRefPos	–	X	–	–	–
moveAfterRefFeed	–	X	–	–	–

Machine parameters	RESET	Strobe with SYNC_CALC	DRIVE OFF	AXIS DISABLED	REF
CfgPositioniLimits					
swLimitSwitchPos	–	X	–	–	–
swLimitSwitchNeg	–	X	–	–	–
lubricationDist	–	X	–	–	–
CfgControllerAuxil					
driveOffLagMonitor	–	–	–	–	–
checkPosStandstill	–	–	–	–	–
maxPosDiff	–	–	–	–	–
posDiffCountDir	–	–	–	–	X
CfgEncoderMonitor					
checkAbsolutPos	–	X	X	–	–
checkSignalLevel	–	–	–	–	–
checkFrequency	–	–	–	–	–
checkRefDistance	–	–	–	–	–
movementThreshold	–	–	–	–	–
thresholdDistance	–	–	–	–	–
CfgSpindle					
All parameters	–	–	–	–	–
CfgPosControl					
kvFactor	–	–	–	–	–
servoLagMin1	–	–	–	–	–
servoLagMax1	–	–	–	–	–
servoLagMin2	–	–	–	–	–
servoLagMax2	–	–	–	–	–
feedForwardFactor	–	–	–	–	–
controlOutputLimit	–	–	–	–	–
CfgSpeedControl					
vCtrlPropGain	–	–	–	–	–
All further parameters	–	–	–	–	–
CfgCurrentControl					
All parameters	–	–	–	–	–
CfgControllerTol					
All parameters	–	–	–	–	–
CfgPowerStage					
ampName	–	X	–	X	X
ampPowerSupplyType	–	X	–	X	X
ampBusVoltage	–	X	–	X	–
ampPwmFreq	–	X	–	X	–
ampVoltProtection	–	X	–	X	–
ampReadyWaitTime	–	X	–	X	–
ampAcFailSelection	–	X	–	X	–
ampFactorI2t	–	X	–	–	–

Machine parameters	RESET	Strobe with SYNC_CALC	DRIVE OFF	AXIS DISABLED	REF
powStatusCheckOff	–	X	–	–	–
ampAdditionalInfo	–	–	–	–	–
limitOfDcVoltage	–	–	–	–	–
PowerStage-Table					
AmpPeakCurrent	–	X	X	X	–
AmpNomCurrent	–	X	X	X	–
AmpCurrentSensorVoltage	–	X	X	X	–
CfgServoMotor					
motName	–	X	X	–	X
starDelta	–	X	X	–	–
motEncCheckOff	–	X	–	–	–
motFactorI2t	–	X	–	–	–
motSlipTimeConstant	–	X	X	–	–
motEmergencyStopRamp	–	X	–	–	–
motPbrMax	–	X	–	–	–
motPMax	–	X	–	–	–
motPbrMaxAcFail	–	X	–	–	–
motMMax	–	X	–	–	–
motSupply	–	X	X	–	–
motEncType	–	X	–	X	X
motDir	–	X	–	X	X
motStr	–	X	–	X	X
motTypeOfFieldAdjust	–	–	–	–	–
motFieldAdjustMove	–	–	–	–	–
motPhiRef	–	–	–	–	–
motEncSerialNumber	–	–	–	–	–
motAdditionalInfo	–	–	–	–	–
motSpeedSwitchOver	–	–	–	–	–
motSpeedSwitchBack	–	–	–	–	–
CfgControllerComp					
All parameters	–	–	–	–	–
CfgPositionFilter					
filter1Shape	–	X	–	–	–
filter1LimitFreq	–	X	–	–	–
filter2Shape	–	X	–	–	–
filter2LimitFreq	–	X	–	–	–
manualFilterOrder	–	–	–	–	–
CfgAxisComp					
All parameters	–	–	–	–	–
CfgAxisCoupling					
All parameters	–	–	–	–	–

Machine parameters	RESET	Strobe with SYNC_CALC	DRIVE OFF	AXIS DISABLED	REF
CfgLaAxis					
axTransJerk	–	–	–	–	–
axPathJerk	–	–	–	–	–
axPathJerkHi	–	–	–	–	–
axFilterErrWeight	–	–	–	–	–
CfgAxisHandwheel					
All parameters	–	–	–	–	–
CfgCCAuxil					
All parameters	–	–	–	–	–

Sequence

A parameter set is selected for an axis with a strobe via the PLC program using Module 9434. Here you must indicate the logical axis number (index from CfgAxis\axisList) and the name of the parameter set (key name from Axes\ParameterSets). The strobe can then be acknowledged, and machining continues with the new parameter set.

Module 9434 responds immediately after execution and reports in the result whether the parameter set could be selected (result=0). It may take some time to switch to the selected parameter set. Use Module 9435 to check which parameter set is active.

Please remember:

Note

Module 9434 does **not** switch off the drive controller. You can switch the drive controller off via your PLC program using **PP_AxDriveOnRequest** and **PP_AxPosControlRequest** and reactivate it after the parameter set has been switched.

If the axis is assigned to a machining channel, the execution of an NC program must be synchronized with the module call.

Module 9434 Select parameter set

The module activates the parameter set programmed for the drive motor.

Constraints:

- The PLC program can activate another parameter block even while a PLC positioning movement is being executed.
- A PLC positioning movement can also be started while a new parameter block is being selected via Module 9434. In this case, the PLC program must ensure the safety of the machine. The PLC program must ensure that not other parameter sets containing machine parameters that are not suitable for this drive are selected. Unsuitable parameter sets can cause incorrect positioning!

Call:

PS	B/W/D/K	<>Axis number> Index from **MP_CfgAxes/axisList** If you are using the iTNC-compatible programming interface (API 1.0), you must enter 15 for the spindle.
PS	B/W/D/K	<>Parameter set index> 0: Parameter set index 0 1: Parameter set index 1 Etc.
CM	9434	
PL	B/W/D	<>Result> 0: New parameter set selected. 1: Addressed control loop does not exist. 2: Addressed parameter set does not exist. 3: Module was not executed because the axis is active in an NC program. 4: Module was not executed because another command is being performed for this control loop. 5: The module was not executed because the axis is deactivated.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	No error
	1	Error (see above)

Module 9435 Status of the parameter set of an axis
The module returns the active parameter set and (if it exists) the selected parameter set of the control loop.

Call:

PS	B/W/D/K	<>Axis number> Index from MP_CfgAxes/axisList If you are using the iTNC-compatible programming interface (API 1.0), you must enter 15 for the spindle.
CM	9435	
PL	B/W/D	<>Index of active parameter block>
PL	B/W/D	<>Index of selected parameter block> –1: No selected parameter set present

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	No error
	1	Error, addressed control loop not present (indicated axis does not exist)

4.10 Overview of Machine Parameters

4.10.1 "System" Group

Global basic settings of the control. After each machine parameter you will find the page number with the detailed description:

System
- CfgAxes
  - **axisList**: **386**
  - **spindleIndices**: **687**
  - **specCoordSysList**: **387**
- CfgHardware
  - **hardwareType**: **396**
  - **i32stopsMonitoring**: **644**
  - **i32driveEnabling**
  - **currentControlAdjust**: **745**
  - **maxTouchFeed**
  - **driveOffGroupInput**: **601**
- CfgMachineSimul:
  - **simMode**: **888**
  - **skipReferencing**: **889**
  - **skipEmStopTest**: **889**
- CfgCycleTimes
  - **ipoCycle**: **559**
  - **plcCount**: **1089**
  - **watchdogTime**: **680**
- CfgFilter
  - **typeFilter1**: **544**
  - **orderFilter1**: **544**
  - **typeFilter2**: **544**
  - **orderFilter2**: **544**
- CfgPosCorrection
  - **enable**: **906**
  - **feed**: **906**
- CfgHandwheel
  - **type**: **968**
  - **initValues**: **968**
  - **incrPerRevol**: **969**
  - **rasterPerRevol**: **969**
  - **countDir**: **969**
  - **sensitivity**: **969**
  - **speedFactor**: **970**
  - **feedFactor**: **970**
  - **crossShortSafety**: **971**
- DisplaySettings
  - CfgUnitOfMeasure
    - **unitOfMeasure**: **880**
  - CfgDisplayLanguage
    - **ncLanguage**: **882**
    - **plcDialogLanguage**: **883**
    - **plcErrorLanguage**: **883**
    - **helpLanguage**: **883**
  - CfgOsciColor
    - **background**: **815**
    - **channel1**: **816**

CfgOemBool
[Key name of the user parameter]
value: **1255**
ignorePlc: **1255**
CfgOemInt
[Key name of the user parameter]
value: **1255**
ignorePlc: **1255**
CfgOemString
[Key name of the user parameter]
value: **1255**
ignorePlc: **1255**
CfgOemPosition
[Key name of the user parameter]
value: **1255**
ignorePlc: **1255**
CfgPlcSymName
stoppingAngle: **1035**
maxSpeedSpindle: **1035**
dbLoadDisplay: **1036**
readTsfData: **1036**
displayMode: **1037**
setToolPlace: **1037**
CfgPlcOperTimes
displayPlcTimes: **1008**
resetPlcTimes: **1008**
resetNcTimes: **1008**
textNumber: **1009**
CfgSystemTime
offsetToUTC: **1013**
TableSettings
CfgTableProperties
[Key name of the table type]
columnKeys: **1193**
primaryKey: **1194**
foreignKey: **1194**
modificationKey: **1194**
Columns
[Key name of the table extension]
[Key name of the table column]
CfgColumnDescription
width: **1196**
unit: **1197**
initial: **1197**
minimum: **1198**
Maximum: **1198**
charset: **1198**
unique: **1199**
readonly: **1199**
unitIsInch: **1199**
CfgColumnText
dialogText: **1189**
softkeyIcon: **1189**
iconVariant: **1189**
choice: **1189**
lockValue: **1189**

CfgConfigSettings
- **undoListSize**: **257**
- **suppressUserMsg**: **930**
- **dispParamNumbers**: **257**
- **hideWriteProtected**: **264**

Network
- Serial
  - CfgSerialPorts
    - **activeRs232**: **1363**
    - **interfaceRs232**: **1363**
    - **interfaceRs422**: **1363**
    - **interfacePlc**: **1363**
    - **baudRateLsv2**: **1364**
  - CfgSerialInterface
    - [Key name with properties of the serial port]
      - **baudRate**: **1366**
      - **protocol**: **1366**
      - **dataBits**: **1367**
      - **parity**: **1367**
      - **stopBits**: **1368**
      - **flowControl**: **1370**
      - **fileSystem**: **1370**
      - **bccAvoidCtrlChar**: **1371**
      - **rtsLow**: **1372**
      - **noEotAfterEtx**: **1372**
  - CfgServiceRequest
    - **name**: **1380**
    - **host**: **1381**
    - **port**: **1381**
    - **content**: **1381**
    - **period**: **1381**
    - **timeout**: **1382**
    - **serverIp**: **1382**
    - **serverIpMask**: **1382**

Key code
- CfgOemPassword
  - [Password as key name]
    - **funcList**
- CfgModOemSoftkey
  - [Key name]
    - **activation**
    - **skPos**
    - buttonText
      - **dialogRes**
      - **text**
    - **buttonImage**
    - **funcKey**
    - **helpId**
- CfgCfgEditActivate
  - CONFIGEDIT-PLCUSERPARAMETER
    - **layoutFile**: **282**
    - **dispLangText**: **283**
    - **readOnly**: **282**
  - CONFIGEDIT-USER123
    - **layoutFile**: **282**
    - **dispLangText**: **283**
    - **readOnly**: **282**

4.10.2 "Channels" group

Channel-specific machine parameters. After each machine parameter you will find the page number with the detailed description:

- Channels
  - Kinematics
    - CfgKinComposModel
      - [Key name of kinematics model]
        - **subKinList**: **495**
        - **activeSpindle**: **495**
        - **tiltingAllowed**: **496**
    - CfgKinSimpleModel
      - [Key name of the kinematics chain]
        - **kinObjects**: **494**
    - CfgKinSimpleTrans
      - [Key name of the transformation]
        - **dir**: **493**
        - **val**: **493**
        - **realtimeComp**
    - CfgKinSimpleAxis
      - [Key name of the machine axis]
        - **dir**: **492**
        - **axisRef**: **492**
    - CfgKinAnchor
      - [Key name of the anchor]
        - **kindOfAnchor**: **490**
    - KinematicsByDir
      - CfgKinModel
        - [Key name of kinematics model]
          - **axesToolSide**: **508**
          - **trafoToolSide**: **508**
          - **trafoDirToolSide**: **508**
          - **trafoAngleToolSide**: **509**
          - **toolCoordSys**: **509**
          - **axesWpSide**: **510**
          - **trafoWpSide**: **510**
          - **trafoDirWpSide**: **510**
          - **trafoAngleWpSide**: **510**
          - **machineTableSys**: **511**
          - **activeSpindle**: **511**
      - CfgTrafoByDir
        - [Key name of transformation]
          - **location**: **513**
          - **zDir**: **514**
          - **xDir**: **514**
      - CfgTrafoByAngle
        - [Key name of transformation]
          - **location**: **515**
          - **angleDef**: **515**
          - **angle1**: **515**
          - **angle2**: **515**
          - **angle3**: **515**

4.10.3 "Axes" group

Axis-specific machine parameters. After each machine parameter you will find the page number with the detailed description:

Axes
- CfgProgAxis
  - [Key name of the axis]
    - **axName**: **399**
    - **dir**: **399**
    - **progKind**: **400**
    - **index**
    - **relatedAxis**: **401**
- PhysicalAxis
  - [Key name of the axis]
    - CfgAxis
      - **isAng**: **402**
      - **isModulo**: **403**
      - **restoreModuloCntr**: **403**
      - **isHirth**: **411**
      - **axisHw**: **406**
      - **axisMode**: **407**
      - **testMode**: **407**
      - **parList**: **408**
      - **realAxis**: **408**
      - **noActToNomAtEmSt**: **409**
      - **deactivatedAtStart**: **448**
      - **advancedSettings**: **409**
    - CfgAxisPropKin
      - **specKinCoordSys**: **412**
      - **kindOfRotAxis**: **412**
      - **presetToAlignAxis**: **413**
      - **hasSpecAxisData**: **413**
      - **parAxComp**: **1079**
    - CfgAxisSafety
      - **safe**
      - **axisGroup**
      - **sgSpindle**
      - **spindleStop2ToSOS**
      - **chkStandstill**
      - **positionDiffRun**
      - **positionDiffRef**
      - **positionMatch**
      - **refPosition**
      - **positionDiffNom**
      - **speedDiffNom**
      - **speedLimit**
      - **speedLimitSom2**
      - **speedLimitSom4**
      - **absLimitPos**
      - **absLimitNeg**
      - **positionRangeVmin**
      - **positionOffset**
      - **distLimitJog**
      - **timeLimitStop1**
      - **timeLimitStop2**

4.10.4 "KeySynonym" group

If parameter objects with the same content but different key names are needed, you can define a synonym name. You make this definition under "KeySynonym." After each machine parameter you will find the page number with the detailed description:

KeySynonym
CfgKeySynonym
[Key name = synonym name of the configuration object]
relatedTo: **290**
excludeList: **290**

4.10.5 "Aggregates" group

"Aggregates" contains the machine parameters for configuring tool carriers and tool holders. After each machine parameter you will find the page number with the detailed description:

Aggregates
- General
  - CfgAggregateKeys
    - **toolHolderKeys**: **1027**
    - **tailstockKeys**: **Reserved**
    - **steadyRestKeys**: **Reserved**
    - **caxisKeys**: **1025**
  - CfgGlobalProperties
    - **lifeTime**: **1038**
    - **iStopT**: **1042**
    - **threadDwell**: **1043**
    - **threadLiftOff**: **1044**
    - **threadHandWheelOn**: **1045**
    - **protectionZone**: **1022**
    - **doProgAfterTCall**: **1072**
    - **freezeVconst**: **1024**
- ToolHolder
  - [Key name of the tool carrier]
    - CfgTHDescription
      - **ordinalNr**: **1028**
      - **type**: **1028**
      - **spindleNr**: **1029**
      - **maxSwivelPosition**: **1029**
      - **xDimToSlideRef**: **1029**
      - **zDimToSlideRef**: **1030**
      - **yDimToSlideRef**: **1030**
    - CfgToolMountKeys
      - **toolMountKeys**: **1030**
- ToolMount
  - [Key name of the tool holder]
    - CfgToolMount
      - **mountPosWAPP**: **1031**
      - **freeTnr**: **1032**
      - **distCarrierRefX**: **1032**
      - **distCarrierRefZ**: **1032**
      - **distCarrierRefY**: **1033**
      - **correctionX**: **1033**
      - **correctionZ**: **1033**
      - **correctionY**: **1033**
      - **kinModelToModify**: **501**
      - **kinModel**: **501**
      - **coorTrafoToModify**
      - **coorTrafo**
      - **mirroringAxes**: **1034**
      - **convTblNr**: **1034**

4.10.6 "ProcessingData" group

General settings like setup clearance definitions are entered under ProcessingData. After each machine parameter you will find the page number with the detailed description:

4.11 Parameter Overview Sorted by MP Numbers

Parameter groups

Machine parameters	Group
100000 to 199999	System configuration + miscellaneous
200000 to 299999	Channel-dependent settings
300000 to 399999	Axis configuration
400000 to 499999	Parameter blocks
600000 to 699999	Range of parameters for CNC lathe machining
700000 to 799999	Reserved
800000 to 899999	Reserved
900000 to 999999	Numbers for OEM parameters

Note

The machine parameters available depend on the control model, the scope of function and the NC software. Some of the machine parameters listed below may not be available on your control.

MP number	Configuration object or machine parameter
100000	CfgAxes
100001	axisList
100002	spindleIndices
100003	specCoordSysList
100100	CfgHardware
100101	hardwareType
100102	l32stopsMonitoring
100103	currentControlAdjust
100104	maxTouchFeed
100105	i32driveEnabling
100106	driveOffGroupInput
100200	CfgMachineSimul
100201	simMode
100202	skipReferencing
100203	skipEmStopTest
100300	CfgCycleTimes
100301	ipoCycle
100302	plcCount
100303	watchdogTime
100400	CfgFilter
100401	typeFilter1
100402	orderFilter1
100403	typeFilter2
100404	orderFilter2
100500	CfgPosCorrection
100501	enable
100502	feed
100600	CfgHandwheel
100601	type
100602	initValues
100603	incrPerRevol
100604	rasterPerRevol
100605	countDir
100606	sensitivity
100607	speedFactor

MP number	Configuration object or machine parameter
100608	feedFactor
100609	crossShortSafety
100800	CfgDisplayData
100801	axisDisplayOrder
100802.x	axisDisplayOrderRef
100803	positionWinDisplay
100804	statusWinDisplay
100805	decimalCharacter
100806	axisFeedDisplay
100807	spindleDisplay
100900	CfgStatusAndQPar
100901	clearMode
101000	CfgPosDisplayPace
101001	displayPace
101002	displayPaceInch
101100	CfgUnitOfMeasure
101101	unitOfMeasure
101200	CfgProgramMode
101201	programInputMode
101202	cycleFormat
101300	CfgDisplayLanguage
101301	ncLanguage
101302	plcDialogLanguage
101303	plcErrorLanguage
101304	helpLanguage
101400	CfgOsciColor
101401	background
101402	channel1
101403	channel2
101404	channel3
101405	channel4
101406	channel5
101407	channel6
101408	logicTrace
101409	select
101410	grid
101411	cursorText
101500	CfgStartupData
101501	powerInterruptMsg

MP number	Configuration object or machine parameter
101600	CfgShutDown
101601	shutdownOnConfig
101602	shutdownOnError
101603	shutdownOnUser
101604	shutdownOnOem
101605	maxTermTime
101606	powerOffPort
101607	powerOffDelay
101608	powerOffDevice
101609	powerOffSlot
101700	CfgTable
101701	tableView
101702	enableNotify
101703	dispCompCol
101800	CfgKeyboard
101801	axisKeyI
101802	axisKeyII
101803	axisKeyIII
101804	axisKeyIV
101805	axisKeyV
101900	CfgJhPath
101901	jhTable
101902	jhCycle
101903	jhCycleDataFile
101904	jhCycleTreeFile
101905	sysCycleDataFile
101906	sysCycleTreeFile
101907	dspPath
101908	runtimePath
102000	CfgOemPath
102001	oemTable
102002	dialogTextfile
102003	cycleMainTreeFile
102004	cycleSubTreeFiles
102005	oemCycle
102006	ncDir
102100	CfgComApiPath
102101	errorFilePath
102102	errorFileName
102200	CfgUserPath
102201	ncDir

MP number	Configuration object or machine parameter
102300	CfgPlcPath
102301	mainPgm
102302	pwmPgm
102303	errorTable
102304	errorText
102305	dialog
102306	softkeyProject
102307	compErrorTable
102308	compCfgFile
102309	events
102310	keymapFile
102311	magazineRules
102312	fn14ErrorText
102400	CfgSimModelPath
102401	simModelPath
102500	CfgTablePath
102501	path
102600	CfgSystemCycle
102601	path
102700	CfgOemScript
102701	path
102800	CfgBinFileCache
102801	cachePath
102802	maxFiles
102803	freeSpace
102900	CfgFileType
102901	unitOfMeasure
102902	standardEditor
102903	fileSize
102904	alternateEditor
102905	softkeyIcon
102906	softkeyIconVariant
102907	protect
103000	CfgDisplFileExt
103001	extensions
103100	CfgPathProtection
103101	protection
103200	CfgPlcOptions
103201	trackMState
103202	apiErrorReaction
103203	trackApiState
103204	revokeMState

MP number	Configuration object or machine parameter
103300	CfgPlcOemError
103301	notifyInfo
103302	notifyWarning
103303	notifyError
103400	CfgPlcPeriphery
103401	plNumber
103402	iocProject
103403	iocOption
103404	pt100Discrete
103405	tempCompensation
103406	overrideFullRatio
103407	overrideDelta
103408	overrideIntegDelta
103500	CfgPlcTimer
103501	unit
103502	value
103503	run
103504	start
103505	gate
103600	CfgPlcDeferCutoff
103601	port
103602	time
103700	CfgPlcFastInput
103701	number
103702	significance
103703	operand
103800	CfgPlcTechnology
103801	operand
103900	CfgPlcMStrobe
103901	Min.
103902	Max.
103903	signal
103904	acknowledge
103905	code
103906	data
103907	revoke
103908	split
103909	singular
103910	blockEnd
103911	blockSearch
103912	sync
103913	macro

MP number	Configuration object or machine parameter
103914	track
103915	group
104000	CfgPlcSStrobe
104001	condition
104002	signal
104003	acknowledge
104004	spindleSpeed
104005	badSpeed
104006	spindleMode
104007	gearCode
104008	gearSpeed0
104009	gearSpeed1
104010	gearStop
104011	sCode
104012	revoke
104013	singular
104014	blockSearch
104015	sync
104016	syncGear
104017	type
104018	cuttingSpeed
104100	CfgPlcTStrobe
104101	type
104102	condition
104103	signal
104104	acknowledge
104105	toolNumber
104106	toolIndex
104107	toolMagazine
104108	pocketNumber
104109	unloadTool
104110	externalTool
104111	internalTool
104112	specialTool
104113	revoke
104114	singular
104115	blockSearch
104116	sync
104200	CfgPlcStrobeAlias
104201	type
104202	mCode
104203	mOffset

MP number	Configuration object or machine parameter
104300	CfgPlcOverrideDev
104301	source
104302	mode
104303	values
104304	mop
104400	CfgPlcOverrideS
104401	minimal
104402	maximal
104403	source
104500	CfgOemBool
104501	value
104502	ignorePlc
104600	CfgOemInt
104601	value
104602	ignorePlc
104700	CfgOemPosition
104701	value
104702	ignorePlc
104800	CfgOemReal
104801	value
104900	CfgOemString
104901	value
104902	ignorePlc
105000	CfgPlcOperTimes
105001	displayPlcTimes
105002	resetPlcTimes
105003	resetNcTimes
105004	textNumber
105100	CfgSafety
105101	timeToEmStopTest
105102	timeToAxGrpStop1
105103	timeToSpGrpStop1
105104	timeToAxGrpStop23
105105	timeToSpGrpStop23
105106	specialModeOn
105107	inpNoAxSwitchPos
105108	inpNoAxSwitchNeg
105109	inpNoMachine
105110	inpNoHandwheel
105111	limitFactorSRG
105200	CfgSystemTime
105201	offsetToUTC

MP number	Configuration object or machine parameter
105300	CfgEditorSelect
105301	elementList
105400	CfgEditorSettings
105401	createBackup
105402	deleteBack
105403	cursorAround
105404	lineBreak
105405	stdTNCHelp
105406	toggleCyclDef
105407	warningAtDEL
105408	maxLineGeoSearch
105409	blockIncrement
105410	useProgAxes
105411	enableStraightCut
105500	CfgTableProperties
105501	columnKeys
105502	primaryKey
105503	foreignKey
105504	modificationKey
105600	CfgColumnDescription
105601	width
105602	unit
105603	initial
105604	minimum
105605	maximum
105606	charset
105607	unique
105608	readonly
105609	unitIsInch
105700	CfgColumnText
105701	dialogText
105702	softkeyIcon
105703	iconVariant
105704	choice
105705	lockValue
105800	CfgTableBinding
105801	binds
105900	CfgColumnBinding
105901	column
105902	id
105903	number
105904	index
105905	subindex

MP number	Configuration object or machine parameter
106000	CfgTableFilter
106001	softkeyText
106002	softkeyIcon
106003	iconVariant
106004	helpColumn
106005	helpValue
106006	helpPicture
106007	select
106008	from
106009	option
106010	exclude
106100	CfgTablePrototype
106101	path
106102	enableReset
106200	CfgTableSelect
106201	filter
106202	column
106203	target
106300	CfgConfigDataFiles
106301	oemPath
106302	usrPath
106303	dataFiles
106304	updateFiles
106400	CfgJhConfigDataFiles
106401	oemConfigDataFile
106402	jhDataFiles
106403	jhAttrFiles
106404	jhUpdateFiles
106405	sysFile
106500	CfgConfigSettings
106501	undoListSize
106502	suppressUserMsg
106503	dispParamNumbers
106504	hideWriteProtected
106600	CfgSerialPorts
106601	activeRs232
106602	interfaceRs232
106603	activeRs422
106604	interfaceRs422
106605	interfacePlc
106606	baudRateLsv2

MP number	Configuration object or machine parameter
106700	CfgSerialInterface
106701	baudRate
106702	protocol
106703	dataBits
106704	parity
106705	stopBits
106706	flowControl
106707	fileSystem
106708	bccAvoidCtrlChar
106709	rtsLow
106710	noEotAfterEtx
106800	CfgPassword
106801	funcList
106900	CfgOemPassword
106901	funcList
107000	CfgModSoftkey
107001	activation
107002	skPos
107003	buttonText
107004	buttonImage
107005	funcKey
107100	CfgModOemSoftkey
107101	activation
107102	akPos
107103	buttonText
107104	buttonImage
107105	funcKey
107200	CfgCfgEditActivate
107201	layoutFile
107202	dispLangText
107203	readOnly
107300	CfgPlcVersion
107301	plcVersion
107302	versionText
107400	CfgCompVersion
107401	componentType
107402	componentVersion
107500	CfgNcVersion
107501	ncType
107502	ncVersion

MP number	Configuration object or machine parameter
107600	CfgFirmwareCcu422
107601	path
107602	filenameIdent
107603	filenameBoot
107604	autoFilename
107605	filenameCur
107606	filenameVel
107607	filenameVelSafe
107700	CfgFirmwareCcu424
107701	path
107702	autoFilename
107703	filename
107704	filenameSafe
107800	CfgFirmwareCcu520
107801	client
107802	path
107803	autoFilename
107804	filename
107900	CfgFirmwareCcu524
107901	client
107902	path
107903	autoFilename
107904	filename
108000	CfgTouchProbe
108001	mStrobeUTurn
108002	mStrobePos
108003	posAngle
108004	posTolerance
108005	considerPreset
108006	probingPasses
108007	confidenceRange
108100	CfgProbeTool
108101	orientProperty
108102	orientSpeed
108103	activateType
108104	activateMStrobe
108105	activateTimeout
108106	activateRetries
108107	deactivateType
108108	deactivateMStrobe
108109	deactivateTimeout
108110	deactivateRetries
108111	autoDeactivateTime

MP number	Configuration object or machine parameter
108200	CfgRingGauge
108201	centerCoord
108202	plungingDist
108300	CfgSystemInfo
108301	systemType
108302	simMode
108303	demoVersion
108304	testVersion
108305	configUpdate
108306	startsToSetFcl
108400	CfgLaPathControl
108401	feedHysterAbs
108402	feedHysterRel
108403	aMaxShortfall
108500	CfgLaPolygon
108501	maxPolyAngle
108502	maxPolyTimeFactor
108503	maxPolyRadius
108504	polygonTolFactor
108505	ignoreCornerFilter
108506	raiseCurveFeed
108600	IpoTraceInfo
108601	active
108602	trigger
108603	channel
108604	index
108700	CfgOsci
108701	channelAmount
108702	debug
108703	channelList
108704	signalList
108705	operandList
108706	test
108707	curjmp
108800	CfgAuxil
108801	serviceFileCntrMax
108802	errorList
108803	ipoKey
108804	ipoDebugPring
108805	debugChannel
108806	overrideForMeasure
108807	clearIpoSRAM
108808	ipoDebugPrint2

MP number	Configuration object or machine parameter
108900	CfgSimulation
108901	autoOn
108902	digitalOverride
108903	numCCU
108904	hardwareType
108905	plcInputSet
108906	plcInputReset
108907	plcInputEmStop
108908	plcInputNcStart
108909	plcInputNcStop
108910	plcInputRapid
108911	plcInputSpStart
108912	plcInputStStop
108913	plcInputSpTipp
108914	plcInput1Plus
108915	plcInput1Minus
108916	plcInput2Plus
108917	plcInput2Minus
108918	plcInput3Plus
108919	plcInput3Minus
108920	plcInput4Plus
108921	plcInput4Minus
108922	plcInput5Plus
108923	plcInput5Minus
108924	plcInputME
108925	plcInputRef
108926	axisPT1time
108927	plcInputSpindleM3
108928	plcInputSpindleM4
108929	plcInputSpindleM5
109000	CfgEndatSimulation
109001	resolution
109002	absValue
109003	error
109004	iresolution
109005	serialNumber
109006	multiturn
109007	linear
109100	CfgNcPgmSettings
109101	dwellStopAtNcStop
109102	fn16MaxScreenOutp
109103	fn16MaxLogfileEnt

MP number	Configuration object or machine parameter
109104	fn16ScreenfilePath
109200	cfgTNCOptConfig
109201	pathTuningPrograms
109202	prgExtensionDinIso
109203	prgDinIsoLanguage
109204	prgHHPlainLanguage
109205	prgSingleFolder
109206	prgDinIsoDialect
109207	tuningControlAdapt
109300	CfgSysDataColumn
109301	column
109302	number
109400	CfgSysDataTable
109401	whereTag
109402	columns
109500	CfgKeySynonym
109501	relatedTo
109502	excludeList
109600	CfgChannelGroup
109601	channelList
109602	omgList
109700	CfgOmgSettings
109701	channelList
109702	errorBehaviour
109800	CfgRootKeys
109801	ncKey
109802	simKey
109900	CfgNcPgmParameters
109901	numberOfReals
109902	realDefaultZero
109903	numberOfStrings
109904	stringDefaultEmpty
110000	CfgProbeToolList
110001	probeList
110100	C50Base4X
110101	data
110200	G50BaseSPI
110201	data
110300	G50Referencing4X
110301	data
110400	G50ReferencingSPI
110401	data

MP number	Configuration object or machine parameter
110500	G50Measure4X
110501	data
110600	G50MeasureSPI
110601	data
110700	G50Endat4X
110701	data
110800	G50EndatSPI
110801	data
110901	CfgNestingLevels
110902	programCalls
110903	macroCalls
111000	CfgPlcPositions
111001	value
111100	CfgSysPaths
111101	sysTable
111102	sysGfun
111103	sysGfunCycle
111104	sysPalletCycle
111105	sysSystemCycle
111106	sysToolCycle
111200	CfgOemPaths
111201	oemTable
111202	oemGfunCycle
111203	oemPalletCycle
111204	oemPlcCycle, oemMfunCycle
111205	oemSystemCycle
111206	oemToolCycle
111300	CfgUserPaths
111301	userCycle
111302	userKey
111400	CfgPalletCycle
111401	path
111500	CfgToolCycle
111501	path
111600	CfgSysNcPgmParameters
111601	numberOfReals
111602	numberOfStrings
111700	CfgOemNcPgmParameters
111701	numberOfReals
111702	numberOfStrings
111800	CfgUserNcPgmParameters
111801	numberOfReals
111802	numberOfStrings

MP number	Configuration object or machine parameter
111900	CfgNcContourLabels
111901	numberOfLabels
112000	CfgSysCycleOptions
112001	hideGfunMacro
112002	hidePalletMacro
112003	hideSystemMacro
112004	hideToolMacro
112005	hideNcPgmParameter
112100	CfgOemCycleOptions
112101	hideGfunMacro
112102	hidePalletMacro
112103	hidePlcMacro, hideMfunMacro
112104	hideSystemMacro
112105	hideToolMacro
112106	hideNcPgmParameter
112200	CfgUserCycleOptions
112201	hideMacro
112300	CfgMachinePicture
112301	path
112302	picture
112400	CfgZeroOffsetSettings
112401	useExtended
112500	CfgFirmwareCbe
112501	path
112502	autoFilename
112503	filename
112600	CfgProfiNetPrjdata
112601	profiNetActive
112602	path
112603	prjDataType
112700	CfgMandPrjData
112701	prjDataNumber
112702	filename
112800	CfgOptPrjData
112801	prjDataNumber
112802	filename
112900	CfgPNStation
112901	logStationAddress
112902	type
113000	CfgPNSlot
113001	logStationAddress
113002	slot

MP number	Configuration object or machine parameter
113003	telegramType
113004	subSlotTelegram
113005	refSpeed
113006	station
113100	CfgDashboardElemnt
113101	dashboardpicType
113102	attribute
113103	entityList
113200	CfgDashboard
113201	Array (e.g. [0]: DB_X1_POS)
113201	dashboardpicType
113202	attrib
113203	entityList
113300	CfgOemGuiColor
113301	color
113400	CfgSysGuiColor
113401	color
113500	CfgSimPosition
113501	value
113600	CfgThreadSpindle
113601	thrdWaitingTime
113602	thrdPreSwitchTime
113700	CfgPNPrj
113701	profiNetActive
113702	path
113703	prjDataType
113704	mandatoryPrjData
113800	CfgPNPrjData
113801	prjDataNumber
113802	filename
113900	CfgPNDrive
113901	ctrlTelegramType
113902	subSlotTelegram
113903	refSpeed
114000	CfgPNAxis
114001	drive
114002	acitvePosEncoder
114003	driveDataSet
114004	commandDataSet
114100	CfgToolMeasurement
114101	stylusType
114102	spindleOrientMode

MP number	Configuration object or machine parameter
114103	probingDirRadial
114104	offsetToolAxis
114105	rapidFeed
114106	probingFeed
114107	probingFeedCalc
114108	spindleSpeedCalc
114109	maxPeriphSpeedMeas
114110	maxPeriphSpeedEdge
114111	maxSpeed
114112	measureTolerance1
114113	measureTolerance2
114114	stopOnCheck
114115	stopOnMeasurement
114116	adaptToolTable
114200	CfgTTRoundStylus
114201	centerPos
114202	stylusDimension
114203	safetyDistToolAx
114204	safetyDistStylus
114300	CfgTTRectStylus
114301	dimX
114302	dimY
114303	dimZ
114304	posXminus
114305	posXplus
114306	posYminus
114307	posYplus
114308	posZminus
114309	posZplus
114310	safetyDistX
114311	safetyDistY
114312	safetyDistZ
114400	CfgObsDiskWatch
114401	warnLevels
114402	scanRate
114403	osType
114500	CfgObsMemWatch
114501	warnLevels
114502	scanRate
114503	startBurstDuration
114504	osType
114505	timedCommands

MP number	Configuration object or machine parameter
114600	CfgServiceRequest
114601	name
114602	host
114603	port
114604	content
114605	period
114606	timeout
114607	serverIp
114608	serverIpMask
114700	CfgPresetView
114701	firstViewFilter
114702	secondViewFilter
114800	CfgSimGeneral
114801	restartAtM99
114802	pathDelay
114900	CfgCoordSystem
114901	coordSystem
115000	CfgTimeDetGeneral
115001	toolChangeTime
115002	gearShiftingTime
115003	mFunTimeAllow
115100	CfgTimeDetMfun
115101	mFun
115102	timeAllow
115200	CfgSimWindowSize
115201	zeroPosX
115202	zeroPosZ
115203	deltaX
115204	deltaZ
115300	CfgSimBlank
115301	outsideDiameter
115302	blankLength
115303	rightBlankEdge
115304	insideDiameter
115400	CfgFunKeys
115401	mFunTimeKeys
115500	CfgConfigVersion
115501	configVersion
115502	versionText
115600	CfgGeneratedMFct
115601	MDCoolantOn
115602	MCoolantOff

MP number	Configuration object or machine parameter
115700	CfgVncServer
115701	active
115702	alwaysActive
115800	CfgSystBckFiles
115801	path
116100	CfgPlcSymName
116101	stoppingAngle
116102	maxSpeedSpindle
116103	dbLoadDisplay
116104	readTsfData
116105	displayMode
116106	setToolPlace

Note

The machine parameters available depend on the control model, the scope of function and the NC software. Some of the machine parameters listed below may not be available on your control.

MP number	Configuration object or machine parameter
200000	CfgKinModel
200001	axesToolSide
200002	trafoToolSide
200003	trafoDirToolSide
200004	trafoAngleToolSide
200005	toolCoordSys
200006	axesWpSide
200007	trafoWpSide
200008	trafoDirWpSide
200009	trafoAngleWpSide
200010	machineTableSys
200011	activeSpindle
200100	CfgTrafoByDir
200101	location
200102	zDir
200103	xDir
200200	CfgTrafoByAngle
200201	location
200202	angleDef
200203	angle1
200204	angle2
200205	angle3
200300	CfgChannelAxes
200301	progAxis
200302	grindAxis
200303	refAxis
200304	refAllAxes
200305	restoreAxis
200306	kinModels
200307	deactFastClamping
200400	CfgChannelFile
200401	geoChainInit
200402	geoIniProgram
200403	geoIniBlock
200404	geoIniCycle

MP number	Configuration object or machine parameter
200405	geoCycleEnd
200406	geoCancelCycle
200407	geoTCallCycPath
200408	geoTDefCycPath
200409	geoAutoTCallCycle
200410	geoPalletCtrlCycle
200411	plcSetPresetCycle
200412	progSelectCycle
200413	afterMdiCycle
200500	CfgSqlProperties
200501	tables
200502	bindings
200600	CfgNcErrorReaction
200601	warningLevel
200700	CfgNcPgmParState
200701	persistent
200702	currentSet
200800	CfgNcPgmBehaviour
200801	operatingTimeReset
200900	CfgGeoTolerance
200901	circleDeviation
201000	CfgGeoCycle
201001	pocketOverlap
201002	displaySpindleErr
201003	displayDepthErr
201004	apprDepCylWall
201005	mStrobeOrient
201100	CfgStretchFilter
201101	filterType
201102	tolerance
201103	maxLength
201200	CfgRotWorkPlane
201201	rotateWorkPlane
201202	planeOrientation
201204	autoMoveAxes
201206	rotPreference
201300	CfgTCPM
201301	tolerance
201302	toolRefPoint
201400	CfgLiftOff
201401	on
201402	distance

MP number	Configuration object or machine parameter
201500	CfgLaPath
201501	minPathFeed
201502	minCornerFeed
201503	maxG1Feed
201504	maxPathJerk
201505	maxPathJerkHi
201506	pathTolerance
201507	pathToleranceHi
201508	maxPathYank
201509	curveTolFactor
201510	curveJerkFactor
201511	angleTolerance
201512	angleToleranceHi
201600	CfgPlcStrobes
201601	mStrobes
201602	sStrobes
201603	tStrobes
201604	aliasStrobes
201605	unitOfMeasure
201700	CfgPlcToolChange
201701	squT0Text
201702	squT0Tint
201703	squT0TintS
201704	sequTextT0
201705	sequTextText
201706	sequTextTint
201707	sequTextTintS
201708	sequTintT0
201709	sequTintT0S
201710	sequTintText
201711	sequTintTextS
201712	sequTintTint
201713	sequTintTintS
201714	sequTintTintF
201715	sequTintTintSF
201716	followUpS
201717	followUpT
201718	updateTable
201719	sequText
201720	sequTint
201800	CfgPlcBlockScan
201801	mFirst
201802	mLast

MP number	Configuration object or machine parameter
201900	CfgPlcOverrideF
201901	minimal
201902	maximal
201903	source
202000	CfgPlcOverrideR
202001	minimal
202002	maximal
202003	source
202100	CfgPalletBehaviour
202101	stopAt
202102	editTableWhileRun
202200	CfgToolBehaviour
202201	varPocketCoding
202202	toolDefMode
202203	cleanToolTblAtRun
202204	modifyToolTblFrom
202205	modifyToolTblTo
202300	CfgPrefForPolarKin
202301	kindOfPref
202400	CfgChannelType
202401	type
202500	CfgChannelSysData
202501	id50Table
202502	id50Columns
202503	id51Table
202504	id51Columns
202505	id52Table
202506	id52Columns
202507	id500Table
202508	id500Columns
202509	basistrafoTable
202510	basistrafoColumns
202600	CfgKinSimpleTrans
202601	dir
202602	val
202603	realtimeComp
202700	CfgKinSimpleAxis
202701	dir
202702	axisRef
202800	CfgKinSimpleModel
202801	kinObjects

MP number	Configuration object or machine parameter
202900	CfgKinComposModel
202901	subKinList
202902	activeSpindle
202904	tiltingAllowed
203000	CfgKinList
203001	kinCompositeModels
203100	CfgScalingOptions
203101	dimension
203102	mode
203200	Cfg2DRadiusCorrectionSettings
203201	thresholdAngle
203202	intermediateCircleTolerance
203300	CfgKinematicCorrections
203301	toolSide
203400	CfgChannDashboard
203401	defaultDashboard
203402	opmodeStartup
203403	opmodeReference
203404	opmodeManual
203405	opmodeMDI
203406	opmodeAutomatic
203407	opmodeSGTest
203408	opmodeManualLarge
203409	opmodeAutomaticLarge
203500	CfgKinematicalOffset
203501	placeOfModification
203502	location
203600	CfgKinSimplFixture
203601	kindOfFixture
203700	CfgKinAnchor
203701	kindOfAnchor
203800	CfgChannelProperties
203801	slideType
203802	posCrossSlide
203803	assCoolantCirc
203804	kinManualMode
203900	CfgAssignAggregate
203901	assignToolHolder
204000	CfgActivateKinem
204001	kinemToActivate

4.11.3 Axis-specific parameters

Note

The machine parameters available depend on the control model, the scope of function and the NC software. Some of the machine parameters listed below may not be available on your control.

MP number	Configuration object or machine parameter
300000	CfgProgAxis
300001	axName
300002	dir
300003	progKind
300004	index
300005	relatedAxis
300100	CfgAxis
300101	isAng
300102	isModulo
300103	isHirth
300104	axisHw
300105	axisMode
300106	testMode
300107	parList
300108	realAxis
300109	noActToNomAtEmSt
300110	deactivatedAtStart
300111	restoreModuloCntr
300112	advancedSettings
300200	CfgAxisPropKind
300201	specKinCoordSys
300202	kindOfRotAxis
300203	presetToAlignAxis
300204	hasSpecAxisData
300205	parAxComp
300300	CfgAxisSafety
300301	safe
300302	sgSpindle
300303	chkStandstill
300304	positionDiffRun
300305	positionDiffRef
300306	positionMatch
300307	speedLimit
300308	absLimitPos
300309	absLimitNeg
300310	positionRangeVmin

MP number	Configuration object or machine parameter
300400	CfgRollOver
300401	shortestDistance
300402	startPosToModulo
300403	showModuloDisp
300500	CfgMachDatumExtra
300501	distFromMachDatum
300600	CfgSpindleProperties
300601	spindleTyp
300602	gearStages
300603	assCoolantCirc
300700	CfgMachineTable
300701	sys
300702	basisTrans
300703	sysKinSimple
300704	basisTransKinSim
300800	CfgCAxisProperties
300801	blockBrake
300802	spindlePrePosit
300803	relatedWpSpindle
300900	CfgAxisProperties
300901	diameterRadiusProg
300902	threadSafetyDist
301000	CfgProtectionZone
301001	limitPositive
301002	limitNegative

4.11.4 Parameters for configuring the parameter sets

Note

The machine parameters available depend on the control model, the scope of function and the NC software. Some of the machine parameters listed below may not be available on your control.

MP number	Configuration object or machine parameter
400000	CfgAxisHardware
400001	signCorrActualVal
400002	signCorrNominalVal
400003	posEncoderType
400004	distPerMotorTurn
400005	posEncoderDist
400006	posEncoderIncr
400007	posEncoderRefDist
400008	posEncoderInput
400009	posEncoderSignal
400010	posEncoderFreq
400011	posEncoderResistor
400012	speedEncoderInput
400013	pwmSignalOutput
400014	hsciCcIndex
400015	driveOffGroup
400016	checkPhiFieldRef
400017	genExtIntPolFactor
400018	ctrlPerformance
400100	CfgAxisAnalog
400101	analogOutput
400102	analogOffset
400103	kvFactor2
400104	kvSpeedLimit
400105	maxFeedAt9V
400106	accForwardFactor
400107	compStrength
400108	compWidth
400109	compTimeOffset
400110	compFFAdjust
400111	compRefAcc
400112	noOffsetAdjust
400113	unipolar
400114	compLimitFactor

MP number	Configuration object or machine parameter
400200	CfgAxisHandwheel
400201	input
400202	countDir
400203	distPerRevol
400204	incPerRevol
400205	rasterPerRevol
400206	encoderSignal
400207	encoderFreq
400208	encoderResistor
400209	decToSoftLimit
400210	hsciCcIndex
400300	CfgFeedLimits
400301	minFeed
400302	maxFeed
400303	rapidFeed
400304	manualFeed
400305	maxAcceleration
400307	M19MaxSpeed
400308	nominalSpeed
400309	restoreFeed
400310	M19NcSpeed
400311	maxAccSpeedCtrl
400312	maxDecSpeedCtrl
400400	CfgReferencing
400401	refType
400402	endatSerial
400403	refPosition
400404	refSwitchActive
400405	endatDiff
400406	refFeedLow
400407	refFeedHigh
400408	refDirection
400409	moveAfterRef
400410	moveAfterRefType
400411	moveAfterRefPos
400412	moveAfterRefFeed
400413	externRefPulse
400500	CfgPositionLimits
400501	swLimitSwitchPos
400502	swLimitSwitchNeg
400503	lubricationDist

MP number	Configuration object or machine parameter
400600	CfgControllerAuxil
400601	driveOffLagMonitor
400602	checkPosStandstill
400603	checkPosDiff
400604	posDiffCountDir
400605	maxPosDiff
400700	CfgEncoderMonitor
400701	checkAbsolutPos
400702	checkSignalLevel
400703	checkFrequency
400704	checkRefDistance
400705	movementThreshold
400706	thresholdDistance
400800	CfgPosControl
400801	kvFactor
400802	servoLagMin1
400803	servoLagMax1
400804	servoLagMin2
400805	servoLagMax2
400806	feedForwardFactor
400807	controlOutputLimit
400808	posCtrlSLPropGainF
400900	CfgSpeedControl
400901	vCtrlPropGain
400902	vCtrlIntGain
400903	vCtrlIntTime
400904	vCtrlDiffGain
400905	vCtrlFiltLowPassT
400906	vCtrlEncInputFilt
400907	vCtrlFiltType1
400908	vCtrlFiltFreq1
400909	vCtrlFiltBandWidth1
400910	vCtrlFiltDamping1
400911	vCtrlFiltType2
400912	vCtrlFiltFreq2
400913	vCtrlFiltBandWidth2
400914	vCtrlFiltDamping2
400915	vCtrlFiltType3
400916	vCtrlFiltFreq3
400917	vCtrlFiltBandWidth3
400918	vCtrlFiltDamping3
400919	vCtrlFiltType4

MP number	Configuration object or machine parameter
400920	vCtrlFiltFreq4
400921	vCtrlFiltBandWidth4
400922	vCtrlFiltDamping4
400923	vCtrlFiltType5
400924	vCtrlFiltFreq5
400925	vCtrlFiltBandWidth5
400926	vCtrlFiltDamping5
400927	vCtrlSwitchOffDelay
400928	vCtrlTimeSwitchOff
401000	CfgCurrentControl
401001	iCtrlPropGain
401002	iCtrlIntGain
401003	iCtrlPwmType
401004	iCtrlPwmInfo
401005	iCtrlMotVRedFact
401006	iCtrlMotVRedSpeed
401007	iCtrlDiffFreqFF
401008	iCtrlAddInfo
401100	CfgControllerTol
401101	posTolerance
401102	timePosOK
401103	speedTolerance
401104	timeSpeedOK
401105	syncTolerance
401106	timeSyncOK
401200	CfgPowerStage
401201	ampName
401202	ampPowerSupplyType
401203	ampBusVoltage
401204	ampPwmFreq
401205	ampVoltProtection
401206	ampReadyWaitTime
401207	ampAcFailSelection
401208	ampFactorI2t
401209	powStatusCheckOff
401210	ampAdditionalInfo
401211	limitOfDcVoltage
401300	CfgServoMotor
401301	motName
401302	starDelta
401303	motEncCheckOff
401304	motFactorI2t

MP number	Configuration object or machine parameter
401305	motSlipTimeConstant
401306	motEmergencyStopRamp
401307	motPbrMax
401308	motPMax
401309	motPbrMaxAcFail
401310	motMMax
401311	motEncType
401312	motDir
401313	motStr
401314	motTypeOfFieldAdjust
401315	motFieldAdjustMove
401316	motPhiRef
401317	motEncSerialNumber
401318	motAdditionalInfo
401319	motSpeedSwitchOver
401320	motSpeedSwitchBack
401321	motSupply
401400	CfgControllerComp
401401	compFriction0
401402	compFrictionT1
401403	compFrictionT2
401404	compFrictionNS
401405	compCurrentOffset
401406	compAcc
401407	compIpcT1
401408	compIpcT2
401409	compIpcJerkFact
401410	compActDampFact
401411	compActDampTime
401412	compTorqueRipple
401413	compTorsionFact
401414	compSwitchOff
401500	CfgSpindle
401501	fastInputType
401502	fastInput
401503	zeroPosEdge
401504	stopOnSwitchSpeed
401505	gearShiftSpeed
401506	gearShiftVoltage
401509	changeTurnDir

MP number	Configuration object or machine parameter
401600	CfgPositionFilter
401601	filter1Shape
401602	filter1LimitFreq
401603	filter2Shape
401604	filter2LimitFreq
401605	manualFilterOrder
401700	CfgLaAxis
401701	axTransJerk
401702	axFilterErrWeight
401703	axPathJerk
401704	axPathJerkHi
401800	CfgAxisComp
401801	active
401802	backLash
401803	linearCompValue
401804	compType
401805	filterTime
401806	posCtrlRevErr
401807	posCtrlRevErrTime
401900	CfgDefaultAnalogParams
401901	signCorrActualVal
401902	signCorrNominalVal
401903	kvFactor
401904	servoLagMin1
401905	servoLagMax1
401906	servoLagMin2
401907	servoLagMax2
401908	feedForwardFactor
401909	controlOutputLimit
401910	maxAcceleration
401911	driveOffLagMonitor
401912	checkPosStandstill
401913	checkAbsolutPos
401914	checkSignalLevel
401915	movementThreshold
401916	posTolerance
401917	filter1Shape
401918	filter2Shape
401919	manualFilterOrder
401920	maxFeedAt9V
401921	accForwardFactor

MP number	Configuration object or machine parameter
402000	CfgSyncAxis
402001	syncAxisCoupleAxis
402002	syncAxisRefSync
402003	syncAxisTorqueBias
402004	syncAxisPropGain
402005	syncAxisTorqueDist
402006	syncAxisCorrect
402007	syncAxisInvVeloc
402100	CfgAxisProfiNet
402101	driveUnit
402102	driveObject
402200	CfgCCAuxil
402201	miscCtrlFunct0
402202	miscCtrlFunct1
402300	CfgAxisCoupling
402301	masterAxis
402302	mode
402303	type
402304	typeOfOffset
402305	posOffset
402306	offsetFeed
402307	maxPosDiff
402308	ultimatePosDiff
402309	scalingFactor
402310	torqueBias
402311	propGain
402312	torqueDistrFactor
402313	speedCorrectRatio
402314	inverseVelocity
402315	accFilterTime

4.11.5 Parameters for configuring tool carriers and tool holders

Note

The machine parameters available depend on the control model, the scope of function and the NC software. Some of the machine parameters listed below may not be available on your control.

MP number	Configuration object or machine parameter
600000	CfgAggregateKeys
600001	toolHolderKeys
600002	tailstockKeys
600003	steadyRestKeys
600004	caxisKeys
600100	CfgToolMountKeys
600101	toolMountKeys
600200	CfgTHDescription
600203	ordinalNr
600204	type
600207	spindleNr
600208	maxSwivelPosition
600210	xDimToSlideRef
600211	zDimToSlideRef
600212	yDimToSlideRef
600400	CfgToolMount
600401	mountPosWAPP
600402	freeTnr
600407	distCarrierRefX
600408	distCarrierRefZ
600409	distCarrierRefY
600410	correctionX
600411	correctionZ
600412	correctionY
600414	coorTrafoToModify
600415	coorTrafo
600416	mirroringAxes
600417	convTblNr
600418	kinModelToModify
600419	kinModel

MP number	Configuration object or machine parameter
601800	CfgGlobalProperties
601801	lifeTime
601802	iStopT
601803	threadDwell
601804	threadLiftOff
601805	protectionZone
601806	doProgAfterTCall
601807	threadHandWheelOn
601808	freezeVconst

Note

The machine parameters available depend on the control model, the scope of function and the NC software. Some of the machine parameters listed below may not be available on your control.

MP number	Configuration object or machine parameter
602000	CfgGlobalTechPara
602005	safetyDistBlankOut
602006	safetyDistBlankIn
602007	safetyDistWorkpOut
602008	safetyDistWorkpIn
602009	DefaultG14
602010	DefaultCLT
602011	DefaultG60
602012	DefGlobG47P
602013	DefGlobG147SCI
602014	DefGlobG147SCK
602015	DefGlobOverMeasI
602016	DefGlobOverMeasK
604600	CfgToolMeasuring
604601	measuringType
604602	feed
604603	distance
604700	CfgProbePosition
604701	positionProbePos
604702	positionProbeNeg
604703	maxMeasuringFeed
604800	CfgGlbDispSettings
604801	plcSpindleSelect
604802	plcChannelSelect
604803	axesDisplayMode
604900	CfgMMISettings
604901	extManualMode
604902	extProgramMode

5 Modules and PLC Operands

5.1 Overview of Modules

Module	Function	SW version	Page
9000/ 9001	Copy in the marker or word range	597 110-01	1342
9002	Read the inputs of a PLC input/output unit	597 110-01	999
9003	Transfer the analog input of the MC	597 110-01	1002
9004	Read the edges of PLC inputs	597 110-01	1000
9005	Set the outputs of PLC input/output unit	597 110-01	1001
9006	Set and start PLC timer	597 110-01	1132
9007	Read the diagnostic information of a PLC input/output unit	597 110-01	1391
9010/ 9011/ 9012	Read in the word range	597 110-01	1343
9019	Size of the processing stack	597 110-01	1257
9020/ 9021/ 9022	Write in the word range	597 110-01	1344
9025	Write as BCD code	597 110-01	–
9030	Read machine parameter for Hirth coupling	597 110-01	–
9034	Load a machine parameter subfile	597 110-01	297
9035	Read NC status information ■ Function 9: Read assigned handwheel axis ■ Function 26: Read jog increment	597 110-01	979
9036	Write NC status information ■ Function 6: Select handwheel axis ■ Function 7: Set handwheel transmission ratio ■ Function 10: Limit value for jog increment	597 110-01	978
9037	Read the safety-oriented status	597 110-01	–
9038	Read the status information of axes	597 110-01	439
9040	Read axis coordinates (format 0.001 mm)	597 110-02	441
9041	Read axis coordinates (format 0.0001 mm)	597 110-01	442
9049	Read position value and speed value of an axis	597 110-03	440
9050	Number conversion binary –> decimal	597 110-01	1345
9051	Number conversion binary –> decimal (format)	597 110-01	1346
9052	Convert decimal string to decimal number with an exponent	597 110-01	1347
9053	Conversion binary –> ASCII/hexadecimal	597 110-01	1347
9054	Conversion ASCII/hexadecimal –> binary	597 110-01	1348
9055	Convert time (binary) to formatted string	597 110-01	1014
9065	Status of the commissioning function	597 110-05	–
9066	Status of HEIDENHAIN hardware	597 110-04	675
9067	Status of HEIDENHAIN software	597 110-04	676

Module	Function	SW version	Page
9070	Copy a number from a string	597 110-01	1320
9071	Find the string length	597 110-01	1321
9072	Copy a byte block into a string	597 110-01	1321
9073	Copy a string into a byte block	597 110-01	1322
9084	Display PLC error messages with additional data	597 110-01	922
9085	Display PLC error messages	597 110-01	923
9086	Delete PLC error message	597 110-01	924
9087	Status of PLC error message	597 110-01	925
9095	Activate axis-error compensation	597 110-01	480
9100	Assign data interface	597 110-01	1385
9101	Release data interface	597 110-01	1385
9102	Status of data interface	597 110-01	1386
9103	Transmit string through data interface	597 110-01	1387
9104	Receive string through data interface	597 110-01	1388
9105	Transmit binary data through data interface	597 110-01	1389
9106	Receive binary data through data interface	597 110-01	1390
9107	Binary data from receive buffer	597 110-01	1391
9110	Transmit a message by LSV2	597 110-01	
9111	Receive a message via LSV2	597 110-01	1392
9112	Transmit ASCII characters via data interface	597 110-01	1393
9113	Receive ASCII characters via data interface	597 110-01	1394
9117	Reset a BALLUFF Identification system BIS C-6002	597 110-04	–
9118	Read and implement data from a BALLUFF identification system BIS C-6002	597 110-04	–
9119	BALLUFF BIS C-6002 evaluation unit: Write tool data to a data carrier	597 110-04	–
9120	Position PLC axis	597 110-01	457
9121	Stop PLC axis	597 110-01	458
9122	Status of PLC axis	597 110-01	458
9123	Traverse the reference marks of PLC axes	597 110-01	459
9124	Feed rate override for PLC axis	597 110-01	460
9128	Torque limiting by PLC (in mA or %)	597 110-04	616
9129	Status of torque limiting by the PLC	597 110-04	617
9133	Download the internal ADCs	597 110-01	659
9137	Read diagnostic information of the PL 510	597 110-01	995
9138	Read analog input of the PL 510	597 110-01	1004
9139	Monitoring functions for the PL 510 PLC input/output units	597 110-01	997
9142	Position preset for programmed axis	597 110-04	400
9144	Configuration of the emergency stop test	597 110-04	684
9145	Actual-to-nominal value transfer	597 110-01	572
9146	Saving and restoring actual position values	597 110-04	–
9147	Preset reference value	597 110-02	–
9149	Read or set the commutation angle of an axis	597 110-05	–

Module	Function	SW version	Page
9155	Switch axes from closed-loop to open-loop condition	597 110-04	451
9156	Switch axes from open-loop to closed-loop condition	597 110-04	452
9157	Transfer the status of the drive controller	597 110-01	603
9158	Torque limiting by PLC (in mA)	597 110-04	617
9159	200 ms early warning for standstill of the drives	597 110-01	603
9160	Interrogate the status of temperature and I2t monitoring	597 110-01	667
9161	Enable the drive controller (current controller)	597 110-01	604
9162	Interrogate the status of the drive controllers (speed controllers)	597 110-01	604
9163	Switch between wye/delta	597 110-01	707
9164	Read the actual speed value of a motor	597 110-01	576
9165	Sample the current motor temperature	597 110-01	660
9166	Read momentary utilization of drive motor	597 110-01	674
9167	Supply voltage monitoring	597 110-01	658
9168	Read commissioning status	597 110-05	–
9169	Axis-specific input "speed 0"	597 110-03	604
9171	Spindle orientation (when using the symbolic API, use **Module 9414**)	597 110-01	697
9173	Activate speed-dependent monitoring of the wye/delta switchover	597 110-01	635
9174	Read momentary spindle status in reference to the wye/ delta switchover	597 110-01	635
9180	Keystroke simulation	597 110-01	961
9181	Disable NC key by PLC	597 110-01	961
9182	Re-enable NC key by PLC	597 110-01	962
9183	Disable NC key groups by PLC	597 110-01	962
9184	Re-enable groups of NC keys by PLC	597 110-01	963
9189	Shut down the control	597 110-01	958
9190	Start the PLC operating hours counter	597 110-01	1010
9191	Stop the PLC operating hours counter	597 110-01	1010
9192	Transfer the operating hours counter	597 110-01	1011
9193	Set the operating hours counter	597 110-01	1011
9194	Alarm when operating time exceeded	597 110-01	1012
9195	Transfer the real-time clock	597 110-01	1013
9196	Find the PLC cycle time	597 110-01	1089
9197	Start cycle timers	597 110-01	1133
9203	Activate the resource-based soft-key structure	597 110-02	–
9204	Refresh the soft key menu	597 110-02	–
9208	Request the status information of PLC soft keys	597 110-02	–
9220	Traverse the reference mark	597 110-01	520
9221	Start a PLC positioning movement	597 110-01	462
9222	Request the status of a PLC positioning movement	597 110-01	463
9224	Stop PLC positioning movements	597 110-01	464

Module	Function	SW version	Page
9226	Preset the status of an axis	597 110-03	–
9227	Position auxiliary axes and NC axes	597 110-04	464
9231	Compensation of thermal expansion	597 110-01	483
9240	Open a file	597 110-01	1204
9241	Close a file	597 110-01	1205
9242	Positioning in a file	597 110-01	1206
9243	Read from a file line by line	597 110-01	1207
9244	Write to a file line by line	597 110-01	1208
9245	Read a field from a table	597 110-01	1209
9246	Write to a field in a table	597 110-01	1210
9247	Search for a condition in a table	597 110-01	1211
9248	Copy, rename or delete file	597 110-01	–
9249	Transfer BS variable "errno"	597 110-01	1212
9250	Start table editor by PLC	597 110-05	1213
9251	Stop table editor by PLC	597 110-05	1214
9252	Positioning the cursor in the table editor	597 110-05	1215
9255	Read a field from a table	597 110-01	1216
9256	Write to a field in a table	597 110-01	1217
9260	Receive events, wait for events	597 110-01	1328
9261	Send events	597 110-01	1330
9262	Context change between spawn processes	597 110-01	1331
9263	Interrupting a spawn process for a defined time	597 110-01	1331
9264	Wait for a condition	597 110-01	1332
9270	Read an entry from the config object CfgOemString	597 110-02	–
9271	Write an entry to the config object CfgOemString	597 110-02	–
9275	Write ASCII data into the log	597 110-01	926
9276	Write operand contents into the log	597 110-01	927
9277	Write strings from a cyclic PLC program in one's own log	597 110-01	–
9279	Shutdown by PLC	597 110-01	959
9285	Set OEM access rights	597 110-04	892
9321	Find the current block number	597 110-01	899
9322	Request information of the current NC program	597 110-02	899
9330	Read the status of an OEM application	597 110-01	–
9331	Send data to the OEM application	597 110-01	–
9332	Receive data from the OEM application	597 110-01	–
9360	Monitor the temperature of the power modules	597 110-01	661
9367	I2t monitoring	597 110-01	668
9382	Profibus: Read data from a DPV1 data block of analog SIEMENS AS-i slaves (to Module 9385)	597 110-01	–
9383	Profibus: Set data for a DPV1 data block of analog SIEMENS AS-i slaves (from Module 9386)	597 110-01	–
9385	Profibus: Read DPV1 data	597 110-01	–
9386	Profibus: Transmit DPV1 data	597 110-01	–

Module	Function	SW version	Page
9404	Beginning of a movement within an active strobe (channel-specific)	597 110-01	902
9405	Translate symbolic operands	597 110-01	1122
9410	Read spindle status	597 110-01	694
9411	Read spindle position / speed	597 110-01	443
9412	Stop the spindle	597 110-01	695
9413	Move the spindle	597 110-01	696
9414	Position the spindle	597 110-01	701
9415	Synchronize spindles	597 110-01	714
9429	Interrogate the status of the executed NC program	597 110-04	–
9430	Temporarily change numerical machine parameters	597 110-01	300
9431	Read numerical machine parameter	597 110-01	302
9432	Temporarily change string machine parameters	597 110-01	303
9433	Read string machine parameters	597 110-01	304
9434	Select the parameter block	597 110-01	314
9435	Find the active parameter block	597 110-01	315
9440	SQL: Open a transaction	597 110-01	1232
9441	SQL: Save changes and end transaction	597 110-01	1233
9442	SQL: Find a record in the result set	597 110-01	1234
9443	SQL: Get a record from the result set	597 110-01	1235
9444	SQL: Change a record in the result set	597 110-01	1236
9445	SQL: Read a single value in a table	597 110-01	1237
9447	SQL: Delete record from result set	597 110-01	1238
9448	SQL: Load column description	597 110-01	1239
9449	SQL: Extract a value from a list separated by comma	597 110-01	1240
9450	Execute SQL command	597 110-01	1241
9451	SQL: Reject changes and close transaction	597 110-01	1242
9452	SQL: Find next record in the result set of a query	597 110-01	1243
9453	SQL: Pull binary data from the query result	597 110-01	1244
9454	SQL: Update binary data in the result set of a query	597 110-01	1245
9455	SQL: Read a single numeric value in a table	597 110-01	1246
9458	SQL: Unload column description	597 110-01	1247
9459	SQL: Change or insert a value in a list separated by comma	597 110-01	1248
9480	Select the channel display	597 110-04	1040
9481	Find the channel display	597 110-04	1041
9482	Select the spindle display	597 110-04	1041
9483	Find the spindle display	597 110-04	1042

5.2 Overview of the PLC Operands

5.2.1 PLC operands of the General Data group

	Operand	Description	SW version	Page
	General – control configuration			
D	NN_GenOmgCount	Number of configured groups of operating modes		–
D	NN_GenChnCount	Number of configured machining channels		386
D	NN_GenAxCount	Number of configured logical axes (including spindles)	597 110-01	386
D	NN_GenSpiCount	Number of configured spindles	597 110-01	687
	General – control status			
D	NN_GenOmgManual	Selected operating mode group for manual operation	597 110-01	890
D	NN_GenChnManual	Selected machining channel for manual control	597 110-01	904
D	NN_GenSpiManual	Selected spindle for manual control	597 110-01	904
M	NN_GenCycleAfterPowerOn	1st PLC scan after power on	597 110-01	949
M	NN_GenCycleAfterPlcStop	1st PLC scan after PLC interruption	597 110-01	949
M	NN_GenCycleAfterReConfig	1st PLC scan after changing of the configuration data	597 110-01	949
M	NN_GenNcInitialized	Control is being initialized (after start-up cycles)	597 110-01	949
M	NN_GenNcEmergencyStop	Control in external emergency stop state	597 110-01	681
	General – error handling			
M	NN_GenApiModuleError	An error occurred while using an API module.	597 110-01	–
D	NN_GenApiModuleError Code	Error code that appeared while using an API module.	597 110-01	–
M	PP_GenReactApiModuleError	Execute the configured reaction to error in API module: The reaction planned in CfgPlcOptions apiErrorReaction is executed only if this marker is set.		–
	General – key information			
D	NP_GenKeyCode	Code of the depressed key	597 110-01	960
D	NP_GenModCode	Code of the code number last entered	597 110-01	886
M	PP_GenHandwheelLocked	Disable handwheel motion	597 110-01	971
	General – touch probe			
M	NN_GenTchProbeReady	Touch probe ready (hardware signal)		–
M	NN_GenTchProbeDeflected	Stylus deflected (hardware signal)		–
M	NN_GenTchProbeBatteryLow	Battery voltage too low (hardware signal)		–
M	NN_GenTchProbeX13	TT (which means X13) active for tool measurement		–

	Operand	Description	SW version	Page
	General – safe control			
M	NN_GenSafetyInputs	Safety oriented: Inputs 0 to 15 bit-encoded	597 110-01	–
M	NN_GenSafetyStopActive	Safety oriented: Stop is activated	597 110-01	–
M	NN_GenSafetySelftest	Safety related: control conducts a self test		–
M	NN_GenSafetyFeedLimit Active	Safety-oriented: Limitation of the feed rate or spindle infeed active	597 110-01	–
	General – table editor			
M	NN_GenTableEditEnd	PLC: Table editor was exited by the machine operator via the END key	597 110-01	–

5.2.2 PLC operands of the Operating Mode Group group

	Operand	Description	SW version	Page
	OMG – Operating modes			
D	NN_OmgChnCount	Number of machining channels in operating mode group		–
D	NN_OmgChn	Assigned channels in this operating mode group		–
M	NN_OmgManual	Manual Operation operating mode	597 110-01	890
M	NN_OmgHandwheel	Electronic Handwheel operating mode	597 110-01	890
M	NN_OmgMdi	Positioning with Manual Data Input operating mode	597 110-01	890
M	NN_OmgProgramSingle	Program Run, Single Block operating mode	597 110-01	890
M	NN_OmgProgramRun	Program Run, Full Sequence operating mode	597 110-01	890
M	NN_OmgReference	Reference operating mode	597 110-01	890
M	NN_OmgJogIncrement	Incremental Jog operating mode	597 110-01	890
D	NN_OmgAuxiliaryMode	Product specific code for special modes and submodes	597 110-01	891
M	PP_OmgHandwheelNotAllowed	Reserved for expanded handwheel functions (disable activation of HR 420/ 5** handwheel for these operating modes)	597 110-01	–
M	NN_OmgHandwheelControl	Reserved for expanded handwheel functions (operation of HR 420/5** handwheel for these operating modes)	597 110-01	–
	OMG – Program run			
M	PP_OmgNcStart	NC start for all machining channels of this operating mode group	597 110-01	893
M	PP_OmgNCStop	NC stop for all machining channels of this operating mode group	597 110-01	893

5.2.3 PLC operands of the Machining Channels group

	Operand	Description	SW version	Page
	Channel – Configuration			
D	NN_ChnAxisCount	Number of axes of this machining channel	597 110-01	389
D	NN_ChnAxis	Array D[8] of the axes of this machining channel	597 110-01	389
	Channel – Error handling			
M	NN_ChnErrorWarning	Error or warning occurred	597 110-01	903
M	NN_ChnErrorFStop	Feed stopped because of an error	597 110-01	903
M	NN_ChnErrorNCStop	NC stop due to an error	597 110-01	903

	Operand	Description	SW version	Page
M	NN_ChnErrorCancel	Program canceled because of an error	597 110-01	903
M	NN_ChnErrorEmergency Stop	Emergency stop because of an error	597 110-01	903
M	NN_ChnErrorReset	Reset because of an error	597 110-01	903
	Channel – Program run			
M	PP_ChnNcStart	NC start or Cycle on	597 110-01	894
M	PP_ChnNCStop	NC stop or Cycle off	597 110-01	896
M	NN_ChnNcStartExtern Request	External request for NC start	597 110-01	894
M	NN_ChnNcStopExtern	NC stop or Cycle off	597 110-01	897
M	NN_ChnControlInOperation	Control is in operation	597 110-01	901
M	NN_ChnProgStoppedAsync	Asynchronous NC program interruption	597 110-01	897
M	NN_ChnProgStopped	NC program interruption	597 110-01	897
M	NN_ChnProgCancel	NC program cancellation	597 110-01	898
M	NN_ChnProgEnd	NC program end has been reached	597 110-01	895
M	NN_ChnAutostart	Autostart function: activate		–
M	NN_ChnAutostartTimeExpired	Autostart function: request for program start (NC start is triggered only be the PLC program, MP7683 # BIT 7 is no longer supported)		–
M	PN_ChnAutostartEnable	Release of autorun		–
M	NP_ChnProgSelected	NC program was selected: support depends on the product		–
M	NN_ChnBlockScan	Mid-program startup (or block scan) active	597 110-01	898
M	NN_ChnBlockScanStrobe Transfer	Restore status at block scan (M/S/T/Q transfer)	597 110-01	898
M	NN_ChnProgManTraverse	Manual traverse of the axes active (for lathe controls: inspection operation)	597 110-01	897
M	NN_ChnProgReturnContour	Return to contour active (after manual traverse or block scan)	597 110-01	897
M	NN_ChnTchProbeCycle	Probing process is active		–
M	PP_ChnTchProbeMonitor	NC stop in all operating modes if stylus is deflected		–
	Channel – Feed rate			
D	NN_ChnProgFeedMinute	Programmed feed per minute [mm/rev]	597 110-01	567
D	NN_ChnProgFeedRevolution	Programmed feed per revolution [mm/min]	597 110-01	567
D	NN_ChnProgFeedThread	Programmed feed rate per thread [mm/rev]	597 110-01	567
M	NN_ChnFeedMinuteActive	Feed rate per minute active	597 110-01	567
M	NN_ChnFeedRevolution Active	Feed rate per revolution active	597 110-01	567
M	NN_ChnFeedThreadActive	Feed rate per thread active	597 110-01	567
M	NN_ChnFeedRapidTraverseActive	Rapid traverse active (FMAX)	597 110-04	567

	Operand	Description	SW version	Page
D	NN_ChnContourFeed	Current contouring feed rate [mm/min]	597 110-01	567
D	PP_ChnContourFeedMax	Max. feed rate from PLC [mm/min]	597 110-01	567
D	NN_ChnFeedOverrideInput	Feed-rate override set [%]	597 110-01	992
D	PP_ChnFeedOverride	Feed-rate override entered by the PLC [%]	597 110-01	992
D	NN_ChnRapidFeedOverride Input	Rapid traverse override set [%]	597 110-01	–
D	PP_ChnRapidFeedOverride	Rapid traverse override entered by the PLC [%]	597 110-01	–
D	PP_ChnConfigOverride	Configurable override (e.g. rapid traverse)	597 110-01	–
M	PP_ChnFeedEnable	Feed-rate enable for all axes	597 110-01	573
M	PP_ChnWorkFeedEnable	Enable rapid traverse for all axes	597 110-01	573
M	NN_ChnToolLifeExpired	Tool life 1 expired		–
	Channel – Status			
M	PP_ChnRapidTraverseKey	Rapid-traverse key	597 110-01	–

5.2.4 PLC operands of the Axis group

	Operand	Description	SW version	Page
	Axis – Configuration			
D	NN_AxLogNumber	Logical axis number (identical to the axis number from "Axes of the machining channel")	597 110-01	386
	Axis – Drive			
M	NN_AxDriveReady	Axis drive is ready	597 110-01	569
M	PP_AxDriveOnRequest (CM9161)	Switch axis drive on	597 110-01	569
M	NN_AxDriveOn (CM9162)	Axis drive is switched on (and is at least speed-controlled)	597 110-01	569
M	PP_AxPosControlRequest	Position-control the axis	597 110-01	569
M	NN_AxPosControl	Axis is position-looped	597 110-01	569
M	PP_AxValueActToNominal	Actual-to-nominal value transfer	597 110-01	572
M	NN_AxCorrectingLagError	Following error eliminated	597 110-01	572
M	PP_AxClampModeRequest	Prepare to open the position control loop	597 110-01	571
	Axis – Control			
M	NN_AxReferenceAvailable	Reference mark not yet traversed	597 110-01	531
M	PP_AxReferenceEndPosition	Reference end position	597 110-01	531
D	PP_AxManualFeedMax	Maximum manual axis feed rate [mm/rev]	597 110-01	568
M	PP_AxFeedEnable	Axis-specific feed-rate enable	597 110-01	573
M	NN_AxMotionRequest	Axis movement by the interpolator	597 110-01	655

	Operand	Description	SW version	Page
M	NN_AxInMotion	Axes in motion	597 110-01	655
M	NN_AxInPosition	Axes in position	597 110-01	654
M	PP_AxTraversePos	Manual traverse in positive direction	597 110-01	568
M	PP_AxTraverseNeg	Manual traverse in negative direction	597 110-01	568
M	PP_AxHandwheelLocked	Disable handwheel motion for specific axes	597 110-01	971
M	PP_AxDeactivateMonitoring	Deactivate monitoring functions	597 110-01	645
M	NN_AxLubricationPulse	Lubrication pulse: Value in MP_lubricationDist exceeded	597 110-01	455
M	PP_AxLubricationDistReset	Reset the accumulated distance	597 110-01	455

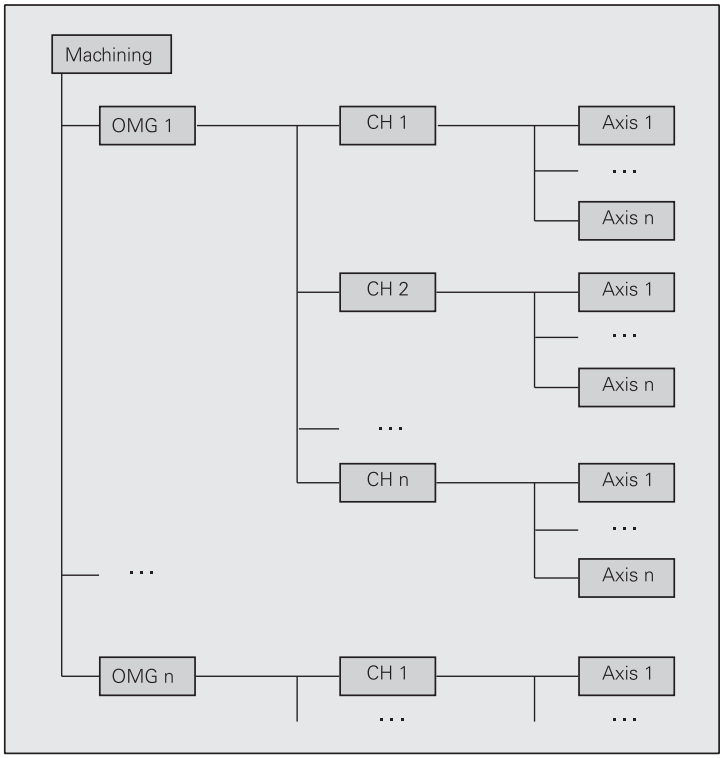
5.2.5 PLC operands of the Spindle group

	Operand	Description	SW version	Page
	Spindle – Configuration			
D	NN_SpiLogNumber	Logical axis number of the spindle	597 110-01	687
	Spindle – Drive			
M	NN_SpiDriveReady	Spindle drive is ready	597 110-01	692
M	PP_SpiDriveOnRequest (CM9161)	Switch spindle drive on	597 110-01	692
M	NN_SpiDriveOn (CM9162)	Spindle drive is switched on (and is at least speed-controlled)	597 110-01	692
	Spindle – Control			
M	NN_SpiReferenceAvailable	Reference position found	597 110-01	688
M	PP_SpiReferenceMarkSignal	Trip dog	597 110-01	688
D	PP_SpiSpeedMax	Maximum spindle speed	597 110-01	693
M	PP_SpiEnable	Spindle enabling	597 110-01	693
M	NN_SpiMotionRequest	Spindle movement by interpolator	597 110-01	655
M	NN_SpiInMotion	Spindle in motion	597 110-01	693
M	NN_SpiSpeedOK	Spindle speed reached	597 110-01	700
M	NN_SpiControl	Spindle in closed loop	597 110-01	702
M	NN_SpiControlInPos	Spindle in position	597 110-01	702
M	NN_SpiSyncSpeed	Rotational speed synchronism active	597 110-01	714
M	NN_SpiSyncAngle	Angle synchronism active	597 110-01	714
M	NN_SpiSyncReached	Synchronous operation reached	597 110-01	714
D	NN_SpiOverrideInput	Speed override set [%]	597 110-01	990
D	PP_SpiOverride	Speed override [%] entered by the PLC	597 110-01	990
M	NN_SpiTapping	Tapping active	597 110-01	702
M	NN_SpiRigidTapping	Tapping with spindle interpolated with Z axis active	597 110-01	702

6 Configuring the Axes and Spindle

6.1 Machine Structure

6.1.1 MANUALplus 620 Adapting to the machine



Legend:

- OMG: Operating mode group (OperatingModeGroup)
- CH: Machining channel (NC channel)
- Axis: Axis or spindle

Use the following organizational structure to configure the machine structure in the MANUALplus 620:

- The machine consists of operating mode groups (OperatingModeGroups).
- Every operating mode group manages machining channels.
- Every machining channel manages axes.

The operating mode groups of a machine as well as the machining channels of an operating mode group operate independently of each other.

All machines have this organizational structure—even if a simple machine requires only one OMG and one channel.

A structure for simulation corresponding to the physical structure of the machine must be configured. As a rule, the machine structure and the structure for simulation have the same configuration.

Settings in the configuration editor	MP number
System CfgAxes **axisList** **specCoordSysList**	 100001 100003

PLC operand / Description	Type
NN_GenChnCount Number of configured machining channels	D

Defining the logical axes

Within the geometry and interpolator processing, a unique identifier (=logical axis number) must be assigned to each axis. The identifier corresponds to the index in **MP_axisList.** Logical axes are defined by sequential numbering starting from the index [0].

The definition of the logical axes is independent of their assignment to the machining channels.

- Enter the axes of all channels of the machine, including spindles and PLC axes.

MP_axisList

Key names for all axes on the machine
Available from NCK software version: 597 110-01.

Format: Array [0...9]
Input: (# = logical axis number)
String of max. 18 characters
Default settings:
[0]: X1 (logical axis number 0)
[1]: Z1 (logical axis number 1)
[2]: S1 (logical axis number 2)
[3]: S2 (logical axis number 3)
...
Default: –
Access: LEVEL3
Reaction: RESET

The PLC indicates the number of configured logical axes in **NN_GenAxCount. NN_AxLogNumber** contains the logical axis number for using general modules (such as Module 9165). The axis number corresponds to the index from **NN_ChnAxis** (see "Configuring a machining channel" on page 388).

PLC operand / Description	Type
NN_GenAxCount Number of configured logical axes (including spindles)	D
NN_AxLogNumber Logical axis number (identical to the axis number of "axes of the machining channel")	D

Spindles

In the control software, spindles and axes are treated in largely the same way. Spindles are considered a special kind of axis and are defined as a logical axis.

For parameters, PLC operands and spindle functions, See "Spindles" on page 687.

Special kinematics axes

Axes that are used in the kinematics model but are not entered in **MP_CfgAxes/axisList** are defined in **MP_specCoordSysList.**

- In **MP_specCoordList**, enter the axes for which in **CfgAxisPropKin/specKinCoordSys** one of the following attributes is defined (See "Kinematic properties of axes" on page 412):
  - FixedTransAxis
  - DefPointTrans
  - DefPointRot

The special kinematics axes are indicated by sequential numbering starting with the index [0].

MP_specCoordSysList

	Key names of special axes for the kinematics description Available from NCK software version: 597 110-01.
Format:	Array [0...9]
Input:	Max. 18 characters
Default:	[0]: Y1
Access:	LEVEL3
Reaction:	RESET

6.2 Configuration of Machining Channels

6.2.1 Configuring a machining channel

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannelAxes	
progAxis	200301
restoreAxis	200305
kinModels	200306
deactFastClamping	200307
CfgNcErrorReaction	
warningLevel	200601
CfgNcPgmParState	
persistent	200701
currentSet	200702
CfgNcPgmBehaviour	
operatingTimeReset	200801

Use the machining channel key names defined in **MP_channelList** as "Key for channel."

Axes of machining channel

In the parameter object **CfgChannelAxes,** you specify the axes of the machining channel (NC channel) and define the behavior of the axes during reference run.

In **MP_progAxis,** enter the axes which can be used within the NC program. Axes that are **not** included are, for example, slave axes in master-slave operation or axes that are for display only.

MP_progAxis

Programmable axes
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: Programmable axis names and axis names for the position display in the workpiece system. The key names of the axes are taken from the System/CfgAxes/axisList parameter and automatically shown as a selection menu.

Default: –

Access: LEVEL3

Reaction: RESET

Deactivating "fast" clamping

There is an accelerated procedure for clamping axes: the unclamping command for an axis is already sent to the PLC before the nominal position value filter. A clamping command is not sent to the PLC until after the nominal position value filter.

Note

This procedure cannot be used for a central drive, since the clamping and unclamping commands refer to the same axis for a central drive.

MP_deactFastClamping

Deactivation of "fast" clamping
Available from NCK software version: 597 110-03.
Format: Selection menu
Selection: **TRUE**
Deactivate fast clamping
FALSE
Axis works with fast clamping
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

In **NN_ChnAxisCount,** the NC informs the PLC of the number of axes assigned to this machining channel. The axes assigned to this machining channel are indicated in the array **NN_ChnAxis.**

PLC operand / Description	Type
NN_ChnAxisCount Number of axes of this machining channel	D
NN_ChnAxis Only the axes (and not the spindles) are entered. The order of the entries has no meaning.	D

Kinematics of machining channel

Up to HEIDENHAIN NCK software version 597 110-02:

In **MP_kinModels,** enter the kinematics models of this machining channel. After control start-up, the last entry will be activated.

MP_kinModels

Key names of the available kinematic models for this channel
Available from NCK software version: 597 110-01.

Format: Array [0...15]
Input: Key names from MP_CfgKinModels
Default: –
Access: LEVEL3
Reaction: REF

Please also note the documentation regarding the configuration of the machine kinematics under „Machine kinematics for lathes (up to NC software 548 328-02)" on page 502.

Error behavior of machining channel

The parameter **MP_warningLevel** specifies the behavior when FN14 errors occur (there are no FN14 errors for lathe controls!).
Errors are triggered only if according to the PET table the warning level of the error is maximally as high as the warning level set here. Note that errors with warning level 0 are always triggered and errors with warning level 5 are never triggered.

MP_warningLevel

Warning level of channel
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to 4
0: FN14 errors with warning level = 0 are triggered
1: FN14 errors with warning level </= 1 are triggered
2: FN14 errors with warning level </= 2 are triggered
3: FN14 errors with warning level </= 3 are triggered
4: FN14 errors with warning level </= 4 are triggered
Default: 0
Access: LEVEL2
Reaction: RUN

Saving Q/QS parameters

In the parameter object **CfgNcPgmParState,** you specify whether and where Q/QS parameters are to be stored in non-volatile memory. If **MP_persistent** is set to **TRUE**, the tool that was last active is displayed again after the control is booted. If the machine parameter is set to **FALSE**, T0 is displayed after the control is booted.
If **MP_currentSet** is not defined, the name of the machining channel is used as name for the Q/QS parameter set.

MP_persistent

Defines the storage of Q/QS parameters
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Q/QS parameters are stored in non-volatile memory.
The tool that was last active will be displayed again after the control is booted.
FALSE
Q/QS parameters are not stored
The tool T0 will always be displayed after the control is booted.
Default: FALSE
Access: LEVEL2
Reaction: RUN

MP_currentSet

Name of Q/QS parameter set
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 18 characters
Name of active Q/QS parameter set
If no name is indicated, the key name of the machining channel is used for storage
Default: –
Access: LEVEL2
Reaction: RUN

Totaling the program machining time

With the **CfgNcPgmBehaviour/operatingTimeReset** parameter you specify whether the machining time is to be totaled, or whether it is reset at each program start.

MP_operatingTimeReset

Specify the totaling of the program machining time
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
The machining time is reset at each program start
FALSE
The machining time is totaled
Default: FALSE
Access: LEVEL3
Reaction: RESET

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannelAxes	
refAxis	200303
refAllAxes	200304

In **MP_refAllAxis,** you specify whether all axes are to be referenced in the sequence defined in **MP_refAxis,** or whether the reference point in these axes is to be traversed by pressing the axis-direction keys.

The automatic or MDI operating mode cannot be used until all axes entered in **MP_refAxes** have been referenced.

MP_refAllAxes

Reference all axes in succession after an NC START
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Traverse the reference mark of all axes in succession after an NC START
FALSE
A reference run is not performed for all axes
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_refAxis specifies the axes to be referenced. The sequence of the reference run is determined by the index.
Index [0] = First axis

MP_refAxis

Axes in which the reference mark is to be traversed
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: The key names of the axes are taken from the CfgAxes/axisList parameter and automatically shown as a selection menu.
Default: –
Access: LEVEL3
Reaction: RESET

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the machining channel] CfgFeedLimits **restoreFeed**	400309
Channels ChannelSettings [Key name of the machining channel] CfgChannelAxes **restoreAxis**	200305

With the MANUALplus 620 you can resume an interrupted NC program at the desired block number by scanning the previous blocks (please refer to the User's Manual).

Moving the axes:

- After an NC STOP the axes are moved to the last interpolated position (stop position).
- During a block scan, they are moved to the calculated restore position.

Markers inform the PLC about individual conditions during mid-program startup (block scan). Depending on these markers you can enable certain functions such as the axis-direction buttons for **MANUAL TRAVERSE.**

NN_ChnProgManTraverse is set if the **MANUAL TRAVERSE** soft key is pressed.

NN_ChnProgReturnContour is set if the **RESTORE POSITION** soft key is pressed ("Return to Contour").

NN_ChnBlockScan is set if the **BLOCK SCAN TO BLOCK N** soft key is pressed. NN **ChnBlockScan** is reset if the **RESTORE POSITION** or **INTERNAL STOP** soft key is pressed.

To move to the restore position:

▶ In **MP_restoreAxis**, specify the sequence in which the axes are to move.

MP_restoreAxis

Sequence for returning to the contour
Available from NCK software version: 597 110-03.

Format: Selection menu
Selection: The key names of the axes are taken from the System/CfgAxes/axisList parameter and automatically shown as a selection menu.
Default: –
Access: LEVEL3
Reaction: RESET

- In **MP_restoreFeed**, specify the feed rate at which positions are to be approached.

MP_restoreFeed

Feed rate for returning to the contour
Available from NCK software version: 597 110-02.

Format: Numerical value
Input: Feed rate in [mm/min] or [°/min] for returning to the contour after an NC stop or block scan.
Default: 1200 [mm/min] or [°/min]
Access: LEVEL3
Reaction: RUN

6.3 Configuration of Axes

Overview of the types of axes supported by the MANUALplus 620:

- **Digital axes (not MC 320T):**
  Closed-loop axes with encoder; controlled by a controller unit (CC)
- **Analog axes:**
  Closed-loop and open-loop axes with and without encoder; with analog output of nominal value
- **Manual axes:**
  Axes with encoder but without motor; these axes can be programmed and are taken into account in the kinematics (e.g. manually operated axes with mechanical handwheels)
- **Display axes:**
  Axes with encoder but without motor; these axes cannot be programmed, and are not taken into account in the kinematics
- **Virtual axes:**
  Axes with neither an encoder nor a motor; however, their nominal values are added to real axes (non-circular grinding)
- **PLC axes:**
  Axes that are controlled by the PLC run-time system, e.g. magazine axes or spindles with coded spindle speed output.

Settings in the configuration editor	MP number
System CfgHardware **hardwareType**	 100101

▶ Specify the type of the drive-controller hardware with the **MP_hardwareType** parameter:

MP_hardwareType

Type of drive controller hardware
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **automatic**
Automatic identification of controller unit
CC422
CC 422 controller unit for conventional axes
CC424
CC 424 controller unit for direct drives with high control loop requirements (very short cycle times)
CC61xx
CC 61xx controller unit with HSCI interface
NoCC
No controller unit installed—select this setting for analog controlled machines (TNC 320 or MANUALplus 620 with MC 320T or analog interface to the drives)

Default: automatic

Access: LEVEL3

Reaction: RESET

6.3.1 Axis designations and coordinates

Principal, parallel and rotary axes are distinguished.

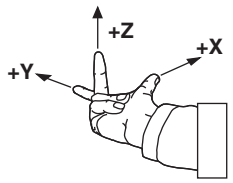
Properties of the principal axes X, Y, Z

X, Y and Z axes are principal axes. These axes have a defined spatial orientation in a coordinate system model, and are always linear.

It is of no importance to the editor whether the current coordinate system is that of the machine base system, or is aligned otherwise.

They are the principal coordinates for programming in the editor.

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

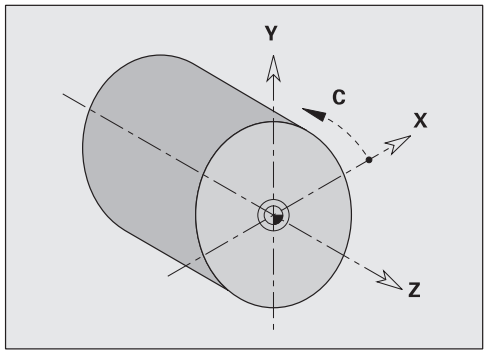


Algebraic signs of the axes

When the programmer writes an NC program, he always assumes that the tool (not the workpiece) is in motion. If the machine moves its workpiece holding element (spindle with C axis) in a particular axis instead of the tool, then the direction of actual motion is opposite to the direction of axis motion. In this case the direction of motion is designated with the same algebraic sign as the axis direction, but with an apostrophe: +X′, +Y′ and +Z′

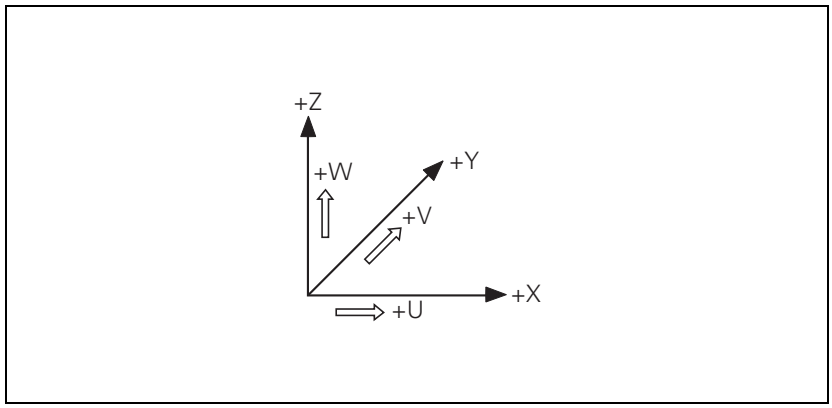
Properties of the rotary axis C

For the rotary axis C, the turning axis is in the direction of the principal coordinate Z. The C axis is therefore a "parallel" rotary axis. The direction of the rotary axis C follows the "right-fist rule." The fingers of the closed right hand point in the proper rotation direction of an axis when the thumb points in the direction of the associated linear axis Z. The angle values of the C axis are given with respect to the zero point of the C axis.



Properties of the linear axes U, V, W

The additional axes U, V and W are parallel linear axes.
They are parallel coordinates for programming in the editor.



Standard coordinates

The meanings of the coordinates X, Y, Z, A, B, C, U, V and W are specified in **DIN 66 217.**

The MANUALplus 620 views coordinates whose **MP_axName** parameter begins with X, Y, Z, A, B, C, U, V or W as standardized coordinates.

For these standardized coordinates, the parameters of the object **CfgProgAxis** must obey the following rules:

First letter of parameter MP_axName	Parameter MP_dir	Parameter MP_progKind
X	XAxis	MainLinCoord
Y	YAxis	MainLinCoord
Z	ZAxis	MainLinCoord
U	XAxis	ParallelLinCoord
V	YAxis	ParallelLinCoord
W	ZAxis	ParallelLinCoord
A	XAxis	ParallelAngCoord
B	YAxis	ParallelAngCoord
C	ZAxis	ParallelAngCoord

Settings in the configuration editor	MP number
Axes	
CfgProgAxis	
[Key name of the axis]	
axName	300001
dir	300002
progKind	300003
index	300004
relatedAxis	300005

In the parameter object **CfgProgAxis,** define and describe all axes that are programmable and/or are displayed. This description is independent of the assignment to NC channels.

For information about programmable axes, see "Standard coordinates" on page 398.

The name entered in **MP_axName** is used for programming/editing and in the position display.

MP_axName

Designation of the axis for position display
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 1 letter
e.g.: X, Y, Z, A, B, C, U, V, W
Default: –
Access: LEVEL3
Reaction: RESET

MP_dir

Spatial orientation of the axis or center of rotation
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **XAxis**
Motion/rotation in X direction
YAxis
Motion/rotation in Y direction
ZAxis
Motion/rotation in Z direction
SpecAxis
Free/undefined spatial orientation, e.g. for spindle
Default: –
Access: LEVEL3
Reaction: RESET

MP_progKind

Type of axis
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
MainLinCoord
Main coordinate, always linear (X, Y, Z)
ParallelLinCoord
Secondary linear coordinate (U, V, W)
ParallelAngCoord
Parallel coordinate, rotary (A, B, C)
SatelliteLinCoord
Minor coordinate, linear: Reserved, not used at present.
SatelliteAngCoord
Minor coordinate, rotary: Reserved, not used at present.
Spindle
Spindle

Default: –
Access: LEVEL3
Reaction: RESET

Position preset for programmable axes

Module 9142 Position preset for programmed axis

Use Module 9142 to preset a programmable axis to a new position value. The position value of another axis or a freely selected position value can be transmitted as the target position.

Constraints:

- The module functions only in the cyclic PLC program.
- The position preset is only possible when the control is not active (NN_ChnControlInOperation=0) or if an M/S/T/T2/G strobe is pending.
- When calling the module for an NC axis during a strobe, the synchronization SYNC_CALC must be configured for this strobe.

Call:

PS B/W/D/K <Type of position preset>
1: For <logical axis number target>, set the new position to the <position value>,
<Logical axis number source> is not necessary.
(corresponds to CM 9147)
2: For <logical axis number target>, set the new position to the value of <logical axis number source>, <position value> is not necessary

PS B/W/D/K <Logical axis number target>

PS B/W/D/K <Logical axis number source>

PS B/W/D/K <Position value>
Input unit: [0.0001 mm]

CM 9142

PL D <Error code>

0: Position preset has been read
1: <Logical axis number target> axis is outside the valid range
2: <Logical axis number target> axis is not configured
3: Missing strobe or control is active
4: <Logical axis number source> axis is outside the valid range
5: <Logical axis number source> axis is not configured
6: <Logical axis number source> axis has not traversed the reference position
7: <Type of position preset> not known
8: The module was called in a submit/spawn job.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule ErrorCode	0	No error
	2	An invalid axis number was transferred.
	2	An invalid type of position preset was transferred.
	21	Missing strobe or control is active.
	24	The module was called in a submit/spawn job.

Axis without a separate drive motor

Enter the axis name of the assigned physical axis in **MP_relatedAxis** if the key name of the programmable axis does not correspond to the key name of the physical axis. This links the axis with the physical axis. The axes concerned are usually axes that do not have a separate drive motor.

Example: If the spindle drive is used for the C axis, you link the C axis with the physical axis "spindle."

MP_relatedAxis

Assigned physical axis
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: Select the key name of the programmable axis from the selection menu.
Only enter an axis name if the key name of the programmable axis does not correspond to the key name of the physical axis.
The key names of the axes are taken from the System/CfgAxes/axisList parameter and automatically shown as a selection menu.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

Settings in the configuration editor	MP number
Axes	
PhysicalAxis	
[Key name of the axis]	
CfgAxis	
isAng	300101
isModulo	300102
restoreModuloCntr	300111
isHirth	300103
axisHw	300104
axisMode	300105
testMode	300106
parList	300107
realAxis	300108
noActToNomAtEmSt	300109
deactivatedAtStart	300110
advancedSettings	300112

In the parameter object **PhysicalAxis,** define all axes that can be instructed by the interpolator to execute a command. The description in **PhysicalAxis** is independent of the assignment to NC channels.

In the parameter object **CfgAxis,** you specify the axis type and drive interface, you assign a parameter set to the axis and define the operating mode of the axis.

MP_isAng

Rotary axis
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **TRUE**
This axis is a rotary axis.
FALSE
This axis is a linear axis (not a rotary axis).
Note:
Rotary axes are not allowed as principal axes.

Default: FALSE

Access: LEVEL3

Reaction: RESET

Position display for rotary axes and PLC auxiliary axes (modulo display)

In **MP_isModulo** you define whether the modulo limit of 360 degrees should apply to the position display of rotary axes or PLC auxiliary axes.

MP_isModulo

Modulo display
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Position display for rotary axes: modulo 0 to 360°
FALSE
Position display is not modulo display.
Default: FALSE
Access: LEVEL3
Reaction: RESET

Note

You must always configure spindles as modulo axes:
MP_isModulo = True.

In **MP_restoreModuloCntr**, you specify whether the modulo counter of the axis is to be saved cyclically in SRAM and restored during control startup.

MP_restoreModuloCntr

Save modulo counter of the axis in SRAM
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **TRUE**
The modulo counter of the axis is saved cyclically in SRAM and restored during startup of the MANUALplus 620.
FALSE
The modulo counter of the axis is not saved in SRAM and not restored during control startup.
Default: -
Access: LEVEL3
Reaction: RUN

Settings in the configuration editor	MP number
Axes	
PhysicalAxis	
[Key name of the axis]	
CfgRollOver	
shortestDistance	300401
startPosToModulo	300402
showModuloDisp	300403

Roll-over axes are rotary axes with modulo counting mode that are able to execute several or any number of revolutions. Example:
A machine with C axis as rotary table.

If you want to configure a rotary axis as rollover axis, you have to insert the optional configuration object **CfgRollOver** in the machine configuration for the respective axis.

With the three machine parameters **MP_shortestDistance**, **MP_startPosToModulo** and **MP_showModuloDisp** you define the characteristics of the rollover axis.

Set the parameter **MP_shortestDistance** = **on** so that the axis approaches the target position on the shortest path (<= 180°) if programming is non-incremental. The nominal and the target value display is always between 0° and 360°, also with incremental programming.

MP_shortestDistance

Traverse path of rotary axis with modulo counting mode
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **on**
Always approach target position on the shortest path (<= 180°)
off
Approach target position without passing over zero
Default: No value, parameter optional (= off)
Access: LEVEL3
Reaction: RUN

Set the parameter **MP_startPosToModulo** = **on**, to move the position of the rotary axis into the module limits (0...360°) before the beginning of the positioning block. The **MP_startPosToModulo** parameter is effective only if you configure **MP_shortestDistance** = **off.**

MP_startPosToModulo

Rollover axis moves start angle into the range of 0 to 360°
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **on**
Move the rotary axis into the 0 to 360° range at the beginning of the positioning block.
off
Do not move the rotary axis into the 0 to 360° range.

Default: No value, parameter optional (= off)

Access: LEVEL3

Reaction: RUN

The **MP_showModuloDisp** parameter defines the display type for the modulo axis. Set the parameter to **on** to display the position of the rotary axis within the modulo limits 0...360°. The **MP_showModuloDisp** parameter is effective only if you configure **MP_shortestDistance** = **off**.

MP_showModuloDisp

Display of rollover axis in the range of 0..360°
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **on**
Display of rollover axis within the modulo limits of 0..360°
off
Display of rollover axis not within the modulo limits

Default: No value, parameter optional (= off)

Access: LEVEL3

Reaction: RUN

Example:

Effect of the **MP_shortestDistance** and **MP_startPosToModulo** parameters

Programmed movement	shortestDistance = on			startPosToModulo = on			shortestDistance = off startPosToModulo = off		
	Nominal	Ref	Distance	Nominal	Ref	Distance	Nominal	Ref	Distance
G110 C0	0	0	–	0	0	–	0	0	–
G110 C185	185	–175	–175	185	185	+185	185	185	+185
G110 C–5	355	355	+170	–5	–5	–190	–5	–5	190
G110 C180	180	180	–175	180	180	–175	180	180	+185
G110 C0	0	0	–180	0	0	–180	0	0	–180
G110 Ci 720	0	720	+720	720	720	+720	720	720	+720
G110 Ci –800	280	–800	–800	–800	–800	–800	–80	–80	–800

Setting the drive interface

Use the **MP_axisHw** machine parameter to define the drive interface. The parameter defines the hardware type of the axis at the time of the control startup. The parameter is evaluated only if the value "active" was entered in **MP_axisMode**.

MP_axisHw

Hardware to which the axis is connected
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
None
No hardware connection (virtual axis)
InOutCC
Digital axis with connection to a controller unit
AnalogMC
Analog drive interface; encoder connection to the MC
AnalogCC
Analog drive interface; encoder connection to the CC
DisplayMC
Display axis; encoder connection to the MC
DisplayCC
Display axis; encoder connection to the CC
ManualMC
Open-loop axis; encoder connection to the MC
ManualCC
Open-loop axis; encoder connection to the CC
ProfiNet
Digital Profinet axis (reserved, function not available yet)

Default: InOutCC
Access: LEVEL3
Reaction: RESET

Activating an axis

An axis can only be moved in a closed loop after it has been activated with the **MP_axisMode** machine parameter.

For commissioning, use the **MP_testMode** parameter to switch an axis to test mode. In this operation, the interpolator views the axis as a fully-functional axis, but the nominal values are not passed on to the drive motor. There is only an internal nominal-to-actual transfer. Such an axis does not need to have all the hardware connections made (position and speed input, PWM output), but can be "positioned."

For the linear axes X, Y and Z, the default value **`Active`** of the parameter **MP_axisMode** must not be changed to **`NotActive`**; otherwise an error message will be issued. If you want to deactivate one of these axes, you should set the parameter **MP_testMode** to **`True`**.

MP_axisMode

Operational mode of the axis
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
NotAllowed
Reserved, do not use!
NotActive
Axis does not exist
Active
Axis physically present
Virtual
Virtual axis for superimposed movements
PlcControlled
e.g. for encoded spindles; axis is controlled solely by the PLC

Default: Active
Access: LEVEL3
Reaction: RESET

MP_testMode

Axis in test mode
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
TRUE
Test mode for commissioning, i.e. the axis need not be connected
FALSE
Test mode not active.
If MP_axisMode = Active: Axis must be connected electrically

Default: FALSE
Access: LEVEL3
Reaction: RESET

Note

Please note that hanging axes cannot be supported in test mode. The PLC must ensure that these axes are braked in test mode.

Hanging axes need 100 % weight compensation.

Assigning parameter sets

In **MP_parList** enter the key name of the parameter set that is assigned to this axis. The parameter set describes the axis control response, the encoder connection, the encoder signals, etc.

You can create more than one parameter set for one axis. This enables you to define different controller settings, for example.

Examples:

a) You define different controller settings to ensure appropriate control response depending on the load.

b) The spindle and the C axis are realized by using a physical axis. This enables you to define separate parameter sets for the spindle and the C axis.

MP_parList

	List of all parameter sets of this axis Available from NCK software version: 597 110-01.
Format:	Array [0...9]
Input:	Key name of the parameter set Max. 18 characters e.g.: ParSetX-0 (parameter set for the X axis)
Default:	–
Access:	LEVEL3
Reaction:	RESET

Note

The first parameter set must be fully defined. In **KeySynonym/ CfgKeySynonym,** you can relate the other parameter sets to the first one. Then you only have to define the parameters that differ from the ones of the parameter set to which you have related the present parameter set.

Virtual axis

In **MP_realAxis**, enter the key name of the associated real axis.

For virtual axes, See "Configuration of Axes" on page 395.

MP_realAxis

	Key name of the associated real axis Available from NCK software version: 597 110-01.
Format:	Selection menu
Selection:	The key name of the associated real axis is only to be entered if the current axis is a virtual axis (e.g. for noncylindrical grinding). The key names of the axes are taken from the System/CfgAxes/ axisList parameter and automatically shown as a selection menu.
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

Actual-to-nominal value transfer

In **MP_noActToNomAtEmSt,** you can define whether an actual-to-nominal value transfer is to be conducted for all axes in an EMERGENCY STOP.

MP_noActToNomAtEmSt

Actual-to-nominal value transfer for all axes in an EMERGENCY STOP

Format: Selection menu

Selection: **TRUE**
No actual-to-nominal value transfer for any axis in an EMERGENCY STOP
FALSE
The standard control behavior is actual-to-nominal value transfer for all axes in an EMERGENCY STOP.

Default: -

Access: LEVEL3

Reaction: RESET

Advanced settings for individual axes

In **MP_advancedSettings**, you can configure advanced settings for individual axes. Set bit 0 = 1 if you do not want the PLC positioning movement of an axis to be canceled when the touch probe is deflected by another axis.
If you want to configure a very fast axis, you can activate a faster acceleration and filter calculation for the PLC movements of the respective axis by setting bit 1 = 1.

MP_advancedSettings

Advanced settings for individual axes
Available from NCK software version: 597 110-04.

Format: Numerical value (32 bits)

Input: Bit 0 = 0:
All axes are stopped when the touch probe is deflected.
Bit 0 = 1:
The PLC positioning movement of this axis is not canceled if the touch probe is deflected by another axis.
Bit 1 = 0:
Faster acceleration and filter calculation during PLC movements is not active for this axis.
Bit 1 = 1:
Faster acceleration and filter calculation during PLC movements is active for this axis.

Default:

Access: LEVEL3

Reaction: RUN

Settings in the configuration editor	MP number
Axes PhysicalAxis [Key name of the axis] CfgAxis **isHirth**	300103
Axes ParameterSets [Key name of the parameter set] CfgReferencing **moveAfterRefType**	400410

Hirth coupling describes a type of clamping of rotary axes and swivel heads. Finely splined disks mesh together in order to create a rigid connection.

With the **MP_isHirth** parameter you specify whether the axis is capable of Hirth coupling. The parameter is optional. The Hirth axis is active as soon as you insert the **MP_isHirth** parameter and enter a grid increment in [°] measured from the machine datum.

With the **MP_moveAfterRefType** machine parameter you specify the type of movement after crossing the reference mark. For the Hirth axis you can specify either positive or negative direction of approach to the next Hirth grid position after a reference run.

Use PLC Module 9038 (Read Status Information of Axes) number 4 to ascertain the active Hirth axes. Use PLC Module 9038 number 5 to read the Hirth grid increment from the machine configuration.

With Module 9125 you stop a PLC axis at the next Hirth grid position.

- Configure the exact positioning of the axis in the Hirth grid as PLC positioning.

Manual Operation mode

As soon as an axis direction key is pressed, the MANUALplus 620 resets the marker in **NN_AxInPosition** (axis in position).

- As soon as the axis-in-position bit is set again, compare the nominal position with the Hirth grid and derive from it a PLC positioning command to the next grid point.

Electronic Handwheel operating mode

For the current handwheel axis, the marker is reset in **NN_AxInPosition** (axis in position).

As soon as you select another handwheel axis, the marker **NN_AxInPosition** is set for the previous axis.

The Hirth axis can be positioned with the handwheel.

- Check the actual position with the Hirth grid and derive from it a PLC positioning command to the next grid point.

Controlled positioning

The positions of the Hirth axis must be programmed in the grid.

- Check the positions in the PLC during program run.
- As soon as "axis in position" is reset, check the target position with the Hirth grid.
- You must output a PLC error message if the target position is not on the Hirth grid.

MP_isHirth

Axis with Hirth coupling
Available from NCK software version: 597 110-02.

Format: Numerical value
Input: Prescribed grid increment in [°] for Hirth coupling, measured from the machine datum
If the parameter is inactive or the value 0 is entered, no Hirth grid is supported
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_moveAfterRefType

Type of axis movement after traversing the reference mark
Available from NCK software version: 597 110-02.

Format: Selection menu
Selection: **absolute**
Absolute positioning
relative
Incremental positioning
HirthRasterPos
Approach next Hirth grid position in positive direction
HirthRasterNeg
Approach next Hirth grid position in negative direction
Default: absolute
Access: LEVEL3
Reaction: REF

Module 9125 Stop PLC axis at next Hirth grid position

Stop an already started PLC positioning movement of an axis at the next Hirth grid position.

Call:

PS	B/W/D/K	<Axis> 0 to 8 represent axes 1 to 9
CM	9125	
PL	B/W/D	<Error code> 0: Positioning is canceled 1: Axis does not exist 2: Not a PLC axis 3: Axis was already stationary 4: Axis is not a Hirth axis

Settings in the configuration editor	MP number
Axes	
PhysicalAxis	
[Key name of the axis]	
CfgAxisPropKin	
specKinCoordSys	300201
kindOfRotAxis	300202
presetToAlignAxis	300203
hasSpecAxisData	300204

In the parameter object **CfgAxisPropKin,** specify the properties important for the kinematics configuration.

In **MP_specKinCoordSys** define whether the assigned coordinate transformation is used for defining a fixed translation axis or a datum (DefPoint).

MP_specKinCoordSys

Type of special coordinate system
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **FixedTransAxis**
Translation axis for which no physical axis exists
DefPointTrans
Coordinate system in the kinematics model of a translation axis to which no physical axis is assigned
DefPointRot
Coordinate system in the kinematics model of a rotational axis to which no physical axis is assigned

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

For rotary axes, specify in **MP_kindOfRotAxis** whether the axis concerned can rotate completely or has a limited angle of rotation.

MP_kindOfRotAxis

Type of rotational axis, only for rotary axes
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **RollOver**
Axis can roll over
NotRollOver
Axis has a limited angle of rotation

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

MP_presetToAlignAxis controls the treatment of presets for rotational axes. If the attribute is set to TRUE, the offset from the preset is subtracted from the axis value before the kinematics calculation. If it is set to FALSE, the offset is only effective for the position display of the axis.

MP_presentToAlignAxis

Controls the treatment of the preset for rotational axes
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Offset is subtracted
FALSE
Offset is only effective for the display
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_hasSpecAxisData is only for special axis data—the parameter is not used at present.

MP_hasSpecAxisData

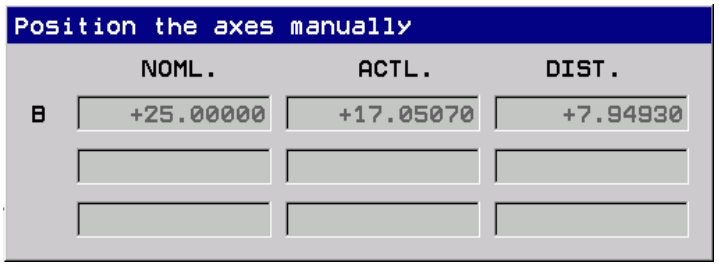
Reserved: Special axis data available, only for special axes
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Special axis data available
FALSE
No special axis data
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

6.3.6 Manual axis (counter axis)

An open-loop axis has a position encoder for determining and displaying the current position value. There is no nominal value output for the axis. The target position is set by the machine operator manually (e.g. via mechanical handwheels).

An open-loop axis can be programmed together with closed-loop axes in an NC program. When an NC block with an open-loop axis is reached during machining, a dialog window appears, prompting the machine operator to traverse the axis to the nominal coordinates:



The dialog window shows the nominal and actual position, as well as the distance-to-go, in **axis coordinates**. The MANUALplus 620 automatically triggers an NC STOP when opening the window. The axis must now be traversed by the machine operator until the target has been reached, taking into account the positioning window defined under **MP_posTolerance.** The MANUALplus 620 then automatically closes the dialog window. Machining of the NC program is continued with NC START.

Note

If an open-loop axis is programmed together with closed-loop axes in an NC block, then the open-loop axis does not have to be traversed to the target position immediately upon appearance of the dialog window. For example, you can first retract the open-loop axis, then position the close-loop axes with the NC START key, and then finish positioning the open-loop axis. The dialog window then disappears. Press NC START again to continue the NC program.

Settings in the configuration editor	Input:
Axes	
PhysicalAxis	
[Key name of the axis]	
CfgAxisHardware	
axisHw	ManualMC
axisMode	Active

- For the manually operated axis, set the **MP_axisHw** and **MP_axisMode** machine parameters to the values indicated in the table.
- Deactivate movement monitoring for the axis: Set **MP_movementThreshold** = 0.
- Deactivate standstill monitoring. Set **MP_checkPosStandstill** = 0.

6.4 Encoders

Encoders report positions and movements of the machine to the MANUALplus 620. HEIDENHAIN contouring controls operate with incremental encoders with a 1 V_{PP} signal and absolute encoders with EnDat interface.

In the parameter object **CfgAxisHardware,** define the connections of the encoders, the type of encoder, the type of signals, etc.

6.4.1 Type of position encoder

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgAxisHardware	
posEncoderType	400003
distPerMotorTurn	400004
posEncoderDist	400005
posEncoderIncr	400006
CfgReferencing	
endatSerial	400402

The parameter object CfgAxisHardware is not required for:

- Virtual axes (MP_axisMode = Virtual)

- In **MP_posEncoderType,** define the type of position measurement and the type of position encoder or speed encoder.
- In **MP_distPerMotorTurn,** define the traverse distance per motor revolution.

MP_posEncoderType

Position measurement via position encoder or motor encoder
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
MC_DISTANCE_CODED
Distance-coded position encoder on the MC (X01 to X06 and X35 to X38)
MC_NOT_DISTANCE_CODED
Not a distance-coded position encoder on the MC (X01 to X06 and X35 to X38)
MC_ENDAT
EnDat position encoder on the MC (X01 to X06 and X35 to X38)
NO_ENCODER
No position measurement
CC_MOTOR_ENCODER
Position measurement by speed encoder on the CC (X15 to X20 and X80 to X83)
CC_DISTANCE_CODED
Distance-coded position encoder on the CC (X201 to X210)
CC_NOT_DISTANCE_CODED
Not a distance-coded position encoder on the CC (X201 to X210)
CC_MOTOR_ENDAT
Position measurement by EnDat speed encoder on the CC (X15 to X20 and X80 to X83)
CC_EXTERN_ENDAT
External EnDat encoder on the CC (X201 to X210)
CC_MOTOR_ENDAT_INCR
HEIDENHAIN EnDat interface box (EIB) on the CC (X201 to X210)
CC_EXTERN_ENDAT_INCR
HEIDENHAIN EnDat interface box (EIB) on the CC (X201 to X210)
MC_NOT_DISTANCE_CODED_CC_ENDAT
In development
CC_EXTERN_ENDAT_2_2
External EnDat 2.2 encoder on the CC (X201 to X210)

Default: CC_MOTOR_ENCODER
Access: LEVEL3
Reaction: RESET

Connectors X35 to X38, X80 to X83 and X201 to X210 are not available on the MC 420 and MC 320T (only X1 to X4, X5 optional). X35 to X38 are available only on the MC 422x with 10 control loops, X80 to X83 only on the CC 42x with 10 or 12 control loops, and X201 to X210 only on the CC 424.

Note

When EnDat encoders that have not yet been adjusted are put into service, field orientation must be performed prior to operation, see "Field orientation – fundamentals" on page 735.

MP_distPerMotorTurn

Travel of one motor revolution
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 001 to 1000 [mm] or [°]
Default: 5 [mm] or [°]
Access: LEVEL3
Reaction: REF

Note

The parameter **MP_distPerMotorTurn** is not effective for analog axes.

Connecting encoders through the EIB

The EIB (External Interface Box) makes it possible to connect encoders with incremental 1 V_{PP} signals to the purely serial EnDat 2.2 interface of the control. This has the advantage that single-shielded cables with smaller connectors and a smaller diameter can be used as encoder cables.
In **MP_endatSerial**, enter the value TRUE if you want to connect incremental encoders to the EnDat 2.2 interface of the control via an EIB:

MP_endatSerial

Connecting incremental encoders via the EIB
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **TRUE**
An incremental encoder is connected to the serial EnDat 2.2 interface via the EIB.
FALSE
No EIB is used.
Default: -
Access: LEVEL3
Reaction: RUN

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgAxisHardware	
distPerMotorTurn	400004
posEncoderDist	400005
posEncoderIncr	400006
posEncoderRefDist	400007

For any given distance the position encoder supplies a fixed number of signal periods. In addition, the signal is interpolated in the control.

The settings you need to define may vary depending on the type of position measurement on your machine.

Linear encoder

General information

The MANUALplus 620 requires the following values:

- **MP_posEncoderDist**: For each axis, enter the distance required for the number of grating periods given in **MP_posEncoderIncr**.
- **MP_posEncoderIncr**: Enter the number of grating periods required for the distance given in **MP_posEncoderDist**.
- **MP_distPerMotorTurn**: Enter the traverse distance per motor revolution.

The MANUALplus 620 calculates the quotient:

$$\text{Signal period} = \frac{\textbf{MP\_posEncoderDist}}{\textbf{MP\_posEncoderIncr}}$$

For linear encoders with **EnDat 2.1 interface** or linear encoders with **EnDat 2.2 interface supplying a grating period**, the MANUALplus 620 requires the following values:

- **MP_posEncoderDist**: For each axis, enter the length of one grating period of the encoder.
- **MP_posEncoderIncr**: Enter the number of increments per grating period.
- **MP_distPerMotorTurn**: Enter the traverse distance per motor revolution for the axis.

Note

The MC 320 (TNC 320) and MC 320T (MANUALplus 620) main computers do not yet support the connection of EnDat 2.2 encoders.

Linear encoders with an **EnDat 2.2 interface without grating period** may be available in future. In this case, the controller unit will treat the linear encoder as if it had an interpolation of "1." The MANUALplus 620 requires the following values:

- **MP_posEncoderDist**: Enter the **resolution** of the linear encoder (indicated on the ID label, e.g. 0.005 µm).

Note

The resolution always has to be entered in the unit [mm]!

- **MP_posEncoderIncr**: Enter the value 1.
- **MP_distPerMotorTurn**: Enter the traverse distance per motor revolution for the axis.

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

- For encoders with distance-coded reference marks, enter for each axis the nominal increment between two fixed reference marks in **MP_posEncoderRefDist.**

Examples:

- **LS 487C:**
  Incremental linear encoder with distance-coded reference marks. Grating period of 20 µm (1 signal period covers 0.02 mm). Nominal increment between reference marks is 20 mm. The axis moves 10 mm per motor revolution.
  MP_posEncoderDist = 0.02
  MP_posEncoderIncr = 1
  MP_distPerMotorTurn = 10
  MP_posEncoderRefDist = $\frac{20\ \text{mm}}{0{,}02\ \text{mm}}$ =1000 [signal periods]

- **LC 483:**
  Absolute linear encoder with EnDat. Grating period of 20 µm (1 signal period covers 0.02 mm). The axis moves 10 mm per motor revolution.
  MP_posEncoderDist = 0.02
  MP_posEncoderIncr = 1
  MP_distPerMotorTurn = 10

- **LC 483 (EnDat 2.2 without signal period):**
  Absolute linear encoder with EnDat 2.2 without indication of the grating period. Resolution 0.005 µm. The axis moves 10 mm per motor revolution.
  MP_posEncoderDist = 0.000005
  MP_posEncoderIncr = 1
  MP_distPerMotorTurn = 10

Rotary encoders angle encoder, motor encoder

General information

The MANUALplus 620 requires the following values:

- **MP_posEncoderDist**: For each axis, enter the traverse distance per motor revolution.
- **MP_posEncoderIncr**: For each axis, enter the number of grating periods per motor revolution.
- **MP_distPerMotorTurn**: For each axis, enter the traverse distance per motor revolution.

Note

Ensure that the line count per rotary encoder revolution specified by the manufacturer is entered in the **MP_posEncoderIncr** parameter. This value is used for the plausibility check of the measured value. If you enter a line count that differs from the one specified by the manufacturer, the MANUALplus 620 displays an error message.

The MANUALplus 620 calculates the quotient:

$$\text{Signal period} = \frac{\textbf{MP\_posEncoderDist}}{\textbf{MP\_posEncoderIncr}}$$

For **motor encoders with EnDat 2.2 interface,** the MANUALplus 620 requires the following values:

Note

Motor encoders with EnDat 2.2 interface usually do not supply any incremental signals. Only a purely digital measured value is transferred to the MANUALplus 620. For this reason, the number of signal periods per encoder revolution (**MP_posEncoderIncr**) must be set to the value "1".

Note

The MC 320 (TNC 320) and MC 320T (MANUALplus 620) main computers do not yet support the connection of EnDat 2.2 encoders.

The controller unit assumes that the motor encoder has only one line per revolution. Set the following:

- **MP_posEncoderDist**: For each axis, enter the traverse distance per motor revolution.
- **MP_posEncoderIncr**: Enter the value "1".
- **MP_distPerMotorTurn**: For each axis, enter the traverse distance per motor revolution.

▶ **CfgServoMotor - MP_motStr** (401313): Enter the value "1" for the line count of the motor encoder.

Note

HEIDENHAIN recommends:

The optimum manufacturing precision can only be attained with linear encoders. A control without linear encoders finds the axis position through the pitch of the ball screw. The problem is, the ball screw gets hot during machining and expands. And the result is a position measurement that deviates from the actual position, causing error. With linear encoders, the control always determines the correct slide position. This means that your machine positions its axes with constantly high precision.

More information is available on the Internet at:

http://www.heidenhain-shows-the-way.com

Examples:

- **ERN 1387 – Only motor encoder (no linear encoder):**
  Incremental rotary encoder for servo drives. 2048 lines per 360°. The axis moves 10 mm per motor revolution.
  MP_posEncoderDist = 10
  MP_posEncoderIncr = 2048
  MP_distPerMotorTurn = 10

- **RON 785C on rotary table with direct drive:**
  Incremental angle encoder with integral bearing and distance-coded reference marks. 18 000 lines per 360°. 36 reference marks. Nominal increment between reference marks is 20°.
  MP_posEncoderDist = 360
  MP_posEncoderIncr = 18000
  MP_distPerMotorTurn = 360

  MP_posEncoderRefDist = $20 \div \frac{360}{18000}$ =1000 [signal periods]

- **RCN 729 with gear transmission on rotary table:**
  Absolute angle encoder with EnDat interface and integral bearing. 32 768 lines per 360°. Gear transmission ratio: 17:1
  MP_posEncoderDist = 360
  MP_posEncoderIncr = 32768

  MP_distPerMotorTurn = $\frac{360}{17}$ = 21.17647059

- **EQN 1325 – Only motor encoder (no linear encoder):**
  Absolute multiturn rotary encoder for servo drives with EnDat 2.1. 2048 lines per 360°. The axis moves 10 mm per motor revolution.
  MP_posEncoderDist = 10
  MP_posEncoderIncr = 2048
  MP_distPerMotorTurn = 10

- **EQN 1337 – Only motor encoder (no linear encoder):**
  Absolute multiturn rotary encoder for servo drives with EnDat 2.2. The axis moves 10 mm per motor revolution.
  MP_posEncoderDist = 10
  MP_posEncoderIncr = 1
  MP_distPerMotorTurn = 10

For **spindles**, the MANUALplus 620 requires the following values:

- **MP_posEncoderDist**: Enter 360.
- **MP_posEncoderIncr**: Enter the line count of the encoder used.
- **MP_distPerMotorTurn**: Enter 360.

MP_posEncoderDist

Distance for number of signal periods from MP_posEncoderIncr
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 001 to 100 000 [mm] or [°]
For spindles: 360°
For multiturn encoders with EnDat interface:
Distance per encoder revolution
Default: 5 [mm] or [°]
Access: LEVEL3
Reaction: REF

MP_posEncoderIncr

Number of signal periods for distance from MP_posEncoderDist
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 1 to 100 000 [incr.]
Enter the number of increments of the external encoder. For spindles you must enter the line count of the rotary encoder used.
For multiturn encoders with EnDat interface:
Signal periods per encoder revolution
Default: 2048 [incr.]
Access: LEVEL3
Reaction: REF

MP_posEncoderRefDist

Nominal increment between two fixed reference marks
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 1 to 65 535 [signal periods]
Number of signal periods between the reference marks. The parameter is only relevant for encoders with distance-coded reference marks.
Default: 1 000 [signal periods]
Access: LEVEL3
Reaction: REF

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgAxisHardware **posEncoderRefDist**	 400007

The parameter object CfgAxisHardware is not required for:

- Virtual axes (MP_axisMode=Virtual)

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

- For encoders with distance-coded reference marks, enter for each axis the nominal increment between two fixed reference marks in **MP_posEncoderRefDist.**

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

MP_posEncoderDist = 0.02
MP_posEncoderIncr = 1

$$\textbf{MP\_posEncoderRefDist} = \frac{20\ \text{mm}}{0{,}02\ \text{mm}} = 1000\ \text{[signal periods]}$$

MP_posEncoderRefDist

	Nominal increment between two fixed reference marks Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	1 to 65 535 [signal periods] Number of signal periods between the reference marks. The parameter is only relevant for encoders with distance-coded reference marks.
Default:	1 000 [signal periods]
Access:	LEVEL3
Reaction:	REF

External interpolation (only CC 61xx and CC 424)

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgAxisHardware **genExtIntPolFactor**	 400017

If you connect encoders with TTL signals and external interpolation electronics through the TTL/1 V_{PP} adapter to the control:

- In **MP_genExtIntPolFactor**, enter the interpolation factor of the external interpolation unit.

MP_genExtIntPolFactor

	External interpolation for encoder signals Available from NCK software version: 597 110-03.
Format:	Numerical value
Input:	0 to 99 0, 1: No external interpolation (default)
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgAxisHardware	
posEncoderInput	400008
posEncoderSignal	400009
posEncoderFreq	400010
posEncoderResistor	400011
speedEncoderInput	400012
pwmSignalOutput	400013
hsciCcIndex	400014

On the CC 61xx there is a **fixed assignment** between the input of the speed encoder, the PWM output and the input of the position encoder.

- You must wire all connections as listed in the table below.
- Enter the machine parameters MP_posEncoderInput, **MP_speedEncoderInput** and **MP_pwmSignalOutput** as shown in the table.

■ Fixed connector assignment when using a CC 61xx:

PWM output (MP pwmSignalOutput)	Speed input (MP_speedEncoderInput)	Position input (MP_posEncoderInput)
X51	X15	X201
X52	X16	X202
X53	X17	X203
X54	X18	X204
X55	X19	X205
X56	X20	X206

MP_speedEncoderInput

Assignment of a speed encoder input to the axis
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **none**
No speed encoder connected
X15...X20
Speed encoder inputs are X15 to X20
X80...X83
Speed encoder inputs are X80 to X83
Default: none
Access: LEVEL3
Reaction: RESET

MP_pwmSignalOutput

Assignment of the speed command output to the axis
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **none**
Not a PWM output, no power module connected
X51...X56
PWM outputs are X51 to X60
X80...X85
Only on UEC 11x: Motor connections are X80...X85
Default: none
Access: LEVEL3
Reaction: RESET

MP_posEncoderInput

Assignment of a position encoder input to the axis
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **none**
No position encoder connected
X01...X06
Position encoder inputs are X01 to X06 (on the MC 42x)
X35...X38
Position encoder inputs are X35 to X38 (on the MC 42x)
X201...X210
Position encoder inputs are X201 to X210 (on the CC)
Default: none
Access: LEVEL3
Reaction: RESET

Assigning axes to the drive-control motherboard

Use the **MP_hsciCcIndex** parameter to assign the axes to the individual drive-control motherboards of the CC 61xx in the HSCI system. The HSCI address of the respective drive-control motherboard serves as entry in **MP_hsciCcIndex**. The HSCI address is obtained from the position of the drive-control motherboard in the HSCI system. However, the HSCI address to be entered only depends on the drive-control motherboards in the system. I/O units (PLs) and machine operating panels (MBs) are not taken into account. This means that for the first drive-control motherboard, you have to enter the address 0 in **MP_hsciCcIndex**, regardless of whether I/O units or machine operating panels are located before the CC in the HSCI chain.

The parameter is optional and, when it is hidden, automatically has the default value 0. Therefore you do not have to enter anything when using a CC 6106.**MP_pwmSignalOutput** is used for further assignment of the axes to the outputs of the respective drive-control motherboard.

MP_hsciCcIndex

	Index of the CC 61xx controller unit in the HSCI chain Available from NCK software version: 597 110-03.
Format:	Numerical value
Input:	Index of the CC in the HSCI system, e.g. "0"
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

Position encoder signal

Incremental position encoders with 1 V_{PP} signals and absolute encoders with EnDat interface can be connected to the CC 61xx.

Define the type of signal, input frequency and terminating resistance in the following machine parameters.

- **MP_posEncoderSignal:** 1 V_{PP}signal On encoders with EnDat interface you set the signal type 1 V_{PP}.

Note

The CC 61xx does not support encoders with 11 μA_{PP} signal!

- **MP_posEncoderFreq:** Maximum input frequency
- **MP_posEncoderResistor:** Terminating resistor

MP_posEncoderSignal

Signal amplitude at position encoder input
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **1 Vpp**
Input signal of the encoder is 1 V_{PP}
11 µA
Input signal of encoder is 11 μA_{PP}
Default: 1 Vpp
Access: LEVEL3
Reaction: RUN

MP_posEncoderFreq

Input frequency of position encoder inputs
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **fast**
Input frequency is
At 11 V_{PP}: 350 kHz
At 11 μA_{PP}: 150 kHz
slow
Input frequency is 33 kHz
Default: fast
Access: LEVEL3
Reaction: RUN

With **MP_posEncoderResistor** you define whether a terminating resistor is required. (120 ohms)

This parameter may be required if the encoder signals are looped through a drive motor, or if Y cables are used. It is usually sufficient to set the parameter to **without.**

MP_posEncoderResistor

Terminating resistor at position encoder input
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **without**
Without terminating resistor
120 ohms
With resistor
Default: without
Access: LEVEL3
Reaction: RUN

Note

The incremental track data must be entered for the corresponding position encoder inputs for encoders with EnDat interfaces.

6.4.5 Connecting the encoders to the UEC 11x

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgAxisHardware	
posEncoderInput	400008
posEncoderSignal	400009
posEncoderFreq	400010
posEncoderResistor	400011
speedEncoderInput	400012
pwmSignalOutput	400013

On the UEC 11x there is a **fixed assignment** between the input of the speed encoder, the motor connection and the input of the position encoder.

The PWM outputs of the controller unit integrated in the UEC 11x are connected internally with the inverter and are not routed toward the outside. Therefore you have to enter the number of the motor connection (X80 to X84) in the **MP_pwmSignalOutput** parameter.

- ▶ You must wire all connections as listed in the table below.
- ▶ Enter the machine parameters MP_posEncoderInput, **MP_speedEncoderInput** and **MP_pwmSignalOutput** as shown in the table.

■ Fixed connector assignment when using a UEC 11x:

Motor connection (MP pwmSignalOutput)	Speed input (MP_speedEncoderInput)	Position input (MP_posEncoderInput)
X80	X15	X201
X81	X16	X202
X82	X17	X203
X83	X18	X204
X84	X19	X205

MP_speedEncoderInput

Assignment of a speed encoder input to the axis
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **none**
No speed encoder connected
X15...X20
Speed encoder inputs are X15 to X20
X80...X83
Speed encoder inputs are X80 to X83

Default: none

Access: LEVEL3

Reaction: RESET

MP_pwmSignalOutput

Assignment of the speed command output to the axis
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **none**
Not a PWM output, no power module connected
X51...X56
PWM outputs are X51 to X60
X80...X85
Only on UEC 11x: Motor connections are X80...X85

Default: none

Access: LEVEL3

Reaction: RESET

MP_posEncoderInput

Assignment of a position encoder input to the axis
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **none**
No position encoder connected
X01...X06
Position encoder inputs are X01 to X06 (on the MC 42x)
X35...X38
Position encoder inputs are X35 to X38 (on the MC 42x)
X201...X210
Position encoder inputs are X201 to X210 (on the CC)

Default: none

Access: LEVEL3

Reaction: RESET

6.4.6 Defining the traverse direction

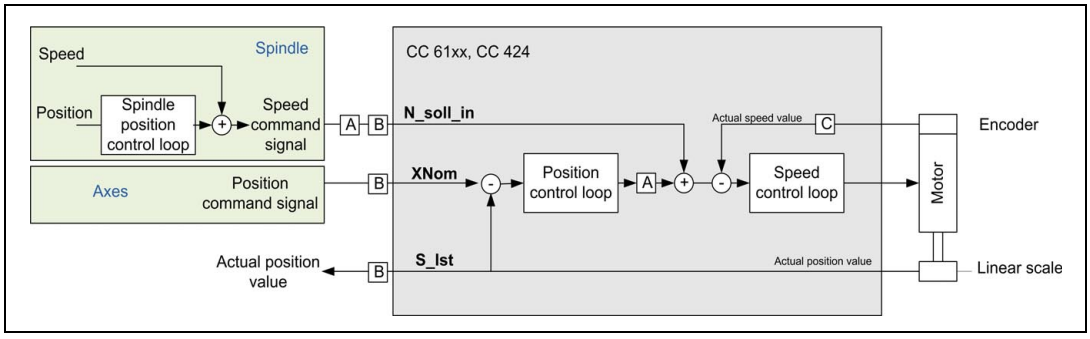
Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgAxisHardware	
signCorrActualVal	400001
signCorrNominalVal	400002
CfgSpindle	
changeTurnDir	401509

The parameter object CfgAxisHardware is not required for:

- Virtual axes (MP_axisMode=Virtual)

When using the CC 61xx, define the counting directions and the traverse direction of the axis, depending on the mounting attitude of the encoders.

Block diagram: Defining the traverse direction



Legend:

Signal	Parameter	Entry / Effect
A	MP_signCorrActualVal	Invert the value if the counting direction of the position encoder is opposite to the direction of rotation of the motor.
B	MP_signCorrNominalVal	Changing the value inverts the nominal and actual values of the position encoder. The axis changes the traverse direction.
C	DIR column from motor table	Counting direction of the motor encoder

Direction of rotation for spindles with C axis

If a spindle is also to be used as a C axis, the rotational direction of the C axis must first be set in the parameters **MP_signCorrActualVal** and **MP_signCorrNominalVal** in such a way that the angle moves in the mathematically positive sense (counterclockwise) if the physical spindle axis rotates in the mathematically negative sense (i.e. clockwise).
Since, if there is no separate drive for the C axis, the two parameters also influence the spindle, the rotational direction of the spindle might need to be corrected if M3 and M4 are used.
Therefore, if the functions M3 and M4 are called, correct the rotational direction of the spindle in **MP_changeTurnDir**, as well as the status display (M3/M4) of the spindle in the dashboard.

MP_signCorrActualVal

Reversal of the algebraic sign of the position encoder signal
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **on**
The algebraic sign of the position encoder is inverted
off
The algebraic sign of the position encoder is not inverted
Default: off
Access: LEVEL3
Reaction: REF

MP_signCorrNominalVal

Reversal of the algebraic sign of the nominal speed value
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **on**
The algebraic sign of the nominal speed value is inverted
off
The algebraic sign of the nominal speed value is not inverted
Default: off
Access: LEVEL3
Reaction: REF

Note

The counting direction of the motor encoder for speed control is defined in the motor table (DIR column). An incorrect entry results in the error message **C3B0 Motor does not rotate <AXIS>** appearing.

MP_changeTurnDir

Rotational direction reversal with M3 and M4
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **TRUE**
Reversal of rotational direction is active with M3/M4
FALSE
Reversal of rotational direction is not active with M3/M4
Default: TRUE
Access: LEVEL3
Reaction: RUN

6.4.7 Encoder monitoring

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgEncoderMonitor	
checkAbsolutPos	400701
checkSignalLevel	400702
checkFrequency	400703
checkRefDistance	400704

HEIDENHAIN contouring controls monitor the signal transmission from the encoders.

The parameter object **CfgEncoderMonitor** is not required for:

- Virtual axes (MP_axisMode = Virtual)

Position encoder

Activate the following monitoring functions for the position encoders:

- **MP_checkAbsolutPos**:
  Monitor the absolute position of position encoders with distance-coded reference marks
- **MP_checkSignalLevel**:
  Monitor the encoder signal level of the position measuring systems
- **MP_checkFrequency**:
  Monitor the edge separation of the position measuring systems

Note

The edge separation is monitored by the CC controller unit. Therefore, the **MP_checkFrequency** parameter is currently not available on the TNC 320 and the MANUALplus 620 with analog command interface to the drive motors.

- **MP_checkRefDistance**:
  Monitor the reference mark of the spindle encoder (distance between zero crossovers of the spindle)

The interpolator calculates the absolute position when a reference mark of a distance-coded encoder is crossed over. If **MP_checkAbsolutPos** is active, the nominal values are compared to the actual values. If deviations are found, an error message is displayed and an EMERGENCY STOP is initiated.

MP_checkAbsolutPos

Monitoring the absolute position with distance-coded encoder
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **on**
Monitor the absolute position
off
No monitoring
Default: off
Access: LEVEL3
Reaction: RESET

MP_checkSignalLevel

Monitoring the encoder amplitude
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **on**
Monitor the encoder amplitude
off
No monitoring
Default: on
Access: LEVEL3
Reaction: RUN

MP_checkFrequency

Monitoring the edge separation of the position encoders
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **on**
Monitor the encoder frequency
off
No monitoring
Default: off
Access: LEVEL3
Reaction: RUN

Criterion	Error message
Absolute position with distance-coded reference marks	**Encoder <AXIS> DEFECTIVE**
Amplitude of encoder signals	**Encoder AMPLITUDE TOO LOW <AXIS>**
Edge separation of encoder signals	**Encoder <AXIS>: FREQUENCY TOO HIGH**

Monitoring of encoders with EnDat interface

In the event of a disturbance, the error message **EnDat defective <error code> <axis>** will appear.

The error code is shown in hexadecimal notation. Error codes may also appear combined, in which case they are added together.

There are two possible types of errors:

- The encoder reports an error.
- Access to the encoder via the EnDat interface is faulty.

Codes for errors reported by the encoder:

Error code	Meaning
0x00000001	Light source defective
0x00000002	Signal amplitude too low
0x00000004	Incorrect position value
0x00000008	Overvoltage
0x00000010	Undervoltage
0x00000020	Overcurrent
0x00000040	Replace the battery
0x00000080	Reserved
0x00000100	Reserved
0x00000200	Reserved
0x00000400	Reserved
0x00000800	Reserved
0x00001000	Reserved
0x00002000	Reserved
0x00004000	Reserved
0x00008000	Reserved

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

Error code	Meaning
0x80010000	Delete the alarm bit
0x80020000	Read the alarm status
0x80040000	Read the number of pulses
0x80080000	Read the number of signal periods
0x80100000	Read the number of differentiable revolutions
0x80200000	Read the measuring steps
0x80400000	Read the serial number
0x80800000	Read the type of encoder
0x81000000	Read the position value
0x82000000	Reserved
0x84000000	Reserved
0x88000000	Read the checksum
0x90000000	Alarm bit remains set
0xA0000000	Timeout while waiting for data—signal "high"
0xC0000000	Timeout while waiting for data—signal "low"
0x80000000	Error during access to EnDat interface

Speed encoder

The MANUALplus 620 uses the **Type of encoder** entry in the "motor.mot" motor table. If an encoder with Z1 track is entered in the motor table, the message **C310 Z1 track error** appears in the event of an error. If an encoder with EnDat interface is entered in the motor table, the MANUALplus 620 attempts to communicate with the encoder. If this fails, the error message **C3F0 EnDat not found <axis>** appears.

Attention

If you use the HEIDENHAIN standard motor table motor.mot and motors with EnDat encoders, you might have to change the entry for the motor in the SYS column (type of encoder) of the motor table or enter a new motor.

- SYS = 1: Incremental rotary encoder with Z1 track
- SYS = 2: Absolute speed encoder with EnDat interface

6.5 Reading and Writing Axis Information

6.5.1 Reading axis information

Module 9038 Read the status information of axes
With Module 9038 you can interrogate the general status information of the axes. You can interrogate the status of a specific axis or of all axes at once.

With bit-coded information, the status request for a specific axis returns code 0 or 1. The meaning of the return codes is explained in the table below.

The desired information can be read for all axes in one run-through. For this the axis number –1 must be transferred. In this case the status request only returns bit-coded information. The information is then passed on in the bit corresponding to the axis.

Status information	Bit information	Meaning
0	x	0: Axis not active (MP_axisMode is not "Active" or no encoder) 1: Axis active
1	x	Axis in interpolation context? 0: Axis is currently in interpolation context or not active. 1: Axis is currently not in interpolation context or this axis is a spindle. (Spindles are not in interpolation context.)
2	x	0: Open-loop axis 1: Closed-loop axis (MP_axisMode=Active)
3	–	Maximum temperature of the motor [°C]
4	x	0: Not a Hirth axis 1: Hirth axis
5	–	Hirth grid [1/10 µm]
6	–	Reserved
7	x	0: Linear axis or not active 1: Rotary axis (MP_isAng=True)
8	x	0: Analog axis (MP_axisHw=Analog) or not active 1: Digital axis
9	x	0: Axis is not a slave axis 1: Axis is a slave axis

Call:

PS	B/W/D/K	<Axis> Axis-specific: Index from MP_CfgAxes/axisList For all axes: –1
PS	B/W/D/K	<Status information> See table above
CM	9038	
PL	B/W/D	<Information> Axis-specific: Status information according to table For all axes: Bit-coded (Bit 0 corresponds to logic axis 0, etc.)

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Information was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Status information not available
	2	Axis does not exist

Module 9049 Read position value and speed value of an axis

Module 9049 reads the position value or speed value of an axis

Call:

PS	B/W/D/K	<Logical axis number>
PS	B/W/D/K	<Desired axis information> 2: Actual position in the reference system 3: Following error 8: Offset of the axis due to kinematic compensation in real time 9: Distance traversed [mm] since the last lubricating pulse 10: Actual speed 11: Nominal speed
CM	9049	
PL	D	<Axis information> For 2, 3, 8: Value in 0.0001 mm or 0.0001° For 9, 10, 11: Value in mm/min or °/min

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Axis information read
	1	No axis information read, error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Interrogated axis number or axis information invalid

Reading the axis coordinates

▶ Read the axis coordinates with Module 9040 (only API 1.0) or Module 9041.

The values are saved in double words beginning at the given address.

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

The coordinate value of an axis remains undefined until the reference point of an axis has been traversed.

Module 9040 Reading of axis coordinates by the PLC in the format 1/1000 (0.001) mm

Module 9040 loads the axis coordinates from the control loop for all NC axes. The actual values in the reference system, the servo lag, the distance-to-go and the deflection of a triggering touch probe can be loaded.

The values are saved in 10 double words in the format 1/1000 mm, beginning at the given target address.

The module is only supported if you use the iTNC-compatible programming interface (API 1.0). The API 1.0 cannot be used on the MANUALplus 620.

Note

This PLC module was introduced in order to remain compatible with older PLC programs (with API version 1.0) of older HEIDENHAIN contouring controls. This PLC module is not supported if the symbolic programming interface is used! Use Module 9041 instead.

Possible errors:

- The argument for the type of coordinate is outside the permitted range (2).
- The specified target address is not a double word address (4).
- The double word block cannot be written to the specified target address (4).
- You are using the symbolic programming interface.

Call:

```
PS    K/B/W/D  <Target address Dxxxx>
PS    K/B/W/D  <Type of coordinate>
                 2: Actual values in the reference system
                 3: Following error
                 4: Distance-to-go
                 5: Deflection (measuring touch probe)
                 6: Actual values in the datum system
                 8: Temperature compensation
                 9: Distance counter [mm]
CM    9040
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Data was read
	1	Faulty call data

Module 9041 Reading of axis coordinates by the PLC in the format 1/10000 (0.0001) mm

Module 9041 loads the axis coordinates from the control loop for all NC axes. The actual values in the reference system, the servo lag, the distance-to-go and the deflection of a triggering touch probe can be loaded.

The values are saved in 10 double words in the format 1/10000 mm, beginning at the given target address.

Possible errors:

- The argument for the type of coordinate is outside the permitted range (2).
- The specified target address is not a double word address (4).
- The double word block cannot be written to the specified target address (4).

Call:

PS K/B/W/D <Target address Dxxxx>
PS K/B/W/D <Type of coordinate>
2: Actual values in the reference system
3: Following error
4: Distance-to-go
5: Deflection (measuring touch probe)
6: Actual values in the datum system
7: Actual values in the reference system with backlash
8: Temperature compensation
9: Distance counter
CM 9041

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Data was read
	1	Faulty call data

Reading the actual spindle values

▶ Read the spindle coordinates with Module 9411.

Note

You can use this module only if you are working with the new symbolic API, see "The API 3.0 symbolic memory interface" on page 1090.

Module 9411 Read the actual spindle values (speed, coordinates)
Use Module 9411 to read the position and speed values of the spindle.

This module is only supported by the symbolic memory interface (API 3.0). If you are using the iTNC-compatible memory interface (API 1.0), the module returns the error code 99.

In order to receive the "Request switchover" status value, the speed-dependent wye/delta switchover (Module 9417) must be active.

Call:

PS	B/W/D/K	<Spindle index> 0: Spindle 1 1: Spindle 2 etc.
PS	B/W/D/K	<Desired spindle information> 1: Actual position 2: Nominal position 3: Following error 10: Actual speed 11: Nominal speed 20: Information about wye/delta operation
CM	9411	
PL	D	<Spindle information> For 1 to 3: Value in 0.0001° For 10 to 11: Value in 0.001 rpm For 20: 0: Spindle in wye operation 1: Request for wye/delta switchover 2: Spindle in delta operation 3: Request for delta/wye switchover

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Actual spindle value read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Interrogated spindle number or spindle information invalid
	99	Module is not supported (control operates with iTNC-compatible API).

6.5.2 Writing axis information—activating and deactivating axes

This chapter describes how to activate and deactivate axes during operation and without rebooting the MANUALplus 620, e.g. in order to exchange milling heads.

The MANUALplus 620 has two machine parameters in the CfgAxis configuration object. They describe the type of connection and the behavior of axes:

- **MP_axisHw** (300104)
- **MP_axisMode** (300105)

Note

For activation and deactivation of axes to be possible at all, the **MP_axisMode** parameter has to have the value **Active**.

With **MP_axisMode** you specify whether the affected axis is known to the system (= Active) or not (= NotActive). If axes are not known, there is also no need for any further parameters. Virtual axes (= Virtual) and axes controlled by the PLC (= PlcControlled) are ignored in the checks for activating and deactivating.

With the **MP_axisHw** parameter you define the hardware type of the axis at the time of the control startup. For the settings possible, see "MP_axisHw" on page 406.

Axes connected to the MANUALplus 620 have different activation statuses:

Activation status	Parameter MP_axisHw	Data required in the parameter set	Functions
Closed-loop axis	InOutCC AnalogMC AnalogCC ProfiNet	There must be at least one parameter set with the controller parameters for the axis. The digital CC parameters are not required for analog axes.	Fully functional control loop, programmable physical axis
Manually operated axis	ManualMC ManualCC	All parameters except the controller parameters must be available.	Axis with a linear encoder but without drive motor—nevertheless programmable. Behaves like an active axis with respect to the PLC (axis is included in the geometry description)
Displayed axis	DisplayMC DisplayCC	All parameters except the controller parameters must be available.	Axis with a linear encoder but without servo drive—not programmable. (Axis included in the geometry information.)
Deactivated axis	(Axis is deactivated either over Module 9226 or 9418, or over the parameter MP_deactivatedAtStart.)		No monitoring active. Encoder and drive motor can be removed mechanically.

The following table shows the assignment of available and configurable monitors of the axes to the various activation statuses.

Activation status	Monitoring active	Monitoring inactive
Closed-loop axis	■ Following error monitoring ■ Standstill monitoring ■ Movement monitoring ■ Monitoring of the amplitude of the encoder ■ Monitoring of the frequency of the encoder ■ Positioning window	
Manually operated axis	■ Monitoring of the amplitude of the encoder ■ Monitoring of the frequency of the encoder ■ Positioning window	■ Following error monitoring ■ Standstill monitoring ■ Movement monitoring
Displayed axis	■ Monitoring of the amplitude of the encoder ■ Monitoring of the frequency of the encoder ■ Positioning window	■ Following error monitoring ■ Standstill monitoring ■ Movement monitoring
Deactivated axis		■ Following error monitoring ■ Standstill monitoring ■ Movement monitoring ■ Monitoring of the amplitude of the encoder ■ Monitoring of the frequency of the encoder ■ Positioning window

Switching parameter sets

If when activating or deactivating axes over the PLC you also switch parameter sets (e.g. when exchanging milling heads), you have to observe particular conditions (See "Switching Parameter Sets" on page 306).

Switching the activation status

The activation status of an axis is switched through PLC Module 9226, and that of the spindle through PLC Module 9418. It must be kept in mind that the activation status cannot be increased beyond that configured in **MP_axisHw**. An axis that is only displayed (**MP_axisHw** = DisplayMC or DisplayCC), for example, cannot be switched up to the "manually operated axis" or "closed-loop axis" activation status. The following table shows the permissible changes of the activation status.

Parameter MP_axisHw	**Permissible change of the activation status**			
	Closed-loop axis	**Manually operated axis**	**Displayed axis**	**Deactivated axis**
InOutCC AnalogMC AnalogCC Profinet	X	X	X	X
ManualMC ManualCC	–	X	X	X
DisplayMC DisplayCC	–	–	X	X

For example, proceed as follows if you want to activate individual axes without rebooting:

- Switch off the servo control of the motor.
- Through Module 9226 for axes and 9418 for the spindle, use mode 14 to set the motor to the "deactivated axis" activation status. This also deactivates all monitors of the drive motor. Now the servo motor can be mechanically removed, for example to exchange a milling head.

Reverse the sequence if you want to reactivate the previously deactivated axis.

- With Module 9226 or 9418, set the axis or spindle to the desired activation status (mode 11 to 13)
- The servo control of the motor can now be activated.

Note

After the axis is activated, the interpolator automatically conducts an actual-to-nominal value transfer through mode 11 to 13.

Deactivate axis during startup

With the **MP_deactivatedAtStart** parameter you can deactivate an axis or spindle already during startup and then activate it later for control operation.

Note

For all axes that are removable, (e.g. exchangeable milling heads), HEIDENHAIN recommends setting the **MP_deactivatedAtStart** parameter to the value **TRUE**. In this case, the PLC must ensure that the respective physically installed axes are correctly activated.

An axis that was deactivated through **MP_deactivatedAtStart** behaves as if you have deactivated the axis through Module 9226 or 9418 with mode 14. A change of the parameter does not go into effect until the next startup of the MANUALplus 620.

MP_deactivatedAtStart

Deactivate axis during startup
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: **TRUE**
During startup of the MANUALplus 620 the axis is deactivated—irrespective of the **MP_axisHw** parameter. The same effect is attained with the parameter as from calling PLC module 9226 or 9418 with mode 14.
FALSE
During startup of the MANUALplus 620 the axis is treated as configured in the parameter axisHw.

Default: No value, parameter optional (= behavior as for FALSE)

Access: LEVEL3

Reaction: RUN

Module 9226 Set status for axes

The module sets a new status for an NC axis or auxiliary axis (not for spindles).

Constraints for modes 10 to 14:

- The motor of the affected axis must be switched off during module call. The current and speed controllers must be inactive.
- The activation status must not be switched except in a strobe with synchronization of the advance calculation (SYNC_CALC, interpreter stop).

Call:

PS B/W/D/K <Axis>
Index from the parameter CfgAxes/axisList

PS B/W/D/K <Mode>
1: Renewed evaluation of the axis reference mark
10: Restore activation status from startup
11: **Closed-loop axis** activation status
12: **Manually operated axis** activation status
13: **Displayed axis** activation status
14: **Deactivated axis** activation status

CM 9226

PL B/W/D <Error code>
0: Module successfully executed
1: Faulty module call (invalid axis number)
2: Faulty module call (invalid mode)
3: Faulty module call (axis is not an NC axis or auxiliary axis)
4: Axis is controlled by the NC
5: Activation status is being switched
6: Activation status is already selected

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Module executed successfully
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Invalid task data transferred (see error number in returned value)

Module 9418 Set status for spindle

The module sets a new status for a spindle.

Constraints:

- This module is only supported by the symbolic memory interface (API 3.0). If the iTNC-compatible memory interface (API 1.0) is used, the module returns an error.

Constraints for modes 10 to 14:

- The motor of the spindle must be switched off during module call. The current and speed controllers must be inactive.
- The activation status must not be switched except in a strobe with synchronization of the advance calculation (SYNC_CALC, interpreter stop).

Call:

PS B/W/D/K <Spindle index>
PS B/W/D/K <Mode>
1: Renewed evaluation of the spindle's reference mark
2: Determine the actual speed value for speed control from the motor encoder
3: Determine the actual speed value for speed control from the position encoder
10: Restore activation status from startup
11: **Closed-loop axis** activation status
14: **Deactivated axis** activation status
CM 9418
PL B/W/D <Error code>
0: Module successfully executed
1: Faulty module call (invalid spindle number)
2: Faulty module call (invalid mode)
3: Activation status is being switched
4: Activation status is already selected

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Module executed successfully
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid task data transferred (see error number in returned value)
	99	Module is not supported (control operates with iTNC-compatible API).

Module 9155 Switch axes from closed-loop to open-loop condition
Module 9155 switches axes from the closed-loop to open-loop state.

Note

This module was implemented to ensure compatibility with earlier HEIDENHAIN contouring controls. HEIDENHAIN recommends:
Use preferably the Modules 9226 and 9418 described above.

Constraints:

- The module functions only in the cyclic PLC program.
- The function is only possible when the MANUALplus 620 is not active (NN_ChnControlInOperation = 0) or if an M/S/T/T2/G strobe is pending.
- The function is only possible when the MANUALplus 620 is not active or if an M/S/T/T2/G strobe is pending.
- Only the last module call per PLC cycle is included.

Call:
PS B/W/D/K <bit-coded axis mask>
CM 9155

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Module executed successfully
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid axis mask given
	21	Missing strobe or control is active.
	24	Module was called in a submit/spawn job

Module 9156 Switch axes from open-loop to closed-loop condition
Module 9156 switches axes that were previously switched by Module 9155 to the open-loop state back to the closed-loop state.

Note

This module was implemented to ensure compatibility with earlier HEIDENHAIN contouring controls. HEIDENHAIN recommends:
Use preferably the Modules 9226 and 9418 described above.

Constraints:

- The module functions only in the cyclic PLC program.
- The function is only possible when the MANUALplus 620 is not active (NN_ChnControlInOperation = 0) or if an M/S/T/T2/G strobe is pending.
- Only the last module call per PLC cycle is included.

Call:
PS B/W/D/K <bit-coded axis mask>
CM 9156

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Module executed successfully
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid axis mask given
	21	Missing strobe or control is active.
	24	Module was called in a submit/spawn job

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgPositionLimits	
swLimitSwitchPos	400501
swLimitSwitchNeg	400702

The parameter object CfgPositionLimits is not required for:

- Virtual axes (MP_axisMode = Virtual)
- Axes that are for display only (MP_axisMode = Display)

Define the software limit switches in the parameter object **CfgPositionLimits:**

- The datum is the machine datum (MP_refPosition).
- If the geometry detects that a software limit switch will be traversed, the traverse path concerned will not be executed and an error message will be displayed.
- If a software limit switch is traversed, the MANUALplus 620 stops the corresponding axis and displays an error message. The axis can be retracted in the opposite direction, however.
- On milling controls, the software limit switches can usually be overwritten from the NC program (see below).
- On lathe controls, the software limit switches can only be overwritten from the PLC.
- Limit-switch monitoring can be deactivated by entering 0 for positive and negative limit values.

MP_swLimitSwitchPos

Positive software limit switch
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 000 000 to +100 000 [mm] or [°]
Default: 0 [mm] or [°]
Access: LEVEL3
Reaction: RUN

MP_swLimitSwitchNeg

Negative software limit switches
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 000 000 to +100 000 [mm] or [°]
Default: 0 [mm] or [°]
Access: LEVEL3
Reaction: RUN

Note

If positive and negative software limit switches = 0, monitoring is switched off.

6.7 Lubrication Pulse

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgPositionLimits **lubricationDist**	 400503

The parameter object CfgPositionLimits is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)

- In **MP_lubricationDist,** you define the traverse distance at which the lubrication pulse for the axis guideways is to be output. The NC reports in **NN_AxLubricationPulse** when the entered distance in an axis has been exceeded.
- Reset **PP_AxLubricationDistReset** after lubrication. This resets the distance counter to 0.

Note

After the MANUALplus 620 has been reset, the accumulated distance is reset.

MP_lubricationDist

Path-dependent lubrication of axis
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to + 100 000 [mm] or [°]
0= no output of lubrication pulse to PLC
Default: 100 [mm] or [°]
Access: LEVEL3
Reaction: RUN

PLC operand / Description	Type
NN_AxLubricationPulse Lubrication pulse: Value in MP_lubricationDist exceeded 0: Value not exceeded 1: Value exceeded	M
PP_AxLubricationDistReset Reset the accumulated distance: 0: Do not reset accumulated distance 1: Reset accumulated distance	M

Axes that are **not** in an interpolation context can be used by the PLC as required. The PLC can start more than one axis simultaneously, but they are not interpolated with each other.

Note

Please keep in mind that the axis interpolation context can by changed by the PLC at any time. Example: By activating another kinematics model.

Stopping/Starting axes by PLC

Module 9120 Position PLC axis

The module positions an axis. The target position and feed rate are transferred in the module call.

The axis is positioned regardless of any other processes in the control. In particular, there is no interpolation with other axes.

Constraints:

- The axis must **not** be in an interpolation context.
- The parameter values for rapid traverse, acceleration, etc. must be set correctly.
- Rotary axes are positioned in the direction of the shortest path, except if the target position was transferred as an incremental value.
- Software limit switches are not active.
- The axis must be stationary. Any positioning movement must be aborted beforehand with Module 9121.
- Feed-rate override is disabled.
- If no reference mark has been traversed, the positioning process builds on the counter value as it was upon switch-on.
- If Modules 9120, 9121 and 9122 are called more than once for the same axis during one PLC scan, only the last command is transferred.
- A "positioning error" status set in this axis is cleared. The status must be evaluated by Module 9122.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
PS	B/W/D/K	<Target position> Input unit: [0.0001 mm]
PS	B/W/D/K	<Feed rate> Input unit: [mm/min]
PS	B/W/D/K	<Mode> Bit 0 – Definition of the target position: 0: Absolute, i.e. relative to the machine datum 1: Incremental
CM	9120	
PL	B/W/D	<Error code> 0: No error. Positioning was started. 1: Axis does not exist 2: Axis is still in interpolation context 3: Axis is already being positioned 4: Absolute position is outside of modulo range 5: Programmed axis not in closed loop

Module 9121 Stop PLC axis

The module stops a positioning movement that has been started by Module 9120 or 9123.

Constraints:

- If Modules 9120, 9121 and 9122 are called more than once for the same axis during one PLC scan, only the last command is transferred.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
CM	9121	
PL	B/W/D	<Error code> 0: Positioning is canceled 1: Axis does not exist 2: Axis is still in interpolation context 3: Axis was already stationary

Module 9122 Status of PLC axis

The module provides information on the present operating status of the axis.

Constraints:

- Status changes through a PLC positioning command (Modules 9120, 9121, 9123) are not detected until the next PLC scan.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
CM	9122	
PL	B/W/D	<Status> Bit 0 – Axis in interpolation context? 0: Axis does not exist or is in interpolation context 1: Axis is not in interpolation context Bit 1 – Reference mark 0: Reference mark not yet traversed 1: Reference mark traversed Bit 2 – Positioning 0: Positioning inactive 1: Positioning active Bit 3 – Direction of motion 0: Positive direction of motion 1: Negative direction of motion Bit 4 – Positioning error 0: No positioning errors occurred 1: Positioning error Bit 5 – Closed-loop or open-loop axis 0: Closed-loop axis 1: Open-loop axis Bit 6 – Target position reached? 0: Target position not yet reached 1: Target position reached

Module 9123 Traverse the reference marks of PLC axes

The module starts a positioning movement in a defined direction. The positioning movement is continued until a reference mark is found or until the positioning movement is canceled by Module 9121.

Note

Use Module 9123 only if no conventional procedure for traversing the reference marks is possible.

Constraints:

- The axis must **not** be in an interpolation context.
- The parameter values for rapid traverse, acceleration, etc. must be set correctly.
- Software limit switches are not active.
- The axis must be stationary. Any positioning movement must be aborted beforehand with Module 9121.
- Feed-rate override is disabled.
- If Modules 9120, 9121 and 9122 are called more than once for the same axis during one PLC scan, only the last command is transferred.
- A "positioning error" status set in this axis is cleared.
- The "find reference point" status is set for the axis.
- Any pre-existing reference point in this axis is cleared, but the numerical axis value remains. It will not be reinitialized until the reference point is found.
- The positioning movement is interrupted as soon as the reference point is found. However, due to the braking distance, the axis comes to a standstill somewhat beyond the reference mark.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
PS	B/W/D/K	<Feed rate> Input unit: [mm/min]
PS	B/W/D/K	<Mode> Bit 0: Direction of traverse 0: Positive 1: Negative
CM	9123	
PL	B/W/D	<Error code> 0: No error. Positioning was started. 1: Axis does not exist 2: Axis is still in interpolation context 3: Axis is already being positioned 5: Programmed axis not in closed loop

Module 9124 Feed rate override for PLC axis

The override value set in this module influences the traversing speed of an axis traversed by the PLC with Module 9120 or 9123.

Constraints:

- The axis must **not** be in an interpolation context.
- The override value is transferred as integral number (0 to 10000), which may be in the range from 0% to 100.00% (resolution 0.01%).
- 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 is set to 100.00%.
- The override value can be changed during positioning.
- The module can be called in addition to a module from the group (9120/ 9121/9123) during the same PLC scan.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
PS	B/W/D/K	<Override> Input unit: 0 to 10 000, corresponds to 0 to 100% in 0.01% steps.
CM	9124	
PL	B/W/D	<Error code> 0: No error, override value was set 1: Axis does not exist 2: Not a PLC axis 3: Override value incorrect

Positioning of axes by PLC

You start a PLC positioning movement with Module 9221, and you can interrogate the status with Module 9222.

The following conditions apply to a PLC positioning command:

- Tool compensation is not included. Before a PLC positioning command you must end any tool compensation.
- A PLC positioning movement is not displayed in the test graphic.

The NC cancels a PLC positioning movement under the following conditions:

- If in the Manual or Handwheel modes there is an NC STOP.
- If in the automatic operating modes there is an NC STOP and "internal stop."
- If there is an EMERGENCY STOP.
- If there is an error message that results in a STOP.

Module 9221 Start a PLC positioning movement

The module positions an axis. The target position and feed rate are transferred in the module call. Limit switch interrogation can be activated in a separate transfer parameter.

The axis is positioned regardless of any other processes in the control. In particular, there is no interpolation with other axes.

Note

When calling the module for an NC axis during a strobe, the synchronization with the advance calculation (strobe with **MP_sync** = SYNC_CALC) must be configured for this strobe.

Constraints:

- The module must only be called if no program is running, or if an M/G/S/T/T2/Q strobe is pending. No axis direction key may be pressed in the Manual operating mode.
- For rotary axes with transition to zero, positioning is by the shortest path.
- If you wish to change a parameter (e.g. target position, feed rate) of a positioning command already in progress, you must first abort positioning, then change the parameter and start again.
- A simultaneous PLC positioning movement of several axes is interpolated. If you start an additional axis while already positioning another, the first movement is aborted, and then all the programmed axes (e.g. X, Y and Z) are positioned together.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
PS	B/W/D/K	<Target position> Input unit: [0.0001 mm]
PS	B/W/D/K	<Feed rate> Input unit: mm/min
PS	B/W/D/K	<Mode> Bit 0 – Definition of the target position: 0: Absolute, i.e. relative to the machine datum 1: Incremental Bit 1 – Software limit switch: 0: Inactive 1: Active
CM	9221	
PL	B/W/D	<Error code> 0: Positioning is being started 1: Axis is not in a closed loop or is an auxiliary axis 2: Inadmissible values for the feed rate 3: Axis has not traversed the reference mark 4: No M/S/T/Q strobe during running program 5: Programmed axis not in closed loop 6: Positioning already started

Module 9222 Status request of a PLC positioning movement
The module provides the PLC positioning status.

Status information is collected for a certain axis or for all axes. This includes the current status of PLC positioning movements. The respective status information of a certain axis, or bit-encoded for all axes, can be interrogated.

Constraints:

- The status of an axis remains until a new status is set when the next PLC positioning of this axis occurs.

Call:

PS B/W/DK <Axis>
Index from MP_CfgAxes/axisList
or bit-encoded output for all axes:
1: Target position reached
2: PLC positioning was started
3: PLC positioning canceled
4: Limit switch
5: Positioning impossible
6: Positioning temporarily stopped

CM 9222

PL B/W/D <Status>
0: No PLC positioning was started
1: Target position reached
2: PLC positioning was started
3: Due to cancelation, target position not reached
4: Target position is outside of traverse range
5: Positioning not possible (e.g. due to "free rotation")
6: PLC positioning temporarily stopped (stop in Automatic operating modes)

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Status was transferred
	1	Error (error code in NN_GenApiModuleErrorCode)
NN_GenApiModuleErrorCode	1	Invalid status information was requested
	2	The status of an open-loop axis, auxiliary axis or slave axis is being interrogated

Module 9224 Stop PLC positioning movements
The module stops the positioning movement of an NC axis that has been started by Module 9220 or 9221.

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

Call:

PS	B/W/DK	<Axis> Index from MP_CfgAxes/axisList
PS	B/W/DK	<Mode> 0: Mode is not used at present
CM	9224	
PL	B/W/D	<Error code> 0: Stop PLC positioning 1: Invalid axis number 2: Invalid axis type 3: Axis is not in motion 4: Axis is controlled by NC 5: Invalid mode

Module 9227 Position auxiliary axes and NC axes
The module starts the positioning of an NC axis (PLC positioning like with Module 9221) or the positioning of an auxiliary axis (like with PLC Module 9120). With Module 9227, acceleration and jerk can be programmed in addition to the target position and feed rate parameters.

Further modes can be programmed for positioning NC axes (like with Module 9221):

- Rapid-traverse feed rate (only operating panels with rapid traverse override)
- Active limit switch interrogations
- Deactivated collision monitoring

See also documentation for PLC Modules 9120 and 9221

Conditions:

- The machine parameter configuration of an axis to be started with Module 9227 determines whether the positioning movement of an auxiliary axis or an NC axis is started.

Positioning of NC axes (PLC positioning):

- The module can only be called if no program is running, or if an M/G/S/T/T2/Q strobe is pending. No axis direction key may be pressed in the Manual operating mode. The entered positions are referenced to the machine datum. Rotary axes with transition to zero are positioned by the shortest path.
- If you wish to change a parameter (e.g. target position, feed rate) of a positioning command already in progress, you must first abort positioning, then change the parameter and start again.
- A simultaneous PLC positioning movement of several axes is interpolated. If you start an additional axis while already positioning another, the first movement is aborted, and then all the programmed axes (e.g. X, Y and Z) are positioned together.
- As soon as a PLC positioning with rapid traverse is active (bit 2 is set), all active PLC positioning movements are at rapid traverse, and instead of the feed-rate override the rapid-traverse override is effective.
- Error code 7 not possible.

Positioning auxiliary axes:

- Axes with automatic reduction (modulo value in MP_isModulo) are always moved to the target position in the direction of the shortest traverse, unless the target position was given as an incremental value.
- The system does not check for limit switch overshoot.
- The axis must be stationary. Any positioning movement must be interrupted beforehand with Module 9121.
- The feed-rate override is not offset.
- If the axis was in the "search for reference mark" state before, this state is canceled. The positioning movement always starts from the current counter value.
- If modules 9120, 9227 (Start Auxiliary Axis), 9121 (Stop Auxiliary Axis) and 9122 (Pass over Reference in Auxiliary Axis) are called several times for the same axis during a PLC scan, only the latest command is followed.
- Error codes 3 and 4 not possible

Call:

PS	B/W/D/K	<Axis>
PS	B/W/D/K	<Target position/Increment> in [0.0001 mm], ref system
PS	B/W/D/K	<Feed rate> in [mm/min]
PS	B/W/D/K	<Acceleration> in [mm/s2] 0: Value from MP_maxAcceleration is used
PS	B/W/D/K	<Jerk> in [mm/s3] 0: Value is transferred but not evaluated
PS	B/W/D/K	<Mode> NC and auxiliary axes: Bit 0 = 0: Absolute positioning Bit 0 = 1: Incremental positioning Only NC axes: Bit 1 = 1: Software limit switch active Bit 2 = 1: Rapid traverse override effective Bit 3 = 1: DCM collision monitoring is deactivated
CM	9227	
PL	B/W/D	<Status> 0: Function performed 1: Illegal group number 2: Incorrect parameterization via bit mask 5: Axis is not controlled 6: Axis is already being positioned 20: Module was not called in a spawn or submit job

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Function performed
	1	Error (error code in NN_GenApiModuleErrorCode)
NN_GenApiModule ErrorCode	1	Invalid group number
	2	Invalid value for bit mask
	20	Module was not called in a spawn or submit job

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgAxisComp **active**	 401801

The parameter object CfgAxisComp is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)

Axis errors are compensated for by changing the command variables for the position.

The MANUALplus 620 compensates the following mechanically caused axis errors:

- Backlash
- Linear axis errors
- Nonlinear axis errors (direction-dependent)
  - Screw-pitch error
  - Axis sag
- Thermal expansion
- Sliding friction (for digital axes, compensation is carried out in the speed controller)

You can activate either linear or nonlinear axis-error compensation per axis.

Backlash compensation can be activated in addition to linear axis-error compensation.

As of NCK software version 597110-04, if nonlinear axis-error compensation is active, **MP_backLash** can be used to activate backlash compensation in addition to the compensation-value tables.

You can also add other types of compensation.

▶ In **MP_active,** you switch all compensations (except stiction) on or off.

MP_active

Switch all axis compensations on/off
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: on
Backlash compensation, linear or nonlinear axis-error compensation, reversal-error compensation and thermal compensation are all active
off
Axis compensations are not active
Default: off
Access: LEVEL3
Reaction: RUN

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgAxisComp	
backLash	401802
filterTime	401805
posCtrlRevErr	401806
posCtrlRevErrTime	401807

The parameter object CfgAxisComp is not required for:

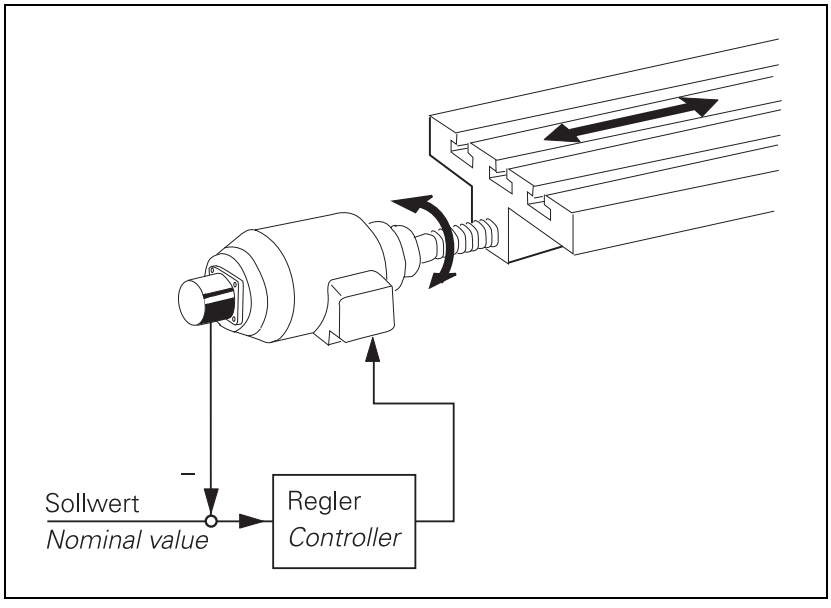
- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)

Cause outside of the control loop

During a reversal in axis direction, there is often a little play between the rotary encoder and table. This play is referred to as backlash.

If the distance is measured using a speed encoder, the backlash compensation compensates the play between the rotary encoder and the table.

Positive backlash: The rotary encoder reading is ahead of the table. The table traverse is too short.
Negative backlash: The rotary encoder reading is behind the table. The table traverse is too long.

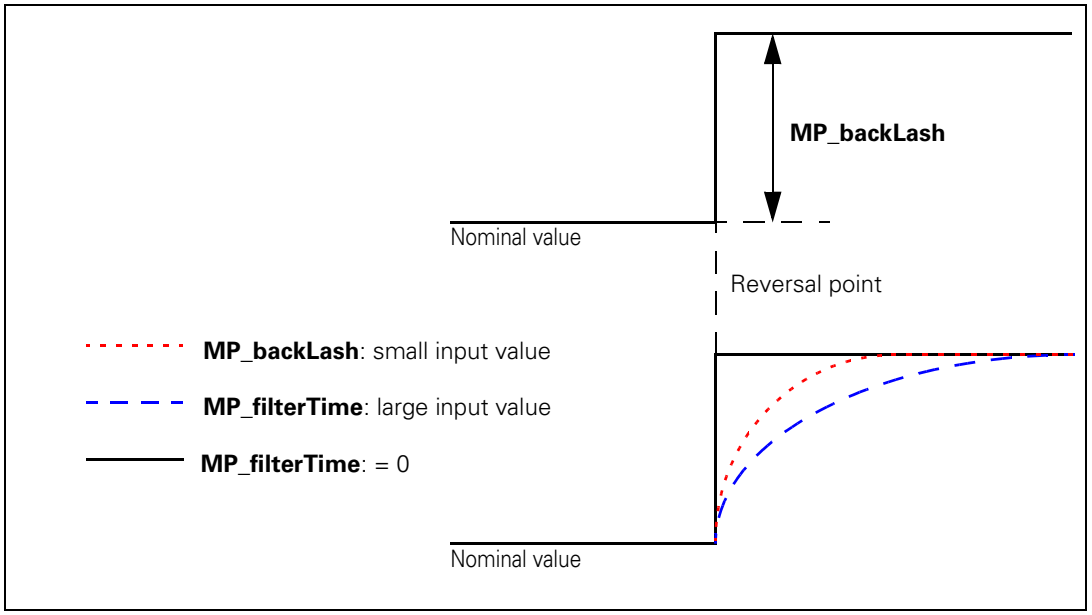


Compensation:

▶ Enter the backlash in **MP_backLash.**

▶ In **MP_filterTime** enter the time in which the distance to be compensated should be traversed.

The value of the backlash is added to the position value at every reversal of direction (even if it results from nonlinear axis-error compensation, for example) and is considered by the position controller. The value of the k_V factor therefore influences the settling time for backlash compensation.



MP_backLash

	Backlash compensation; backlash outside of the control loop Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	–1.000 000 000 to +1 [mm] or [°]
Default:	0
Access:	LEVEL3
Reaction:	RUN

MP_filterTime

	Time constant for backlash compensation Available from NCK software version: 597 110-03.
Format:	Numerical value
Input:	1 to 1000 [ms] 0: Compensation is output as a step
Default:	No value, parameter optional (= 0)
Access:	LEVEL3
Reaction:	RUN

Example:

MP_backLash: 0.03 mm
MP_filterTime: 15 ms

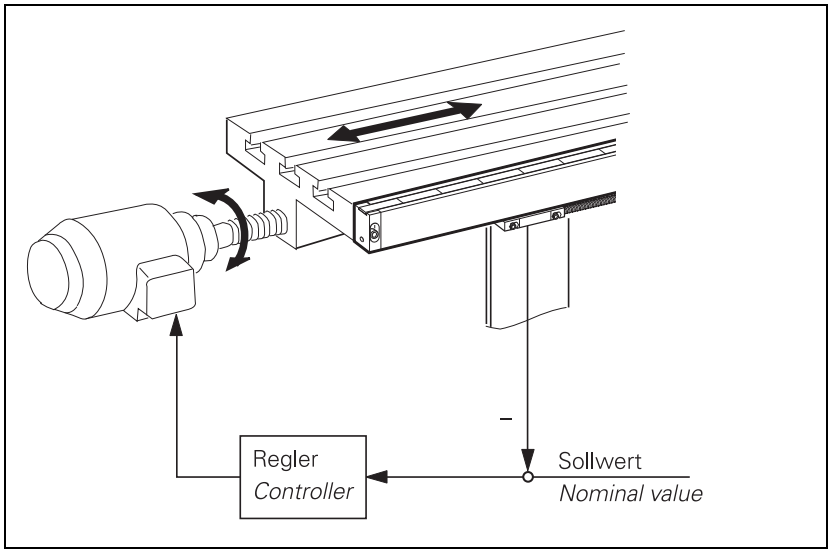
For every change in direction, a nominal speed command signal is output for 15 ms, which corresponds to a feed rate of 120 mm/min:

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

Cause within the control loop

Only possible with digital drive control!

If a position encoder is used for direct distance measurement, the MANUALplus 620 can compensate the play between the motor and the table. At the same time, this compensates the reversal peaks in circular movements.



Compensation

- In **MP_posCtrlRevErr** enter the reversal error in [mm].
- In **MP_posCtrlRevErrTime** enter the time in which the distance to be compensated should be traversed.

MP_posCtrlRevErr

	Backlash compensation (distance)
	Available from NCK software version: 597 110-03.
Format:	Numerical value
Input:	–9.9999 to +9.9999 [mm] or [°]
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RUN

MP_posCtrlRevErrTime

	Backlash compensation (time)
	Available from NCK software version: 597 110-03.
Format:	Numerical value
Input:	0 to 1000 [ms]
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgAxisComp **linearCompValue** **compType**	 401803 401804

The parameter object CfgAxisComp is not required for:

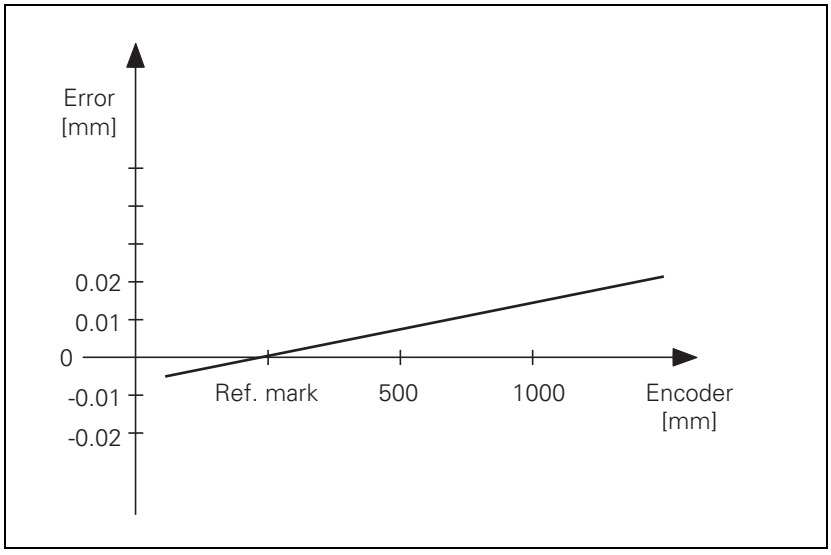
- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)

Note

Linear axis error compensation is not available for rotary axes!

For every linear axis you can compensate a linear axis error.

- Positive linear axis error: The table moves too far.
- Negative linear axis error: The table moves too short a distance.



Compensation:

- In **MP_linearCompValue**, enter the axis error [mm/m].
- With **MP_compType**, activate the linear axis error compensation.

MP_linearCompValue

Linear axis error compensation
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –1.000 000 000 to +1 [mm/m]
Default: 0
Access: LEVEL3
Reaction: RUN

MP_compType

Selection of linear/nonlinear axis error compensation
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: linear
Linear axis error compensation is active
non-linear
Nonlinear axis error compensation is active
Default: linear
Access: LEVEL3
Reaction: RUN

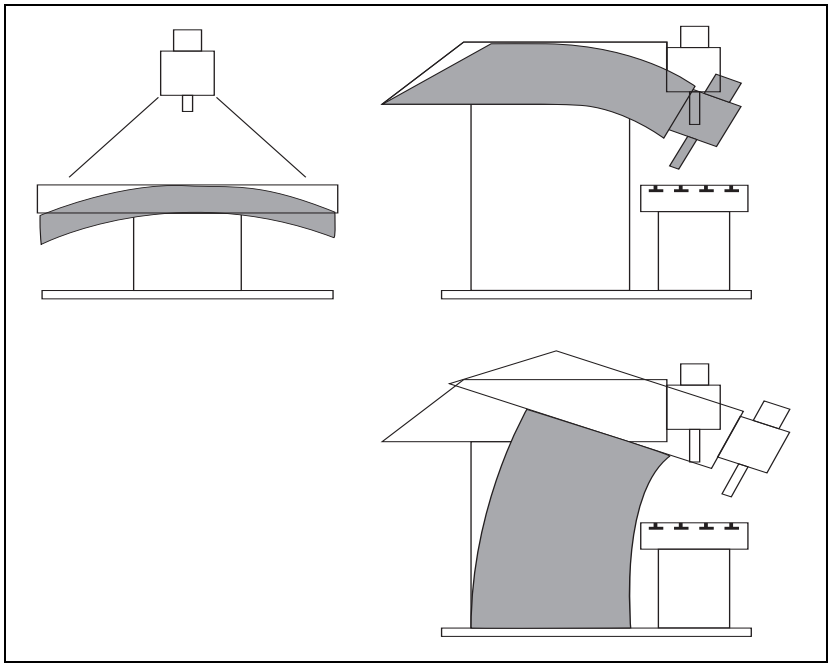
Note

If **nonlinear** axis-error compensation is active (**MP_compType** = non-linear), linear axis-error compensation is not available.

6.9.3 Nonlinear axis error compensation

Depending on the design of the machine, production tolerances, or external factors (e.g. temperature), a nonlinear axis error can occur. Typical errors are screw-pitch errors and axis sag.

These graphics show typical nonlinear axis errors:



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

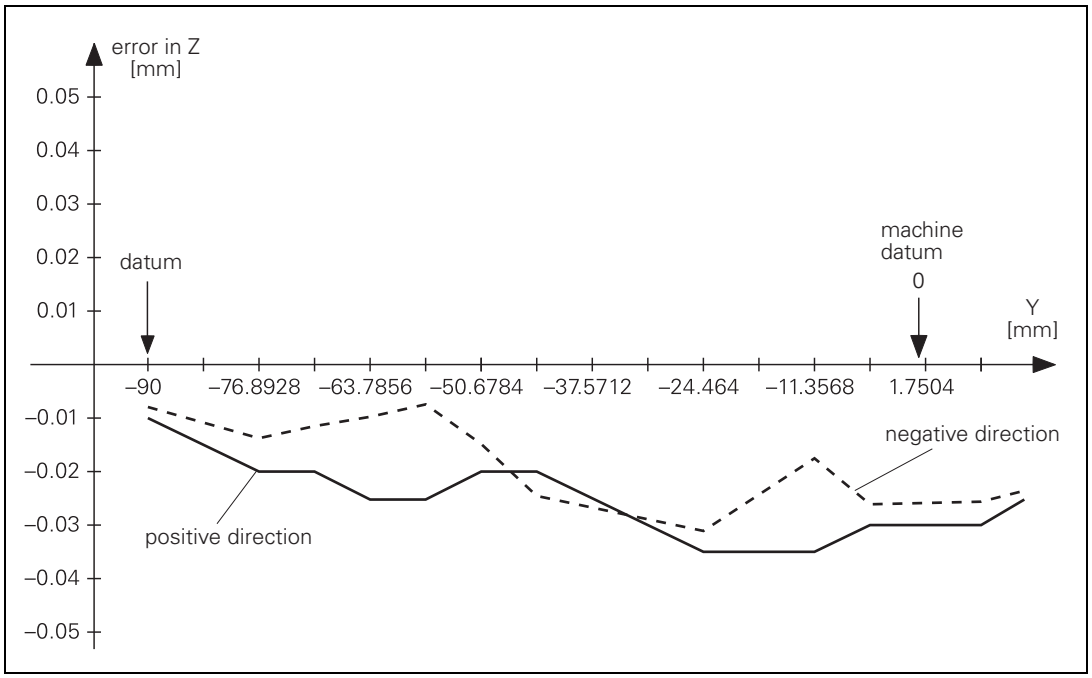
Note

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

Nonlinear axis error compensation is also effective for an open loop. In this case the compensation value is considered when the control loop is closed.

Nonlinear axis error compensation supports one compensation value group each for the positive and negative directions of traverse.

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



Compensation value tables

The compensation values for nonlinear axis error compensation are stored in the following tables:

- The ***.COM** tables contain the compensation values for max. 256 compensation points. A *.COM table is required for each axis and spindle. It consists of the following columns:
  - **AXISPOS:** Compensation points that are assigned compensation values. The compensation points are given with respect to the machine datum.
    Equidistant spacing between the measuring points is **necessary**.
  - **BACKLASH:** Compensation values for screw pitch errors in negative direction of traverse. The BACKLASH column is defined for the axis for which this compensation-value table is created. This way the backlash can be compensated directly via the compensation-value table.
  - **Axis * – axis to which the table refers:** Compensation values for screw pitch errors in positive direction of traverse
  - **Axis * – adjoining axis:** Compensation values for sag errors with respect to the adjoining axis
  - **Spindle:** The compensation values for a spindle are entered in this column

- In the ***.CMA** table, the *.COM tables are assigned to the error-causing axis.
  - **ACTIVE:** The character * activates the compensation value tables.
  - *** axis:** File name of the *.COM file with the compensation values of this axis.
  - **Spindle:** File name of the *.COM file with the compensation values of this spindle.

You will find the path of the *.CMA tables in the parameter object **System/Paths/CfgTablePath/TABCMA** (standard name of the file: config.cma). The *.CMA file contains the file names of the *.COM files. The directory path of the *.COM tables is entered in the parameter object **System/Paths/CfgOemPath/oemTable**.

Entering compensation values

The following information must be entered in the <*.COM> tables:

- In the AXISPOS column, enter the compensation points for the compensation values. The positions are given with respect to the machine datum (MP_refPosition).
- If required, enter the compensation values measured in the negative direction of traverse in the BACKLASH column.
- Enter the compensation values to which the compensation points belong in the column of the axis for which a dependency relationship exists. The name of the column is the name from MP_CfgAxes/axisList (see Table Format).

Example: The following dependencies apply to the Y axis and Z axis:

- Ball screw pitch error in Z and Y: Z = F(Z) and Y = F(Y)
- Axis sag in Z depending on Y
- Range of traverse: Z axis = 800 mm, Y axis = 500 mm
- Start point for compensation values: Z = –200 mm, Y = –90 mm
- Desired spacing of compensation points: 5 mm

Number of compensation points:

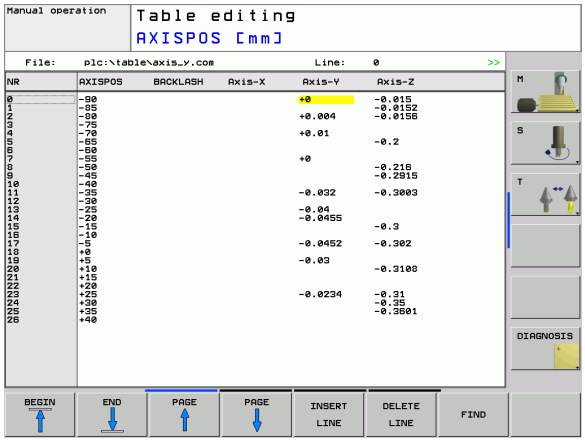
$$\frac{500\ \text{mm}}{5\ \text{mm}} = 100 \text{ compesation points in Y axis}$$

$$\frac{800\ \text{mm}}{5\ \text{mm}} = 160 \text{ compensation points in Z axis}$$

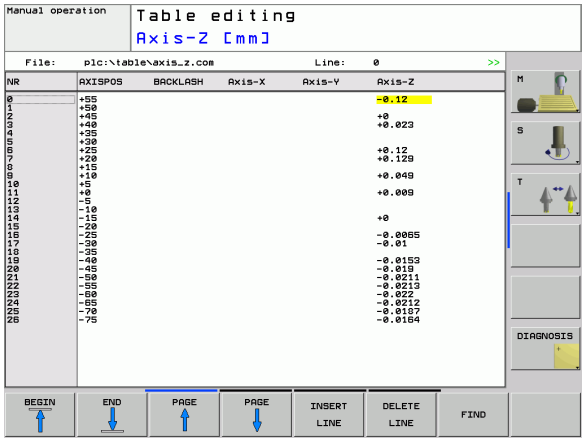
How to access the tables:

- Switch to the **Organization** operating mode
- Press the soft key.
- Enter the code number 95148.
- Press the **PGMMGT** soft key (drive PLC: becomes visible).
- In the PLC:/table drive, open the tables *.COM and *.CMA.

Entries:



Y axis:
Screw-pitch error in Y axis, sag error in Z axis



Z axis:
Screw-pitch error in Z axis

Assigning the compensation value tables to the axes

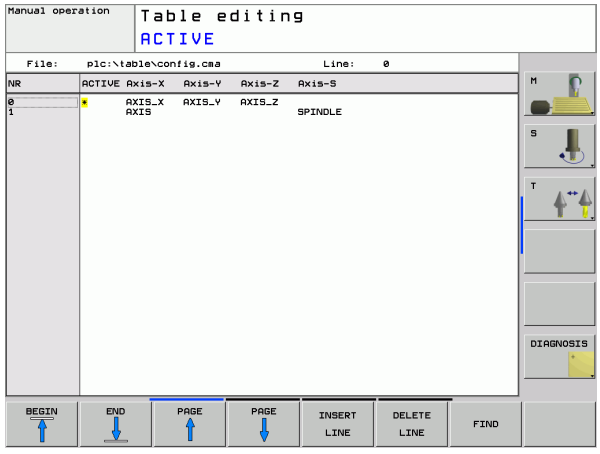
General relationship for *.CMA tables:
[Axis in column from *.com] = F(Axis in column from *.cma, in which *.com is entered)

Enter the compensation-value tables in a table of the type <*.CMA> (standard name config.cma).
(for table formatting, see the chapter Tables):

- Enter a column for each axis to be compensated. The column names must match the axis keys from MP_CfgAxes/axisList.
- Enter the names of the compensation-value tables (*.com) line-by-line in the appropriate axis columns.
  You can assign more than one compensation value table to each axis, however only one table can be active.
- Activate the compensations with an * in the column ACTIVE, which can be entered via the table editor or via the PLC (SQL server).
  All compensations in this line become active.

Example:
Z axis = F(Y axis); axis sag compensation
Y axis = F(Y axis); nonlinear compensation
The first line is active.

Entries:



CMA table

Activate error compensation

Three requirements must be fulfilled for activating nonlinear axis error compensation:

- Activate the general compensation procedures with **MP_active = ON.**
- Activate the axis-specific nonlinear axis error compensation with **MP_compType=non-linear** (See "Linear axis error compensation" on page 473).
- In the config.cma file, activate a line with an ***** in the ACTIVE column or with Module 9095. The active line can be interrogated using Module 9035.

Note

Compensation is not available for axis and spindle positioning by PLC.

Module 9095 Activate axis-error compensation

Module 9095 activates a line in the selected file (*.CMA) and assigns the arguments for the compensation value tables (*.COM). Multiple measurement series (e.g. x=f(), y=f()..) can be stored in the compensation value tables. After the module has been executed, the argument is assigned. In this way the screw pitch error x=f(x) and axis sag x=f(y) can be compensated simultaneously, for example.

Constraints:

- The transferred line remains selected as the active line even after a control reset.
- Once the NC program has started, the module operates only during the output of an M/G/S/T/T2/Q strobe.
- The nominal axis values may change slightly when the compensation value table is switched over.

Call:

PS	B/W/D/K	<Active line>
CM	9095	
PL	B/W/D	<Error code>

0: Compensation was selected
1: Line was not found in the *.CMA table
2: Compensation value table (*.COM) is missing
3: Compensation value table > 256 entries
4: Maximum total number of compensation points exceeded
5: Too many compensation value tables (>10)
6: *.CMA file does not exist
7: Call was not from a submit job
8: Call during running program without strobe
10: *.CMA file is protected

Module 9035 Read NC status information
Module 9035 reads status information. A function number specifying the desired status information is transferred.

Function number 19: Display active line in the *.CMA file

Note

The line number is displayed even if the active line does not contain any *.COM file.

Call:

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

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	No error
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	The transferred input parameter does not identify any status information available with the software version being used.
	20	Information which is only available in a submit/ spawn job was requested in the cyclic PLC program.

Special case: Rotary axis

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

Settings in the configuration editor	MP number
System PLC CfgPlcPeriphery **tempCompensation**	103405
Axes ParameterSets [Key name of the parameter set] CfgAxisComp **active**	401801

The parameter object CfgAxisComp is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)

To compensate thermal expansion, exact measurements of machine thermal behavior as a function of temperature (e.g., the center of axis expansion, the amount of the expansion) are necessary. Since the thermal expansion of the axes is largely proportional to the temperature, you can determine the amount of expansion by multiplying the temperature value by a certain factor.

The temperature values measured by the Pt100 thermistors are transferred using Module 9003. Module 9231 activates the compensation for thermal expansion according to the lag tracking method.

Compensation:

- Activate the general compensation procedures with **MP_active=ON** (See "MP_active" on page 468).
- Transfer the distance to be compensated to Module 9231. At the same time, "lag tracking" becomes active. This means that the actual position is offset by a certain value per PLC cycle until the complete value is compensated.
- Define the amount of compensation per PLC cycle for lagged-tracking axis error compensation in MP_tempCompensation.

MP_tempCompensation

	Compensation of thermal expansion Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.0000 to 359999.6400 [mm/min] 0 = Compensation not active
Default:	0
Access:	LEVEL2
Reaction:	NOTHING

Module 9231 Compensation of thermal expansion
Thermal expansion is compensated by Module 9231. The axis number and the compensation value are transferred.

The module activates lag tracking. This means that the actual position is offset by a certain value per PLC cycle until the complete value is compensated. The increment of change per PLC cycle must be defined in MP_tempCompensation.

This does not change the value in the actual position display.

The module functions only in the cyclic PLC program.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
PS	B/W/D/K	<Compensation value> Range: –30000 to +30000 [1/10 µm]
CM	9231	

Error recognition:

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

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgSpeedControl **vCtrlIntTime**	 400903

The parameter object CfgSpeedControl is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)
- Analog axes (MP_axisHw=Analog)

In machines with a great deal of stiction, a high integral-action component can accumulate over time if there is a position error at standstill. This can result in a jump in position when the axis begins moving. In such cases you can limit the integral-action component of the speed controller in **MP_vCtrlIntTime:**

▶ Enter a limit in **MP_vCtrlIntTime.** Realistic input values: 0.1 to 2.0

MP_vCtrlIntTime

	Limit of integral factor of the speed controller Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 30 [s]
Default:	0 [s]
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgControllerComp	
compFriction0	401401
compFrictionT1	401402
compFrictionT2	401403
compFrictionNS	401404

The parameter object CfgControllerComp is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)
- Analog axes (MP_axisHw=Analog)

CC 61xx, CC 424

Note

Machine parameters **MP_compFrictionT1** and **MP_compFrictionT2** now function with respect to distance rather than time (unit: [mm] or [°]).

With the CC 61xx and CC 424, this makes it possible to compensate quadrant transitions independently of velocity, acceleration, and diameter.

The CC calculates the distance from the zero crossover of velocity. The compensation current is reduced starting from the distance **before** the zero crossover defined in **MP_compFrictionT1.** After the zero crossover, the compensation current is increased again. **MP_compFrictionT2** defines the point **after** the zero crossover at which 63% of the compensation current is reached.

- In **MP_compFrictionT1,** define the distance before the reversal point from which a reduction of the current from **MP_compFriction0** is to go into effect.
- In **MP_compFrictionT2,** define the distance after the reversal point from which the current from MP_compFriction0 is to go into effect again.

MP_compFriction0

Friction compensation at low speed
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 30 [A]
Default: 0 [A]
Access: LEVEL3
Reaction: RUN

MP_compFrictionT1

Distance before the reversal point for current from MP_compFriction0
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 1 [mm] or [°]
The following units of measure apply:
MP_compFrictionT1 = [s] if **MP_compFrictionT2** = 0
(same behavior as CC 422)
MP_compFrictionT1 = [mm] if **MP_compFrictionT2** > 0
0: No friction compensation
Default: 0
Access: LEVEL3
Reaction: RUN

MP_compFrictionT2

Distance after the reversal point for current from MP_compFriction0
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 1 [mm] or [°]
0: Friction compensation same as with CC 422
Default: 0 [mm]
Access: LEVEL3
Reaction: RUN

6.10 Machine kinematics on lathes (as of NC software 548328-03)

As of software version 548328-03, a new kinematics model is available for describing the machine kinematics of the MANUALplus 620. The MANUALplus 620 can be operated as selected with the old or the new kinematic model. The use of KinematicsDesign is possible only with the new kinematic model, which is also activated in the factory default condition.

Machine parameters in the MANUALplus 620 describe the machine kinematics. It is a precondition that the kinematics models consist of translation axes and rotation axes which are linked to each other. This structure can also be used for configuring axes that are not perpendicular with respect to each other.

Multiple sets of kinematics can be configured for one machining channel. Multiple sets of kinematics are needed, for example, if the spindle and C axis on a lathe are driven by the same motor. The kinematics description consists of a transformation chain, starting from the tool reference point to the the linear axes, all the way to the last axis or transformation.

The transformation chain consists of

- fixed lengths (machine dimensions)
- variable lengths (linear axes)
- fixed rotations (machine conditions)
- variable rotations (rotary axes)

starting from the tool reference point (e.g. spindle housing, machine base, linear axes, machine envelope). Each translation is described.

All machine axes in the kinematics model are described in the position REF=0 (machine center).

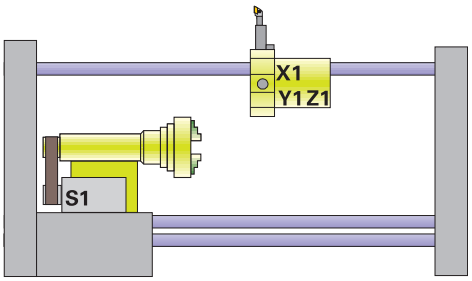
Note

The KinematicsDesign software for PCs can only be used in conjunction with the new kinematics model described below.

Example 1:

Preconfigured kinematics in the factory default setting of the NC software

In the factory default setting of the NC software, the kinematics properties of a lathe with spindle, X axis, Y axis and Z axis is already preconfigured in the kinematics model K4_CH1_S1.



You must enter the following settings in the configuration editor in order to configure this three-axis machine. These settings were already performed for the factory default setting of the NC software.

Settings in the configuration editor		MP number
Channels		
Kinematics		
CfgKinComposModel		
K4_CH1_S1 (Key name of kinematics model)		
subKinList		
[0]:	TOOL_TH1	K4_CH1_S1.202901.000
[1]:	K_XYZ_CH1	K4_CH1_S1.202901.001
[2]:	K_WP_S1	K4_CH1_S1.202901.002
activeSpindle:	S1	K4_CH1_S1.202902
CfgKinSimpleModel		
K_XYZ_CH1 (Key name of the subkinematics)		K_ZYX.202800
kinObjects		
[0]:	TransMaAxDummy2	K_XYZ_CH1.202801.000
[1]:	MachAxisY1	K_XYZ_CH1.202801.001
[2]:	TransMaAxDummy1	K_XYZ_CH1.202801.002
[3]:	MachAxisX1	K_XYZ_CH1.202801.003
[4]:	MachAxisZ1	K_XYZ_CH1.202801.004
[5]:	MACH_BASE	K_XYZ_CH1.202801.005
CfgKinSimpleAxis		
MachAxisX1 (Key name of the machine axis)		
dir:	X	MachAxisX1.202701
axisRef:	X1	MachAxisX1.202702
MachAxisY (key name of the machine axis)		
dir:	Y	MachAxisY1.202701
axisRef:	Y1	MachAxisY1.202702
MachAxisZ (key name of the machine axis)		
dir:	Z	MachAxisZ1.202701
axisRef:	Z1	MachAxisZ1.202702
CfgKinAnchor		
MACH_BASE (Key name of the machine base)		
kindOfAnchor:	MachBase	MACH_BASE.203701
Channels		
ChannelSettings		
CH_NC1 (Key name of the machining channel)		
CfgActivateKinem		
kinemToActivate:	K4_CH1_S1	CH_NC1.204001

6.10.1 Configuring the machine kinematics

Specify the machine kinematics as well as the resulting transformation model, starting from the machine reference point (REF 0).

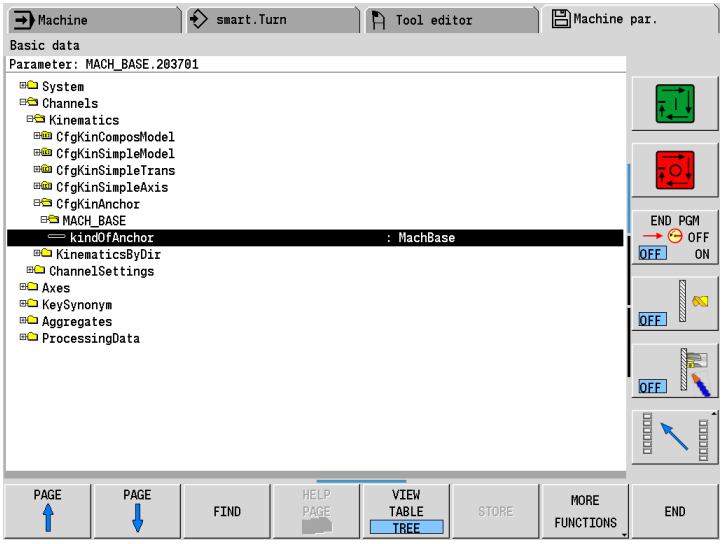
▶ Set the axes to the machine datum

Basic procedure: work your way "from the bottom up" through the kinematics configuration folders (System/Channels/Kinematics).

▶ Configure a machine base

Define the fixed points in the kinematics chain with the **CfgKinAnchor** configuration object. A machine base is such a fixed point.

▶ Create a key name (=folder) for the machine base under **CfgKinAnchor,** e.g. **Base**. Select the value **MachBase** for the parameter **MP_kindOfAnchor**. You will need the machine base later for the configuration of subkinematics.



MP_kindOfAnchor

Fixed point in the kinematics chain
Available from NCK software version: 597 110-03.

Format: Selection menu

Selection: **MachBase**
Specifies a fixed point in the kinematics chain.
In the desired subkinematics (CfgKinSimpleModel), enter the machine base (e.g. with the key name "Base") at the appropriate position in the list, i.e.:
MachAxisY
MachAxisZ
MachAxisX
MACH_BASE

Default: –

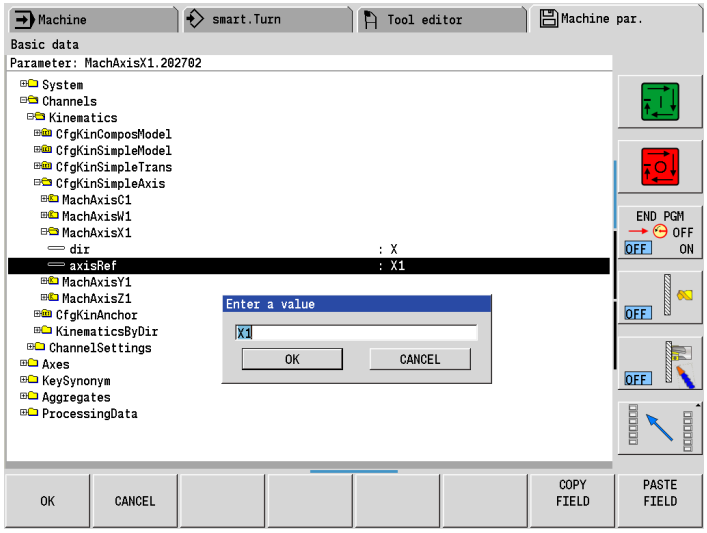
Access: LEVEL3

Reaction: RUN

▶ Now define the machine axes in the kinematics chain

Define via **CfgKinSimpleAxis** the point at which a machine axis becomes effective in the kinematics sequence. Movements by the indicated axis **(MP_axisRef)** change via the entry **MP_dir** the positions of previously defined axes or objects relative to subsequent axes or objects. The start of the sequence is always the tool reference point.

▶ Define under **CfgKinSimpleAxis** a key name (= folder) for each machine axis, e.g. **`MachAxisX.`** Enter via **MP_dir** the direction and via **MP_axisRef** the axis from System/CfgAxes/axisList to which the machine axis belongs.



MP_dir

Direction of the machine axis
Available from NCK software version: 597 110-03.

Format: Selection menu

Selection: **X, Y, Z, A, B, C**
Specifies the direction of the machine axis in the local coordinate system, which is given by the previous coordinate transformations.
An X or A entry means that the machine axis moves or rotates in the X direction of the local coordinate system. Y and B stand for the Y direction, Z and C for the Z direction.

Default: –

Access: LEVEL3

Reaction: RUN

MP_axisRef

Reference to the associated machine axis
Available from NCK software version: 597 110-03.

Format: String

Input: Max. 18 characters
Enter here the key name of the associated axis from System/CfgAxes/axisList.

Default: –

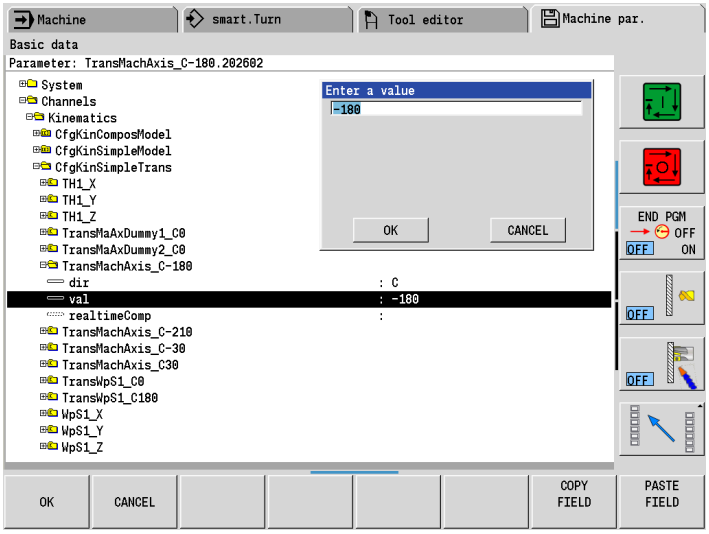
Access: LEVEL3

Reaction: RUN

▶ Now define the transformations in three dimensions

Starting from the tool reference point, define in sequence the shifts or rotations via the linear, rotary and tilting axes to a center point of a (rotary) table.

▶ Enter the shifts and rotations under **CfgKinSimpleTrans.** Define key names for the transformations. Create a folder for each transformation. A transformation is performed in the axis direction entered under **MP_dir** in the current kinematics sequence by the value entered in **MP_val.** This can be a linear translation as well as a rotation about an axis.



Example: Rotation of the coordinate system around the C axis by –180°:
MP_dir = C
MP_val = -180
Enter in **MP_val** the value for the transformation. For linear axes (X, Y, Z, ...) the value is entered in [mm] or for rotary axes (A, B, C) in [°]. Example:
MP_val = 47.092 [mm] or
MP_val = 45.05 [°]
The units are not entered. The MANUALplus 620 interprets the units of measurement from the axis designation indicated in **MP_dir**.

As a rule, the MANUALplus 620 takes changes in the mechanical offset into account, meaning that these changes do not have to be compensated with a PLC datum shift.

MP_dir

Direction of the transformation
Available from NCK software version: 597 110-03.
Format: Selection menu
Selection: **X, Y, Z, A, B, C**
The selection of X, Y and Z indicates that it is a shift of the coordinate system in X, Y or Z direction. If A, B, or C is indicated, then it is a rotation of the coordinate system about the A, B or C axis.
Default: –
Access: LEVEL3
Reaction: RUN

MP_val

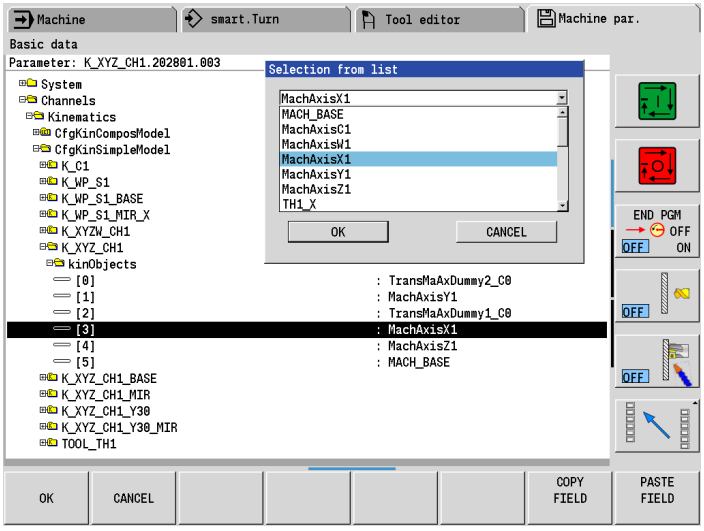
Value of the transformation
Available from NCK software version: 597 110-03.
Format: Numerical value
Input: –999 999 999.999 999 999 to +999 999 999.999 999 999 [mm] or [°]
Default: –
Access: LEVEL3
Reaction: RUN

▶ Now define the subkinematics

Subkinematics can be included separately, and so increase the flexibility of the machine kinematics. On machines with exchangeable components you can use the subkinematics in multiple kinematics models.

Subkinematics are defined under **CfgKinSimpleModel:**

▶ Create a folder (= key name of the subkinematics) for each subkinematics under **CfgKinSimpleModel.** Under **MP_kinObjects** you describe the structure of the subkinematics. Then, in sequence from the tool to the machine base you enter the names of the kinematics objects comprising the subkinematics. Kinematics objects can be machine axes (key names from CfgKinSimpleAxis), transformations (key names from CfgKinSimpleTrans) and fixed points (e.g. the machine base key name from CfgKinAnchor).



MP_kinObjects

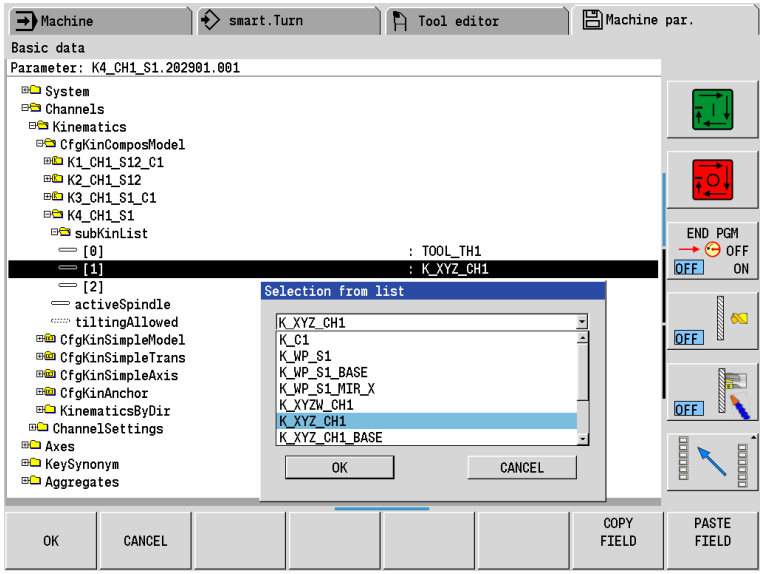
List of key names of objects in the kinematics chain
Available from NCK software version: 597 110-03.

Format: Array [0...49]
Default: –
Access: LEVEL3
Reaction: RUN

▶ Now collect the subkinematics to form a kinematics model

Define kinematics models under **CfgKinComposModel.**

▶ Create a folder (= key name) for the kinematics model under **CfgKinComposModel.** Via **MP_subKinList** you enter the subkinematics that comprise the kinematics model. The list must be in the sequence from the tool [0] to the machine table. It is also possible to enter just one subkinematics. Enter under **MP_activeSpindle** the name of the spindle (from System/CfgAxes/axisList) that belongs to the kinematics model. If tilting of the working plane in allowed for the kinematic model, enter the value TRUE in the parameter **MP_tiltingAllowed**.



MP_subKinList

List of key names of the subkinematics
Available from NCK software version: 597 110-03.

Format: Array [0...5]
Input: Enter the subkinematics comprising the machine kinematics, going from the tool to the workpiece.
Default: –
Access: LEVEL3
Reaction: RUN

MP_activeSpindle

Key name of the active spindle of this kinematics model
Available from NCK software version: 597 110-03.

Format: String
Input: The name of the spindle can be taken from System/CfgAxes/AxisList, e.g. S, Spindle1, etc.
Default: –
Access: LEVEL3
Reaction: RUN

MP_tiltingAllowed

Tilting the working plane is allowed
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: TRUE
Tilting the working plane is allowed with this kinematic model.
FALSE
Tilting the working plane is not allowed with this kinematic configuration. The MANUALplus 620 displays an error message.

Default: No value, parameter optional (= TRUE)

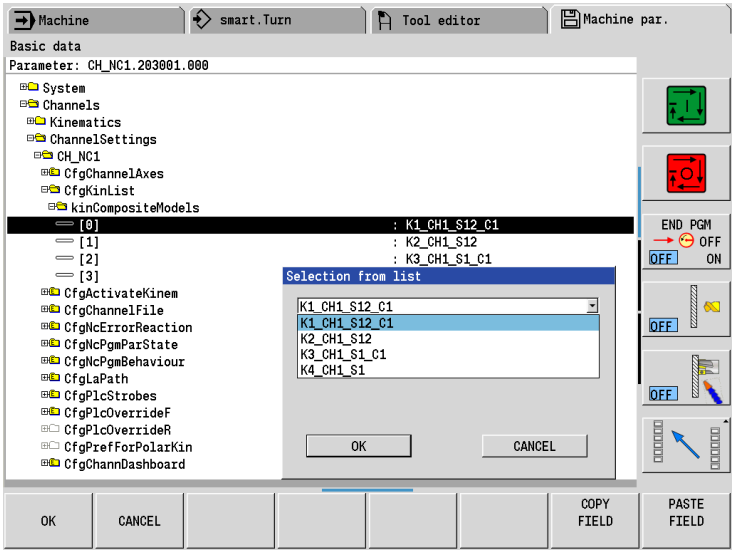
Access: LEVEL3

Reaction: RUN

- Last step: Assign the configured kinematics models to the machining channel.

The settings for the machining channel (ChannelSettings/CH_NC1) must include the kinematics models valid for this channel:

- Open the **CfgKinList** configuration object. Enter in the **kinCompositeModels** list parameter the key names of the kinematic models for the machining channel.
- Open the **CfgAcitvateKinem** configuration object. Choose from the selection menu of the **MP_kinemToActivate** parameter the key name of the kinematics model to be activated.



MP_kinCompositeModels

List of key names of kinematics models for this machining channel
Available from NCK software version: 597 110-03.

Format: Selection menu

Selection: Key names from Channels/Kinematics/CfgKinComposModel

Default: –

Access: LEVEL3

Reaction: RUN

MP_kinemToActivate

Kinematics to be activated / Active kinematics
Available from NCK software version: 597 110-03.

Format: Selection menu
Selection: Key names from Channels/Kinematics/CfgKinComposModel
Select the key name of the kinematics configuration to be activated. You can also see from this parameter which kinematics configuration is currently active.
Default: –
Access: LEVEL3
Reaction: RUN

Note

As an alternative you can configure the kinematics with the PC software KinematicsDesign. Registered customers can download the software from the HEIDENHAIN FileBase on the Internet. The User's Manual is included with the software as online help.

6.10.2 Preconfigured subkinematics

The following table lists all preconfigured subkinematics that are included in the control when it is shipped.

Subkinematics	Meaning	Transformation	Meaning
TOOL_TH1	Tool carrier	TH1_Y TH1_X TH1_Z	Direction of the tool length in Y, X, Z
K_XYZ_CH1	Linear axes	TransMaAxDummy2_C0	Placeholder for automatic mirroring of the X axis[1]
		MachAxisY1	Position of Y axis
		TransMaAxDummy1_C0	Placeholder for automatic mirroring of the X axis[1]
		MachAxisX1	Position of X axis
		MachAxisZ1	Position of Z axis
		Mach_BASE	Fixed position of the kinematics chain
K_C1	C axis	K_C1	Position of C axis
K_WP_S1	Workpiece spindle	WpS1_Y	Position of workpiece spindle Y
		WpS1_X	Position of workpiece spindle X
		TransWpS1_C0	Placeholder for automatic mirroring of the X axis[1]
		WpS1_Z	Position of workpiece spindle Z

Subkinematics	Meaning	Transformation	Meaning
K_XYZ_CH1_MIR		TransMachAxis_C-180	Reverse rotation of X axis
		MachAxisY1	Position of Y axis
		TransMaAxDummy1_C0	Placeholder for mirroring of the X axis[1]
		MachAxisX1	Position of X axis
		MachAxisZ1	Position of Z axis
		Mach_BASE	Fixed position of the kinematics chain
K_WP_S1_MIR_X		WpS1_Y	Position of workpiece spindle Y
		WpS1_X	Position of workpiece spindle X
		TransWpS1_C180	Mirroring of X axis
		WpS1_Z	Position of workpiece spindle Z
K_XYZ_CH1_Y30		TransMachAxis_C-30	Reverse rotation of Y axis
		MachAxisY1	Position of Y axis
		TransMachAxis_C30	Oblique-axis position (Y=30°)
		MachAxisX1	Position of X axis
		MachAxisZ1	Position of Z axis
		Mach_BASE	Fixed position of the kinematics chain
K_XYZ_CH1_Y30_MIR		TransMachAxis_C-210	Reverse rotation of: – Mirroring of X axis – Oblique-axis position (Y30)
		MachAxisY1	Position of Y axis
		TransMachAxis_C30	Oblique-axis position (Y=30°)
		MachAxisX1	Position of X axis
		MachAxisZ1	Position of Z axis
		Mach_BASE	Fixed position of the kinematics chain
K_WP_S1_MIR_X		WpS1_Y	Position of workpiece spindle Y
		WpS1_X	Position of workpiece spindle X
		TransWpS1_C180	Mirroring of X axis
		WpS1_Z	Position of workpiece spindle Z

[1] TransMaAxDummy1_C0, TransMaAxDummy2_C0 and TransWpS1_C0 are required for automatic activation of kinematics "in front of the workpiece" for configurations with tool turrets and MultiFix.

6.10.3 Standard kinematics models

In the factory default setting, the following subkinematics are assigned to the existing standard kinematics models:

Kinematics models	Subkinematics
K1_CH1_S12_C1 – Driven tool – With C axis	CfgKinComposModel -> subKinList [0] = TOOL_TH1 [1] = K_XYZ_CH1 [2] = K_C1 [3] = K_WP_S1 -> activeSpindle = S2
K1_CH1_S12 – Driven tool – No C axis	CfgKinComposModel -> subKinList [0] = TOOL_TH1 [1] = K_XYZ_CH1 [2] = K_WP_S1 -> activeSpindle = S2
K1_CH1_S1_C1 – Spindle – With C axis	CfgKinComposModel -> subKinList [0] = TOOL_TH1 [1] = K_XYZ_CH1 [2] = K_C1 [3] = K_WP_S1 -> activeSpindle = S1
K1_CH1_S1 [Standard kinematics model] – Spindle – No C axis	CfgKinComposModel -> subKinList [0] = TOOL_TH1 [1] = K_XYZ_CH1 [2] = K_WP_S1 -> activeSpindle = S1

6.10.4 Find/activate kinematics through the PLC

Module 9098 Find active kinematics configuration

The PLC uses Module 9098 to find the active kinematics configuration. Both the index from CfgKinList as well as the key name of the kinematic model can be interrogated.

Constraints:

- The module is not executable in the cyclic PLC program.

Call:

PS	B/W/D/K	<String number for key name of the kinematic model> 0...15: String number for key name (the index from CfgKinList is also acquired) –1: Only find index from CfgKinList; do not acquire the name
CM	9098	
PL	B/W/D	<Index from CfgKinList> –1: Index could not be acquired

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Kinematic configuration was selected
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid parameter programmed for string number
	20	Module was not called in a spawn or submit job
	30	Configuration datum CfgKinList or CfgActivateKinem is not defined, or the entry from CfgActivateKinem was not found in CfgKinList
	58	No NC channel configured, or more than one configured

Settings in the configuration editor	MP number
Aggregates	
ToolMount	
[Key name of the tool holder]	
CfgToolMount	
kinModelToModify	600418
kinModel	600419

Axis mirroring on lathes is required, for example, when carrying out machining operations above or below the turning center or on the rear side. Such axis mirroring is done within a kinematics group; there is no kinematics change.

Axis mirroring is activated/deactivated either via NC commands or by assigning the activation/deactivation to a tool holder.

With the tool holder (parameter object **CfgToolMount**), you can control axis mirroring with the following parameters:

- **MP_kinModelToModify**: Subkinematics to be replaced of the axis or axes to be mirrored
- **MP_kinModel**: Subkinematics of the mirrored axis/axes, which is to be activated

Axis mirroring that is configured with the tool holder will be activated when the tool holder is inserted. Axis mirroring will remain active until the next tool change.

MP_kinModelToModify

Keys of the subkinematics to be replaced
Available from NCK software version: 597 110-04.

Format: Array [0...99]
Default: -
Access: LEVEL3
Reaction: RESET

MP_kinModel

Keys of the subkinematics to be activated
Available from NCK software version: 597 110-04.

Format: Array [0...99]
Default: -
Access: LEVEL3
Reaction: RESET

6.11 Machine kinematics for lathes (up to NC software 548 328-02)

Machine parameters in the control describe the machine kinematics. It is a precondition that the kinematics models consist of translation axes and rotation axes which are linked to each other. This structure can also be used for configuring axes that are not perpendicular with respect to each other.

Multiple sets of kinematics can be configured for one machining channel. Multiple sets of kinematics are needed, for example, if the spindle and C axis on a lathe are driven by the same motor.

After control startup, the last kinematics model entered in **CfgKinModel** will be active. If required, activate another kinematics model.

Note

The prepared machine kinematics of the MANUALplus 620 include all common lathe applications. If adjustments of the machine kinematics are nevertheless required, you will find all necessary information in the following sections.
The KinematicsDesign software for PCs **cannot** be used in conjunction with the old kinematics model described below.

6.11.1 Configuration of the machine kinematics

Overview of machine parameters for the kinematics configuration

Settings in the configuration editor	MP number
Channels	
Kinematics	
CfgKinModel	
[Key name of the kinematics model]	
axesToolSide	200001
trafoToolSide	200002
trafoDirToolSide	200003
trafoAngelToolSide	200004
toolCoordSys	200005
axesWpSide	200006
trafoWpSide	200007
trafoDirWpSide	200008
trafoAngleWpSide	200009
machineTableSys	200010

Describe the kinematics models in the parameter object **CfgKinModel/Key name of kinematics model.**

Two transformation sequences are defined based on a **machine base system** $\mathbf{C}_{mb}$:

- Transformation sequence on the **tool side**
- Transformation sequence on the **workpiece side**

Each axis on the machine is represented by a coordinate system in one of the two sequences.

The **Z axis** for these coordinate systems is always defined as the **direction of movement** (for translation axes) or the **rotary axis** (for rotation axes) (convention for internal kinematics model).

The Z axis of **translation axes** always indicates the positive direction of movement for the tool, and for the workpiece the negative direction of motion of the assigned physical axis. Positive direction of motion means that the REF display increases when the axis moves in this direction. On the other hand, a negative direction of motion means that the REF display decreases when the axis moves in this direction.

This means that, for **rotational axes**, the Z axis of a coordinate system indicates the positive direction of rotation for the tool, and for the workpiece the negative direction of rotation of the assigned physical axis. Positive direction of rotation for rotational axes means that the REF display increases when the axis rotates in this direction. On the other hand, a negative direction of rotation means that the REF display decreases when the axis rotates in this direction.

For machines with mutually perpendicular axes, the following results from this convention:
The machine base coordinate system C_{mb} must be selected so that its axes are parallel to the physical axes of the machine.

If the algebraic sign of an axis is defined oppositely on the machine, then the coordinate system of the affected axis must be rotated in the transformation so that its Z axis points in the opposite direction.

Important rule of thumb:

The position (location and orientation) of a coordinate system is always expressed in the coordinates of the **previous** coordinate system.

Example:

Position of C_Y in coordinates of C_{mb}

Position of C_x in coordinates of C_y

Position of C_{mt} (machine table) in coordinates of C_X

etc.

The following are defined as well:

- **Tool system** (C_{tool}) in **MP_toolCoordSystem** – in addition to the transformation sequence on the tool side
- **Machine table system** (C_{mt}) in **MP_machineTableSys** – in addition to the transformation sequence on the workpiece side

The transformation sequence can also contain other systems, such as the coordinate system of a 45× rotary axis (for horizontal/vertical spindles).

This system also represents an axis that can have the axis values 0× (vertical position) and 180× (horizontal position). Such an axis is moved to the appropriate position manually, via the PLC or an NC linear block.

Other coordinate systems can be auxiliary systems, which do not represent axes, but are only used to enter the values of the relevant factors in the kinematics chain. These systems are described as DefPoint systems. No axis values can be assigned to these DefPoint systems (as a default, the axis values are always null).

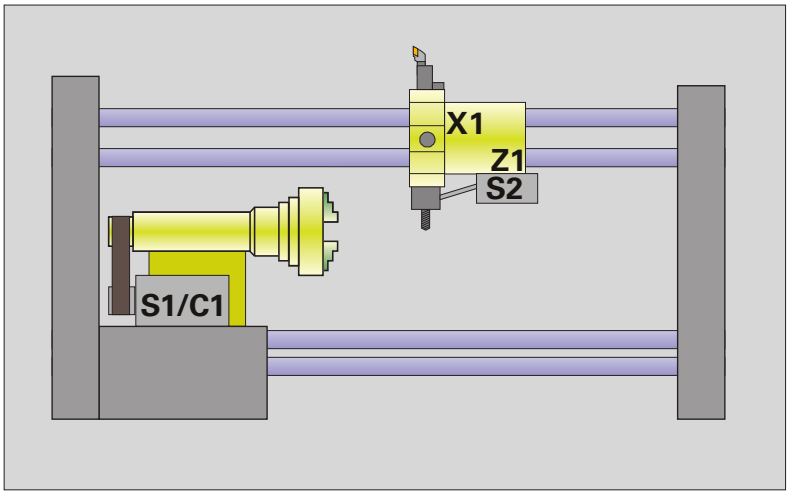
All axes defined in the kinematics chain must be entered in the parameter object **CfgAxes**. The axes with axis values are entered in **MP_CfgAxes/ axisList,** and those without axis values (DefPoint systems) are entered in **MP_specCoordSysList**.

Example:

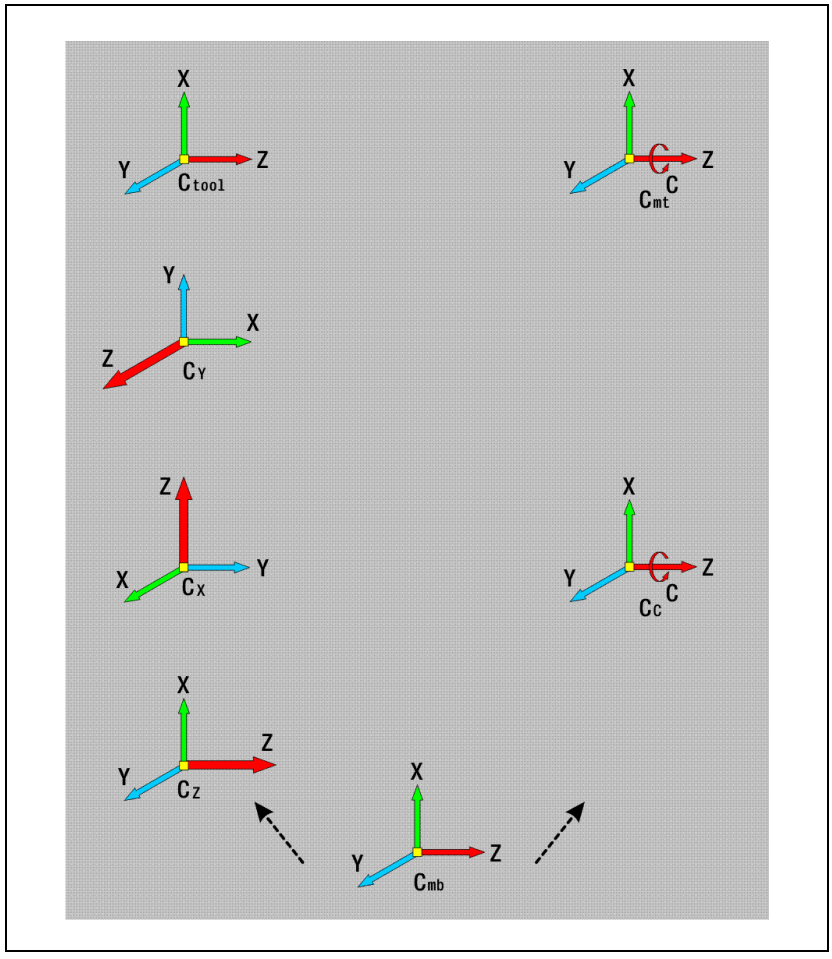
Below is a sketch of a simple lathe with the linear axes X and Z and a C axis. Starting from the machine bed, the X axis is on the Z axis on the tool side.

The coordinate system of the machine bed is determined by the position of the main spindle and the arrangement of the tool carrier (here: "behind workpiece".) As the kinematics chains is evaluated three-dimensionally, an (imaginary) Y axis on the X axis is taken into account.

For the transformation sequence on the workpiece side, only the "machine table" must be taken into account. For kinematics with a C axis, the "machine table" and the C axis must be taken into account.



The kinematics chain for the example machine illustrated is as follows:



Note

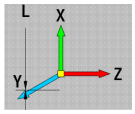
For purposes of clarity, the origins of the coordinate systems are drawn distinct from each other even when they are at the same location.

As the origin of both kinematics chains is the machine base, it is not necessary to describe the coordinate system of the machine base.

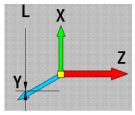
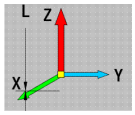
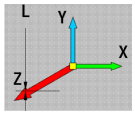
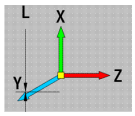
In the machine configuration of the control, the kinematics chain shown above is described with vectors.

The following pages will inform you of all machine parameters necessary for this. In the following table you can see how the kinematics configuration of the example machine is represented in the machine configuration: Usually the axis directions but not the dimensions are specified in the kinematics model.

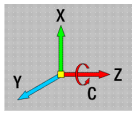
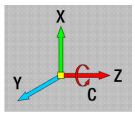
Kinematics configuration of a lathe with the axes X, Z and a C axis

Machine base	Coordinate system
$\mathbf{C}_{mb}$: Coordinate system of the machine base	

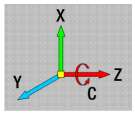
Transformation sequence on the tool side

	location		zDir		xDir		Coordinate system
$\mathbf{C}_Z$: Kinematics of the Z axis	[0]	0	[0]	0	[0]	1	
	[1]	0	[1]	0	[1]	0	
	[2]	0	[2]	1	[2]	0	
$\mathbf{C}_X$: Kinematics of the X axis	location		zDir		xDir		Coordinate system
	[0]	0	[0]	1	[0]	0	
	[1]	0	[1]	0	[1]	1	
	[2]	0	[2]	0	[2]	0	
$\mathbf{C}_Y$: Kinematics of the Y axis	location		zDir		xDir		Coordinate system
	[0]	0	[0]	1	[0]	0	
	[1]	0	[1]	0	[1]	1	
	[2]	0	[2]	0	[2]	0	
$\mathbf{C}_{tool}$: Kinematics of tool system	location		zDir		xDir		Coordinate system
	[0]	0	[0]	1	[0]	0	
	[1]	0	[1]	0	[1]	1	
	[2]	0	[2]	0	[2]	0	

Transformation sequence on the workpiece side for kinematics with a C axis

	location		zDir		xDir		Coordinate system
$\mathbf{C}_c$: Kinematics C axis	[0]	0	[0]	0	[0]	1	
	[1]	0	[1]	0	[1]	0	
	[2]	0	[2]	1	[2]	0	
$\mathbf{C}_{mt}$: Kinematics of machine table	location		zDir		xDir		Coordinate system
	[0]	0	[0]	0	[0]	0	
	[1]	0	[1]	1	[1]	0	
	[2]	0	[2]	0	[2]	1	

Transformation sequence on the workpiece side for kinematics without a C axis

	location		zDir		xDir		Coordinate system
$\mathbf{C}_{mt}$: Kinematics of machine table	[0]	0	[0]	0	[0]	0	
	[1]	0	[1]	1	[1]	0	
	[2]	0	[2]	0	[2]	1	

Transformations on the tool side

The machine kinematics, i.e. the coordinate transformations, are described in the following parameters.

The key names of all axes on the tool side are entered in **MP_axesToolSide.** The sequence of the entries reflects the physical arrangement of the axes. Enter the axis on which the other axes are based at position [0].

The control assumes a three-dimensional kinematics model. If the NC channel does not have all of the three principal axes, replace the missing principal axes by dummy axes.

MP_axesToolSide

	Key names of the axes on the tool side Available from NCK software version: 597 110-01.
Format:	Array [0...9]
Input:	Key names from MP_CfgAxes/axisList (for dummy axes from MP_CfgAxes/specCoordSysList)
Default:	–
Access:	LEVEL3
Reaction:	RESET

Enter the key names of the coordinate transformations on the tool side in **MP_trafoToolSide.** The sequence must correspond to the axes entered in **MP_axesToolSide.**

MP_trafoToolSide

	Coordinate transformations on the tool side Available from NCK software version: 597 110-01.
Format:	Array [0...9]
Input:	Key names for coordinate transformations
Default:	–
Access:	LEVEL3
Reaction:	RESET

Key names of the coordinate transformations defined by direction vectors are entered in **MP_trafoDirToolSide.** They must also be entered in **MP_trafoToolSide,** but cannot appear in **MP_trafoAngleToolSide.**

MP_trafoDirToolSide

	Coordinate transformation defined by direction vectors Available from NCK software version: 597 110-01.
Format:	Array [0...9]
Input:	Key names for coordinate transformations
Default:	–
Access:	LEVEL3
Reaction:	RESET

Key names of transformations defined by angles are entered in **MP_trafoAngleToolSide.** They must also be entered in **MP_trafoToolSide,** but cannot appear in **MP_trafoDirToolSide.**

MP_trafoAngleToolSide

Coordinate transformations defined by angle
Available from NCK software version: 597 110-01.

Format:	Array [0...9]
Input:	Key names for coordinate transformations
Default:	–
Access:	LEVEL3
Reaction:	RESET

MP_toolCoordSys is the end of the kinematics chain on the tool side.

MP_toolCoordSys

Key name of the tool coordinate system
Available from NCK software version: 597 110-01.

Format:	String
Input:	Max. 18 characters
Default:	–
Access:	LEVEL3
Reaction:	RESET

Transformations on the workpiece side

The key names of all axes on the workpiece side are entered in **MP_axesWpSide.** The sequence of the entries reflects the physical arrangement of the axes. Enter the axis on which the other axes are based at position [0].

MP_axesWpSide

Key names of the axes on the workpiece side
Available from NCK software version: 597 110-01.

Format: Array [0...9]
Input: Key names from MP_CfgAxes/axisList (for dummy axes from MP_CfgAxes/specCoordSysList)
Default: –
Access: LEVEL3
Reaction: RESET

Enter the key names of the coordinate transformations on the workpiece side in **MP_trafoWpSide.** The sequence must correspond to the axes entered in **MP_axesWpSide.**

MP_trafoWpSide

Coordinate transformations on the workpiece side
Available from NCK software version: 597 110-01.

Format: Array [0...9]
Input: Key names for coordinate transformations
Default: –
Access: LEVEL3
Reaction: RESET

Key names of the coordinate transformations defined by direction vectors are entered in **MP_trafoDirWpSide** They must also be entered in **MP_trafoWpSide,** but cannot appear in **MP_trafoAngleWpSide.**

MP_trafoDirWpSide

Coordinate transformation defined by direction vectors
Available from NCK software version: 597 110-01.

Format: Array [0...9]
Input: Key names for coordinate transformations
Default: –
Access: LEVEL3
Reaction: RESET

Key names of transformations defined by angles are entered in **MP_trafoAngleWpSide.** They must also be entered in **MP_trafoWpSide,** but cannot appear in **MP_trafoDirWpSide.**

MP_trafoAngleWpSide

Coordinate transformations defined by angle
Available from NCK software version: 597 110-01.

Format: Array [0...9]
Input: Key names for coordinate transformations
Default: –
Access: LEVEL3
Reaction: RESET

MP_machineTableSys is the end of the kinematics chain on the workpiece side.

MP_machineTableSys

Key name of the machine-table coordinate system
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 18 characters
Default: –
Access: LEVEL3
Reaction: RESET

Each coordinate transformation on the workpiece or tool side is defined with direction vectors or angles.

Spindle of the kinematics model

Settings in the configuration editor	MP number
Channels Kinematics CfgKinModel [Key name of the kinematics model] **activeSpindle**	 200011

In **MP_activeSpindle** you specify the spindle used in the kinematics model. This assignment is required for different calculations (for example: calculate feed per revolution, determine tool life, etc.)

MP_activeSpindle

Key name of the active spindle of this kinematics model
Available from NCK software version: 597 110-01.

Format: String
Input: The key name of the spindle is read from CfgAxes/ spindleIndices, e.g. "S", "Spindle1", etc.
Default: –
Access: LEVEL3
Reaction: RUN

Settings in the configuration editor	MP number
Channels ChannelSettings [Key name of the machining channel] CfgActivateKinem **kinemToActivate**	 204001

In **MP_kinemToActivate**, enter the key name of the kinematics model to be activated for this machining channel. The key name of this model must be contained in **MP_ kinModels** in the list of the kinematics models available for this machining channel.

MP_kinemToActivate

Key name of the kinematics model to be activated
Available from NCK software version: 597 110-01.

Format: String
Input: For the key name, please refer to CfgChannelAxes/kinModels, e.g. "K2_CH1_S12", etc.
Default: -
Access: LEVEL3
Reaction: RUN

Settings in the configuration editor	MP number
Channels	
Kinematics	
CfgTrafoByDir	
[Key name of the transformation]	
location	200101
zDir	200102
xDir	200103
CfgTrafoByAngle	
[Key name of the transformation]	
location	200201
angleDef	200202
angle1	200203
angle2	200204
angle3	200205

A coordinate transformation is defined by the description of the position of a coordinate system in the previous coordinate system. This type of position is described by a position vector **(MP_location)** and an orientation. The two principle possibilities for describing the orientation are described below:

Definition of the transformation with direction vectors

MP_location defines the position of the coordinate origin of the transformed system relative to the previous coordinate system.

MP_location

Origin of this coordinate system in the previous system
Available from NCK software version: 597 110-01.

Format: Array [0...2]
Input: -100 000.00000 to +100 000.00000 [mm]
Default: –
Access: LEVEL3
Reaction: RESET

In **MP_zDir** you define the Z direction of the current coordinate system using the previous coordinate system. For more information about the position of the vector **MP_zDir,** See "Overview of machine parameters for the kinematics configuration" on page 503.

MP_zDir

Z-basis vector expressed in the previous coordinate system
Available from NCK software version: 597 110-01.

Format: Array [0...2]
Input: –1 to +1
Enter the Z-basis vector of the transformed coordinate system relative to the previous coordinate system
Note:
Translation axes move in this direction and rotation axes rotate around this vector.
Default: –
Access: LEVEL3
Reaction: RESET

In **MP_xDir** you define the X direction of the current coordinate system using the previous coordinate system. For more information about the position of the vector **MP_xDir,** See "Overview of machine parameters for the kinematics configuration" on page 503.

MP_xDir

X-base vector expressed in the previous coordinate system
Available from NCK software version: 597 110-01.

Format: Array [0...2]
Input: –1 to +1
Enter the X-basis vector of the transformed coordinate system relative to the previous coordinate system.
Default: –
Access: LEVEL3
Reaction: RESET

Definition of the transformation with angles

MP_location defines the position of the coordinate origin of the transformed system relative to the previous coordinate system.

MP_location

Origin of this coordinate system in the previous system
Available from NCK software version: 597 110-01.

Format: Array [0...2]
Input: -100 000.00000 to +100 000.00000 [mm]
Default: –
Access: LEVEL3
Reaction: RESET

MP_angleDef specifies the interpretation of the angles.

MP_angleDef

Specifies the interpretation of the angles
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **Cardan**
Orientation by Cardan angles
RollPitchYaw
Orientation by rotation around fixed axes
Euler
Orientation by Eulerian angles
Default: Cardan
Access: LEVEL3
Reaction: RESET

MP_angle1

Angle 1 – Meaning as per parameter MP_angleDef
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –360.000 to +360.000 [°]
Default: 0
Access: LEVEL3
Reaction: RESET

MP_angle2

Angle 2 – Meaning as per parameter MP_angleDef
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –360.000 to +360.000 [°]
Default: 0
Access: LEVEL3
Reaction: RESET

MP_angle3

Angle 3 – Meaning as per parameter MP_angleDef
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –360.000 to +360.000 [°]
Default: 0
Access: LEVEL3
Reaction: RESET

Settings in the configuration editor	MP number
Aggregates	
ToolMount	
[Key name of the tool holder]	
CfgToolMount	
coorTrafoToModify	600414
coorTrafo	600415
mirroringAxes	600416

Axis mirroring on lathes is required, for example, when carrying out machining operations above or below the turning center or on the rear side. Such axis mirroring is done within a kinematics group; there is no kinematics change.

Axis mirroring is activated/deactivated either via NC commands or by assigning the activation/deactivation to a tool holder.

With the tool holder (parameter object **CfgToolMount**), you can control axis mirroring with the following parameters:

- **MP_coorTrafoToModify**: Kinematics of the axis/axes to be mirrored
- **MP_coorTrafo**: Kinematics of the mirrored axis/axes

Axis mirroring that is configured with the tool holder will be activated when the tool holder is inserted. Axis mirroring will remain active until the next tool change.

6.12 Reference Marks

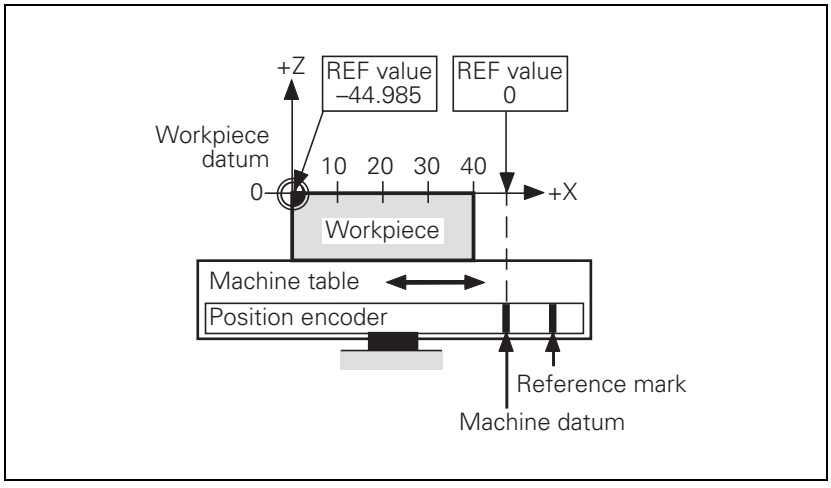
6.12.1 Definition

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

HEIDENHAIN linear encoders are designed with one or more reference marks. The reference marks identify an axis position at a known distance from the machine datum. The position of the freely selectable datum is defined with respect to the machine datum.

The datum and the actual position can be reproduced as soon as the reference marks are traversed.

HEIDENHAIN recommends position encoders with distance-coded reference marks. With distance-coded reference marks, the position value can be reestablished after traverse of a short distance over any two reference marks.



6.12.2 Traversing the reference marks

The reference marks must be traversed after any interruption in power. Specify which axes are homed, and in which sequence, in **MP_refAllAxes** or **MP_refAxis.**

- Press the machine START button: The reference marks are automatically traversed (MP_refAllAxes=True).

or:

- Press the machine axis-direction buttons: The user determines the sequence of the axes (MP_refAllAxes=False).

After the reference marks have been traversed:

- The software limit switches are activated.
- The most recently set datum or workpiece datum and the machine datum are reproduced.

Distance between the scale reference point and the machine datum

For position encoders with distance-coded reference marks, the machine datum is defined with respect to the scale reference point, which is at the first reference mark after the beginning of the measuring length. On angle encoders, the scale reference point is marked:

- In MP_refPosition, enter the distance between the scale reference point and the machine datum.

For position encoders without distance-coded reference marks but with more than one reference mark, every reference mark to be traversed must be evaluated.

- For each reference mark to be traversed, create another parameter set, and enter in MP_refPosition the distance between the scale reference point and the reference mark.
- Activate the parameter set that corresponds to the traversed reference mark.

Assigning a reference value

In some cases it may be necessary to assign a new reference value to an axis, e.g. if an axis is mechanically fixed and the encoder is moved. A typical application is, for example, with (Hirth) clamped axes

Since due to the mechanical fixing the position of the axis cannot be changed, you can assign it a new reference value.

- Enter the new reference value in Module 9147.

Note

Please note the following restriction if you are using the **CC 422** controller unit:

If you are using Module 9147, the following settings in the **MP_refType** parameter (functional sequence for traversing the reference marks) are not permissible when using the CC 422:
MP_refType = **`distance coded + on the fly`**
MP_refType = **`without switch + on the fly`**

Module 9147 Assign a reference value to an axis
If a new reference value is assigned to an axis, the corresponding bit in NN_AxReferenceAvailable (W1032) is reset.

Note

When calling the module for an NC axis during a strobe, the synchronization with the advance calculation (strobe with **MP_sync** = SYNC_CALC) must be configured for this strobe.

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

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	No error
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Invalid axis number
	21	Missing strobe in M4176 = 1
	24	Module was called in a spawn or submit job

Encoders with EnDat interface

Position encoders and speed encoders with EnDat interface can be connected to the MANUALplus 620. With these encoders there is no need to traverse the reference marks. The position value is only read when the MANUALplus 620 is switched on. It cannot be read again.

When connecting a position encoder with EnDat interface, or a speed encoder with EnDat interface as a position encoder:

- Enter **MP_refType = Endat Encoder**

Note

If use of multiturn encoders with EnDat interfaces results in overflows, the corresponding information is stored temporarily. If the control is exchanged, **MP_refPosition** must be re-adjusted.

Renewed traversing of the reference marks

Module 9220 Traverse the reference mark

The module starts the reference mark traverse in an axis or servo-controlled spindle. If the reference mark has already been evaluated, it can be evaluated again by this module. The module can be called in all operating modes.

Constraints:

- Software limit switches are not effective.
- The sequence of functions is determined by **MP_refType.**
- The velocity and the direction for traversing the reference marks are either taken from **MP_CfgReferencing**//refFeedHigh and **MP_CfgReferencing/ refDirection** or they are defined in the module.
- An axis cannot be started for referencing until all other axes are in position.
- If an axis is started for reference point traverse although the reference mark has already been traversed, NN_AxReferenceAvailable is reset and the reference mark is evaluated again. The same constraints apply as when traversing the reference mark for the first time.
- If the spindle is started for reference point traverse, the marker NN_SpiReferenceAvailable is set.
- The spindle must be started from a standstill to traverse the reference mark.

Note

The direction of traverse should be defined in the module only in exceptional cases. Since the reference end positions are not considered in this case, the limits of the traverse range may be violated.

Call:

PS	B/W/D/K	<Axis/spindle> Index from MP_CfgAxes/axisList
PS	B/W/D/K	<Feed rate/shaft speed> 0: Feed rate/shaft speed from MP_CfgReferencing/ refFeedHigh >0: Feed rate in mm/min or shaft speed in 1/1000 min^{-1}
PS	B/W/D/K	<Direction of traverse> –1: Negative direction 0: Direction from MP_CfgReferencing/refDirection 1: Positive direction
CM	9220	
PL	B/W/D	<Error code> 0: Reference mark traverse is commanded 1: Axis does not exist, or not a closed-loop spindle 2: Inadmissible values for the feed rate / direction 3: Incorrect operating mode 4: Reference traverse already started 5: Axis is already being positioned or the spindle is in motion 6: Other axis is already being positioned 8: Programmed axis not in closed loop

6.12.3 Traversing the reference marks

The reference marks must be traversed after any interruption in power. Specify which axes are homed, and in which sequence, in **MP_refAllAxes** or **MP_refAxis.**

- After acknowledging the power interruption with the CE key or pressing the Control voltage On key, you can preselect individual axes to be homed or use the ALL soft key.
- After the NC start key has been pressed, the axes are homed one after the other in the sequence defined in parameter **MP_refAxis**.

After the reference marks have been traversed:

- The software limit switches are activated.
- The most recently set datum or workpiece datum and the machine datum are reproduced.

Distance between the scale reference point and the machine datum

For position encoders with distance-coded reference marks, the machine datum is defined with respect to the scale reference point, which is at the first reference mark after the beginning of the measuring length. On angle encoders, the scale reference point is marked:

- In MP_refPosition, enter the distance between the scale reference point and the machine datum.

For position encoders without distance-coded reference marks but with more than one reference mark, every reference mark to be traversed must be evaluated.

- For each reference mark to be traversed, create another parameter set, and enter in MP_refPosition the distance between the scale reference point and the reference mark.
- Activate the parameter set that corresponds to the traversed reference mark.

Assigning a reference value

In some cases it may be necessary to assign a new reference value to an axis, e.g. if an axis is mechanically fixed and the encoder is moved. A typical application is, for example, with (Hirth) clamped axes

Since due to the mechanical fixing the position of the axis cannot be changed, you can assign it a new reference value.

- Enter the new reference value in Module 9147.

Note

Please note the following restriction if you are using the **CC 422** controller unit:

If you are using Module 9147, the following settings in the **MP_refType** parameter (functional sequence for traversing the reference marks) are not permissible when using the CC 422:
MP_refType = **`distance coded + on the fly`**
MP_refType = **`without switch + on the fly`**

Module 9147 Assign a reference value to an axis
If a new reference value is assigned to an axis, the corresponding bit in NN_AxReferenceAvailable (W1032) is reset.

Note

When calling the module for an NC axis during a strobe, the synchronization with the advance calculation (strobe with **MP_sync** = SYNC_CALC) must be configured for this strobe.

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

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	No error
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Invalid axis number
	21	Missing strobe in M4176 = 1
	24	Module was called in a spawn or submit job

Encoders with EnDat interface

Position encoders and speed encoders with EnDat interface can be connected to the MANUALplus 620. With these encoders there is no need to traverse the reference marks. The position value is only read when the MANUALplus 620 is switched on. It cannot be read again.

When connecting a position encoder with EnDat interface, or a speed encoder with EnDat interface as a position encoder:

- Enter **MP_refType = Endat Encoder**

Note

If use of multiturn encoders with EnDat interfaces results in overflows, the corresponding information is stored temporarily. If the control is exchanged, **MP_refPosition** must be re-adjusted.

Renewed traversing of the reference marks

Module 9220 Traverse the reference mark

The module starts the reference mark traverse in an axis or servo-controlled spindle. If the reference mark has already been evaluated, it can be evaluated again by this module. The module can be called in all operating modes.

Constraints:

- Software limit switches are not effective.
- The sequence of functions is determined by **MP_refType.**
- The velocity and the direction for traversing the reference marks are either taken from **MP_CfgReferencing**//refFeedHigh and **MP_CfgReferencing/ refDirection** or they are defined in the module.
- An axis cannot be started for referencing until all other axes are in position.
- If an axis is started for reference point traverse although the reference mark has already been traversed, NN_AxReferenceAvailable is reset and the reference mark is evaluated again. The same constraints apply as when traversing the reference mark for the first time.
- If the spindle is started for reference point traverse, the marker NN_SpiReferenceAvailable is set.
- The spindle must be started from a standstill to traverse the reference mark.

Note

The direction of traverse should be defined in the module only in exceptional cases. Since the reference end positions are not considered in this case, the limits of the traverse range may be violated.

Call:

PS	B/W/D/K	<Axis/spindle> Index from MP_CfgAxes/axisList
PS	B/W/D/K	<Feed rate/shaft speed> 0: Feed rate/shaft speed from MP_CfgReferencing/ refFeedHigh >0: Feed rate in mm/min or shaft speed in 1/1000 min^{-1}
PS	B/W/D/K	<Direction of traverse> –1: Negative direction 0: Direction from MP_CfgReferencing/refDirection 1: Positive direction
CM	9220	
PL	B/W/D	<Error code> 0: Reference mark traverse is commanded 1: Axis does not exist, or not a closed-loop spindle 2: Inadmissible values for the feed rate / direction 3: Incorrect operating mode 4: Reference traverse already started 5: Axis is already being positioned or the spindle is in motion 6: Other axis is already being positioned 8: Programmed axis not in closed loop

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgReferencing	
refType	400401
refSwitchActive	400404
refFeedLow	400406
refFeedHigh	400407
refDirection	400408

The parameter object CfgReferencing is not required for:

- Virtual axes (MP_axisMode = Virtual)

You define the process of traversing the reference marks in the following machine parameters:

- In **MP_refDirection** and **MP_refFeedHigh** (for rotary encoders also in MP_refFeedLow) you define the direction and velocity for traversing the reference marks.
- In **MP_refAxis**, define the sequence of axes for traversing the reference marks.
- In **MP_refType** you select the functional sequence (type of reference marks) for each axis.

MP_refType

Sequence for finding the reference mark
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **None**
No traversing of the reference marks
If the reference run with the spindle is performed via a trip dog, this value must be set.
Switch, changing Dir
For linear axes with speed encoder; reference run with NC start
Switch, no changing Dir
For linear axes with speed encoder; reference run with NC start
without Switch
For spindle, rotary table with angle encoder; reference run with NC start
distance coded
For distance-coded linear encoders; reference run with NC start
distance coded + on the fly
For distance-coded linear encoders; reference run with axis-direction keys or NC start
without switch + on the fly
For spindle; reference run with M3, M4
Endat Encoder
For axes with EnDat encoder; reference-mark traverse not necessary

Default: Switch, changing Dir

Access: LEVEL3

Reaction: REF

Direction and velocity

In **MP_refDirection** you specify the direction of traverse. If the axis traverses the reference-end-position trip dog, and **PP_AxReferenceEndPosition** is set, the direction of traverse is reversed.

In **MP_refFeedHigh** and **MP_refFeedLow,** define the velocity for traversing the reference marks.

It depends on the entry in **MP_refType** whether the low or high reference-run velocity is used.

MP_refDirection

Direction for traversing the reference marks
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **Positive**
Positive traverse direction
Negative
Negative traverse direction
Default: Negative
Access: LEVEL3
Reaction: REF

MP_refFeedLow

Low speed when finding the reference mark
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 10.000 000 000 to 36 000 000 [mm/min]
Default: 600
Access: LEVEL3
Reaction: REF

MP_refFeedHigh

High velocity for traversing the reference mark
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 80.000 000 000 to 36 000 000 [mm/min]
Default: 1200 [mm/min]
Access: LEVEL3
Reaction: REF

The parameter **MP_refSwitchActive** defines the status of the trip dog for reference end position.

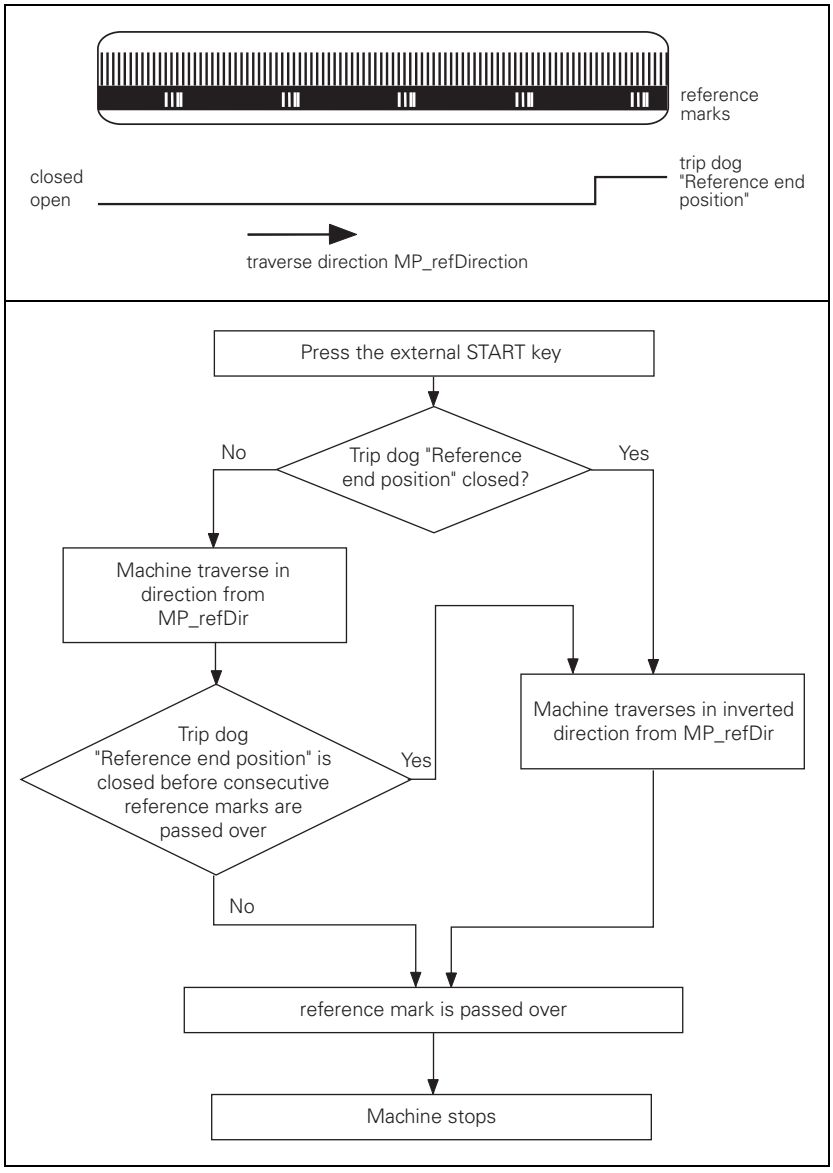
MP_refSwitchActive

Active level of the trip dog for reference end position
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **high**
Reference-end-position trip dog is active at high level
low
Reference-end-position trip dog is active at low level
Default: high
Access: LEVEL3
Reaction: REF

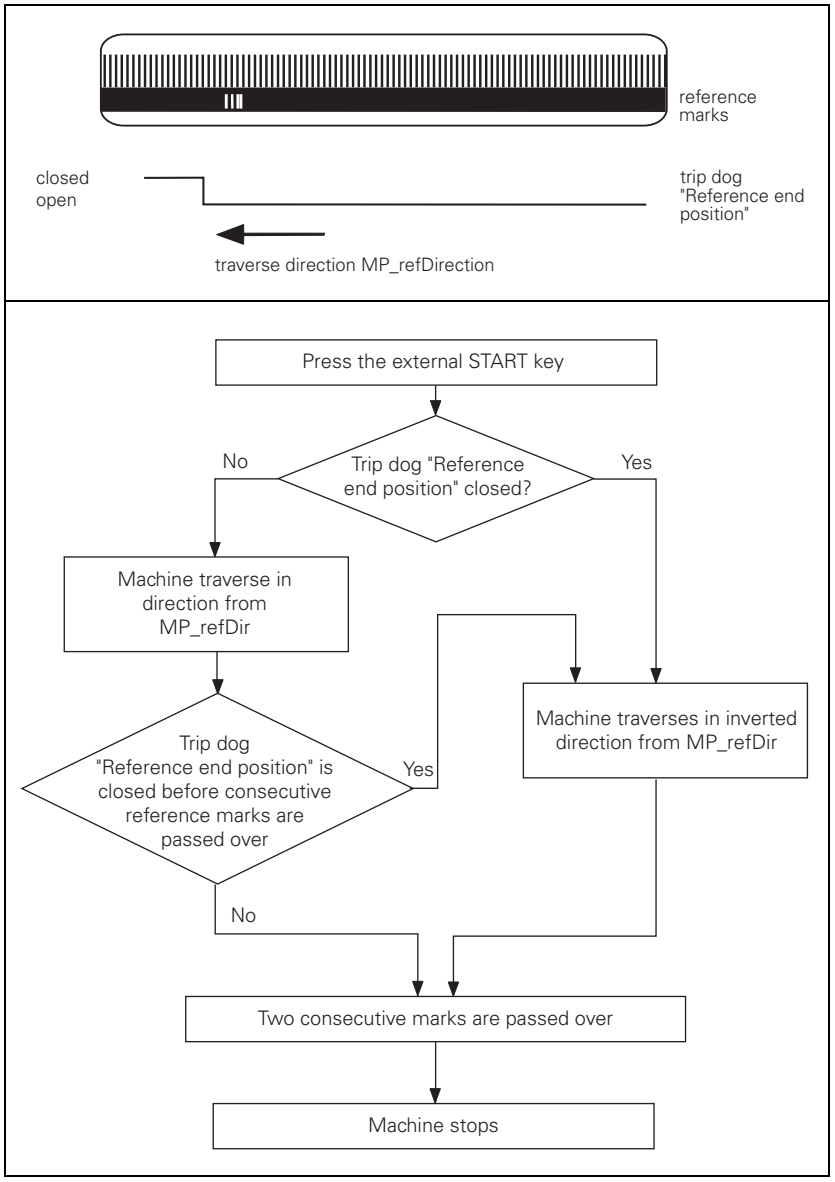
Position encoder with distance-coded reference marks

Functional sequence if **MP_refType=distance coded**



Position encoder with one reference mark

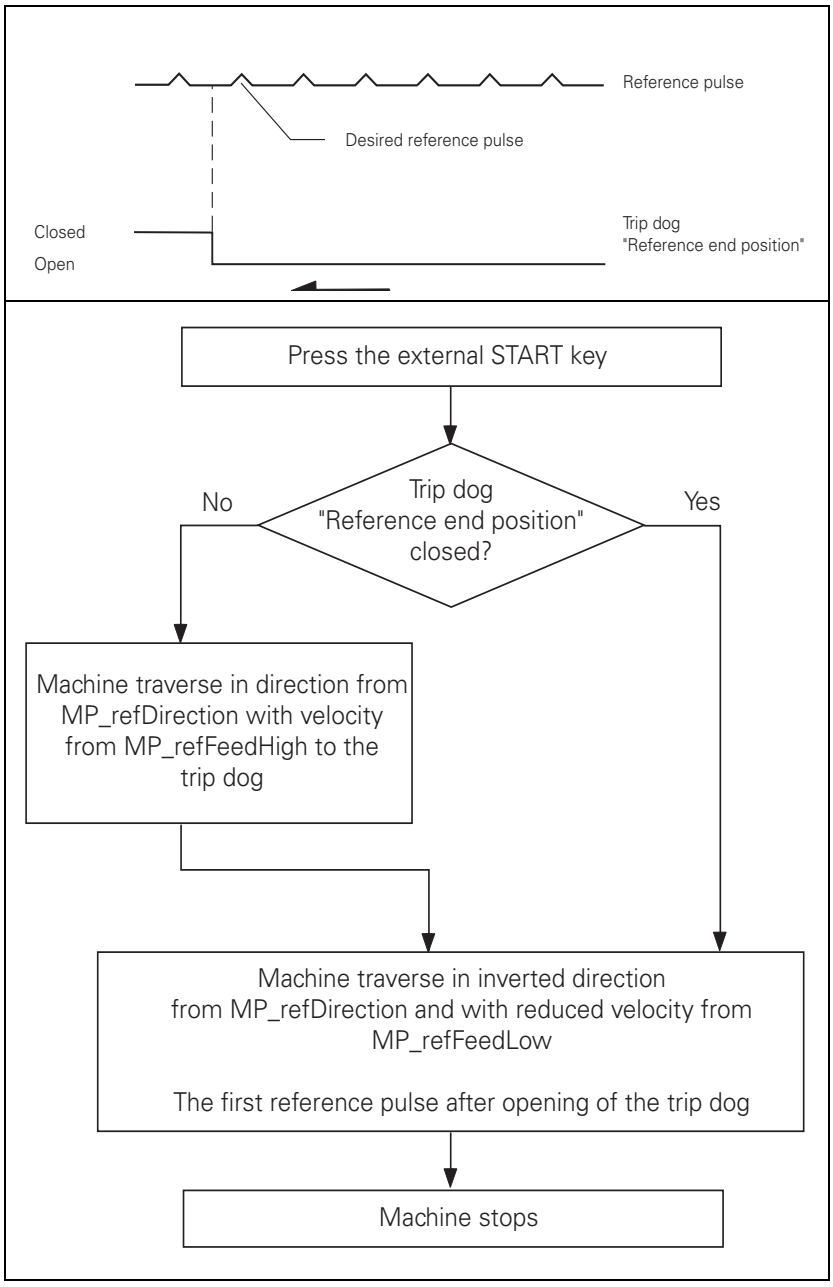
Functional sequence if **MP_refType=Switch, no changing Dir**



Linear measurement through rotary encoder

Functional sequence if **MP_refType=Switch, changing Dir**

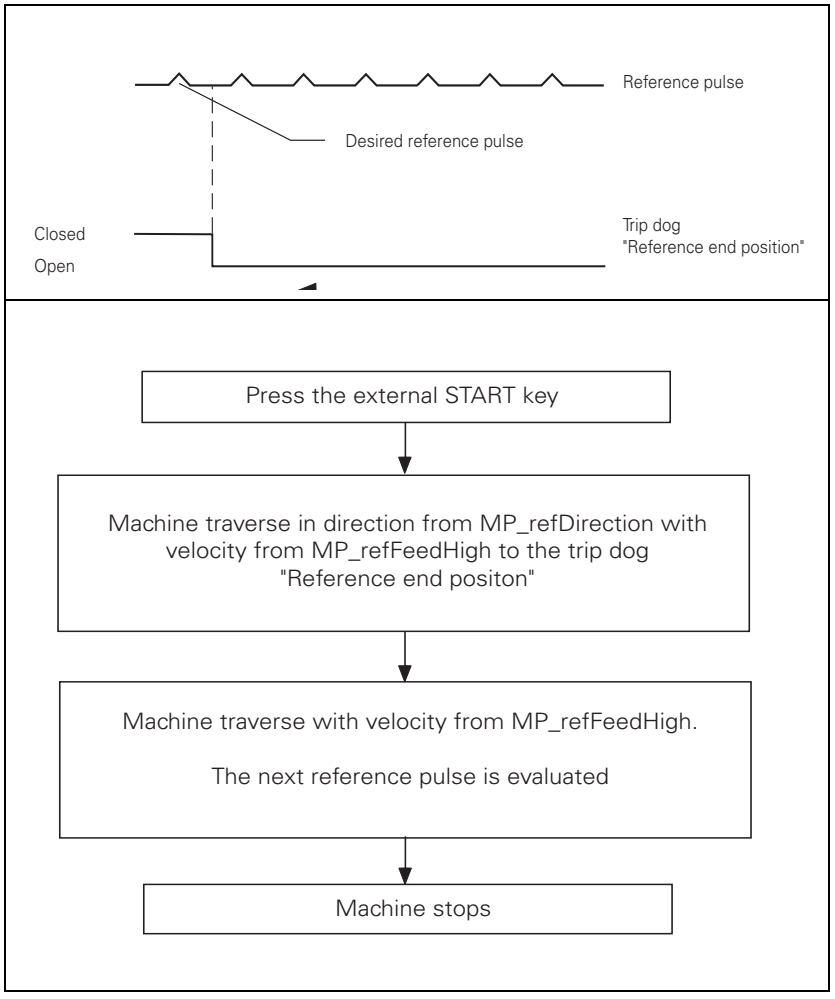
For linear measurement using a rotary encoder, a reference pulse is produced at each revolution of the encoder. Ensure that during referencing the same reference pulse is always evaluated. This can be realized with the trip dog for reference end position.



Functional sequence if **MP_refType=without Switch**

For linear measurement using a rotary encoder, a reference pulse is produced at each revolution of the encoder. During the reference run the first reference pulse traversed after the trip dog for reference end position is closed is evaluated. This ensures that the same reference pulse is always evaluated.

For linear measurement using a rotary encoder, HEIDENHAIN recommends using the **MP_refType=Switch, changing dir** method.



Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgReferencing	
refPosition	400403
moveAfterRef	400409
moveAfterRefType	400410
moveAfterRefPos	400411
moveAfterRefFeed	400412

The parameter object CfgReferencing is not required for:

- Virtual axes (MP_axisMode=Virtual)

In **NN_OmgReference**, the NC informs the PLC of the **`Pass Over Reference Point`** operating mode. In **NN_AxReferenceAvailable,** the NC reports whether the reference marks of this axis were traversed.

If you switch the operating mode before all reference marks are traversed, the MANUALplus 620 identifies this state and prompts you to traverse the remaining reference marks.

Reference end position

To prevent the axes from violating their traverse limits when traversing the reference marks, each axis requires a trip dog (at the reference end position). The trip dogs must be installed by the machine tool builder at the ends of the traverse range. The switch signals from the trip dogs are sent to free PLC inputs. The PLC program must gate these PLC inputs with **PP_AxReferenceEndPosition** for "reference end position." Setting the reference end position causes a reversal of the traverse direction from **MP_refDirection.**

PLC operand / Description	Type
NN_AxReferenceAvailable 0: Reference mark not traversed 1: Reference mark traversed	M
PP_AxReferenceEndPosition 0: Trip dog not triggered 1: Trip dog triggered	M

Machine datum

MP_refPosition defines the position of the machine datum relative to the reference point of the scale. For encoders with distance-coded reference marks, the position is relative to the scale reference point; for encoders with EnDat interface, relative to the absolute encoder datum.

MP_refPosition

	Position of the machine datum Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	–100 000.000 000 000 to +100 000 [mm] or [°]
Default:	100
Access:	LEVEL3
Reaction:	REF

Positioning after reference mark traverse

The axis can automatically be moved to a certain position after reference mark traverse is completed. This behavior is activated with **MP_moveAfterRef.**

Define the following information for positioning after reference mark traverse:

- The type of movement after finding the reference mark: absolute, relative or positioning on the Hirth grid
- In **MP_moveAfterRefPos** the end position.
- In **MP_moveAfterRefFeed**, the feed rate.

MP_moveAfterRef

Activate movement after finding the reference mark
Available from NCK software version: 597 110-02.

Format: Selection menu
Selection: **on**
Activate positioning after reference-mark traverse
off
No positioning after reference-mark traverse
Default: off
Access: LEVEL3
Reaction: REF

MP_moveAfterRefType

Type of movement after finding the reference mark
Available from NCK software version: 597 110-02.

Format: Selection menu
Selection: **absolute**
Absolute positioning This function is mainly intended for positioning rotary tables. Please ensure that no collision occurs as a result of this positioning. The software limit switches are already active.
relative
Incremental positioning
HirthRasterPos
Approach next Hirth grid position in positive direction.
HirthRasterNeg
Approach next Hirth grid position in negative direction.
Default: absolute
Access: LEVEL3
Reaction: REF

MP_moveAfterRefPos

Position for positioning after traversing the reference mark
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 000 000 to +100 000 [mm] or [°]

Default: 0
Access: LEVEL3
Reaction: REF

MP_moveAfterRefFeed

Feed rate for positioning after traversing the reference mark
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 10.000 000 000 to 36 000 000 [mm/min] or [°/min]
Default: 6 000
Access: LEVEL3
Reaction: REF

6.13 The Control Loop

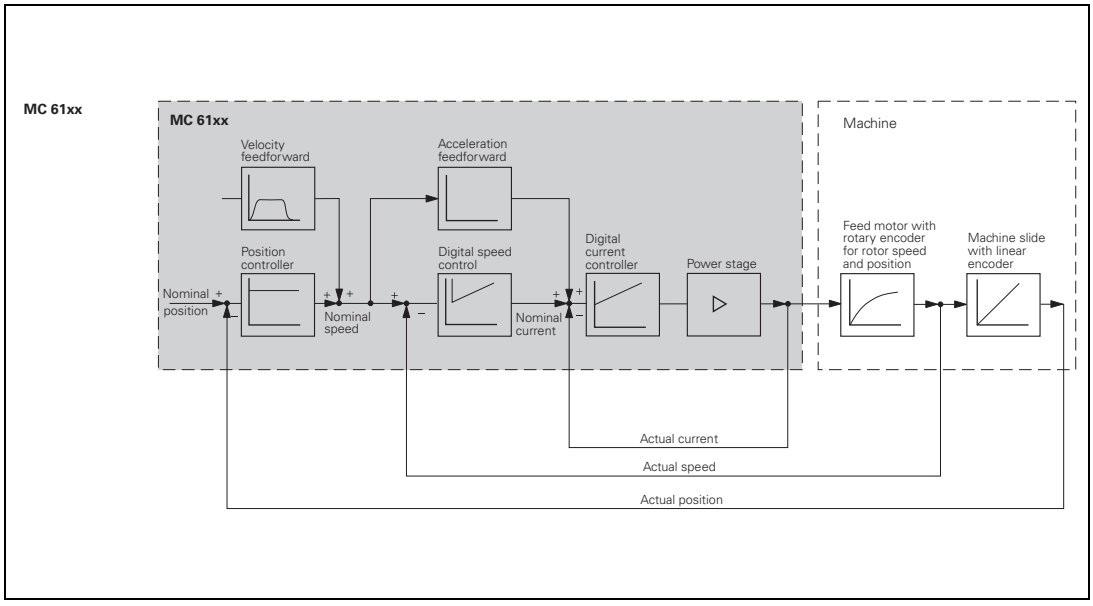
Machine tools normally function on the principle of cascade control. Here the position control loop is prior to the speed and current control loops.

Benefits of cascade control:

- Transparent structure of the individual control loops.
- Disturbances can be compensated through the subsequent controllers. This relieves the prior controller.
- The respective outer control loop protects the inner control loop by limiting the command variable.
- Individual commissioning of each control loop, starting with the innermost loop.

6.13.1 Block diagram of control loop

The position, speed, and current controllers are integrated in the MANUALplus 620. The power module is driven by the CC controller unit through PWM signals (PWM = pulse width modulation).



There is a separate time interval for each control loop:

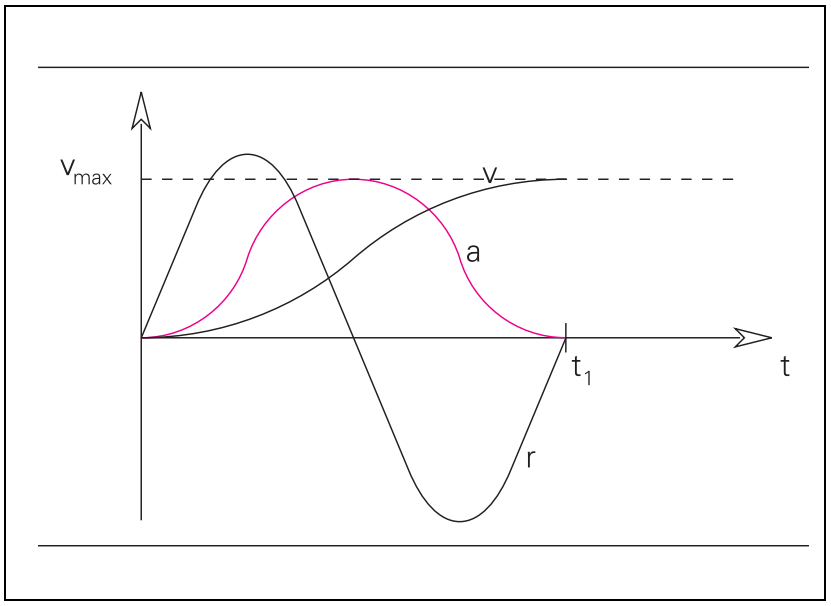
- **Position controller cycle time:** Time interval during which the interpolation points on the path are calculated.
- **Speed controller cycle time:** Time interval in which the actual speed value is compared to the calculated nominal speed value.
- **Current controller cycle time:** Time interval in which the actual current value is compared to the calculated nominal current value.

The cycle times that apply vary depending on the CC used or the CC's settings.

6.13.2 Relation between jerk, acceleration, velocity and distance

Acceleration and jerk

Taking into account the motor and the power module, the machine should be designed in such a way that acceleration during the acceleration phase is as constant as possible. This ensures maximum utilization of the drive current. On the other hand, the machine should also be designed to fulfill the dynamic requirements. The jerk should be kept to a minimum and the jerk phase should be maximized in order to prevent the machine from oscillating. This results in a bell-shaped acceleration curve (see figure).



Legend:

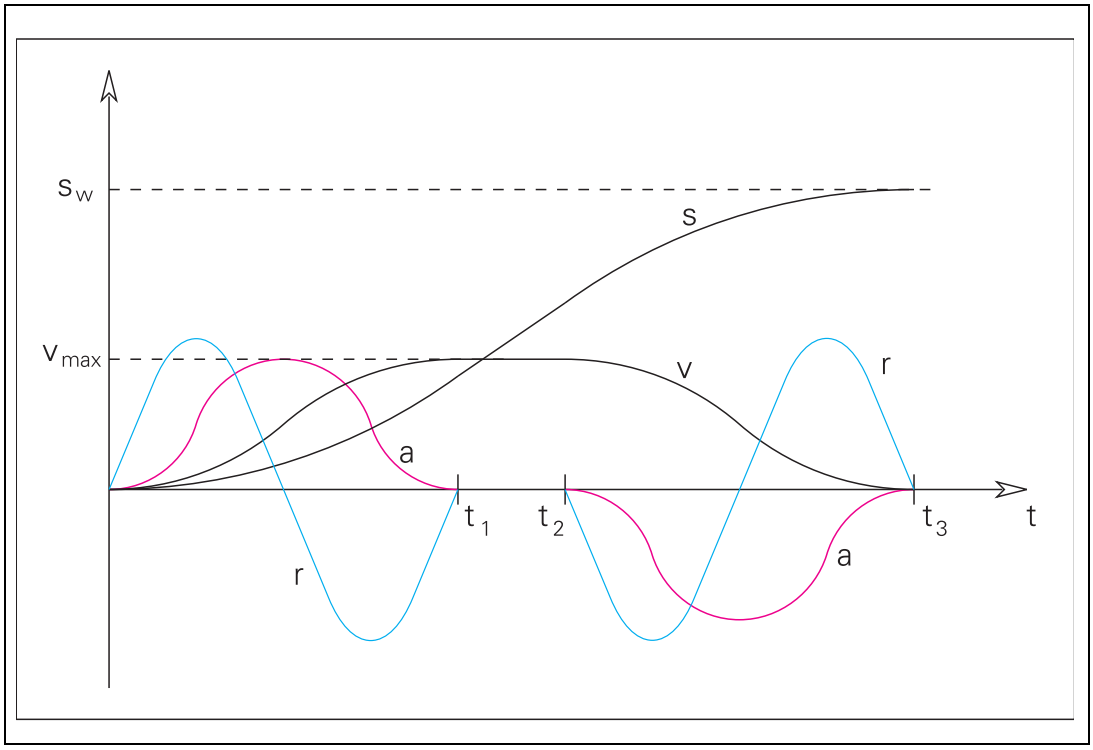
- v: Velocity
- a: Acceleration
- r: Jerk
- v_{max}: Maximum velocity

Distance

To attain the maximum velocity, a certain minimum distance must be traversed. This also applies to the braking phase.

If the traverse distance is greater than the distance covered during the acceleration and braking phases, a movement at constant (maximum) speed is inserted (see period of time from t_1 to t_2 in the figure below).

If the traverse distance is shorter, the maximum velocity is not attained.



Legend:

- v: Velocity
- a: Acceleration
- r: Jerk
- s: Distance
- v_{max}: Maximum velocity
- S_w: Traverse distance
- t_1: End of acceleration phase
- t_2: Start of braking phase
- t_3: End of traverse distance

Settings in the configuration editor	MP number
System	
CfgFilter	
typeFilter1	100401
orderFilter1	100402
typeFilter2	100403
orderFilter2	100404
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgPositionFilter	
filter1Shape	401601
filter2LimitFreq	401602
filter2Shape	401603
filter2LimitFreq	401604

To attain a high machining velocity while maintaining a high surface quality, the workpiece contour can be adapted to the machine dynamics by means of a nominal position value filter.

Two types of low-pass filters are available for limiting the bandwidth of the dynamics of nominal position and speed values.

- Average (mean-value filter)
- Triangle filter

Mean-value and triangle filter

The mean-value and triangle filters are classic low-pass filters that always smooth a contour towards the inside at changes in direction. They can be used when a very high surface quality is required, or when a high machining speed is required and larger tolerances are permitted.

HEIDENHAIN recommends: Do not use the mean-value filter. This filter has a low edge steepness. Prefer the triangle filter instead.

Depending on the cycle time, tolerance and axis-jerk limit values, the control automatically increases the filter frequency actually used.

Selection criteria for the nominal position value filters

The settings for the nominal position value filters mainly depend on the emphasis of the requirements for machining the workpiece. Speed and accuracy, in connection with clean and smooth surfaces, are the decisive criteria.

At the same time, the oscillation and resonance tendencies of the machining system (the machine tool) are to be considered, and taken into account in the settings for the nominal position value filters. The following recommendations can be made for the settings:

Clean surface

Definition of the term "surface":

- A clean and smooth surface has the highest priority
- Application: Finishing
- Oscillations in the axes must be damped, since following errors of 1 µm are still visible on the surface
- Tolerances are typically between 0.01 and 0.02 mm (may be slightly exceeded in order to achieve a better surface).

Settings guidelines for "surface":

- Low jerk values **(MP_maxPathJerk)**
- Switch off consideration of the tolerance for curvature changes **(MP_pathTolerance)**
- High jerk values for MP_axJerk so that no limitations take effect (example: test up to a value of 1000)

Accuracy

Definition of the term "accuracy":

- Maintaining the tolerances has the highest priority
- Slight oscillations can be seen on the surface
- Tolerances are typically between 0.005 and 0.01 mm.

Settings guidelines for "accuracy":

- Lower jerk values than for "speed"
- Ideally, circular paths should be checked with a KGM grid encoder from HEIDENHAIN. However, in many cases the circular interpolation test with the integrated oscilloscope or TNCopt suffices.
- **MP_pathTolerance** = 1 (consideration of tolerance limits at curvature changes)
- The adjustment should be tested with suitable NC programs. The **TNCopt** software from HEIDENHAIN features suitable NC programs for this. The advantage is that the speed and the contour deviations can be seen directly.

Speed

Definition of the term "speed":

- Surface quality is secondary; short machining times have the highest priority
- Application: Roughing
- Tolerances typically between 0.1 and 0.2 mm

Settings guidelines for "speed":

- Filter selection
  - For large tolerances (greater than 50 µm), preferential use of the triangle filter (MP_filter1Shape or MP_filter2Shape = Triangle)
- High jerk values
- The adjustment should be tested with suitable NC programs. The **TNCopt** software from HEIDENHAIN features suitable NC programs for this. The advantage is that the speed and the contour deviations can be seen directly.
- When setting the jerk and acceleration values, as well as selecting the suitable filters, take into account
  - the running noises of the machine
  - the mechanical load (wear)
  - the desired machining speed.

Global settings of the nominal position value filters

Two filters are located before the position control loop to prevent the machine from oscillating.

- **MP_typeFilter1** only affects **rotary axes.** This filter is usually not required for lathes. In this case you can enter "off" for the parameter.
- **MP_typeFilter2** only affects **linear axes.**

With **MP_orderFilter1** and **MP_orderFilter2** you define the global order of the filter. Then you define for each axis whether a filter is used and which filter is used to optimize the axis. In addition, the frequency of the filter is defined axis-specifically.

The smoothing function of these filters causes contour errors. The velocity profile is adjusted by the look-ahead function so that the contour error does not exceed the given tolerance (see "Look-ahead" on page 547).

Note

Filters delay the processing time of NC blocks by the control, since multiple NC blocks must be considered for the filter functions.

Take the machine setup into account when you configure filters.

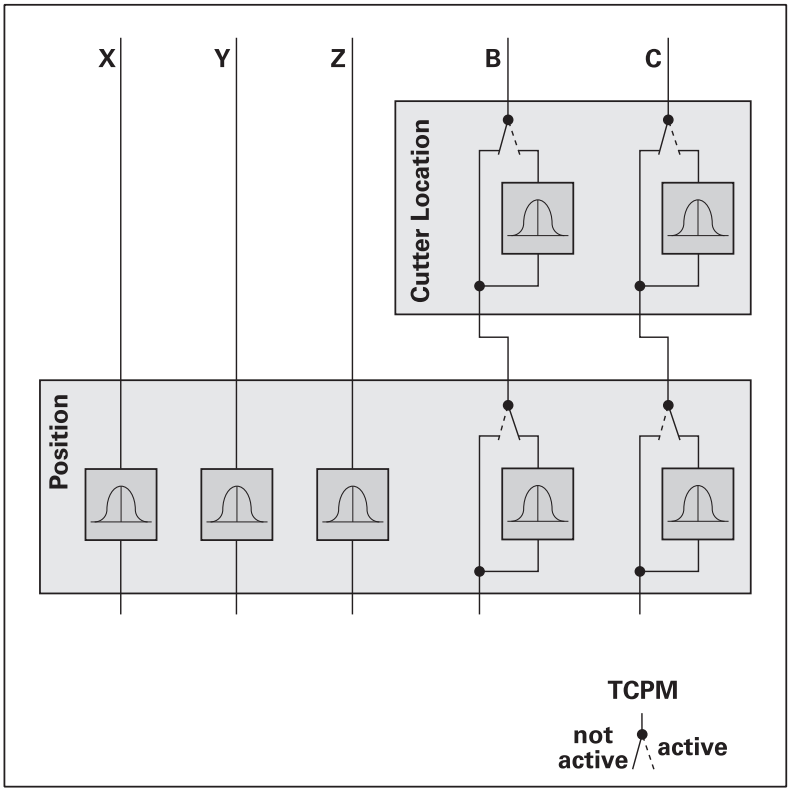
HEIDENHAIN recommends that you configure the filters of standard machines (TCPM is not used) as follows:

- Define **MP_typeFilter1** = Off.
- Define **MP_typeFilter2** = Position and specify the filter order in **MP_orderFilter2.**
- For **rotary and linear axes** you define the triangle filter **MP_filter2Shape** = Triangle. Enter in **MP_filter2LimitFreq** the limit frequency of the machine.

Function of the nominal position value filters

The function of the filters for rotary axes depends on TCPM (see figure):

- TCPM is not active:
  - The filters of the "cutter location" type are not active
  - The filters of the "position" type are active for all axes
- TCPM is active (not available on lathes):
  - The filters of the "cutter location" type are active
  - The filters of the "position" type are not active for rotary axes



This method provides the following advantage:

- The values optimized by the cutter-location-point filter are used as the basis for the filter for linear axes. This means that the rotary axis values that have already been corrected are evaluated by the linear axis filter.

Recommended settings for filter limit frequency and filter order

On the MANUALplus 620 the filter limit frequency is set via the globally effective filter order. In the following you will find recommendations on how to set the filter order.

Triangle and mean-value filter:

The higher you set the filter order, the more "relaxed" the machine behaves—this always results in a degree of inaccuracy. In addition, the higher the filter order, the more the speed of the axes is reduced.

These settings must always be considered in relation to their results. The speed at corners and arcs is reduced because of the tolerance.

However, the behavior is exactly the opposite at contour transitions (straight line to arc, arc to arc): the "relaxed" filter settings make contour transitions at higher speeds possible.

You must perform tests to find the best compromise between high transition speeds and high curve speeds for each machine type.

Settings of the nominal position value filters

Now that the basic global settings for the nominal position value filter have been made, the filter and look-ahead parameters can be specified.

- The control can distinguish between machining and rapid-traverse positioning movements. This makes it possible to optimize the path control for positioning movements at rapid traverse:
  Define the maximum machining feed rate in the **MP_maxG1Feed** parameter. As with rapid traverse, the parameters **MP_maxPathJerkHi** and **MP_pathToleranceHi** apply as of this feed rate.
- Enter the permissible axis-specific jerk:
  MP_maxPathJerk: Maximum jerk for acceleration processes.
  MP_axJerk: Is effective for acceleration processes due to the workpiece geometry (at curvature changes, e.g. tangential transition from a line to an arc)
- Enter the permissible axis-specific jerk for rapid traverse in the parameter **MP_maxPathJerkHi**. This value also applies to feed rates greater than the parameter **MP_maxG1Feed.**
- In the **MP_pathTolerance** parameter you must define the tolerance for contour transitions with motions at the machining feed rate. The control monitors the contour deviation to ensure that it does not exceed this maximum.
- In the parameter **MP_pathToleranceHi** you define a tolerance for contour transitions with motions at rapid traverse. The value also applies to feed rates greater than the limit value in **MP_maxG1Feed.** This tolerance can **not** be overwritten by the machine user with Cycle 32 "Tolerance."
- When selecting the limit frequencies (parameters **MP_filter1LimitFreq** and **MP_filter2LimitFreq**) for mean-value filters and triangle filters, take into account the lowest resonant frequency of your machine's axes and the desired damping at this frequency. If the limit frequency is set to 0, the filter is switched off.
  MP_filter1LimitFreq and **MP_filter2LimitFreq** have no significance for the Triangle and Average filter types. You can switch off these filters only if you enter the value **`Off`** in **MP_filter1Shape** and **MP_filter2Shape**.

Note

The tolerance (**MP_pathTolerance**) always refers to the nominal value, meaning the servo lag also affects the contour accuracy.
For example, if the servo lag S = 5 µm and the tolerance T = 10 µm, then the total deviation is 15 µm.

- With the parameters **MP_filter1Shape** and **MP_filter2Shape,** select from the mean-value and triangle filters the nominal position value filters for the **`Program Run, Single Block, Program Run, Full Sequence`** and **`Positioning with Manual Data Input`** operating modes. The triangle filter (triangle) smoothes the contour towards the inside at changes in direction. This results in an excellent surface quality without overshoot. The mean-value filter (average) is a very smooth filter with low edge steepness, which should be used only for testing, and not for machining.
- With the parameter **MP_manualFilterOrder,** select the order of the mean-value filter for the **`Manual, Electronic Handwheel, Jog Increment`** and **`Pass Over Reference Point`** operating modes.
- In order to achieve the optimum results for your machine or application, test the various filter settings with a test part consisting of short, straight paths.

MP_typeFilter1

Type of the first nominal position value filter
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **Off**
Filter 1 is deactivated (recommended for lathes)
Position
Axis position (for linear and rotary axes)
CutterLocation
For rotary axes
Default: **Off**
Access: LEVEL3
Reaction: RESET

MP_orderFilter1

Order of first nominal position value filter
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 1 to 31
Default: 11
Access: LEVEL3
Reaction: RESET

MP_typeFilter2

Type of the second nominal position value filter
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **Off**
Filter 2 is not active
Position
Axis position (for linear and rotary axes)
(recommended for lathes)
CutterLocation
for rotary axes (do not use for lathes)
Default: Position
Access: LEVEL3
Reaction: RESET

MP_orderFilter2

Order of second nominal position value filter
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 1 to 31
Default: 11
Access: LEVEL3
Reaction: RESET

MP_filter1Shape

Form of the first nominal position value filter
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **Off**
Turned off
Average
Mean-value filter
Triangle
Triangle filter
HSC
High-speed cutting filter (cannot be used for lathes)
Default: Off
Access: LEVEL3
Reaction: RUN

MP_filter1LimitFreq

Limit frequency of the first nominal position value filter
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 10.000 000 000 to 100 [Hz]
Default: 66 [Hz]
Access: LEVEL3
Reaction: RUN

MP_filter2Shape

Form of the second nominal position value filter
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **Off**
Turned off
Average
Mean-value filter
Triangle
Triangle filter
HSC
High-speed cutting filter (cannot be used for lathes)
Default: Off
Access: LEVEL3
Reaction: RUN

MP_filter2LimitFreq

Limit frequency of the second nominal position value filter
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 10.000 000 000 to 100 [Hz]
Default: 40 [Hz]
Access: LEVEL3
Reaction: RUN

Note

- For filters of the "position" type, the cutoff frequencies of the axes should not differ significantly.
- Values between 20 Hz and 40 Hz are recommended for the cutoff frequencies of the second nominal position value filter.

6.13.4 Look-ahead

Under consideration of certain limit values, **look-ahead** cyclically calculates the maximum possible contouring speed up to 5000 blocks in advance.

The calculated values are transferred to the **interpolator** in feed-rate profiles. The interpolator calculates axis-specific nominal values from the position polynomials and feed-rate profile.

The programmed contouring feed rate, maximum axis accelerations, permissible axis/path jerk, filter parameters and tolerances are taken into account in these feed-rate profiles. The feed-rate profiles are also influenced by changes to the override potentiometer, and by whether SINGLE BLOCK or FULL SEQUENCE is active.

Small variations in the feed rate, which appear during calculation of the feed-rate profile, are suppressed in order to achieve a smooth feed rate.

Contour smoothing

In order to achieve smooth machining surfaces with a minimum of machining time, the following must be kept in mind:

- Each jerk (da/dt), which is caused by a change in direction on the contour, or by a change in the acceleration or in the feed rate, excites vibrations in the machine. Therefore, the jerk must be limited to a permissible size.
- For feed rates above the machining feed rate, an increased jerk and increased tolerance are both permissible, since they no longer have any effect on the machining quality.
- The tool may go to the limits of the adjustable path tolerance (deviation from the contour), but must not exceed the tolerance.
- Each machine axis is programmed for a certain maximum feed rate, and has a specified capability for acceleration. For interpolating axes, the acceleration of the slowest axis is decisive.
- Feed rates must not fall beneath the minimum value.

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgLaPath	
minPathFeed	201501
minCornerFeed	201502
maxG1Feed	201503
maxPathJerk	201504
maxPathJerkHi	201505
pathTolerance	201506
pathToleranceHi	201507
maxPathYank	201508

- In **CfgLaPath**, define the path-specific limit values for feed rate, acceleration and jerk.

MP_minPathFeed allows you to define the minimum feed rate in an NC block. The parameter is intended for technological purposes, in particular for preventing dwell marks in small radii. If this setting is too high, it will disable the feed-rate optimization function and can cause excessive dynamic load on the machine.

Note

As a rule, **MP_minPathFeed** is limited to the programmed feed rate. The look-ahead will only go below the defined feed-rate value if the lower feed rate is programmed in the NC block or has been set with the override potentiometer.

MP_minPathFeed

Minimum feed rate on the path
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 600 000 [mm/min]
Default: 60 [mm/min]
Access: LEVEL3
Reaction: RUN

In **MP_minCornerFeed** you can define the minimum feed rate at block transitions (corners and line-to-arc transitions). The parameter is intended for technological purposes, in particular for preventing dwell marks in corners. If this setting is too high, it will disable the feed-rate optimization function and can cause excessive dynamic load on the machine.

If the value defined in **MP_minCornerFeed** is greater than **MP_minPathFeed**, **MP_minPathFeed** will also be effective for the minimum feed rate at block transitions.

Note

As a rule, **MP_minCornerFeed** between two segments is limited to the programmed feed rate. The look-ahead will only go below the defined feed-rate value if the lower feed rate is programmed in the NC block or has been set with the override potentiometer.

MP_minCornerFeed

	Minimum feed rate at corners Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 600 000 [mm/min]
Default:	30 [mm/min]
Access:	LEVEL3
Reaction:	RUN

MP_pathTolerance limits the feed rate at corners and curvatures. The parameter specifies the degree to which corners are rounded and defines the maximum deviation from the circle radius. This keeps the errors of nominal position value filter within certain limits. To meet the defined tolerance, the MANUALplus 620 reduces the feed rate at corners, circles and curved contours.

MP_pathTolerance includes errors caused by the filter before the position control loop. The feed-rate override does not affect which jerk or tolerance is in effect.

The tolerance defined in **MP_pathToleranceHi** is effective for feed rates exceeding than those defined in **MP_maxG1Feed**.

MP_pathTolerance

Path tolerance for contour transitions after the filter
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.0001 to 10.000 000 000 [mm]
Default: 0.01 [mm]
Access: LEVEL3
Reaction: RUN

MP_pathToleranceHi

Path tolerance after the filter at rapid traverse
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.0001 to 10.000 000 000 [mm]
Default: 0.01 [mm]
Access: LEVEL3
Reaction: RUN

All "-Hi" parameters (jerk and tolerance) go into effect from the feed rate defined in **MP_maxG1Feed**.

Please note:

- **Programmed feed rate <= MP_maxG1Feed:**
  Tool in contact with workpiece, lower jerk values and tolerances are effective
- **Programmed feed rate > MP_maxG1Feed:**
  Rapid traverse, greater jerk values and tolerances are effective

Note

HEIDENHAIN recommends:
Set **MP_maxG1Feed** to a high value when optimizing the machine. This prevents inadvertent use of the "-Hi" values on the MANUALplus 620 while you are adjusting the axes.

MP_maxG1Feed

Maximum machining feed rate. If this value is exceeded, the "-Hi" parameters go into effect.
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 99 999 [mm/min]
Default: 99 999 [mm/min]
Access: LEVEL3
Reaction: RUN

The jerk defined in **MP_maxPathJerk** is effective throughout the channel for machining feed rates that do not exceed the maximum machining feed rate from **MP_maxG1Feed. MP_maxPathJerk** has a higher priority than **MP_axPathJerk** (CfgLaAxis). This means that **MP_maxPathJerk** limits the vector sum of the axis-specific path jerks.

MP_maxPathJerk

Maximum jerk on the path
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 to 1 000 000.000 [m/s3]
Default:	40 [m/s3]
Access:	LEVEL3
Reaction:	RUN

The jerk defined in **MP_maxPathJerkHi** is effective for feed rates greater than the feed rate defined in **MP_maxG1Feed**. **MP_maxPathJerkHi** has a higher priority than **MP_axPathJerkHi** (CfgLaAxis). This means that **MP_maxPathJerkHi** limits the vector sum of the axis-specific path jerks at rapid traverse.

MP_maxPathJerkHi

Maximum jerk on the path at rapid traverse
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 to 1 000 000.000 [m/s3]
Default:	40 [m/s3]
Access:	LEVEL3
Reaction:	RUN

- Independent of the feed rate, define the maximum yank (dj/dt) in **MP_maxPathYank.**

The maximum yank normally does not have to be limited in the path direction with **MP_maxPathYank** (= change of the jerk per time). Therefore, you can usually set the parameter to its maximum value.

Under unfavorable conditions, very short jerk phases (in the range of the first natural frequency of the axis) may lead to oscillations in the workpiece. By reducing **MP_maxPathYank**, you can prolong the jerk phases to a certain extent and thus reduce the oscillations in the workpiece.

Note

Limiting the yank is not suitable for smoothing successive acceleration and deceleration processes ("pumping" of feed rate). In this case, reduce the path jerk (**MP_axTransJerk**) instead. The use of a stretch filter (CfgStretchFilter) and feed-rate smoothing filter (**MP_filterFeedTime**) might also be helpful.

Feed-rate "pumping" often occurs in poor-quality CAM-generated programs that carry noise. Therefore check whether the CAM program is the cause of the problem and read it out again if necessary.

MP_maxPathYank

Maximum yank on the path (dj/dt)
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 000 000 to 1 000 000 [mm/s4]
Default:	4 000 [mm/s4]
Access:	LEVEL3
Reaction:	RUN

Axis-specific limit values

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgFeedLimits	
minFeed	400301
maxFeed	400302
rapidFeed	400303
manualFeed	400304
maxAcceleration	400305
maxAccSpeedCtrl	400311
maxDecSpeedCtrl	400306
CfgLaAxis	
axTransJerk	401701
axPathJerk	401703
axPathJerkHi	401704

The parameter objects **CfgFeedLimits** and **CfgLaAxis** are not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)

▶ In **CfgFeedLimits** and **CfgLaAxis**, define the axis-specific limit values for feed rate, acceleration and jerk.

MP_minFeed

Applies only to the main spindle: minimum spindle speed
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 36 000 000 [°/min]

Default: 0
Access: LEVEL3
Reaction: RUN

MP_maxFeed

Maximum axis feed rate (rapid traverse) or maximum spindle speed
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 36 000 000 [mm/min] or [°/min]
Default: 16 000
Access: LEVEL3
Reaction: RUN

MP_rapidFeed is used as the maximum axis feed rate in the **Manual Operation** and **El. Handwheel** modes in conjunction with the rapid traverse key.

MP_rapidFeed

	Rapid traverse in manual mode Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.0 to 36 000 000.0 [mm/min] or [°/min]
Default:	4 999.98
Access:	LEVEL3
Reaction:	RUN

In the **El. Handwheel** mode, the value entered in **MP_manualFeed** is multiplied by the value entered in **MP_CfgHandwheel/feedFactor.** For rotary axes and spindles, the velocity is specified in [°/min]:

MP_manualFeed

	Maximum manual feed rate Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.0 to 36 000 000.0 [mm/min] or [°/min]
Default:	4 999.98
Access:	LEVEL3
Reaction:	RUN

MP_maxAcceleration defines the axis-specific acceleration. Use this parameter for all position-looped axes. The parameter is also effective for spindles operated with position control, such as while tapping or with M19. The value entered also applies to braking the axis. For rotary axes and spindles, the acceleration is specified in the unit [1000°/s^2].

MP_maxAcceleration

	Max. permissible axis acceleration during position control Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 1000 [m/s^2] or [1000°/s^2]
Default:	3 [m/s^2] or [1000°/s^2]
Access:	LEVEL3
Reaction:	RUN

The two optional parameters **MP_maxAccSpeedCtrl** and **MP_maxDecSpeedCtrl** are only of interest if you want to use values for the acceleration and braking ramp of the spindle (M3, M4, M5) that differ from those for the axis acceleration **(MP_maxAcceleration).** If this is the case, then enter the parameters. If the two parameters are not entered, then as a default the value from **MP_maxAcceleration** is used.

MP_maxAccSpeedCtrl

Optional acceleration for the spindle with shaft speed control
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0.000 000 000 to 1000 [1000°/s^2]
This parameter is only effective for spindles. If 0 is entered or there is no parameter, the acceleration from **MP_maxAcceleration** is used.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

MP_maxDecSpeedCtrl

Optional braking ramp for the spindle with shaft speed control
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0.000 000 000 to 1000 [1000°/s^2]
This parameter is only effective for spindles. If the value 0 is entered or the parameter is not available, the value from **MP_maxAccSpeedCtrl** will be used for the brake acceleration. If **MP_maxAccSpeedCtrl** is also 0 or not present, the value from **MP_maxAcceleration** will be used.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

There are two ways to configure the maximum jerk on the path:

- **MP_maxPathJerk(Hi)** (551) limits the jerk in the direction of traverse.
- **MP_axTransJerk** limits the jerk for specific axes in the transverse path direction (transition jerk).

If the emphasis of commissioning is primarily on speed (with a sufficiently clean surface), you can set the transition jerk to a value that is 2 to 3 times higher than the path jerk. Only if a clean surface takes highest priority should you use the same setting for both the path jerk and the transition jerk.

MP_axTransJerk

Maximum axis jerk
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 1 000 000 [m/s^3]
Default: 0.1 [m/s^3]
Access: LEVEL3
Reaction: RUN

Two axis-specific jerk parameters are available for very high requirements on the surface quality. With the parameters **MP_axPathJerk** and **MP_axPathJerkHi** (rapid traverse) you limit the axis-specific jerk on path contours.

Independent of the values set in the two parameters, the values defined in **MP_maxPathJerk** and **MP_maxPathJerkHi** (551) are always effective as the maximum jerk limits in the path direction. To allow maximum flexibility in calculating the jerk, taking all the axis components in the traverse movements into account, you should set **MP_maxPathJerk** and **MP_maxPathJerkHi** to values that are at least 1.75 times higher than those defined in **MP_axPathJerk** and **MP_axPathJerkHi**. This ensures that only the axis-specific jerk values go into effect.

MP_axPathJerk and **MP_axPathJerkHi** have a great influence on the speed stability, surface quality and machining time.

The MANUALplus 620 ensures that the nominal values do not exceed the permissible axis jerk.

Adjusting **MP_axPathJerk** and **MP_axPathJerkHi**:

- The adjustment should be performed with a suitable NC program. The NC programs "ff_100_[axis].nc" are already available on the control for feedforward adjustment. Record the following error **s-diff** and the jerk **j-nom** caused by a change in the feed rate ("ramp"), e.g. from 100 to 1000 mm/min.

Note

Recording the jerk is very useful because it allows you to check whether modifying **MP_axPathJerk** has actually changed the jerk on the axis. If the change in feed rate is too great, it may happen that the maximum acceleration rather than the jerk limits the ramp.

- Modify the setting in **MP_axPathJerk** until the resulting following error is just acceptable enough.
- The value set in **MP_axPathJerkHi** is typically five times higher than that in **MP_axPathJerk**.

Note

The following error **s-diff** should be no more than a few micrometers (µm) to ensure smooth surfaces during acceleration and deceleration. If speed is required, the parameter can be increased until the machine gets loud, or the following error **s-diff** too large.

Note

In the above procedure, which uses feed rate changes instead of only positioning movements, disturbances from sliding friction and stiction transitions are avoided.

▶ The following formula can be used to calculate a proposed value for **MP_axPathJerk**. The formula is conceived for the calculated jerk to be large enough that the acceleration of the axis is not impaired.

$$MP\_axPathJerk \geq \frac{MP\_maxAcceleration^2 \cdot 60000}{MP\_rapidFeed}$$

Note

HEIDENHAIN recommends entering the permissible jerk for each axis in **MP_axPathJerk and MP_axPathJerkHi**. This way the jerk is based on the weakest axis participating in a motion.

MP_axPathJerk

Axis-specific maximum jerk on path contours
Available from NCK software version: 597 110-04.

Format: Numerical value
Input: 0 to 1 000 000 [m/s^3]
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

MP_axPathJerkHi

Axis-specific maximum jerk during rapid traverse

Format: Numerical value
Input: 0 to 1 000 000 [m/s^3]
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

Tolerance for rotary axes

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgLaAxis **axFilterErrWeight**	 401702

The parameter object CfgLaAxis is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)

The filter error for a rotary axis with a large radius can be multiplied by a factor. Weighting the factor smoothes the feed-rate profile for the rotary axis.

By enlarging the value of the rotary axes, the tolerance of these axes has a more stringent effect and the error of the tool center point becomes smaller.

Reducing the value is also useful if the filter error of an axis has only a minor influence on the workpiece.

Enter **MP_axFilterErrWeight** = 1 for linear axes.

MP_axFilterErrWeight is evaluated for smoothing the contour.

MP_axFilterErrWeight

	Factor for filter error (for rotary axes) Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.010 000 000 to 100 For linear axes: 1
Default:	1
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
System CfgCycleTimes **ipoCycle**	 100301

The interpolator operates at the clock rate defined in **MP_ipoCycle.** The axis-specific nominal position values are calculated at this clock rate from the feed rate profiles transferred by look-ahead.

MP_ipoCycle

Cycle time of position controller (interpolation clock pulse)

Format: Selection menu

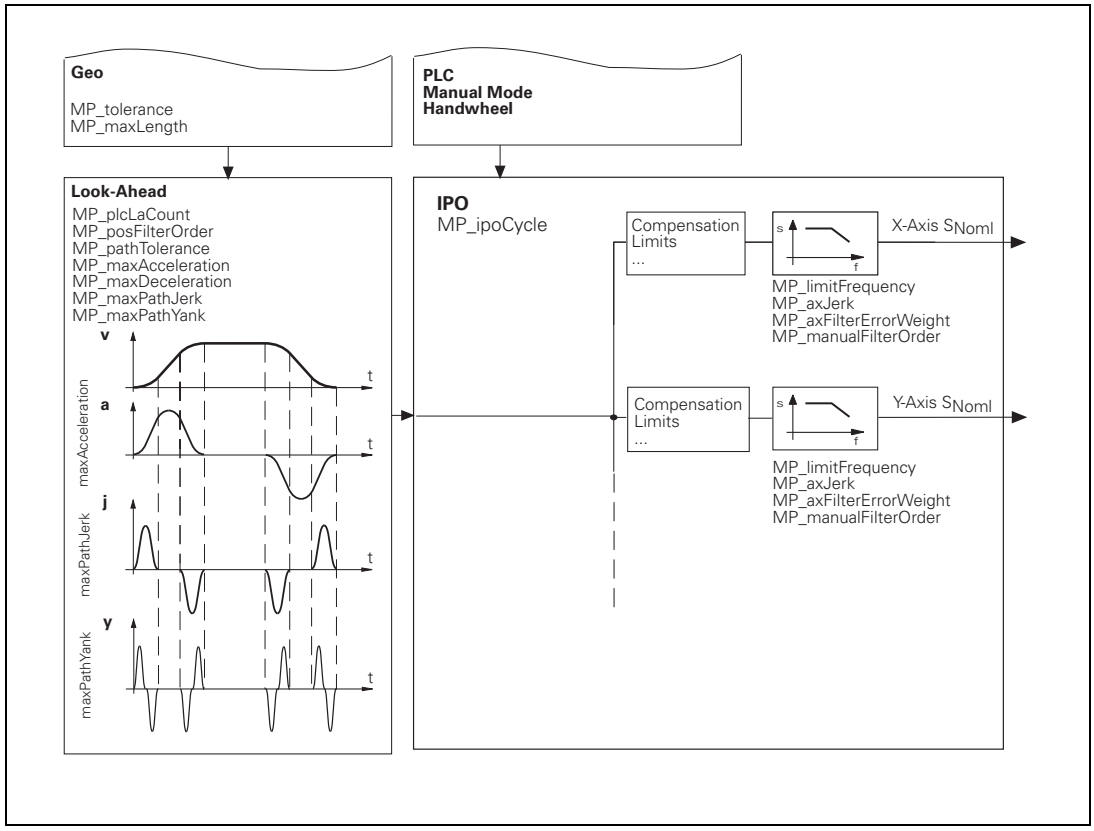
Selection: 3 ms
At this time only a position controller cycle time of 3 ms is supported. Therefore no other values can be selected.

Default: 3 ms

Access: LEVEL3

Reaction: RESET

Schematic of the Interpolator:



Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgPosControl	
kvFactor	400801
feedForwardFactor	400806
controlOutputLimit	400807

The parameter object **CfgPosControl** is not required for:

- Virtual axes (**MP_axisMode**=Virtual)
- Axes that are for display only (**MP_axisMode**= Display)

The position controller uses the axis-specific nominal position values transferred by the interpolator. The nominal speed values are determined and transferred to the speed controller.

Feedback control

The MANUALplus 620 operates with following error (servo lag) or with velocity feedforward. It is defined in **MP_feedForwardFactor.**

- If **MP_feedForwardFactor** = 0, operation with 100% following error is in effect.
- **MP_feedForwardFactor** = >0 activates velocity semifeedforward control (for analog axes).
- If **MP_feedForwardFactor** = 1, machining will be carried out using 100% velocity feedforward control (for digital axes).

With analog control, the shaft speed nominal value is calculated with the following formula:

$$U_{out} = (P_{err} \cdot \boldsymbol{kvFactor} + \frac{V_{nom}}{60} \cdot \boldsymbol{feedForwardFactor} + 1000 \cdot A_{nom} \cdot \boldsymbol{accForwardFactor}) \cdot \frac{9\,V \cdot 60}{\boldsymbol{maxFeedAt9V}}$$

Simplified:

U_{out} = **MP_kvFactor** · P_{err} + V_{nom} · **MP_feedForwardFactor**

Value, parameter	Unit	Description
U_{out}	V	Output voltage (analog nominal speed value)
P_{err}	mm	Following error (servo lag)
kvFactor	1/s	Kv factor (proportional component of position controller)
V_{nom}	mm/min	Nominal velocity
feedForwardFactor		Factor for velocity feedforward control
A_{nom}	m/s^2	Nominal acceleration
accForwardFactor	(s)	Factor for acceleration feedforward control
maxFeedAt9V	mm/min	Assumed velocity of the axis at 9 V

MP_feedForwardFactor

	Factor for velocity feedforward control Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	**0**: Feedback control with following error **>0,<1**: Feedback control with velocity semifeedforward **1**: Feedback control with velocity feedforward
Default:	1
Access:	LEVEL3
Reaction:	RUN

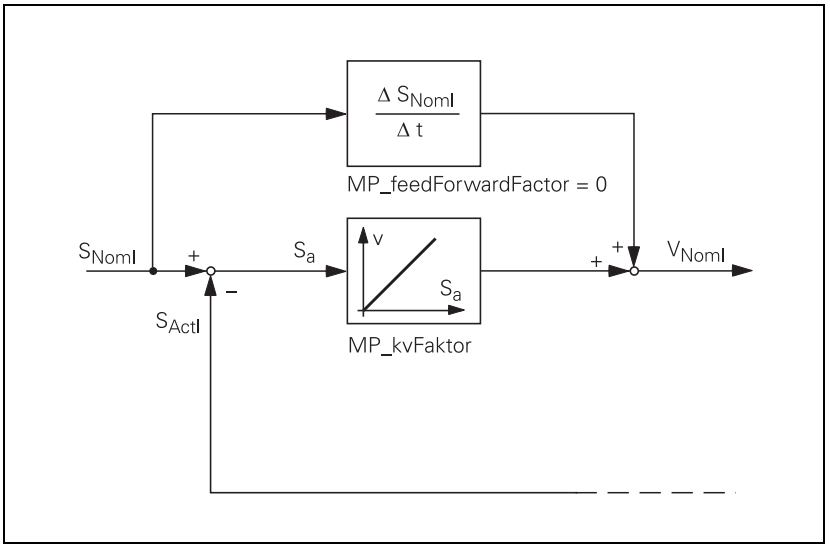
Note

For axes that are interpolated with each other, the k_V factor and the factor for velocity feedforward control must be equal. In this case the smaller k_V factor determines the input value for all axes.

Feedback control with following error

Following error (also known as servo lag) is a gap that remains between the nominal position commanded by the NC and the actual position of the axis.

Simplified representation:



The nominal position value s_{nomI} for a given axis is compared with the actual position value s_{actI} and the resulting difference is the following error s_a:

$s_a = s_{nomI} - s_{actI}$

s_a = following error
s_{nomI} = nominal position value
s_{actI} = actual position value

The following error is multiplied by the k_v factor and passed on as nominal velocity value:

$v = k_v \cdot s_a$

v_{nomI} = nominal velocity value

The control loop gain, known as the k_v factor, defines the amplification of the position control loop. You must find the optimum k_v factor by trial and error.

If you choose a k_v factor that is too large, the following error will become very small. However, this can lead to oscillations.

If you choose too small a k_v factor, the axis will move to a new position too slowly.

For axes that are interpolated with each other, the k_v factors must be equal to prevent contour deviations.

▶ Define the k_v factor in **MP_kvFactor.**

Interrelation of k_v factor, feed rate, and following error

The following formula shows the interrelation of k_v factor, feed rate, and following error:

$$k_v = \frac{v_e}{s_a} \quad \text{or} \quad s_a = \frac{v_e}{k_v}$$

k_v = kv factor [(mm/s)/mm]
v_e = rapid traverse [mm/sec]
s_a = following error [mm]

Note

The unit for the kv factor of the MANUALplus 620 differs from the one used for the other TNC contouring controls, such as the iTNC 530.
Unit for the kv factor of the MANUALplus 620: mm / (mm · s)
Unit for the kv factor of the iTNC 530: m / (mm · min)
Therefore:
iTNC 530 kv factor · 1000 / 60 = MANUALplus 620 kv factor

MP_kvFactor

kv factor (proportional component of the position controller)
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 000 000 to 1000 [1/s]
Default:	0
Access:	LEVEL3
Reaction:	RUN

Controller output limit

The controller output limit **MP_controlOutputLimit** is used only during switch-on of position control without actual-to-nominal value transfer. Example:

Clamped or hanging axes cause a following error when the position control loop is open. When closing the control loop without actual-to-nominal value transfer, this difference in the position is corrected by the MANUALplus 620. The deviation is corrected at the maximum feed rate entered in **MP_controlOutputLimit.**

Note

The axis parameters entered for jerk and acceleration have no effect. Enter only values that are non-critical to the axis.
HEIDENHEIN recommends:
In **MP_controlOutputLimit,** enter a value that is approximately 0.1 · **MP_manualFeed**.

If **MP_controlOutputLimit** = 0, the resulting following error is not corrected until the next positioning block.

MP_controlOutputLimit

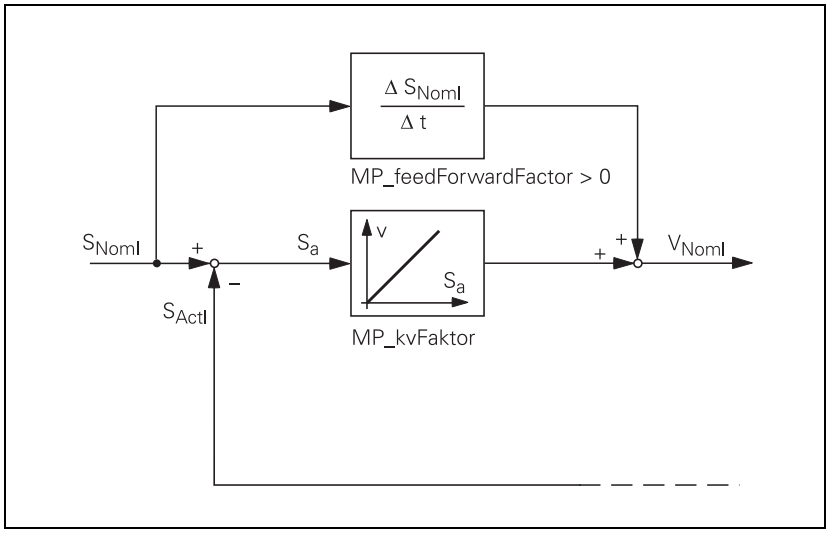
	Controller output limit for the position controller Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 1666 [mm/min]
Default:	0 [mm/min]
Access:	LEVEL3
Reaction:	RUN

Feedback control with velocity feedforward

For feedback control with velocity feedforward, the nominal velocity value consists of an open-loop and a closed-loop component.

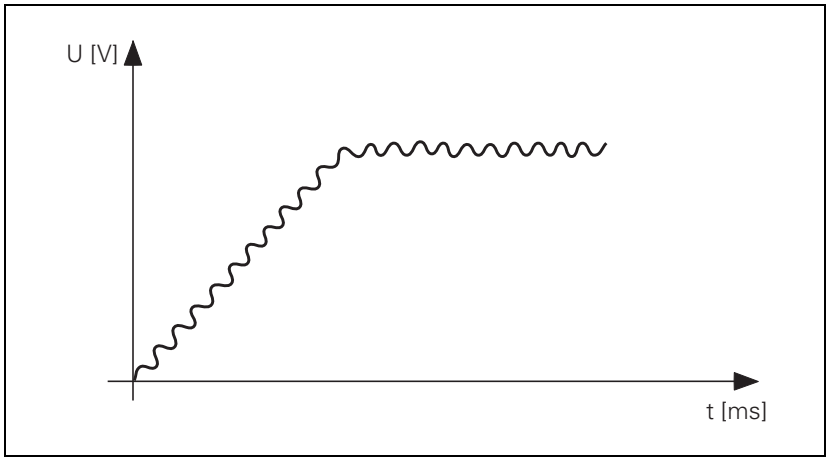
The machine-adjusted nominal velocity value is the open-loop controlled component. The closed-loop velocity component is calculated through the following error. The following error is small.

In most cases, machines are controlled with velocity feedforward, since it makes it possible to machine exact contours even at high speeds.



You can influence feedback control with velocity feedforward with the k_v factor:

- Enter a k_v factor in **MP_kvFactor**.



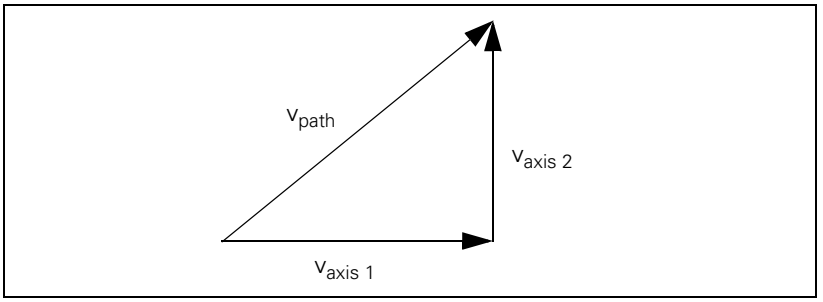
Attention

If the k_v factor that you select is too large, the system will oscillate around the forward-fed nominal velocity value.

Unlike operation with following error, you must enter the optimum k_v factor for each axis when operating with interpolated axes (See "Interrelation of k_v factor, feed rate, and following error" on page 563).

Rapid traverse and feed rate limitation

If more than one axis is moved simultaneously, the rapid traverse on the path v_{path} is formed from the appropriate axis components (see "Axis-specific limit values" on page 553).



- In **MP_maxFeed,** define the maximum rapid traverse for this axis.

Feed rate and rapid traverse are significantly lower for **`Manual Operation:`**

- Define the feed rate for manual mode in **MP_manualFeed.**
- Define the feed rate for rapid traverse in **MP_rapidFeed.**

If the value in **PP_ChnContourFeedMax** is greater than the value in **MP_maxFeed,** the parameter value applies. After the MANUALplus 620 is switched on, or after an interruption of the PLC program, **PP_ChnContourFeedMax** is assigned the value 300 000 so that the value in **MP_maxFeed** automatically becomes effective.

Note

The absolute maximum velocity of this axis is defined in **MP_maxFeed.** This value is not exceeded.

The maximum possible feed rate depends on the encoder being used.

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

v_{max} = Maximum traversing speed
P = Signal period of the encoder
f_i = Input frequency of the encoder input, see "Encoder connections" on page 174.

Digital axes:
For digital axes, the maximum feed rate also depends on the number of pole pairs in the drive motor and the pitch of the ball screw.

$$v_{max}[\text{mm/min}] = \frac{24\,000}{\text{numberofpolepairs}}[1/\text{min}] \times \text{pitch of the ballscrew [mm]}$$

Analog axes:
The rapid traverse rate at an analog voltage of 9 V is defined in **MP_maxFeedAt9V**.

Feed rate values in PLC operands

The feed rate values are stored in PLC operands (see the following tables).

The PLC can influence the following values:

- **PP_ChnContourFeedMax:** Maximum feed rate
- **PP_AxManualFeedMax:** Maximum manual axis feed rate
  Only effective for specific axes if the optional machine parameter **MP_axisFeedDisplay** (CfgDisplayData) is set to the value **`at axis key.`** If the parameter is not set, the feed-rate value is effective globally for all axes.

The following PLC operands contain channel-specific feed rate values.

PLC operand / Description	Type
NN_ChnProgFeedMinute Programmed feed rate per minute [mm/min]	D
NN_ChnProgFeedRevolution Programmed feed rate per revolution [mm/rev]	D
NN_ChnProgFeedThread Programmed feed rate per thread [mm/rev]	D
NN_ChnFeedMinuteActive Feed per minute is active. 0: Per-minute feed rate is not active. 1. Feed per minute is active.	M
NN_ChnFeedRevolutionActive Feed rate per revolution active 0: Per-revolution feed rate is not active 1: Per-revolution feed rate is active	M
NN_ChnFeedThreadActive Feed rate per thread active 0: Thread feed rate is not active 1: Thread feed rate is active	M
NN_ChnFeedRapidTraverseActive Rapid traverse active (FMAX) 0: Rapid traverse is not active 1: Rapid traverse active	M
NN_ChnContourFeed Current contouring feed rate [mm/min] In the manual operating modes, the highest axis feed of all axes is stored in this operand	D
PP_ChnContourFeedMax Max. feed rate from the PLC [mm/min]	D

The following PLC operands contain axis-specific feed rate values.

PLC operand / Description	Type
PP_AxManualFeedMax Maximum manual axis feed rate [mm/rev] (Only effective for specific axes if the machine parameter **MP_axisFeedDisplay** (CfgDisplayData) is set to the value **`at axis key.)`** If the parameter is not set, the feed-rate value is effective globally for all axes.	D
PP_AxTraversePos Manual traverse in positive direction 0: Do not move axis 1: Move axis	M
PP_AxTraverseNeg Manual traverse in negative direction 0: Do not move axis 1: Move axis	M

Position loop resolution for digital axes

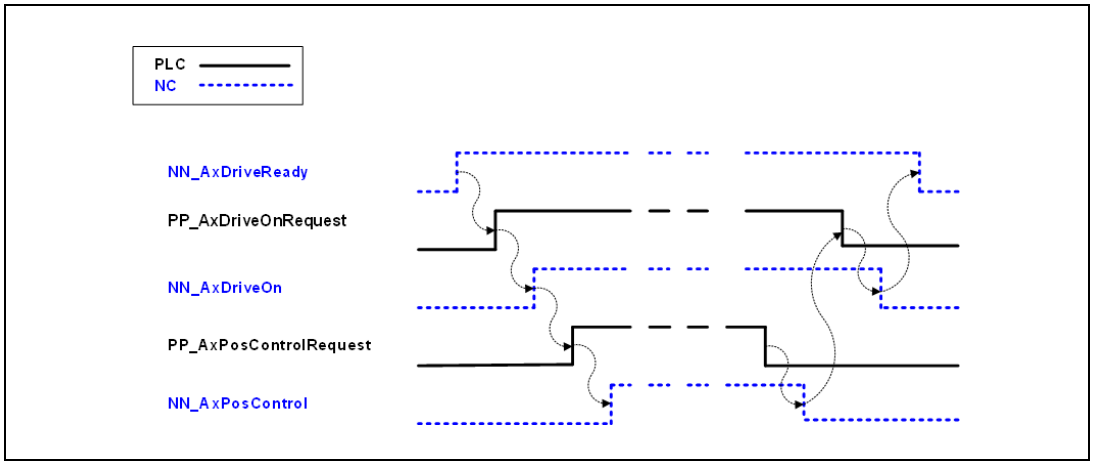
The encoder signals are interpolated 1024-fold.

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

6.13.7 Activating and deactivating position control loops

Opening the position control loop

The following figure shows the procedure for switching on the drive motor and activating the position control loop as well as the procedure for opening the position control loop and deactivating the drive.



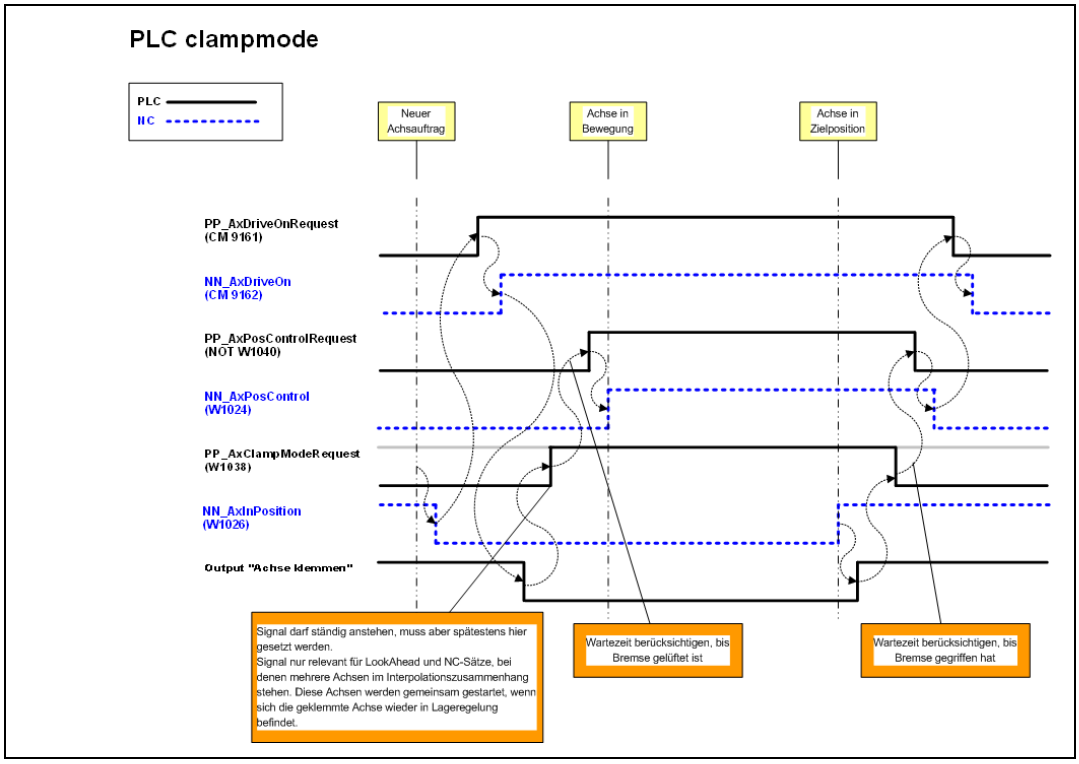
PLC operand / Description	Type
NN_AxDriveReady Axis drive is ready 0: Drive not ready for operation 1: Drive ready for operation	M
PP_AxDriveOnRequest Switch axis drive on 0: Do not activate the drive 1: Activate the drive	M
NN_AxDriveOn Axis drive is switched on (and is at least speed-controlled) 0: Drive is off 1: Drive is on	M
PP_AxPosControlRequest Position-control the axis 0: No position feedback control for axis 1: Position feedback control for axis	M
NN_AxPosControl Axis in position feedback control 0: Axis not in position feedback control 1: Axis in position feedback control	M

Clamping the axes

After running an NC block you can clamp the axes.

The MANUALplus 620 considers an axis as clamped as soon as position control is switched off, i.e. as soon as **NN_AxPosControl** decreases towards 0.
The PLC uses the signal **NN_AxInPosition** to determine whether an axis movement is pending.

The following figure shows the procedure for clamping the axes as well as for unclamping them.



If **NN_AxInPosition** is set, no further movement of the respective axis is pending. The PLC must initiate a clamping operation if **PP_AxClampModeRequest** is set. As illustrated in the figure, first the axis is clamped, then position control is switched off, and finally the drive is switched off. The MANUALplus 620 resumes NC program run as soon as position control is switched off.

If **NN_AxInPosition** decreases, the axis should be moved. The PLC initiates the unclamping operation for the respective axis. As illustrated in the figure, first the drive is switched on, then the clamping is released, and then position control is activated. The MANUALplus 620 resumes NC program run as soon as position control is switched on.

Activating the clamping mode for NC channels

Clamping mode can be switched on/off for an NC channel if no machining path is active in the NC channel. This constraint is important for the surface quality of the workpiece, because activation or deactivation of a brake influences all axes of a channel.
Feed-rate enabling should therefore be withdrawn for all axes of the NC channel while the clamping mode is being switched.
As an alternative, switchover can also take place while a strobe is being executed, because—in addition to normal clamping mode—axes can also be clamped/unclamped by means of M commands.

Activating the clamping mode for individual axes

When switching an axis between clamped and unclamped mode, the following constraints apply:

- The **PP_AxClampModeRequest** signal may be set only if the axis is not in a closed loop, i.e. the clamping mode is activated.
- The **PP_AxClampModeRequest** signal may be reset only if the axis is in a closed loop, i.e. the clamping mode is deactivated.
- If a strobe (M command) is used for switchover, the strobe must not be acknowledged until the **PP_AxClampModeRequest** signal has assumed the correct state.

Clamping mode with active handwheel

If a handwheel is active for an axis in clamping mode, the axis may remain switched on even if no further programmed movement of this axis is pending, i.e. **NN_AxInPosition** is set.

For axes in clamping mode, the handwheel should not be activated during a movement.

PLC operand / Description	Type
PP_AxClampModeRequest Axis in clamping mode 0: No clamping mode Axis is to remain continuously in a closed loop 1: Clamping mode Axis is to be clamped as soon as it is no longer moved (NN_AxInPosition=1) Axis is to be unclamped when it is to be moved (NN_AxInPosition=0)	M
NN_AxInPosition Axis in position 0: Axis not in position 1: Axis in position	M

Actual-to-nominal value transfer

During actual-to-nominal value transfer, the current position is saved as the nominal position value. This becomes necessary, for example, if the axis has been moved when the position control loop is open.

There are two ways to turn the actual position into the nominal position:

- Place the request for actual-to-nominal value transfer in the **`Manual`** and **`Electronic Handwheel`** operating modes in **PP_AxValueActToNominal** and check the elimination of the following error in **NN_AxCorrectingLagError.**
- To transfer the actual position in all operating modes, use Module 9145.

PLC operand / Description	Type
PP_AxValueActToNominal Actual-to-nominal value transfer (Request to eliminate following error) 0: Request to eliminate following error 1: No request to eliminate following error	M
NN_AxCorrectingLagError Following error eliminated 0: Following error is not eliminated 1: Following error is eliminated	M

Module 9145 Actual-to-nominal value transfer

Module 9145 is used for an actual-to-nominal value transfer for the axes entered.

Constraints:

- The module functions only in the cyclic PLC program.
- An actual-to-nominal value transfer is possible only if the MANUALplus 620 is not active (NN_ChnControlInOperation=0) or if an M/S/T/T2/G strobe is pending. Actual-to-nominal value transfer can always be performed for axes that are not in an interpolation context.

Call:

PS B/W/D/K <Axes bit-encoded>
(Bit 0 represents logic axis 0, etc.)

CM 9145

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Actual-to-nominal value transfer performed
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid axis number
	21	Missing M/S/T/T2/G strobe in NN_ChnControlInOperation=1
	24	Module was called in a spawn or submit job

6.13.8 Feed-rate enable

To move the axes, you must first enable the feed rate through the PLC. Until "feed-rate enable" is set, the nominal velocity value zero is output. The status display shows that the feed rate enable is set / not set.

You can set the feed rate enable for all axes of the NC channel or for specific axes. The PLC run-time system combines **PP_ChnFeedEnable** and the corresponding axis-dependent feed rate enable **PP_AxFeedEnable** with an OR gate.

Feed-rate enable for all axes of an NC channel:

- Set **PP_ChnFeedEnable**

Axis-specific feed-rate enable:

- Reset **PP_ChnFeedEnable**
- Set **PP_AxFeedEnable**

PLC operand / Description	Type
PP_ChnFeedEnable Feed-rate enable for all axes 0: No feed-rate enable 1: Feed-rate enable	M
PP_AxFeedEnable Axis-specific feed rate enable 0: No feed-rate enable 1: Feed-rate enable	M

The PLC sets **PP_ChnWorkFeedEnable** if rapid traverse movements are allowed. This marker is set, for example during a tool change or turret actuation, in order to use the time for rapid-traverse movements.

PLC operand / Description	Type
PP_ChnWorkFeedEnable Enable rapid traverse for all axes 0: No rapid traverse enable 1: G0 movements are allowed (rapid traverse enable)	M

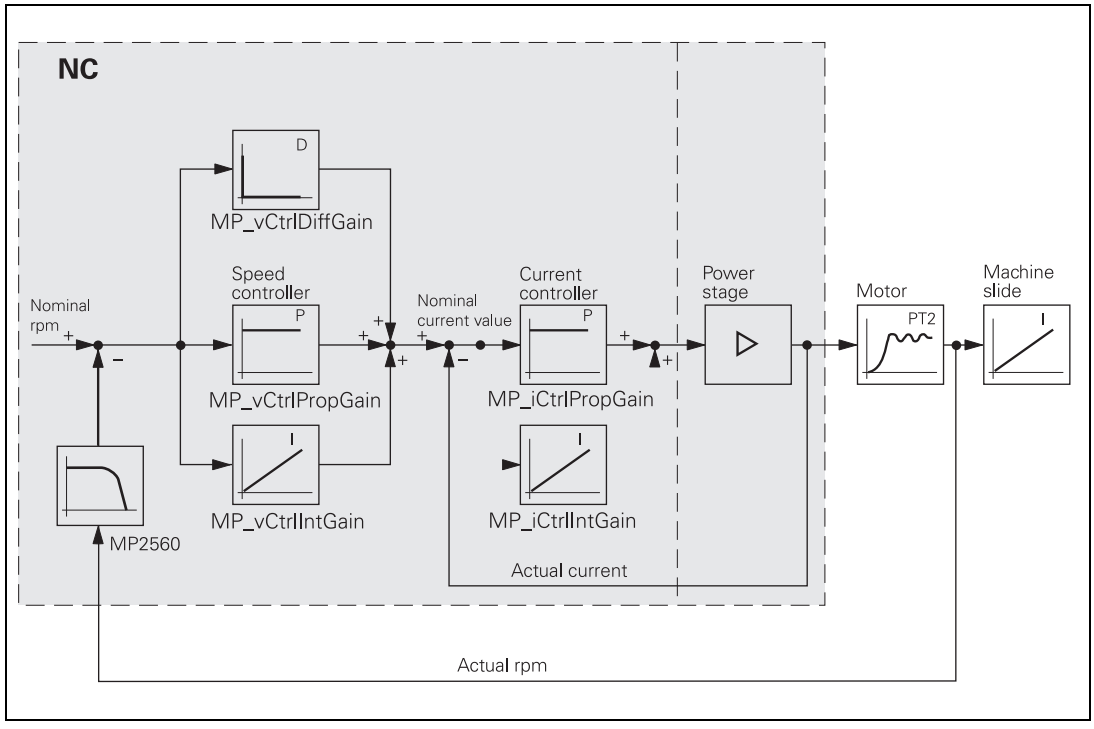
Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgSpeedControl	
vCtrlPropGain	400901
vCtrlIntGain	400902
vCtrlDiffGain	400904

The parameter object CfgSpeedControl is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)
- Analog axes (MP_axisHw=Analog)

The actual speed values are measured directly at the motors with HEIDENHAIN rotary encoders. The position controller provides the nominal speed value. The speed controller is driven by the difference between nominal and actual speed values. It provides the nominal current value as output.

Use Module 9164 to read the actual speed value of the motors.



Adjust the step response of the speed controller:

- With the position controller switched off **(PP_AxPosControlRequest),** enter with **MP_VCtrlPropGain** a proportional factor and with **MP_VCtrlIntGain** an integral factor for the speed controller. Adjust the step response so that only one overshoot is visible and the settling time t_{off} is as small as possible.
  Realistic values for the settling time: 3 ms to 15 ms.

MP_vCtrlPropGain

	Proportional factor of the speed controller Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 1000 [As/rev.]
Default:	1 [As/rev.]
Access:	LEVEL3
Reaction:	RUN

MP_vCtrlIntGain

	Integral factor of the speed controller Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 100 000 [A/rev.]
Default:	100 [A/rev.]
Access:	LEVEL3
Reaction:	RUN

Module 9164 Read the actual speed value of the drive motor
The module provides the actual speed value of a motor controlled by an integral current controller in 1/1000 revolutions per minute.

The resolution of the actual speed value depends on the encoder being used:

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

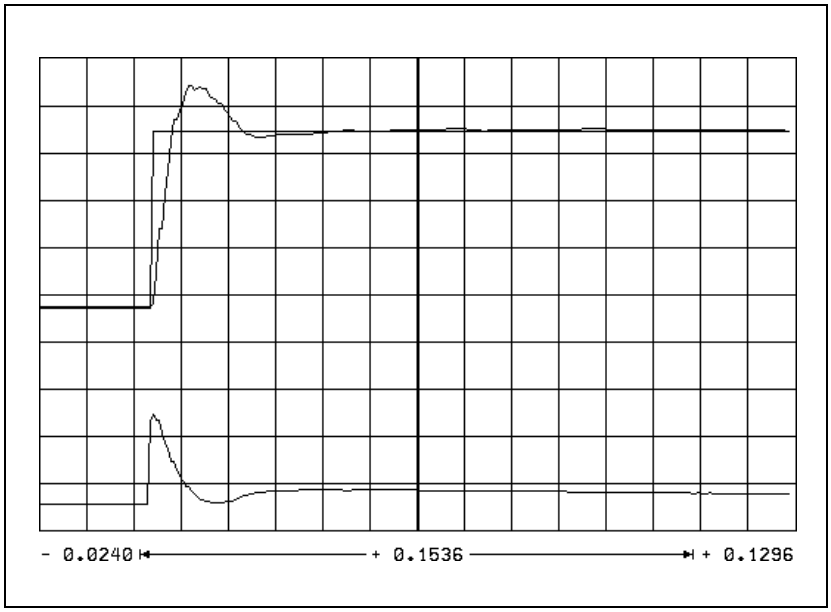
The value 0 is read for axes that are not connected.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
CM	9164	
PL	B/W/D	<actual speed value in the format 0.001 rpm>

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Actual speed value was read
	1	Control has no integrated current controller



The step response illustrated above is idealized. In practice, interfering oscillations are superimposed on the step response.

You can reduce these interference oscillations with the differential factor, thePT_2 second-order time-delay element, the band rejection filter and the low-pass filter of the speed controller.

Differential factor

The differential factor reduces low-frequency oscillations. However, it increases the tendency to oscillate in the high frequency range.

▶ In **MP_vCtrlDiffGain,** enter a differential factor.

Note

Ensure that the system is stable enough!

The differential factor is not recommended on machines with motors that have belt couplings. The influence of aging and temperature is too great.

Estimating the differential factor:

$$MP\_vCtrlDiffGain \approx \frac{T \cdot MP\_vCtrlPropGain}{8}$$

MP_vCtrlDiffGain: Differential factor of the speed controller [As^2]

MP_vCtrlPropGain: Proportional factor of the speed controller

T: Period duration of the lowest disturbance frequency [s]

MP_vCtrlDiffGain

	Differential factor of the speed controller Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 1 [As^2/rev.]
Default:	0 [As^2/rev.]
Access:	LEVEL3
Reaction:	RUN

Note

The filters in the speed control loop of the CC 422 and CC 61xx/CC 424 differ in their organization. This chapter describes the filter functions available when using the CC 61xx and CC 424.

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgSpeedControl	
vCtrlFiltLowPassT	400905
vCtrlEncInputFilt	400906
vCtrlFiltLowPassT	400905
vCtrlFiltDamping1	400910
vCtrlFiltDamping2	400914
vCtrlFiltDamping3	400918
vCtrlFiltDamping4	400922
vCtrlFiltDamping5	400926
vCtrlFiltFreq1	400908
vCtrlFiltFreq2	400912
vCtrlFiltFreq3	400916
vCtrlFiltFreq4	400920
vCtrlFiltFreq5	400924
vCtrlFiltType1	400907
vCtrlFiltType2	400911
vCtrlFiltType3	400915
vCtrlFiltType4	400919
vCtrlFiltType5	400923
vCtrlFiltBandWidth1	400909
vCtrlFiltBandWidth2	400913
vCtrlFiltBandWidth3	400917
vCtrlFiltBandWidth4	400921
vCtrlFiltBandWidth5	400925

The parameter objects CfgSpeedControl and CfgControllerComp are not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)
- Analog axes (MP_axisHw=Analog)

Multifunction filter

With the CC 61xx and CC 424, you can influence the manipulated variable of the speed controller (= nominal current) and the position controller (= nominal speed) by means of up to five freely definable filters per axis. These filters are multifunctional filters, which means that the filter type of each individual filter order can be selected as desired. They are also effective for the spindle(s).

Objective of the filters

The first objective when adjusting a machine is the optimization of the control loop in the current and speed controller. The increase of the P component of the control loops in order to raise the dynamics of the machine is the main aspect of this. If a control loop is at the oscillation limit, these oscillations can be damped with filter functions, so that the P components can be increased again.

The second objective when adjusting a machine is the optimization of the position controller. Here it is attempted to increase the k_V factor in the position controller, in order to simultaneously increase the machine's performance (the acceleration behavior, for example). The procedure is always to increase the k_V factor to the oscillation limit, damp these oscillations with the filters, and then increase the k_V factor again.

Types of filters

Three different types of filters per axis are available for selection:

- **PT2 low-pass**
  - Use:
    - Oscillations in the upper frequency range (typically: from 500 Hz)
    - High-frequency noises on axes (such as during switch-on)
- **Band-rejection filter**
  - Use:
    - Oscillations in the middle frequency range (typically: between 100 Hz and 2.5 Hz)
  - Typical settings:
    - Damping from 6 to 9 dB
    - Bandwidth: equal to the center frequency, constant from 500 Hz
  - Disadvantage:
    - These can strengthen oscillations in the lower frequency range
- **Phase increase**
  - Use:
    - Oscillations in the lower to middle frequency range, which occur because of an insufficient phase reserve
    - Oscillations in the lower frequency range, for which band rejection would excessively decrease the amplitude
  - Typical settings:
    - Phase from 20° to 80°
    - Center frequency: Frequencies from 3 to 400 Hz
    - Bandwidth: Equal to the center frequency (oscillation frequency)
  - Disadvantage:
    - The control-loop gain above the center frequency is increased. The increased use of band-rejection filters can become necessary, or the P component might need to be reduced.
  - Note: After the settings have been made, the stability of the control loop must be checked again (P and I component)

Recommended types of filters

Experience has shown that the band rejection of the multifunction filters is to be used for damping oscillations in the **speed controller.**

On the other hand, the tendency of the **position controller** to oscillate should be counteracted with IPC (Integral Phase Compensation). Only if this adjustment does not lead to the desired result can the multifunction filters such as the phase increase (better, since it does not facilitate oscillations at lower frequencies as much) or the band-rejection filter be used.

Since the ambient conditions can be so different, the use of the filters must be checked separately in every case. The TNCopt PC software from HEIDENHAIN should always be used, so that the sequence of the adjustment matches the ideal case. This manual can only present recommended guidelines and procedures.

Possible multifunction filter settings

	Filters 1 to 5
Selection of filter type: **0** = Filter not active **1** = PT2 low-pass filter (speed controller) **2** = Band-rejection filter (speed controller) **3** = Phase increase (speed controller) **11** = PT2 low-pass filter (position controller) **12** = Band-rejection filter (position controller) **13** = Phase increase (position controller)	**MP_vCtrlFiltType1...5**
■ **PT2 low-pass filter:** No effect ■ **Band rejection:** Damping [dB] ■ **Phase increase:** Phase [0 - 90°]	**MP_vCtrlFiltDamping1...5**
■ **PT2 low-pass filter:** Corner frequency [Hz] ■ **Band-rejection filter:** Center frequency [Hz] ■ **Phase increase:** Center frequency [Hz]	**MP_vCtrlFiltFreq1...5**
■ **PT2 low-pass filter:** No effect ■ **Band-rejection filter:** Bandwidth [Hz] ■ **Phase increase:** Bandwidth [Hz]	**MP_vCtrlFiltBandWidth1...5**

The filters can be used in the position controller or speed controller as desired. For example, the second filter can be used although the first filter is not active.

- In **MP_vCtrlFiltType1...5,** define the filter type and specify whether the filter is to take effect in the speed controller or in the position controller. The other parameters to be entered depend on the filter type:
- Enter **MP_vCtrlFiltFreq1...5** to set a **PT2 second-order time-delay element.** Enter a base frequency of 3 dB as parameter value.

▶ The parameters **MP_vCtrlFiltFreq1...5, MP_vCtrlFiltBandWidth1...5** and **MP_vCtrlFiltDamping1...5** are required to set a **band-rejection filter.**

Machine parameters

MP_vCtrlFiltType1...5

Filter type for filters 1 to 5
Format: Numerical value
Input: 0 to 3
0: No filter
1: PT_2 low-pass filter (speed controller)
2: Band-rejection filter (speed controller)
11: PT_2 low-pass filter (position controller)
12: Band-rejection filter (position controller)
Default: 0
Access: LEVEL3
Reaction: RUN

MP_vCtrlFiltFreq1...5

Center frequency of band-rejection filter for filters 1 to 5
Available from NCK software version: 597 110-01.
Format: Numerical value
Input: 0.000 000 000 to 5000 [Hz]
Default: 0 [Hz]
Access: LEVEL3
Reaction: RUN

MP_vCtrlFiltBandWidth1...5

Bandwidth of band-rejection filter for filters 1 to 5
Available from NCK software version: 597 110-01.
Format: Numerical value
Input: 0.000 000 000 to 5000 [Hz]
Default: 0 [dB]
Access: LEVEL3
Reaction: RUN

MP_vCtrlFiltDamping1..5

Damping of band-rejection filter for filters 1 to 5
Available from NCK software version: 597 110-01.
Format: Numerical value
Input: 0.000 000 000 to 40 [dB]
Default: 0 [dB]
Access: LEVEL3
Reaction: RUN

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgSpeedControl **vCtrlEncInputFilt**	 400906

Application

If a low-pass filter is used with the CC 61xx or CC 424, the filter order of the low-pass filter can be set in **MP_vCtrlEncInputFilt.** Enter MP_vCtrlEncInputFilt=0 under standard conditions.

Spindle: High-frequency spindles often cause considerable current noise (shown by **`I nom`** or **`Utilization`** on the oscilloscope). The optimization goal is as little current noise as possible at high spindle speeds. MP_vCtrlEncInputFilt = 10 to 20 should be used as a starting value.

Axes: The low-pass filter should be used if the actual speed has "a lot of noise," for example MP_vCtrlEncInputFilt = 1 or 2. If there is no improvement, set MP_vCtrlEncInputFilt = 0.

For adjustment of the filters, See "Filters in the speed controller and position controller when using the CC 61xx and CC 424" on page 579.

Machine parameters

MP_vCtrlEncInputFilt

Filter order of the FIR filter (low-pass filter)
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to 20
0: No low-pass filter
1: 1st-order low-pass filter
. . .
20: 20th-order low-pass filter
Default: 0
Access: LEVEL3
Reaction: RUN

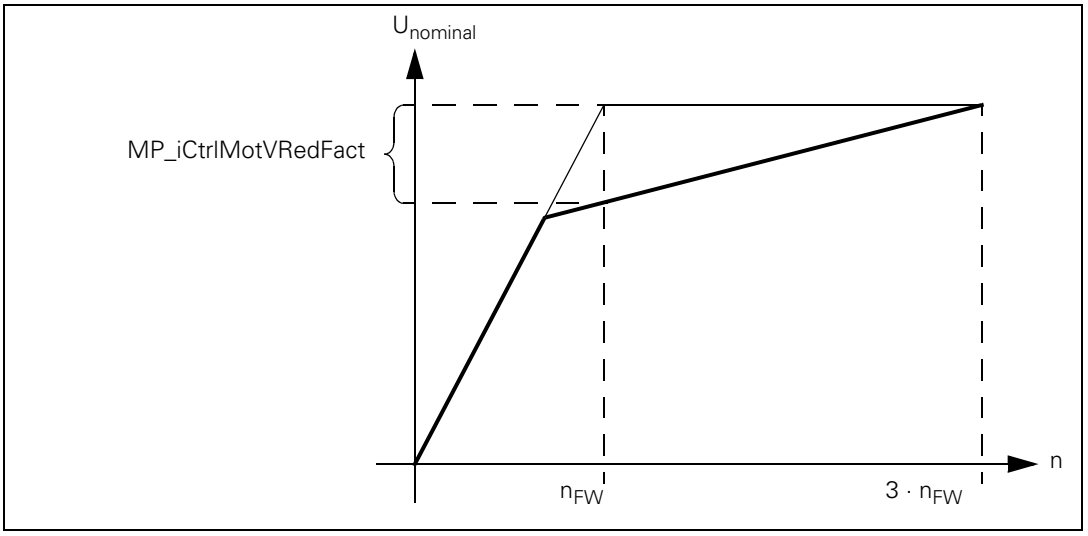
Reduction of the magnetization current

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgCurrentControl **iCtrlMotVRedFact**	401005

Some of the asynchronous spindle motors require a high magnetizing current at low speeds ($n < n_{field\ weakening}$). This magnetizing current can, however, lead to thermal motor problems at the rpm for field weakening $\mathbf{n}_{FW}$ when there is no load.

The nominal voltage (and so also the nominal magnetization current) at the rpm for field weakening $\mathbf{n}_{FW}$ when there is no load can be reduced with **MP_iCtrlMotRedFact.** The entered reduction results in a profile of the nominal voltage as shown in the figure below. The maximum nominal voltage is reached when $n = 3 \cdot n_{field\ weakening}$. The nominal voltage can be reduced by max. 60% (MP_iCtrlMotRedFact = 60).

If a load is placed on the drive, then the nominal voltage is increased again automatically in order to improve the dynamics.



MP_iCtrlMotVRedFact

	Reduction of the nominal voltage at the rpm for field weakening when there is no load Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0 to 60 [%] 0: Function inactive
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RUN

Note

Please note that the reduction of the magnetization current can lead to a limitation of the drive's dynamics. However, this depends on the drive, and must be judged separately in each case.

Setting the reduction of the magnetization current

If thermal problems arise on an asynchronous spindle with no load during weakened-field operation, and these problems cannot be traced to an overload or other mechanical problems, then please reduce the magnetization current as follows:

- Reduce the nominal voltage in steps of 10% in MP_iCtrlMotRedFact
- Reduce MP_iCtrlMotRedFact until a stable temperature within the motor specifications is reached when there is no load.

Note

This function is **not** available for a CC 61xx or UEC 11x!

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgControllerComp	
compActDampFact	401409
compActDampTime	401410

The active damping of low-frequency oscillations is suitable for damping noise oscillations of approx. 10 to 30 Hz. The damping factor is set in **MP_compActDampFact**, and the damping time constant in **MP_compActDampTime**. It can be calculated according to the following formula:

$$\text{MP\_comActDampTime} = \frac{k}{2 \cdot \pi \cdot f}$$

k: Factor from 0.8 to 1.0
f: Frequency to be damped (approx. 10 Hz < f < 30 Hz)

Note

The active damping should only be used if improvements actually occur, since the damping could also lead to lower and higher frequencies being fortified.

Activating the active damping:

- Ascertain the deepest resonant frequency (e.g. with the frequency diagram in TNCopt when adjusting the IPC and k_V factor).
- Set **MP_compActDampFact** = 1.5.
- Calculate the damping time constant according to the above formula with k = 0.9, and enter this value in MP_compActDampTime.
- Record **`I (n int)`** or **`s diff`** with the integrated oscilloscope, and move the axis with the axis-direction buttons.
- Vary the value of k up and down somewhat (between 0.8 and 1.0), calculate **MP_compActDampTime** and compare the oscilloscope recordings in order to find the correct value for **MP_compActDampTime** (the value with the lowest amplitude).
- Vary **MP_compActDampFact**, and compare with the recordings in the frequency diagram in TNCopt for the adjustment of the IPC and k_V factor.
- Select the value for **MP_compActDampFact** by evaluating the advantages and disadvantages of the active damping.

MP_compActDampFact

	Damping factor for active damping Available from NCK software version: 597 110-02.
Format:	Numerical value
Input:	0 to 30.000 0 = No damping 1.5: Typical damping factor
Default:	No value, parameter optional (= 0)
Access:	LEVEL3
Reaction:	RUN

MP_compActDampTime

	Damping time constant for active damping Available from NCK software version: 597 110-02.
Format:	Numerical value
Input:	0 to 0.9999 [s] 0 = No damping 0.005 to 0.02: Typical damping time constant
Default:	No value, parameter optional (= 0)
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgControllerComp **compAcc**	 401406

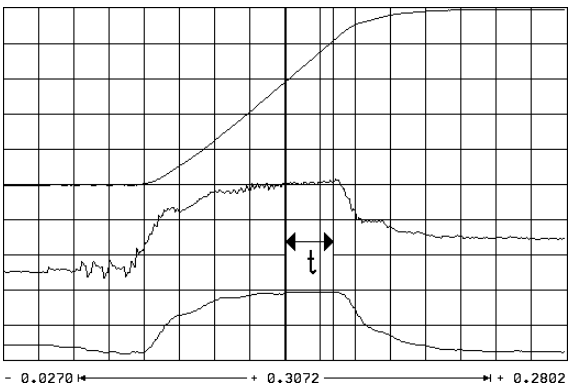
The parameter object CfgControllerComp is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)
- Analog axes (MP_axisHw=Analog)

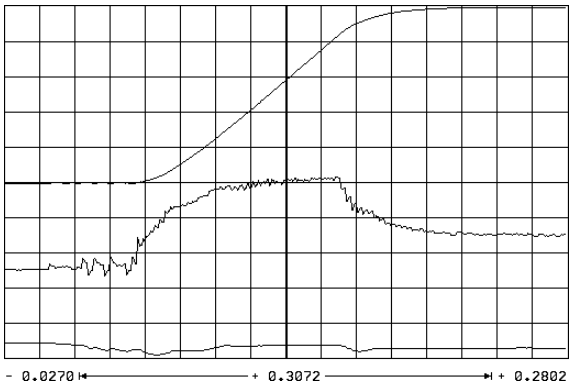
Acceleration feedforward functions only in velocity feedforward control in parallel with the speed controller.

At every change in velocity, spikes of short duration appear in the following error. With acceleration feedforward control you can minimize these spikes:

- First adjust the friction compensation (See "Compensation of sliding friction" on page 485).
- From the integral-action component of the nominal current value I (N INT) calculate the input value for **MP_compAcc.**
- Adjust the acceleration feedforward control with MP_compAcc.



MP_compAcc = 0



MP_compAcc = optimum

For calculation of the acceleration feedforward, the integral-action component of the nominal current value I (N INT) is recorded with the internal oscilloscope. The actual speed value V (ACT RPM) and nominal current value I (NOM) are also recorded for better illustration.

$$\text{MP\_compAcc} = \frac{\text{I (N INT) [A]} \cdot \text{ t [s]} \cdot \text{ 60 [s/min]} \cdot \text{ MP\_distPerMotorTurn [mm]}}{\Delta\text{V (N IST) [mm/min]}}$$

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

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

ΔV (ACT RPM) = actual-speed-value change

MP_distPerMotorTurn = traverse distance per motor revolution

MP_compAcc

Acceleration feedforward
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 000 000 to 6 [As^2]
Default:	0
Access:	LEVEL3
Reaction:	RUN

Limiting the integral factor

In machines with a great deal of stiction, a high integral-action component can accumulate if there is a position error at standstill. This can result in a jump in position when the axis begins moving. In such cases you can limit the integral-action component of the speed controller:

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgSpeedControl **vCtrlIntTime**	 400903

- Enter a limit in MP_vCtrlIntTime.
  Realistic input values: 0.1 to 2.0

MP_vCtrlIntTime

Limit of integral factor of the speed controller
Available from NCK software version: 597 110-02.

Format:	Numerical value
Input:	0.000 000 000 to 30 [s]
Default:	0
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgControllerComp	
compCurrentOffset	401405
compIpcT1	401407
compIpcT2	401408
compIpcJerkFact	401409

The parameter object **CfgControllerComp** is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)
- Analog axes (MP_axisHw=Analog)

Integral Phase Compensation (IPC)

Use the I factor **MP_vCtrlIntGain** of the speed controller to attain a short settling time. However, the I factor has a negative influence on the position controller. The position controller is more likely to oscillate, which often makes it impossible to adjust the k_V factor **MP_kvFactor** to the desired size.

The IPC compensates the negative influence of the I factor on the speed controller, and makes it **possible** to increase the k_V factor.

The IPC is beneficial on the following types of machines:

- Machine type 1: Machines with a dominant natural frequency between 15 Hz and 80 Hz, for which it is not possible to set a sufficiently high k_V factor.
- Machine type 2: Small-to-medium size machines that are driven directly.

Note

- The acceleration feedforward **(MP_compAcc)** must already have been carefully adjusted for both types of machines.
- If after commissioning the IPC you wish to optimize the speed controller again, you must switch off the IPC beforehand, because the IPC influences the curve form.
- Use the same test program to commission the IPC as is used to measure the jerk and the k_V factor.

Machine type 1:

- The machine is commissioned as usual until the k_V factor is to be determined.
- Enter **MP_compIpcT1** = 1 and **MP_compIpcT2** = 0.
- Increase the k_v factor **(MP_kvFactor)** until you reach the oscillation limit.
- Starting value: $\text{MP\_compIpcT2} = \frac{2}{3} \cdot \frac{\text{MP\_compAcc}}{\text{MP\_vCtrlPropGain}}$
- Change **MP_compIpcT2** until you have found the maximum k_v factor. If you cannot find a maximum k_v factor, use the default value.
- Starting value: $\text{MP\_compIpcT1} = \frac{\text{MP\_compAcc}}{\text{MP\_vCtrlPropGain}}$
- Increase **MP_compIpcT1** until you have found the maximum k_v factor. If the value found is significantly greater than the starting value (> factor 2), you should adjust **MP_compIpcT2** again by enlarging and reducing it to find the optimum value.
- $\text{MP\_kvFactor} = 0{,}65 \cdot$ ermittelter k_V

Machine type 2:

- The machine is commissioned as usual until the k_V factor is to be determined.
- Enter **MP_compIpcT1** = 1 and **MP_compIpcT2** = 0.
- Increase the k_v factor **(MP_kvFactor)** until you reach the oscillation limit.
- Starting value: $\text{MP\_compIpcT2} = \frac{2}{3} \cdot \frac{\text{MP\_compAcc}}{\text{MP\_vCtrlPropGain}}$
- Change **MP_compIpcT2,** normally by reducing it, until the following error is at its minimum.
- $\text{MP\_kvFactor} = 0{,}65 \cdot$ ermittelter k_V

MP_compIpcT1

IPC time constant T1
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 100 000 to 1 [s] 0: IPC inactive
Default:	0
Access:	LEVEL3
Reaction:	RUN

MP_compIpcT2

IPC time constant T2
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 100 000 to 1 [s] 0: IPC inactive
Default:	0
Access:	LEVEL3
Reaction:	RUN

Optimizing the jerk feedforward control

In the same manner as the other feedforward controls in a control loop, the jerk feedforward control causes following errors to be reduced or compensated during a dynamic phase (in this case the phase of the change in acceleration).

When the feedforward control is initially configured, the necessary jerk feedforward control is calculated for the known control-loop factors, but the mechanical deformation of the transmission components (e.g. due to torsion of the ball screw or the slack sides and tight sides of belt drives) and the resulting following error are unknown. This can be minimized or compensated with the dimensionless multiplier **MP_compIpcJerkFact,** which affects the jerk feedforward control directly.

When optimizing the jerk feedforward control all other controller parameters must already have been adjusted.

Note

The **MP_compIpcJerkFact** parameter is not available on the CC 61xx or UEC 11x. Use the much more suitable parameter **MP_compTorsionFact** instead (See "Torsion compensation" on page 642).

With the **CC 61xx or CC 424** you can also use this function without active IPC. A prerequisite for the activation of the possibility of compensation with the CC 422 is an active and already adjusted IPC (Integral Phase Compensation).

Commissioning:

- Enter the following test program:

```
BEARBEITUNG [MACHINING]
[maximum traverse range]
N1 G0 X400
N2 G0 X0
N3 M99
END
```

- Run the program at high speed.
- Use the integrated oscilloscope to record the following error.

- **CC 424:** Adjust MP_compIpcJerkFact until a minimal following error occurs in the jerk phase. Here you can compensate the following error for the jerk phase.
  Typical input values: 0 to 0.5 (e.g. 0.01).

Note

Please note that values over 0.5 in MP_compIpcJerkFact do not make sense for the CC 424 controller unit, and are therefore not permissible. Therefore, an error message will be issued for values greater than 0.5.

- **CC 422:** Adjust MP_compIpcJerkFact until a minimal following error occurs in the jerk phase. Here you can compensate positive following errors (MP_compIpcJerkFact > 1) as well as rare negative following errors (MP_compIpcJerkFact < 1) for the jerk phase.
  Typical input values: 0.5 to 2.

MP_compIpcJerkFact

Following error in the jerk phase
Available from NCK software version: 597 110-02.

Format:	Numerical value
Input:	0.000 000 000 to 10
Default:	0
Access:	LEVEL3
Reaction:	RUN

Holding torque

The holding torque is the torque that is required to keep a vertical axis at a standstill.

The holding torque is given by the MANUALplus 620 through the integral-action component of the nominal current value. In most cases the holding torque is constant. The required holding current can therefore be fed forward through MP_compCurrentOffset. This relieves the speed controller.

- To prevent the effect of stiction, measure the current at low speed in both directions (e.g. 10 min^{-1}).
- Calculate the holding current from the mean of the measured current values and enter the result in MP_compCurrentOffset.

Mean:

$$MP\_compCurrentOffset = \frac{I\ SOLL_1 + I\ SOLL_2}{2}$$

Note

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

MP_compCurrentOffset

Holding current
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	–30.000 000 000 to +30 [A]
Default:	0
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
System CfgHardware **driveOffGroupInput**	 100106
Axes ParameterSets [Key name of the parameter set] CfgAxisHardware **driveOffGroup** CfgPowerStage **ampReadyWaitTime**	 400015 401206

The parameter object

- CfgAxisHardware is not required for:
  - Virtual axes (MP_axisMode=Virtual)
- CfgPowerStage is not required for:
  - Virtual axes (MP_axisMode=Virtual)
  - Axes that are for display only (MP_axisMode=Display)
  - Analog axes (MP_axisHw=Analog)

The procedure recommended by HEIDENHAIN for switching the drives on and off, including the correct triggering of the brakes, can be configured with the PLC basic program. HEIDENHAIN recommends using the PLC basic program.

Readiness of the drives

The drives can only be switched on once the drive system is ready. This includes

- the readiness of the supply unit (RDY.PS signal at X69 pin 17a – interrogation via Module 9066 bit 5), the inverters (RDY signal at X51 to X64 pin 10b of the PWM outputs) and of the motors.
- global drive enabling with EMERGENCY STOP via -ES.B.
- the conclusion of the motor orientation (determination of the field angle).
- for optional use of axis groups, these must be enabled via the PLC inputs specified in **MP_driveOffGroupInput**.

Basic settings:

- Ensure that the clamping or brakes activate when an emergency stop is effected.
- To enable drives use the input configured as -ES.B. on the PL 62xx or UEC 11x PLC system module.
- Ensure via Module 9159 (drive controllers are switched off) that the clamping or brakes activate before the drive controllers (current and speed controllers) are switched off.
- Before switching off the drives, save the positions of axes with common encoders.

- Ensure that all control components are wired correctly. (HEIDENHAIN recommends always wiring the system according to the basic circuit diagram of the MANUALplus 620.) As a registered customer, you can find the current basic circuit diagram on the HEIDENHAIN FileBase on the Internet at http://filebase.heidenhain.de.
- If you are using axis groups, assign the axis to the axis group required for drive enabling in **MP_driveOffGroup**.

To switch on the drive controller:

- With **PP_AxDriveOnRequest**, switch the drive motor on (See "Opening the position control loop" on page 569).
- Use **NN_AxDriveOn** to check the status of the drive motor.

If drive enabling through X150/X151 is missing, the drive motor will **always** be switched off and the position controller will be informed accordingly.

If drive enabling of axis groups is used, the PLC program should perform a plausibility test to see if all axes used are defined in **MP_driveOffGroup.**

Danger

If **MP_driveOffGroup** is configured incorrectly, the drive motor might not switched be off correctly!

Note

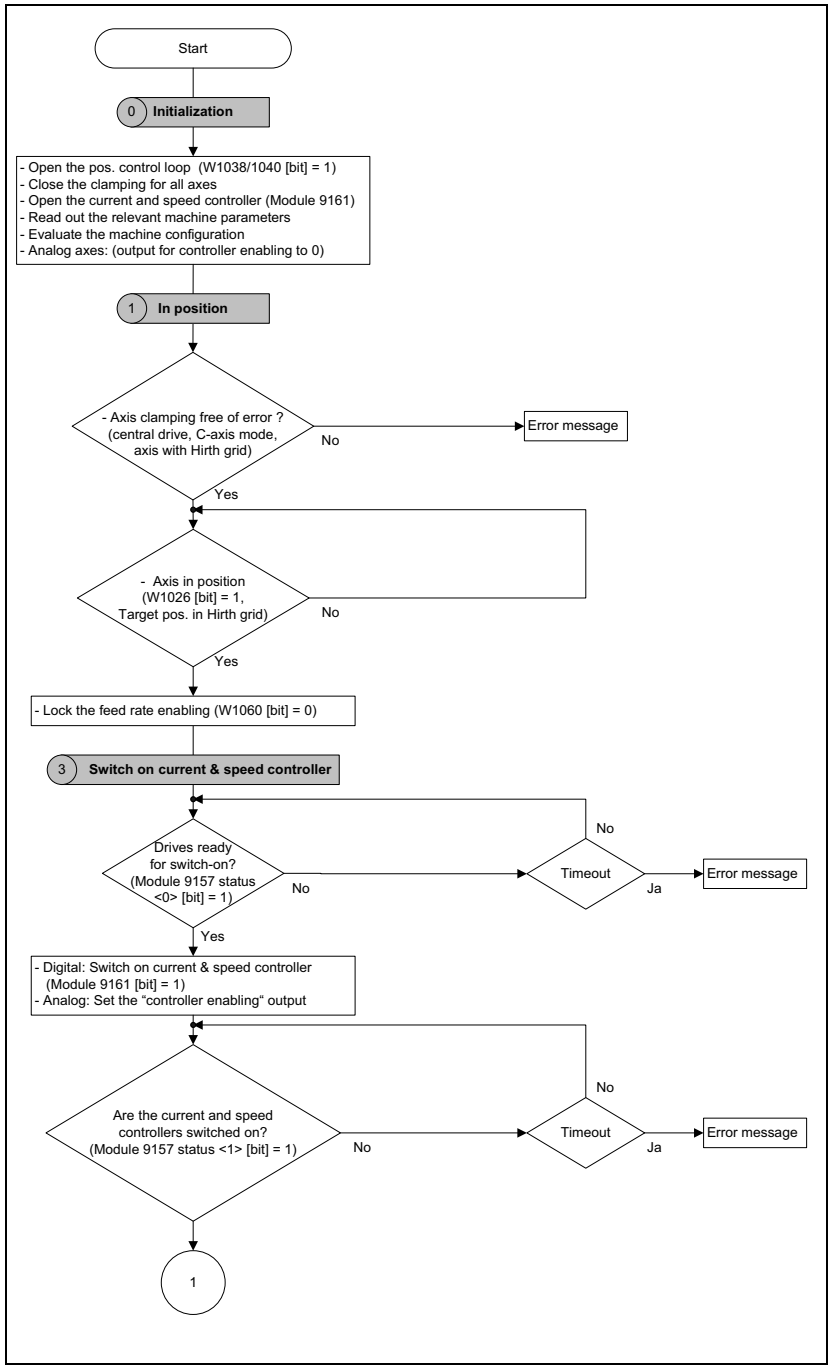
- If drive enabling is missing, the error message **8B40 No drive release <axis>** appears.
- If you do **not** want to use drive enabling for axis groups, but rather just global drive enabling through -ES.B, remove the optional parameter **MP_driveOffGroup** from the configuration of your machine.
- The MANUALplus 620 monitors the time between the switch-on of the drive controller (via Module 9161) and the READY signal of the power module (via the PWM cables). If the READY signal is missing after the waiting time has passed, the error message **8B40 No drive release <axis>** appears. In **MP_ampReadyWaitTime** you preset the permissible waiting time.
- If the readiness signal is reset at the PWM outputs, the drive controllers (current and speed controllers) are switched off. Normally, the error message **MOVEMENT MONITORING IN <AXIS> B** is output through the position control loop. Subsequently, the PWM signal release is switched off by the reset signal.

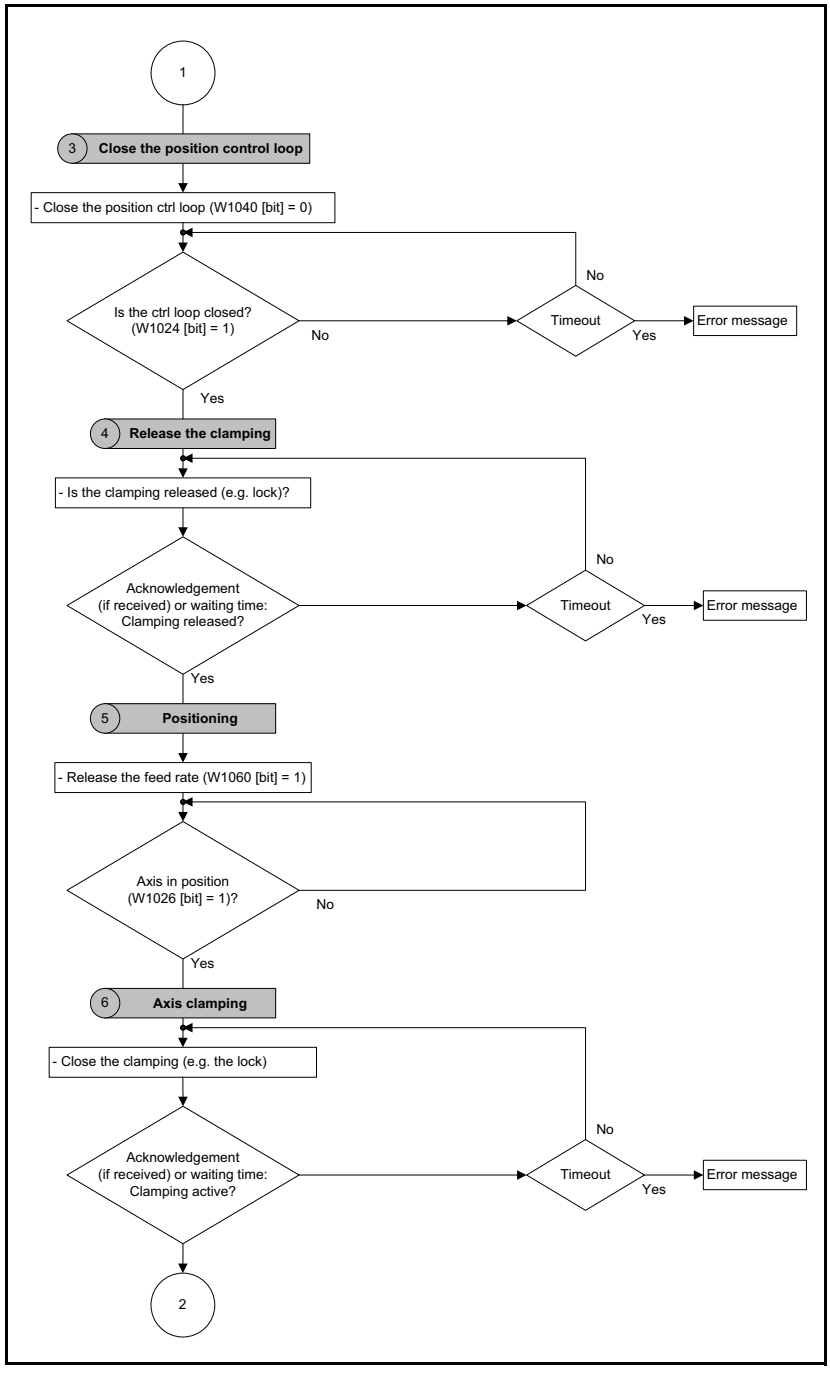
The drive controller cannot be switched on until the control-is-ready signal of the inverter, the global drive enabling through -ES.B and the axis group enabling are available.

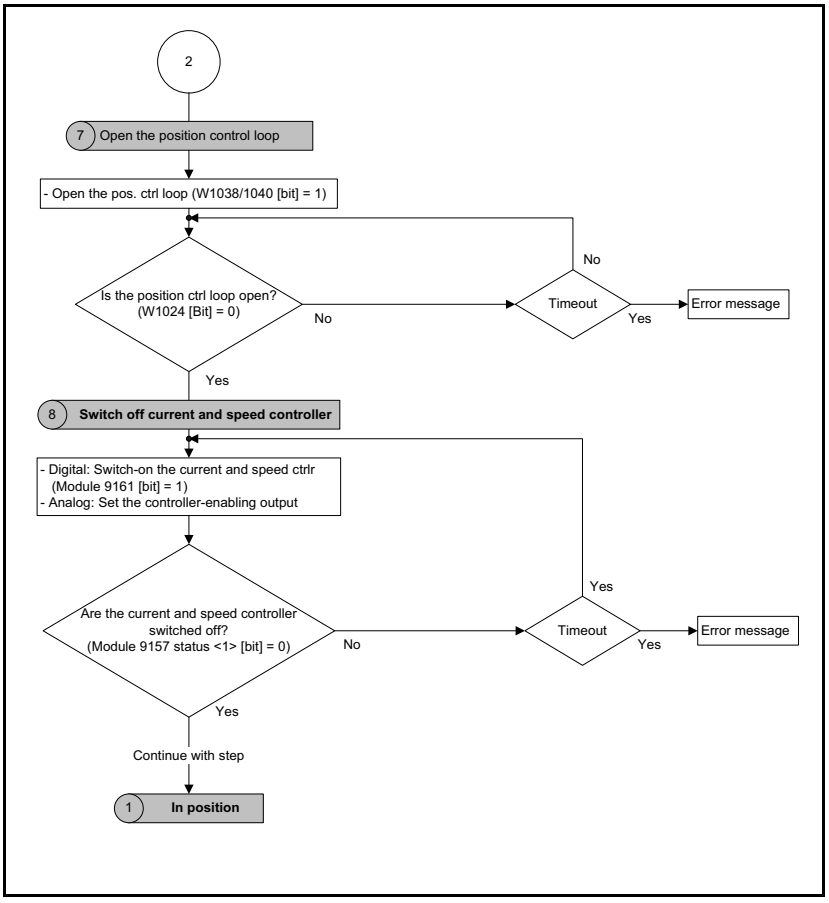
Note

If you do **not** want to use drive enabling for axis groups, but rather just global drive enabling through -ES.B, set **MP_driveOffGroup=NONE** for all axes, or remove the parameter from your configuration.

Procedure for "switching drives on and off"




1
3 Close the position control loop
- Close the position ctrl loop (W1040 [bit] = 0)
Is the ctrl loop closed? (W1024 [bit] = 1)
No
Timeout
Yes
Error message
No
Yes
4 Release the clamping
- Is the clamping released (e.g. lock)?
Acknowledgement (if received) or waiting time: Clamping released?
No
Timeout
Yes
Error message
Yes
5 Positioning
- Release the feed rate (W1060 [bit] = 1)
Axis in position (W1026 [bit] = 1)?
No
Yes
6 Axis clamping
- Close the clamping (e.g. the lock)
Acknowledgement (if received) or waiting time: Clamping active?
No
Timeout
Yes
Error message
2


2
7 Open the position control loop
- Open the pos. ctrl loop (W1038/1040 [bit] = 1)
Is the position ctrl loop open?
(W1024 [Bit] = 0)
No
Timeout
Yes
Error message
No
Yes
8 Switch off current and speed controller
- Digital: Switch-on the current and speed ctrlr
(Module 9161 [bit] = 1)
- Analog: Set the controller-enabling output
Are the current and speed controller
switched off?
(Module 9157 status <1> [bit] = 0)
No
Timeout
Yes
Error message
Yes
Yes
Continue with step
1 In position

MP_driveOffGroup

Circuit of X150/X151
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **None**
Axis not assigned (disabling only through I32)
Group1
Axis is assigned to X150 pin 1
Group2
Axis is assigned to X150 pin 2
Group3
Axis is assigned to X150 pin 3
Group4
Axis is assigned to X150 pin 4
Group5
Axis is assigned to X151 pin 1
Group6
Axis is assigned to X151 pin 2
Group7
Axis is assigned to X151 pin 3
Group8
Axis is assigned to X151 pin 4

Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

The MANUALplus 620 monitors the time between the switch-on of the drive controller (by means of **PP_AxDriveOnRequest**) and the RDY signal (via the PWM cable). If the READY signal is missing after the waiting time has passed, the error message **8B40 NO DRIVE ENABLE <axis>** appears.

▶ Enter in **MP_AmpReadyWaitTime** the permissible waiting time.

MP_ampReadyWaitTime

Waiting time between switch-on of the power module and the RDY signal
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.001 000 000 to 4999 [s]
0: Waiting time of 2 [s]
Default: 0 [s]
Access: LEVEL3
Reaction: RESET

Note

If drive enabling is missing, the error message **8B40 No drive release <axis>** appears.

If you disconnect the voltage

- -ES.B, all drive controllers are switched off.
- on the PLC inputs for axis group enabling, the drive controllers of the corresponding axis group(s) are switched off.

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

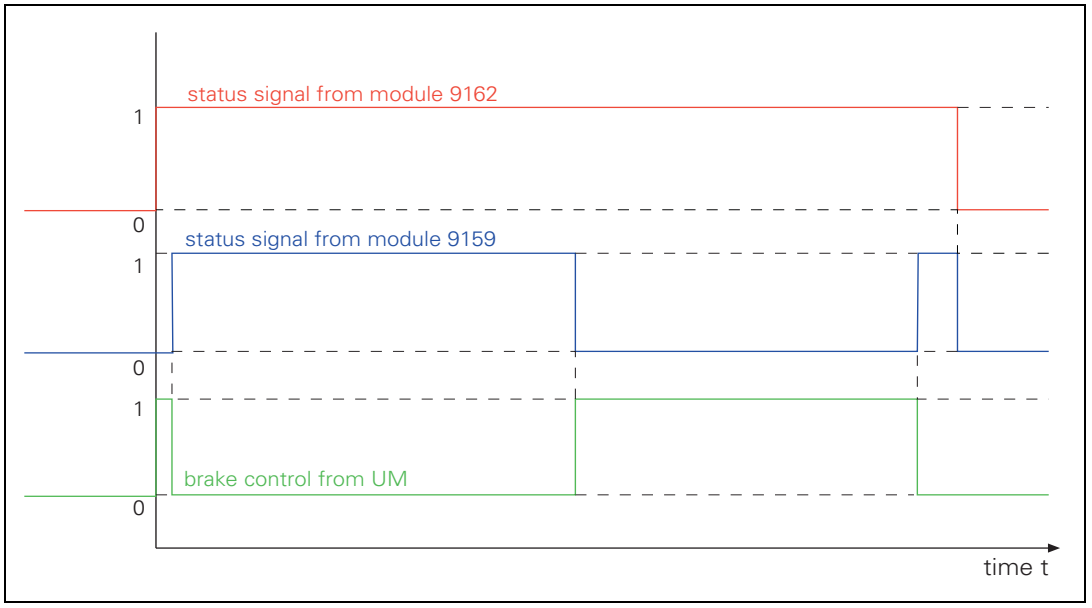
- Call Module 9159. The drive motors that are switched off are returned in bit code.

Note

The bit-encoded status signal supplied by Module 9159 is a request to lock the brake/clamping of the servo drive. Do not use the signal as a request to switch off the drive concerned.

The following logic diagram shows the status signals of the PLC modules 9159 and 9162 as well as the trigger signal for the brake from the HEIDENHAIN inverter.

A switch-on procedure with a test of the motor brake and a switch-off procedure with an overlap time of 200 ms are recorded:



The logic diagram shows that the trigger signal for the motor brake can be calculated from the status signals of the Modules 9159 and 9162:

Brake is not active = Status signal of Module 9162 AND NOT status signal of Module 9159

This enables you to use the PLC to apply the test of the motor brakes to non-HEIDENHAIN power modules, too.

The status of the drive controller is shown in **NN_AxDriveOn.** Use Module 9157 to inquire whether the drive controller is ready to be switched on.

Module 9157 Status request for drive hardware
The module provides information about the drive controller status.

Information about drive controller status: The drive is ready if

- the DSP processor board has been enabled.
- the drive was activated through external switching.
- the current controller and the speed controller have been activated.

Call:

PS B/W/D/K <Selection of status information>
0: Drive controller is ready
Result 0: Drive is not ready and cannot be switched on
Result 1: Drive is ready and can be switched on
1: Drive controller status
Result: Bit-encoded (bit 0 corresponds to logical axis 0, etc.)
2: Axis enabled via axis group (bits 0 to 13 = 1) or axis not enabled (bits 0 to 13 =0)
3: Signals available at X150/X151
(bits 0 to 7 = X150; bits 8 to 15 = X151)
4: Reserved

CM 9157

PL B/W/D <Axis status information bit-encoded>
(Bit 0 represents logic axis 0, etc.)

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Status information was ascertained
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid status information was programmed
	24	Module was called in a spawn job or submit job

Module 9159 Early warning for stopping the drives
The module provides an early warning indicating that the drives will be stopped. This early warning is made available approx. 200 ms before the drive readiness is canceled.

Call:

CM 9159

PL W/D <Drives, in bit code, that are switched off in 200 ms>
(Bit 0 represents logic axis 0, etc.)

Module 9161 Enable the drive controller

With this module you can switch the drive controllers (speed and current controllers) on and off for specific axes. A nominal speed value is also output when the drive controller is not enabled.

Call:

PS W/D/K <Enabled axes>
Bit 0...13 -> axis 1...14, Bit 15 -> spindle
0: No drive controller enabling
1: Drive controller enabling

CM 9161

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	No error
	1	Control has no current controller, or the call is in a spawn or submit job

Module 9162 Status request of the drive controller

Call:

CM 9162

PL B/W/D <Drive is ready>
Bit 0...13 -> axis 1...14, Bit 15 -> spindle
0: Not ready
1: Ready

Module 9169 Axes for which I32 does not switch off the drives

Call:

PS B/W/D/K <Axes bit-encoded>

CM 9169

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgCurrentControl	
iCtrlPropGain	401001
iCtrlIntGain	401002

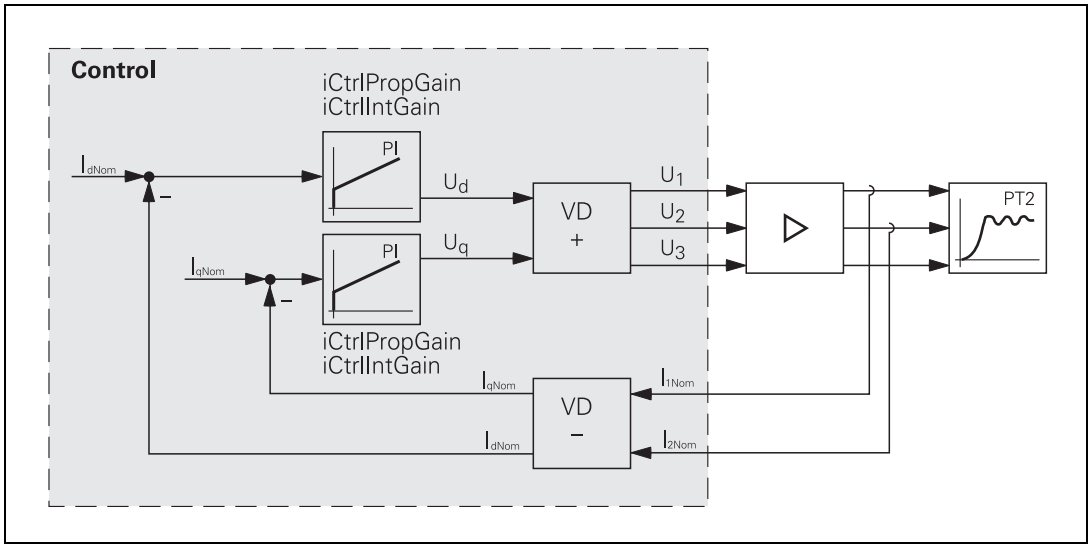
The parameter object CfgCurrentControl is not required for:

- Virtual axes (MP_axisMode = Virtual)
- Axes that are for display only (MP_axisMode = Display)
- Analog axes (MP_axisHw = Analog)

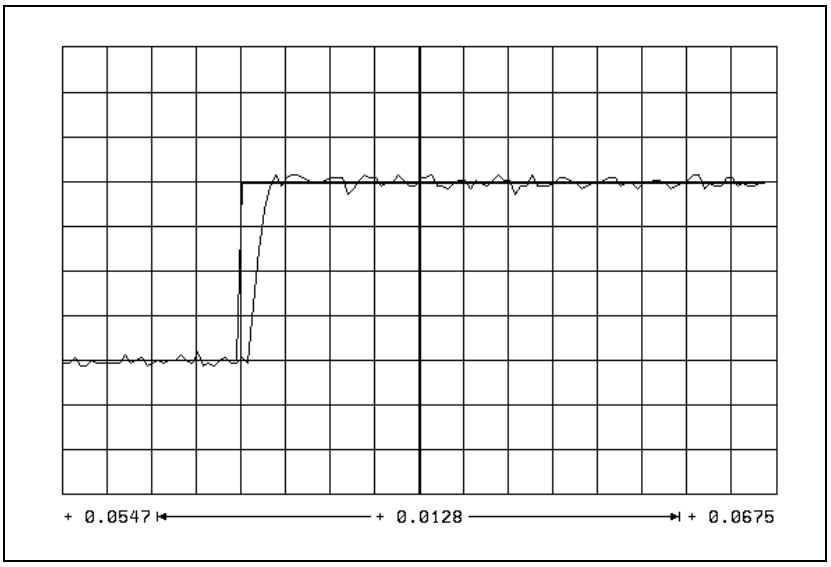
The nominal values for magnetizing current I_{dnom} and torque current I_{qnom} are divided into the PWM signals U_1, U_2 and U_3 through a PI controller and vector rotator VD+, and are transferred to the power module through X51 to X60.

The actual current values I_{1act} and I_{2act} are determined by the power module and are transferred to vector rotator VD– through X51 to X60. The vector rotator determines the actual values of magnetizing current I_{dist} and torque current I_{qnom}.

Circuit diagram:



You adjust the current controller to attain the optimum result, with the position and speed controller switched off.



The step response is adjusted such that there is no overshoot and the rise time is as small as possible:

- In **MP_iCtrlPropGain,** define the P factor of the current controller.
- In **MP_iCtrlIntGain,** define the I factor of the current controller.

MP_iCtrlPropGain

	Proportional component for digital current controller Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 9999.99 [V/A]
Default:	0 [V/A]
Access:	LEVEL3
Reaction:	RUN

MP_iCtrlIntGain

	Integral-action component for digital current controller Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 9 999 999.9 [V/As]
Default:	0 [V/As]
Access:	LEVEL3
Reaction:	RUN

Expanded current controller bandwidth (only CC 61xx)

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgCurrentControl	
iCtrlDiffFreqFF	401007
iCtrlAddInfo	401008

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

- Using machine parameter **MP_iCtrlAddInfo** to increase the controller factors in the current control loop,
- Proportionally differentiating feedforward (D feedforward) to optimize the command action of the current control loop (**MP_iCtrlDiffFreqFF** parameter).

Increased controller factors in the current control loop:

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

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

D feedforward for optimization of the current controller command action:

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

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

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

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

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

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

Note

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

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

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

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

- In the corresponding parameter set of the axis **MP_iCtrlAddInfo**, set bit 0 = 1 to activate the function.
- Readjust the current controller with TNCopt (Bode diagram function; see TNCopt User's Manual).

Note

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

MP_iCtrlDiffFreqFF

Cutoff frequency of the current controller feedforward in Hz
(This function is only available when using the CC 61xx controller unit!)
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: 0 Hz to f_g
The cutoff frequency f_g depends on the PWM frequency (see table).
0: Current controller feedforward deactivated

Default: No value, parameter optional

Access: LEVEL3

Reaction: RUN

MP_iCtrlAddInfo

Switching the increased current controller factors on and off.
(This function is only available when using the CC 61xx controller unit!)
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: Bit0 = 0:
Increased current controller factors switched off
Bit0 = 1:
Increased current controller factors switched on

Default: No value, parameter optional

Access: LEVEL3

Reaction: RUN

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgServoMotor	
motEmergencyStopRamp	401306
motPbrMax	401307
motPbrMaxAcFail	401308

The parameter object CfgServoMotor is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)
- Analog axes (MP_axisHw=Analog)

In an emergency stop and power failure the axes and the spindle must be braked as quickly as possible. If the braking energy cannot be drawn off quickly enough, the dc-link voltage increases sharply. If, in unfavorable cases, the maximum regenerative power of the inverter is exceeded, the inverter is switched off and the axes and spindle(s) coast to a stop. A powerful braking of the drives also puts a high strain on the mechanics of the machine. With gantry axes a mechanical offset can occur between the master and slave axes.

Problems with **non-regenerative** inverters (with braking resistors) during braking mostly arise if the inverter is switched off too early. The strain on the mechanics during braking is reduced, but can also be influenced with braking strategies.

Regenerative inverters usually do not develop problems if they are switched off. As a rule, the mechanics of the machine must be taken into account.

During a **power failure** ("powerfail"), the "SH1B" signal at X51 to X60 is maintained for 10 more seconds, in order to brake the spindle(s). At the same time, the control tries to reset the PLC outputs.

Braking on the brake ramp

Define **MP_motEmergencyStopRamp** (brake ramp in an emergency stop):

- Use the value in MP_maxAcceleration as the minimum value
- Use the emergency stop to brake the axis from rapid traverse, or the spindle from high speed.
- Increase the value entered in MP_motEmergencyStopRamp until the braking time is as short as possible and the mechanics of the axis or spindle are not overloaded.

Note

If the brake ramp defined in **MP_motEmergencyStopRamp** has a smaller gradient than the one defined in **MP_maxAcceleration,** the brake ramp from **MP_maxAcceleration** is used for braking.

Limitation of the braking power

The use of energy recovery and/or a braking resistor must also be taken into account when braking the drives in an emergency stop or in the event of a power failure:

Inverter **without energy recovery** (with braking resistor):

- **MP_motPbrMax:** Calculate the maximum braking power from the formula below.
- **MP_motPbrMaxAcFail:** Calculate the maximum braking power from the formula below.

Inverter **with energy recovery** (without braking resistor):

- Enter **MP_motPbrMax**= 0 so as not to limit the braking power.
- **MP_motPbrMaxAcFail**: The braking power cannot be limited in a power failure. Axes and spindles will coast to a stop.

Inverter **with energy recovery and additional braking resistor:**

- Enter **MP_motPbrMax**= 0 so as not to limit the braking power.
- **MP_motPbrMaxAcFail:** Calculate the maximum braking power from the formula below.

Calculation of maximum braking power:

- For **inverters with braking resistor**, calculate the maximum braking power from the following formula:

$$\text{Maximum braking power} = \frac{U_Z^2}{R \cdot 1000}$$

R = Braking resistance [Ω]
U_Z = dc-link voltage [V]

Note

- If the braking power defined in **MP_motPbrMax** or **MP_motPbrMaxAcFail** is smaller than the one defined in **MP_maxAcceleration,** the braking power from **MP_maxAcceleration** is used for braking.
- If after entry of a value in **MP_motPbrMax** or **MP_motPbrMaxAcFail,** the mechanics are overloaded by the braking process, reduce the value until you have found an optimum between braking time and mechanical load.

Machine parameters

MP_motEmergencyStopRamp

Braking ramp in an emergency stop
Available from NCK software version: 597 110-02.

Format: Numerical value
Input: 0.000 000 000 to 999.9 [m/s^2] or [$1000°/s^2$]
0: Function inactive
Default: 0
Access: LEVEL3
Reaction: RESET

MP_motPbrMax

Maximum brake power for emergency stop
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.0 to 3200.0 [kW]
0: Braking power is not limited
Default: 0
Access: LEVEL3
Reaction: RESET

MP_motPbrMaxAcFail

Maximum brake power for power failure
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 3200 [kW]
0: Braking power is not limited
Default: 0
Access: LEVEL3
Reaction: RESET

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgServoMotor	
motEncCheckOff – Bit2	401303
motPMax	401308
motMMax	401310

The parameter object CfgServoMotor is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Analog axes (**MP_axisHw** = Analog)

You can limit the power of your spindle motor to achieve wider gear ranges. Wide-range motors are characterized by a larger speed range with higher torque at low speed.

One solution for bringing about this behavior is to use an oversized motor, and to limit the maximum power. However, power limiting does not reduce the high torque to the speed at which power limiting becomes effective. This high torque (until power limiting takes effect) can be reduced with torque limiting, in order to keep the mechanics of the machine from becoming overloaded.

With torque limiting you can also limit the torque of the axis motors, in order to keep the mechanics of the machine from becoming overloaded. Power limiting is not useful for axis motors.

For **axes and spindles**, the torque is limited to the value taken from either the table of power modules or the motor table, whichever is lower.

On supply units where the $\overline{\text{ERR.IZ.GR}}$ signal is available, the power is limited via MP_motPMax in case of error.

HEIDENHAIN recommends activating this monitoring function (not with UE 2xx).

The torque can be calculated for any speed:

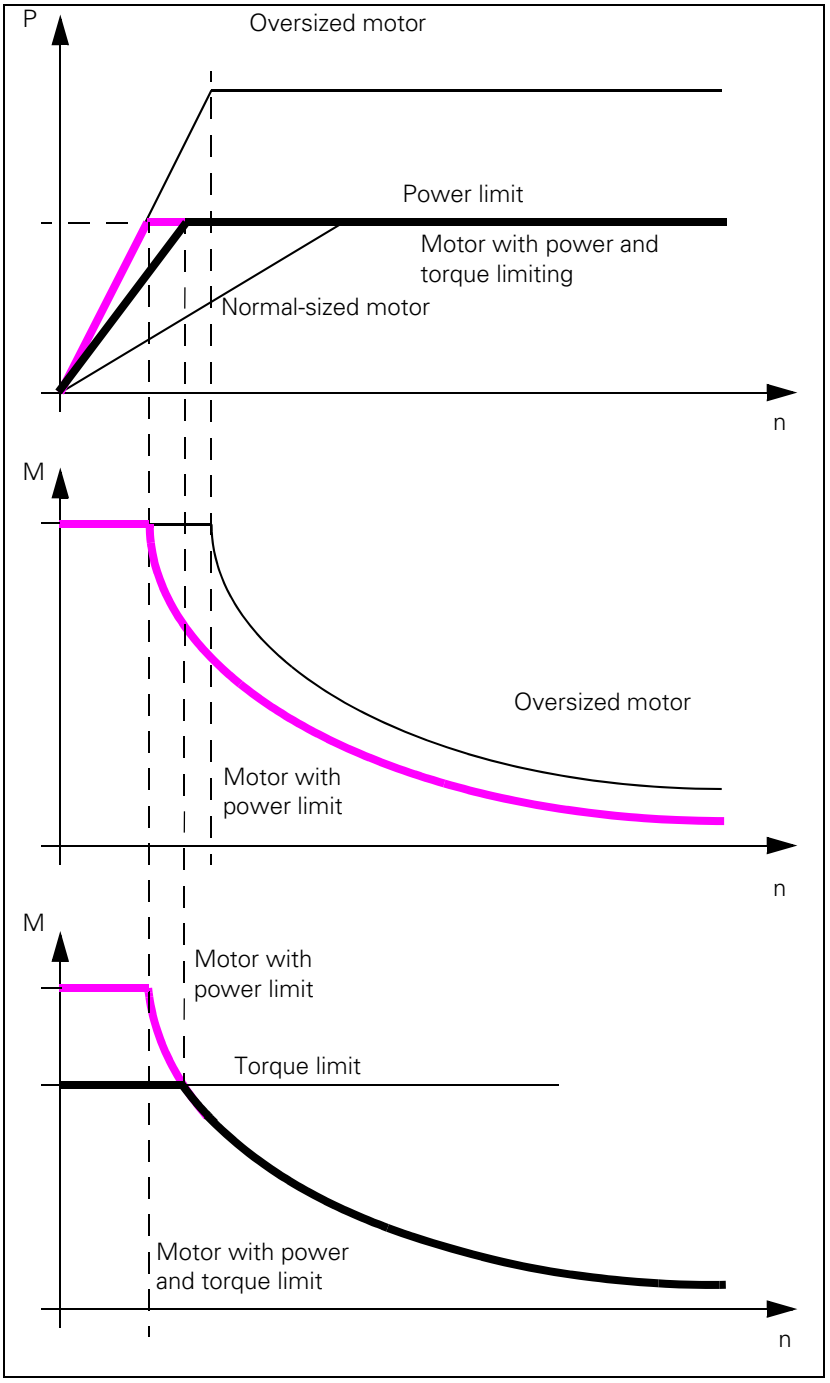
$$M = \frac{P \cdot 60}{n \cdot 2 \cdot \pi}$$

M: Torque [Nm]
P: Power [W]
n: Speed [min^{-1}]

Note

The power and torque limiting can have an effect on the braking of the spindle in an emergency stop.

- Enter the maximum power for the spindle in **MP_motPMax.**
- Enter the maximum torque for the drive motor in **MP_motMMax.**
- Activate the power limiting of the spindle at $\overline{\text{ERR.IZ.GR}}$ via **MP_motEncCheckOff – Bit2=0** (not for UE 2xx)



Machine parameters

MP_motEncCheckOff

Monitoring functions
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: **Bit 2** – Power limit of spindle with $\overline{\text{ERR.IZ.GR}}$ (only for HEIDENHAIN inverters, except UE 2xx)
0: Power limit active
1: Power limit inactive

Default: 0

Access: LEVEL3

Reaction: RESET

MP_motPMax

Power limiting of motor
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: 0.000 000 000 to 3200 [kW]
0: No power limiting

Default: 0

Access: LEVEL3

Reaction: RESET

MP_motMMax

Maximum torque
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: 0 to 30 000 [Nm]
0: No torque limiting

Default: 0

Access: LEVEL3

Reaction: RESET

Module 9128 Torque limiting by the PLC

Module 9128 can be used to program a maximum torque for the programmed axis. The torque of the drive is limited to the programmed value. The value –1 cancels the torque limitation and the value from the motor data becomes effective again. The torque can be limited in [mA] or in [0.1%] of the rated current.

Condition:

- The module is only executable in the cyclic PLC program.
- The programmed value for the maximum torque may not be higher than the value in the motor data. If the programmed value is higher than the value in the motor data, the value in the motor data is used as the limit.
- A torque value of 0 cannot be programmed.
- Programming a torque value of –1 cancels limitation. The original value from the motor data becomes effective again.
- The unit of the resulting torque is [mA].
- If a drive is switched off, the torque from the motor data becomes effective again when it is switched on.

Call:

PS	B/W/D/K	<Mode> 0: Current in [0.1%] of the rated current 1: Current in [mA] (like Module 9158)
PS	B/W/D/K	<Axis number / spindle number>
PS	B/W/D/K	<Torque> –1: Cancel the torque limiting
CM	9128	

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Torque limiting programmed
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Invalid value for torque
	2	Invalid value for axis number or mode, axis is an open-loop axis or is temporarily not a closed-loop axis
	24	Module was called in a spawn or submit job

Module 9129 Status of torque limiting by the PLC

Module 9129 is used to determine the current status of torque limiting for the programmed axis. The momentary maximum torque can be determined in [mA] or in [0.1%] of the rated current.

Condition:

- The module is only executable in the cyclic PLC program.
- The greatest possible return value is the value resulting from the motor data.
- If torque limiting is not active, the maximum current can be determined from the motor data.

Call:

PS	B/W/D/K	<Mode> 0: Limiting active/inactive 1: Current in [mA] 2: Current in [0.1%] of the rated current
PS	B/W/D/K	<Axis number / spindle number>
CM	9129	
PL	B/W/D	<Status> Mode 0: 0 = Limiting active / 1 = Limiting inactive Mode 1: Current in [mA] Mode 2: Current in [0.1%] of the rated current

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Status ascertained
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid value for axis number or mode, axis is an open-loop axis or is temporarily not a closed-loop axis
	24	Module was called in a spawn or submit job

Module 9158 Maximum torque

With Module 9158 you can limit the torque of an axis or spindle. The maximum torque resulting from the data in the control's motor table cannot be exceeded. In this case the torque is limited to the value from the motor specifications. After the drive is switched off, the original torque becomes effective again.

If torque limiting is active, the standstill monitoring is inactive; only the motion monitoring remains active.

The torque-producing current required for the desired torque must be transferred to the module:

Synchronous motor	Asynchronous motor
$k_M = \frac{P_N}{I_N \cdot 2 \cdot \pi \cdot \frac{n_N}{60}}$ $I_q = \frac{M}{k_M}$ I_q: Torque-producing current M: Desired torque k_M: Torque constant n_N: Rated speed (from motor table) I_N: Rated current (from motor table) P_N: Rated power output (from motor table)	■ Armature control range ($n < n_{FS}$) $I_q = \frac{M \cdot n_N \cdot 2 \cdot \pi \cdot \sqrt{I_N^2 - I_0^2}}{P_N \cdot 60}$ ■ Field weakening range ($n > n_{FS}$) $I_q = \frac{M \cdot n_N \cdot n \cdot 2 \cdot \pi \cdot \sqrt{I_N^2 - I_0^2}}{P_N \cdot n_{FS} \cdot 60}$ I_q: Torque-producing current M: Desired torque n_N: Rated speed (from motor table) n: Current speed I_N: Rated current (from motor table) I_0: No-load current (from motor table) P_N: Rated power output (from motor table) n_{FS}: Threshold speed for field weakening (from motor table)

Danger

If Module 9158 is used, then certain monitoring functions regarding the drives must be switched off. Please note the following error messages and their possible consequences (see DSP error messages)

- **C380 Motor <axis> not controllable**
- **C3B0 Motor <axis> does not rotate**

Attention

If you are using a CC 422 controller unit, then please note that the current values in the oscilloscope are all peak values (and not effective values as with the CC 424). Measured values then appear greater by a factor of the square root of 2 than the calculated values.

Call:
PS B/W/D/K/S<Axis number / spindle number>
PS B/W/D/K/S<Torque-producing current in mA>
–1 = Torque given in motor specifications
CM 9158

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Torque preset active
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	0 Nm torque transferred
	2	Invalid value for axis number or mode, axis is an open-loop axis or is temporarily not a closed-loop axis
	24	Module was called in a spawn or submit job

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgPositionFilter **manualFilterOrder**	 401605

Filter before position control loop

The **MP_manualFilterOrder** parameter differentiates between axes and spindles. In the **`Manual Operation`** and **`El. Handwheel`** operating modes, or if axes are moved by PLC, a mean-value filter is used as a nominal position value filter for axes.

MP_manualFilterOrder for spindles: See "Filtering the acceleration values" on page 691

- Define the order of the mean-value filter for axes in **MP_manualFilterOrder.**

MP_manualFilterOrder

Order of mean-value filter in Manual mode
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	1 to 51
Default:	11
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgPosControl **kvFactor** **feedForwardFactor**	 400801 400806
Axes ParameterSets [Key name of the parameter set] CfgAxisAnalog **analogOutput** **analogOffset** **kvFactor2** **kvSpeedLimit** **maxFeedAt9V** **accForwardFactor** **compStrength** **compWidth** **compTimeOffset** **compFFAdjust** **compRefAcc** **compLimitFactor** **noOffsetAdjust** **unipolar**	 400101 400102 400103 400104 400105 400106 400107 400108 400109 400110 400111 400114 400112 400113

General information

Analog axis feedback control is based on the following formula:

$$U_{out} = (P_{err} \cdot \textbf{kvFactor} + V_{nom} \cdot \textbf{feedForwardFactor} + A_{nom} \cdot \textbf{accForwardFactor}) \cdot \frac{9\,V}{\textbf{maxFeedAt9V}}$$

Value, parameter	Unit	Description
U_{out}	V	Output voltage (analog nominal speed value)
P_{err}	mm	Following error (servo lag)
kvFactor	1/s	Kv factor (proportional component of position controller)
V_{nom}	mm/min	Nominal velocity
feedForwardFactor		Factor for velocity feedforward control
A_{nom}	m/s^2	Nominal acceleration
accForwardFactor		Factor for acceleration feedforward control
maxFeedAt9V	mm/min	Assumed velocity of the axis at 9 V

The parameter object **CfgAxisAnalog** is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Digital axes **MP_axisHW** = CC or None)

Characteristic curve kink point (only for analog axes)

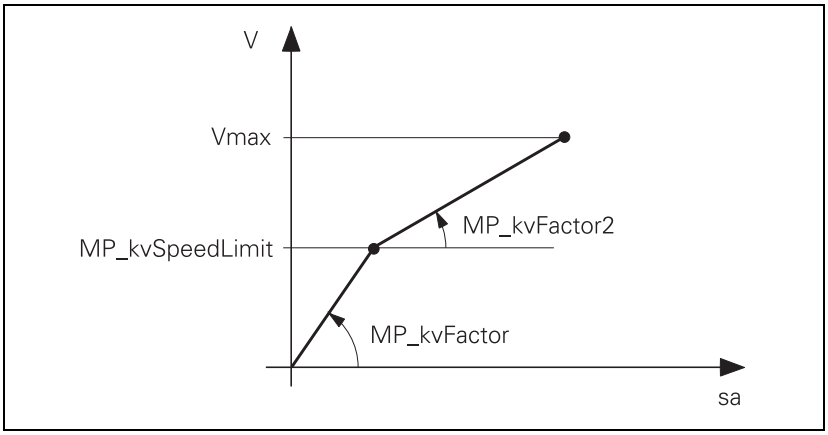
For machines with high rapid traverse, you can not increase the k_V factor enough for an optimum control response to result over the entire velocity range (from standstill to rapid traverse).

In this case, define a characteristic curve kink point, which has the following advantages:

- High k_V factor in the low range
- Low k_V factor in the upper range (beyond the machining velocity range)

▶ Define the position of the kink point in **MP_kvSpeedLimit.**

▶ In **MP_kvFactor2,** enter the k_V factor for the upper range.



sa: Following error

The characteristic curve kink point must lie above the tool feed rate!

MP_kvFactor2

	Proportional component of position controller above MP_kvSpeedLimit Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 1000 [1/s]
Default:	0 [1/s]
Access:	LEVEL3
Reaction:	RUN

MP_kvSpeedLimit

	Limit velocity for MP_kvFactor2 Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0.000 000 000 to 36 000 000 [mm/min]
Default:	0.0 [mm/min]
Access:	LEVEL3
Reaction:	RUN

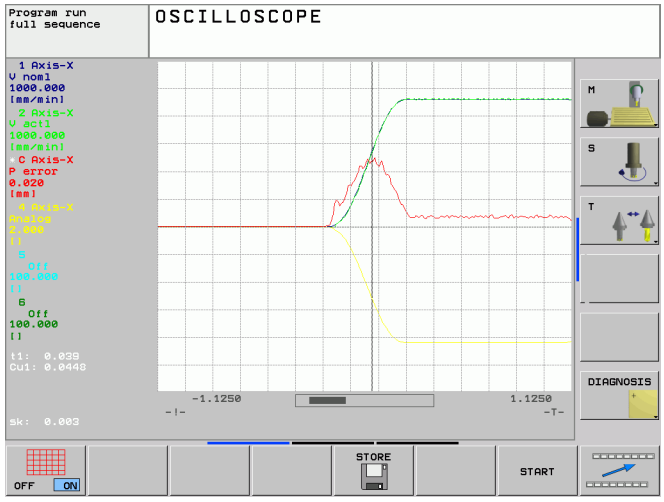
Acceleration feedforward control for analog axes

MP_accForwardFactor allows you to influence acceleration feedforward control for analog axes.

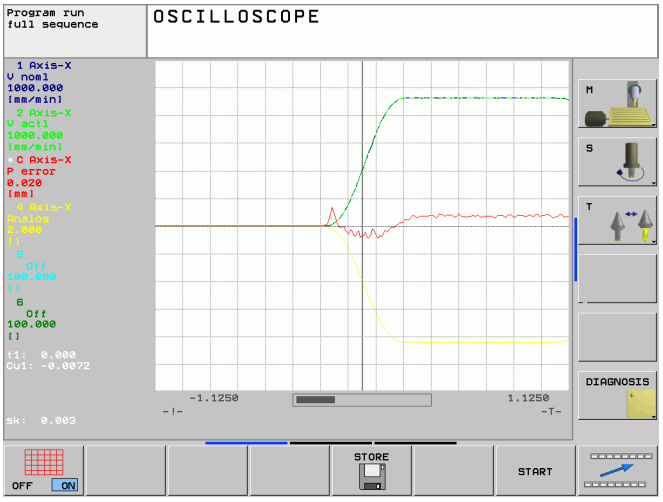
The following error (servo lag) should be as small as possible during the acceleration phase. The **MP_accForwardFactor** parameter directly affects the following error.

The best method for ascertaining the appropriate value for the parameter is by carefully approaching the ideal value. Proceed as follows:

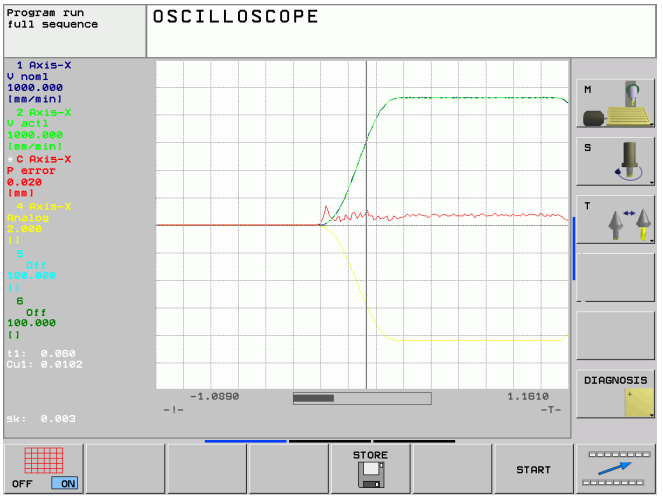
- Enter 0.005 as the preliminary starting value in the parameter **MP_accForwardFactor.**
- Press the axis-direction key and record the following error with the integrated oscilloscope.
- Ascertain the correct setting by carefully increasing the preliminary starting value; i.e. use the value 0.01 for the next measurement.
- Determine the value just before an undershoot forms with the measured following error.



MP_accForwardFactor not adjusted



MP_accForwardFactor adjusted too high



MP_accForwardFactor correctly adjusted

MP_accForwardFactor

Factor for acceleration feedforward control
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 0.01
Default: 0
Access: LEVEL3
Reaction: RUN

Compensation of reversal peaks for analog axes

The compensation of the reversal peaks affects the nominal speed value, which is output at the analog nominal value output of the control (X8). If an axis reverses its direction of movement, a time-dependent compensation curve is superimposed on the nominal speed value.

Machine parameters

MP_compStrength

Strength of the compensation
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	–999 999 999.999 999 999 to +999 999 999.999 999 999 [mm] 0: No compensation Positive input value: Compensation works in the direction of acceleration. Negative input value: Compensation works against the acceleration.
Default:	0
Access:	LEVEL3
Reaction:	RUN

With the **MP_compWidth** parameter you define the distance to the reversal peak. The distance given refers to the offset entered under **MP_compTimeOffset.** Enter the traverse path in which the compensation curve is to be superimposed over the nominal speed value.

MP_compWidth

Specify, with respect to MP_compTimeOffset=0, the distance from the reversal point at which compensation is to begin.
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0 to +999 999 999.999 999 999 [mm], only positive values
Default:	0.001
Access:	LEVEL3
Reaction:	RUN

The parameter **MP_compTimeOffset** shifts the compensation curve along the time axis. The width is not changed. The velocity of the axis at which the compensation function is to reach its maximum is defined. This means the higher the acceleration at the reversal point, the closer the maximum will be to the reversal point at the time of direction reversal.

MP_compTimeOffset

Time offset of the compensation
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: –999 999 999.999 999 999 to +999 999 999.999 999 999 [mm/min]
0: Compensation parabola reaches its maximum at the time of direction reversal
Positive input value:
The compensation curve is delayed, which means that the maximum will not be reached until after the direction reversal.
Negative input value:
The compensation curve is moved to an earlier position, which means that the maximum is reached before the direction reversal.

Default: 0

Access: LEVEL3

Reaction: RUN

Use machine parameter **MP_compFFAdjust** to adjust the area below the compensation curve as a function of the velocity at the reversal point. The area entered in the machine parameter **MP_compStrength** is valid for the acceleration entered in **MP_compRefAcc.** The compensation strength is increased or decreased during accelerations that differ from the acceleration in **MP_compRefAcc.**

MP_compFFAdjust

Additive correction of the compensation strength to the machine parameter MP_compStrength
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: –999 999 999.999 999 999 to +999 999 999.999 999 999 [mm]
0: The compensation strength is constant over all acceleration values and is equal to the value in MP_compStrength.
> 0: The area below the compensation curve becomes larger for low accelerations.
< 0: The area below the compensation curve becomes smaller for low accelerations.

Default: 0

Access: LEVEL3

Reaction: RUN

MP_compRefAcc

For the acceleration given here, the area below to compensation curve is set equal to the value entered in MP_compStrength.
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –999 999 999.999 999 999 to +999 999 999.999 999 999 [m/s^2]
Default: 0.03
Access: LEVEL3
Reaction: RUN

Limiting the compensation

Parameter **MP_compLimitFactor** (400114) is used to limit reversal-spike compensation. As a precaution, the effective internal software limit is 3 mm/s. You can use **MP_compLimitFactor** to raise or lower the limit. To do this, add the optional parameter to the machine configuration.

MP_compLimitFactor

Limit of reversal-peak compensation
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0.5 to 5 with up to 9 decimal places
Factor for raising or lowering the internal default software limit of 3 mm/s.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

Compensation of reversal peaks

For adjusting the compensation of the reversal peaks, proceed as follows:

- Set machine parameters **MP_compFFAdjust** = 0 and **MP_compRefAcc** = 0 to ensure that the compensation strength is constant over all feed-rate values.

Now select a typical machining speed and adjust the **MP_compStrength, MP_compWidth** and **MP_compTimeOffset** parameters for the selected speed.

MP_compWidth specifies the duration of compensation and should contain values in the range of a few microns.

- In **MP_compStrength** specify the distance (in [mm]) which the axis will travel if it ideally complies with the transferred nominal speed value. The reversal peak height resulting without compensation is a suitable starting value for the compensation.
- In **MP_compWidth,** enter the duration of compensation. The value entered should lie in the range of a few microns.
- Set the parameter **MP_compTimeOffset** = 0.
- Run a circular interpolation test.
- If the path traversed in the circular test deviates at the reversal point from the nominal path, first toward the inside and then toward the outside, the compensation is performed too early. In this case, you must increase the value in **MP_compTimeOffset.**

- If the path traversed deviates at the reversal point from the nominal path, first toward the outside and then toward the inside, the compensation is performed too late. In this case, you must decrease the value in **MP_compTimeOffset.**

With the following formula you can estimate the magnitude of useful values for the **MP_compTimeOffset** parameter:

$$\textbf{\textit{compTimeOffset}} = \sqrt{2 \cdot [Acceleration] \cdot \textbf{\textit{compWidth}}} \cdot 6$$

It is easier to adjust the **MP_compWidth** and **MP_compTimeOffset** parameters at low feed rates. HEIDENHAIN recommends:

- First roughly adjust **MP_compStrength.**
- Then select a low feed rate.
- Then set **MP_compWidth** and **MP_compTimeOffset.**
- Return to the original feed rate and optimize the value for **MP_compStrength.**

You use these three parameters **(MP_compStrength, MP_compWidth, MP_compTimeOffset)** to adjust the compensation of reversal peaks for a specific feed rate and radius. Under certain circumstances, however, it can be necessary to correct the compensation strength based on the feed rate. Proceed as follows:

- Switch to the **Oscilloscope** mode of operation.
- Set the following value in the oscilloscope by pressing the **SELECTION** soft key:
  Display mode: YT
  Sampling time: IPO clock
  Channel 1: A nom
  Trigger: Free run
- Press the **OSCI** soft key to switch to the curve representation.
- Now get the nominal acceleration for the axis **(A nom)** at the reversal point for the feed rate that you have selected for the adjustment up to now.
- Press the **START** soft key to start recording.
- Press the axis-direction key of each axis.
- Press the **STOP** soft key to stop recording.
- Enter the determined acceleration in the parameter **MP_compRefAcc**.

Note

The oscilloscope shows mm/s^2, but the unit of the parameter is m/s^2. This means that you must divide the value by 1000.

- Now use the parameter **MP_compFFAdjust** to correct the compensation strength for lower or higher feed rates.

Analog offset

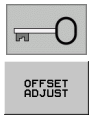
The offset voltage required for analog axes is stored in **MP_analogOffset.** You either enter these values manually or determine them by using the **offset adjustment** function.

During the offset adjustment, the control receives the axis offset values determined by the IPO and enters them in the parameters **MP_analogOffset.**

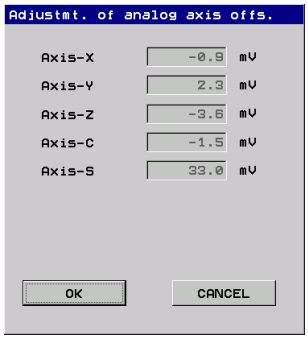
For the offset adjustment, the axes must be in position feedback control.

To adjust the offset:

- Switch to the **Organization** mode of operation.
- Press soft key and enter the key number 75368
- Press the **ADJUST OFFSET** soft key.



The control opens the **Adjustment of analog axis offset** dialog box and displays the values determined.



- Press the **OK** button to transfer the offset values to the parameters **MP_analogOffset.**

The maximum permissible offset voltage in the control is +/– 1 V. If this voltage is exceeded, the **offset adjustment** function limits the value.

MP_analogOffset

Offset on analog axis
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –1 to +1 [V]
Default: 0 [V]
Access: LEVEL3
Reaction: RUN

MP_noOffsetAdjust

Excludes the axis from automatic offset adjustment
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Axis will be excluded from offset adjustment.
FALSE
Offset of the axis is adjusted.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

Position loop resolution for analog axes

The analog voltage is subdivided 65536-fold with a 16-bit D/A converter. This results in a smallest voltage step of 0.15 mV.
This results in the voltage delta U per position error or following error s_a:

The control outputs one voltage per position error.

$$\Delta U = \frac{10\ 000\ [mV]}{S_a[\mu m]}$$

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

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgPowerStage **ampVoltProtection**	 401205

The parameter object CfgPowerStage is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Analog axes (**MP_axisHw** = Analog)

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

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

Attention

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

For synchronous motors, the operation with a weakened field is automatically activated if **MP_ampVoltProtection=installed** or **special installed** has been entered. No further settings must be made.

- Enter MP_ampVoltProtection=not installed if you want to operate synchronous motors **without** a weakened field.
- Enter MP_ampVoltProtection=installed if you want to operate synchronous motors with a weakened field **and** use voltage protection modules.

The EcoDyn synchronous motors from HEIDENHAIN are operated with a limited field weakening. No voltage protection module is necessary here.

- Select from the motor table the motors with the designation **`QSY1xxx EcoDyn`** or **`QSY1xxx EcoDyn EnDat`** for MP_motName.
- Enter MP_ampVoltProtection=special installed if you are using **EcoDyn synchronous motors from HEIDENHAIN.**

MP_ampVoltProtection

Field weakening for synchronous motors
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **not installed**
No voltage-protection module
installed
Voltage-protection module present
special installed
Limited field weakening without voltage protection module for EcoDyn motors

Default: Not installed

Access: LEVEL3

Reaction: RESET

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgServoMotor **starDelta**	 401302

The parameter object CfgServoMotor is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Analog axes (**MP_axisMode** = Analog)

Define different parameter sets if a wye/delta connection switchover is carried out for an asynchronous motor. Use the Modules 9434 and 9435 to activate the parameter set for operation with wye connection or operation with delta connection (see "Switching Parameter Sets" on page 306).

For the two operating modes, you can use different machine parameters for the current and speed controllers:

The switchover can be carried out during standstill or with a revolving spindle.

Danger

The contactor for the wye/delta switchover must not be switched under load!

With Module 9173, a monitoring function dependent on the speed can be realized for the wye/delta switchover of the spindle. You can interrogate the current spindle status in reference to the wye/delta switchover with Module 9174.

Performing a wye/delta connection switchover:

- Switch the drive controller off by using **PP_AxDriveOnRequest.**
- Use Module 9434 to select the corresponding parameter set.
- Use Module 9435 to check if this parameter set is active.
- Switch the drive controller back on by using **PP_AxDriveOnRequest.**

MP_starDelta

Motor with wye/delta switchover
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **star-connection**
Wye connection
delta-connection
Delta connection
Default: star-connection
Access: LEVEL3
Reaction: RESET

Module 9173 Speed-dependent monitoring of the wye/delta switchover
Module 9173 is used to realize a monitoring function dependent on the speed for the wye/delta switchover of the spindle.

You can use Module 9174 to determine a requirement for switching. Module 9174 supplies the current status for wye/delta operation.

Do not call the module cyclically. A single call is enough for activation, deactivation or changing the setting.

Call:

PS	B/W/D/K	<Mode> Bit 0 = 0: Monitoring off Bit 0 = 1: Monitoring on Useful values for <Mode> include: xx0 = Deactivate monitoring 001 = Monitoring on, Switchover compatible
PS	B/W/D/K	<Switchover speed from wye to delta operation>
PS	B/W/D/K	<Switchover speed from delta to wye operation>
CM	9173	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Speed-dependent monitoring activated
	1	Faulty call parameters, see error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Invalid shaft speeds were given (e.g. wye switchover speed >= delta switchover speed, or shaft speed negative)

Module 9174 Read current spindle status
With Module 9174 you read a the current spindle status in reference to wye/delta operation.

In order to receive one of the two "Request switchover" status values, you must activate the speed-dependent wye/delta switchover via Module 9173.

Call:

CM	9174	
PL	B/W/D	<Status> 0: Spindle in wye operation 1: Request for wye/delta switchover 2: Spindle in delta operation 3: Request for delta/wye switchover

Note

This function is only available when using the CC 61xx, UEC 11x or CC 424 controller unit!

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgServoMotor	
motSpeedSwitchOver	401318
motSpeedSwitchBack	401319

This function is used with high-speed spindle drives. This switchover is only possible for double-speed control loops. (Software option)

In **MP_motSpeedSwitchOver** and **MP_motSpeedSwitchBack**, a speed-dependent hysteresis for switching the PWM frequency is specified. It only takes effect if the value in **MP_motSpeedSwitchBack** is less than the value in **MP_motSpeedSwitchOver**.

This function is associated with **MP_iCtrlPwmType** and **MP_iCtrlPwmInfo**. Only if **MP_iCtrlPwmType** = 2, and **MP_ampPwmFreq** ≤= 5 kHz, does the switching of the PWM frequency take effect. Please note that the adjustment of the current controller (**MP_iCtrlPropGain**, **MP_iCtrlIntGain**) is based on the lower PWM frequency ≤5 kHz. Adaptation of the current-controller parameters and consideration of the power-module derating are performed automatically.

Using this function provides several benefits:

- At lower speeds and therefore a lower PWM frequency, the power module provides a comparatively high current. This results in a relatively high maximum motor torque.
- On the one hand, losses due to harmonics in the motor become more important as the speed increases, and on the other hand, the relationship between the electrical frequency and the PWM frequency worsens. These two disadvantages can be counteracted by increasing the PWM frequency. The resulting reduction of the current normally is insignificant, since in part due to motor characteristics, very high motor currents are mostly no longer possible or needed at higher speeds.

Note

The speed-dependent switching of the PWM frequency is only permitted with power modules from HEIDENHAIN.

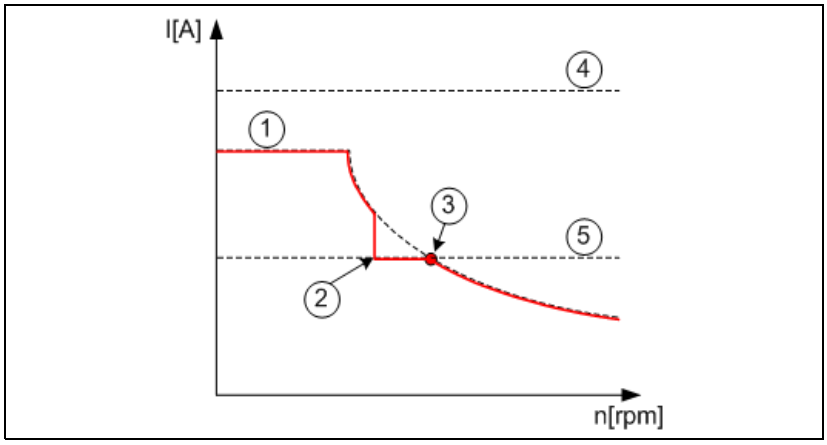
Danger

Speed-dependent switching of the PWM frequency with non-HEIDENHAIN power modules can lead to malfunctions, and possibly to damage of the power modules. Therefore, only use this function with power modules from HEIDENHAIN.

When determining the optimum switching speed for the PWM frequency, you should consider that the maximum motor current decreases as the speed rises, due to the finite dc-link voltage.

The current for the drive depends on the maximum permissible motor current and power-module current. The smaller of the two values limits the current for the drive. The value of the maximum power-module current is reduced by approx. 30% because of the derating when doubling the PWM frequency.

The optimum switching speed results from the intersection of the maximum current curve of the motor with the maximum current curve of the power module for the high PWM frequency. You can determine the maximum current curve of the motor spindle by using the TNCscopeNT software. Record the current Inoml/S in dependency of the speed. The spindle must be accelerated to the maximum speed so that the derating behavior can be seen in the curve. The figure shows the behavior when the values entered for the switching speed (**MP_motSpeedSwitchOver** and **MP_motSpeedSwitchBack**) are too low. This then results in a speed range where the current for the motor is less than the permitted and maximum current, resulting in inconsistencies in the motor's torque behavior.



- 1: Red line: Maximum current for the drive, resulting from the entries in **MP_motSpeedSwitchOver** and **MP_motSpeedSwitchBack**.
- 1: Broken line:
  Maximum possible current for the drive (Imax motor)
- 2: Switching point for the PWM frequency set too low
- 3: Optimum switching point for the PWM frequency
- 4: Maximum power-module current at low PWM frequency
- 5: Maximum power-module current at high PWM frequency
  (Imax of power module)

Summary:

- The speed is switched at the intersection of the two current curves (Imax of motor, Imax of power module) so that no inconsistencies in the torque behavior of the motor occur.
- For better controllability (no harmonics at higher PWM frequencies), it might already make sense to switch at lower speeds.
- The best speed to switch at must be determined by experimenting. The value above should serve as an initial value.

Overcurrent switch-off

The momentary current is monitored by the CC, and the following errors now lead to an immediate cutoff:

- Excessive actual current: If the actual current exceeds the maximum permissible current (depending on the power module and motor), it triggers the error 0xC3C0 (Motor current too high).
- Excessive current offset: If the offset current exceeds the permissible threshold (50% of the maximum current), it triggers the error 0xC600 (Current offset too high). One possible reason could be a phase to ground fault.

MP_motSpeedSwitchOver

Shaft speed for PWM switchover
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0 to 100 000 [rpm]
Specifies the shaft speed at which the PWM frequency is switched to twice the PWM frequency. Use only in combination with HEIDENHAIN power stages!
Function only available in combination with CC 6106 or CC 424.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_motSpeedSwitchBack

Shaft speed for PWM switchover
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0 to 100 000 [rpm]
Specifies the shaft speed at which the factor 2 PWM frequency is switched to a factor 1. Use only in combination with HEIDENHAIN power stages!
Function only available in combination with CC 6106 or CC 424.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

General information

Note

This function is only available when using the CC 61xx, UEC 11x or CC 424 controller unit!

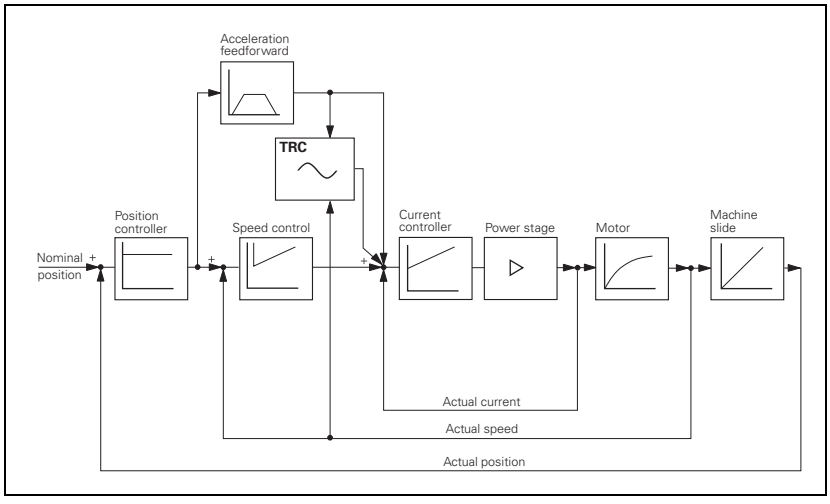
Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgControllerComp	
compTorqueRipple	401412
compSwitchOff	401414

Certain motors with permanent magnets (linear, torque and some synchronous motors) have an increased, position-dependent variation of the motor torque (not QSY motors from HEIDENHAIN). This can be the result of two things:

- During idle running, the cogging due to attractive forces of the permanent magnets
- When under load, the torque ripple from higher harmonics, resulting from the opposing electromotive forces (generator effect)

In practice, both causes always occur together, i.e. the torque of the motor is subject to periodic oscillations. This can have a negative effect on the controllability of the motor, which can result in a greater following error, and under circumstances, lower surface quality of the workpiece.

To compensate for the **cogging,** a compensation current ascertained specifically for each motor can now be added.



Activating TRC

TRC can only be activated via a special compensation file. The settings in this file can only be made with the TNCopt commissioning software from HEIDENHAIN (as of version 2.3). Please refer to the documentation for the TNCopt software. The compensation current is determined with a special method for measurement, and the parameters for calculating this compensation are stored in a compensation file on the control. The MANUALplus 620 then takes these parameters into account when calculating the controller parameters.

- Directory: The MANUALplus 620 saves the compensation files in the directory configured under
  Paths / CfgOemPath / **MP_oemTable**.
  Default: **%OEM%\table**
- File name: **xx_<Motor_name>.TRC**
  - **xx**: Index of the axis (e.g. 00 = 1st axis, X axis)
  - **<Motor_name>**: Name of the motor from the motor table (max. 29 characters)
  - **.TRC**: File extension for "Torque Ripple Compensation"

An entry in **MP_compTorqueRipple** specifies whether torque ripple compensation is to be performed for an axis. If the optional parameter **MP_compTorqueRipple** is not part of the configuration, no compensation current is calculated for the axis concerned.

Example:

```
MP_compTorqueRipple:      00_MotNameAusMotTab
```

Note

- The TRC function can only be used with PWM frequencies up to 5 kHz.
- A TRC file can only be used on the control on which the adjustment has been made.
- A TRC file must be re-created if the motor or even the encoder is exchanged.
- A TRC file can only be generated for synchronous motors or for linear or torque motors.

Switching off TRC

The TRC can be switched off by setting the machine parameter **MP_compSwitchOff**, bit 0.

MP_compTorqueRipple

Name of the file for "torque ripple compensation" (TRC)
Available from NCK software version: 597 110-03.

Format: String
Input: xx_<MotorNamefromMotorTable>.TRC (generated in TNCopt)
No entry: No compensation
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

MP_compSwitchOff

Switch-off of compensation in speed controller or current controller

Format: String
Input: Binary value
Bit 0: Torque ripple compensation
0: Torque ripple compensation is switched on
1: Torque ripple compensation is switched off
Default: %00000000000000000000000000000000
Access: LEVEL3
Reaction: RUN

Note

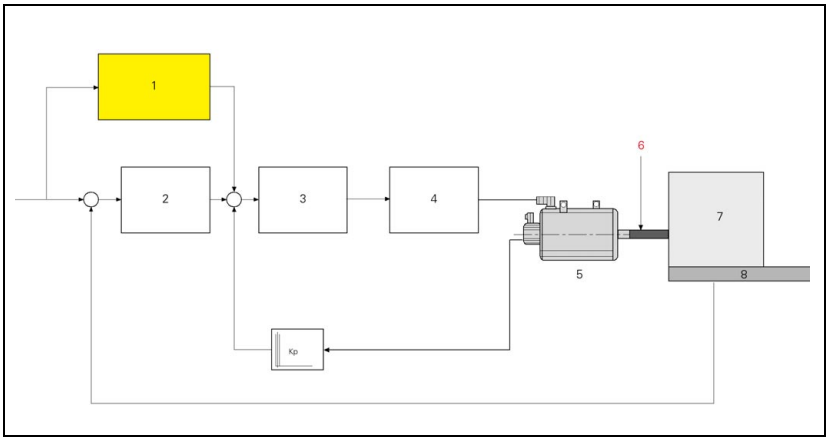
This function is only available when using the CC 61xx, UEC 11x or CC 424 controller unit!

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgControllerComp **compTorsionFact**	401413

With **MP_compTorsionFact** you can perform a torsion compensation between the position and speed measuring systems.

The torsion compensation regulates the difference in position that results from the elasticity between the motor (rotary encoder) and the position measuring system. An additional torsion motion is added to the speed controller.

Enter a specific drive elasticity factor in **MP_compTorsionFact**.



- 1: Torsion compensation
- 2: Position controller
- 3: Speed controller
- 4: Current controller – power module
- 5: Motor
- 6: Elastic coupling
- 7: Machine
- 8: Linear encoder

MP_compTorsionFact

	Torsion compensation between position encoder and speed encoder Available from NCK software version: 597 110-03.
Format:	Numerical value
Input:	0.001 to 30.000 [µm/A] 0: Not active
Default:	No value, parameter optional (= 0)
Access:	LEVEL3
Reaction:	RUN

6.14 Monitoring Functions

6.14.1 Monitoring the drives

Settings in the configuration editor	MP number
System CfgHardware **I32stopsMonitoring**	 100102

The NC monitors the dynamic response of the machine by using the following monitoring functions:

- Position monitoring
- Standstill monitoring
- Movement monitoring

If the specified values are exceeded, it displays an error message and stops the machine.

You can switch off the monitoring functions for individual axes or for all axes (globally) if drive enabling is canceled (I32 = 0).

Attention

Safe machine operation is not possible if the monitoring functions are switched off. Uncontrolled axis movements are not detected.

Switching off monitoring functions globally

The monitoring functions for all drives are switched off if I32 = 0 and **MP_I32stopsMonitoring** = On:

MP_I32stopsMonitoring
Behavior of input I32 (drive enabling)
Available from NCK software version: 597 110-01.
Format: Selection menu
Selection: **on**
If I32 = 0, all monitoring functions that can be influenced by the PLC are switched off.
off
Input I32 has no effect on the monitoring functions.
Default: off
Access: LEVEL3
Reaction: RESET

Switching off monitoring functions for individual axes

Set **PP_AxDeactivateMonitoring** to switch off monitoring for individual axes.

PLC operand / Description	Type
PP_AxDeactivateMonitoring Deactivate monitoring functions 0: Monitoring functions active 1: Monitoring functions inactive	M

The following table shows the status of monitoring as a function of the axis-specific PLC operand **PP_AxDeactivateMonitoring** and drive enabling (I32) if **MP_I32stopsMonitoring** = On.

Monitoring functions for		PP_AxDeactivate Monitoring =	MP_I32stopsMonitoring = On; I32=
Individual drive motor	All drive motors		
Inactive	Inactive	0	0
Active	Active	0	1
Active	Inactive	1	0
Inactive	Active	1	1

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgPosControl	
servoLagMin1	400802
servoLagMax1	400803
servoLagMin2	400804
servoLagMax2	400805
CfgControllerAuxil	
driveOffLagMonitor	400601
maxPosDiff	400605
posDiffCountDir	400604
CfgReferencing	
endatDiff	400405

Note

The parameters **MP_maxPosDiff** and **MP_posDiffCountDir** are available only with digital drive control.

The parameter object

- CfgControllerAuxil is not required for:
  - Virtual axes **(MP_axisMode** = **`Virtual`)**
  - Axes that are only for display (**MP_axisMode** = **`Display`**)
- CfgReferencing is not required for:
  - Virtual axes **(MP_axisMode** = **`Virtual`)**

The axis positions are monitored by the MANUALplus 620 as long as the control loop is closed.

The input values for position monitoring depend on the maximum possible following error (servo lag). Therefore the input ranges for operation with following error and velocity feedforward are separate.

For both modes of operation there are two range limits for position monitoring.

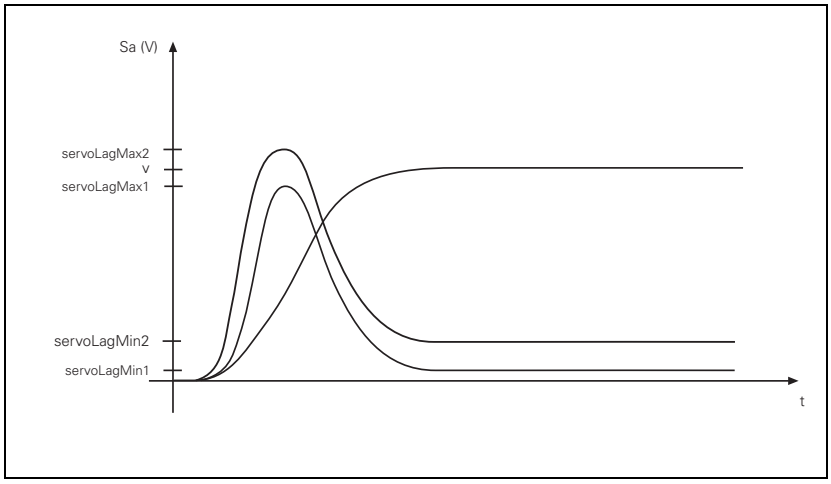
If the first limit **(MP_servoLagMin1, MP_servoLagMax1)** is exceeded, the warning **`Excessive servo lag in [axis]`** appears. The machine stops.

This message can be cleared. An actual-to-nominal value transfer is then executed for the respective axes.

If the second limit **(MP_servoLagMin2, MP_servoLagMax2)** is exceeded, the emergency-stop error message "Excessive servo lag in [axis]" appears.

The control-is-ready signal output is reset. The machine stops. You cannot clear this message. You must restart the control to correct the error.

- In the machine parameters given below, define two range limits for position monitoring.
- Adjust the input values to the machine dynamics.



MP_servoLagMin1 and MP_servoLagMin2 apply at a constant feed rate, whereas **MP_servoLagMax1** and MP_servoLagMax1 apply to changes in the feed rate.

MP_servoLagMin1

Minimum for following-error monitoring (clearable)
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 100 [mm] or [°]
Default: 1 [mm] or [°]
Access: LEVEL3
Reaction: RUN

MP_servoLagMax1

Maximum for following-error monitoring (clearable)
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 100 [mm] or [°]
Default: 5 [mm] or [°]
Access: LEVEL3
Reaction: RUN

MP_servoLagMin2

Minimum for following-error monitoring (emergency stop)
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 100 [mm] or [°]
Default: 1 [mm] or [°]
Access: LEVEL3
Reaction: RUN

MP_servoLagMax2

Maximum for following-error monitoring (emergency stop)
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 100 [mm] or [°]
Default: 5 [mm] or [°]
Access: LEVEL3
Reaction: RUN

Clamped axes, hanging axes

Clamped axes or hanging axes are monitored when the drive motor is switched off if **MP_driveOffLagMonitor** is activated. The value from **MP_servoLagMax2** is monitored.

MP_driveOffLagMonitor

Following-error monitoring with drive switched off
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **on**
Monitoring of hanging axes is active
off
Monitoring of hanging axes is not active
Default: off
Access: LEVEL3
Reaction: RUN

Note

The monitoring functions for hanging axes can **not** be switched off by using **MP_l32stopsMonitoring.**

Difference between position at switch-on and shutdown

When the MANUALplus 620 is switched off, the actual position of the axes is saved with an absolute encoder. During switch-on it is compared with the position values read by the encoder.

If the positions differ by more than the difference defined in MP_endatDiff, a pop-up window appears with both positions. The new position must be confirmed with a soft key. If it is not confirmed, the error message **231-8C50 Position input<axis> measured value not saved (EnDat)** appears.

The pop-up window may appear, stating that the positions at switch-on and shutdown differ by more than **MP_endatDiff**. If the motor is located at the correct position, you can confirm the message.

Note

The cause for one of the above listed messages can also be a defect in the encoder or control.

MP_endatDiff

	Permissible difference of EnDat encoders during switch-on Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	–100 000.000 000 000 to +100 000 [mm] or [°] 0: Monitoring is switched off
Default:	0
Access:	LEVEL3
Reaction:	REF

Determining the difference between speed and position encoder

If shaft-speed and position measuring systems are connected to a machine, then the MANUALplus 620 can ascertain and monitor deviations between the two systems. These differences can be displayed in the oscilloscope (**`PosDiff`** oscilloscope signal).

- Specify in **MP_maxPosDiff** the maximum permissible position difference in [mm] between the position and shaft-speed measuring systems.
- The **MP_maxPosDiff** parameter is optional. Deactivate the parameter in order to switch monitoring off.

MP_maxPosDiff

	Maximum position difference between position and shaft-speed measuring system Available from NCK software version: 597 110-03.
Format:	Numerical value
Input:	0.000 000 000 to 1 0000 000 [mm] 0: Monitoring is switched off
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RUN

In some situations it may occur that the interpolator cannot clearly recognize if the position or shaft-speed measuring system is counting in the opposite direction. That is why you can configure the counting direction of the motor encoder in the **MP_posDiffCountDir** parameter.

- Use the integrated oscilloscope (**`PosDiff`** signal) to inspect the position deviation between the shaft-speed and position measuring system.

If you determine via the oscilloscope that the actual position of the axis "wanders," then you must invert the parameter **MP_posDiffCountDir.**

MP_posDiffCountDir

	Counting direction of the motor encoder Available from NCK software version: 597 110-03.
Format:	Selection menu
Selection:	**positive** Motor encoder counts in positive direction **negative** Motor encoder counts in negative direction
Default:	positive
Access:	LEVEL3
Reaction:	REF

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgEncoderMonitor	
movementThreshold	400705
thresholdDistance	400706

Movement monitoring is possible during operation both with velocity feedforward and with following error. This requires, however, that position control occurs in the interpolator of the MC, and not in the CC.

If the **MP_movementThreshold** parameter is set to a value greater than 0, the manipulated value of the position controller is totaled in the IPO clock as soon as the threshold configured in the parameter is exceeded. In **MP_thresholdDistance** you enter the distance at which the comparison is to be performed, and so define the monitoring cycle.
The MANUALplus 620 now calculates a nominal path and compares it with the actual path traversed after every 5 mm.
An error message is output if the actual path traversed is…

- less than a quarter of the nominal path
  or
- four times greater than the nominal path.

If the encoder is defective, the axis moves no more than the distance defined in **MP_thresholdDistance**.

Intentional movements smaller than indicated in **MP_movementThreshold** are ignored.

- In **MP_movementThreshold,** enter the distance over time at which movement monitoring is to start.
- Enter in **MP_thresholdDistance** the distance at which the nominal and actual paths are to be checked.

Attention

- If **MP_movementThreshold = 0,** movement monitoring is not active.
- If you are using drive motors with digital interface, set **MP_movementThreshold** to 0 and use **MP_vCtrlFiltLowPassT** of the speed controller.

MP_movementThreshold

Threshold as of which movement monitoring becomes active
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 36 000 000 [mm/min] or [°/min]
0: Monitoring switched off
Default: 5 [mm/min] or [°/min]
Access: LEVEL3
Reaction: RUN

MP_thresholdDistance

Tolerance at and above which the following error is included
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 36 000 000 [mm] or [°]
Default: 5 [mm] or [°]
Access: LEVEL3
Reaction: RUN

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgControllerAuxil **checkPosStandstill**	 400602

Standstill monitoring is effective during operation both with velocity feedforward and with following error, as soon as the axes have reached the positioning window.

If the position difference is greater than the value defined in **MP_checkPosStandstill,** the error message **`Standstill monitoring in [axis]`** appears. The message also appears if, while moving to a position, an overshoot occurs that is larger than the value entered in **MP_checkPosStandstill,** or if the axis moves in the opposite direction when beginning a positioning movement.

Manually moving a spindle while the control loop is open (e.g. during a manual tool change) causes following error to build up. This following error could trigger activation of the standstill monitoring. Therefore, monitoring is only active in the following situations:

- If the axis is not a spindle and the parameter **MP_checkPosStandstill** > 0. Here it is not relevant whether the axis is in a control loop or not.
- If the affected axis is a spindle and the parameter **MP_checkPosStandstill** > 0, then the axis must be in a closed loop for standstill monitoring to be effective.

▶ Enter in **MP_checkPosStandstill** the threshold from which the standstill monitoring should go into effect:

MP_checkPosStandstill

	Standstill monitoring Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0: Monitoring switched off 0.001 000 000 to 10 000 [mm]
Default:	10 000 [mm]
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgControllerTol	
posTolerance	401101
timePosOK	401102

The parameter object CfgControllerTol is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)

If the axes have reached the positioning window after a movement, the status is shown in **NN_AxInPosition.** This also applies to the status after the machine control voltage is switched on.

The NC resets **NN_AxInPosition** as soon as you start a positioning movement or traverse the reference marks.

In the **Electronic Handwheel** mode of operation, **NN_AxInPosition** for the current handwheel axis is reset.

NN_AxInPosition is not set for contours that can be machined with constant surface speed.

Axes in position

The MANUALplus 620 reports "axis in position" (**NN_AxInPosition**), if

- the axis is stationary for the time **MP_timePosOK** within the positioning window **MP_posTolerance**
- no axis direction key for the axis is pressed.

After the position has been reached, the control begins running the next block. The position controller can correct a disturbance inside this window without activating the "Return to the Contour" function. **NN_AxInPosition** is reset as soon as there is a request to move an axis (NN_AxMotionRequest = 1).

- Specify in **MP_posTolerance** the size of the positioning window.
- Specify in **MP_timePosOK** how long the axis is to remain within the positioning window.

The control window is monitored during positioning as concerns various parameters. A following error is calculated from the control parameters, and compared with the actual following error. If the deviation is greater than the calculated value as well as the parameter value in **MP_posTolerance**, then the error message **[Axis] does not attain the control window** is output after 20 seconds.

This error message can be cleared, and traverse can continue.

MP_posTolerance

Positioning window
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.001 000 000 to 2 [mm]
Default: 0.005
Access: LEVEL3
Reaction: RUN

MP_timePosOK

Hysteresis time reached for positioning window
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 20 [s]
Default: 0.01 [s]
Access: LEVEL3
Reaction: RUN

PLC operand / Description	Type
NN_AxInPosition Axis in position 0: Axis not in positioning window 1: Axis in positioning window	M

Axis movement by interpolator

If the interpolator wants to move the axes, the markers **NN_AxMotionRequest** or **NN_SpiMotionRequest** are set. The PLC then switches on the drive, activates the control loop, releases the brake, and enables the feed rate or spindle.

PLC operand / Description	Type
NN_AxMotionRequest Axis movement by interpolator 0: No axis movement by interpolator 1: Axis movement by interpolator	M
NN_SpiMotionRequest Spindle movement by interpolator 0: No spindle movement by interpolator 1: Spindle movement by interpolator	M

Axes in motion

During an axis movement, the NC sets NN_AxInMotion.

PLC operand / Description	Type
NN_AxInMotion Axes in motion 0: Axis not in motion 1: Axis in motion	M

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgPowerStage **ampAcFailSelection**	 401207

The parameter object CfgPowerStage is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)
- Analog axes (MP_axisHw=Analog)

The rectified supply voltage of the power supply unit is monitored. The supply voltage must lie within a defined range
(400 V +/– 10%). If this is not the case the power supply unit reports an AC fail (PF.PS.AC).

At the same time, the dc-link voltage is monitored:

- If approx. 760 V– (UV 120, UV 140, UV 150, UR 2xx: approx. 800 V) is exceeded, the NC revokes the pulse release (reset) for the IGBT of the power module. The motors coast out of loop to a stop. No energy is returned to the dc link.
- If the dc-link voltage falls below approx. 385 V– (UV 120, UV 140, UV 150. UR 2xx: approx. 410 V), the power supply unit reports a powerfail (PF.PS.ZK signal)
- If the dc-link voltage falls below approx. 155 V– (UV 120, UV 140, UV 150, UR 2xx, UV 105: approx. 200 V), the control is reset (signal RES.PS).
- Below approx. 135 V– (UV 120, UV 140, UV 150, UR 2xx, UV 105: approx. 180 V), the power supply unit switches off.

The UV 105 power supply unit reports a powerfail if the dc-link voltage is < approx. 385 V and the supply voltage is < approx. 330 V.

▶ Define in **MP_ampAcFailSelection** which inverter signal is to trigger the **`Powerfail`** on the control.

Inverter signal	Meaning
AC fail (PF.PS.AC)	Failure of supply voltage for inverter
Powerfail (PF.PS.ZK)	DC-link voltage failure

Since the AC-fail signal is reported to the control before the powerfail, the MANUALplus 620 has more time to react to the subsequent dc-link voltage failure.

Note

Only specific HEIDENHAIN power supply units provide the AC-fail signal (see the Technical Manual for "Inverter Systems and Motors"). If you are using power supply units that do not provide this signal, you must not select the AC-fail signal in **MP_ampAcFailSelection.**

If a powerfail is triggered on the control, all drives are brought to a controlled stop. The PLC outputs are switched off and the control freezes to ensure that the hard disk can no longer be accessed.

The MANUALplus 620 must be turned off and on again.

- Define in **MP_ampAcFailSelection** whether the powerfail in the control is generated via the AC-fail and/or the powerfail signal of the power module.

MP_ampAcFailSelection

	Signal for power fail Available from NCK software version: 597 110-01.
Format:	Selection menu
Selection:	**AC fail only generated** Only AC-fail signal **power fail and AC fail generated** Powerfail and AC-fail signals **AC fail / power fail inactive** AC fail and powerfail deactivated **power fail only generated** Only powerfail signal
Default:	AC fail only generated
Access:	LEVEL3
Reaction:	RESET

Module 9167 Monitoring of dc-link voltage

With this module you can switch the dc-link voltage monitoring for powerfail (U_Z <approx. 385 V or 410 V) on and off.

If you don't call the module during the first PLC scan, the supply voltage monitoring is automatically started after the first PLC scan.

Call:

PS	B/W/D/K	<Command code> 0: DC-link voltage monitoring for >385 V– off 1: DC-link voltage monitoring for >385 V– on
CM	9167	
PL	B/W/D	<Error code> 0: Command executed –1: Transferred parameter invalid

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	DC-link voltage monitoring on or off
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Transferred parameter invalid

6.14.7 Temperature monitoring

Temperature of the MC

The internal temperature of the MC is continuously monitored. Beginning at about 60 °C the error message **`Temperature too high <temperature> °C`** appears and an emergency stop is triggered. If the temperature does not fall below 60 °C when the machine is switched on again, the error message reappears after 10 to 20 seconds.

Interrogate the values of the internal ADC

Module 9133 allows you to interrogate the internal values of the analog-to-digital converter of the MC.

Module 9133 Interrogate the values of the internal ADC

Call:

PS B/W/D/K <Code>
0: Internal temperature sensor in [°C]
1: Temperature CPU1 (basic PCB) in [°C]
2: Temperature CPU2 (additional PCB) in [°C]
3: Voltage of buffer battery in [mV]

CM 9133
PL B/W/D <Value>

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Value ascertained
	1	Value could not be determined; error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid code programmed
	8	No second processor present (for code 2)

The following warnings have not yet been integrated in the basic PLC program (including version V04-03):

- A PLC warning should already be issued starting from an operating temperature of 45 °C, because the permissible operating temperature is between 5 °C and 45 °C.
- If the battery voltage < 2.6 V, an NC error message is issued. Therefore, a PLC early warning should already be issued at a battery voltage < 2.7 V.

Motor temperature

To measure the motor temperature, a KTY 84 must be connected at pins 13 and 25 of X15 to X20, X80 to X83. The temperature value is ascertained at least once per second. The maximum permissible motor temperature is taken from the motor table.

As soon as the given temperature is exceeded, the error message: **231-8B30 MOTOR TEMPERATURE <AXIS> TOO HIGH** appears and the drives are automatically switched off.

Appropriate measures can be taken before the motor reaches the maximum temperature.

Module 9165 Read the temperature of the drive motor

The module provides the temperature of a drive motor controlled by the integral current controller in degrees Celsius.

Constraints:

- The measuring range is 0 to 255 °C.
- For non-existing axes the value 0 °C is read.
- The temperature value is measured every 5 seconds.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
CM	9165	
PL	B/W/D	<Temperature> Range: 0 to 255 °C

Error recognition:

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

Temperature of the power module's heat sink

At X51 to X60 the temperature warning signal is available at pin 10a.

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

Attention

To avoid destroying the power module, the drives must be brought to a standstill immediately after a temperature warning.

Data on maximum permissible temperatures are available from the manufacturer of your power supply unit.

The temperature warning signal is not evaluated in the NC.

- Use Module 9160 (recommended, See "Module 9160 Status request of temperature monitoring and I^2t monitoring" on page 667) or Module 9360 to interrogate the temperature warning, and take appropriate measures.

Module 9360 Monitor the temperature of the power modules

This module provides the temperature of the power modules. If the temperature is exceeded (ERR_TEMP signal at X51to X60), the axis-specific bits are reset.

Call:

CM	9360	
PL	D	<Temperature monitoring in bit code> Bit 15876543210 Axis:Sxxxxx987654321

Error recognition:

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

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgServoMotor	
motFactorI2t	401304
CfgPowerStage	
ampFactorI2t	401208

The parameter objects CfgServoMotor and CfgPowerStage are not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Analog axes (**MP_axisHw** = Analog)

General information

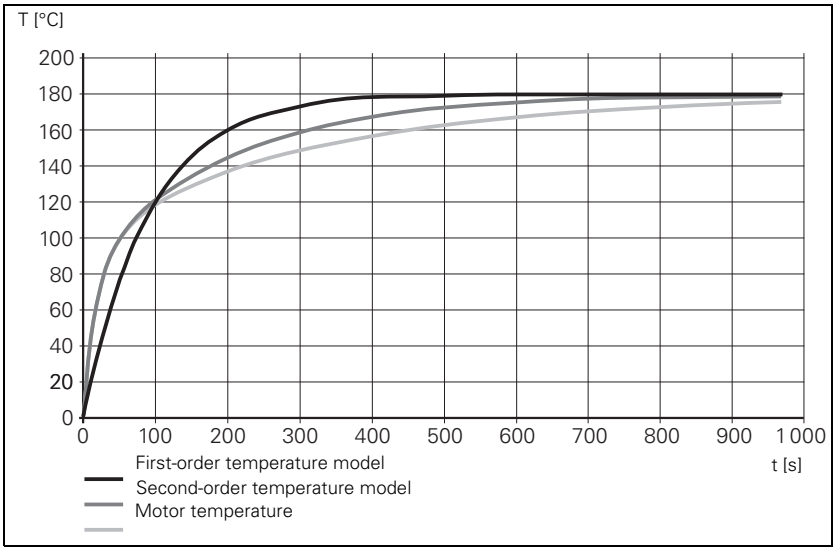
HEIDENHAIN inverter systems feature individual I^2t monitors, one for each power module and motor.

Function

An I^2t monitor calculates and supervises the temperature pattern in a thermal motor or power-stage model during operation, even if temperature measurement is also provided by the hardware.

The basis for calculation are the active current, the rated or stall current, (multiplied by MP_motFactorI2t for motors and by MP_ampFactorI2t for power modules) and a device-specific temperature model. A first-order temperature module is available for monitoring power modules, first and second-order modules are available for motors. These modules make it possible to permanently calculate the temperature of the stator winding in the motor or the semiconductor in the power module.

Temperature model in an example comparison (motor)



The I^2t monitor responds if this calculated temperature exceeds a certain limit.

Because temperature increase and heat dissipation are uneven when the motor is stationary or moving slowly, the I²t monitor distinguishes between standstill and traversing mode.

This limit range is defined in a motor table or power module table The following entries are important:

- F-AC **(transition frequency in traversing mode [Hz])**
- F-DC **(transition frequency at standstill [Hz]; only CC 424)**

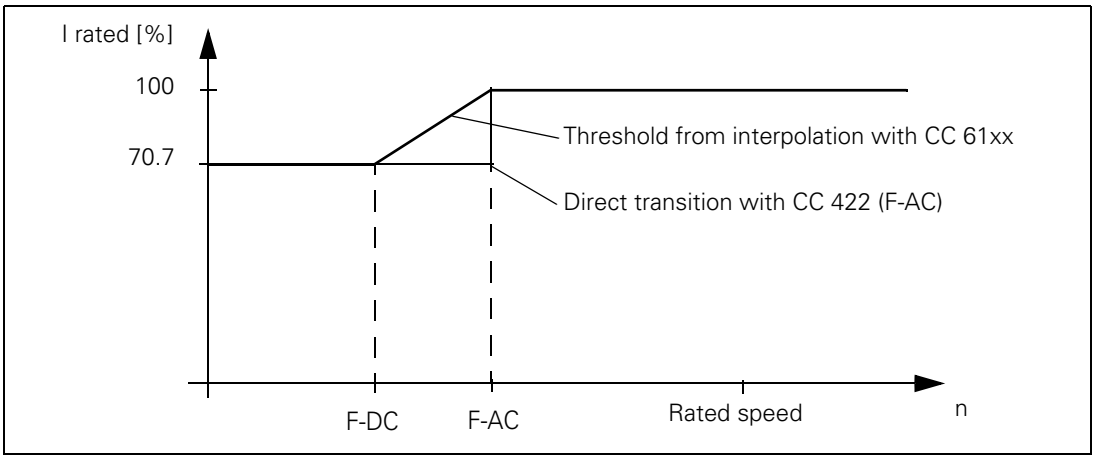
Fundamentals

The following graphics illustrate these parameters in relation to the reference voltage. Remember here that the parameters of the CC 422 may differ from those of the CC 61xx or CC 424.

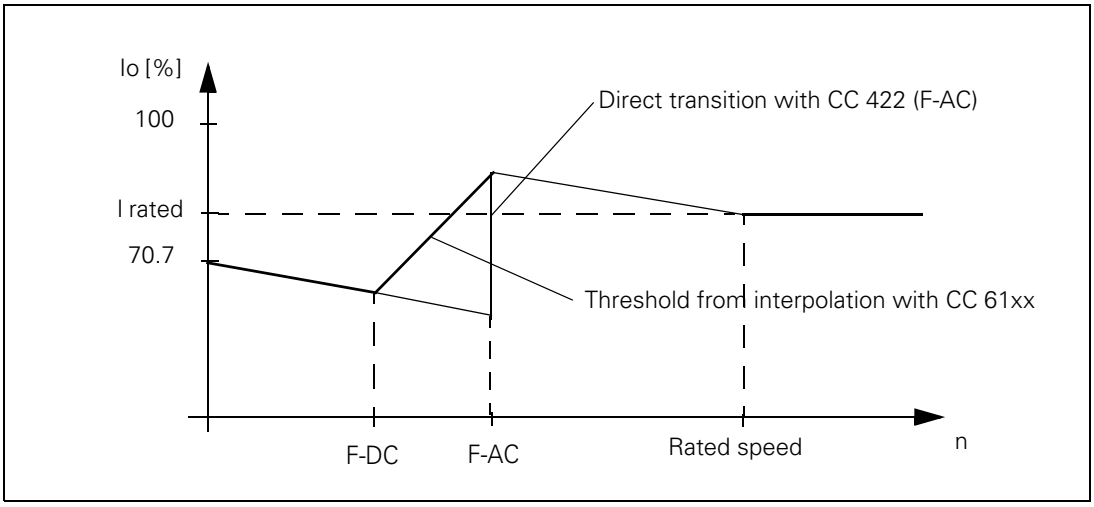
With the CC 61xx and CC 424 it is possible to use an interpolated current range for the transition from standstill to traverse. This allows a more exact calculation of the temperature model.

For the CC 422, no difference is made between F-DC and F-AC. Instead, F-AC is used as a rigid limit frequency for the transition between standstill and traversing mode.

If there is no stall torque value given in the motor table, the following model of current (with respect to the rated current) is used to calculate the temperature in the motor. The factors for MP_motFactorI2t and MP_ampFactorI2t are not yet taken into account.


I rated [%]
100
70.7
Threshold from interpolation with CC 61xx
Direct transition with CC 422 (F-AC)
F-DC
F-AC
Rated speed
n

If the stall current value is given in the motor table, the following model of current (with respect to the stall current) is used to calculate the temperature in the motor. This is only used for synchronous motors, however. For asynchronous motors the above model of current applies, which is used if no stall current is given. For synchronous motors, the factors from MP_motFactorI2t and MP_ampFactorI2t are not yet taken into account in the following description.



Commissioning and evaluation

- Enter in **MP_motFactorI2t** the factor for the I^2t monitoring of the motor. The input value is a factor for the reference current (1 = 100% of the motor's standstill current or rated current). If you enter zero, the I^2t monitoring for the motor (not for the power module) is switched off.
- Enter in **MP_ampFactorI2t** the factor for the I^2t monitoring of the power module. The input value is a factor of the power module's rated current (1 = 100%). If you enter zero, the I^2t monitoring for the power module (not for the motor) is switched off.
- All required entries for calculation of a temperature model have to be available in the motor table or power module table (See "Temperature models" on page 669).
- Use Module 9160 (recommended) or Module 9367 to interrogate the I^2t monitoring (See "Module 9160 Status request of temperature monitoring and I^2t monitoring" on page 667).

Limit values

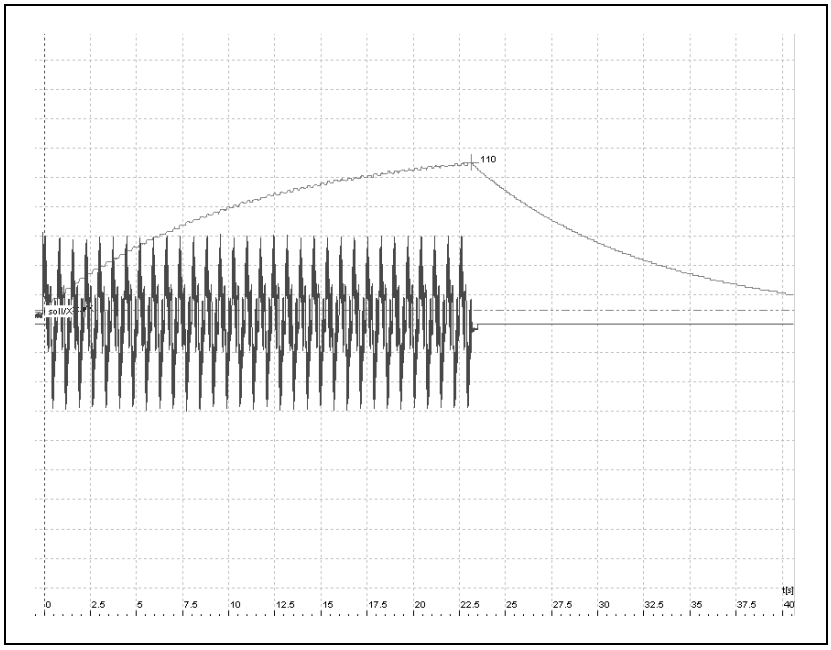
The limit values for the I^2t value (dimension for the permissible temperature in the device [%]) are handled by the NC side of the control and are composed of the following:

- Value exceeds 100%: An axis-specific I^2t early warning is sent to the PLC (for evaluation and possible countermeasures such as reduction of the feed rate with error message by PLC program with the aid of PLC Module 9367). If the value does not exceed 110% and falls below 90%, the axis-specific early warning is reset.
- 110%: An NC stop is triggered and the drives are switched off.

Note

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

Motor overload with I^2**t monitoring**



Machine parameters

MP_motFactorI2t

Reference value for I^2t monitoring of motor
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 000 000 to 1000 [· rated or stall current of the motor]
Default:	0
Access:	LEVEL3
Reaction:	RESET

MP_ampFactorI2t

Reference value for I^2t monitoring of the power module
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 000 000 to 1000 [· rated current of power module] 0: I^2t monitoring of the power module switched off 1: Rated current of power module is reference value
Default:	0
Access:	LEVEL3
Reaction:	RESET

Interrogation through PLC module

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

The I^2t monitoring reported by the module is given with respect to the first I^2t monitor response (power stage or motor) if both I^2t monitors are activated (MP_motFactorI2t and MP_ampFactorI2t). This early warning is withdrawn as soon as the limit for reset is reached. For the response behavior, See "Limit values" on page 666.

Constraints:

- Unconnected axes respond with all error bits as 0.
- The I^2t monitoring responds when the current value exceeds 110% of the rated current.
- After an e-function, the current is limited to 110% of I_{nomI}. The time constant for this is approx. 2.56 seconds.
- The response time for the I^2t monitoring at $2 \cdot I_{nomI}$ without preload is approx. 2.1 seconds. With a nominal load hysteresis every 10 seconds the monitoring responds after approx. 200 ms.
- The message "I^2t early warning" is set approx. 200 ms before limitation starts. The message "I^2t monitoring" appears when limitation is activated.
- Only the I^2t early warning is possible for the spindle axis.

Call:

CM	9160	
PL	D	<Temperature monitoring> Bit 15876543210 Axis:Sxxxxx987654321
PL	D	< I2t monitoring I²t early warning> Bit 15876543210 15 876543210 Axis Sxxxxx987654321 Sxxxxx987654321

Error recognition:

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

Module 9367 I^2t monitoring

The module reports

- an I^2t early warning
- an I^2t error (temperature exceeded)

The I^2t monitoring reported by the module is given with respect to the first I^2t monitor response (power stage or motor), if both I^2t monitors are activated (MP_motFactorI2t, MP_ampFactorI2t).

The I^2t early warning is withdrawn as soon as the limit for reset is reached. For the response behavior, See "Limit values" on page 666.

Constraints:

- Unconnected axes respond with all error bits as 0.
- The I^2t monitor responds if the current value exceeds 110% of the rated current.
- The "I^2t early warning" is set approx. 200 ms before the limit takes effect. The "I^2t monitoring" is set when the limit is reached.
- For the spindle axis, only the I^2t early warning is possible.

Call:

CM	9367	
PL	D	<I2t early warning, bit encoded> (Bit 0 represents logic axis 0, etc.)
PL	D	<I2t error, bit encoded> (Bit 0 represents logic axis 0, etc.)

Error recognition:

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

Temperature models

The temperature model of the motor or power module is defined by the entries in the motor table or power module table, respectively (motor.mot, motor.amp). Remember that calculation of which temperature model to use depends exclusively on the availability or nonavailability of the parameters. In addition, the parameters for motors and power modules are to be evaluated separately.

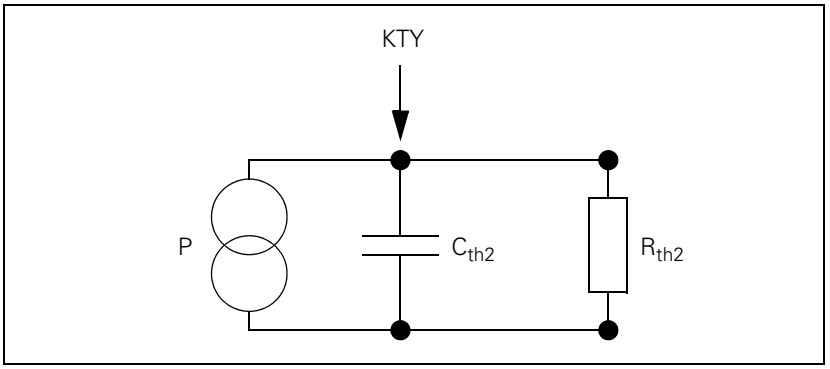
These parameters are explained below using the respective temperature model for the calculation.

Temperature model, first order

The following values (entries in the motor table or power module table) are required for the first-order temperature model to calculate the temperature.

- F-DC [Hz]:
  This parameter is not evaluated for the CC 422.
  Lower limit frequency for the transition of traverse to standstill with the CC 61xx and CC 424.
  F-DC = 0 – Default value (0) is active
  F-DC > 0 – Input value in Hz is active
- T-DC [s]:
  Thermal time constant for operation at standstill (not evaluated at present)
- F-AC [Hz]:
  Upper limit frequency for the transition from standstill to traverse.
  F-AC = 0 – Default value (0) is active
  F-AC > 0 – Input value in Hz is active
- T-AC [s]:
  Thermal time constant for the motor or power stage. Identifies the point in the temperature curve at which 63% of the maximum temperature is reached.
  T-AC = 0 – Default value: 10 s for axes, 150 s for ball screw
  T-AC > 0 – Input value [s] for power modules. For motors, this input value is active if $Tth_2 = 0$.
- Only for motors Tth_2 [s]:
  Thermal time constant for the motor. Identifies the point in the temperature curve at which 63% of the maximum temperature is reached.
  $Tth_2 = 0$ – Default value: 10 s for axes, 150 s for ball screw
  $Tth_2 > 0$ – Input value in [s] for motors

First-order temperature model of the motor



- P: Heat output of the three phases
- KTY: KTY temperature sensor in the winding
- C_{th2}: Thermal capacity of the motor housing
- R_{th2}: Thermal resistance on the motor housing
- T_{th2}: Thermal time constant $R_{th2} \cdot C_{th2}$

Temperature model, second order

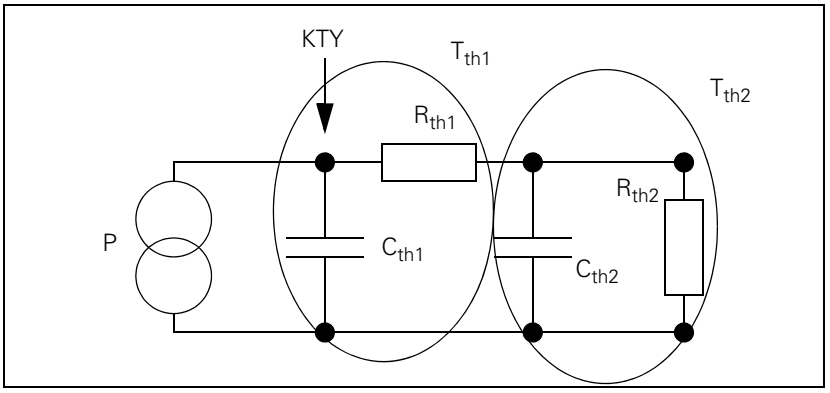
The following values (entries in the motor table or power module table) are required for the second-order temperature model to calculate the temperature (default values [axis/spindle] are valid for the entry "0"):

- F-DC [Hz]:
  This parameter is not evaluated for the CC 422.
  Lower limit frequency for the transition of traverse to standstill with the CC 61xx and CC 424.
  F-DC = 0 – Default value (0 Hz) is active
  F-DC > 0 – Input value in Hz is active
- T-DC [s]:
  Thermal time constant for operation at standstill (not evaluated at present)
- F-AC [Hz]:
  Upper limit frequency for the transition from standstill to traverse.
  F-AC = 0 – Default value (0 Hz) is active
  F-AC > 0 – Input value in Hz is active
- Tth_1 [s]:
  Thermal time constant for the transition from winding to housing
  Tth_1 = 0 – Default value (0 s) is active
  Tth_1 > 0 – Input value in [s] is active
- Rth_1 [K/W]
  Thermal resistance for the transition from winding to housing.
  Rth_1 = 0 – Default value: 0 K/W
  Rth_1 > 0 – Input value in [K/W] is active
- Tth_2 [s]:
  Thermal time constant for the transition from housing to coolant
  Tth_2 = 0 – Default value: 10 s for axes, 150 s for ball screw
  Tth_2 > 0 – Input value in [s] is active
- Rth_2 [K/W]:
  Thermal resistance for the transition from winding to coolant
  Rth_2 = 0 – Default value: 0 K/W
  Rth_2 > 0 – Input value in [K/W] is active
- When the CC starts up, the current motor temperature (KTY sensor) is taken into the calculation model in order, for example, to compensate any excessive temperatures.

Note

- All parameters have to be entered for the model to become active. If a parameter is missing, the first-order temperature model becomes active, either with the thermal time constant "Tth2" or with "T-AC."

Second-order temperature model of the motor



- P: Heat output of the three phases
- KTY: KTY temperature sensor in the winding
- C_{th1}: Thermal capacity of the winding
- C_{th2}: Thermal capacity of the housing
- R_{th1}: Thermal resistance winding/housing
- R_{th2}: Thermal resistance housing/coolant
- $T_{th1} = R_{th1} \cdot C_{th1}$
- $T_{th2} = R_{th2} \cdot C_{th2}$

Compatibility

Old motor tables are also usable in newer software versions. If the columns/ parameters in the temperature models are missing, however, it is of course impossible to calculate a second-order temperature model.

In such a case the entries **F-DC, T-DC, F-AC, T-AC** are used for a first-order temperature model. If this model, too, has no entries (entries "0"), the default values of the above temperature models apply.

6.14.9 Momentary utilization of drive motors

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

Utilization means:

Speed range	n_{actl} < rated speed	n_{actl} Š rated speed
Asynchronous motor	$\frac{\lvert M \rvert}{\lvert M_{Rated} \rvert}$	$\frac{\lvert P \rvert}{\lvert P_{Rated} \rvert}$
Synchronous motor	$\frac{\lvert M \rvert}{\lvert M_{Rated} \rvert}$	–

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

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

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

$$\text{Utilization} = 1000 \cdot \frac{I_{qMean}}{I_{qRated}}$$

For asynchronous motors:

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

I_N: Rated current of motor
I_{mag}: Magnetizing current

For synchronous motors:

I_{qRated} = <Rated current of motor>

The utilization display of synchronous motors is with respect to the rated torque (M/M_{rated}).

Module 9166 Read momentary utilization of drive motor

The module provides the momentary utilization of a drive motor controlled by the integral current controller in percent (%) of the rated utilization.

The value 0 is read for axes that are not connected.

Call:

PS	B/W/D/K	<Axis> Index from MP_CfgAxes/axisList
CM	9166	
PL	B/W/D	<Utilization of the drive in %>

Error recognition:

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

6.14.10 Status of HEIDENHAIN hardware and software

Module 9066 is used to determine the status information of the HEIDENHAIN components. With Module 9067 you can interrogate the status of software settings—at present with code 0 of the set SIK options.

Module 9066 Status of HEIDENHAIN hardware
Module 9066 interrogates the status information of HEIDENHAIN hardware components and the SIK component.

Constraints:

- The module can only be called in the cyclic PLC program.

Call:

PS B/W/D/K <Code for hardware component>
0: HEIDENHAIN inverter (only with digital control)
1: SIK ID
Note for users switching from the iTNC 530:
iTNC: SIK ID as BCD number
NCK: SIK ID as decimal number
2: I^2t early warning (only with digital control)
3: I^2t monitoring (only with digital control)
4: Error of analog part (only with digital control)

CM 9066

PL B/W/D <Status information>
Bit 0: Reserved
Bit 1: dc-link voltage too high
Bit 2: Heat sink temperature too high
Bit 3: Reserved
Bit 4: dc-link current too high
Bit 5: Power supply unit not ready
Bit 6: Leakage current too high
Code 1: SIK ID
Codes 2 to 4: Axis masks

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Status ascertained
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid code for HEIDENHAIN hardware
	24	Module was called in a spawn or submit job

Module 9067 Status of software settings

Module 9067 enables you to request status information about software settings. Module 9067 can currently be used to interrogate the software options set in the SIK.

Call:

PS	B/W/D/K	<Mode> 0: Interrogate whether SW option is set in the SIK
PS	B/W/D/K	<Number> If mode is 0: Number of SIK option
CM	9067	
PL	B/W/D	<Status> Status of SIK option (if mode is 0) 0: Not set 1: Set

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Function performed successfully
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Invalid value for number
	2	Invalid value for mode

Example:

Query if option 10 is set in the SIK:

```
PS      K0
PS      K10
CM      9067
PL      DL_statusinfo_9067
M_display_module_error(KG_error_module_9067)
```

Suppressing the status signals of the HEIDENHAIN supply unit

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgPowerStage **powStatusCheckOff**	 401209

The parameter object CfgPowerStage is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Analog axes (**MP_axisHw** = Analog)

The HEIDENHAIN power supply units have several status signals which lead to error messages on the control. **MP_powStatusCheckOff** is used to suppress the error message for each status signal.

HEIDENHAIN does not recommend suppressing the error messages from the power supply units. If you are using a UE 2xx, the signals must be suppressed because the UE 2xx compact inverter does not provide these signals.

The handling of status signals from HEIDENHAIN power supply units, which are already inactive during control start-up, varies depending on **MP_ampStatusCheckOff** – bit 0.

MP_powStatusCheckOff

- **Bit 0 = 0:** Missing signals do not result in an error message when the drive is switched on.
- **Bit 0 = 1:** Missing signals result in an error message when the drive is switched on. Signals that are not provided by the power supply unit must be suppressed with MP_powStatusCheckOff (bit 1 to bit 6), because non-existent signals are always identified as errors.

Note

Signals that change their status during operation are always identified as errors.

MP_powStatusCheckOff

Suppress error messages of the HEIDENHAIN supply units
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: **Bit 0** – Status signals that are already active during control power-up.
0: Missing signals are ignored
1: Missing signals are evaluated
Bit 1 – $\overline{\text{ERR.UZ.GR}}$ signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 2 – $\overline{\text{ERR.TMP}}$ signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 3 – Reserved
Bit 4 – $\overline{\text{ERR.IZ.GR}}$ signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 5 – RDY.PS signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 6 – $\overline{\text{ERR.ILEAK}}$ signal
0: Error message is not suppressed
1: Error message is suppressed
Bit 7 – Reserved

Default: 0

Access: LEVEL3

Reaction: RESET

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgSpeedControl **vCtrlSwitchOffDelay** CfgServoMotor **motEncCheckOff – Bit3**	 400927 401303

The parameter objects CfgSpeedControl and CfgServoMotor are not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Analog axes (**MP_axisHw** = Analog)

Controlling the motor brakes

The motor brakes are controlled with the $\overline{\text{BRK}}$ braking signal, which is transmitted to the **HEIDENHAIN inverters** via the PWM interface (X51 to X62). The corresponding outputs are activated there. See the basic circuit diagrams. Registered customers can download the current basic circuit diagram from the "HESIS-Web Including Filebase" on the Internet (http://hesis.heidenhain.de).

Control of the motor brakes via the PWM interface must be deactivated for non-HEIDENHAIN inverters that do not support this function.

The motor brakes are opened no later than 50 ms after the speed controller is switched on. For safety reasons, the controller is not switched off until the braking signal has been output:

- Enter in **MP_vCtrlSwitchOffDelay** the time (overlap time) after which the controller is to be switched off (after the braking signal has been output).

If the inverter sends the $\overline{\text{RES.PS}}$ reset signal, then the $\overline{\text{BRK}}$ braking signals are output immediately upon switch-off of the controllers, i.e. without any overlap time.

Activated brakes cause a change in the controlled system. The motor with the changed controlled system is controlled during the overlap time. This can lead to oscillations when the controller is switched off. These oscillations are suppressed by the NC software. **MP_motEncCheckOff** bit 3 can be used to deactivate the suppression of the oscillations. HEIDENHAIN does not recommend switching off the suppression of the oscillations.

MP_motEncCheckOff

Monitoring functions
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: **Bit 3** – Switching off the controller when the motor brakes are activated
0: Suppress oscillations
1: Oscillations are allowed
Default: 0
Access: LEVEL3
Reaction: RESET

MP_vCtrlSwitchOffDelay

Dwell time of speed controller during feedback control switch-off
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 0.5 [s]
Default: 0 [s]
Access: LEVEL3
Reaction: RUN

6.14.12 Emergency stop monitoring

Internal EMERGENCY STOP

If an **internal EMERGENCY STOP** is triggered (e.g. due to standstill monitoring), the MANUALplus 620 switches

- the control-is-ready output off (STO.A.G; responsible watchdog reacts after 10 ms at the latest)
- the inverter enablings off (STO.B.X; responsible watchdog reacts after the time set in **MP_watchdogTime** of 1 to 6 s), and the inverters are now without power.

An error message appears and the PLC program is stopped. Depending on the error class, it might be possible that it **cannot** be cleared with the CE key:

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

MP_watchdogTime

Delay for the SH1B signal (inverter enable)
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 1 to 6 [s]
Default: 3 [s]
Access: LEVEL3
Reaction: RESET

Testing an internal EMERGENCY STOP

For test purposes, an internal EMERGENCY STOP can be simulated in order to inspect the correct wiring of the machine. The control-is-ready output is reset, and the NC and PLC are no longer operable.

Attention

It is essential that you support hanging axes before the test in order to prevent damage to the machine in case of error.

- Under MOD, enter the code number **6871232**.
- Press the ENT key and the test will start.

After the test has been completed, you must restart MANUALplus 620.

External emergency stop

For the EMERGENCY STOP routine, controls with HSCI and without Functional Safety (FS) have a "control-is-ready" output X9/3a (MC.RDY or STO.A.G) and the two equal EMERGENCY STOP inputs ES.A (X9/7a) and ES.B (X9/7b). A "0" signal at ES.A or ES.B triggers an emergency stop reaction and the error message "external emergency stop." In addition **NN_GenNcEmergencyStop** is set.

The PLC inputs I3 and I32 known up to now in the HEIDENHAIN controls are being replaced in HSCI systems by ES.A and ES.B.

PLC operand / Description	Type
NN_GenNcEmergencyStop Control in "external EMERGENCY STOP" state 0: Control is not in external EMERGENCY STOP state 1: Control is in external EMERGENCY STOP state	M

Axis enabling

On controls with HSCI but without Functional Safety (FS), a pure axis enable, which up to now has been realized over I32, can only be realized through the function of the former X150. However, a CC 61xx and the UC 11x do not have an X150 connector for axis-specific or axis-group-specific drive controller enabling.

But in the **MP_driveOffGroupInput** machine parameter (100106), you can now enter eight numbers of those PLC inputs that simulate the inputs of X150. If one of these PLC inputs is switched off, the corresponding drives are decelerated on the provided braking ramp (usually at the limit of current). Over PLC module 9161 you have to switch off the axes of the affected axis groups. To do so, you can use PLC module 9157 to scan the corresponding PLC inputs.

The axes are assigned to individual axis groups through the axis-specific parameter **MP_driveOffGroup** (400015).

Emergency stop test

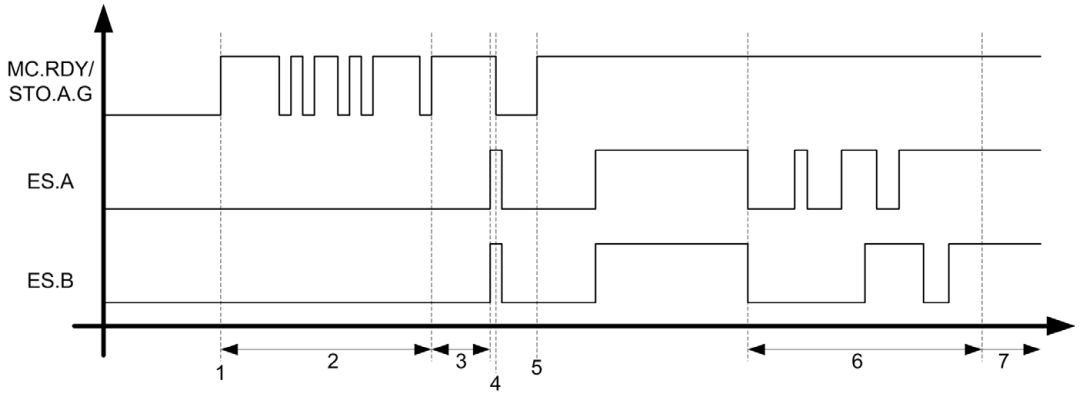
After the emergency stop test or a self-test is started, internal signals of the HSCI participants are tested for proper function. In a further, second phase the emergency stop test and brake test is conducted with external signals (ES.A, ES.B, STO.A.G).

In the event of an error, a drop-off of the control-is-ready output (MC.RDY or STO.A.G) always triggers an emergency stop.

Note

The circuitry recommended by HEIDENHAIN is illustrated in the Basic Circuit Diagram of the MANUALplus 620. Registered customers can download the current basic circuit diagram from the HEIDENHAIN FileBase on the Internet at http://filebase.heidenhain.de.

Time diagram of essential signals after the MANUALplus 620 is booted and during the self test:



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

The following error messages can occur during the test:

- **Timeout during self test**
  At least one HSCI participant has not answered a request or has not correctly detected a signal condition to be tested. Possible causes:
  - HSCI participant/device is defective
  - HSCI cabling is faulty
  - The hardware components used and/or software are not compatible with each other.
- **Error during self test**
  The sequence of the individual test steps and processes in the self test do not fulfill the requirements. A signal condition to be tested is not in the required initial condition. Possible causes:
  - HSCI participant/device is defective
  - HSCI cabling is faulty
  - The hardware components used and/or software are not compatible with each other.
- **Error in self test**
  Has same causes as **Error during self test**. However, after the cause of error is corrected (e.g. by closing the guard door), the test can be continued without the MANUALplus 620 having to restart.

Module 9144 Configuration of the EMERGENCY STOP test

With this module you can specify whether a repeated self test should be started immediately (mode 0) or by the user via soft key (mode 1). Before the repeated self test begins, the PLC program must switch off the drives.

Call:

PS B/W/D/K <Mode>
0: Start self test immediately
1: Prompt the user to start the self test

PS B/W/D/K <Parameter1>
0: No evaluation, but must be programmed
1: No evaluation, but must be programmed

PS B/W/D/K <Parameter2>
0: No evaluation, but must be programmed
1: No evaluation, but must be programmed

CM 9144

PL B/W/D <Error/Result>
Mode 0:0 = Function is run
1 = Error in NN_GenApiModuleErrorCode

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Function is being run
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid value for mode
	43	Control is not an HSCI system

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgServoMotor **motEncCheckOff – Bit4...8**	 401303

The parameter object CfgServoMotor is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Analog axes (**MP_axisHw** = Analog)

The following monitoring functions are only available when you are using the CC 61xx and CC 424. Use **MP_motEncCheckOff** to activate/deactivate these functions:

- **Bit 4, 5**: The KTY temperature sensor of the motors is monitored by the MANUALplus 620 for excessive and insufficient temperatures. If the KTY is not to be evaluated (e.g. because the temperature sensor is not doubly isolated), this function must be deactivated.
- **Bit 7:** The CC 61xx or CC 424 monitors the input frequency of the speed encoders. If this leads to problems (e.g. unjustified responses), the monitoring function can be deactivated. The following error messages can appear:
  - Speed encoder: **8860 Input frequency from speed encoder <axis>**
  - Position encoder: **8870 Input frequency from position encoder <axis>**
- **Bit 8:** Position encoders are not used with linear or torque motors. If such an axis is removed from the closed-loop control and later reintroduced, a mechanical offset can occur. This offset is not fixed "in one blow," but instead is adjusted by gradually raising the k_V factor from 0 to the original value.

MP_motEncCheckOff

Monitoring functions
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: **Bit 4:** – Monitoring for excessive motor temperature
0: Monitoring active
1: Monitoring inactive
Bit 5: – Monitoring for too low a motor temperature
0: Monitoring active
1: Monitoring inactive
Bit 7 – Monitoring the input frequency of the speed encoder
0: Monitoring active
1: Monitoring inactive
Bit 8 – Adjust mechanical offset by gradually increasing the k_V factor
0: Function active
1: Function inactive

Default: 0

Access: LEVEL3

Reaction: RESET

6.15.1 Configuring spindles

Settings in the configuration editor	MP number
System CfgAxes **spindleIndices**	 100002

The list index of a spindle key name defines the programmable spindle number used by the PLC to identify the spindle. Spindles are indicated by sequential numbering starting from the index [0].

The key names shown here in the selection menu of the MP_spindleIndices parameter are automatically taken from the **MP_CfgAxes/axisList** parameter. Select the key name(s) of the spindle(s) from the menu.

MP_spindleIndices

Key names of all spindles on the machine
Available from NCK software version: 597 110-02.

Format: Selection menu

Selection: The key names from MP_CfgAxes/axisList are automatically presented as a selection menu. Select the key name of the spindle.
The index number determines the logical spindle number.

Default: –

Access: LEVEL3

Reaction: RESET

The PLC indicates the number of configured spindles and the logical spindle number in the following PLC operands.

PLC operand / Description	Type
NN_GenSpiCount Number of configured spindles	D
NN_SpiLogNumber Logical axis number of the spindle. Equals the index of the spindle from CfgAxes > **MP_axisList** 0...n: Logical axis number –1: Spindle does not exist –2: Spindle deactivated (example: alternation between C axis and spindle)	D

6.15.2 Position encoder of the spindle

Analog and digital spindles can be driven in a closed position control loop. In this case the spindle needs its own position encoder, or you use the speed encoder to measure the position of a digital spindle.

Due to the higher required accuracy, the position encoder should be mounted directly on the spindle.

If the position encoder cannot be mounted to the spindle, the encoder will output several reference pulses per revolution. For example, with a transmission of 4:1 (motor to spindle), you will receive four reference pulses (every 90°) per spindle revolution.

- Evaluate the reference mark with Module 9220 (See "Renewed traversing of the reference marks" on page 520).

In **NN_SpiReferenceAvailable,** the NC reports whether the reference position of the spindle has been determined.

PLC operand / Description	Type
NN_SpiReferenceAvailable Reference position found 0: Reference position not found 1: Reference position found	M
PP_SpiReferenceMarkSignal Trip dog 0: Trip dog not triggered 1: Trip dog triggered	M

6.15.3 Speed encoder of the spindle

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgServoMotor	
motEncCheckOff – Bit0, 1	401303
CfgEncoderMonitor	
checkRefDistance	400704

The parameter object

- CfgServoMotor is not required for:
  - Virtual axes (**MP_axisMode** = Virtual)
  - Axes that are for display only (**MP_axisMode** = Display)
  - Analog axes (**MP_axisHw** = Analog)
- CfgEncoderMonitor is not required for:
  - Virtual axes (**MP_axisMode** = Virtual)

Digital speed control requires a speed encoder.

The MANUALplus 620 monitors the reference mark of the speed encoder. The monitor checks whether the line count for one revolution from reference mark to reference mark is equal to the line-count entry in the motor table. If differences occur, the DSP error message **C3A0 Incorrect reference position S** appears. If this happens, check the speed encoder, encoder cable, and whether you have selected the correct motor.

With a gear wheel encoder, even if it is properly installed, monitoring can result in this error message due to its inherent inaccuracy:

▶ In this case, switch the monitoring off with **MP_motEncCheckOff – Bit0 = 1**

The MANUALplus 620 monitors the direction of rotation. If the nominal value of current exceeds the limit value for a certain time, the DSP error message **C380 motor <spindle n, axis> not controllable** appears.

At lower speeds, high-frequency spindles only have a low amount of torque. If such a spindle is having its speed controlled, the tool changer may slightly twist the spindle, causing the limit of current to be exceeded. This leads to the above error message:

▶ In this case, switch the monitoring off with **MP_motEncCheckOff – Bit1 = 1**

MP_motEncCheckOff

Monitoring functions
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: **Bit 0** – Monitoring the reference mark
0: Monitoring active
1: Monitoring inactive
Bit 1 – Monitoring the direction of rotation
0: Monitoring active
1: Monitoring inactive

Default: 0

Access: LEVEL3

Reaction: RESET

Attention

For axes, monitoring of the rotational direction (**MP_motEncCheckOff** – Bit 1) must **not** be deactivated. An error (e.g. one motor phase interchanged with another or incorrect entry in the **DIR** column of the motor table) might cause uncontrolled acceleration of the motor in one direction if the monitoring function for the rotational direction is deactivated. This also applies to spindles. For spindles, however, an incorrect acceleration in one direction is less dangerous than for axes.

Monitoring of the direction of rotation (M**MP_motEncCheckOff** – Bit 1) for synchronous motors (entry **SM** in the column **TYPE** in the motor table) cannot be switched off.

- In **MP_checkRefDistance,** define the monitoring of the datum-to-datum distance of the spindle.

MP_checkRefDistance

Monitoring the reference mark of the spindle speed encoder
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **on**
Monitor the reference mark of the speed encoder
off
No monitoring

Default: off

Access: LEVEL3

Reaction: RESET

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgPositionFilter **manualFilterOrder**	 401605

The parameter object CfgPositionFilter is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)

The **MP_manualFilterOrder** parameter differentiates between axes and spindles. Linear acceleration is used for the operation of spindles. **MP_manualFilterOrder** allows you to filter the acceleration values.

MP_manualFilterOrder for axes: See "Filter before position control loop" on page 620

▶ Define the order of the mean-value filter for spindles in **MP_manualFilterOrder.**

MP_manualFilterOrder

Order of mean-value filter in Manual mode
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	1 to 51
Default:	11
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgControllerTol **speedTolerance** **timeSpeedOK**	 401103 401104

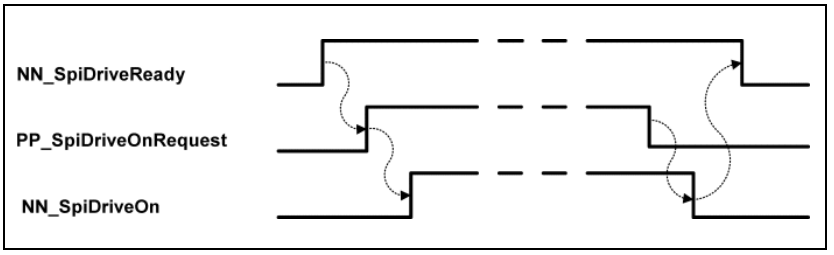
The parameter object **CfgControllerTol** is only required for spindles.

Also, the parameter object is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)

Switch on / switch off spindle drive

The following figure shows the procedure for switching the spindle drive on/off.



The NC or PLC store the information on controlling the spindle in the following PLC operands.

PLC operand / Description	Type
NN_SpiDriveReady Spindle drive is ready 0: Drive not ready for operation 1: Drive ready for operation	M
PP_SpiDriveOnRequest Switch spindle drive on. With this operand the control of the spindle is switched on or off. It can be switched on only in the ready condition (**NN_SpiDriveReady** = 1). 0: Do not activate the drive 1: Activate the drive	M
NN_SpiDriveOn Spindle drive is switched on (and is at least speed-controlled) 0: Drive is off 1: Drive is on	M

PLC operand / Description	Type
PP_SpiSpeedMax Maximum spindle speed	D
PP_SpiEnable Spindle enabling 0: Spindle not enabled 1: Spindle enabled	M
NN_SpiInMotion Spindle in motion 0: Spindle not in motion 1: Spindle in motion The operand is set if the nominal velocity of a spindle is not equal to 0, e.g. when - The spindle is controlled by the NC (tapping) - It is controlled by the PLC (9000 modules for M3/M4) The operand is not set (not even if the above conditions are fulfilled): - With spindle override = 0 - PLC has withdrawn the spindle enabling (PP_SpiEnable = 0)	M

Note

The PLC program offers the following ways to detect a real movement of the spindle:

- Find the actual speed of the spindle through Module 9411 (<desired spindle information> : 10)
- Connect "NN_SpiInMotion = 0" and "NN_SpiSpeedOK = 1". This attains the information that the spindle speed is less than 5 rpm.

Spindle control by PLC

The PLC controls the spindle by using the following modules:

- Module 9412: Stop the spindle
- Module 9413: Rotate the spindle
- Module 9414: Position the spindle (M19)

The status is queried by

- Module 9410: Read spindle status

Note

PP_SpiEnable = 1 must be set for a spindle movement to be executed.

In order to remain compatible with old PLC programs that use the numerical programming interface API 1.0, PLC Module 9171 is available as an alternative (See "Module 9171 Start of a spindle orientation with adjustable parameters" on page 697).

Module 9410 Read spindle status

The module reads the status of the specified spindle.

This module is only supported by the new symbolic memory interface. If the iTNC-compatible programming interface is used (API 1.0), the module returns an error.

Call:

PS	B/W/D/K	<Spindle index>
CM	9410	
PL	D	<Spindle status> 1: No job active – Last job was OK 2: No job active – Last job was faulty 3: Job is being executed
PL	D	<Spindle mode> 1: Spindle at standstill 2: Spindle turning clockwise 3: Spindle turning counterclockwise 4: Spindle is position-looped (M19)

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Status ascertained
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Invalid spindle number
	99	Module is not supported (control does not operate with symbolic memory interface)

Module 9412 Stop the spindle

Use Module 9412 to stop the specified spindle (M5 status).

This module is only supported by the new symbolic memory interface. If the iTNC-compatible interface (API 1.0) is used, the module returns an error.

Call:

PS	B/W/D/K	<Spindle index>
CM	9412	
PL	D	<Error code> 0: Module successfully executed 1: Incorrect module call 2: No permission for module call (example: NC is cutting a thread at the time it is instructed to execute a command)

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Spindle is stopped
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid spindle number
	9	Module call not possible at this time
	99	Module is not supported (control does not operate with symbolic memory interface)

Module 9413 Move the spindle

The module rotates the specified spindle in CW/CCW direction (M3/M4) at a constant rotational speed or at a constant cutting speed.

Constraints:

- This module is only supported by the new symbolic memory interface. If the iTNC-compatible interface is used (API 1.0), the module returns an error.
- With an analog spindle the voltage to be output for a reciprocation movement is read from the configuration data, and the shaft speed given is ignored. With a digitally controlled spindle the given shaft speed is interpreted as the motor shaft speed for the reciprocation movement.

Call:

PS	B/W/D/K	<Spindle index>
PS	B/W/D/K	<Mode> Bit 0 and bit 1: Direction of rotation 01 = Rotation CW (M3) 10 = Rotation CCW (M4) Bits 2 to 4: Type of rotation 001 = Constant surface speed 010 = Constant shaft speed 100 = Constant motor shaft speed for reciprocation movement
PS	D/K	<Rotational speed or cutting speed> Constant cutting speed in [m/min] Constant rotational speed in [rpm]
CM	9413	
PL	D	<Error code> 0: Module successfully executed 1: Incorrect module call 2: No permission for module call 3: Rotational speed not allowed 4: Mode not allowed

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Status ascertained
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Invalid spindle number
	9	Module call not possible at this time
	99	Module is not supported (control does not operate with symbolic memory interface)

Module 9171 Start of a spindle orientation with adjustable parameters
Module 9171 can be used to start an orientation of the spindle. The orientation speed, orientation angle and the direction of rotation can be set. The module sets M4130, which displays that the positioning is running and for how long.

Note

This PLC module was introduced in order to remain compatible with older PLC programs (with API version 1.0) of older HEIDENHAIN contouring controls. This module is **not** supported if the symbolic programming interface is used. Use Module 9414 instead.

Constraints:

- If no speed output has occurred for the spindle, the call will have no effect.
- If the marker M4130 is set in the same PLC scan and Module 9171 is called, the spindle is oriented with the parameters from the module call.
- If the module is called several times in the same scan, the spindle will be oriented with the parameters of the last call.
- If the module is called although an orientation from an earlier PLC scan is not yet finished, the call will have no effect.
- The module functions only in the cyclic PLC program.
- If the module is called while the spindle is turning, the direction of orientation that was transferred will be ignored. The spindle is always oriented in the direction of spindle rotation.
- If the values +2 to +4 are transferred as direction of rotation, the spindle can be oriented to the angle last defined in CYCL DEF 13. The transferred angle of orientation is added to the value from CYCL DEF 13. Therefore the PLC can transfer an additional spindle preset.
- If 0 is transferred as the speed, the speed from MP3520.1 is used.

■ The module is only supported for PLC programs that use API version 1.0.

Call:

PS B/W/D/K <Orientation angle [1/10000 degrees]>
or additional preset if there is a value from CYCLE DEF 13
PS B/W/D/K <Speed [1/1000 rpm]>
PS B/W/D/K <Direction of rotation>
–1: Negative direction (M04)
0: Direction of the shorter path
1: Positive direction (M03)
2: Same as –1, but angle from CYCLE DEF 13
3: Same as 0, but angle from CYCLE DEF 13
4: Same as +1, but angle from CYCLE DEF 13
CM 9171

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Spindle is oriented, M2712/M4130=1
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	The value for direction of rotation or rotational angle is invalid
	2	The spindle number or speed is incorrect, or there has been no speed output
	19	Spindle is not a closed-loop spindle
	24	Module was called from a spawn/submit job
	27	A spindle orientation is already running.

Tolerances for "rotational speed reached"

The MANUALplus 620 reports "Rotational speed reached" (**NN_SpiSpeedOK** = 1), if

- the actual speed has reached the tolerance band specified under **MP_speedTolerance** for the time from **MP_timeSpeedOK**,
- the nominal speed does not lie below the minimum speed of the current parameter set as specified in **MP_minFeed**,
- the spindle is no longer in the acceleration/braking ramp.

The programmed speed serves for comparison.

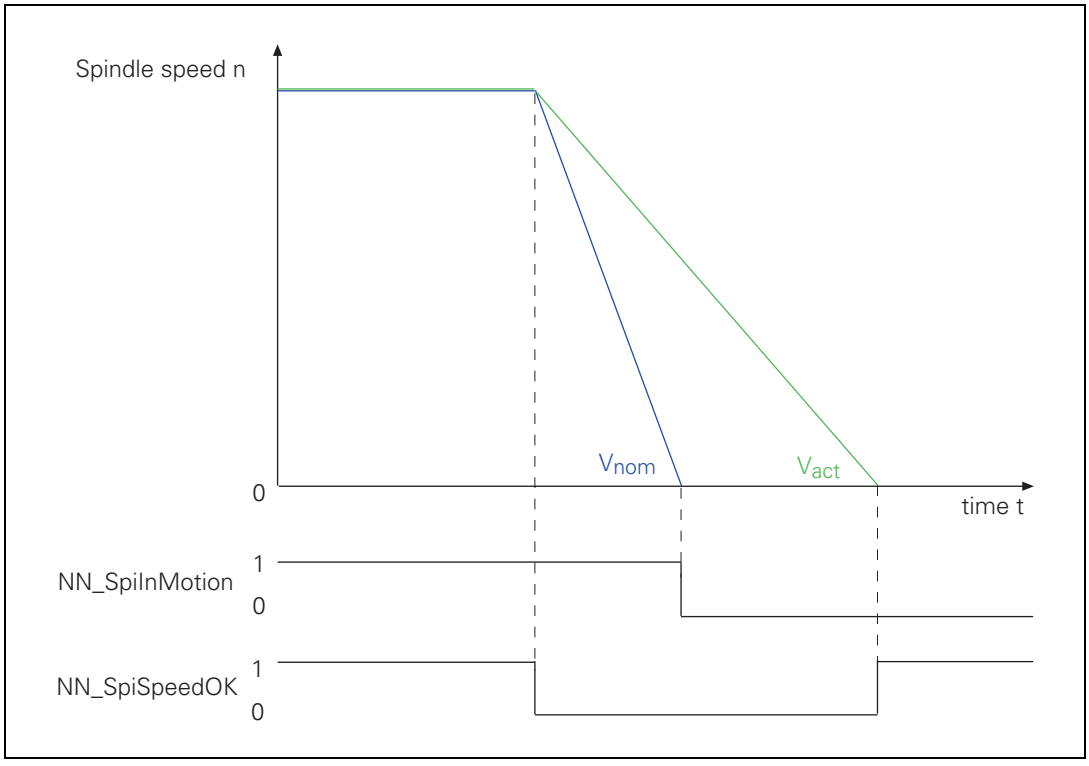
Special characteristics:

- If the programmed speed is not equal to 0 and **MP_speedTolerance** = 100%, the MANUALplus 620 sets **NN_SpiSpeedOK** = 1.
- If the spindle movement is to be stopped (programmed speed = 0), the MANUALplus 620 monitors the spindle for a speed n < 5 rpm.

▶ In **MP_speedTolerance**, define the size of the control window.

▶ In **MP_timeSpeedOK**, define the period of time which the rotational speed (feed rate) is to remain within the control window.

▶ In **MP_minFeed** of the respective spindle parameter set, define the minimum permissible speed for the spindle.



MP_speedTolerance

Rotational speed (feed rate) window
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 1000 [mm/min] or [°/min]
Default: 3 000 [mm/min] or [°/min]
Access: LEVEL3
Reaction: RUN

MP_timeSpeedOK

Hysteresis time for monitoring the speed deviation
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 20 [s]
Default: 0.01 [s]
Access: LEVEL3
Reaction: RUN

MP_minFeed

Applies only to spindles: minimum spindle speed
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 36 000 000 [mm/min] or [°/min]
Default: 0
Access: LEVEL3
Reaction: RUN

The NC provides the information "rotational speed reached" in **NN_SpiSpeedOK**.

PLC operand / Description	Type
NN_SpiSpeedOK Spindle speed reached 0: Spindle speed not reached 1: Spindle speed reached	M

Positioning the spindle (M19/trip dog position)

The PLC command defined in

- Module 9414 instructs the NC to activate spindle positioning.
- Module 9412 instructs the NC to deactivate spindle positioning.

Use Module 9414 to switch on position feedback control. Position feedback control is effective until it is switched off by Module 9412.

Module 9414 Position the spindle

The module is used for the following functions:

- Position the spindle (M19): The spindle is stopped at the specified position. The mode defines the direction of rotation.
- Stop the spindle at the trip dog position (mode bit 3=1): The spindle is positioned to the trip dog at the specified rotational speed (see "Stopping/ referencing the spindle at trip dog position" on page 708).

This module is only supported by the new symbolic memory interface. If the iTNC-compatible interface (API 1.0) is used, the module returns an error.

Call:

PS	B/W/D/K	<Spindle index>
PS	B/W/D/K	<Mode> Bits 0 to 2: Direction of rotation 000 = Shortest direction of rotation, absolute position entry 001 = Rotation CW, absolute position entry 010 = Rotation CCW, absolute position entry 100 = Relative position entry, rotational direction from algebraic sign of position entry Bit 3: Select the function 0 = "Position the spindle (M19)" function 1 = "Stop spindle at trip dog position" function
PS	D/K	<Absolute or relative position in [0.0001°]>
PS	D/K	<Rotational speed in [0.0001 rpm]>
CM	9414	
PL	D	<Error code> 0: Module successfully executed 1: Incorrect module call 2: No permission for module call 3: Rotational speed not allowed 4: Mode not allowed

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Status ascertained
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid spindle number
	9	Module call not possible at this time
	99	Module is not supported (control does not operate with symbolic memory interface)

The NC stores the status information on spindle positioning in the following PLC operands.

PLC operand / Description	Type
NN_SpiControl Spindle in closed loop 0: Spindle is not in position control loop 1: Spindle is in position control loop	M
NN_SpiControlInPos Spindle in position 0: Spindle is not in position 1: Spindle is in position	M

Tapping

The NC puts the spindle in the position control loop during tapping and thread cutting (Cycle 18 for TNC controls). Please be aware that the spindle is position-looped (NN_SpiControl = 1) even for tapping with a floating tap holder (Cycle 2). The job is not transmitted by the PLC.

The NC shows in the following PLC operands that a tapping operation is currently being executed (both markers are set simultaneously).

PLC operand / Description	Type
NN_SpiTapping Tapping active 0: Tapping not active 1: Tapping active	M
NN_SpiRigidTapping Tapping with spindle interpolated with Z axis active 0: Tapping not active 1: Tapping active	M

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgFeedLimits	
maxAccSpeedCtrl	400307
maxDecSpeedCtrl	400310
M19MaxSpeed	
M19NcSpeed	

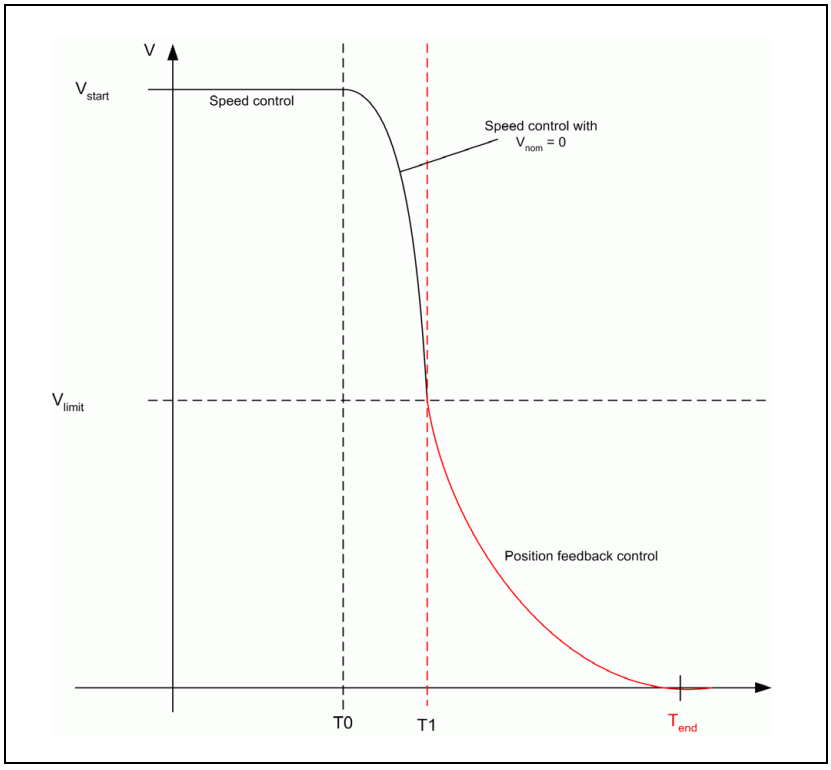
Note

The spindle position must be measured by an encoder before an oriented spindle stop can be executed. If the parameter CfgAxisHardware/ posEncoderType is set to **`no encoder`**, an oriented spindle stop is not possible.

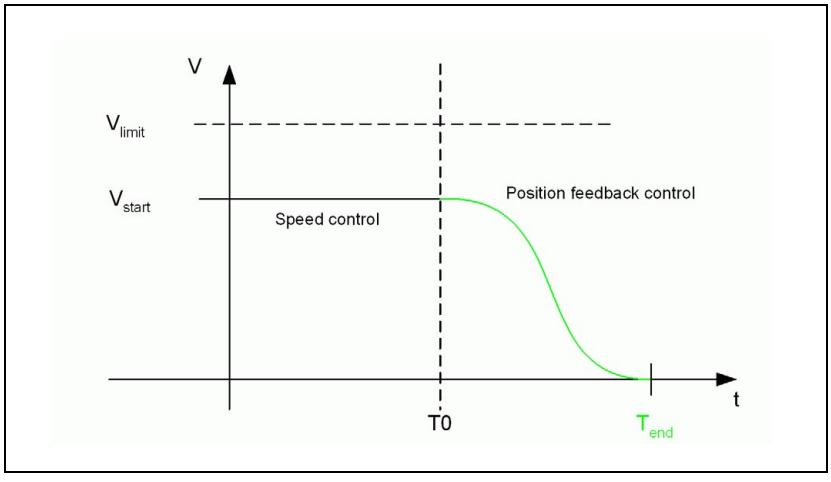
Spindle speed for rotating spindle

An oriented spindle stop (spindle point stop) with **rotating spindle** and at high speeds is executed in two steps if you set **MP_M19MaxSpeed** accordingly:

- Rotational speed > 120% of **MP_M19MaxSpeed** (phase 1):
  The spindle is braked at the limit of current until the speed limit is reached (speed control with $V_{nom} = 0$).
- Rotational speed < 120% of **MP_M19MaxSpeed** (phase 2):
  The position controller is switched on at the rotational speed limit. The actual position, actual speed and actual acceleration at the time of the transition of phase 1 to 2 are the initial values for positioning under position feedback control. This results in a continuous movement, speed and acceleration until the target position is reached. The jerk is limited during deceleration and positioning, whereby the maximum jerk can be set.



The oriented spindle stop with **rotating spindle** and at a speed < 120% of **MP_M19MaxSpeed** is executed under position feedback control.



Make sure that the rotational speed limit does not become too low. Otherwise, due to the position-controller cycle time it will be impossible to generate jerk-limited positioning commands. The rotational speed limit of 3 000 rpm is therefore not violated. If you enter smaller parameter values, the limit is increased to 3 000.

MP_M19MaxSpeed

Maximum rotational speed limit for M19
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 1000 to 20 000 [1/min]
Default: 1 000 [1/min]
Access: LEVEL3
Reaction: RUN

For a spindle orientation requested by the NC (tapping, measuring) the parameter **MP_M19NcSpeed** (400310) can now be used to specify the speed for it.

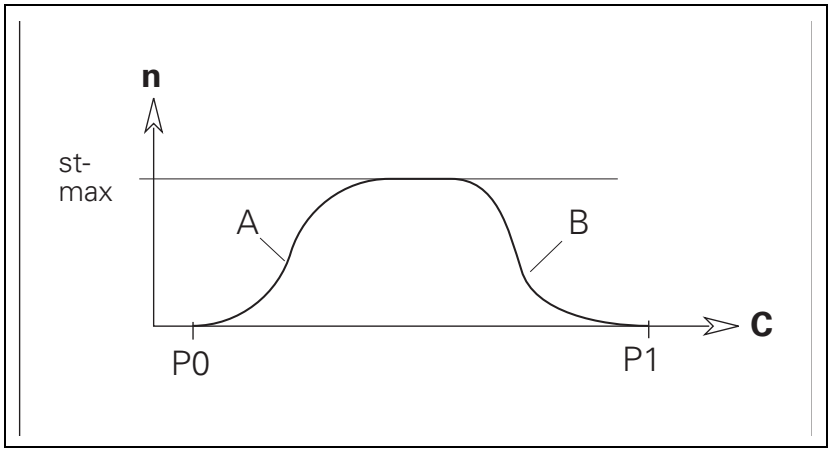
MP_M19NcSpeed

Nominal speed for an oriented spindle stop commanded by the NC
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: RPM [1/min]
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

Oriented spindle stop with stationary spindle

A ramp algorithm determines the acceleration with a **stationary spindle. MP_maxAcceleration** determines the steepness of the ramp during acceleration, **MP_maxAcceleration** determines the steepness of the ramp during braking. The **MP_maxFeed** parameter or the speed from Module 9414 limits the ramp.



Legend:

- **P0:** Initial position
- **P1:** Target position
- **st-max:** MP_maxFeed or the rotational speed from 9414
- **A: MP_maxAcceleration**
- **B: MP_maxAcceleration**

Spindle orientation via PLC module

See "Spindle control by PLC" on page 693.

6.15.7 Switching the operating modes

As soon as the operating mode is switched with Module 9163, the NC switches the drive controller of the spindle off and activates the parameter set from the motor table and the machine parameters. You can check this with Module 9162. After the operating mode has been switched, you must reactivate the drive controller of the spindle with Module 9161.

For a spindle motor, two parameter sets with the same name can be saved in the motor table. This may be necessary if

- Another parameter set applies to a spindle motor at the higher speed range.
- A wye/delta connection switchover is carried out for a motor.

With Module 9163 you can switch between the drive parameters for wye and delta connections of the spindle. The switchover can be carried out during standstill or with a revolving spindle.

Danger

The contactor for the wye/delta switchover must not be switched under load!

To use the operating-mode switchover:

- Enter the two parameter sets of your spindle motor with the same name in the motor table. Identify parameter set 1 by entering 0 in the **MODE** column, and parameter set 2 by entering 1.
- Switch between the two operating modes with Module 9163.
- With Module 9161, reactivate the drive controller.

For the two operating modes, you can use different machine parameters for the spindle (e. g. current and speed controller):

Settings in the configuration editor	MP number
System PLC CfgPlcSStrobe [Key name of S strobe] **gearSpeed0** **gearSpeed1**	 104008 104009

- Enter the key names of spindle parameter sets either in the **MP_gearSpeed0** parameter (wye connection) or **MP_gearSpeed1** (delta connection). These parameter sets define the gear stages for operating modes 0 or 1, if they are selected with PLC Module 9163. Then create the parameter sets with the desired parameters in the configuration editor.

Note

Use the KeySynonym function to create a new parameter set, See "The KeySynonym Function" on page 289

The configuration object **CfgFeedLimits** of the respective parameter set defines the minimum and maximum spindle shaft speed for each gear stage. The list must be sorted in ascending order, with the smallest shaft speed at the top. Gear ranges are not supported if the list is missing or empty.

Module 9163 reports an error if it is called from a PLC program using the numerical TNC API version 1.0.

MP_gearSpeed0

Key names of parameter sets for gear ranges (operating mode 0)
Available from NCK software version: 597 110-02.

Format: Array [0...99]
Input: List with key names for spindle parameter sets for operating mode 0.
Default: –
Access: LEVEL3
Reaction: RESET

MP_gearSpeed1

Key names of parameter sets for gear ranges (operating mode 1)
Available from NCK software version: 597 110-02.

Format: Array
Input: List with key names for spindle parameter sets for operating mode 1.
Default: –
Access: LEVEL3
Reaction: RESET

Module 9163 Switch the drive parameters for delta and wye connection
The module cancels the pulse release of the designated axis, and activates the given parameter set for the drive. When 0 is transferred the parameter set for wye connection is activated, and when 1 is transferred the set for delta connection is activated. Define in **MP_gearSpeed0** the parameters for wye connection, and in **MP_gearSpeed1** the parameters for delta connection.

Call:
PS B/W/D/K <Axis>
PS B/W/D/K <Type of connection>
0: Wye connection
1: Delta connection
CM 9163

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Parameters were switched
	1	Faulty call parameters, see error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Incorrect axis given (currently only 15 = 1st spindle possible)
	2	Incorrect type of connection transferred

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgSpindle	
fastInputType	401501
fastInput	401502
zeroPosEdge	401503
stopOnSwitchSpeed	401504
CfgFeedLimits	
maxDecSpeedCtrl	400312
CfgReferencing	
externRefPulse	400413

The parameter object CfgSpindle is not required for:

- Axes that are **not** defined as spindles (not entered in **MP_spindleIndices**)
- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)

A spindle that is not driven in a position control loop can also be stopped at a defined position (trip dog position). It is a prerequisite that this position be reported to a fast PLC input. A fast PLC input is not read in PLC cycle time but in the IPO clock.

On HSCI controls, a vacant PLC input on the PL 62xx PLC system module or a vacant PLC input of the UEC 11x is used as the input for the spindle reference signal. Use one of the first two slots from the left on the PL 62xx. Use the IOconfig PC software to determine a vacant PLC input and identify its input number.

During a spindle stop, the PLC transfers the rotational speed for "waiting for the input signal" (See "Module 9414 Position the spindle" on page 701) in Module 9414.

The spindle is stopped in three steps.

1. The spindle is decelerated until the defined rotational speed is reached.
2. The spindle continues rotating at the defined rotational speed.
3. The spindle is stopped immediately (without ramp) as soon as the input signal of the spindle trip dog is detected.

Use the **MP_fastInputType** parameter to define how the control is to handle the signal of the trip dog for reference end position. Select **`forStopping`** to stop the spindle at trip dog position with M19 upon receiving this input signal.

Select **`forReferencing`** to use the input signal as spindle reference pulse.

Note

If you use the **`forReferencing`** function, you have to set the parameter **MP_refType** = **`None`** in the **CfgReferencing** configuration object.

Reference run process:

- **With M3 or M4:**
  The spindle accelerates to the programmed speed and picks up the reference point during this process.
- **With M19:**
  If the spindle was not referenced before, it rotates at 10 rpm in positive direction until the reference pulse has been recognized. Then the spindle stops and orients.

Referencing with external reference signal

Set the value **`spindleRef`** in **MP_externRefPulse** if you want to use an external reference signal instead of the reference signal of the connected motor encoder or position encoder (e.g. when gear ranges are used). Only one axis in the entire system can be homed with an external reference signal.

MP_fastInputType

Treatment of the fast input for the spindle
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **none**
Spindle does not have any trip dog for reference end position (no signal) or the signal is not evaluated.
forStopping
For M19 without rotary encoder, the spindle is positioned at the trip dog upon receiving this input signal.
forReferencin
The input signal is used as spindle reference signal. The control handles the signal of the trip dog for reference end position as reference pulse of the encoder.

Default: None
Access: LEVEL3
Reaction: RUN

MP_fastInput

Number of the fast PLC input for the spindle reference run
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to 32767
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

MP_zeroPosEdge defines the trip dog edge, which defines the spindle stop position in positive direction of rotation.

MP_zeroPosEdge

Evaluation of the edge
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: zeroOne
Zero-one transition is evaluated.
oneZero
One-zero transition is evaluated.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

MP_stopOnSwitchSpeed defines the speed for positioning at the trip dog. If you defined the value **`forStopping`** in the **MP_fastInputType** parameter, you have to enter the positioning speed here.

MP_stopOnSwitchSpeed

Shaft speed for positioning to the trip dog
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: RPM [1/min] with up to 9 decimal places
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

MP_maxDecSpeedCtrl

Brake ramp for the spindle
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: 0.000 000 000 to 1000 [m/s2] or [1000°/s^2]
Define a brake ramp for the spindle different from the acceleration ramp. You usually enter the same value as in **MP_maxAcceleration**.

Default: 3 [m/s2] or [1000°/s^2]

Access: LEVEL3

Reaction: RUN

MP_externRefPulse

Referencing with external reference signal
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: **Off**
The reference signal of the connected motor encoder or the position encoder is used.
spindleRef (X30)
On systems with HSCI, the reference signal at -SP.REF+ and -SP.REF- of the external PL is evaluated; on systems with MC 320T or MC 420 the reference signal at input X30 is evaluated.

Default: Off

Access: LEVEL3

Reaction: RUN

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgAxisAnalog **unipolar**	 400113

Note

This function is only available if the spindle is controlled via an analog nominal-value output!

The same polarity of the nominal voltage value is output for both directions of rotation (M3 and M4) for "unipolar" spindles. (For example, for a speed of 9 V, or 1000 rpm, a voltage of +4.5 V is output for S500 for both M3 and M4.) The motor is then switched over by means of a contactor controlled by a PLC output.

You configure a unipolar spindle as follows:

- Open the parameter set of the spindle concerned and move the cursor to the **CfgAxisAnalog** config object.
- Add the optional parameter **MP_unipolar** (400113) to the configuration.
- Configure the parameter according to your needs (see the parameter description below)

MP_unipolar

Algebraic sign with analog unipolar drives
Available from NCK software version: 597 110-03.

Format: Selection menu

Selection: **off**
Not a unipolar drive
always positive
A positive voltage is output for each direction of rotation (M3, M4)
always negative
A negative voltage is output for each direction of rotation (M3, M4)

Default: No value, parameter optional

Access: LEVEL3

Reaction: RUN

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgControllerTol	
syncTolerance	401105
timeSyncOK	401106

The parameter object CfgControllerTol is not required for:

- Virtual axes (MP_axisMode=Virtual)
- Axes that are for display only (MP_axisMode=Display)

Tolerances for synchronism

The control reports that the synchronism of two spindles is reached if the spindle remains within the control window of MP_syncTolerance for the period of time defined in MP_timeSyncOK.

- In MP_syncTolerance, define the size of the control window.
- In MP_timeSyncOK, define the period of time which the spindle is to remain within the control window.

MP_syncTolerance

Angle tolerance for spindle synchronism
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.001 000 000 to 2 [°]
Default:	0.01
Access:	LEVEL3
Reaction:	RUN

MP_timeSyncOK

Hysteresis time for spindle synchronism
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 000 000 to 20 [s]
Default:	0.01 [s]
Access:	LEVEL3
Reaction:	RUN

Spindle synchronism

The NC stores the spindle synchronism in the following operands.

PLC operand / Description	Type
NN_SpiSyncSpeed Rotational speed synchronism active 0: Rotational speed synchronism not active 1: Rotational speed synchronism active	M
NN_SpiSyncAngle Angle synchronism active 0: Angle synchronism not active 1: Angle synchronism active	M
NN_SpiSyncReached Synchronous operation reached 0: Synchronism not reached 1: Synchronism reached	M

Module 9415 Synchronize spindles

The module synchronizes the specified master spindle and slave spindle.

This module is only supported by the new symbolic memory interface. If the iTNC-compatible interface is used, the module returns an error.

Rotational speed: The rotational speed of the master spindle is programmed. The speed factors then determine the speed of the slave spindle.

Remember that:
Master speed * FM = Slave speed * FS

- FM: Speed factor of master
- FS: Speed factor of slave

The angle offset is set as default.

Call:

PS	B/W/D/K	<Logic number of master spindle>
PS	B/W/D/K	<Logic number of slave spindle>
PS	B/W/D/K	<Mode> 0: Switch off all synchronism functions (the spindle numbers have no meaning) 1: Switch off synchronization of master spindle and slave spindle 2: Switch on synchronization of master spindle and slave spindle
PS	D/K	<Speed factor of master>
PS	D/K	<Speed factor of slave>
PS	D/K	<Angle offset in [0.0001°]>
CM	9415	
PL	D	<Error code> 0: Module successfully executed 1: Incorrect module call 2: No permission for module call 3: Invalid mode 4: Invalid speed factor

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Status ascertained
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid spindle number
	9	Module call not possible at this time
	99	Module is not supported (control does not operate with symbolic memory interface)

6.15.11 Spindle of the kinematics model (as of NC software 548 328-03)

Settings in the configuration editor	MP number
Channels Kinematics CfgKinComposModel [Key name of the kinematics model] **activeSpindle**	 202902

In **MP_activeSpindle** you specify the spindle used in the kinematics model. This assignment is required for different calculations (for example: calculate feed per revolution, determine tool life, etc.)

MP_activeSpindle

	Key of the active spindle of this kinematics model Available from NCK software version: 597 110-01.
Format:	String
Input:	The key name of the spindle is read from CfgAxes/ spindleIndices, e.g. "S", "Spindle1", etc.
Default:	–
Access:	LEVEL3
Reaction:	RUN

For more information on the configuration of the machine kinematics of your control, See "Machine kinematics on lathes (as of NC software 548328-03)" on page 487

6.15.12 Spindle of the kinematics model (until NC software 548 328-02)

Settings in the configuration editor	MP number
Channels Kinematics CfgKinModel [Key name of the kinematics model] **activeSpindle**	 200011

In **MP_activeSpindle** you specify the spindle used in the kinematics model. This assignment is required for different calculations (for example: calculate feed per revolution, determine tool life, etc.)

MP_activeSpindle

Key of the active spindle of this kinematics model
Available from NCK software version: 597 110-01.

Format: String
Input: The key name of the spindle is read from CfgAxes/ spindleIndices, e.g. "S", "Spindle1", etc.
Default: –
Access: LEVEL3
Reaction: RUN

For more information on the configuration of the machine kinematics of your control, See "Machine kinematics for lathes (up to NC software 548 328-02)" on page 502

6.15.13 Gear shifting

The PLC is responsible for gear shifting. The PLC also manages the parameters that are required for gear shifting. Please refer to "Gear shifting" on page 1167 in the PLC section of this documentation.

A separate parameter set can be created for every gear range.

The PLC is instructed by M function (M40 to M44) to switch on a certain gear range.

6.15.14 Tapping

Tapping is executed with position feedback control. The spindle and the tool axis interpolate with each other. A floating tap holder is not required.

An oriented spindle stop is performed before tapping.
I.e., a certain spindle position is assigned to each axis position. This synchronization makes it possible to cut the same thread more than once. The NC orients the spindle.

The feed-rate override can be changed during tapping. The MANUALplus 620 automatically adjusts the rotational speed to the changed feed rate. The speed override has no function during tapping.

- Define another parameter set and switch to this parameter set if you want to achieve a specific control response for tapping.

6.15.15 C-axis operation

The MANUALplus 620 supports the following configurations for C-axis operation:

- The C axis and spindle use the same servo drive. Because the speed encoder is built into the motor, it measures both the C axis and the spindle.
- The C axis and spindle each use their own servo drive. Because the speed encoder is built into the motor, there are separate speed encoders for the C-axis and the spindle.
- The C axis and spindle use one position encoder.
- The C axis and spindle each use their own position encoder.
- The axis and spindle can be operated as an analog or digital axis or spindle.

Separate parameter sets for the spindle and C axis are required in all configurations. In these parameter sets, you define the servo drive, the position encoder and speed encoder as well as the other details separately for the spindle and C axis.

- Define separate parameter sets for spindle and C-axis operation.
- The commissioning must be performed for spindle operation as well as for C-axis operation.

Note

The C-axis position controller should be commissioned in the gear range that is actually used for positioning. If possible, use the lowest gear range to ensure optimum controllability.

If you use only one position encoder for both the spindle and the axis, the axis display keeps running while the spindle is in operation:

- Before switching from the axis to the spindle, save the actual position value of the axis with Module 9146. This ensures that the axis display remains at the last value, even when the spindle is rotating.
- Before switching from the spindle to the axis, recover the actual position value of the axis with Module 9146.

If you save the actual position value with Module 9146 and then close the position control loop, or if the position control loop is closed and the actual position value is then saved with Module 9146, the error message **`Actual position value saved <Axis>`** appears. The error message triggers an emergency stop.

Switching from **spindle to C axis:** (example):

- Stop the spindle
- Change the axis by switching the parameter set
- Start C-axis operation

Switching from **C-axis to spindle** (example):

- Stop C-axis operation
- Change the axis by switching the parameter set
- Start spindle operation

Module 9146 Saving and reestablishing actual position values

Module 9146 saves and later reestablishes the actual position values of axes. If the actual position values were saved, the last value displayed remains until they are reestablished.

Call:

PS B/W/D/K <Axes bit-encoded>
PS B/W/D/K <Mode>
0: Save actual position values
1: Reestablish actual position values

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Actual position values saved or reestablished
	1	Error code in NN_GenApiModuleError
NN_GenApiModule ErrorCode	1	Invalid mode
	2	Invalid axes
	24	Module was called in a spawn job or submit job

6.15.16 Volts-per-hertz control mode

In volts-per-hertz control mode (U/f control mode), the motor is speed-controlled in an open loop. The motor is digitally controlled using HEIDENHAIN or non-HEIDENHAIN inverter systems.

You can use the following HEIDENHAIN inverters to set up the volts-per-hertz control mode:

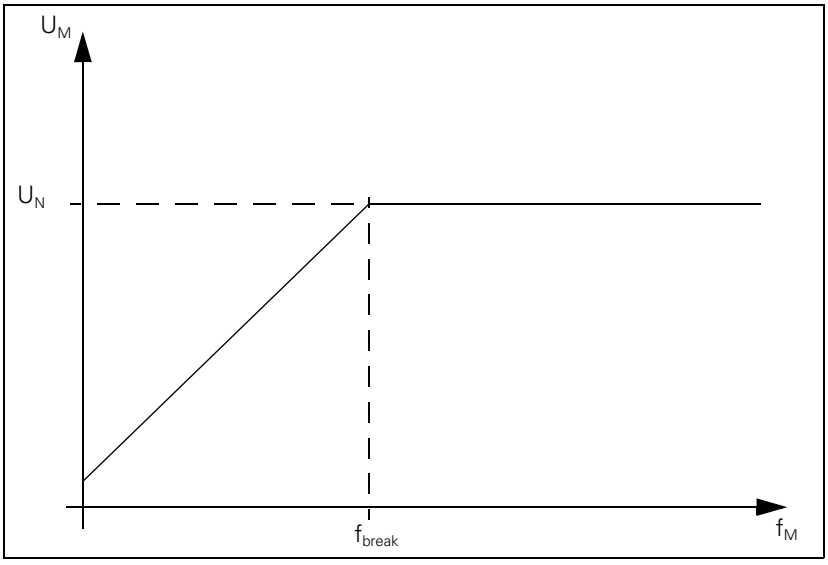
- UM 1xx
- UE 2xx
- UR 2xx
- UEC 1xx

You need a control loop to use a motor in volts-per-hertz control mode, and the appropriate axis has to be enabled in the SIK. Connect the PWM output of the CC with the inverter (not required for UEC 11x)—the encoder inputs remain unconnected.

Note

The volts-per-Hertz control mode is a type of digital motor control that does not return the rotational speed. This control mode cannot be used for axes controlled through the analog nominal shaft-speed interface (± 10 V signals).

The motor voltage increases in proportion to frequency up to the break (= threshold rpm for field weakening). Then the motor voltage remains constant (= rated voltage of motor); only the frequency continues to increase.
If the spindle reaches the maximum current due to excessive load, the error message **C380 Motor <Spindle 1/2> not controllable** appears. For the maximum current, the value from either the motor table or the power module table of the drive (whichever is lower) applies.



The maximum speed in the volts-per-hertz control mode corresponds to the maximum speed in closed loop operation.

To drive a motor with a U/f component:

- In the motor table, enter for your motor in the column **Motor model** (TYPE) UASM, in the column **Encoder line count** (STR.) the value 0, in the column **Type of encoder** (SYS) the value 0 and in the column **Maximum temperature** [°C] T-MAX the value 255.
- The machine parameters for current controller and speed controller have no effect.
- The acceleration and braking ramp (**MP_maxAcceleration**) must be set so that the maximum current is not exceeded.

During volts-per-hertz (U/f) control mode, no speed encoder is used.
Therefore: Actual speed value = Nominal speed value

- Use Module 9164 to determine the speed while the spindle is running, but not during the acceleration and braking phases.

Note

The oscilloscope shows the actual current instead of the nominal current (I NOML), since there is no nominal current with U/f components.

6.16 Configuring the Controller Unit and Drive Motors

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

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

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

The CC 61xx family has a modular structure. The addressed DSP is on the motherboard of the controller. These drive-control motherboards have two control loops, and can therefore control two axes. The same applies to each drive-control expansion board, which does not have its own DSP, however. A drive-control expansion board has two control loops, but can only be used in combination with a drive-control motherboard (with DSP).

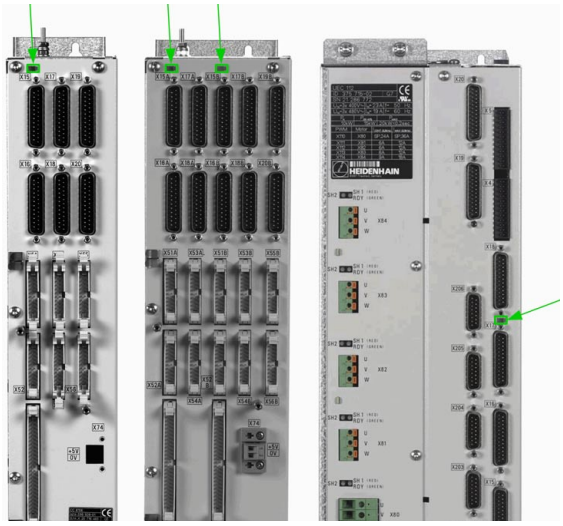
Up to two controller expansion boards can be connected to a drive-control motherboard (with DSP). This configuration results in a maximum of six axes that a single DSP can control. If more than six axes are to be controlled, then another drive-control motherboard is necessary, providing another DSP.

If two drive-control motherboards (two DSPs) are housed in one CC, then the inputs and outputs of the CC are uniquely assigned to one of the two DSPs via the letters A and B.

Index A means that these inputs and outputs are controlled by the first DSP (first drive-control motherboard). Index B means that these inputs and outputs are controlled by the second DSP (second drive-control motherboard).

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

There is one LED on each drive-control motherboard indicating its HSCI address relevant for **MP_hsciCcIndex** by a blink code.



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

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

Speed output: PWM output	**Input: Speed encoder**	**Input: Position encoder**
X51(A/B)	X15(A/B)	X201(A/B)
X52(A/B)	X16(A/B)	X202(A/B)
X53(A/B)	X17(A/B)	X203(A/B)
X54(A/B)	X18(A/B)	X204(A/B)
X55(A/B)	X19(A/B)	X205(A/B)
X56(A/B)	X20(A/B)	X206(A/B)

PWM frequencies with the CC 61xx

The same PWM frequency must be set for both PWM outputs of a controller group in Mp2180.x.

- Controller group 1: X51, X52
- Controller group 2: X53, X54
- Controller group 3: X55, X56

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgPowerStage	
ampPwmFreq	401204
CfgCurrentControl	
iCtrlPwmType	401003

In **MP_ampPwmFreq,** you assign different PWM frequencies to the CC 61xx controller groups.

There are three fundamental PWM frequencies: 3.333 kHz, 4 kHz and 5 kHz. A control loop is operated at the fundamental PWM frequency or at double the fundamental PWM frequency (6.666 kHz, 8 kHz, 10 kHz).

The same PWM frequency must be assigned to both PWM outputs of a controller group with **MP_ampPwmFreq.** Otherwise, the DSP error message **`C440 PWM frequency <axis> incorrect`** will appear.

- Controller group 1: X51, X52
- Controller group 2: X53, X54
- Controller group 3: X55, X56

▶ In **MP_ampPwmFreq,** enter the PWM frequency.

MP_ampPwmFreq

PWM frequency
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
0
5 kHz (for HEIDENHAIN inverters)
3.333 kHz
4.166 kHz
5 kHz
6.666 kHz
8.333 kHz
10 kHz

Default: 0
Access: LEVEL3
Reaction: RESET

The cycle time of the current controller depends on the PWM frequency:

$$T_I = \frac{1}{2 \cdot f_{PWM}}$$

If a control loop is operated at double the fundamental PWM frequency, the current controller cycle time is halved. However, this is possible only with double-speed control loops.

If you want to operate single-speed control loops at double the fundamental PWM frequency,

- Set **MP_iCtrlPwmType** = 1 in order to calculate the current controller cycle time from the fundamental PWM frequency although the control loop will be operated at double the fundamental PWM frequency.

If you operate a single-speed control loop at double the fundamental PWM frequency and half the current controller cycle time (**MP_iCtrlPwmType** = 0), the error message **C017 PWM frequency too high** will appear. Current controller cycle time depending on the PWM frequency:

PWM frequency	Single-speed control loop MP_iCtrlPwmType		Double-speed control loop MP_iCtrlPwmType	
	=0	=1	=0	=1
3.333 kHz	150 µs	Error C013!	150 µs	Error C013!
4 kHz	125 µs	Error C013!	125 µs	Error C013!
5 kHz	100 µs	Error C013!	100 µs	Error C013!
6.666 kHz	Error C017!	150 µs[a]	75 µs	150 µs[a]
8 kHz	Error C017!	125 µs[b]	62.5 µs	125 µs[b]
10 kHz	Error C017!	100 µs[c]	50 µs	100 µs[c]

a. Fundamental PWM frequency of 3333 Hz
b. Fundamental PWM frequency of 4000 Hz
c. Fundamental PWM frequency of 5000 Hz

MP_iCtrlPwmType

Current controller cycle time
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: 0: Cycle time = 1 / (2 · f_{PWM})
1: Cycle time = 1 / f_{PWM}
2: Reserved
The value from CfgPowerStage/ampPwmFreq is used as the PWM frequency f_{PWM}.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RUN

6.16.3 PWM frequency with INDRAMAT "POWER DRIVE" inverters

- In **MP_ampPwmFreq,** enter the PWM frequency 4.166 kHz.

6.16.4 PWM frequency with SIEMENS "SIMODRIVE" inverters

The MANUALplus 620 operates with a PWM frequency of 5 kHz. SIEMENS power modules are normally driven with a PWM frequency of 3.2 kHz (spindle) and 4 kHz (axes).
The rated current values I_N are defined for these frequencies. If power modules are operated with a higher PWM frequency (e.g. 5 kHz), high temperatures can be caused in these modules in some cases.

This applies particularly to these SIEMENS power modules:

- 6SN1123-1AA00-0CA0 (as axis module)
- 6SN1123-1AB00-0CA0 (as axis module)

Machines that are not under full load do not exceed the maximum permissible temperature.

There are two ways to prevent the undesired heating:

- In **MP_ampPwmFreq,** enter the required PWM frequency (3.333 kHz or 4.166 kHz).
  or
- Reduce the factor for I^2t monitoring
  or
- Reduce rated current I_N in the table of power modules.

Note

A reduction of the PWM frequency has no effect on the maximum rotational speed, but it means that the axes and the spindle(s) must be commissioned again.

When a new machine is put into service, HEIDENHAIN recommends the PWM frequency fitting for the axis modules (normally 4 kHz, see the SIEMENS documentation). If the power module of the spindle gets too warm in spite of a reduction of the PWM frequency from 5 kHz to 4 kHz, then the reference value for the I^2t monitoring or the rated current I_N must be reduced in the table of power modules.

Reduction of the reference value for I^2t monitoring or rated current I_N in the table of power modules

The reduction of the rated current I_N of the power modules, as well as the reference value for I^2t monitoring can be calculated from two values (X1, X2) that are given in the SIEMENS documentation.

The percent reduction of the rated current can be calculated with the following formula:

$$X_R[\%] = 100 - \left(\frac{(100 - X1) \cdot (8\ \text{kHz} - f_{PWM})}{8\ \text{kHz} - X2} + X1 \right)$$

- X1 = Reduction factor of the current in % at a PWM frequency of 8 kHz
- X2 = PWM threshold frequency in kHz at which the electrical power reduction begins
- f_{PWM} = frequency in kHz set in MP_ampPwmFreq

This results in the reference value for I^2t monitoring:

$$X_B = 1 - \frac{X_R[\%]}{100}$$

- Reduce the rated current values I_N of your power modules in the power module table.

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

or

- Reduce the reference value (MP_CfgServoMotor/motFactorI2t) for the I^2t monitoring.

  Reference value $= X_B$

Note

A reduction of the rated current of the power module can cause a reduction of the rated torque and, as a consequence, the rated power of the motor, if equal values for rated current of the power module and the rated current of the motor were chosen.

Example of a 50-A power module:

- Axis power module with 50 A, PWM frequency of 5 kHz, X1 = 40%, X2 = 4 kHz

$$X_R[\%] = 100 - \left(\frac{(100 - 40) \cdot (8\ \text{kHz} - 5\ \text{kHz})}{8\ \text{kHz} - 4\ \text{kHz}} + 40 \right) = 15\ \%$$

$$X_B = 1 - \frac{15}{100} = 0,85$$

- Spindle power module with 50 A, PWM frequency of 5 kHz, X1 = 40%, X2 = 3.2 kHz

$$X_R[\%] = 100 - \left(\frac{(100 - 40) \cdot (8\ \text{kHz} - 5\ \text{kHz})}{8\ \text{kHz} - 3,2\ \text{kHz}} + 40 \right) = 22,5\ \%$$

$$X_B = 1 - \frac{22,5}{100} = 0,78$$

- Axis power module with 50 A, PWM frequency of 4 kHz, X1 = 40%, X2 = 4 kHz

$$X_R[\%] = 100 - \left(\frac{(100 - 40) \cdot (8\ \text{kHz} - 4\ \text{kHz})}{8\ \text{kHz} - 4\ \text{kHz}} + 40 \right) = 0\ \%$$

$$X_B = 1 - \frac{0}{100} = 1,00$$

- Spindle power module with 50 A, PWM frequency of 4 kHz, X1 = 40%, X2 = 3.2 kHz

$$X_R[\%] = 100 - \left(\frac{(100 - 40) \cdot (8\ \text{kHz} - 4\ \text{kHz})}{8\ \text{kHz} - 3,2\ \text{kHz}} + 40 \right) = 10\ \%$$

$$X_B = 1 - \frac{10}{100} = 0,90$$

6.16.5 Comparison of the CC 61xx and CC 424 controller units

Function	CC 61xx	CC 424
Hardware	Position encoder inputs X201(A) to X206(A) and X201B to X206B on the CC 61xx	Position encoder inputs X201 to X206 and X207 to X210 on the CC 424
Assignment of speed encoder inputs to the PWM outputs	Permanent assignment, position encoder inputs are also permanently assigned. Always in one row, e.g. X16A, X51A and X202A are permanently assigned to each other.	Fixed assignment, MP_speedEncoderInput not required
MP_hsciCcindex	Assignment of axes and spindles to drive-control motherboards	Not present
MP_pwmSignalOutput	Assignment of axes and spindles to the speed outputs (X15 to X18, X15 to X20)	Assignment of axes and spindles to speed outputs
MP_posEncoderType	New possible selection "CC_EXTERN_ENDAT_2_2" for CC 61xx.	The possible selection "CC_EXTERN_ENDAT_2_2" has no function
Encoders with EnDat 2.2 interface	Encoders with EnDat 2.2 interface can be operated without analog encoder signals. The position is evaluated purely digitally via a serial data protocol. The parameter MP_posEncoderSignal has no effect. EnDat 2.2 encoders can be selected as motor encoders in parameter MP_motEncType.	Encoders can only be operated via analog encoder signals.
PWM frequency	Same as CC 424, only the controller groups for which the same PWM frequency must be set have changed, see page 723.	Can be set via MP_ampPwmFreq (switchable during operation); the calculation of the current-controller cycle time must be adapted via MP_iCtrlPwmType
Control loops can be switched from single speed to double speed for higher controller performance	See "PWM frequencies with the CC 61xx" on page 725.	
Control-loop cycle times (at 5000-Hz PWM frequency) (position/speed/current)	Single-speed: 200 µs/200 µs/100 µs Double-speed: 100 µs/100 µs/100 µs (with position encoder) 100 µs/100 µs/100 µs (without position encoder)	Single-speed: 200 µs/200 µs/100 µs Double-speed: 200 µs/100 µs/100 µs (with position encoder) 100 µs/100 µs/100 µs (without position encoder)
Following error in the jerk phase (MP_complpcJerkFact)	Same as CC 424	Typical input values: 0.001 to 0.005

Function	CC 61xx	CC 424
Stick-slip friction compensation (MP_compFriction0, MP_compFrictionT1, MP_compFrictionT2)	Same as CC 424	Feed-rate independent; MP_compFriction0 same meaning as previously (effective values, readjustment necessary), MP_compFrictionT1 new meaning MP_compFrictionT2 new
Filter in the speed control loop	Same as CC 424	MP_vCtrlFiltLowPassT omitted, new machine parameters: MP_vCtrlFiltDamping1 to MP_vCtrlFiltDamping5 MP_vCtrlFiltFreq1 to MP_vCtrlFiltFreq5 MP_vCtrlFiltType1 to MP_vCtrlFiltType5 MP_vCtrlFiltBandWidth1 to MP_vCtrlFiltBandWidth5 MP_vCtrlEncInputFilt new meaning
Master-slave torque control	The PWM outputs of the master and slave axes must always be operated on the same DSP (meaning the same controller basic board for the CC 61xx), i.e. the master and slave axes must be operated with the same power. More slave axes are possible for the CC 61xx than before, e.g. up to five slave axes are possible for a CC for six axes.	The PWM outputs of the master and slave axes must always be operated on the same DSP ("single-speed" setting)
Reading the absolute value of encoders with EnDat interface	Same as CC 424	The absolute value can be read out again via the **PASS OVER REFERENCE** soft key or via Module 9220 (i.e. after the exchange of milling heads).
MP_motEncCheckOff	Same as CC 424	Bit 4: Monitoring for excessive temperature Bit 5: Monitoring for insufficient temperature Bit 6: Reserved Bit 7: Monitoring of encoder input frequency Bit 8: Adjust mechanical offset by gradually increasing the k_V factor
MP_motTypeOfFieldAdjust	Same as CC 424	Determining the field angle for unaligned encoders
MP_plcCount	Same as CC 424	PLC cycle time [ms]

Function	CC 61xx	CC 424
MP_ipoCycle	Same as CC 424	MP_ipoCycle omitted, path interpolation fixed at 3 ms (does not influence the position controller cycle)
Display in internal oscilloscope and in TNCopt	Same as CC 424	Effective values

6.16.6 Configuring the servo motor

Settings in the configuration editor	MP number
Axes ParameterSets [Key name of the parameter set] CfgServoMotor **motName** **motEncType** **motDir** **motStr**	 401301 401311 401312 401313
System Paths CfgTablePath MOTOR_OEM **path**	 MOTOR_OEM.102501

The parameter object CfgServoMotor is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Analog axes (**MP_axisHw** = Analog)

Configure the servo motor in the CfgServoMotor parameter object. You can use the entries in the motor table for the parameters **MP_motEncType, MP_motDir** and **MP_motStr,** or use the three parameters to overwrite the values in the motor table.

The motor table currently active for the MANUALplus 620 is opened when making an entry in **MP_motName.** Select the name of the motor.

Along with the motor table supplied by HEIDENHAIN, the OEM has the possibility of saving his own motor table on the MANUALplus 620 as well. The soft keys OEM and SYS are used to open the respective tables in the table editor. The SYS motor table is write-protected.

In the dialog window for selecting a motor, all entries from the OEM motor table have an asterisk (*) in the OEM column to identify them as entries made by the OEM.

You must configure the path to the OEM motor table. In the "Settings in the configuration editor" table at the top of the page you will find the **MP_path** parameter.

▶ Specify the path to the OEM motor table under the key name "MOTOR_OEM" in the **MP_path** parameter.

MP_motName

Name of the motor
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Name of the motor
Choose the name from the motor table.
Default: –
Access: LEVEL3
Reaction: RESET

Enter the type of encoder in **MP_motEncType.** If "off" is entered, then the type of encoder entered in the motor table is used.

MP_motEncType

Type of speed encoder
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **off**
The encoder type entered in the motor table is valid.
ROTATING_WITH_Z1
Incremental rotary encoder with Z1 track
ROTATING_ENDAT_ADJUSTED
Aligned absolute rotary encoder
LINEAR_ENDAT
Absolute linear encoder
LINEAR_INCREMENTAL
Incremental linear encoder
ROTATING_ENDAT_NOT_ADJUSTED
Unaligned absolute rotary encoder
ROTATING_WITHOUT_Z1
Incremental rotary encoder without Z1 track
ROT_DIST_CODED_NOT_ADJUSTED
Unaligned incremental rotary encoder with distance-coded reference marks
LIN_DIST_CODED_NOT_ADJUSTED
Unaligned linear encoder with distance-coded reference marks
DIG_ENDAT_2_2_ADJUSTED
Purely digital and aligned EnDat 2.2 rotary encoder
DIG_ENDAT_2_2_NOT_ADJUSTED
Purely digital and unaligned EnDat 2.2 rotary encoder
LIN_ENDAT_2_2
Purely digital EnDat 2.2 linear encoder
Default: off
Access: LEVEL3
Reaction: RESET

You define the counting direction of the encoder in **MP_motDir.** If "off" is entered, then the counting direction entered in the motor table is used.

MP_motDir

	Counting direction of the motor encoder Available from NCK software version: 597 110-02.
Format:	Selection menu
Selection:	**Off** The counting direction entered in the motor table is valid. **Positive** Positive counting direction. **Negative** Negative counting direction.
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

Use the parameter **MP_motStr.** to define the line count of the motor encoder. If "0" is entered, then the line count entered in the motor table is used.

MP_motStr

	Line count of the motor encoder Available from NCK software version: 597 110-02.
Format:	Numerical value
Input:	**0:** The line count entered in the motor table is used. **> 0:** Line count of the motor encoder
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

Note

Note: Please consider the following chapter on field orientation to be a "preliminary description." The procedure for determining the field angle, the messages that appear, etc. are still being developed, and will be available completely with a later software version.

If a linear, torque or synchronous motor is used with an incremental encoder without a Z1 track or an unaligned encoder with EnDat interface, there is no association between the encoder and the rotor magnets. The field angle must be determined before this motor can be moved.

The control uses the "field orientation" function to determine the field angle for the motors listed above. The association between the encoder and the rotor magnet (field angle) is determined and stored.

Regarding the **motor.mot** motor table, the field orientation must be performed for the following drives:

- Linear motor with absolute encoder with EnDat interface **(SYS = 3)**
- Synchronous or torque motor with unaligned rotary encoder with EnDat interface **(SYS = 5)**
- Synchronous or torque motor with incremental rotary encoder without Z1 track **(SYS = 6);** one reference mark per revolution
- Synchronous or torque motor with incremental rotary encoder with distance-coded reference marks **(SYS = 7)**
- Linear motor with incremental linear encoder with distance-coded reference marks **(SYS = 8)**
- Synchronous or torque motor with unaligned rotary encoder with EnDat 2.2 interface **(SYS = 10)**

Encoders for direct drives

- An absolute encoder with EnDat interface should be used, since the absolute position value is available directly after switch-on, and the field angle can be associated immediately. This means that the motor can be controlled immediately.
- The encoder should have a high line count. This leads to better controllability.
- With incremental encoders the motor must first be moved a "minimum" distance in order to determine a field angle with which the motor can be moved until the reference mark. Only after the reference mark has been traversed can the field angle determined during commissioning be assigned.
- If excessive clamping of the axis prevents the "minimum" motion for determining the field angle, then no field angle can be determined and the axis cannot be controlled. In this case the clamping must be undone for the field angle to be determined. If this is not possible, because the axis would fall down, then an absolute encoder with EnDat interface must be used.

Field orientation with absolute encoder (EnDat)

As soon as the absolute position of the encoder has been read, the absolute position and determined field angle are associated.
The field angle is associated with the zero position of the encoder.

Field orientation via encoder with Z1 track

After switching on the control, the motor orients itself (rough orientation) via the Z1 track of the encoder. The drive is ready for operation after this procedure. The field angle is determined and associated as soon as the reference mark is traversed during the first motor motion.

Field orientation via the control

The control uses the "field orientation" function to ascertain the assignment between the encoder and the rotor magnet (field angle).

The possibilities for ascertaining the field angle depend on the CC model:

- **CC 422:**
  - Automatic field orientation when starting the control, with motion of the motor.
  - Field orientation via code number during commissioning of the motor, with motion of the motor. This method determines the field angle more accurately than the "Automatic field orientation."

Note

The "field orientation" function can be performed only if the current controller is already adjusted!

Note

For synchronous spindles, the field angle should be determined via the code number, since this is a more exact determination.

Danger

- An encoder with absolute values or an encoder with a Z1 track must be used when determining the field angles for hanging axes (or braked axes that could move on their own).
- If the speed encoder is exchanged, the Field Orientation function must be rerun.

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgServoMotor	
motTypeOfFieldAdjust	401314
motFieldAdjustMove	401315
motPhiRef	401316
motEncSerialNumber	401317
CfgAxisHardware	
checkPhiFieldRef	400016

The parameter object CfgServoMotor is not required for:

- Virtual axes (**MP_axisMode** = Virtual)
- Axes that are for display only (**MP_axisMode** = Display)
- Analog axes (**MP_axisHw** = Analog)

▶ Select the method for field angle determination in **MP_motFieldAdjustMove**. When commissioning unaligned encoders **MP_checkPhiFieldRef** must also be set to "on."

MP_motFieldAdjustMove

Determining the field angle
Available from NCK software version: 597 110-02.

Format: Numerical value
Input: 0: Automatic field orientation without plausibility test
2: Field angle determination via code number with plausibility test; motor motion is permitted
Default: 0
Access: LEVEL3
Reaction: RESET

MP_checkPhiFieldRef

Adjustment for unaligned speed encoders
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **on**
Perform adjustment during initial setup
off
Do not adjust
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

HEIDENHAIN recommends using **MP_motFieldAdjustMove** = 2 when commissioning new drive systems (such as machine prototypes), because the plausibility tests will be run. After successful commissioning, **MP_motFieldAdjustMove** = 0 can be used to save time (such as for series production of the machine).

Plausibility test

During plausibility testing, some machine parameters and part of the wiring are checked.

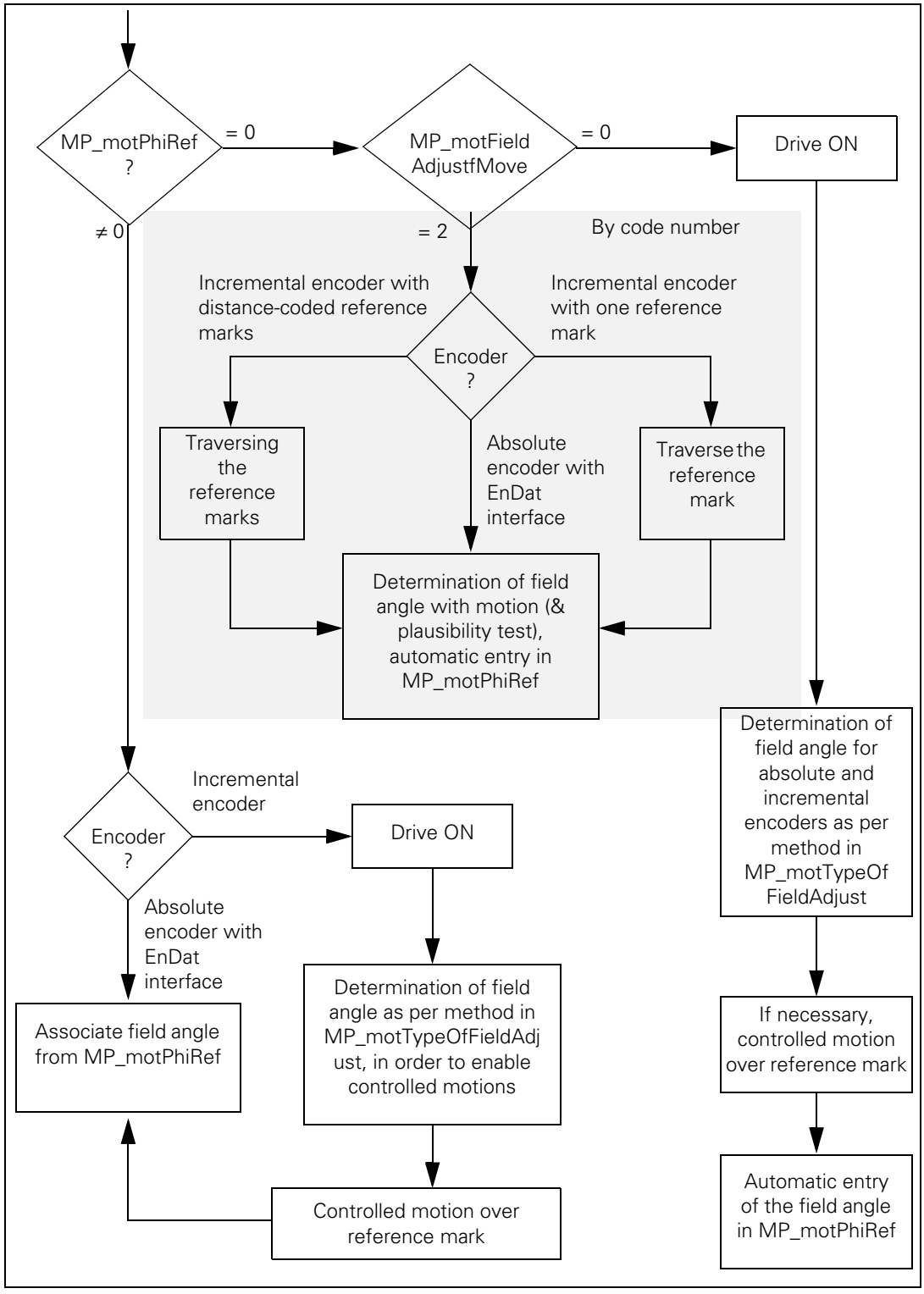
- Encoder line count
- Number of pole pairs
- Rotational direction of the electrical field
- Traverse distance per electrical revolution

Note

Determination of the field angle with plausibility testing is recommended for initial setup, new designs, and other similar situations.

The following messages can appear during the plausibility test:

- **8630 Field orient. successful** indicates that the field angle was successfully determined and stored in MP_motPhiRef.
- **8B10 Wrong traverse direction** indicates that the rotational direction of the electrical field does not match the counting direction of the encoder. Error fix: Change the entry in MP_motDir.
- **8B20 Error field orientation** indicates that no usable measuring results could be determined. One common reason for this is that the parameters for the encoder are incorrect. A further cause could be that the motor is moving against a resistance (e.g. brake is still active, bellows, limit stop) or that the mechanics are too stiff.


MP_motPhiRef ?
= 0
MP_motField AdjustfMove
= 0
Drive ON
≠ 0
= 2
By code number
Incremental encoder with distance-coded reference marks
Encoder ?
Incremental encoder with one reference mark
Traversing the reference marks
Absolute encoder with EnDat interface
Traverse the reference mark
Determination of field angle with motion (& plausibility test), automatic entry in MP_motPhiRef
Determination of field angle for absolute and incremental encoders as per method in MP_motTypeOf FieldAdjust
Incremental encoder
Encoder ?
Drive ON
Absolute encoder with EnDat interface
Associate field angle from MP_motPhiRef
Determination of field angle as per method in MP_motTypeOfFieldAdj ust, in order to enable controlled motions
If necessary, controlled motion over reference mark
Controlled motion over reference mark
Automatic entry of the field angle in MP_motPhiRef

Determining the field angle without motor motion

A distinction must be made when determining the field angle without motor motion (MP_motFieldAdjustfMove = 0):

- Initial operation, field angle has not been determined yet (**MP_motPhiRef** = 0): Determination of the field angle is started automatically when the MANUALplus 620 is switched on. The method for determining the field angle is set in **MP_motTypeOfFieldAdjust.** The determined angle is stored in **MP_motPhiRef**.
- A field angle has already been determined (**MP_motPhiRef** ≠ 0). The subsequent steps depend on the encoder:
  - Absolute encoder with EnDat interface: The absolute position of the encoder is read immediately after the control has finished booting. The field angle from **MP_motPhiRef** is assigned to this position. Therefore, the first motor motion already occurs with the determined field angle.
  - Incremental encoder: Immediately after the control has finished booting and the control voltage has been switched on, then depending on **MP_motTypeOfFieldAdjust** a field angle is determined with which the motor can be traversed over the reference mark. After traversing the reference mark, the field angle from **MP_motPhiRef** is assigned. The subsequent motor motions utilize the field angle from **MP_motPhiRef.**

The field angle is determined automatically after switching on the drive. This process lasts approximately 4 to 6 seconds (the PLC program must not rescind the drive release during this time). The **`Finding field angle`** progress bar appears.

If the power module is not active before the determination of the field angle begins, the error message **`8B40 No drive release <axis>`** appears. If the power module switches off during the determination, **`8B50 Axis module <axis> not ready`** appears.

Methods for determining the field angle

There are two methods for determining the field angle without motor motion:

- **Method 2:**
  Current pulses are output with the brakes applied, and the absolute rotor position is determined from the reaction. A "minimum" movement of the motor must be possible when the brakes are applied.
- **Method 3:**
  Functions in the same manner as Method 2, but with the difference that the motor brakes are not applied. Therefore, this method is not suitable for hanging axes. However, this method can lead to more exact results than Method 2, so it should be used for synchronous spindles. Minimal spindle movements can occur during field angle determination.

▶ In **MP_motTypeOfFieldAdjust** select the method to be used for determining the absolute rotor position.

MP_motTypeOfFieldAdjust

	Determining the field angle for unaligned encoders Available from NCK software version: 597 110-02.
Format:	Numerical value
Input:	0, 2: Method 2 (brakes applied) 3: Method 3 (same as Method 2, but motor brake is not applied)
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

Note

Standstill monitoring is active while determining the field angle. If it responds for motors without motor brakes, increase the threshold in **MP_checkPosStandstill.** Afterwards, reset **MP_checkPosStandstill** to the original value.

Determining the field angle with motor motion

With this field angle determination method (**MP_motFieldAdjustMove** = 2), the motor moves in a certain direction. It should be approximately halfway in the distance traversed before the field angle is determined.
Axes with linear/torque motors can be slid "by hand" if the brakes are not applied.

While the field angle is being found, the speed controller and position controller are opened and the drive controller is active. This means that the motor is moved (approx. 2 pole pairs) and the brake must be open until the field angle is determined.

Danger

Hanging axes need 100 % weight compensation.
Please contact HEIDENHAIN if this is not the case.

Attention

Limit switches are ignored!
If axes move into an illegal area, press the emergency stop button!

Note

When using incremental encoders with distance-coded reference marks, **MP_posEncoderRefDist** (nominal increment between two fixed reference marks) must be set correctly.

MP_motFieldAdjustMove = 2: The PLC initial servicing program, whose name and path is entered in the **MP_pwmPgm** parameter must ensure that the inverters are ready after the control voltage has been switched on but that the motor brakes are only open while determining the field angle. Alternately, the motor brakes can be opened manually for the duration of the field angle determination.

Before determining the field angle (code number not entered yet) the inverter must be in the following mode of operation:

- Green "READY" LED on
- Red "SH1" LED off
- Red "SH2" LED on (drive controller not ready, brakes closed)

As soon as the drive enable comes from the PLC, the **Finding field angle** message appears, otherwise **8B40 No drive release** appears.

The motor moves and the field angle is determined. Limit switches are not taken into account.

In order to avoid the possible error message resulting from standstill monitoring, enter an appropriately high value in **MP_checkPosStandstill.**

- Switch on the MANUALplus 620.
- Do **not** acknowledge the **Power interrupted** message. Enter the code number **94655.**
- The PLC must
  - switch the drive on/off
  - release and lock the brakes

The motor moves back and forth. The field angle is determined for the reference mark or datum, and is stored automatically. The **Finding field angle** progress bar appears. Then one of the following messages appears:

- **8630 Field orient. successful** indicates that the field angle was successfully determined and stored in **MP_motPhiRef.**
- **8B10 Wrong traverse direction** indicates that the rotational direction of the electrical field does not match the counting direction of the encoder. Error fix: Change the entry in the machine parameter **MP_motDir.**
- **8B20 Error field orientation** indicates that no usable measuring results could be determined. One common reason for this is that the parameters for the encoder are incorrect. A further cause could be that the motor is moving against a resistance (e.g. brake is still active, bellows, limit stop) or that the mechanics are too stiff.

The MANUALplus 620 carries out a reset. If the message **8630 Field orient. successful** appears, then the field angle was associated and is available.

Saving the determined field angle

The determined field angle is automatically entered in **MP_motPhiRef.**

For purposes of reliability and redundancy, either the serial number of the encoder (only for EnDat interface) or a unique control ID is entered as identification in **MP_motEncSerialNumber.**

An error message appears if the current identification does not match the entry in **MP_motEncSerialNumber:**

- When using an encoder with EnDat interface, the error message **8830 `EnDat: no field angle <axis>`** appears. In any case the field angle must be determined anew, since the encoder does not match the field angle from MP_motPhiRef.
- When using an incremental encoder, the error message **`MP2257.<index> incorrect (ID=$<identification>)`** appears. The field angle from **MP_motPhiRef** and the new identification **`(ID=$<identification>)`** for **MP_motEncSerialNumber** can only be assumed after determining that the same drive is meant (e.g. after changing controls).

Danger

In all other cases the field angle must be determined anew, since otherwise uncontrolled drive motions could occur!

Note

You can force a new field angle determination be entering MP_motPhiRef = 0 (for example, after exchanging a motor or encoder).

MP_motPhiRef

	Determined field angle Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	Automatic determination and entry by interpolator in the operating mode for determining the field angle. 0: Field angle does not need to be determined, or has not been determined
Default:	0
Access:	LEVEL3
Reaction:	RESET

MP_motEncSerialNumber

	Control or encoder identification for the field angle from MP_motPhiRef Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0: Field angle does not need to be determined, or has not been determined
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

Definition of the field angle

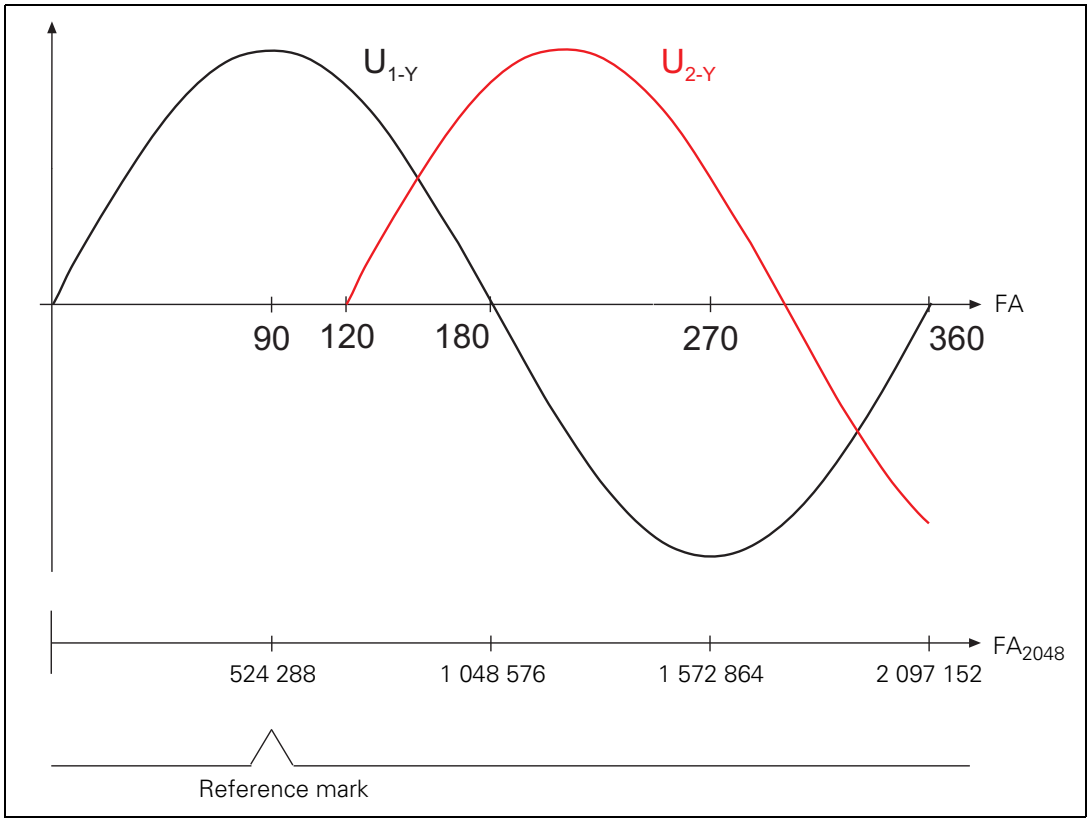
The following applies to the determined field angle, which is entered in **MP_motPhiRef:**

The motor is moved with external power in the positive direction (when viewing the shaft, the shaft rotates clockwise). The voltages U_{1-Y} (phase 1 to star point) and U_{2-Y} (phase 2 to star point) are measured. The positive peak value of U_{1-Y} corresponds to a field angle of 90°.

The field angle at the reference mark is saved in increments in MP_motPhiRef. An increment is formed from <line count> · <interpolation factor>. Therefore:
MP_motPhiRef = (<field angle at reference mark> · <line count> · 1024) / 360°

In the example below, the reference mark is at the field angle 90°, i.e. **MP_motPhiRef** = (90° · 2048 · 1024) / 360° = 524288.

On a "standard" synchronous motor (with aligned speed encoder), the reference mark is at the field angle 0°. If the field angle were to be determined for this motor, the result would be **MP_motPhiRef** = approx. 2097152.



- U_{1-Y}: Motor voltage between phase 1 and star point
- U_{2-Y}: Motor voltage between phase 2 and star point
- FA: Field angle in degrees
- FA_{2048}: Field angle in increments for an encoder with 2048 lines and 1024-fold interpolation (2048 · 1024 = 2 097 152)

Settings in the configuration editor	MP number
System CfgHardware **currentControlAdjust**	 100103

- Enter as much configuration data as possible
- Set **MP_currentControlAdjust** so that the current controller will be adjusted:

MP_currentControlAdjust

Operating mode for current controller adjustment
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **on**
A DSP start (mode for DSP commissioning) is sent to the CCU.
off
No DSP start is sent to the CCU.

Default: off

Access: LEVEL3

Reaction: RESET

Danger

During current controller adjustment of linear and torque motors, the rotor position of the motor is not yet known.
For this reason, if the motor brakes are not active, the motor might move slightly when the current pulses switch on. In other words, it might oscillate about a preferred position. It is also possible to position "manually" to the preferred position. Do not do this, however, during a measurement.

Use the commissioning and diagnostic software TNCopt to adjust the current controller.

During adjustment of the current controller the speed controller and position controller are open. During output of the current pulses the drive controller becomes active.

The PLC commissioning program, whose name and path is entered in the parameter **MP_pwmPgm,** must ensure that the inverters are ready after "Switch on external dc voltage," but that the motor brakes are not opened.

Before and after the output of the current pulses for adjusting, the inverter must be in the following operating mode:

- Green "READY" LED on
- Red "SH1" LED off
- Red "SH2" LED on (drive controller not ready)

During output of the current pulses:

- Green "READY" LED on
- Red "SH1" LED off
- Red "SH2" LED off (drive controller ready)

▶ Commission the current controller with TNCopt. Please refer to the information in the TNCopt User's Manual (Chapter 7.2 Commissioning the Current Controller).

6.18.1 Power module table and motor table

In the configuration editor you select the installed power modules and the motors:

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgPowerStage	
ampName	401201
ampAdditionalInfo	401210
CfgServoMotor	
motName	401301
motEncType	401311
motDir	401312
motStr	401313

- Open the parameter set of the desired axis, and move the cursor to the **MP_ampName** (power stage) or **MP_motName** (motor) parameter.
- Press the ENT key. The MANUALplus 620 opens the list of power modules or motors.

Meaning of the soft keys	
	Scroll back one page in the list
	Scroll forward one page in the list
	Confirm selection
	Cancellation, do not confirm value
	Open OEM motor table in the table editor
	Open HEIDENHAIN motor table in the table editor (write-protected)
	Open power module table in the table editor (write-protected)
	Copy selected value to clipboard
	Paste value from clipboard

In the list of motors, the type of table (OEM or HEIDENHAIN motor table), the type of motor (synchronous, asynchronous, or linear motor), the operating mode (wye/delta) and the maximum current are displayed in addition to the motor designation.

Once you have selected the motor and power module, and have confirmed this with the ENT key or the OK soft key, the selected models are automatically entered in **MP_motName** and **MP_ampName** for the respectively open parameter set.

If you use motors or power modules that are not listed in the menus, please contact HEIDENHAIN.

You can overwrite standard data or add other models to the OEM motor table. The OEM motor table is stored in the PLC partition:

- **PLC:\table\MOTOR_OEM.MOT**

To open the OEM motor table, press the OEM soft key in the table editor. In the list view, the MANUALplus 620 marks all motors from the OEM table with an ***** in the **OEM** column.

The data in the HEIDENHAIN motor table and in the **inverter.inv** power module table on the SYS partition is write-protected. You can open the tables, but you cannot edit them.

Note

The power module table of older HEIDENHAIN contouring controls, **MOTOR.AMP**, is no longer supported by the MANUALplus 620!

If you use a motor that appears in the motor table, but only the data for the speed encoders differs, you can overwrite this data in the motor table with **MP_motStr, MP_motDir** and **MP_motEncType.** The motor table is not actually changed. The changes only take place in the working memory.

Note

The original entry from the motor table is only used if **MP_motStr, MP_motDir and MP_motEncType** are not in the configuration.

MP_ampName

Type of axis power module
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Name of the power module from the inverter.inv power module table (entered by the MANUALplus 620)
Default: –
Access: LEVEL3
Reaction: RESET

MP_ampAdditionalInfo

Configure the switch position of the current sensor (column "S" of the inverter.inv power module table). Is required for the D series of HEIDENHAIN inverters (UM 1xx D) in order to use the higher currents at the lower PWM frequencies.
Available from NCK software version: 597 110-04.

Format: Numerical value (32 bits)
Input: Bit 0 – Configure the switch position of the current sensor in HEIDENHAIN inverters of the "D" series (UM 1xx D).
Bit 0 = 0: "S" in inverter.inv table = 0
Bit 0 = 1: "S" in inverter.inv table = 1
Bit 1- reserved
Bit 2- reserved
Bit 3 – Inverter manufacturer
Bit 3 = 0: Non-HEIDENHAIN inverter
Bit 3 = 1: HEIDENHAIN inverter
Default: -
Access: LEVEL3
Reaction: RUN

MP_motName

Motor model
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Name of the selected motor (entered by the MANUALplus 620)
Default: –
Access: LEVEL3
Reaction: RESET

MP_motDir

Overwrite the "counting direction" of the motor encoder from the motor table
Available from NCK software version: 597 110-03.

Format: Selection menu
Selection: **Off**
Value from the motor table active
Positive
Positive counting direction
Negative
Negative counting direction
Default: No value, parameter optional (value from motor table in effect)
Access: LEVEL3
Reaction: RESET

MP_motStr

Overwrite "Line count" of the motor encoder from the motor table
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0: No speed encoder (volts-per-hertz control mode)
1 to 999 999
Default: No value, parameter optional (value from motor table in effect)
Access: LEVEL3
Reaction: RESET

MP_motEncType

Overwrite the "type of encoder" from the motor table
Available from NCK software version: 597 110-03.

Format: Selection menu
Selection: **off**
Entry from the motor table active
ROTATING_WITH_Z1
Incremental rotary encoder with Z1 track
ROTATING_ENDAT_ADJUSTED
Absolute rotary encoder with EnDat interface (aligned)
LINEAR_ENDAT
Absolute linear encoder with EnDat interface
LINEAR_INCREMENTAL
Incremental linear encoder
ROTATING_ENDAT_NOT_ADJUSTED
Absolute rotary encoder with EnDat interface (nonaligned)
ROTATING_WITHOUT_Z1
Incremental rotary encoder without Z1 track
ROT_DIST_CODED_NOT_ADJUSTED
Incremental rotary encoder with distance-coded reference marks (not aligned)
LIN_DIST_CODED_NOT_ADJUSTED
Incremental linear encoder with distance-coded reference marks (unaligned)
DIG_ENDAT_2_2_ADJUSTED
Absolute rotary encoder with EnDat 2.2 interface (aligned)
DIG_ENDAT_2_2_NOT_ADJUSTED
Absolute rotary encoder with EnDat 2.2 interface (unaligned)
LIN_ENDAT_2_2
Absolute linear encoder with EnDat 2.2 interface
Default: off (value from the motor table active)
Access: LEVEL3
Reaction: RESET

Entries in the power module table (inverter.inv)

- **NAME:** Designation of the power module
- PWM: PWM frequency in [Hz] at which the power module is driven
- **S:** Switch position of the current sensor. Is required for the D series of HEIDENHAIN inverters (UM 1xx D) in order to use the higher currents named above at the lower PWM frequencies.
  Input 0 or 1
- **I-MAX:** Maximum current of the inverter output in [A]
- **I-Nom:** Rated current of the inverter output in [A]
- **U-Imax:** Current sensor voltage in [V] at I-MAX
- **I-N-DC**: Permissible continuous current in stationary rotating field or until F-DC is reached in [A]
- **T-DC:** Time constant, how long maximum current can be applied to a stationary synchronous motor in [s]
- **F-DC**: Lower motor base frequency down to which the motor can be loaded with I-N-DC in [Hz]
- **T-AC**: Cycle duration for the duty cycle S6-40% in [s]
- **F-AC:** Motor frequency from which I-MAX is permissible in [s]
- **T-IGBT:** Protection time of the IGBTs in [s]

Entries in the motor table (motor.mot)

- Motor model (TYPE)
  - UASM = Uncontrolled asynchronous motor (volts-per-hertz control mode)
  - SM = synchronous motor
  - ASM = asynchronous motor
  - LSM = linear motor
- Designation of motor (NAME)
- Operating mode (MODE)
- Rated current (I-N) in A
- Rated voltage (U-N) in V
- Rated speed (N-N) in min^{-1}
- Rated frequency (F-N) in Hz
- No-load voltage (U0) in V
- No-load current (I0) in A
- Stator resistance cold (R1) in mΩ
- Rotor resistance cold (R2) in mΩ
- Stator leakage reactance (XStr1) in mΩ
- Rotor leakage reactance (XStr2) in mΩ
- Magnetizing reactance (XH) in mΩ
- Upper speed X-H characteristic (N-XH) in min^{-1}
- Threshold speed for field weakening (N-FS) in min^{-1}
- Maximum speed (N-MAX) in min^{-1}
- Factor for X-H characteristic (%-XH)
- Factor for stalling torque reduction (%-K)
- Number of pole pairs (PZ)
- Temperature coefficient (TK) in Ω/K
- Line count of the motor encoder (STR)

- Encoder in use (SYS)
  - 0 = No speed encoder (volts-per-hertz control mode)
  - 1 = incremental rotary encoder with Z1 track
  - 2 = Absolute rotary encoder with EnDat interface (aligned)
  - 3 = Absolute linear encoder with EnDat interface
  - 4 = Incremental linear encoder
  - 5 = Absolute rotary encoder with EnDat interface (not aligned)
  - 6 = Incremental rotary encoder without Z1 track (one reference mark)
  - Only CC 424: 7 = Incremental rotary encoder with distance-coded reference marks (not aligned)
  - Only CC 424: 8 = Incremental linear encoder with distance-coded reference marks (not aligned)
- Counting direction of the motor encoder (DIRECT.)
- Maximum temperature (T-MAX) in °C
- Maximum current (I-MAX) in A
- Rated power output (P-N) in W
- Motor mass moment of inertia (J) in kgm^2
- Inductance of the series reactor (L) in µH
- Thermal time constant for direct current (T-DC) in seconds
- Lower thermal limit frequency (F-DC) in Hz
- Thermal time constant for alternating current (T-AC) in seconds
- Upper thermal limit frequency (F-AC) in Hz; above this frequency, the maximum current I-MAX applies
- Thermal time constant for winding (Tth1) in s
- Thermal resistance between winding and lamination (Rth1) in K/W
- Thermal time constant for lamination (Tth2) in s
- Thermal resistance between lamination and coolant (Rth2) in K/W

Series reactor

The inductance of the series reactor is calculated depending on the no-load current I0:

I0 < 26 A:

$$L = \frac{700\ \mu H \cdot 5000\ Hz \cdot U_Z}{f_{PWM} \cdot 600} - \frac{(X_1 + X_2) \cdot 1000}{2 \cdot \pi \cdot f_N}$$

I0 ≥ 26 A:

$$L = \frac{700\ \mu H \cdot 5000\ Hz \cdot U_Z}{23{,}1 \cdot I_0 \cdot f_{PWM}} - \frac{(X_1 + X_2) \cdot 1000}{2 \cdot \pi \cdot f_N}$$

- L: Inductance of the series reactor in µH
- f_{PWM}: PWM frequency [Hz]
- X_1: Stator leakage reactance [mΩ]
- X_2: Rotor leakage reactance [mΩ]
- f_N: Rated frequency [Hz]
- U_Z: DC-link voltage [V]
- I_0: No-load current [A]

A negative result means that there is no series reactor.

Note

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

The series reactor must fulfill the following specifications:

- The required inductance (per phase)
- Load capacity with the maximum spindle current
- The inductance even with the maximum spindle speed (operating frequency)

Determining data for synchronous motors

The motor data for synchronous motors are entered in the motor table after some conversions using the values from the motor data sheet of the respective manufacturer (here using the example of a SIEMENS motor).

Values in the HEIDENHAIN motor table	Values from the motor data sheet
TYPE: SM	Permanently excited synchronous motor
NAME: 1FT6044-4AF7	1FT6044-4AF7
MODE: 0	
Rated current I-N in [A_{eff}] winding I-N: 2.9	Data sheet value Inoml (100 K) IN = 2.9 A
Rated voltage U-N in [V_{eff}] interlinked U-N: 340	Calculation from data sheet values n_{noml}, k_E, R_{Str}, $I_{noml\ (100\ K)}$, L_D: $U\text{-}N = \sqrt{3} \cdot \sqrt{(U_e + U_r)^2 + U_x^2}$ $U_e = (n_{noml} / 1000) \cdot (k_E / \sqrt{3})$ $U_e = (3000 / 1000) \cdot (108 / \sqrt{3})$ $U_e = 187.06\ V_{eff\ L,N}$ $U_r = RStr \cdot I_{noml\ (100\ K)}$ $U_r = 3.05 \cdot 2.9$ $U_r = 8.85\ V_{eff\ L,N}$ $U_x = 2 \cdot \pi \cdot (n_{noml} / 60) \cdot PZ \cdot (LD / 1.5) \cdot I_{noml\ (100\ K)}$ $U_x = 2 \cdot \pi \cdot (3000 / 60) \cdot 2 \cdot (0.016 / 1.5) \cdot 2.9$ $U_x = 19.44\ V_{eff\ L,N}$ $U\text{-}N = \sqrt{3} \cdot \sqrt{(187{,}06 + 8{,}85)^2 + 19{,}44^2}$ $U\text{-}N = 341\ V_{eff\ L,L}$
Rated speed N-N in [min^{-1}] N-N: 3000	Data sheet value n_{noml} N-N = 3000 min^{-1}
Rated frequency F-N in [Hz] F-N: 100	Calculation from data sheet value n_{noml} F-N = (n / 60) · PZ F-N = (3000 / 60) · 2 F-N = 100 Hz
No-load voltage at rated speed U0 in [V_{eff}] interlinked U0: 324	Calculation from data sheet value n_{noml} and k_Et $U0 = (n_{noml} / 1000) \cdot k_E$ U0 = (3000 / 1000) · 108 $U0 = 324\ V_{eff\ L,L}$
No-load current I0 in [A_{eff}] winding I0: 3	Data sheet value $I_{0\ (100\ K)}$ I0 = 3 A_{eff}
Stator resistance at 20 °C R1 in [mΩ] at 20° C R1: 3050	Data sheet value R_{Str} R1 = 3050 mΩ

Values in the HEIDENHAIN motor table	Values from the motor data sheet
Rotor resistance at 20 °C R2 in [mΩ] at 20° C R2: 0	
Stator leakage reactance at F-N Xstr1 in [mΩ] Xstr1: 0	If nothing given, then zero.
Rotor leakage reactance at F-N Xstr2 in [mΩ] Xstr2: 0	
Magnetizing reactance XH for F-N at rated conditions XH in [mΩ] XH: 10052	Calculation from data sheet value L_D, n_{noml} XH = 2 · π · (n_{noml} / 60) · PZ · (L_D / 1.5) XH = 2 · π · (3000 / 60) · 2 · (0.016 / 1.5) XH = 6702 mΩ
Desaturation speed N-XH in [min^{-1}] N-XH: 0	
Rotational speed of beginning field weakening range N-FS [min^{-1}] N-FS: 0	
Maximum speed (mechanical) N-MAX in [min^{-1}] N-MAX: 7700	Data sheet value n N-MAX = 7700 min^{-1}
Saturation factor %-XH in % %XH: 100	
Stalling torque reduction factor %-K in % %-K: 100	
No. of pole pairs (half pole no. of motor) PZ PZ: 2	From data sheet value or model designation
Temperature coefficient of the stator winding TK in 1/K TK: 0.004	
Line count of the speed encoder: STR STR: 2048	From the mounted speed encoder
Type of encoder SYS: 1	Incremental rotary encoder with Z1 track
Counting direction DIRECT. DIRECT.: +	
Max. temperature of motor at temperature sensor T-MAX in [°C] T-MAX: 150	
Maximum motor current I-MAX in [A_{eff}] winding I-MAX: 7.5	Data sheet value I_{max} I-MAX = 11 A_{eff}

Values in the HEIDENHAIN motor table	Values from the motor data sheet
Rated power P-N in [W] P-N: 1350	Calculation from data sheet value n_{noml} and $M_{noml\ (100\ K)}$ $P\text{-}N = 2 \cdot \pi \cdot (n_{noml} / 60) \cdot M_{noml}$ $P\text{-}N = 2 \cdot \pi \cdot (3000 / 60) \cdot 4.3$ P-N = 1351 W
Motor mass moment of inertia J in [kgm^2] J: 0.0006	Data sheet value J_{mot} J = 0.00051 kgm^2
Inductance of the series reactor L in [mH] L: 0	
Thermal time constant for direct current T-DC in [s] T-DC: 2400	Calculation from data sheet value T_{th} $T\text{-}DC = T_{th} \cdot 60$ $T\text{-}DC = 40 \cdot 60$ T-DC = 2400
Lower thermal cutoff frequency F-DC in [Hz] F-DC: 0	
Thermal time constant for alternating current T-AC in [s] T-AC: 2400	Calculation from data sheet value T_{th} $T\text{-}AC = T_{th} \cdot 60$ $T\text{-}AC = 40 \cdot 60$ T-AC = 2400
Upper thermal cutoff frequency F-AC in [Hz] F-AC: 0	

6.18.2 Preparation

Proceed as follows:

- Check the wiring against the grounding diagram
- Acknowledgment of control-is-ready signal (see "Emergency stop monitoring" on page 680)
- Check the EMERGENCY STOP circuit by pressing the EMERGENCY STOP buttons and the EMERGENCY STOP limit switches.
- Determine the machine configuration using the documentation on hand. A basic configuration of the machine kinematics should exist before you start commissioning. Please refer to "Machine kinematics on lathes (as of NC software 548328-03)" on page 487 or "Machine kinematics for lathes (up to NC software 548 328-02)" on page 502 in this manual.
  The machine configuration must contain a parameter set for every axis. The machine parameters should be preassigned with initial values before commissioning.
- Create a PLC program for interfacing the control to the machine (use the PLC development software **PLCdesignNT**). A PLC basic program is available for the MANUALplus 620. Registered customers can download the current version of the PLC basic program from the "HESIS-Web Including Filebase" on the Internet (http://hesis.heidenhain.de).
  HEIDENHAIN recommends using the PLC basic program.
- Ensure that all axis drives are enabled by the PLC. Use the OLM, for example, (See "Actual status 1 of the axes (Ipo Act State 1)" on page 836) to check this before putting the machine into service.
- Before putting the machine into service, get familiar with the machine and the mechanical data of the individual axes.
- Ensure that the axes are located at noncritical positions and that they can be moved safely during adjustment.

DC-link voltage

The MANUALplus 620 uses the dc-link voltage to calculate the maximum motor voltage:

- Enter in **MP_ampBusVoltage** the dc-link voltage at the power module.

MP_ampBusVoltage

	DC-link voltage Uz
	Available from NCK software version: 597 110-03.
Format:	Numerical value
Input:	DC-link voltage U_z in [V]
	Regenerative inverter: 650 [V]
	Non-regenerative inverter: 565 [V]
Default:	650 [V]
Access:	LEVEL3
Reaction:	RESET

Power module model

▶ Define with the MP_ampPowerSupplyType parameter whether you are using a power module with or without power recovery:

MP_ampPowerSupplyType
Power stage with or without energy recovery
Available from NCK software version: 597 110-03.
Format: Selection menu
Selection: **without energy recovery:**
Power stage without energy recovery
with energy recovery:
Power stage with energy recovery
Default: without energy recovery
Access: LEVEL3
Reaction: RESET

Preliminary input values

▶ Enter the following temporary input values when you begin commissioning:

Machine parameters in the configuration editor and preliminary input value		**Meaning**
System		
CfgFilter		
typeFilter1:	Off	Type of 1st nominal position value filter
orderFilter1:	1	Order of 1st nominal position value filter
typeFilter2:	Position	Type of 2nd nominal position value filter
orderFilter2:	11	Order of 2nd nominal position value filter[a]
Channels		
ChannelSettings		
CH_NC		
CfgLaPath		
minPathFeed:	0	Minimum feed rate on the path
minCornerFeed:	0	Minimum feed rate at corners
maxPathJerk:	5	Maximum jerk on the path
maxG1Feed:	99,999	Max. machining feed rate
pathTolerance:	0.01	Tolerance for contour transitions
maxPathYank:	1000000	Maximum yank on the path

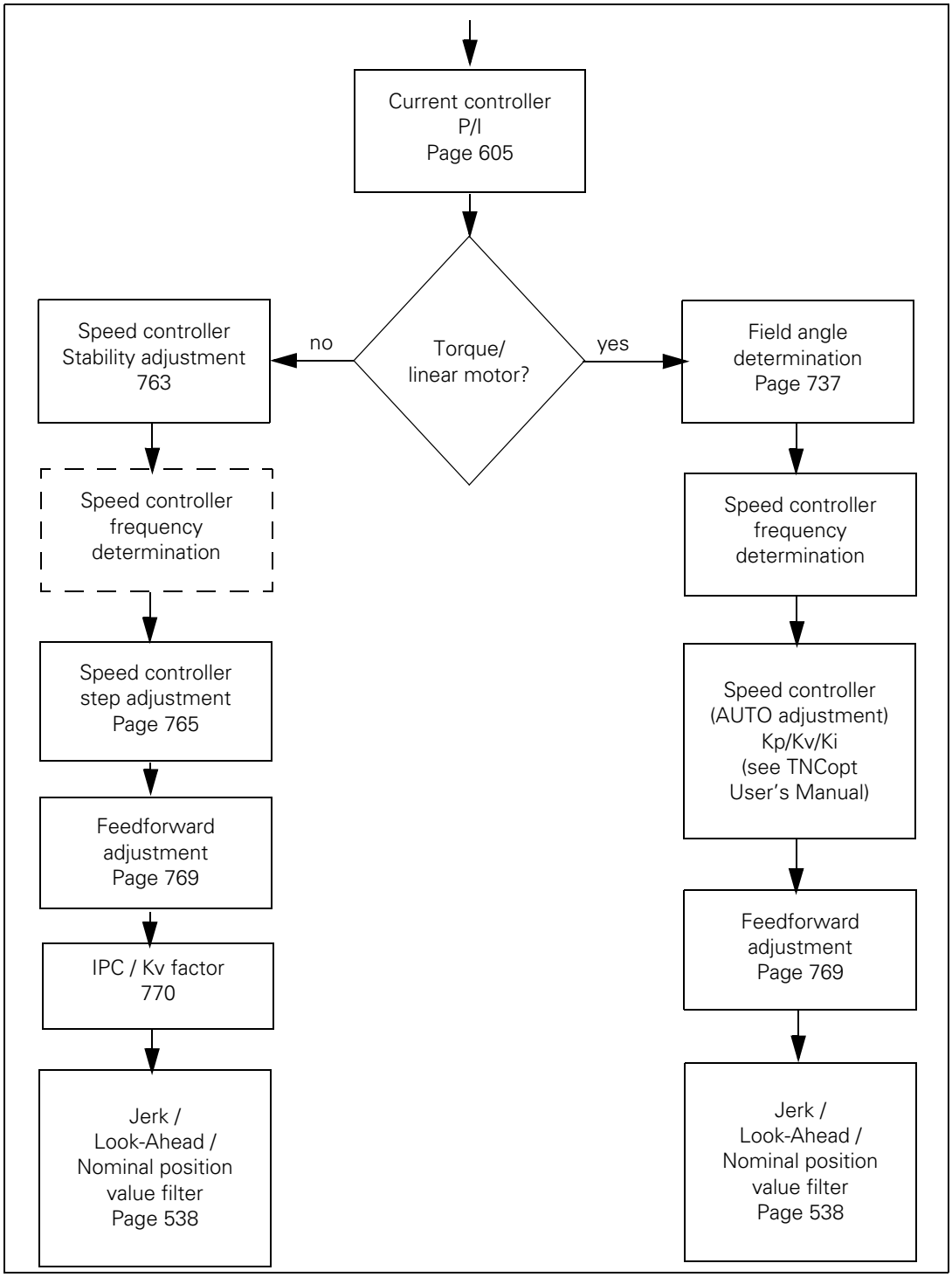
Machine parameters in the configuration editor and preliminary input value		Meaning
Axes		
ParameterSets		
[Key name of the parameter set]		Parameter set of axis
CfgAxisHardware		
signCorrActualVal:	off	Reverse counting direction of actual value
signCorrNominalVal:	off	Reverse counting direction of nominal value
CfgPosControl		
kvFactor:	15	k_V factor
servoLagMin1:	20	Following error limit
servoLagMax1:	20	Following error limit
servoLagMin2:	20	Following error limit
servoLagMax2:	20	Following error limit
feedForwardFactor:	1	100 % feedforward
controlOutputLimit:	1000	Control variable limit for pos. ctrl.
CfgFeedLimits		
maxAcceleration:	0.5	Maximum permissible acceleration
CfgReferencing		
refType:	None	No reference mark evaluation
CfgControllerAuxil		
driveOffLagMonitor:	off	Following-error monitoring
checkPosStandstill:	2	Standstill monitoring
CfgEncoderMonitor		
checkAbsolutPos:	off	Monitoring of distance code
checkSignalLevel:	on	Monitoring the encoder amplitude
movementThreshold:	0	Movement monitoring threshold
CfgControllerTol		
posTolerance:	0.01	Positioning window
CfgPositionFilter		
filter1Shape:	off	Shape of 1st nominal position value filter
filter1LimitFreq:	40	Cutoff frequency of 1st nominal position value filter
filter2Shape:	Triangle	Shape of 2nd nominal position value filter[a]
filter2LimitFreq:	40	Cutoff frequency of 2nd nominal position value filter
manualFilterOrder:	11	Order of the mean-value filter

a. Use filter order 7 only in combination with filter type MP_filter2Shape = Triangle!

Machine parameters in the configuration editor and preliminary input value		Meaning
Axes		
ParameterSets		
[Key name of the parameter set]		Parameter set of axis
CfgSpeedControl		
vCtrlPropGain:	0.5	Proportional factor of speed controller
vCtrlIntGain:	20	Integral factor of speed controller
vCtrlIntTime:	0	Limitation of integral factor
vCtrlDiffGain:	0	Derivative factor of speed controller
vCtrlFiltLowPassT:	0	PT_2 element of speed controller
vCtrlFiltDamping[1...5]:	0	Band-rejection for attenuation, filters 1 to 5
vCtrlFiltFreq[1...5]:	0	Band-rejection for center frequency, filters 1-5
vCtrlFiltType[1...5]:	0	Type of filter
CfgCurrentControl		
iCtrlPropGain:	0	Proportional factor of current controller
iCtrlIntGain:	0	Integral factor of current controller
CfgControllerComp		
compFriction0:	0	Friction compensation, low speed
compFrictionT1:	0	Delay of friction compensation
compFrictionT2:	0	Delay of stick-slip friction compensation
compFrictionNS:	0	Friction compensation at rated speed
compCurrentOffset:	0	Holding current
compAcc:	0	Acceleration feedforward control
compIpcT1:	0	IPC time constant T_1
compIpcT2:	0	IPC time constant T_2
compIpcJerkFact:	0	Following error in the jerk phase

6.18.3 Commissioning of digital axes

Overview:
Commissioning



TNCopt

Commissioning the digital control loops should be performed with the TNCopt commissioning and optimization software. Refer to the detailed information in the TNCopt User's Manual. It is available in PDF format from the HESIS-Web Including Filebase on the Internet (http://hesis.heidenhain.de) and supplied as an online help with TNCopt.

Current controller

The current controller **must** be adjusted with TNCopt. The speed and position control loops are opened when you adjust the current controller. You must therefore activate a special PLC commissioning program:

- Open the machine configuration.
- Enter the path to the PLC commissioning program in the **MP_pwmPgm** parameter of the System/Path/CfgPlcPath configuration object.
- Open the System/CfgHardware configuration object. Set the parameter **MP_currentControlAdjust** to the value **on**.
- Save the changes to the machine configuration.
- Restart the MANUALplus 620.

It suffices to program an **EM** (end module) in the PLC commissioning program.

The drive must be enabled externally and the MANUALplus 620 needs the "ready" signal.

Adjusting the current controller:

- Once the control has booted, acknowledge the **Power interrupted** message and switch the control voltage on.
- Open TNCopt and connect to the MANUALplus 620.
- Load the machine parameters from the control.
- Adjust the current controller as described in the TNCopt User's Manual.
- On completion of the adjustment, transfer the current controller parameters determined by TNCopt to the MANUALplus 620.

Note

If you use the CC 61xx or UEC 11x controller unit, always adjust all control loops (axes and spindle) **without** overshoot.

Commissioning speed and position controllers

- First deactivate the "current controller adjustment" mode. To do so, set the parameter **MP_currentControlAdjust** to the value **`off`**.
- Save the changes to the configuration data and restart MANUALplus 620 by turning the main switch off/on. Do not reboot the control, because rebooting would cause the general system error 320-0010.
- Activate a PLC program that is adapted to the machine.
- Acknowledge power interruption with the CE key and switch on the drives.

Note

It is possible that the MANUALplus 620 generates an error message regarding standstill monitoring. In this case, either change the **MP_signCorrActualVal** or the **MP_signCorrNominalVal** parameter.

More information and a block diagram on the subject of defining the traverse direction is provided at "Defining the traverse direction" on page 433.

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgAxisHardware	
signCorrActualVal	400001
signCorrNominalVal	400002

Commission the speed controller by following the steps described below:

Speed controller stability measurement

- Establish a connection between TNCopt and the control.
- In TNCopt, select the "speed-controller pulse" function.
- Load the machine parameters from the control.
- Disable all filters in the speed controller.
- Enter for PropGain (proportional factor of the speed controller) the value "1" and for IntGain (integral factor of the speed controller) the value "0". You may need to increase the starting value for IntGain (e.g. for hanging axes).
- Click the AUTO button and then the START button to start the measurement.

▶ Continue as described in the TNCopt User's Manual.

Note

If TNCopt reduces the P factor continually without finding a stable value, there might not be sufficient energy available for the speed controller's pulse output. In this case, modify the basic settings in TNCopt in such a way that a higher amplitude and a higher pulse width are achieved:

Settings/General/Speed Controller/Stability Measurement:

- Increase gain and/or
- Pulse width.

Example:

Increase the gain from 0.5 to 0.99 and the pulse width from 600 µs to 1800 µs.

Filter optimization:

During AUTO adjustment, a dominant resonance frequency might occur. If this is the case, you should abort the AUTO adjustment, and dampen this specific resonant frequency with a filter (band-rejection filter with 3 to 9 dB of damping).

▶ Repeat the AUTO adjustment and continue optimizing the filter until the AUTO adjustment has been successfully completed.

▶ Only repeat the filter optimization for as long as Kp clearly increases and the rise time is approx. 3 ms.

▶ Set as few filters as necessary.

Note

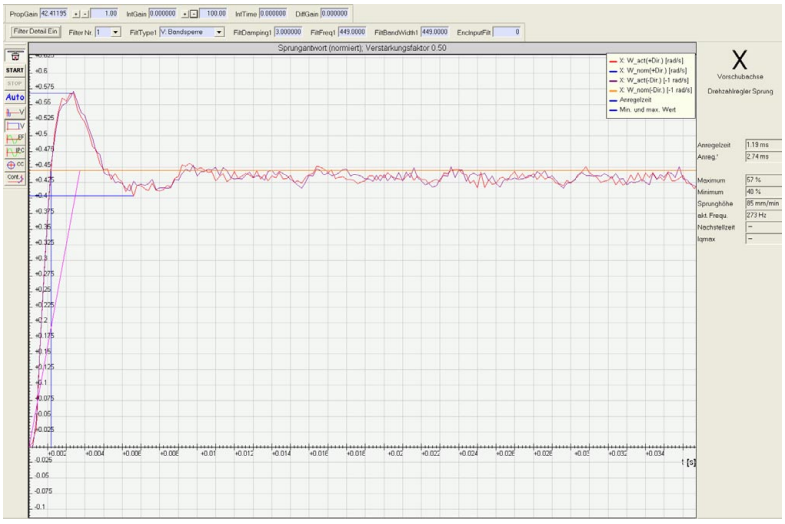
The rise time is machine-dependent. The 3 ms indicated above are not a standard value! On some types of machine, the best results might by achieved with a rise time of 10 ms and higher.

▶ Transfer the determined filter parameters to the control.

Speed controller step adjustment

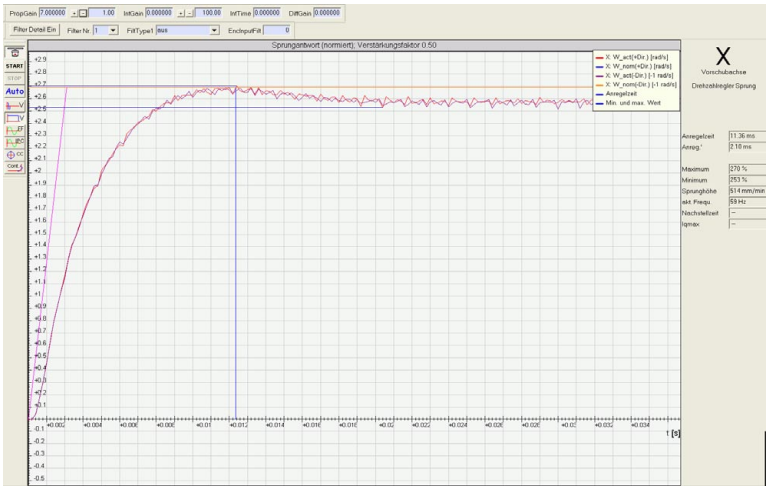
- In TNCopt, select the "speed-controller step adjustment" function.
- The P factor and the filter parameters are transferred from the speed controller stability measurement performed previously.
- Set the I factor (IntGain) to a small value:
  Value 0 for horizontal axes
  Value 100 for vertical axes, e.g. Z axis
- Now check the P factor of the speed controller, which you determined previously. This factor might be too high and need to be reduced. Carry out a measurement and continue reducing the P factor until you:
  - obtain a low oscillation tendency
  - there is (almost) no overshoot
  - obtain a rise time of approx. 3 ms (incl. I factor)
  - obtain a "smooth" step response.
- Click the START button (AUTO is disabled)

Example of a P factor adjusted too high



- Reduce the P factor and disable the filters you do not need (filters for a very high P factor)
- Repeat the measurement to check the P factor.

Example of a correctly adjusted P factor of a speed controller:

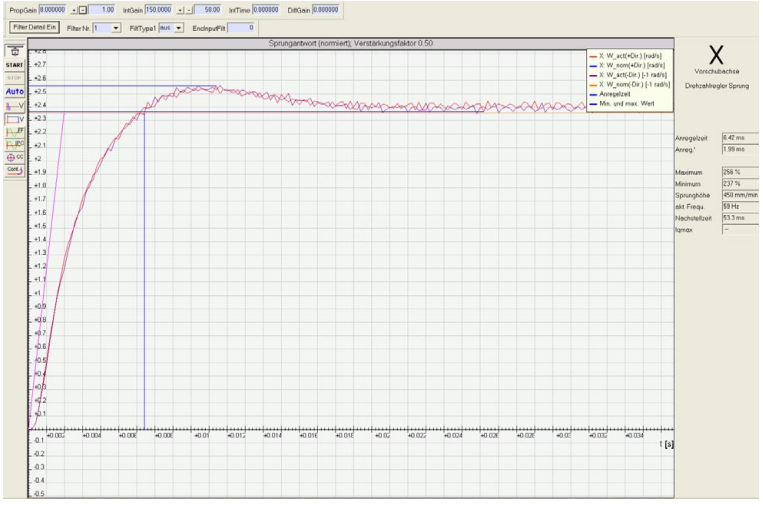


When checking and correcting the P factor is complete, continue with the speed controller step adjustment:

- Click the AUTO button.
- Click the START button.
- TNCopt now performs several measurements and automatically determines the value for the I factor of the speed controller.

As an alternative, you can also determine the I factor manually:

- Disable the AUTO button.
- Increase the I factor until a step response with just a minimal overshoot results. Example:



- Transfer the determined machine parameters to the control.

The CC 61xx and UEC 11x controller units offer the possibility to use a powerful multifunction filter for the resonance frequencies. For more information on the multifunction filter, see „Filters in the speed controller and position controller when using the CC 61xx and CC 424" on page 579

Sequence for traversing the reference marks

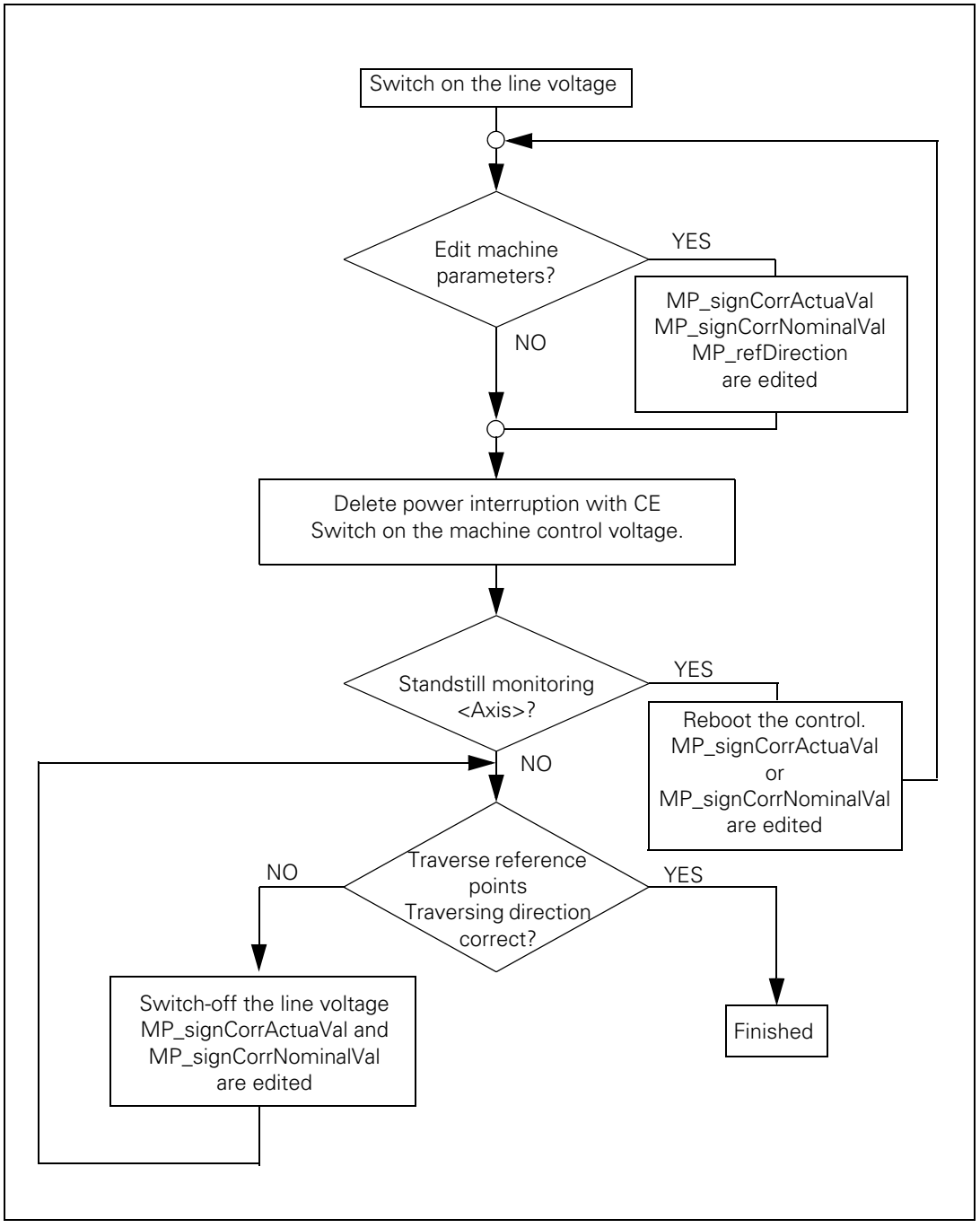
Commission the speed controller by following the steps described below:

Check the counting and traversing direction

More information and a block diagram on the subject of defining the traverse direction is provided at "Defining the traverse direction" on page 433.

Settings in the configuration editor	MP number
Channels ChannelSettings CH_NC CfgChannelAxes **refAxis** **refAllAxes**	 200303 200304
Axes ParameterSets [Key name of the parameter set] CfgAxisHardware **signCorrActualVal** **signCorrNominalVal**	 400001 400002

- Specify via the channel-dependent configuration object **CfgChannelAxes** the sequence in which the reference marks are to be traversed. In **MP_refAllAxis,** you specify whether all axes are to be referenced in the sequence defined in **MP_refAxis,** or whether the reference point in these axes is to be traversed by pressing the axis-direction keys.
- Use the following flow chart to check the traversing direction:


Switch on the line voltage
Edit machine parameters?
YES
MP_signCorrActuaVal
MP_signCorrNominalVal
MP_refDirection
are edited
NO
Delete power interruption with CE
Switch on the machine control voltage.
Standstill monitoring <Axis>?
YES
Reboot the control.
MP_signCorrActuaVal
or
MP_signCorrNominalVal
are edited
NO
Traverse reference points
Traversing direction correct?
NO
YES
Switch-off the line voltage
MP_signCorrActuaVal and
MP_signCorrNominalVal
are edited
Finished

Adjustment of feedforward parameters

The feedforward parameters are adjusted with TNCopt. Please refer to the detailed information in the TNCopt User's Manual.

Note

The Generate NC programs function must not be used on the MANUALplus 620, because this function transfers milling programs to the control. The ff_100_[axis].nc programs required for adjusting the feedforward parameters are already contained in the ncps program folder of the MANUALplus 620.

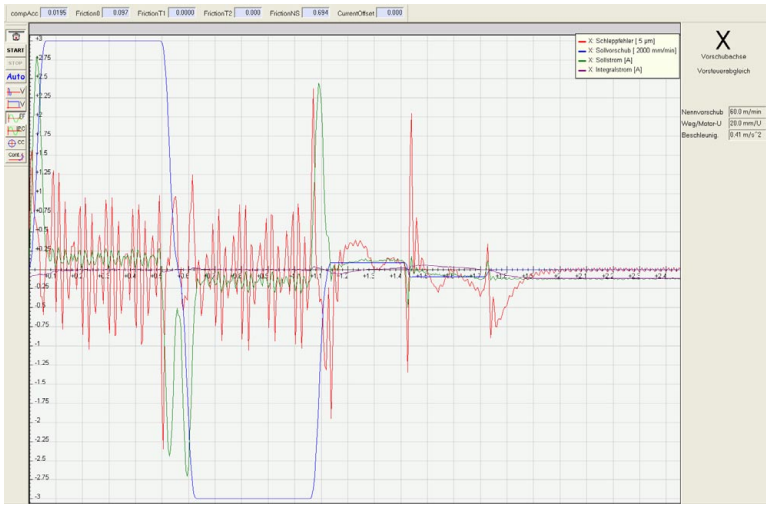
Default settings in the configuration editor of the MANUALplus 620:

Machine parameters in the configuration editor and input value:		**Meaning:**
Axes ParameterSets [Key name of the parameter set] CfgFeedLimits **maxAcceleration**:	0.5	Acceleration [m/s^2]
Channels ChannelSettings CH_NC CfgLaPath **maxPathJerk**:	5	Maximum jerk on the path

- Enter the default values according to the table above and save the changes to the machine configuration.
- Switch to the **Program Run** mode of operation
- Depending on the axis you want to optimize, open the NC program **TNC:\nc_prog\ncps\ff_100_[axis].nc**
  e.g. for the X axis: **ff_100_x.nc**
- Click the AUTO and START buttons in TNCopt.
- Press the NC START button on the machine to start the NC program.
- TNCopt now determines the feedforward parameters automatically:
  - Friction (low speed)
  - Friction (high speed)
  - Acceleration
  - Holding current for vertical axes

  by minimizing the integral current

▶ The integral current should now be almost 0:



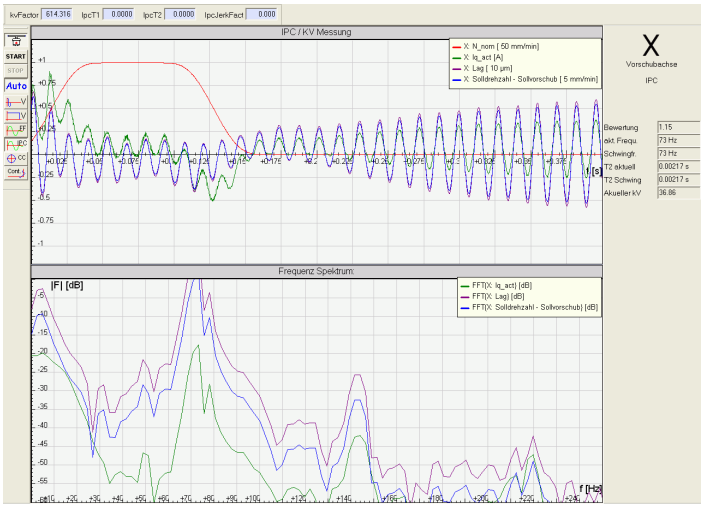
▶ Transfer the determined machine parameters to the control.

IPC / Kv factor

A prerequisite for the correct adjustment of the IPC is that the acceleration feedforward (**MP_compAcc** parameter) has realistic values. It is therefore essential that you adjust the acceleration feedforward control with TNCopt, as described above, before adjusting the IPC and kv factor.

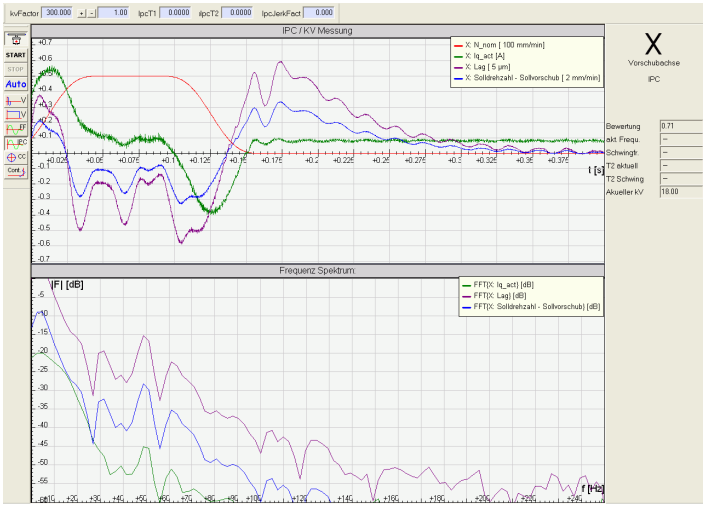
The IPC and the kv factor are adjusted with TNCopt. Continue as described in the TNCopt User's Manual to set the IPC and determine the kv factor.

▶ TNCopt automatically increases the kv factor until a sustained oscillation results:



▶ The values for IpcT1 and IpcT2 are automatically determined by TNCopt.

▶ Correctly adjusted kv factor:



Note

If the measurement cannot be successfully completed, the trigger threshold might be too high. In this case, change the settings in TNCopt under:

Settings/Optimize/IPC=> Trigger CC

In "Trigger threshold," enter a lower value, e.g. 5.

For more information on the IPC, See "IPC, holding torque, following error in the jerk phase" on page 591.

Temporary machine parameters in the configuration editor and input value:	
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgFeedLimits	
maxAcceleration:	0.5
CfgLaAxis	
axJerk:	500
Channels	
CH_NC	
CfgLaPath	
maxPathJerk:	999999
maxPathJerkHi:	999999

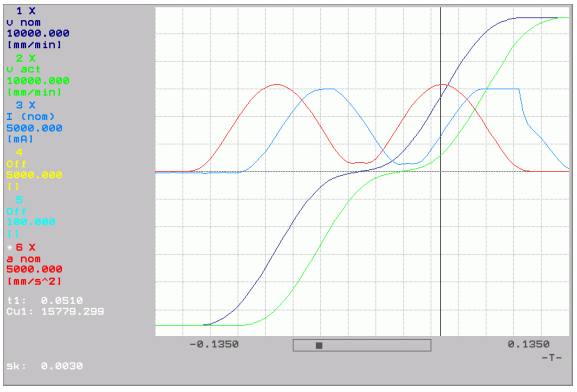
The following machine parameters influence whether the maximum possible acceleration can be reached:

- Maximum permissible axis acceleration **(MP_maxAcceleration)**
- Maximum jerk on the path **(MP_maxPathJerk)**
- Maximum jerk on the path during rapid traverse **(MP_maxPathJerkHi)**

▶ Clamp an object of maximum permissible weight on the machine table.

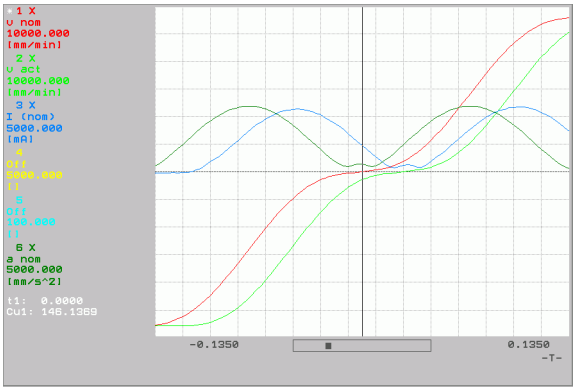
▶ Increase the acceleration (**MP_maxAcceleration** parameter) step by step by a value of 0.5.

▶ The maximum possible acceleration is achieved at the point where the current I (nom) reaches the limitation. In the example below:
15 A motor current at an acceleration of 16 m/s^2

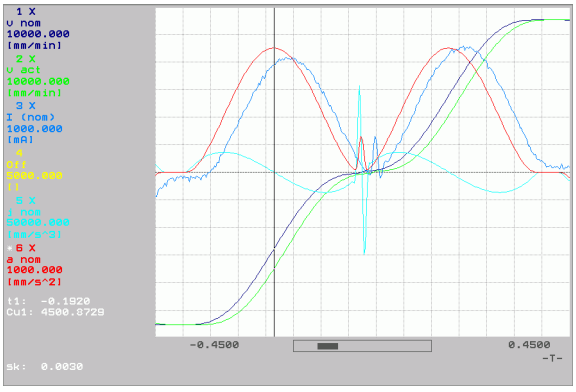


▶ Limit the maximum acceleration to approx. 80% of the maximum motor current, e.g.:
15 A · 80 % = 12 A

▶ Reduce the acceleration and check the result with the oscilloscope until a maximum motor current of I = 80% · Imax is obtained. In the example below, an acceleration of 12 m/s^3 results:



Example: Selected acceleration = 16 m/s^2

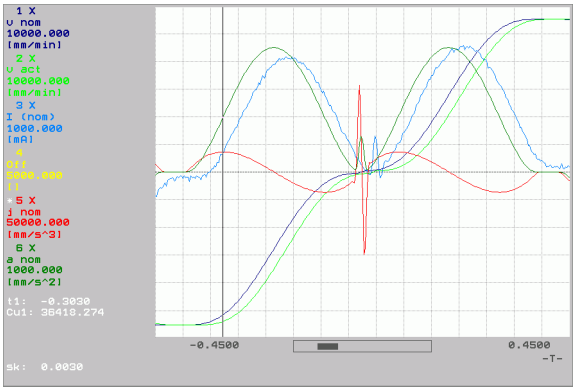


Result: The acceleration actually achieved is only 4.5 m/s^2.

Possible causes:

- Speed too low
- Traverse range too short

▶ Use the integrated oscilloscope to check whether the configured axis jerk (e.g. **MP_axJerk** = 500 m/s^3) is actually reached.



Result: The jerk actually achieved is only 364 m/s^3.

Possible reasons

- Speed too low
- Traverse range too short

Setting the traverse range

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the parameter set]	
CfgPositionLimits	
swLimitSwitchPos	400501
swLimitSwitchNeg	400502

To define the software limit switches, proceed as follows:

- Traverse the reference points
- Enter **`NominalValue`** in the **Axis display** user parameter to display the nominal position with respect to the machine zero point.
- The position displays now show the distance to the machine zero point, without taking the tool lengths or zero point shifts into account.
- 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 with the algebraic signs.
- Enter the noted values in the machine parameters **MP_swLimitSwitchPos** or **MP_swLimitSwitchNeg**.
- Enter **`Default`** in the **Axis display** user parameter to return the position displays to the display of the tool tip position with respect to the workpiece zero point.

Note

You can enter different traverse ranges. You must define a separate parameter set per axis and traverse range. The individual traverse ranges are activated by switching the parameter sets (e.g. by PLC).

Activating monitoring functions

The monitoring functions of the MANUALplus 620 must be activated now.

Note

To ensure that the monitoring functions of the MANUALplus 620 become effective at the right moment, you must enter meaningful values.

- Activate the position monitoring (see "Position monitoring" on page 646). You define two limits in the machine parameters for the position monitoring: one for operation with following error, and one for operation with velocity feedforward control.
- Configure the movement monitoring (See "Movement monitoring" on page 650).
- Configure the standstill monitoring (See "Standstill monitoring" on page 652).

Note

Adjust the input values to the machine dynamics.

6.18.4 Commissioning of analog axes

General information

The MANUALplus 620 features a Commissioning Wizard for analog axes.

The Wizard guides you step by step through the commissioning of any axis parameter set. Press the **ANALOG AXIS** soft key on the opening screen of the internal oscilloscope to start the Commissioning Wizard. The soft key is grayed out as a default. You must enable the commissioning aid with the code number **13852**. The parameter set to be optimized must exist in the system, and is selected from a dialog window.

Soft key	Function
ANALOGE ACHSEN	Grayed out as a default; selectable after the code number 13852 has been entered. Starts the Commissioning Wizard for analog axes.

Attention

Default values are automatically assigned to the axis parameters before commissioning. The control reports this in a message window. The axis is not operated with position control as long as these temporary initial parameter values are in effect. The following error that results is not eliminated. **Hanging axes require a 100 % compensation for weight. Ensure that hanging axes are adequately supported.**

You can define the following machine parameters with the aid of the Commissioning Wizard:

Soft key	Function
VORZEICHEN	Ascertain the algebraic sign of the axis (**MP_signCorrActualVal** or **MP_signCorrNominalVal**)
RICHTUNG ACHSE	Ascertain the axis traverse direction (**MP_signCorrActualVal** or **MP_signCorrNominalVal**)
ABGLEICH MAXFEED	Ascertain the velocity at an analog voltage of 9 V (**MP_maxFeedAt9V**)
MAX. BESCHL.	Ascertain the maximum acceleration of the axis (**MP_maxAcceleration**)
KV-FAKTOR	Ascertain the k_V factor of the axis (**MP_kvFactor**)
BESCHL. VORST.	Ascertain the acceleration feedforward control of the axis (**MP_accForwardFactor**) Not possible with MANUALplus 620 (no HSC filter)!

Press the **CANCEL** or **LEVEL BACK** soft key to leave the commissioning tool at any point. The temporary parameter values or ascertained values are then discarded, and the previous values are restored.

Note

The last step of the commissioning wizard for determining acceleration feedforward cannot be performed on the MANUALplus 620, because there is no HSC filter. The commissioning wizard must therefore be canceled after determining the kv factor. As a result, all values determined through the adjustment are discarded by the control.
It is therefore necessary to write down all values determined through the adjustment, to enter them manually in the corresponding machine parameters and to save them.

Preparation

Proceed as follows:

- Check the wiring against the grounding diagram
- Acknowledgment of control-is-ready signal (see "Emergency stop monitoring" on page 680)
- Check the EMERGENCY STOP circuit by pressing the EMERGENCY STOP buttons and the EMERGENCY STOP limit switches.
- Determine the machine configuration using the documentation on hand. The machine configuration must contain a parameter set for every axis. The machine parameters should be preassigned with initial values before commissioning.
- Create a PLC program for interfacing the control to the machine (use the PLC development software **PLCdesignNT**). A PLC basic program is available for the control. It is already installed on the control when the control is shipped. In addition, registered customers can download the current version of the PLC basic program from the HESIS-Web Including Filebase on the Internet (hesis.heidenhain.de).
  HEIDENHAIN recommends using the PLC basic program.
- Ensure that all axis drives are enabled by the PLC. Use the OLM, for example, (See "Actual status 1 of the axes (Ipo Act State 1)" on page 836) to check this before putting the machine into service.
- Before putting the machine into service, get familiar with the machine and the mechanical data of the individual axes.
- Ensure that the axes are located at noncritical positions and that they can be moved safely during adjustment.

Temporary input values

▶ Enter the following temporary input values when you begin commissioning:

Machine parameters in the configuration editor and preliminary input value		Meaning
Channels		
ChannelSettings		
CH_NC		
CfgLaPath		
maxPathJerk:	1	Maximum jerk on the path
maxG1Feed:	99,999	Max. machining feed rate
pathTolerance:	0.01	Tolerance for contour transitions
Axes		
ParameterSets		
[Key name of the parameter set]		Parameter set of axis
CfgAxisHardware		
signCorrActualVal:	off	Reverse counting direction of actual value
signCorrNominalVal:	off	Reverse counting direction of nominal value
CfgPosControl		
kvFactor:	0	k_V factor
servoLagMin1:	20	Following error limit
servoLagMax1:	20	Following error limit
servoLagMin2:	20	Following error limit
servoLagMax2:	20	Following error limit
feedForwardFactor:	1	100 % feedforward
controlOutputLimit:	1000	Control variable limit for pos. ctrl.
CfgFeedLimits		
maxAcceleration:	0.5	Maximum permissible acceleration
CfgControllerAuxil		
driveOffLagMonitor:	off	Following-error monitoring
checkPosStandstill:	2	Standstill monitoring
CfgEncoderMonitor		
checkAbsolutPos:	off	Monitoring of distance code
checkSignalLevel:	on	Monitoring the encoder amplitude
movementThreshold:	0	Movement monitoring threshold
CfgControllerTol		
posTolerance:	0.01	Positioning window
CfgPositionFilter		
filter2Shape:	Triangle	Shape of 2nd nominal position value filter
manualFilterOrder:	11	Order of the mean-value filter

Adjusting the servo amplifier

Please note:

Note

Adjust the servo amplifier before optimizing the position controller. For instructions on adjustment, refer to the information given by the manufacturer of your servo amplifier.

- Adjust the offset according to the information given by the drive manufacturer.
- Adjust the proportional (P) component and the integral-action (I) component of the speed controller at the servo amplifier.
- Check the polarity of the tachometer signal of the drive by using a battery box, for example.

HEIDENHAIN recommends:

Use a voltage of 9 V for rapid traverse to ensure optimum utilization of the voltage range of +/– 10 V and to attain optimum control loop performance for the axis. The axis velocity to be expected (in [mm/min]) is defined in machine parameter **MP_maxFeedAt9V.** Enter the rapid traverse rate in the machine parameter.

Note

A servo amplifier that has been adjusted according to the information given by the manufacturer is the basic prerequisite for putting the machine into service.

General information

Analog axis feedback control is based on the following formula:

Value, parameter	Unit	Description
U_{out}	V	Output voltage (analog nominal speed value)
P_{err}	mm	Following error (servo lag)
kvFactor	1/s	Kv factor (proportional component of position controller)
V_{nom}	mm/min	Nominal velocity
feedForwardFactor		Factor for velocity feedforward control
A_{nom}	m/s^2	Nominal acceleration
accForwardFactor	(s)	Factor for acceleration feedforward control
maxFeedAt9V	mm/min	Assumed velocity of the axis at 9 V

$$U_{out} = (P_{err} \cdot \boldsymbol{kvFactor} + \frac{V_{nom}}{60} \cdot \boldsymbol{feedForwardFactor} + 1000 \cdot A_{nom} \cdot \boldsymbol{accForwardFactor}) \cdot \frac{9V \cdot 60}{\boldsymbol{maxFeedAt9V}}$$

The temporary input values result in the following reduced formula for the output voltage:

$$U_{out} = (P_{err} \cdot 0 + V_{nom} \cdot 1 + A_{nom} \cdot 0) \cdot \frac{9V}{\boldsymbol{maxFeedAt9V}}$$

Therefore:

$$U_{out} = V_{nom} \cdot \frac{9V}{\boldsymbol{maxFeedAt9V}}$$

Note

Due to the temporary input values the axis is only in a speed control loop, but not in a position control loop, during operation. The resulting following error is not eliminated. For this reason, higher values were defined with the temporary input values in the **MP_servoLagMin1, MP_servoLagMin2,** MP_servoLagMax1 and MP_servoLagMax2 parameters.

Danger

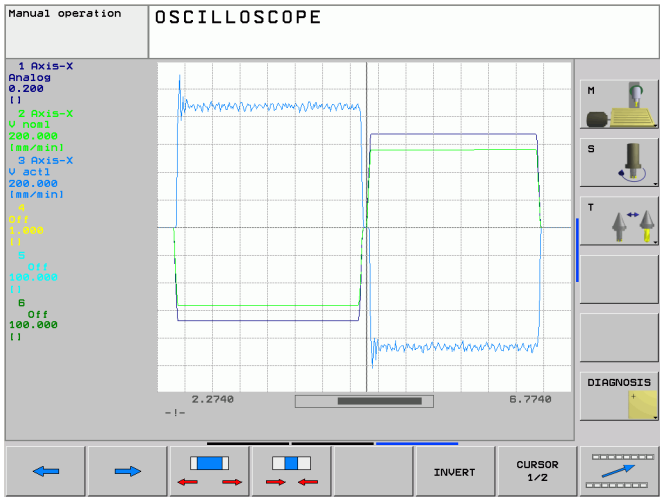
Due to the temporary machine parameters, the position control loop is open at the beginning of commissioning!
Hanging axes need 100 % weight compensation.
Ensure that hanging axes are adequately supported.

1. Checking the counting direction

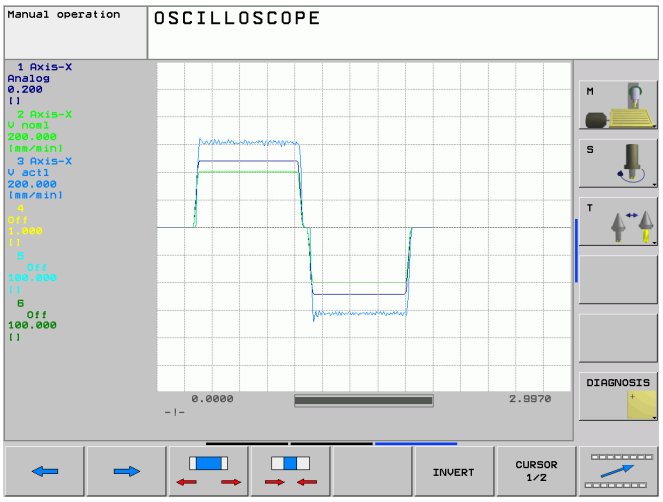
Settings in the configuration editor:	
Axes ParameterSets [Key for parameter set] CfgAxisHardware **signCorrActualVal** **signCorrNominalVal**	

To check the counting direction of the position measuring system, proceed as follows:

- Switch on the machine.
- Select the following operating mode without crossing over the reference marks:
  Manual Operation
- Switch to the **Oscilloscope** mode of operation.
- Set the following values in the oscilloscope by pressing the **SELECTION** soft key:
  Display mode: YT
  Sampling time: IPO clock
  Channel 1: Analog
  Channel 2: v nom
  Channel 3: v act
  Trigger: Free run
- Press the **OSCI** soft key to switch to the curve representation.
- Press the **START** soft key to start recording.
- Press the axis-direction key of each axis to be checked.
- Press the **STOP** soft key to stop recording.



Incorrect algebraic sign



Correct algebraic sign

- If **v nom** and **v act** do not lie in the same direction on the oscilloscope, you must change either **MP_signCorrActualVal** or **MP_signCorrNominalVal.**

Reversal of traverse direction

If the axis does not move in the expected direction after you have pressed the respective axis-direction key (e.g. X axis moves in negative direction although you have pressed the X+ key), you can reverse the traversing direction.

- Invert the two values entered in the parameters MP_signCorrActualVal and MP_signCorrNominalVal.

Settings in the configuration editor:	
Axes ParameterSets [Key for parameter set] CfgAxisAnalog **maxFeedAt9V**	

The aim of speed adjustment is to achieve that the output nominal speed value is equal to the really measured actual speed value ($V_{nom} = V_{act}$).

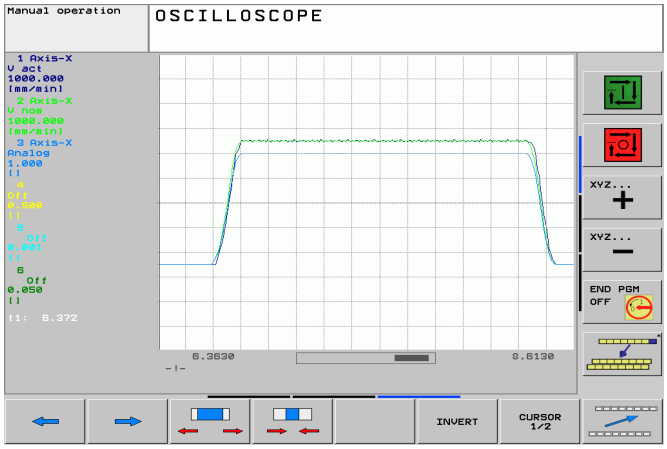
Determine whether the nominal speed value (V_{nom}) differs from the actual value (V_{act}) on the machine. Proceed as follows:

- Switch on the machine.
- Select the following operating mode without crossing over the reference marks:
  `Manual Operation`
- Switch to the **`Oscilloscope`** mode of operation (code number **688379**).
- Set the following values in the oscilloscope by pressing the **`SELECTION`** soft key:
  `Display mode: YT`
  `Sampling time: IPO clock`
  `Channel 1: Analog`
  `Channel 2: V nom`
  `Channel 3: V act`
  `Trigger: Free run`

Note

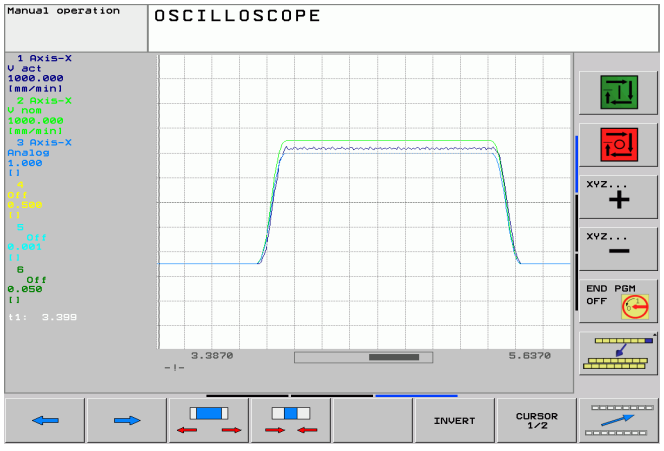
In the internal oscilloscope, the **`analog`** signal corresponds to the output voltage U_{out} (= analog speed command signal) at connector X8.

- Press the **`OSCI`** soft key to switch to the curve representation.
- Press the **`START`** soft key to start recording.
- Press the axis-direction key of each axis to be checked.
- Press the **`STOP`** soft key to stop recording.
- Compare the values measured for V_{nom} and V_{act} with each other.
- Ideally, your oscilloscope measurement should look similar to this:



Vnom = Vact, speed adjusted correctly

However, it may occur that the nominal value differs from the actual value:



Vnom ¼ Vact, speed adjusted incorrectly

In this case, you should at first try to eliminate the difference by using the possible settings of the servo amplifier (please note the information given by the manufacturer). If this fails, refer to the information given below on how to adjust the value in **MP_maxFeedAt9V.**

Due to the temporary input values, the following formula applies to the nominal output voltage at connection X8:

$$U_{out} = V_{nom} \cdot \frac{9V}{\mathbf{maxFeedAt9V}}$$

Therefore, **MP_maxFeedAt9V** is determined as follows:

$$\mathbf{maxFeedAt9V} = V_{act} \cdot \frac{9V}{U_{out}}$$

Perform a measurement with the internal oscilloscope to determine the current difference between **MP_maxFeedAt9V** and the connected drive. Use the parameter formula described above to determine the correct value.

Proceed as follows:

- Switch to the **Oscilloscope** mode of operation.
- Set the following values in the oscilloscope by pressing the **SELECTION** soft key:
  Display mode: YT
  Sampling time: IPO clock
  Channel 1: Analog
  Channel 2: v nom
  Channel 3: v act
  Trigger: Free run
- Press the **OSCI** soft key to switch to the curve representation.
- Press the **START** soft key to start recording.
- Press the axis-direction key of each axis to be checked.
- Press the **STOP** soft key to stop recording.
- Select the values **V nom** and **Analog** by using the arrow keys and write down the measured values, which are displayed at the left side of the oscilloscope (**Cu1:**).
- Enter the two values in the formula for determining the parameter **MP_maxFeedAt9V.**

Example:

The internal oscilloscope measured the following values on the machine:

- **Analog** = 1.21 V
- **v act** = 1517 mm/min
- **v nom** = 1008 mm/min

This measurement makes clear that the actual speed value **v act** differs from the nominal speed value **nom**. The difference can be eliminated by using the formula mentioned above:

$$\mathbf{maxFeedAt9V} = 1517\frac{mm}{min} \cdot \frac{9V}{1,21V} = 11283\frac{mm}{min}$$

- Enter the calculated value in the parameter **MP_maxFeedAt9V** and check the calculated value by performing a measurement with the internal oscilloscope.

3. Determining the acceleration

▶ Clamp an object of maximum permissible weight on the machine table.

Note

Write down the current input values set in **MP_CfgPositionFilter.** You will need to enter these values again after the acceleration has been optimized.

▶ Now enter the temporary machine parameters listed in the table.

Goal of the temporary input values: A jump in the nominal value is output to the axis.

Machine parameters in the configuration editor	Temporary input value	Meaning
Axes		
ParameterSets		
[Key for parameter set]		
CfgPosControl		
kvFactor	0	k_v factor
CfgFeedLimits		
maxAcceleration	999999	Maximum acceleration
CfgPositionFilter		
filter2Shape	Off	Shape of 2nd nominal position value filter
manualFilterOrder	1	Order of mean-value filter in manual mode

Attention

Ensure that the transmitted nominal-value step does not cause any damage to the machine mechanics. It may be necessary to determine the acceleration by careful approximation.

▶ Switch to the **Oscilloscope** mode of operation.

▶ Set the following values in the oscilloscope by pressing the **SELECTION** soft key:
Display mode: YT
Sampling time: IPO clock
Channel 1: Analog
Channel 2: v nom
Channel 3: v act
Trigger: Free run

▶ Press the **OSCI** soft key.

▶ Press the **START** soft key to start recording.

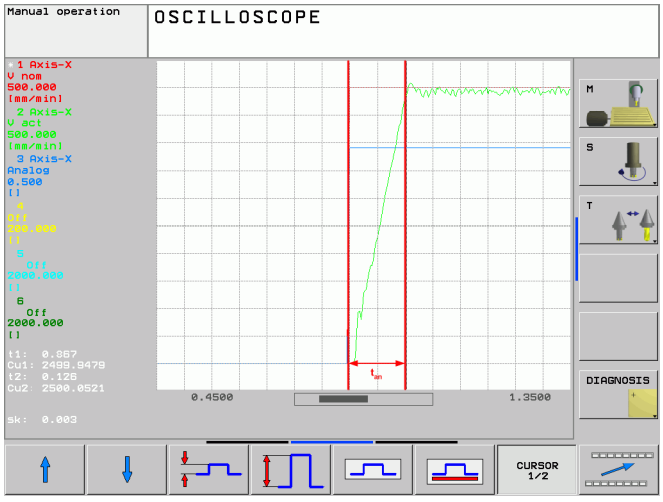
▶ Press the rapid traverse key together with the axis-direction key to output the maximum possible feed rate.

▶ Press the **STOP** soft key to stop recording.

▶ From the step response of the actual velocity **(V act)** you determine the maximum possible acceleration (incl. 10% safety margin).

$$a = \frac{Vnom}{t_{an} \cdot 66\,000}$$

Value, parameter	Unit	Description
a	m/s^2	Acceleration
Vnom	mm/min	Nominal velocity
t_{on}	s	Rise time



Example:

The internal oscilloscope measured the following rise-time value on the machine:

t_{an} = 0.125 s

The nominal speed **v nom** (can be read from the oscilloscope) is a machine-specific parameter, and was determined to be 5000 mm/min in this example.

Calculation of acceleration:

$$a = \frac{5000 \frac{mm}{min}}{0{,}125\ s \cdot 66\,000} = 0{,}61 \frac{m}{s^2}$$

- Enter the calculated value in the parameter **MP_maxAcceleration** and check the calculated value by performing a measurement with the internal oscilloscope.

Attention

Every transmitted nominal value step causes high stress to the machine mechanics. Now reset the temporary input values in **CfgPositionFilter** to the initial values before continuing commissioning.

4. Determining the k_V factor

Machine parameters in the configuration editor	**Temporary starting value**	**Meaning**
Axes ParameterSets [Key for parameter set] CfgPosControl **kvFactor**	15	k_V factor

Note

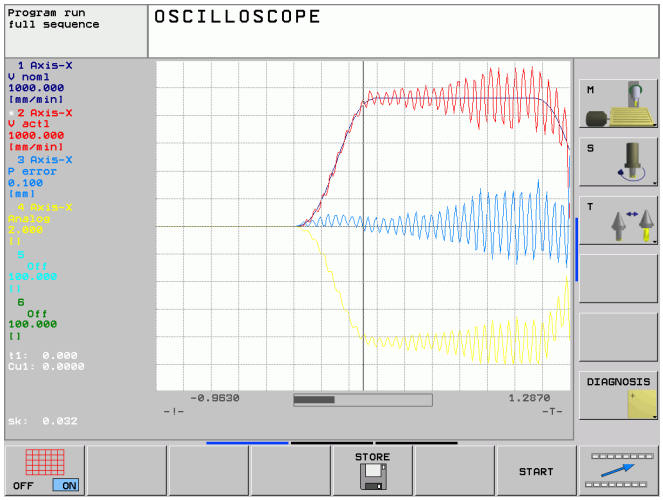
If the starting value entered causes the control loop to oscillate, the value must be reduced.

- Enter the following test program:

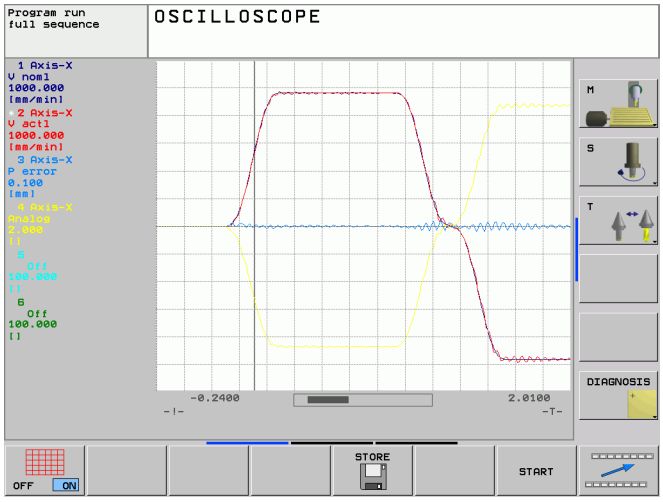
```
BEARBEITUNG [MACHINING]
N 1 G94 F2000
N 2 G1 X100
N 3 G1 X0
N 4 M99
END
```

The test program should be structured so that the axis reaches the nominal velocity.

- Run the program at high speed (feed-rate override = 100 %).
- With the integrated oscilloscope, record the nominal feed rate **(v nom),** the actual feed rate **(v act),** and if desired, the servo lag **(s diff)** as well.
- Perform the first measurement with the temporary k_V factor (15).
- Increase the k_V factor until the oscillation limit is reached.
- Calculate the starting value of the **MP_kvFactor** with the following formula:
  MP_kvFactor = <determined value of the oscillation limit> · 0.5



kv factor too high, axis oscillates



Axis has reached the oscillation limit

5. Determining the jerk

The following machine parameter will be optimized now:

Settings in the configuration editor:	
NCchannel ChannelSettings CH_NC CfgLaPath **maxPathJerk**	

- Run the test program described previously in "Determining the k_V factor."

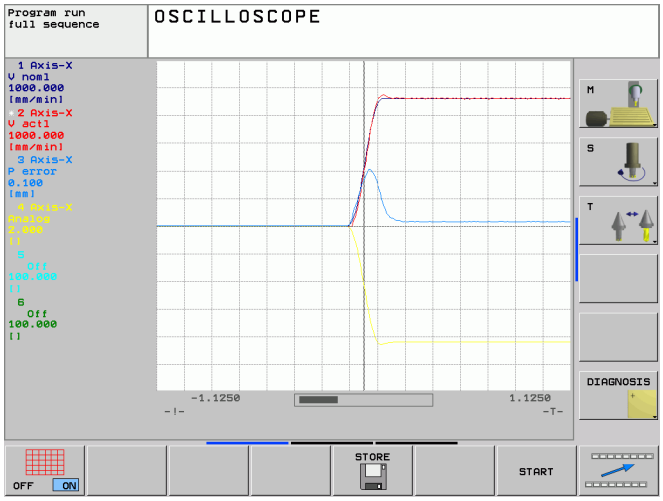
Note

Depending on the position of the axis slide on the ball screw, the axis can have different mechanical properties. Therefore you should repeatedly perform the following measurement several times in a row at different positions within the traverse range.

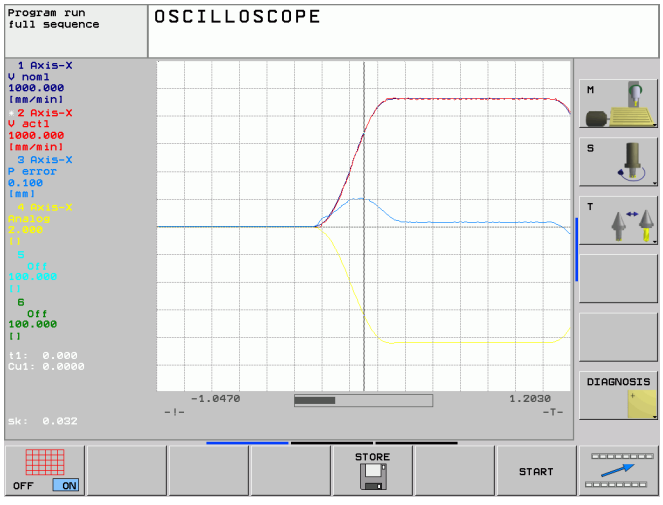
- Run the program at high speed (feed-rate override = 100 %).
- With the integrated oscilloscope, record the actual feed rate **(v act)** and if desired, the servo lag **(s diff)** as well.
- Increase the parameter **MP_maxPathJerk** until the overshoot disappears.

Note

The **MP_maxPathJerk** parameter is globally effective for all axes. Therefore, sequentially determine the jerk for each axis individually. In the parameter you then enter the jerk of the interpolating axis with the smallest determined jerk value. The specific jerk values determined for each axis are then entered in **MP_axJerk.**



Value in MP_maxPathJerk too high



Value in MP_maxPathJerk adjusted correctly

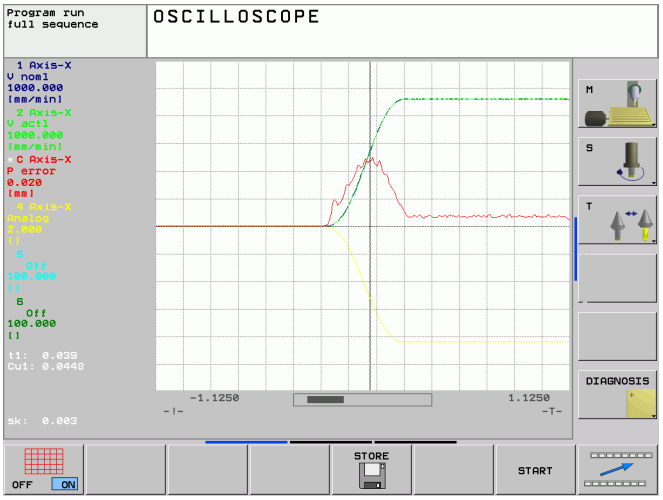
6. Determining acceleration feedforward control

Goal: The following error (servo lag) is to be set as small as possible during the acceleration phase.

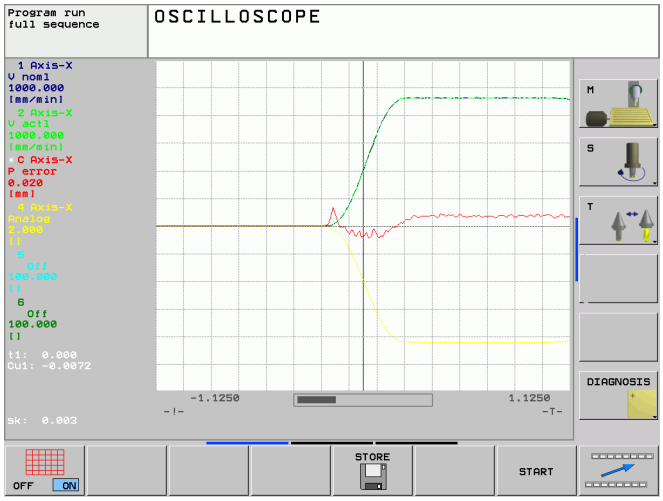
Machine parameters in the configuration editor	Temporary starting value	Meaning
Axes ParameterSets [Key for parameter set] CfgAxisAnalog **accForwardFactor**	0,005	k_V factor

This parameter is determined via step-by-step approximation to the ideal value:

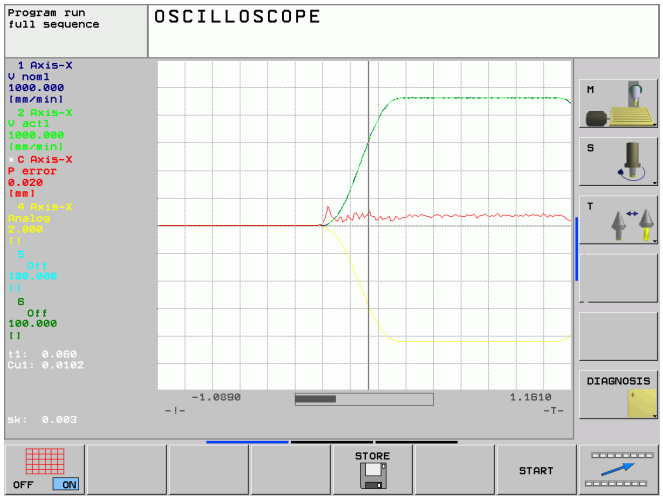
- Carefully increase the temporary starting value to determine the optimum setting for **MP_accForwardFactor.** Use the value 0.001 as a starting value in the next step.
- Determine the value just before an **undershoot** forms with the measured following error **(P err).**



MP_accForwardFactor not adjusted



MP_accForwardFactor adjusted too high



Correctly adjusted MP_accForwardFactor

Settings in the configuration editor:	
Axes ParameterSets [Key for parameter set] CfgPositionLimits **swLimitSwitchPos** **swLimitSwitchNeg**	

To define the software limit switches, proceed as follows:

- Traverse the reference points
- Enter **`NominalValue`** in the **Axis display** user parameter to display the nominal position with respect to the machine zero point.
- The position displays now show the distance to the machine zero point, without taking the tool lengths or zero point shifts into account.
- 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 with the algebraic signs.
- Enter the noted values in the machine parameters **MP_swLimitSwitchPos** or **MP_swLimitSwitchNeg**.
- Enter **`Default`** in the **Axis display** user parameter to return the position displays to the display of the tool tip position with respect to the workpiece zero point.

Note

You can enter different traverse ranges. You must define a separate parameter set per axis and traverse range. The individual traverse ranges are activated by switching the parameter sets (e.g. by PLC).

8. Activating monitoring functions

The monitoring functions of the control must be activated now.

Note

To ensure that the monitoring functions of the control become effective at the right moment, you must enter meaningful values.

- Activate the position monitoring (see "Position monitoring" on page 646)
- You define two limits in the machine parameters for the position monitoring: one for operation with following error, and one for operation with velocity feedforward control.
- Configure the movement monitoring (See "Movement monitoring" on page 650).
- Configure the standstill monitoring (See "Standstill monitoring" on page 652).

Note

Adjust the input values to the machine dynamics.

9. Hysteresis / static friction

For configuring the reversal-spike compensation, see compensation of reversal peaks for analog axes.

6.18.5 Commissioning the digital spindle

Current controller

The current controller is adjusted in the same manner as a digital axis, see "Current controller" on page 605.

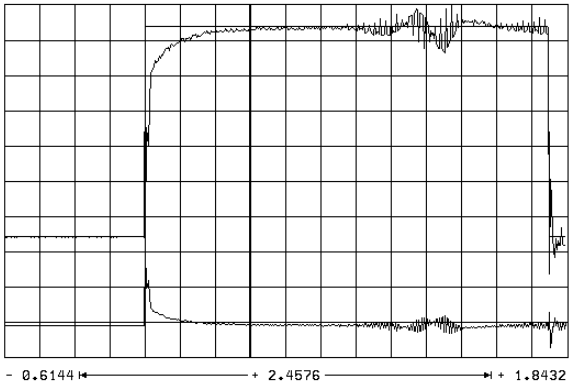
Speed controller

To activate the step function, enter the following machine parameters:

Settings in the configuration editor	
Axes	
ParameterSets	
[Key name of the spindle parameter set]	
CfgFeedLimits	
maxAcceleration:	[high value]
maxAccSpeedCtrl:	[high value]
CfgPositionFilter	
manualFilterOrder:	1

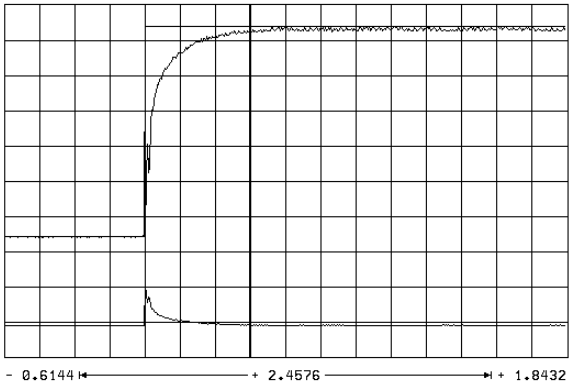
Use **MP_manualFilterOrder** = 1 to activate the lowest filter order for the Manual Operation mode. As a result, the nominal speed value is no longer "averaged".

- Start the step output by Spindle On.
- Activate a spindle speed from the highest gear range.
- Use the integrated oscilloscope to record the nominal feed rate **(v nom),** the actual feed rate **(v act),** and the actual current value **I (nom)**.
- Output a step with the Spindle On function (M03/M04).
- Choose the height of the step function for a very low speed so as not to overload the speed controller, i.e. so that **I (nom)** is not limited.
- Increase the P factor of the speed controller (**MP_vCtrlPropGain**) until the system oscillates or no change is visible. To change the machine parameter, press the END key in the oscilloscope and then the CONFIG EDIT soft key.



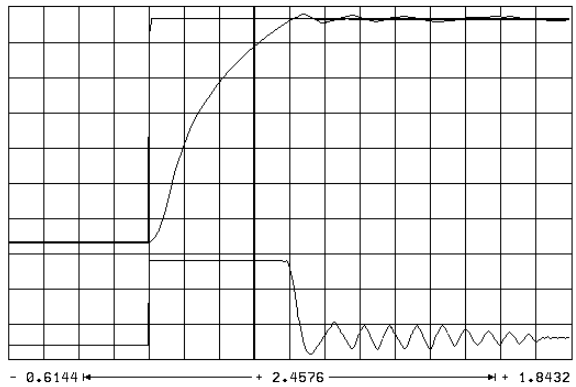
- Calculate the input value for **MP_vCtrlPropGain**:
  MP_vCtrlPropGain = **MP_vCtrlPropGain** · 0.6

▶ Increase the I factor (**MP_vCtrlIntGain**) until the nominal value is reached and there is no overshoot.

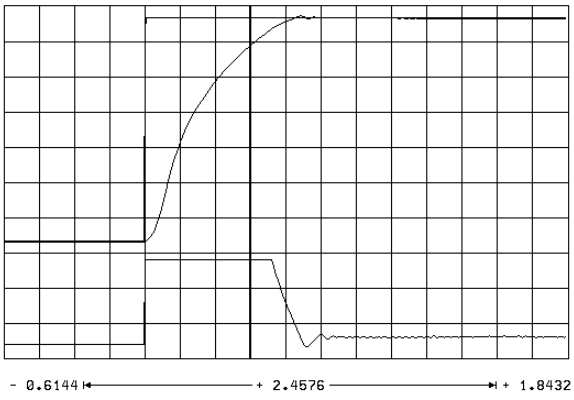


▶ Output the step with maximum shaft speed. **I (nom)** is within the limitation during acceleration. **I (nom)** must not oscillate after reaching the maximum speed. If **I (nom)** oscillates:

Reduce MP_vCtrlPropGain and MP_vCtrlIntGain evenly until the overshoots are minimized.



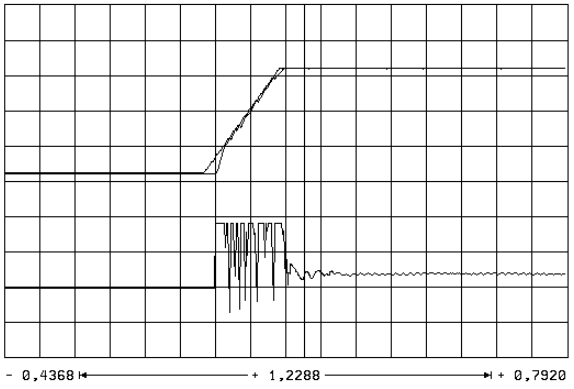
I (nom) oscillates



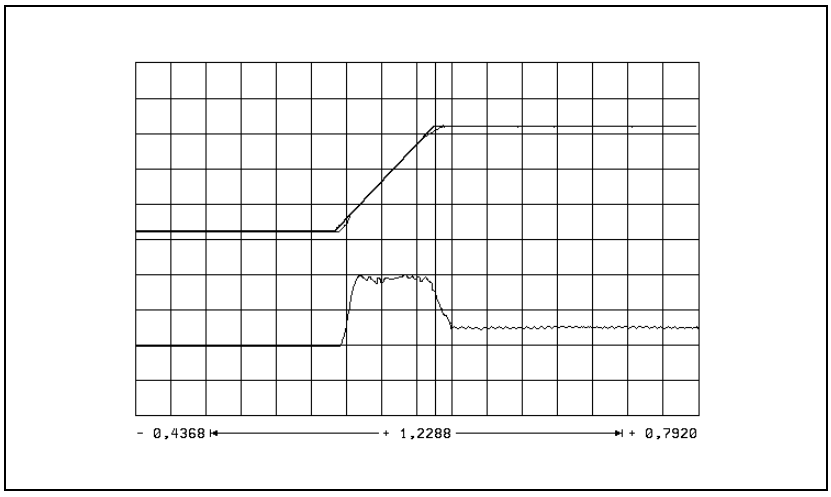
Only one overshoot

Optimize the acceleration for M3/M4/M5 with the parameters MP_maxAcceleration, MP_maxAccSpeedCtrl and MP_maxAccSpeedCtrl

- Optimize the acceleration individually for each gear range.
- **M3/M4**:
  With **MP_maxAccSpeedCtrl** choose a ramp gradient at which the motor almost reaches the electrical current limit.



- **M5:**
  Braking is usually performed at the current limit. Activate the optional parameter **MP_maxDecSpeedCtrl** if you want a braking ramp that is different from the acceleration ramp. If no value is entered the MANUALplus 620 uses the value entered under **MP_maxAccSpeedCtrl** or **MP_maxAcceleration**.



M19:
For tapping and oriented spindle stop, **I (nom)** must not be within the limit during acceleration:

- In **MP_maxAcceleration**, enter a lower value for these operations with closed-loop position control.
- Use the **MP_manualFilterOrder** parameter to influence the transient response of the spindle. Large values result in considerable signal rounding, small values in minor signal rounding.
  Adapt the nominal value curve to the actual value curve.

Check the direction of rotation:

You can check the direction of rotation of the spindle when M03 is output. If the spindle does not rotate in clockwise direction:

- Change **MP_signCorrNominalVal**

Position controller

The position control loop of the main spindle is closed only during the spindle orientation:

- Close the position control loop of the main spindle, See "Oriented spindle stop (spindle point stop)" on page 703.
- If the error message **Nominal speed value too high** appears, you must modify **MP_signCorrActualVal**.
- Optimize the k_v factor (**MP_kvFactor**) for each gear range. A TOOL CALL must be run to transfer the modified gear-specific parameters.

6.19 Integrated Oscilloscope

6.19.1 Fundamentals

The MANUALplus 620 features an integrated oscilloscope. This oscilloscope features 6 channels for recording analog signals and 16 channels for recording digital signals (see the following tables).

The recording of CC signals is limited to four signals.

Overview of signals

Analog signals	Meaning	CC signal
SAVED	The signal last recorded on this channel is "frozen."	–
a act	Actual axis acceleration value [m/s^2] or [°/s^2]. Calculated from position encoder.	–
a nom	Nominal axis acceleration value [m/s^2] or [°/s^2]	–
v act	Actual value of the axis feed rate [mm/min] or [°/min]. Calculated from position encoder.	–
v nom	Nominal value of the axis feed rate [mm/min] or [°/min]. Axis feed rate calculated from the difference from the nominal position values. The following error is not included.	–
Feed rate F	Contouring feed rate [mm/min] or [°/min]	–
Block no.	Block number of the NC program for triggering	–
s act	Actual position [mm] or [°]	–
s nom	Nominal position [mm] or [°]	–
s diff	Following error of the position controller [mm] or [°]	–
PosDiff	Difference between position and speed encoder [mm] or [°]	–
Position: A	Signal A of the position encoder	–
Position: B	Signal B of the position encoder	–
j act	Actual jerk value [m/s^3]. Calculated from position encoder.	–
j nom	Nominal jerk value [m/s^3]	–
v (N act)	Shaft speed actual value [mm/min]. Calculated from speed encoder	CC
v (N nom)	Nominal velocity value [mm/min]. Output quantity of the position controller	CC
I (N int)	Integral-action component of the nominal current value [A]; CC 422: peak value, CC 424: effective value	CC
I (N nom)	Nominal current value [A] that determines torque; CC 422: peak value, CC 424: effective value	CC
PLCPrePgm	The PLC operands (B, W, D, I, O, T, C) are recorded **before** the PLC program run. This means that the values of the operands are read at the beginning of the PLC program cycle. For types B, W and D the contents are recorded, and for the other types the logical state of the operands are recorded.	–

Analog signals	Meaning	CC signal
PLCPostPgm	The PLC operands (B, W, D, I, O, T, C) are recorded **after** the PLC program run. This means that the values of the operands are read at the end of the PLC program cycle. For types B, W and D the contents are recorded, and for the other types the logical state of the operands are recorded.	–
Analog	Analog axis/spindle: Analog voltage = nominal velocity value [mV]	–
IpoDbg	Diagnostic information for internal purposes	–
CCDbg	Diagnostic information for internal purposes	–
Contour deviat.	Circular interpolation test, contour deviation in mm	–
Off	No recording for this channel	–

Digital signals	Meaning
M	PLC marker
I	Input
O	Output
T	Timer
C	Counter
X	Reserved

Note

The PLC operands are addressed with numbers in the oscilloscope. You get the numerical addresses from the PLC diagnostic function **Watch List.**

Sampling rate

The resolution of the internal oscilloscope is at most 600 µs.

This means that even for CC signals, the signal is only recorded every 600 µs. However, since the CC 61xx and CC 424 operate with control-loop cycle times < 600 µs, undersampling results at higher frequencies. This can result in misinterpretation of the oscilloscope image.

For the CC 61xx and CC 424, the internal oscilloscope displays effective values, as opposed to the peak values of the CC 422.

6.19.2 Preparing a recording

Starting the oscilloscope

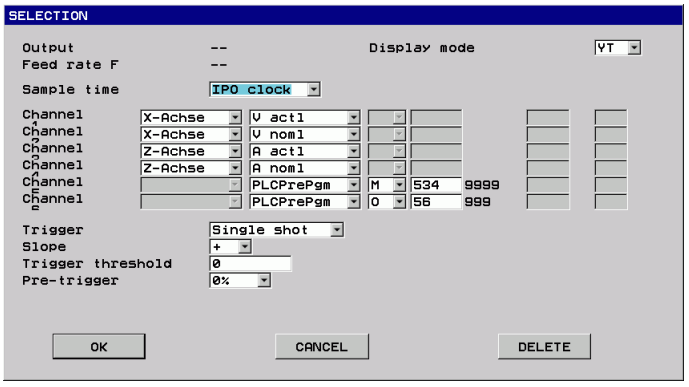
- Switch to the Organization mode of operation.
- Press the soft key.
- Enter the code number **688379** for the control to activate the setup menu.

Setup for analog signals

Call the "Selection" dialog box:

- Press the **SELECT** soft key.

In the dialog box, set:



- **Type of display:** Set the time interval for recording the signals.
  - YT: Chronological depiction of the channels
  - YX: Graph of two channels
  - Circle: Circular interpolation test, see page 814
- **Sampling time:** Set the time interval for recording the signals.
  - CC clock: Time interval = 0.6 ms
  - IPO clock: Time interval = IPO clock (from MP_System/CfgCycleTimes/ipoCycle)
  - PLC clock: Time interval = PLC clock (results from MP_System/CfgCycleTimes/plcCount * Ipo clock)

3000 grid points (events) are stored. The time grid determines the duration of recording.
Examples:

- 0.6 ms · 3000 = 1.8 s
- 3 ms · 3000 = 9 s
- 21 ms · 3000 = 63 s

Channel 1 to channel 6
Specify the signals to be recorded:

- Assign the channels of the analog signals to be recorded to the respective axes.
- Specify the operand type (B,W,D,I,O,T,C) and the address for the recording of PLC operands
- Use the **SAVED** setting to "freeze" the signal last recorded for this channel. This means that the recorded values remain available on the display. For example, you can use them to record a reference curve for use in future measurements.

Trigger conditions:
Specify the trigger conditions in the following input fields:

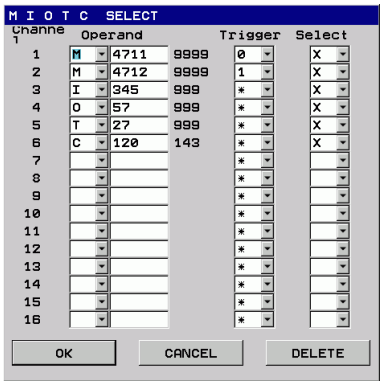
- **Trigger:** Set the trigger condition.
  - **Single shot**: After pressing the soft key, the next 3000 events are stored.
  - **Free run:** The recording is started and ended by soft key. If you press the STOP soft key, the last 3000 events (at most) are stored.
  - **Channels 1** to **6**: Recording begins when the trigger condition of the selected channel is fulfilled.
  - **Channel 1 + L** to **channel 6 + L:** Recording begins when the trigger condition of the channel selected here as well as the trigger conditions of the digital signals (trigger condition "logic") are fulfilled. The trigger conditions are AND-gated.
  - **Logic:** Recording begins when the trigger condition of the digital signals is fulfilled (trigger condition "logic").
- **Edge:** Set when triggering is to occur:
  - **+**: Trigger at rising edge
  - **–**: Trigger at falling edge
- **Trigger threshold:** Enter the trigger threshold (you will find the appropriate units in the signals table on 802).
- **Pre-Trigger:** Select a value from the selection box.
  Recording begins at a time preceding the trigger time point by the value entered here

The **Output** and **Feed rate F** fields are reserved.

MIOTC AUSWAHL

Call the "M, I, O, T, C selection" dialog box:

- Press the **SELECT MIOTC** soft key.



Set:

- **Operand**: Type and number of the PLC operand's symbolic name
  - M: Marker
  - I: Input
  - O: Output
  - T: Timer
  - C: Counter
  - X: Diagnostic information for internal purposes
  - s: Symbolic API operand
    After selection, the symbolic API operand can be selected over the SYMBOL LIST soft key.
- **Trigger:**
  - X: No trigger
  - 0: Trigger at 0-level
  - 1: Trigger at 1-level

The trigger is only taken into consideration if **Selection=X** is set.

- **Selection:**
  - X: Signal is displayed and considered as trigger
  - Empty field: Signal is not displayed and not considered as trigger

Note

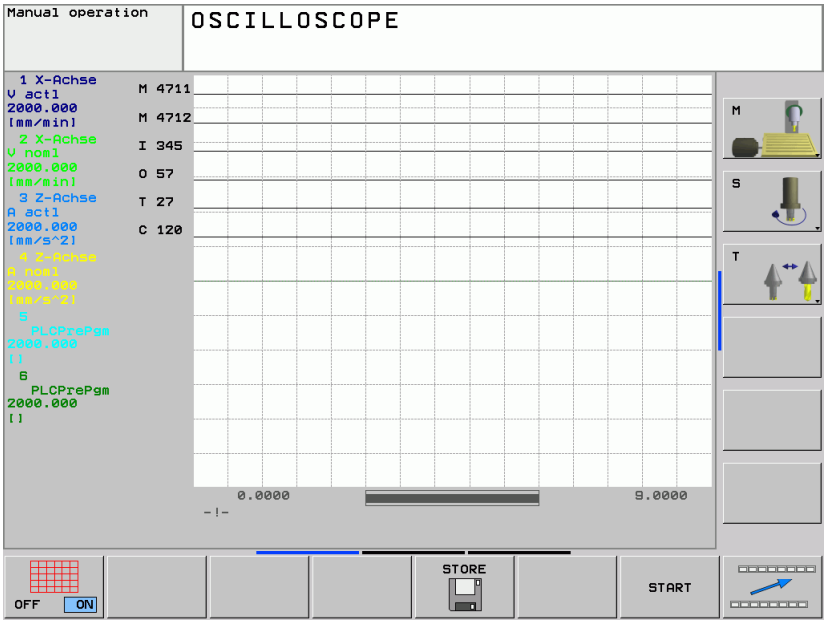
- You define the general trigger conditions ("Trigger" input field) and the pre-trigger in the setup for analog signals.
- The trigger condition "logic" is fulfilled when all triggers set in "M I O T C selection" are fulfilled (AND-gating).

6.19.3 Recording signals

OSCILLOS.

Change to the Recording operating mode:

- Press the **OSCI** soft key.



For every channel, the type and resolution of the analog signal are shown in the left status field. The operand type and address are listed for digital signals.

Starting and stopping the recording

START

To start recording:

- Press the **START** soft key.

STOP

To stop recording:

- Press the **STOP** soft key.

DISPLAY STOP

To stop the display:

- Press the **DISPLAY STOP** soft key.

Starting signal recording before the first PLC scan

To start the signal recording in the oscilloscope before the first PLC scan, proceed as follows:

- Start the control.
- Do **not** acknowledge the Power interrupted message with the CE key; instead, enter the code number 807667 to switch to the PLC programming mode of operation.
- Press the **COMPILE** soft key.
- Press the **SELECT + COMPILE PLC PGM** soft key and press the **SELECT** soft key to compile the **MAIN_PGM.SRC** PLC main program manually. The "PLC program could not be started" error message can be ignored.
- Select the desired data in the oscilloscope and press the **START** soft key to start recording.
- Acknowledge the power interruption with the CE key.

Trigger conditions: Trigger and pre-trigger conditions:

- Trigger=Single shot: 3000 events beginning from the start are recorded.
- Trigger=Free run: At most the last 3000 events before the STOP soft key is pressed are recorded.
- Trigger condition defined: The time when recording ends depends on the setting of the pre-trigger.
  - Pre-trigger=0%: 3000 events beginning from the fulfilled trigger condition are recorded.
  - Pre-trigger=25% (or 50%, or 75%): 75% (or 50% or 25%) of the 3000 events beginning from the fulfilled trigger condition are recorded.
  - Pre-trigger=100%: Recording is stopped. The last 3000 events before the fulfilled trigger condition are recorded.

Note

If the trigger condition is fulfilled **before** the corresponding number of events have been stored when the pre-trigger is set to 25 %, 50 %, 75 % or 100 %, then correspondingly fewer events are recorded.

During recording, the selected signals are continuously displayed. You can freeze the display of the signals with the DISPLAY STOP soft key. This does not affect the recording of the signals.

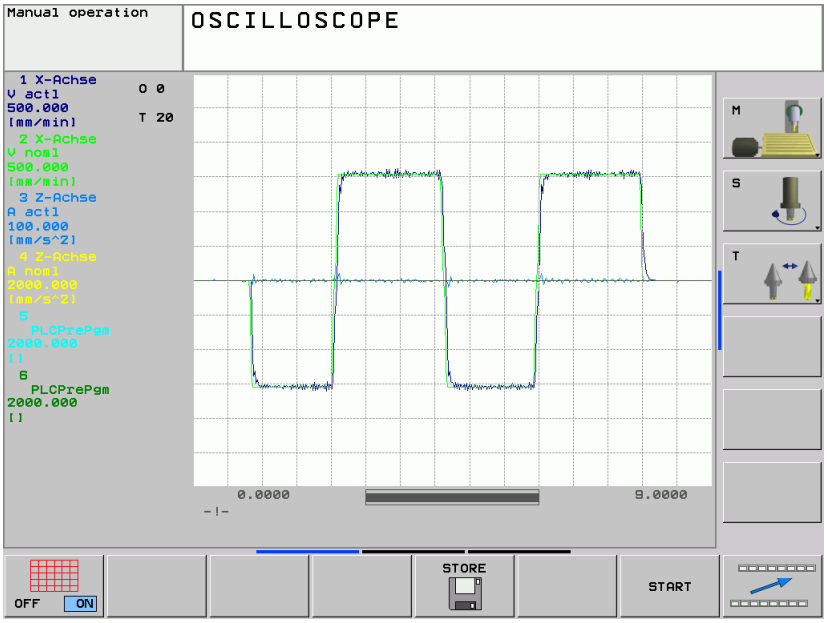
The recorded data remain stored until you start recording again or activate another graphic function.

A fulfilled trigger condition is indicated with a "T" in the status field at right below the display area.

Hide/show **gridlines**:

▶ Press the **GRID** soft key.

6.19.4 Analyzing the recording

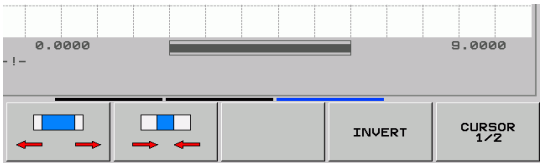


Recording completed

After recording has completed, the oscilloscope shows the memory contents. The information in the status field below the display area refers to the displayed events. It has the following meanings (see figure below):

- Left number: Time the "leftmost" event was recorded
- Right number: Time the "rightmost" event was recorded
- The bar symbolizes the displayed range relative to the memory content.

The fulfillment of the trigger condition is designated as "t0" (t=0). Events that were recorded before the trigger condition was fulfilled are given a negative time. If no trigger condition was defined, the beginning of the recording is designated as "t0."



Changing the display

The following soft keys influence the entire display range (all signals):

▶ Shift the display range to the left.

▶ Shift the display range to the right.

▶ Decrease horizontal resolution.

▶ Increase horizontal resolution.

Increment

The increment used when shifting the display or shifting the cursor is shown in the status field at bottom left, with the code "sk: ....." (see the "cursor information" illustration).

▶ The **Page Up** key increases the increment, and the **Page Down** key decreases it.

Analyzing an individual analog signal

▶ Use the **arrow up** and **arrow down** keys to select the channel to be analyzed.

The selected channel is indicated with an asterisk (*). At the same time, the cursor is activated and placed on the selected channel.

Shown in the status field (bottom left) are (see "cursor information" figure):

- Code "t1: .....": Cursor position in [s], referenced to t0
- Code "Cu1: .....": Signal size at the cursor position

Shifting the cursor:

▶ Shift the cursor with the **right arrow** and **left arrow** keys.

Influencing the signal display

Soft keys that influence the signal display:

▶ Shift the signal downward.

▶ Shift the signal upward.

▶ Decrease the vertical resolution.

▶ Increase the vertical resolution.

▶ Optimum vertical resolution. The signal is centered on the zero line and always remains in the display area.

▶ Undo vertical shifts.

▶ Invert the signal (multiply by –1).

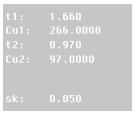
Second cursor

▶ Activate/deactivate second cursor.

The information for the second cursor is **relative to the first cursor.** It is shown in the status field (see "Cursor information" figure):

- Code "t2: .....": Cursor position in [s], referenced to the first cursor
- Code "Cu2: .....": Signal referenced to the first cursor position

Cursor information:



6.19.5 Saving and loading recordings

You can save the recorded signals and all settings in one file. The file must have the extension SCO (oscilloscope trace file).

You can load and evaluate saved SCO files in the oscilloscope. HEIDENHAIN also makes the PC program **TNCscopeNT** available for evaluating SCO files.

Saving and loading oscilloscope recordings:

Switch to the Saving and Loading mode:

- Press the soft key.

Save the oscilloscope trace file:

- Press the **SAVE** soft key.
- Enter the path in the "Save As" dialog box.
- Press the **OK** soft key or button.

Load the oscilloscope trace file:

- Press the **LOAD** soft key.
- Enter the path in the "Open" dialog box.
- Press the **OK** soft key or button.

6.19.6 Circular interpolation test with the integrated oscilloscope

A circular interpolation test can now be conducted with the integral oscilloscope.

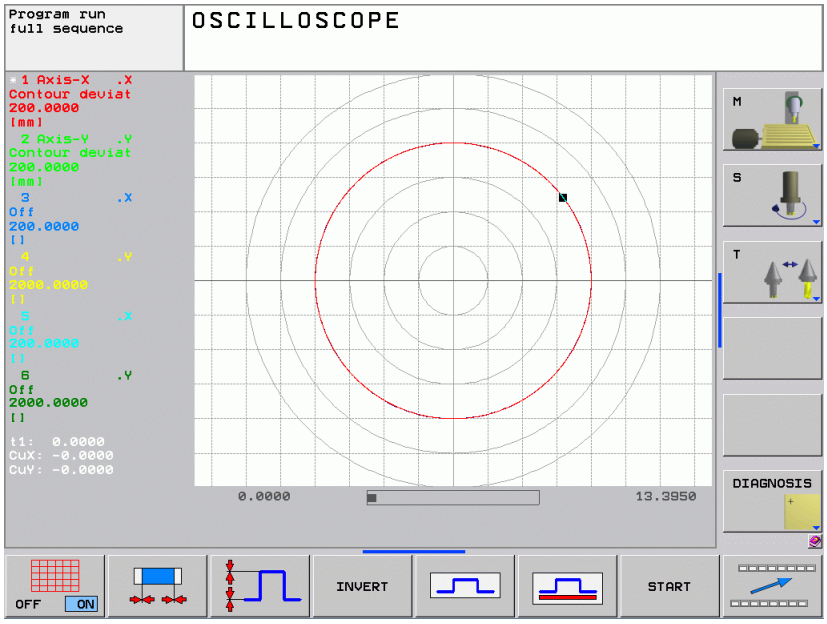
- Start the oscilloscope and press the SELECTION soft key
- Under **display mode**, select the setting **Circle**
- At the desired channels, select as the signal type the setting **Contour deviat.**
- Start the recording. Press the soft keys OSCI and then START.
- Start an NC program in which a circle is programmed. The circle center point must be at the origin of both axes.
- For lathe controls, the programs cir_xz_r10_f2000_f.nc and cir_xz_r10_f2000_r.nc prepared for the circular interpolation test are contained in the folder NCPS.
  The starting points are X0 and Z0.

Example of a circular interpolation test with the integrated oscilloscope:

Actual position:
X +30
Y +0

NC program (example of milling control):

```
0 BEGIN PGM Circular interpolation test MM
1 CC X+0 Y+0
2 CP IPA+360 DR+ F1000
3 M30
4 END PGM Circular interpolation test MM
```



Settings in the configuration editor	MP number
System	
DisplaySettings	
CfgOsciColor	
background	101401
channel1	101402
channel2	101403
channel3	101404
channel4	101405
channel5	101406
channel6	101407
logicTrace	101408
select	101409
grid	101410
cursorText	101411

▶ In the parameter object CfgOsciColor, define the colors for the oscilloscope.

MP_background

Background color
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
black
blue
light_gray
red
dark_gray
light_green
really_light_gray
really_dark_gray
light_violet
light_blue
light_red
medium_gray
yellow
white

Default: black
Access: LEVEL3
Reaction: NOTHING

The colors defined in MP_channels 1 to 6 are used for display of the status information of this channel and the path of the curve. As soon as a channel is selected, the color defined in **MP_select** is switched to.

MP_channel1

Color for channel 1
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: light_green
Access: LEVEL3
Reaction: NOTHING

MP_channel2

Color for channel 2
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: light_violet
Access: LEVEL3
Reaction: NOTHING

MP_channel3

Color for channel 3
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: light_blue
Access: LEVEL3
Reaction: NOTHING

MP_channel4

Color for channel 4
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: light_red
Access: LEVEL3
Reaction: NOTHING

MP_channel5

Color for channel 5
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: light_blue
Access: LEVEL3
Reaction: NOTHING

MP_channel6

Color for channel 6
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: light_red
Access: LEVEL3
Reaction: NOTHING

The color defined in **MP_logicTrace** is used for the display of the digital signals.

MP_logicTrace

Color for logic-trace channels
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: yellow
Access: LEVEL3
Reaction: NOTHING

MP_select

Color for selected channel
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: white
Access: LEVEL3
Reaction: NOTHING

MP_grid

Color for gridlines
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: light_gray
Access: LEVEL3
Reaction: NOTHING

MP_cursorText

Color for gridlines
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See MP_background
Default: dark_gray
Access: LEVEL3
Reaction: NOTHING

6.20 Diagnosis with the Online Monitor (OLM)

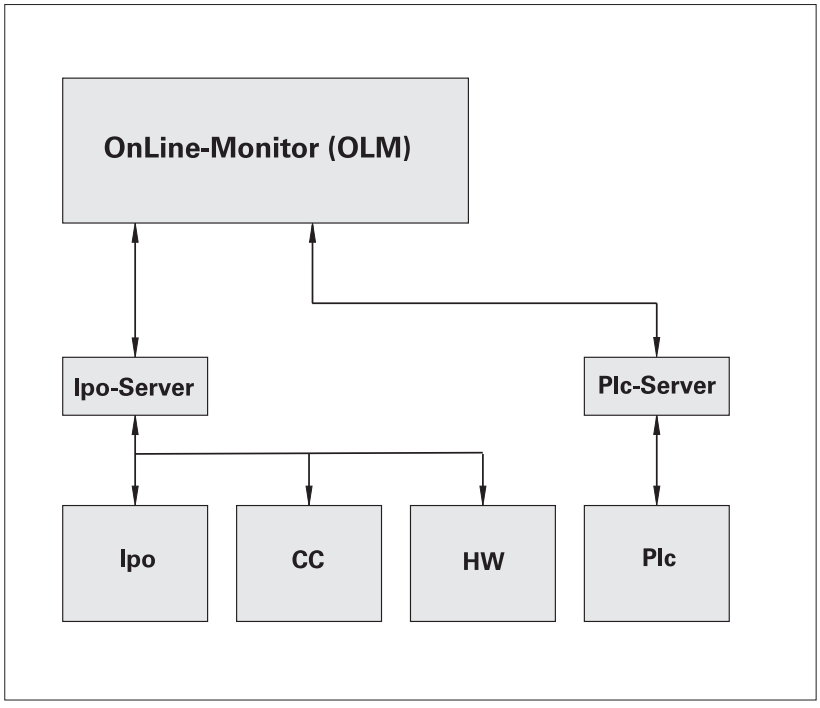
6.20.1 Introduction

The Online Monitor (OLM) assists you in commissioning and provides diagnostic functions to check control components:

- Display of IPO internal variables for axes and channels
- Display of CC internal variables (if a CC is present)
- Display of hardware signal states
- Various trace functions
- Activation of spindle commands
- Enabling IPO internal debug outputs

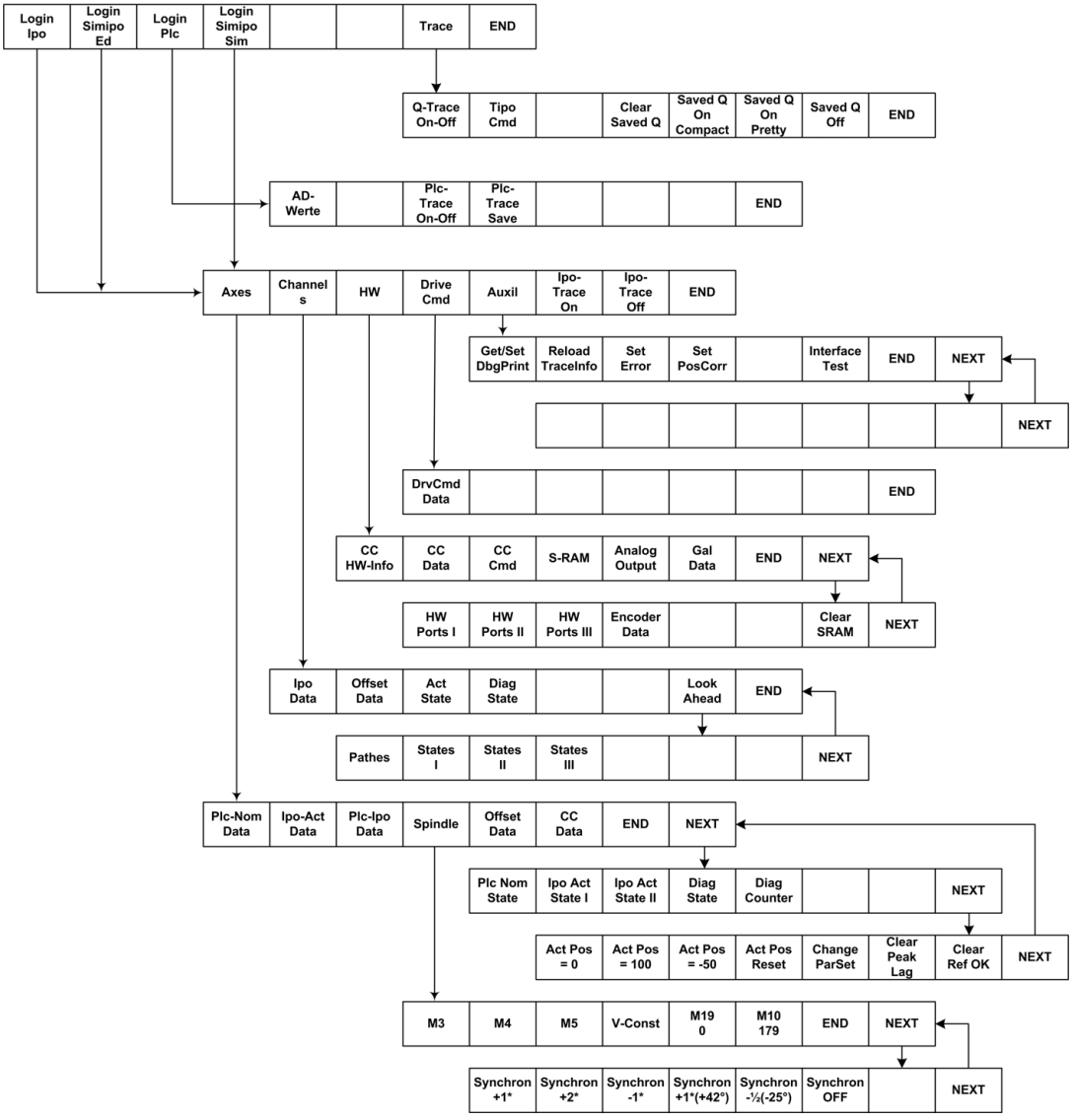
The OLM is included in the control software.

Software structure



6.20.2 Using the OLM

Overview of the menus



END and NEXT soft keys

The NEXT soft key indicates that additional soft keys are available for this menu level. The END soft key switches one level back.

- Press the **NEXT** soft key. The OLM switches to the next soft-key row of this level.

- Press the **END** soft key. The OLM switches one level back. If you press the END soft key on the main level, the OLM is exited.

Keyboard and display

The OLM is operated using the soft keys of the control keyboard. The control screen is used for display.

The OLM distinguishes the following software and function areas:

- IPO
- Simulation IPO (SimIPO)
- PLC
- Trace

The software area or function area is selected by soft key on the "main level."

For hardware reasons, only a subgroup of the IPO functions is available for the SimIpo. The available SimIpo functions work in the same way as the IPO functions.

Starting and exiting the OLM

To start the OLM:

- Press the soft key.
- Enter the code number 654321 for the control to activate the main menu.

To exit the OLM:

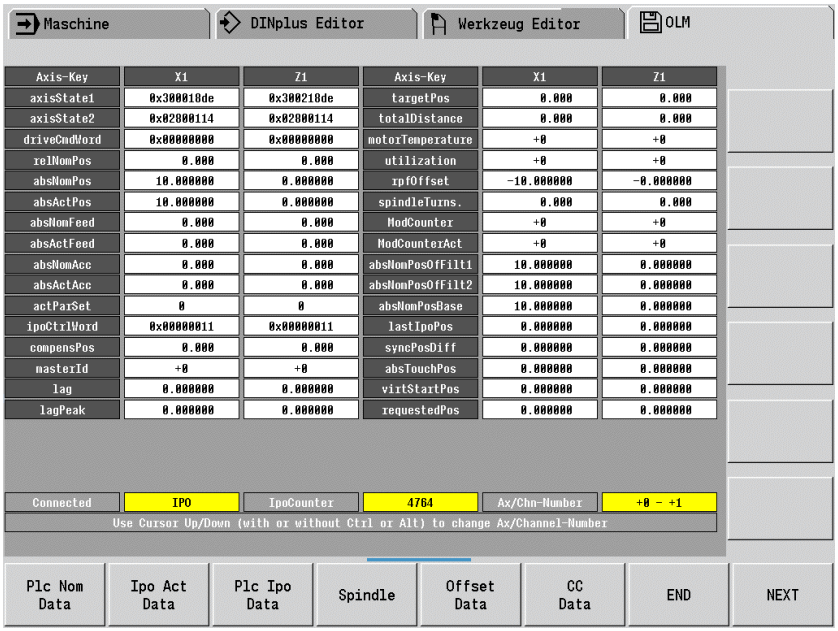
- Change to the main level of the OLM.

- Press the **END** soft key.

6.20.3 Screen layout

Display of variables

Example of screen layout when variables are displayed:



The OLM lists the variable designations, status designations, etc. in the **dark-highlighted boxes** of the "main window."

The **white-highlighted boxes** contain the variable values. The OLM displays the values of two axes or channels.

The axis designation or channel designation defined in the parameters is shown in the **column heading.**

- Parameter for axis designation:
  MP_Sytem/CfgAxes/axisList(axisId)
- Parameter for channel designation for IPO:
  MP_ChannelGroup/CfgChannelGroup/Machining/ChannelList
- Parameter for channel designation for SimIpo:
  MP_ChannelGroup/CfgChannelGroup/Simulation/ChannelList

The term **Index n** in the column heading means that no axis or no channel is defined for this index.

The following general data is displayed in the bottom display line:

- Connected: Indicates the software or function area to which the OLM is connected
  - Ipo
  - SimIpo
  - PLC6
  - none: No connection
- IpoCounter: Counts the IPO clock pulses
  Note: The contents of the IpoCounter are also stored for trace information and error messages.
- Ax/Chn-Number: Logical axis number or channel number (depends on the selected function)
  - Number at left: Left column
  - Number at right: Right column

If the number of variables displayed exceeds the capacity of the main window, use PageDown/PageUp to scroll to the next/previous group of variables. One group consists of eight displayed variables.

Units

The OLM displays data in the following units:

- Linear axes
  - For position, length, etc: in [mm]
  - For speed: in [mm/s]
  - For acceleration: in [mm/s^2]
- Rotary axes (spindles)
  - For position, etc: in [°]
  - For speed: in [°/s]
  - For acceleration: in [°/s^2]

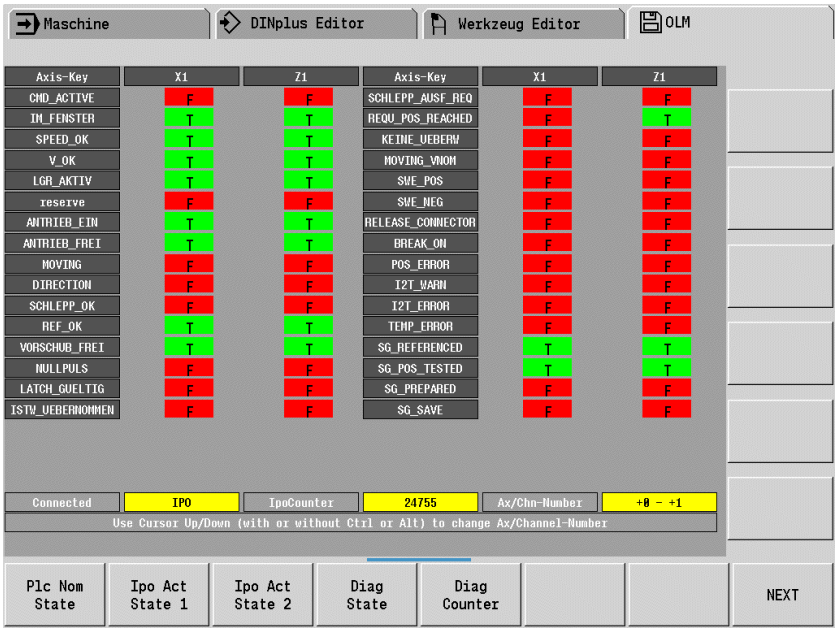
Selecting axes and channels

To select axes or channels, proceed as follows:

- Press **CTRL** + UP ARROW / DOWN ARROW to influence the **left column.**
  - CTRL + UP ARROW: Display the next axis/channel.
  - CTRL + DOWN ARROW: Display the previous axis/channel.
- Press **ALT** + UP ARROW / DOWN ARROW to influence the **right column.**
  - ALT + UP ARROW: Display the next axis/channel.
  - ALT + DOWN ARROW: Display the previous axis/channel.
- Press UP ARROW / DOWN ARROW (without CTRL or ALT) to influence **both columns.**
  - UP ARROW: Display the next axes/channels.
  - DOWN ARROW: Display the previous axes/channels.

Status display

Example of screen layout for status display:



In the main window, the status of the binary variables is displayed. The status is identified by the background color and the code letter.

- Green or "T": true
- Red or "F": false
- Yellow: The status is not defined yet

The information given about the display of variables also applies to the column headings and the bottom display line.

6.20.4 Group of NC axes

Nominal commands of the PLC (Plc Nom Data)

To select the **PLC Nom Data** function:

▶ Press the **`Login Ipo`** soft key.

▶ Press the **`Axes`** soft key.

▶ Press the **`PlcNomData`** soft key.

The function displays the nominal commands of the PLC for each axis in the following variables:

Variable	Display
PlcSollStatus	Axis status of the PLC as a bit line (listed in PLC-Nom_State)
MaxAchsVorschub	Maximum permissible axis feed rate in mm/s
AchsOverride	Override value for each axis (1 = 100%)
TempKorr	Temperature compensation in mm

IPO-internal variables (Ipo Act Data or Spindle)

The OLM displays the current IPO-internal variables of the selected axes (also spindle axes).

To select the **Ipo Act Data** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Axes** soft key.

▶ Press the **Ipo Act Data** soft key.

The display of the current IPO-internal variables is also activated when you select the **Spindle** function.

▶ Press the **Login Ipo** soft key.

▶ Press the **Axes** soft key.

▶ Press the **Spindle** soft key.

The function displays the following variables:

Variable	Display
axisState1	Bit line (listed in IpoActState 1)
axisState2	Bit line (listed in IpoActState 2)
driveCmdWord	Command for universal controller
relNomPos	Relative nominal position
absNomPos	Absolute nominal position
absActPos	Absolute actual position
absNomFeed	Absolute nominal velocity
absActFeed	Absolute actual velocity
absNomAcc	Absolute nominal acceleration
absActAcc	Absolute actual acceleration
actParSet	Current parameter set index
ipoCtrlWord	Internal control bit line of nominal commands in the IPO chain
compensPos	Compensation value
masterId	Active master during synchronism
lag	Current following error
lagPeak	Peak of current following error. The maximum peak of the following error is determined and displayed here. Use CLEARPEAKLAG to delete it.

Variable	Display
targetPos	Absolute target position of the axis
totalDistance	Total travel of the axis
motorTemperature	Motor temperature in degrees Celsius
utilization	Utilization of axis in %
rpfOffset	Coordinate system offset between switch-on position and reference point
spindleTurns.	Spindle revolutions – the value of the active spindle is calculated.
ModCounter	Only available for modulo axes. The counter is updated with each zero crossover of the current nominal position value of a modulo axis. All positions leaving the IPO (display, PLC, etc.) are calculated from the current position (0 to 359.9999 degrees) + moduloCounter * 360. The counter can be set, cleared, stopped and restarted from the geometry module. The modulo counter is saved cyclically in SRAM and restored during control startup.
ModCounterAct	Only available for modulo axes. The counter is updated with each zero crossover of the current actual position value of a modulo axis. All positions leaving the IPO (display, PLC, etc.) are calculated from the current position (0 to 359.9999 degrees) + moduloCounter * 360. The counter can be set, cleared, stopped and restarted from the geometry module.
absNomPosOfFilt1	Absolute nominal position before the nominal position value filter (see configuration data **System/CfgFilter**)
absNomPosOfFilt2	Absolute nominal position before the nominal position value filter (see configuration data **System/CfgFilter**)
absNomPosBase	Nominal position without compensation of virtual axes
lastIpoPos	Last interpolated nominal value of interpolator
syncPosDiff	Position difference during spindle synchronism (for slave spindle)
absTouchPos	Absolute actual position provided by measuring system
virtStartPos	Starting position of virtual axis (reference position). Basis for determining the relative virtual offset.
requestedPos	Position commanded by HMI for returning to the contour
absProgFeed	Currently programmed speed
PosWithChainTime	The value is "pre-calculated" in the spindle module and then processed in the internal path calculation. This value considers the runtime of the interpolator.
realAxisIdent	Number of the real axis to which a virtual axis is connected.

Variable	Display
relActPos	Relative actual position per IPO clock
xNom	Nominal position value in the CC interface (in interpolation steps).
absIstPosNonMod	Absolute actual position without any modulo calculation (for non-modulo axes: absIstPosNonModulo == absIstPos).
acceptableLag1	Permissible following error for warning.
acceptableLag2	Permissible following error for emergency stop error.
axisCommand	Internal axis commands (e.g. freeze override, from interpolator to spindle during thread cutting)
driveCmdWord	Command for universal controller
virtOffsetOn	Feedforward of virtual offset is active.
rpfActiv	Reference run for this axis is active.
measureLatch Active	Touch probe latch is active.
counterAbs PosCheck	Encoder monitoring is active.
checkPosStandstill	Standstill monitoring is active.
toolCorrDelta	Asynchronous tool compensation per IPO clock.
toolCorrNom	Nominal value for asynchronous tool compensation.
toolCorrAkt	Asynchronous tool compensation path executed until now.
wpCorrDelta	Asynchronous position compensation per IPO clock.
wpCorrNom	Nominal value for asynchronous position compensation.
wpCorrAkt	Asynchronous position compensation path executed until now.

Internal working data of PLC-IPO (Plc Ipo Data)

To select the **Plc Ipo Data** function:

▶ Press the **Login Ipo** soft key.

▶ Press the **Axes** soft key.

▶ Press the **Plc Ipo Data** soft key.

The function displays the following variables:

- **act-Cmd** (currently active command)
- **last-Cmd** (last assigned command)

Variable	Display
state	Possible states: ■ Finished: Command acknowledged. ■ Idle: Axis does not work and can be assigned a command. ■ MovingByHand: Manual direction key or PLC positioning is active. ■ StoppingByHand: Deceleration until standstill ■ WaitForPlcPosQuit: Waiting until the last nominal position value has been received by the axis (runtimes in the IPO chain) ■ RpfStart: Status during reference run ■ RpfFastToSwitchPreo: Status during reference run ■ RpfFastToSwitch: Status during reference run ■ RpfFastFromSwitchPrep: Status during reference run ■ RpfFastFromSwitch: Status during reference run ■ RpfSlowToSwitchPrep: Status during reference run ■ RpfSlowToSwitch: Status during reference run ■ RpfAktivatePulse: Status during reference run ■ RpfWaitForPulse: Status during reference run ■ RpfWaitForStop: Status during reference run ■ RpfLatchPos: Status during reference run ■ RpfFinish: Status during reference run ■ WaitForSpindlemoveQuit: Waiting for acknowledgment of spindle (e.g. speed reached or synchronism switched on, etc.) ■ Spindlemove: Spindle rotates at programmed speed.

Variable	Display
typeOfMove	Possible states: ■ NONE = 0: Axis is in IDLE state ■ AXKEY: Manual direction key ■ PLCPOS: PLC positioning ■ LIFTOFF: Lift off at Cycle stop ■ RESTORE_POS: Return to contour (block scan) ■ SYNCHRON: Synchronism (only for spindles) ■ PLCMEASURE: Measuring with PLC axes ■ SG_POS: Positioning with safety-related (SG) package ■ REVOL_FEED: Feed per revolution ■ SPINDLE: Command to spindle (M3, M4, M19 etc.)
noLimitSw	During PLC positioning, software limit-switch monitoring can be switched off (T = switched off).
error	The following errors can occur: ■ AxisAlreadyActive = 1: Axis is already working and cannot be given a new command. ■ PlcposAlreadyActive: Axis is already working and cannot be given a new command. ■ KeyposAlreadyActive: Axis is already working and cannot be given a new command. ■ OnlyOneAxWithKinem: When the kinematics model is switched on, the PLC-IPO can assign commands to only one axis at a time. ■ SweAlreadyActive: Axis is located at software limit switch. ■ MovementCanceled: Movement was canceled. ■ TsSwitched: Touch probe has triggered ■ ErrorPending: Touch probe error must be cleared first (in the error window). ■ NoToolAx: No tool axis. ■ MinFeed: Too small a feed rate is programmed. ■ M19_ACTIVE: M19 is active; axis cannot be moved. ■ NoChannel: Feed-per-revolution command was sent to an axis without NC channel. ■ M19WithoutRef: Not used. ■ ReconfigActive: Do not move any axis while changing parameters. ■ AlreadyMaster: This axis is a master and cannot be a slave at the same time (synchronism). ■ AlreadySlave: Only the SYCHRON_OFF command is allowed for a slave spindle. ■ NoSlave: The SYNCHRON_OFF command was sent to a non-slave spindle.
Vb-Prog	Programmed feed rate (mm/s)
Vb-Act	Active feed rate
Source	Type of handwheel (serial, e.g. HR410, etc., or encoder)
Dist/Revol	Traverse per handwheel revolution

Variable	Display
DistMax	Maximum traverse range (+/–) of handwheel
Factor	Internal conversion factor (dist./rev. / incr./rev.)
Impulse	Handwheel pulses at current IPO clock
ImpulseLast	Handwheel pulses at previous IPO clock
Position	Handwheel position
PosRaster	Handwheel position (for handwheel with detent).
InputsToPLC	Handwheel keys are sent to the PLC as bit line
OutputsFrom PLC	e.g. LEDs on the HR 410.

Data from the IpoOffset module (Offset Data)

The data of the interpolator and the PLC-IPO are collected in the module IpoOffset.

To select the **Offset Data** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Axes** soft key.

▶ Press the **Offset Data** soft key.

The function displays the following variables:

Variable	Display
offsetIpoSteuer	Internal control bit line
IpoSteuer	Internal control bit line
IpoSteuerErlaubt	Internal control bit line
handValid	Nominal values from the PLC-IPO are available.
offsetPosition	Nominal position (relative) from PLC interpolator and/or kinematics.
kinematikOffset	Incremental offset of kinematics.
autoValid	Nominal values from the interpolator are available.
sollPosition	(Absolute) nominal position from the interpolator.
sollPosBase	Absolute position (sum of SollPosition and OffsetPosition)
lastPosition	Absolute position (sum of sollPosBase and virtual offset)
lastPosWithoutG	Position of the axis without grinding offset. The axis has reached this position by executing the movement in the standard interpolator.
grindingValid	Validity of the value in **grindingOffset.** True: The content of **grindingOffset** is added to the nominal axis value.
grindingOffset	Offset value generated by the grinding interpolator.
channelNr	Current channel number
The following variables coordinate the movements of the real and virtual axes. The nominal position value of the virtual axes is added to the nominal position value of the real axis (feedforward of nominal value). This applies only to the relative movements of the virtual axis.	
virtOffsValid	Validity of the value in **virtOffset.** True: **virtOffset** is effective.
virtOffset	Value of the relative movement of the virtual axis. (Only important for real axes.)

Variable	Display
virtOffsActive = true	The feedforward of the nominal position value is configured and effective. (Only important for virtual axes.)
virtStartPos	Starting position of the virtual axis before feedforward of nominal position value is activated. (Only important for virtual axes.)
realAxisNr	Number of the real axis to which the virtual axis was connected. (Only important for virtual axes.)
virtOffsetOn	Feedforward of position value of virtual axis is active. (Only important for real axes.)
channelNr	Assignment of the axis to the channel.

Example of coordinated movements of real and virtual axes:

- Absolute position of virtual axis = 5.0
- The virtual axis is moved to position 8.0.
- Resulting virtual offset (virtOffset) at the end of movement = 3.0.

Data of the active CC (CC Data)

To select the **CC Data** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Axes** soft key.

▶ Press the **CC Data** soft key.

The function displays the following variables:

Variable	Display
ccIndex	Current position of the axis (on CC0 or CC1).
axIndex	Index of this axis on the current CC.
N_SOLL_IN	Nominal velocity transferred to the CC (internal units in the CC).
V_VORST	Nominal feedforward velocity transferred to the CC (internal units in the CC).
N_SOLL_IN_DELTA	Nominal acceleration transferred to the CC (internal units in the CC).
N_SOLL_IN_CHK	Handshake bit
DriveCmdWord	Command word
boDREG_FREI	Controller enable
S_IST	Actual position in the CC interface.
S_IST_CHK	Handshake bit
N_IST	Actual velocity
S_REF	Reference position
S_REF_CHK	Handshake bit
S_3D	Measurement result (position)
S_3D_CHK	Handshake bit
S_IST_ROT_ENC	Position of the motor encoder
S_IST_CHK_ROT_ENC	Handshake bit
Utilization	Utilization of drive
MotTemp	Motor temperature
Dummy01	

Nominal status of the axes (Plc Nom State)

The nominal status of the axes is requested by the PLC.

To select the **PLC Nom State** function:

▶ Press the **Login Ipo** soft key.

▶ Press the **Axes** soft key.

▶ Press the **Plc Nom state** soft key.

The function displays the status of the following binary variables (the descriptions refer to the status "true"):

Variable	Display
RefSwitch	Trip dog
ActualToNom	Transfer the current values as nominal values.
ClampRequest	Request for clamping this axis.
PosCtrlRequest	Request for position feedback control for this axis
NoMonitoring	No monitoring of following error or standstill
NoFeedhold	Feed rate stop is not set
DriveOnReq	Request to switch the drive on
ActToNomStrb	Not used
CurrentOff	Switch off the current for wye/delta connection
SpiChangeDir	Change the direction of spindle rotation
ResetRefOk	Reset the Ref OK flag when changing from 0 to 1
NoPosDiffCheck	Suppress the position difference check

Actual status 1 of the axes (Ipo Act State 1)

To select the **Ipo Act State 1** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Axes** soft key.

▶ Press the **Ipo Act State 1** soft key.

The function displays the status of the following binary variables (the descriptions refer to the status "true"):

Variable	Display
CMD_ACTIVE	Command is active for this axis
IM_FENSTER	Following error is within the positioning tolerance
SPEED_OK	Feed rate is OK
V_OK	No acceleration active
LGR_AKTIV	Position feedback control is active
ANTRIEB_EIN	Drive is on
ANTRIEB_FREI	Drive ready for operation
MOVING	Axis is in motion (feed rate > 0)
DIRECTION	Direction of motion (true: negative direction or standstill)
SCHLEPP_OK	Not used
REF_OK	Axis was homed
VORSCHUB_FREI	Feed rate has been enabled (no feed stop)
NULLPULS	Reference pulse in one IPO cycle
LATCH_GUELTIG	Probe value is valid
ISTW_UEBERNOMMEN	The current value was transferred instead of the nominal value.
SCHLEPP_AUSF_REQ	If position feedback control is activated, the "old" position is approached (no compensation of following error)
REQU_POS_REACHED	Requested position is reached
KEINE_UEBERW	Request from PLC: No servo lag and standstill monitoring
MOVING_VNOM	Axis is in motion (nominal feed rate > 0)
SWE_POS	Positive software limit switch is reached
SWE_NEG	Negative software limit switch is reached
RELEASE_CONNECTOR	Status of axis-specific enabling (X150/X151)

Variable	Display
BREAK_ON	Request to PLC for activating the brake within 100 ms
POS_ERROR	Positioning error
I2T_WARN	Warning during I^2t monitoring
I2T_ERROR	Error during I^2t monitoring
TEMP_ERROR	Error during temperature monitoring
SG_REFERENCED	Axis was homed (SG: safety-oriented control)
SG_POS_TESTED	Axis was "tested" by the user (SG: safety-oriented control)
SG_PREPARED	Axis was homed and "tested" by the user (SG: safety-related control)
SG_SAVE	Safe axis (SG: safety-oriented control)

Actual status 2 of the axes (Ipo Act State 2)

To select the **Ipo Act State 2** function:

- Press the **Login Ipo** (or **Login SimIpo**) soft key.

- Press the **Axes** soft key.

- Press the **Ipo Act State 2** soft key.

The function displays the status of the following binary variables (the descriptions refer to the status "true"):

Variable	Display
NO_CONTROL	Axis is not controlled (internal IPO status)
SPEED_CONTROL	Speed control is active (internal IPO status)
POS_CONTROL	Position control is active (internal IPO status)
INTERNAL_ERROR	Error has occurred (internal IPO status)
CHANNEL_AXIS	Axis belongs to a channel
CHANNEL_ SPINDLE	Axis is a spindle of a channel
PLC_AXIS	Axis received a command from the PLC
PLC_SPINDLE	Axis is used as spindle
IS_ACTIVE	Axis is physically available and can be given a command
IS_MANUAL	Axis is a noncontrolled counter axis
IS_VIRTUAL	Virtual axis whose nominal position values can be added to those of other axes. (Axis does not have its own servo drive.)
IS_DISPLAY	Axis is only displayed. (Axis does not have its own servo drive.)
NORMAL_FEED	Feed rate in "travel/minute."
REVOL_FEED_ MANUAL	Feed rate in "travel per revolution" in the manual control mode.
REVOL_FEED_ PROGR	Feed rate in "travel per revolution" in the automatic mode.
VCONST_FEED	Only for spindles – Feed rate at constant cutting speed
NC_CMD_ACTIVE	Command from the NC is active
PLC_CMD_ACTIVE	Command from the PLC is active
HR_ACTIVE	Handwheel is active
NC_STOP_ACTIVE	NC stop is active in the channel
SP_SPEED_ REACHED	Only for spindles – last spindle speed reached
SP_MASTER	Axis is master spindle (for spindle synchronism)

Variable	Display
SP_SLAVE	Axis is slave spindle (for spindle synchronism)
SP_SYNC_ REACHED	Last spindle speed reached (for spindle synchronism)
LGR_REQUEST	Only for spindles – IPO-internal request for position feedback control
TAKE_CYC_DATA	Use the nominal values from the cyclic message
IS_NOT_ACTIVE	Axis was configured but is not available physically
IS_ENDAT	Axis with EnDat encoder

Status of the axes (Diag State)

To select the **Diag State** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Axes** soft key.

▶ Press the **Diag State** soft key.

The function displays the status of the following binary variables (the descriptions refer to the status "true"):

Variable	Display
OwnFeedhold	No feed-rate enable from the PLC for this axis
OtherFeedhold	No feed-rate enable from the PLC for an axis of the kinematics
NewSlope	Internal use
NoAntriebFrei	No drive enable by the PLC
OwnPosCtrl	Position feedback control not active
OtherPosCtrl	Position feedback control not active in an axis of the kinematics
Override=0	Override at 0%
OffsetCtrl	Internal use
TakeCycData	Internal use
PlcMaxFeed=0	MaxFeed from the PLC is 0
TsMaxFeed=0	MaxFeed from the touch-probe table and/or from parameter "maxTouchFeed" is 0
WorkingOutLag	Following error is corrected

Counters for axis commands (Diag Counter)

To select the **Diag Counter** function:

- Press the **Login Ipo** (or **Login SimIpo**) soft key.

- Press the **Axes** soft key.

- Press the **Diag Counter** soft key.

All counters are reset to 0 when the control is started. The function displays the following counters:

Variable	Display
cntrActToNom	Number of nominal/actual position captures
cntrFeedHold	Number of changes of the feed-rate enable
cntrM3	Number of M3 commands
cntrM4	Number of M4 commands
cntrM5	Number of M5 commands
cntrHwheel	Number of handwheel commands

Group of axis commands

The OLM transfers the axis commands directly to the axis. The current IPO internal variables are displayed in the display boxes.

The commands apply to the **axis selected in the left column.**

To select the **axis commands:**

- Press the **Login Ipo** soft key.

- Press the **Axes** soft key.

- Press the **Plc Ipo Data** soft key.

- Press the **ActPos=0** soft key or a soft key for another command.

Axis commands	
	Set axis to position 0.
	Set axis to position 1000.
	Set axis to position –50.
	Restore original axis data.

Switching the parameter set of an axis (Change ParSet)

The active parameter set of an axis can be switched for test purposes.

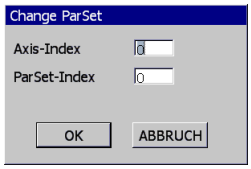
To select the **Change ParSet** function:

- Press the **Login Ipo** (or **Login SimIpo**) soft key.

- Press the **Axes** soft key.

- Press the **Change ParSet** soft key for the OLM to open the "Change ParSet" dialog box.
- Enter the required data in the dialog box (see below).
- Confirm with **OK**—the OLM switches to the defined parameter set.

"Change ParSet" dialog box:



Dialog box entries:

- **Axis-Index:** Logical axis number
- **ParSet-Index:** Index of parameter set

Deleting the following error (Clear PeakLag)

The IPO saves the greatest following error (PeakLag) that occurred. The Clear PeakLag function deletes this variable for all axes.

To select the **Clear PeakLag** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Axes** soft key.

▶ Press the **Clear PeakLag** soft key.

Deleting the reference point (Clear RefOk)

Deleting the reference point is necessary for being able to assign a new reference-run command. The function is effective for all axes.

To select the **Clear RefOK** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Axes** soft key.

▶ Press the **Clear RefOK** soft key.

6.20.5 Group of spindle commands

The OLM transfers the spindle commands directly to the spindle. The current IPO internal variables are displayed in the display boxes.

- The standard spindle commands apply to the **spindle selected in the left column.**
- If you want to use commands for **spindle synchronism,** remember the following assignment:
  - Spindle in left column: Master spindle
  - Spindle in right column: Slave spindle

To select the **spindle commands:**

▶ Press the **`Login Ipo`** soft key.

▶ Press the **`Axes`** soft key.

▶ Press the **`Plc Ipo Data`** soft key.

▶ Press the **`Spindle`** soft key.

▶ Press the **`M3`** soft key or a soft key for another spindle command.

Specify the direction of rotation (M3 or M4) in the commands for spindle synchronism.

Spindle commands	
	Spindle rotates (at 123 rpm) in M3 direction
	Spindle rotates (at 234 rpm) in M4 direction
	Spindle is stopped
	The spindle rotates at constant cutting speed (2000 m/sec in M3 direction)
	Spindle point stop at 0°
	Spindle point stop at 179°

Spindle synchronism commands	
Synchron +1*	Spindle synchronism ■ Same direction of rotation ■ Speed ratio master/slave: 1/1
Synchron +2*	Spindle synchronism ■ Same direction of rotation ■ Speed ratio master/slave: 1/2
Synchron -1*	Spindle synchronism ■ Reversed direction of rotation ■ Speed ratio master/slave: 1/1
Synchron +1* (+42°)	Spindle synchronism ■ Same direction of rotation ■ Speed ratio master/slave: 1/1 ■ Angle offset: 42°
Synchron -1/2 (-25°)	Spindle synchronism ■ Reversed direction of rotation ■ Speed ratio master/slave: 2/1 ■ Angle offset: –25°
Synchron OFF	Switch off spindle synchronism

Data of the interpolator module (Ipo Data)

To select the **Ipo Data** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Channels** soft key.

▶ Press the **Ipo Data** soft key.

The function displays the following variables:

Variable	Display
State	Possible IPO states: ■ Idle: IPO is idle (start-up) ■ RdNextMsg: IPO is waiting for job—reads from its input queue ■ Running: IPO is working (traverses the axes) ■ Waiting: Waiting during synchronism between IPO, PLC and channel object ■ WaitingForLr: Waiting to ensure that the nominal value last generated was received by axes (IPO chain) ■ WaitingForCancel: Waiting for GmCanceled from the input queue ■ WaitingForAxes: Waiting until all axes are in the control window ■ WaitingForLiftOff: Waiting for lift off movement after NcStop ■ WaitingForLA: Waiting for Look-Ahead ■ WaitingForSpindle: Waiting for spindle command to be executed (internal M19 during drilling) ■ ShapeReset: ■ StartThreadCutting: Thread cutting ■ WaitingForPLCPos: Waiting for PLC positioning ■ WaitingForTime: Waiting times for tapping with BCD spindles ■ WaitingForPlcQuit: Waiting for acknowledgment from the PLC during tapping with BCD-encoded spindles (M3, M4, M5)
kanalStatus	Possible channel states, represented as bit codes (see "Current status of the channel (Act State)" on page 850).
chainState	Status of the IPO chain: ■ IPO chain is "full." ■ IPO chain is "empty." ■ IPO chain is "almost empty" (waiting for the last acknowledgment message)

Variable	Display
chainCount	Number of acknowledgment messages in the IPO chain
satzCount	Number of blocks in this NC program.
blockId	ID of the current block
blockNumber	Number of the current block (from NC program)
fileName	Current NC program
syncActState	■ Running: Synchronization not active ■ Wait SS: Waiting in Single Block mode ■ Wait Sync: Waiting during the synchronization of IPO and PLC
syncWaitFor	Is only relevant if syncActState is **not** SyncRunning. ■ Channel object ■ PLC-Sync ■ Step-Contrl
syncWaitId	Number of the current synchronization
CH-syncIdWait	Waiting ID for M97, G62, G63 (channel synchronization), not relevant for MANUALplus 620
CH-syncId	Starting ID for M97, G62, G63 (channel synchronization), not relevant for MANUALplus 620
eomStopId	ID of the End of Move (eom) from the geometry chain
laStopId	ID of the End of Move (EOM) from the real-time LookAhead
startPath	Absolute starting position of current traverse on the path.
endPath	Absolute end position of current traverse on the path.
pathLength	Current path length
S(t)	(Absolute) position on the path
P(s)[0]	Position of the first axis to be interpolated
P(s)[1]	Position of the second axis to be interpolated
P(s)[2]	Position of the third axis to be interpolated
P(s)[3]	Position of the fourth axis to be interpolated
P(s)[4]	Position of the fifth axis to be interpolated
P(s)[5]	Position of the sixth axis to be interpolated
RevolFeedProg	Programmed feed rate per revolution in automatic mode
RevolFeedMan	Programmed feed rate per revolution in manual mode
ProgFeed	Programmed contour speed

Variable	Display
Fmax	(If true:) Rapid traverse was programmed
toolCorrId	ID of the current asynchronous tool compensation
wpCorrId	ID of the current asynchronous workpiece compensation

Internal data of the offset interface (Offset Data)

To select the **Offset Data** function:

- Press the **`Login Ipo`** (or **`Login SimIpo`**) soft key.

- Press the **`Channels`** soft key.

- Press the **`Offset Data`** soft key.

The function displays the following variables from the IpoOffset:

Variable	Display
kindOfKinComp	Type of current kinematics
kindOfKinCompSave	Type of current kinematics
useFrozenAxVal	The "frozen position" is used for rotary axes
v_bahn	Current contour speed
mySpindleNr	Spindle number belonging to this channel
achsAnz	Number of axes of this channel to be interpolated
logAchsNr[0] . . . logAchsNr[8]	First logical axis number of the channel . . . Ninth logical axis number of the channel

Current status of the channel (Act State)

To select the **Act State** function:

- Press the **Login Ipo** (or **Login SimIpo**) soft key.

- Press the **Channels** soft key.

- Press the **Act State** soft key.

The function displays the following binary variables:

Variable	Display
rapidFeed	Rapid traverse is active for this channel
ncStopTaster	PLC request for NcStop at triggering of touch probe
override100	The override is frozen at 100%
singleStep	Single Block is active
ncStart	NC start is active
internStart	Internal NC start is active
systemCycle	A system cycle is active
ncStop	NC stop is active
programStop	Program stop (M00/M01) is active
cancel	Cancellation is active
threadCycle	Thread cycle is active
tProbeCycle	Touch probe cycle is active
threadRevFeed	Feed rate per revolution during thread cutting
tasterMonitorGeo	NC request for monitoring of touch probe
tasterMonitorPlc	PLC request for monitoring of touch probe
measure	The touch probe has triggered.
revolFProgRun	Feed rate per revolution in automatic mode is programmed for this channel.
revolFManualMode	Feed rate per revolution in manual mode is programmed for this channel.

Diagnosis of the channel status (Diag State)

To select the **Diag State** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Channels** soft key.

▶ Press the **Diag State** soft key.

The function displays the following binary variables:

Variable	Display
WaitingForAxes	Waiting until all axes are in the control window after the program is started
noSFromT	No new nominal values from the Look-Ahead
SameSFromT	Cyclically identical nominal values from the Look-Ahead

Look-Ahead information

To select the **Look-Ahead information:**

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **Channels** soft key.

▶ Press the **Look Ahead** soft key.

▶ Press the **Paths** or **States I** or **States II** or **States III** soft key.

The functions display the states and variables of the look-ahead function. Analyzing this data is reserved for look-ahead specialists.

Hardware information of the CC (CC HW Info)

To select the **CC HW Info** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **HW** soft key.

▶ Press the **CC HW Info** soft key.

Note

The hardware information displayed for the CC is only valid if a CC with a parallel interface is used (no HSCI).

On the **left side of the screen** the OLM shows the variables as they are provided by the CC (first CC in the left column, and second CC in the right column):

Variable	Display
CC-Type	
ActiveAxes	Displays the available axes in bit-encoded representation.
CtrlType	Provides information on hardware and software versions, etc.
TestVersion	Information on the DSP software. Indicates either the letter of a test version or the number of a DSP software service pack. The MC must take this additional information into account.
LinkDate	Contains the software generation in 32 bits (Unix format)
SoftIdent	Contains the software ID and the version code.
ReadyInc	After processing the remaining identification cells, the variable is incremented by the controller program or boot program to allow an "activity check."

The data provided by the CC are explained in the **right half of the screen** (first CC in the left column, and second CC in the right column):

Variable	Display
BootHost	Boot software from the host
BootEprom	Boot software from the EEPROM

Variable	Display
Model	Controller model: ■ TMS320C31 ■ TMS320C32 ■ TMS320C32 – spindle board ■ Universal controller
HW-Info	Information on the hardware version
PS-Modul	Module for safety-related controls (SG: safety-oriented controls)
Version	Test version or release version (delivery version)
TestVers.	Is the software a test version?
LinkDate	Contains the software generation in 32 bits (Unix format)
SW-Info	Information on the software version

Transfer of commands between IPO and CC (CC Data)

To select the **CC Data** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **HW** soft key.

▶ Press the **CC Data** soft key.

Note

The hardware information displayed for the CC is only valid if a CC with a parallel interface is used (no HSCI).

The function displays the following variables (first CC in left column – second CC in right column):

Variable	Display
CmdFrmHost	Current command to the CC
AchsInfoFrmHost	Associated axis information
MsgInfoFrmHost0 .. 3	Additional information on the command
CmdToHost	Current command from the CC
AchsInfoToHost	Associated axis information
MsgInfoToHost0 .. 3	Additional information on the command
WatchDogFrmHost	Watchdog to the CC
WatchDogToHost	Watchdog from the CC
Irq_Cycle	Synchronization between CC0 and CC1
UnixTime	Not used
DriveOffMask	Not used
ExtDriveReleaseMsk	Not used
LtRdyState	The 8-bit word (oooo oooo) contains the following information ("x" indicates the bit(s) containing the information): ■ oooo ooxo (x=1): DC-link voltage too high (Port input: "–ERR.UZ.GR" = low level) ■ oooo oxoo (x=1): Temperature of heat sink too high (Port input: "–ERR.TEMP" = low level) ■ ooox oooo (x=1): DC-link current too high (Port input: "–ERR.IZ.GR" = low level) ■ ooxo oooo (x=1): Power supply unit not ready (Port input: "RDY.PS" = low level) ■ oxoo oooo (x=1): Leakage current too high (Port input: "–ERR.ILEAK" = low level)

Variable	Display
DriveStatus	Shows the status of the drives in bit-encoded representation. The bit position of a drive depends on the index in the CC. This index is shown in the **axIndex** field (OLM function: IPO/Axes/CC Data). ■ Bit is 0: Drive is switched off or does not exist ■ Bit is 1: Drive is on
AxRelConnect	■ Bits 0 to 7: Indicate the status at connection X150 (axis-specific enabling) ■ Bits 8 to 15: Indicate the enabling status of the axes of the first CC
AxRelConnectSI	■ Bits 0 to 7: Indicate the status at connection X151 (axis-specific enabling) ■ Bits 8 to 15: Indicate the enabling status of the axes of the second CC

Command to the CC (CC Command)

This function transfers commands directly to the CC.

Danger

Please note that commands transferred to the CC with this function are not checked by the system.

Do **not** use these functions to switch drives on or off, since the brakes will not be activated or deactivated.

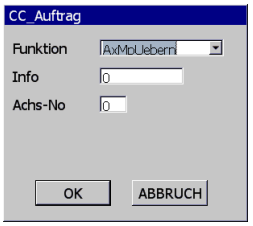
To select the **CC Command** function:

- Press the **Login Ipo** soft key.

- Press the **HW** soft key.

- Press the **CC Command** soft key for the OLM to open the "CC Command" dialog box.
- Enter the required data in the dialog box (see below).
- Enter the required data in the dialog box (see below).
- Confirm with **OK**—the OLM sends the defined command to the CC.

Dialog box **CC_Command**:



Dialog box entries:

- **Function:** Command sent to the CC
  - AxMpUebern
  - MotTemp
  - AntriebEin
  - AntriebAus
  - Status
  - CHK_PHIFELDREF
  - SHOW_HW_CO...
- **Info:** Not used
- **Axis no:** Logical axis number

Data of the static RAM (S-RAM)

This function displays the data stored in the static RAM of the IPO.

To select the **S-RAM** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **HW** soft key.

▶ Press the **S-RAM** soft key.

The function displays the following variables:

Variable	Display
kennung1	Internal code
kennung2	Internal code
absIstPos	Switch-off position of the individual axes
absIstPosNonMod	Switch-off position of the individual axes
refPosition	Reference position of the individual axes
modCounterEndat	Overflow of multiturn EnDat encoder
checkSum	Checksum of current machine parameters
valid	Validity code

Data of the analog outputs (Analog Output)

Use **AnalogOutput** to display the nominal commands. – The values of the outputs are not returned.

To select the **Analog Output** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **HW** soft key.

▶ Press the **Analog Output** soft key.

The function displays the values of the analog outputs in [V]:

- Output1
- Output2
- ...
- Output16

Counter function blocks of the MC (GAL Data)

GAL Data allows you to display the internal registers of the counter function blocks of the MC.

To select the **GAL Data** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **HW** soft key.

▶ Press the **GAL Data** soft key.

The GAL Data variables have only an IPO-internal meaning:

Variable	Display
reg0_low	
reg0_mid	
reg0_hig	
reg1_low	
reg1_mid	
reg1_hig	
init_reg_1	
cntrl_reg_1	
RI_reg	
latch_reg	
irq_reg	
offset00_reg	
offset90_reg	
timer_reg	
cntrl_reg_2	
cntrl_reg_3	

Hardware port states (HW-Ports)

The **HW-Ports** function displays the current status of some hardware ports. For the meaning of the displays, please refer to the description of the MC hardware. To make orientation easier, the relative addresses of the ports are indicated below. To obtain the actual port address, add the base address of the hardware to the relative address.

To select the **HW-Ports** function:

- Press the **Login Ipo** (or **Login SimIpo**) soft key.

- Press the **HW** soft key.

- Press the **HW-Ports I** or **HW-Ports II** or **HW-Ports III** soft key.

The functions display the status of the following hardware signals:

Variable	Text display
HW-Ports I	■ WD (IPO) ■ _SH2_p(CCU): Base address + 0x 330c ■ _NE1_p (I3): Base address + 0x330e ■ _NE2_p (I32): Base address + 0x3304 ■ _24V_plc2on: ■ _24V_plc3on: ■ _SH1AB_1_p: Base address + 0x3208 ■ _SHS1AB_1_p: Base address + 0x3204 ■ EN_SH2:Base address + 0x6000 ■ EN_NE1:Base address + 0x6004 ■ EN_PL:Base address + 0x6004 ■ EN_REG:Base address + 0x6006 ■ EN_MS:Base address + 0x6008 ■ EN_AT:Base address + 0x600a ■ EN_ACFAIL:Base address + 0x600c ■ IRQ_SH2:Base address + 0x6010 ■ IRQ_NE1:Base address + 0x6012 ■ IRQ_PL ■ IRQ_REG:Base address + 0x6016 ■ IRQ_MS:Base address + 0x6018 ■ IRQ_AT:Base address + 0x601a ■ IRQ_ACFAIL:Base address + 0x601c ■ IRQ_SYNCPWM:Base address + 0x601e

Variable	Text display
HW-Ports II	■ 3D-Signal (low active) ■ 3D-Bereit (low active) ■ 3d-Warng. ■ TT-Signal (low active) ■ TT-Bereit (low active) ■ X30-SpRef ■ WD-Reset ■ PLC-2*5V ■ iport1[0]: Base address + 0x3100 ■ iport1[1]:Base address + 0x3102 ■ iport1[2]:Base address + 0x3104 ■ iport1[3]:Base address + 0x3106 ■ iport1[4]:Base address + 0x3108 ■ iport1[5]:Base address + 0x310A ■ iport1[6]:Base address + 0x310C ■ iport1[7]:Base address + 0x310E ■ oport1[0]:Base address + 0x3302 ■ oport1[1]:Base address + 0x3102 ■ oport1[2]:Base address + 0x3104 ■ oport1[3]:Base address + 0x3106 ■ oport1[4]:Base address + 0x3108 ■ oport1[5]:Base address + 0x310A ■ oport1[6]:Base address + 0x310C ■ oport1[7]:Base address + 0x310E ■ _sg_inst_

Variable	Text display
HW-Ports II	■ IRQ SH2 ■ IRQ MNE1 ■ IRQ PLC ■ IRQ Reg/Spi1 ■ IRQ MS ■ IRQ Mitsu/Spi2 ■ IRQ AF ■ IRQ SyncPWM ■ IRQ busTimeout ■ IRQ VART1 ■ IRQ Vart2 ■ IRQ PF ■ IRQ HWM ■ IRQ WD ■ IRQ ■ IRQ ■ MSK SH2 ■ MSK MNE1 ■ MSK PLC ■ MSK Reg/Spi1 ■ MSK MS ■ MSK Mitsu/Spi2 ■ AMK AF ■ MSK SyncPWM ■ MSK busTimeout ■ MSK VART1 ■ MSK VART2 ■ MSK PF ■ MSK HWM ■ MSK WD ■ MSK ■ MSK enable all

Data of the encoder

With Encoder Data, you display the information on the encoder.

To select the **Encoder Data** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **HW** soft key.

▶ Press the **Encoder Data** soft key.

The **index** is structured as follows:

- 0 to 9: Encoders of the MC
- 10: Speed encoder (index 0; CC 0)
- 11: Position encoder (index 0; CC 0)
- 12: Speed encoder (index 1; CC 0)
- 13: Position encoder (index 1; CC 0)
- . . .
- 32: Speed encoder (index 5; CC 1)
- 33: Position encoder (index 5; CC 1)

The OLM displays the following data of the encoder:

Variable	Display
type	■ NotConnected ■ McPosEncoder ■ CcMotorEncoder ■ CcPosEncoder
inUse	The encoder is configured in the system (handwheel, position encoder or speed encoder).
usedFor	■ Nothing ■ Position ■ Speed ■ Handwheel
axisIdent	Index of the axis from axisList
axisKey	Key of the axis from axisList
ipoFactor	Fine interpolation factor
ccIndex	Index of the CC
dpIndex	Index on the CC
latch1Ok	
latch1	Counter increments of latch 1, including fine interpolation
latch2Ok	
latch2	Counter increments of latch 2, including fine interpolation (reference pulse or measuring pulse).

Variable	Display
EndatInfo	(Heading of the subsequent displays)
resolution	Resolution of the absolute track
absValue	Value of the absolute track
error	Error code
iresolution	Resolution of the incremental track
serialNumber	Serial number
multiturn	Maximum number of multiturn revolutions
mstype	Encoder code
incValue	Value of the incremental track during power-on

Data of the "Fast Inputs"

This function displays information on the fast inputs.

To select the **Fast Inputs** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **HW** soft key.

▶ Press the **Fast Inputs** soft key.

The **index** is structured as follows:

- 0 to 4: Fast inputs of the PLC
- 5 to 9: Trip dogs of the spindles

The OLM displays the following data of the fast inputs:

Variable	Display
inUse	Fast input is configured
usedFor	■ Nothing ■ Plc ■ Axis (spindle)
axisIdent	Index of the axis from axisList
axisKey	Key of the axis from axisList
number	Number of the input: ■ No HSCI: I0 to I31 ■ HSCI: All inputs are permitted
set	Status of the input (0/1)

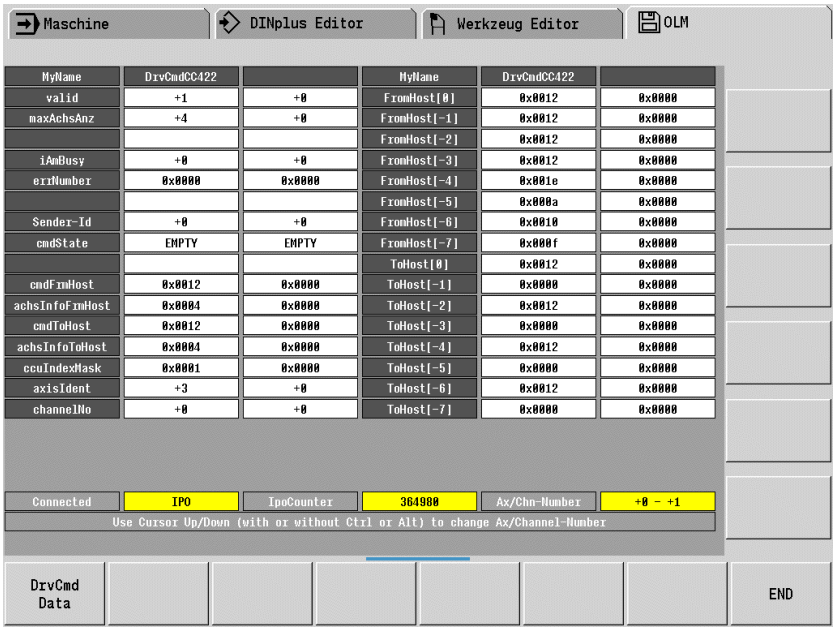
6.20.8 Group of drive commands

To select the **Drive Command** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

▶ Press the **HW** soft key.

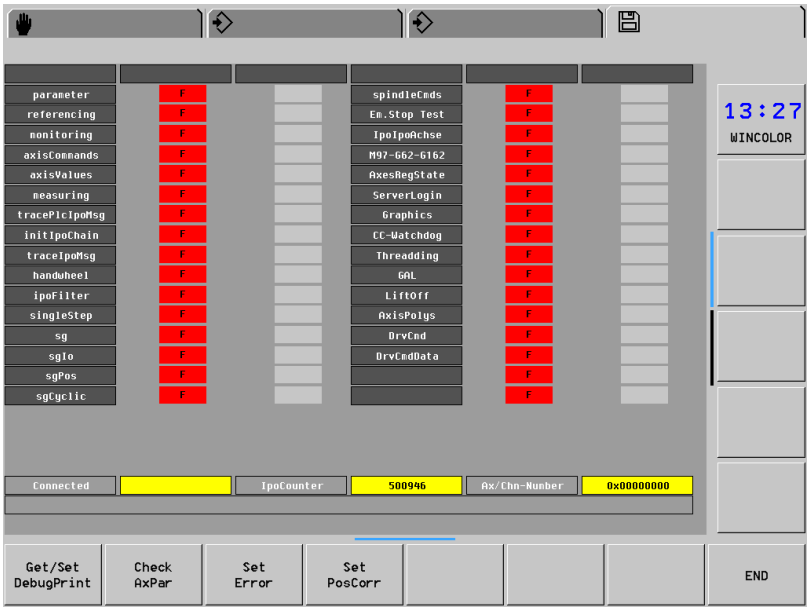
▶ Press the **CC HW Info** soft key.



Enabling debug outputs (Get/Set DebugPrint)

With **Get/Set DebugPrint,** you define the data to be logged and saved in the file r:\runtime_Xprint.txt. Data you identify by a "T" will be saved.

Example of a **selection**:



To select the **Get/Set DebugPrint** function:

Login Ipo

- Press the **Login Ipo** (or **Login SimIpo**) soft key.

Auxil

- Press the **Auxil** soft key.

Get/Set DebugPrint

- Press the **Get/Set DebugPrint** soft key for the OLM to open (Get...) the selection list (see figure below).
- Use the arrow keys to select the data whose identifiers you want to change.
- Use the **ENT** key to change the identifier (T or F).

The OLM uses the selected data to generate a bit line. The bit line is displayed in the bottom screen line at right. You can also use the bit line in the start batch of the IPO to start the IPO with the Debug Print function.

Meaning of the data:

Variable	Display
parameter	Output of information during parameter assignment. In addition, the parameters for every axis that are sent to the CC are written to the file r:\runtime_HelpTrace.txt.
referencing	Output of information during reference run
monitoring	Output of information during standstill monitoring and during monitoring of the absolute position (from zero pulse to zero pulse for distance-coded encoders)
axisCommands	Output of internal axis commands
axisValues	Output of information during actual-value transfer
measuring	Output of information on measuring process (probe on/off, monitoring, etc.)
tracePlcIpoMsg	All messages that are sent to the PLC-IPO are written to r:\runtime_HelpTrace.txt.
initIpoChain	Output during the initialization of the IPO chain
traceIpoMsg	All messages that are sent to the IpoInterpolator are written to r:\runtime_HelpTrace.txt.
handwheel	Output of information during the configuration and selection of the handwheel
ipoFilter	Output of information during the configuration and selection of two filters in the IPO chain
singleStep	Output of information during graphic simulation in the SingleStep mode of operation
sg	Output of additional information from the safety-oriented package (SG: safety-oriented control)
sgIo	Output of additional information from the safety-oriented package
sgPos	Output of additional information from the safety-oriented package
sgCyclic	Output of additional information from the safety-oriented package
spindleCmds	All spindle commands and their acknowledgments are recorded.
EmStopTest	Outputs during the emergency stop test
IpoIpoAchse	Output during the configuration of the axes of a channel (exchanging axes in and removing axes from the interpolation context)
M97-G62-G162	Outputs during the synchronization of several NC channels (not relevant for MANUALplus 620)
AxesRegState	Outputs at status change of axes in the controller

Variable	Display
ServerLogin	Log in to / log out of the IpoData server
Graphics	Request of workpiece positions for the on-line graphics and graphic simulation
CC Watchdog	Not used
Threading	Outputs during thread cutting
GAL	Not used
LiftOff	Lift off of tool during cycle stop
AxisPolys	Trace of the distance polynomials (result in file _HelpTrace.txt)
HirthAxis	Outputs during Hirth axis commands
DrvCmdData	Reserved
EthernetAll	Reserved
PathNames	Output of the names of all selected programs and cycles, including the paths
Watchdog	Output of watchdog states (only output once)
EthNet OnOff	Reserved
Terminal	Enable output to serial terminal
GetValues	Output when capturing positions with internal path calculation
Encoder	Reserved
RequestedPos	Output when approaching the restore position

Reload of trace information

The configuration data for the IPO trace is loaded from the configuration file O:\service\traceInfo.cfg.

To select the **Check AxPar** function:

- Press the **Login Ipo** (or **Login SimIpo**) soft key.

- Press the **Auxil** soft key.

- Press the **Reload TraceInfo** soft key for the OLM to load the configuration data.

Generating error messages (Set Error)

The **Set Error** function generates error messages. You define the error number and the error class.

To select the **Check AxPar** function:

▶ Press the **Login Ipo** (or **Login SimIpo**) soft key.

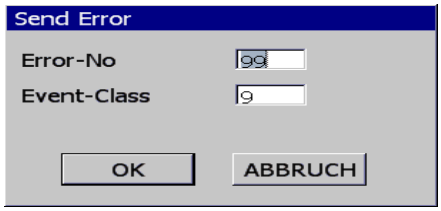
▶ Press the **Auxil** soft key.

▶ Press the **Set Error** soft key for the OLM to open the "Send Error" dialog box (see figure below).

▶ Enter the required data in the dialog box (see below).

▶ Confirm with **OK**—the OLM generates an error message.

"Send Error" dialog box:



Dialog box entries:

- **Error-No:** Error number (hexadecimal)
- **Event-Class:** Error class

Generating an asynchronous position compensation (Set PosCorr)

The **Set PosCorr** function generates an asynchronous position compensation (additive compensation). You define the number of the compensation and the compensation values.

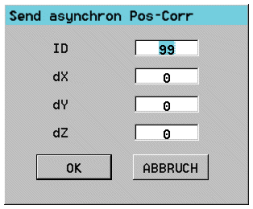
To select the **Set PosCorr** function:

- Press the **Login Ipo** (or **Login SimIpo**) soft key.

- Press the **Auxil** soft key.

- Press the **Set PosCorr** soft key for the OLM to open the "Send asynchron Pos-Corr" dialog box (see figure below).
- Enter the required data in the dialog box (see below).
- Confirm with **OK**—the OLM generates the compensation.

"Send asynchron Pos-Corr" dialog box:



Dialog box entries:

- **ID:** Compensation number
- **dX, dY, dZ:** Compensation values

Interface test

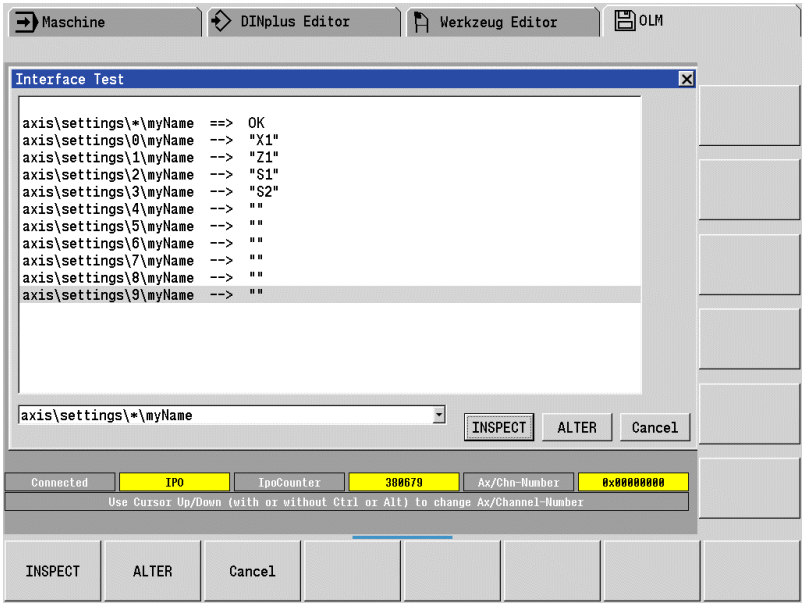
Interface for DNC mode and WinCast.

To select the function:

- Press the **Login Ipo** (or **Login SimIpo**) soft key.

- Press the **Auxil** soft key.

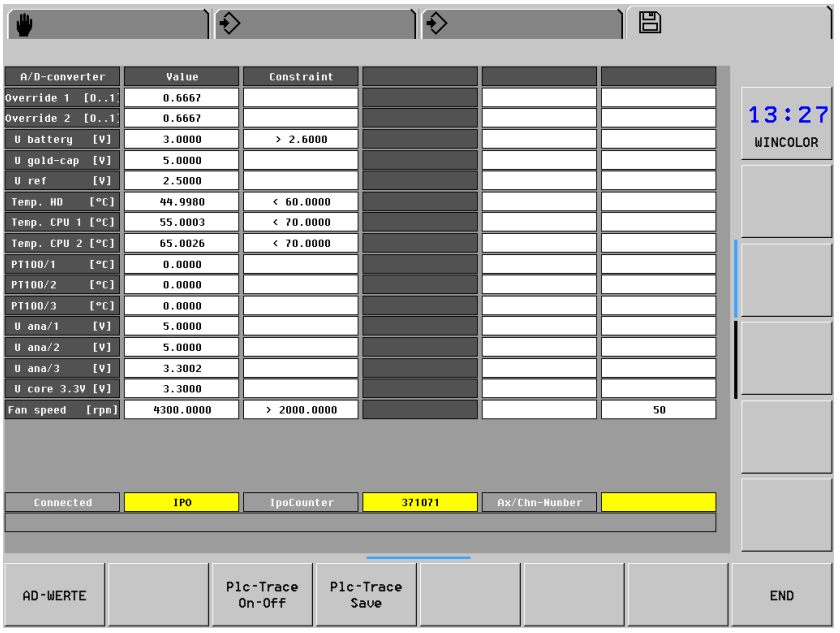
- Press the **Interface Test** soft key for the OLM to open the dialog box (see figure below).
- Set and check the required interface.



Displaying the values of the analog-to-digital converter

Note

The displayed information is specific to the control and the machine. Refer to the technical documentation to find out which analog values are assigned on your control and the meaning they have.



To select the **AD values** function:

▶ Press the **Login Plc** soft key.

▶ Press the **AD VALUES** soft key.

The function displays the values measured by the analog inputs as well as the permissible limits of some temperature and voltage values:

- **Value** column: Measured values, converted to units used internally by the PLC
- **Raw value** column: Values from the encoder, standardized to 16-bit format
- **Constraint** column: Limit values
- **Sample interval** column: PLC interval during which the value was read (0=simulated value)

Displays:

Variable	Display
ovr1 ovr2	Values of the override potentiometer (* 10000)
battery	Battery voltage (* 1000)
goldCap	"Gold-cap" voltage (* 1000)
caseTemp	Temperature of the control (* 10)
supply5V	5 V supply voltage at the main board (* 1000 / 2)
supply3V	3.3 V supply voltage at the main board (* 1000)
pt100_1 .. pt100_3	Temperature inputs (X48) of the MC (* 10)
u_1 .. u_3	Voltage of the analog inputs (X48) of the MC (* 1000)
tempCpu1	Temperature of CPU 1 (* 10)
tempCpu2	Temperature of CPU 2 (* 10)
caseFan	Fan speed [rpm]

PLC trace

The PLC trace function saves the PLC modules called and the errors that have occurred during the module call. Depending on the setting of the PLC-TRACE ON-OFF soft key, the following modules are saved:

- **Plc Trace On:** The PLC trace saves all module calls of the real-time thread and the submit/spawn thread.
- **Plc-Trace Off:** The PLC trace saves only module calls that generate an error.

The PLC trace saves the following information for each module call:

- IPO counter
- Module called
- Error number

Press the PLC-TRACE SAVE soft key for the OLM to save the PLC trace data in the file r:\runtime\=APIModCall.txt.

PLC trace on/off

To define **Plc-Trace On-Off**:

- Press the **`Login Plc`** soft key.

- Press the **`Plc-Trace On-Off`** soft key for the PLC-trace status to be changed.

Plc-Trace Save

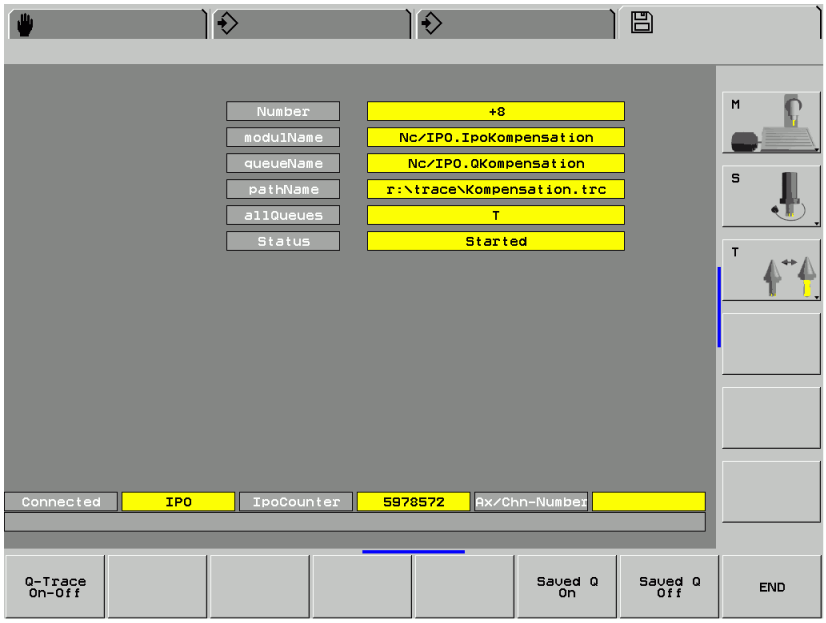
To select the **Plc-Trace Save** function:

- Press the **`Login Plc`** soft key.

- Press the **`Plc-Trace Save`** soft key for the PLC trace data to be saved.

6.20.11 Queue trace

The Q trace records the messages of the selected queues and saves them in a file.



After the trace function has been selected, the OLM displays the data of the queue at the top of the screen (see figure).

Trace

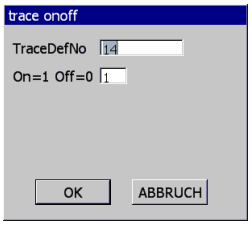
- Press the **Trace** soft key for the OLM to display the data of a queue.
- **Cursor Up**: Displays the next queue.
- **Cursor Down**: Displays the previous queue.
- **CTRL + Cursor Up**: Scrolls forward in increments of 10.
- **ALT + Cursor Up**: Scrolls forward in increments of 100.
- **CTRL + Cursor Down**: Scrolls backward in increments of 10.
- **ALT + Cursor Down**: Scrolls backward in increments of 100.

Activating a Q trace

The OLM saves the queues to be traced in a file. Define the entries in this file as follows:

- Press the **Q-Trace On-Off** soft key for the OLM to open the "trace onoff" dialog box.
- Enter the required data in the dialog box (see below).
- Conclude with **OK**.

"trace onoff" dialog box:



Dialog box entries:

- TraceDefNo: Enter the number of the queue (after selecting Trace, you can view details of the queue—see above).
- On/Off: Enter 0 or 1.

Deleting trace information

When the trace is activated, the data is "appended" to the existing trace file. You can delete the file as follows:

- Press the **Clear Saved Q** soft key for the OLM to delete the existing trace file.

All entries made after that will be entered into a new file.

Saving trace information

Select how the Q trace is to save the trace information:

- Press the **Saved Q On Compact** soft key for the OLM to save the trace information in compact form (single-line).

- Press the **Saved Q On Pretty** soft key for the OLM to save the trace information in structured form.

Stopping a Q trace

- Press the **Saved Q Off** soft key for the OLM to stop the trace.

"Tipo Command" soft key

The "Tipo Command" function is provided for internal tests.

6.20.12 Frequent causes of error

Servo drive cannot be switched on

The servo drive cannot be switched on or does not move:

1 Check whether the drive was enabled by the CC.
Selection: Login Ipo/Axes/Ipo Act State 1: **ANTRIEB_FREI** ()

ANTRIEB_FREI=false: Presumably an error on the CC or a hardware problem

2 Check whether "Drive on" was requested by the PLC.
Selection: Login Ipo/Axes/Plc Nom State: **AntriebEin** ()

AntriebEin=false: probably error in the PLC program

3 Check whether the drive was switched on.

Selection: Login Ipo/Axes/Ipo Act State 1: **ANTRIEB_EIN** ()

ANTRIEB_EIN=false: probably IPO internal error

4 Check whether position feedback control was requested by the PLC.
Selection: Login Ipo/Axes/Plc Nom State: **PosCtrlRequest** (does not apply to spindles)

PosCtrlRequest=false: probably error in the PLC program

5 Check whether position feedback control is active.
Selection: Login Ipo/Axes/Ipo Act State 1: **LGR_AKTIV**

6 Check whether "feed rate enable" was set by the PLC.
Selection: Login Ipo/Axes/Plc Nom State: **VorschubFreigabe**

Servo drive does not move

The servo drive cannot be switched on or does not move although all enabling commands are available—check the following variables:

1 The maximum permissible axis feed rate must be > 0.
Selection: Login Ipo/Axes/Plc Nom Data: **MaxAchsVorschub**

MaxAchsVorschub = 0: probably an error in the PLC program

2 The axis override must be > 0.
Selection: Login Ipo/Axes/Plc Nom State: **AchsOverride**

AchsOverride = 0: probably error in the PLC program

3 The IPO nominal speed must not be equal to 0.
Selection: Login Ipo/Axes/Ipo Act Data: **absSollV**

absSollV = 0: probably IPO internal error

4 The IPO actual speed must not be equal to 0.
Selection: Login Ipo/Axes/Ipo Act Data: **absIstV**

absIstV = 0: probably IPO internal error

7 Machine Interfacing

7.1 Display and Operation

The display screen of the MANUALplus 620 is divided into separate windows. The user can select the operating functions by soft key. Please also refer to the User's Manual.

7.1.1 Unit of measurement for display and operation

Settings in the configuration editor	MP number
System DisplaySettings CfgUnitOfMeasure **unitOfMeasure**	 101101

MP_unitOfMeasure is evaluated by the following functions or modes of operation:

- Machine display
- Entries in the **Manual Operation, El. Handwheel** and **Positioning with MDI** operating modes
- Entries in the configuration editor

NC programs have a specific code for the unit of measurement.

▶ In **MP_unitOfMeasure,** you define whether display and operation are in metric or inch mode.

Input or display	Metric	Inches
Coordinates, linear dimensions, compensation values, etc.	mm	in.
Feed rate (feed rate per minute, feed rate per revolution)	mm/min; mm/rev	in./min; in./rev
Cutting speed	mm/min	ft/min

Number of decimal places	Metric	Inch
Coordinates, linear dimensions, etc.	3	4
Compensation values	3	5

MP_unitOfMeasure

Unit of measure for display and user interface
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **Metric**
Metric system
Inch
Inches

Default: Metric

Access: LEVEL3

Reaction: RUN

Settings in the configuration editor	MP number
System	
DisplaySettings	
CfgDisplayLanguage	
ncLanguage	101301
plcDialogLanguage	101302
plcErrorLanguage	101303
helpLanguage	101304

The MANUALplus 620 distinguishes between conversational languages for the following areas:

- NC operation
- PLC operation
- PLC error messages
- Online help

The **path** for the dialog text files is permanently defined. The language abbreviation is at the end of the path. You define the language abbreviation in the parameters of the **CfgDisplayLanguage** object.

▶ In the parameter object **CfgDisplayLanguage,** you define the languages you want to use.

These directories are:

%OEM%\PLC\LANGUAGE\	cs	(Czech)
	da	(Danish)
	nl	(Dutch)
	en	(English)
	fi	(Finnish)
	fr	(French)
	de	(German)
	it	(Italian)
	pl	(Polish)
	pt	(Portuguese)
	es	(Spanish)
	sv	(Swedish)
	hu	(Hungarian)
	ru	(Russian)
	zh	(Chinese simplified)
	zh-tw	(Chinese traditional)
Additional conversational languages (Option #41)		
	sl	(Slovenian)
	sk	(Slovak)
	lv	(Latvian)
	no	(Norwegian)
	ko	(Korean)
	et	(Estonian)
	tr	(Turkish)
	ro	(Romanian)
	lt	(Lithuanian)

If the dialog text files for the selected language are not on the PLC partition, the error message **LANGUAGE LOAD ERROR** appears. The MANUALplus 620 will then try to open the dialog text file in the directory "..\en" (English).

The **file names** of the dialog text file are the same for all languages. The file names are usually defined in parameters.

- Store the dialog texts you created under the same file name in permanently defined directories.

In **MP_ncLanguage** you define the end of the path indicating the NC conversational language (language abbreviation).

MP_ncLanguage

NC conversational language
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
ENGLISH
GERMAN
CZECH
FRENCH
ITALIAN
SPANISH
PORTUGUESE
SWEDISH
DANISH
FINNISH
DUTCH
POLISH
HUNGARIAN
RUSSIAN
CHINESE
CHINESE_TRAD
SLOVENIAN (software option #41)
SLOVAK (software option #41)
LATVIAN (software option #41)
NORWEGIAN (software option #41)
KOREAN (software option #41)
ESTONIAN (software option #41)
TURKISH (software option #41)
ROMANIAN (software option #41)
LITHUANIAN (software option #41)

Default: ENGLISH
Access: LEVEL2
Reaction: RUN

In **MP_plcLanguage** you define the end of the path indicating the PLC conversational language (language abbreviation).

MP_plcDialogLanguage

PLC conversational language
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See **MP_ncLanguage**
Default: ENGLISH
Access: LEVEL2
Reaction: RUN

In **MP_plcErrorLanguage** you define the end of the path indicating the PLC error messages (language abbreviation).

MP_plcErrorLanguage

Language for PLC error messages
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See **MP_ncLanguage**
Default: ENGLISH
Access: LEVEL2
Reaction: RUN

MP_helpLanguage is used to define the end of the path of the help texts (language abbreviation).

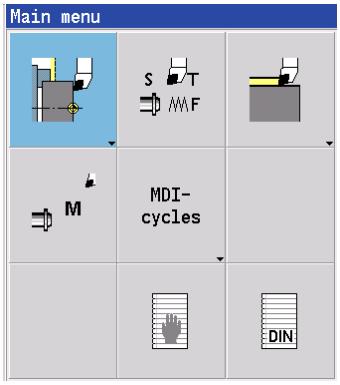
MP_helpLanguage

Language for online help
Available from NCK software version: 597 110-01.

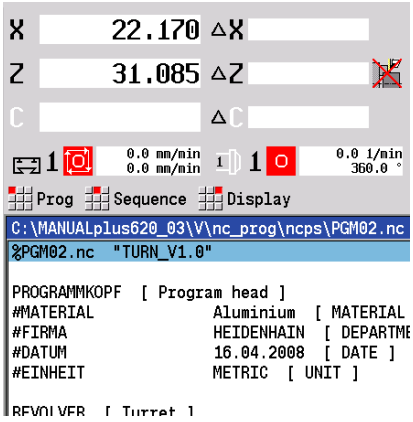
Format: Selection menu
Selection: See **MP_ncLanguage**
Default: ENGLISH
Access: LEVEL2
Reaction: RUN

Settings in the configuration editor	MP number
System DisplaySettings CfgMMISettings **extManualMode** **extProgramMode**	 604901 604902

Machine parameter **MP_extManualMode** can be used to expand the menu guidance of the user interface (3x3 menu) in the **Machine** operating mode. The manual cycles are consolidated to the submode MDI, which vacates menu items so that new functions can be assigned to them.



With **MP_extProgramMode** a horizontal menu is inserted beneath the dashboard in the **Program Run** operating mode in order to enable new functions to be selected. The new functions are described in the User's Manual of the control.



MP_extManualMode

Enhanced Machine operating mode
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: **TRUE**
Enhanced menu guidance in the user interface is active
FALSE
Enhanced menu guidance in the user interface is inactive

Default: -

Access: LEVEL3

Reaction: NOTHING

MP_extProgramMode

Enhanced Program Run operating mode
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: **TRUE**
Horizontal menu shown in Program Run
FALSE
Horizontal menu not shown in Program Run

Default: -

Access: LEVEL3

Reaction: NOTHING

7.1.4 Access rights to NC files

You can assign access rights to NC programs (*.nc) and NC subprograms (*.ncs). If you add an underscore as prefix before the file name (_*.nc, _*.ncs), then the respective file will not become visible in the **TNC:\nc_prog\ncps** directory until the code number 95148 has been entered.

7.1.5 Code numbers

General information

Press the soft key with the key symbol in the **Organization** operating mode in order to enter code numbers. With these code numbers you can activate certain functions.

Overview

The following code numbers have a fixed meaning:

Code number	Function
0	After access, deletion of the soft keys for the **Machine Parameter Programming, Oscilloscope** and **PLC Programming** operating modes.
123	Calls machine parameters that are accessible to the user. Additional enabling of the special function for deleting all tools ("Delete all" soft key) in the tool editor.
1234	PLC user parameters
95148	Selects the **Machine parameter programming** mode of operation. Additional enabling of the input fields "Set axis values" in the menu window for the X and Y axes.
231019	Software update
531210	Delete nonvolatile PLC operands and stop control operation. Then you must restart the control.
654321	Online monitor (OLM)
688379	Oscilloscope
13852	Enable the commissioning tool for analog axes (the soft key for accessing the tool is provided in the startup screen of the internal oscilloscope.)
75368	Adjustment of analog axis offsets
807667	Select the **PLC Programming** mode of operation.
6871232	Tests the internal EMERGENCY STOP (as of SW02)
857282	Reset the operating times
NET123	Ethernet settings
SIK	Open the menu for enabling software options.

PLC operand

The code of the entered code number is entered in **NP_GenModCode.** You can evaluate this code and define your own functions for code numbers, or disable fixed code numbers.

PLC operand / Description	Model
NP_GenModCode Code of the code number last entered	D

Settings in the configuration editor	MP number
System	
CfgMachineSimul	
simMode	100201
skipReferencing	100202
skipEmStopTest	100203

With **MP_simMode**, you can switch the MANUALplus 620 into a programming-station mode. This way the control can be used as a simple programming station. No drives are enabled. You can create and test NC programs. The operation of the machine is simulated in the programming station mode. As OEM, you have access to the machine configuration in the programming station mode. This enables you to adapt the MANUALplus 620 to the machine before actual commissioning.

The **MP_simMode** parameter offers four different setting possibilities for the programming station mode:

If **FullOperation** is set, the MANUALplus 620 starts in normal operation. The programming station mode is deactivated. All drives and the PLC are active.

Choose the **CcOnly** setting in order to simulate the CC controller unit while the PLC is active. In this case all PLC inputs and outputs, as well as the emergency-stop loop (X41/34 and X42/4), must already be connected correctly in order to switch the MANUALplus 620 on correctly. As of NCK software level 597 110-04, analog values are only simulated and are no longer output if axes are moving.

Danger

With the **CcOnly** setting, all axes are automatically switched to test mode. Hanging axes are not braked in test mode.

Hanging axes require a 100% compensation for weight.
Make sure that hanging axes are adequately supported.

Choose **CcAndExt** in order to simulate the CC controller unit and all PLC inputs and outputs. The PLC runs in simulation mode, and the emergency-stop loop and PLC inputs and outputs are not interrogated.

If **Delivery** is set, during startup of the MANUALplus 620 all axes are set to the test mode, and a switch-on of the axes is prevented. The user should then be able to start the control, even with an incomplete or faulty axis configuration in order to put the axes into operation. After the configuration of all axes has been completed, the control can be switched to full operation (FullOperation).

In order for the new settings to become active after changes in **MP_simMode**, the MANUALplus 620 must be restarted.

MP_simMode

Sets type of programming station mode
Available from NCK software version: 597 110-02.

Format: Selection menu

Selection: **FullOperation**
Programming station mode is switched off, the emergency-stop loop (X41/34 and X42/4) must be complete. The drives are moved.

CcOnly
Simulation of the CC controller unit. All PLC inputs and outputs, as well as the emergency-stop loop, must be connected correctly in order to switch the control on correctly.

CcAndExt
Simulation of the CC controller unit and all PLC inputs and outputs. The emergency-stop loop does not need to be complete. The PLC runs in simulation mode.

Delivery
When the control starts up, the axes are put into test mode and cannot be activated.

Default: FullOperation

Access: LEVEL3

Reaction: RESET

Traversing the reference position of the axes can be skipped in programming station mode.

Set the **MP_skipReferencing** parameter to the value TRUE in order to set the axes directly on the reference position when the control is started.

MP_skipReferencing

Fast reference run
Available from NCK software version: 597 110-02.

Format: Selection menu

Selection: **FALSE**
The axes are not set on the reference position.
TRUE
The axes are set directly on the reference position when the control is started.

Default: FALSE

Access: LEVEL3

Reaction: RESET

You can suppress the emergency-stop test with the **MP_skipEmStopTest** parameter.

MP_skipEmStopTest

No emergency-stop test is performed
Available from NCK software version: 597 110-02.

Format: Selection menu

Selection: **TRUE**
Emergency-stop test is not performed
FALSE
Emergency-stop test is performed

Default: FALSE

Access: LEVEL3

Reaction: RESET

7.1.7 Operating modes / control operation in the operating mode group

Operating modes

Note

All machining channels of an operating mode group have the same operating mode.

The symbolic PLC operands available depend on the control model. Sometimes, not all operands listed in the following table will be available on your control.

In the following PLC operands, the NC informs the PLC of the current operating mode of an operating mode group:

PLC operand / Description	Model
NN_OmgManual Manual Operation operating mode 0: Operating mode not active 1: Operating mode active	M
NN_OmgHandwheel Electronic Handwheel operating mode 0: Operating mode not active 1: Operating mode active	M
NN_OmgMdi Positioning with Manual Data Input operating mode 0: Operating mode not active 1: Operating mode active	M
NN_OmgProgramSingle Program Run, Single Block operating mode 0: Operating mode not active 1: Operating mode active	M
NN_OmgProgramRun Program Run, Full Sequence operating mode 0: Operating mode not active 1: Operating mode active	M
NN_OmgReference Reference operating mode 0: Operating mode not active 1: Operating mode active	M
NN_OmgJogIncrement Incremental Jog operating mode 0: Operating mode not active 1: Operating mode active	M

PLC operand / Description	Model
NN_OmgAuxiliaryMode Product specific code for special modes and submodes 101: Cycle in preparation If a program was selected but not yet started with Cycle ON (e.g. an M function is run in manual control) 102: Reference in preparation If the axis to be referenced is selected but the reference run has not yet been started 103: Single block in preparation If a program was selected and SINGLE BLOCK is active but has not yet been started with Cycle ON 104: Continuous mode in preparation If a program was selected and CONTINUOUS is active but has not yet been started with Cycle ON 105: Single block, continuous in preparation If a program was selected and SINGLE BLOCK and CONTINUOUS, but has not yet been started with Cycle ON 106: Measuring in preparation If in the "Measure the tool" overview the "Touch probe" soft key was pressed but the measuring cycle was not yet started 201: Teach-In Teach-In operating mode and cycle is started 202: Single cycle If, for example, an M function is run in manual control 203: Single program If a single cycle is started 204: Program cycle ON and continuous mode If a single cycle is started CONTINUOUS is set	D

Disabling operating modes

PLC Module 9285 "Set the access level" can be used to disable operating modes and write-access to files.

Module 9285 Set the access level

PLC Module 9285 locks/enables the following predefined functions. If the bit is set in the mask, the function is locked. If the bit is set to 0, the function is enabled:

Group 0: Disable operating modes	Bit mask	CNC PILOT 620 / MANUALplus 620
Manual Operation operating mode	0x01	Machine operating mode Disabling function in the main menu (9-item menu): - Setup - TSF menu
Electronic Handwheel operating mode	0x02	Function not available
Positioning with Manual Data Input operating mode	0x04	Machine operating mode Disabling function in the main menu (9-item menu): - Single paths - MDI cycles - Manual programs - DIN macros - M functions
Program Run, Single Block operating mode	0x08	Machine operating mode Disable the Program Run soft key
Program Run, Full Sequence operating mode	0x10	Machine operating mode Disable the Program Run soft key
smarT.NC operating mode	0x20	smart.Turn / Teach-In operating mode Disable editing

Group 1: Disable write-access to files	Bit mask	CNC PILOT 620 / MANUALplus 620
NC programs (*.H, *.I, *.HU, *.HP, *.HC, *.DXF)	0x01	Disable write-access rights smart.Turn: *.nc, *.ncs Teach-In: *.gmz ICP: *.gm*
Tool table	0x02	Tool editor operating mode Disable editing
Pocket table	0x04	Machine operating mode Disable editing
Preset table	0x08	Machine operating mode Disable setting up datums
Pallet table	0x10	Function not available

Group 2: Disable other functions	Bit mask	CNC PILOT 620 / MANUALplus 620
Manual probing	0x01	Machine operating mode Disable the "Tool measurement" soft key
Code numbers	0x02	Organization operating mode Disable the code number input

Call:

PS B/W/D/K <Group number>
0: Disable operating modes
1: Disable write-access to tables
2: Disable other functions

PS B/W/D/K/S<Bit mask>

CM 9285

PL B/W/D <Status>
0: Function performed
1: Illegal group number
2: Incorrect parameterization via bit mask
20: Module was not called in a spawn or submit job

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error (M4203)	0	Function performed
	1	Error code in W1022
NN_GenApiModule ErrorCode (W1022)	1	Invalid group number
	2	Invalid value for bit mask
	20	Module was not called in a spawn or submit job

Start/stop of the machining channels

With the following PLC operands, the PLC informs the NC of the start or stop status:

PLC operand / Description	Model
PP_OmgNcStart NC start for all machining channels of this operating mode group 0: NC start not active 1: NC start active	M
PP_OmgNcStop NC stop for all machining channels of this operating mode group 0: NC stop not active 1: NC stop active	M

7.1.8 Control operation in the machining channel

NC program run

Settings in the configuration editor	MP number
Channels ChannelSettings [Key name of the machining channel] CfgChannelFile **geoIniProgram** **geoCycleEnd** **geoCancelCycle**	 200402 200405 200406
System Paths CfgSystemCycle [Key name of the OEM system cycle] **path**	 102601

Note

The machine parameters described below (**MP_geoIniProgram, MP_geoCycleEnd,** etc.) of the **CfgChannelFile** config object are not evaluated by the MANUALplus 620.

Starting an NC program

The PLC executes an NC start with **PP_ChnNcStart**.

PLC operand / Description	Model
PP_ChnNcStart NC start or Cycle on 0: NC start not active 1: NC start active	M

With **NN_ChnNcStartExternRequest**, the NC asks the PLC to initiate an NC start. The PLC then uses **PP_ChnNcStart** to activate the NC start.

PLC operand / Description	Model
NN_ChnNcStartExternRequest External request for NC start 0: External NC start not requested 1: External NC start requested	M

Before running the actual NC program, the NC first starts the program defined in **MP_geoIniProgram,** and then the OEM program defined in **MP_Path.**

The NC program is executed immediately after the lead programs.

MP_geoIniProgram

Path and name of the lead program
Available from NCK software version: 597 110-01.

Format: String
Input: Path and name of the lead program
No entry: No lead program is executed.
Default: **%SYS%\jhcyc\sys\nc\iniprog.h**
Access: LEVEL3
Reaction: RESET

MP_path

Path and name of the OEM lead program
Available from NCK software version: 597 110-01.

Format: String
Input: Path and name of the OEM lead program
No entry: No OEM lead program is executed.
Default: –
Access: LEVEL3
Reaction: RUN

Terminating the NC program

In **NN_ChnProgEnd** the NC informs the PLC that an NC stop was executed because the program end has been reached.

PLC operand / Description	Model
NN_ChnProgEnd NC program end has been reached A "program end" command was executed (END PGM, M02 or M30). 0: End of NC program not reached 1: End of NC program reached	M

After the NC program has been run, the NC starts the program defined in **MP_geoCycleEnd.** The trailer program is executed immediately after the NC program.

MP_geoCycleEnd

Path/name of the trailer program for program end
Available from NCK software version: 597 110-01.

Format: String
Input: Path and name of the trailer program
No entry: No trailer program is executed.
Default: **%SYS%\jhcyc\sys\nc\progend.h**
Access: LEVEL3
Reaction: RESET

Interrupting an NC program

The PLC or NC can stop execution of the NC program. After interruption, the NC program is continued.

During program interruption, the axes can be traversed manually.

PLC stops NC program run:

The PLC executes an NC stop with **PP_ChnNcStop**.

PLC operand / Description	Model
PP_ChnNCStop NC stop or Cycle off 0: NC stop not active 1: NC stop active	M

NC stops NC program run:

The NC uses the following markers to inform the PLC of NC program interruption and the reason for the interruption:

- **NN_ChnStopExtern:** The program was interrupted because of an external request (e.g. Stop key).
- **NN_ChnProgStopped:** The program was interrupted because of a program stop (M0), the end of a block in Single block mode, etc.
- **NN_ChnProgStoppedAsync:** The program interruption was caused by an error, etc.

PLC operand / Description	Model
NN_ChnNcStopExtern NC stop or Cycle off NC stop is executed by the NC. 0: NC stop not executed 1: NC stop was executed by the NC	M
NN_ChnProgStopped NC program interruption The NC reports an asynchronous program interruption, such as at the end of a block in Single Block mode, M0, etc. 0: No NC program interruption 1: NC program interruption	M
NN_ChnProgStoppedAsync Asynchronous NC program interruption The NC reports an asynchronous program interruption, for example because of an error, etc. 0: No asynchronous NC program interruption 1: Asynchronous NC program interruption	M

Moving the axes during program interruption

During program interruption, the NC distinguishes between "manual traverse of the axes" and "returning to the contour." The NC indicates the status in the following markers:

PLC operand / Description	Model
NN_ChnProgManTraverse Manual traverse of the axes active (for lathe controls: inspection operation) 0: Manual traverse not active 1: Manual traverse active	M
NN_ChnProgReturnContour Return to contour active (after manual traverse or block scan) 0: Return to contour is not active 1: Manual traverse active	M

Canceling an NC program

In **NN_ChnProgCancel**, the NC informs the PLC of a program cancellation.

PLC operand / Description	Model
NN_ChnProgCancel NC program cancellation NC program cancellation due to an internal stop 0: No NC program cancellation 1: NC program cancellation	M

After the NC program has been canceled, the NC starts the program defined in MP_geoCancleCycle. The trailer program is executed immediately after the NC program has been canceled.

MP_geoCancelCycle

Path/name of the trailer program for program cancellation
Available from NCK software version: 597 110-01.

Format: String
Input: Path and name of the trailing program
No entry: No trailer program is executed.
Default: `%SYS%\jhcyc\sys\nc\cancelcyc.h`
Access: LEVEL3
Reaction: RESET

Block scan (mid-program startup)

PLC operand / Description	Model
NN_ChnBlockScan Mid-program startup (or block scan) active 0: Block scan not active 1: Block scan active	M
NN_ChnBlockScanStrobeTransfer Restore status at block scan (M/S/T/Q transfer) 0: Status not restored 1: Status restored	M

The PLC operand **NN_ChnBlockScan** is set when a start block is selected and the Cycle Start key is pressed. After the start block has been reached, the signal is reset.

If the Cycle Start key is pressed again, the signal **NN_ChnBlockScanStrobeTransfer** is set and all strobes collected during the block scan are executed by the control.

After the last strobe signal has been acknowledged, **NN_ChnBlockScanStrobeTransfer** is also reset to zero. This falling edge indicates to the PLC that the control is in a state that corresponds to the start block status.
All data relevant to the PLC has been processed, and therefore the NC program can be started.

Finding the NC program and block number

Module 9321 Find the current block number

Prefer Module 9322 to evaluate the NC program!

Module 9321 finds the current block number of the active NC program.

Constraints:

- Before an NC program is run for the first time, –1 is set as block number. For all traverse blocks that are not generated from an NC program, a cycle or an NC macro (e.g. PLC positioning, return-to-contour logic), –1 is set as block number.
- After the NC program is cancelled or after the end of the NC program, the last block number executed is returned.

Call:

```
PS    B/W/D/K   <String number>
CM    9321
PL    B/W/D     <Current block number>
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Block number has been found
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid string number

Module 9322 Information of the current NC program

The module supplies information about the execution of the current NC program.

When called in the cyclic program part, only the current block number is read. When called from a spawn or submit job, the current block number and the name of the current NC program, subprogram or cycle (depending on the setting) are determined.

Constraints:

- Because of the geometry look-ahead, the call from the cyclic PLC program only supplies the block number in real time, but no information about the NC program.
- Call from the cyclic PLC program:
  For all traverse blocks that are not generated from an NC program, a cycle or an NC macro, block number –1 is read.
- Call from the cyclic PLC program:
  After the NC program is cancelled or after the end of the NC program, the last block number executed is returned.

Call:

PS B/W/D/K <Mode>
When called from a cyclic PLC program, the <Mode> setting is omitted. The block number of the active NC program is always returned.
When called from a spawn job or submit job:
0: String / block number and path refer only to the active NC (sub)program. Block number from block scan.
2: Only the name of the NC main program without information about the block number; block number is set to 0 when executed correctly.

PS B/W/D/K <String number for path of the NC program or cycle>
Call from a cyclic PLC program: Without effect.

CM 9322

PL B/W/D <Block number>
–1: Error if error marker is set
–1: Block number in certain cases, if call was from a cyclic PLC program

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Successful execution of module
	1	Error. See NN_GenApiModuleError
NN_GenApiModule ErrorCode	1	Invalid mode programmed
	2	Invalid string number programmed
	12	Path name is longer than PLC string
	13	Internal error
	20	Module was not started from a spawn or submit job

Control in operation

In the **Positioning with Manual Data Input, Program Run, Single Block** and **Program Run, Full Sequence** operating modes, the NC uses **NN_ChnControlInOperation** to inform the PLC that the control is in operation. The status "control in operation" applies when the NC is executing a program, an M function or an axis movement.

NN_ChnControlInOperation is also set if the NC is processing a macro (cycle) in the background.

This applies to the following situations:

- Program selection
- Leaving the **Positioning with Manual Data Input** operating mode
- Control start-up (execution of the start-up cycle)
- When running cycles in the **Positioning with Manual Data Input** operating mode (the control-in-operation symbol is shown on the screen), e.g. while entering M functions or setting a datum.

PLC operand / Description	Model
NN_ChnControlInOperation Control is in operation 0: Control not in operation 1: Control in operation	M

M, S or T function in parallel with traverse motion

The PLC can execute M, S or T functions in parallel with the movement programmed in the same NC block.

Module 9404 Start movement when an NC strobe is present
The module starts the movement programmed in an NC block when a strobe that is effective at the beginning of the same NC block is still present.

- Do not execute the module in a submit or spawn process.

Call:

PS	B/W/D/K	<Channel number>
CM	9404	
PL	B/W/D	<Error number> 0: Successful 1: Invalid channel number 15: Module was called in a spawn job or submit job

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Movement successfully started
	1	Process not possible
NN_GenApiModule ErrorCode	1	An invalid channel number was transferred.
	2	Processing of NC part program is not synchronized
	24	The module was called in a submit job or spawn process

Error status

The NC informs the PLC of errors occurring in this machining channel. The PLC operands are used to distinguish between the reactions to errors (see "PET table (PLC error table)" on page 919).

PLC operand / Description	Model
NN_ChnErrorWarning Error or warning occurred 0: No error or warning occurred 1: NC error or warning occurred	M
NN_ChnErrorFStop Feed rate stopped because of an error 0: No feed stop triggered 1: Feed stop triggered	M
NN_ChnErrorNCStop NC stop because of an error 0: No NC stop triggered 1: NC stop triggered	M
NN_ChnErrorCancel Program canceled because of an error 0: No program cancellation 1: Program cancellation triggered	M
NN_ChnErrorEmergencyStop Emergency stop because of an error 0: No emergency stop triggered 1: Emergency stop triggered	M
NN_ChnErrorReset Reset because of an error 0: No reset 1: Reset triggered	M

NN_ChnErrorReset is not used at present because the PLC program stops when a reset error occurs.

Assignments in Manual modes of operation

In the manual operating modes, the peripheral user devices, such as monitor or keyboard unit, are assigned to a machining channel and a spindle. The machining channel is specified in the PLC operands **NN_GenOmgManual** and **NN_GenChnManual**, the spindle in **NN_GenSpiManual**.

The machining channel and the spindle are selected specifically for each control.

PLC operand / Description	Model
NN_GenOmgManual Selected operating mode group in manual operation	D
NN_GenChnManual Selected machining channel in manual operation	D
NN_GenSpiManual Selected spindle in manual operation	D

Asynchronous position compensation (additive compensation)

Settings in the configuration editor	MP number
System CfgPosCorrection **enable** **feed**	 100501 100502

The control supports asynchronous compensation. Asynchronous means that compensation values are entered during machining.

The control supports:

- Asynchronous tool compensation: This compensation is assigned to a tool, and corrects the tool lengths. The asynchronous tool compensation is cleared during a tool change.
- Asynchronous position compensation: This type of compensation, also known as additive compensation, is managed independently of channels in tables, and is activated and deactivated via G command. An asynchronous position compensation is in effect until it is deactivated or until the end of the program.

For compensation value tables for asynchronous position compensation, see MP_System/Paths/CfgTablePath/Add_Cor*/path (*=1: machining channel 1; *=2: machining channel 2; etc.)

If an asynchronous compensation is activated, deactivated or changed, then it is "corrected" with the velocity defined in MP_feed.

MP_enable is used to specify whether asynchronous compensations are calculated in the interpolator or already before the interpolator. If **MP_enable** = Off, the compensations are taken into account during interpretation of the NC program. Due to the large block scan required, it takes a certain amount of time for the compensations to take effect. As an alternative you can define **MP_enable** = On. The interpolator then takes the compensations into account. This speeds up the reaction time, but the processing effort necessary by the interpolator is increased considerably.

- Define **MP_enable** = On if you require a quick reaction time for asynchronous compensations. If this is not required, or if there is no asynchronous compensation, then the **MP_enable** = Off setting is recommended.
- In **MP_feed**, define the velocity at which asynchronous compensations are corrected.

MP_enable

Asynchronous position compensation on/off
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **On**
Switch asynchronous position compensation on
Off
Switch asynchronous position compensation off
Default: Off
Access: LEVEL3
Reaction: RUN

MP_feed

Velocity for asynchronous position compensation
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 000 000 to 99 960 [mm/min]
Default: 960
Access: LEVEL3
Reaction: RUN

7.1.9 Error messages and log files

The control displays errors in the header of the screen. Long error messages or error messages extending over more than one line are abbreviated. The complete information on all pending error messages is given in the error window.

Errors and system information (system start, system end, etc.) are entered in the error log file. The control saves every keystroke and the mouse events in the keystroke log file.

Error window

ERR

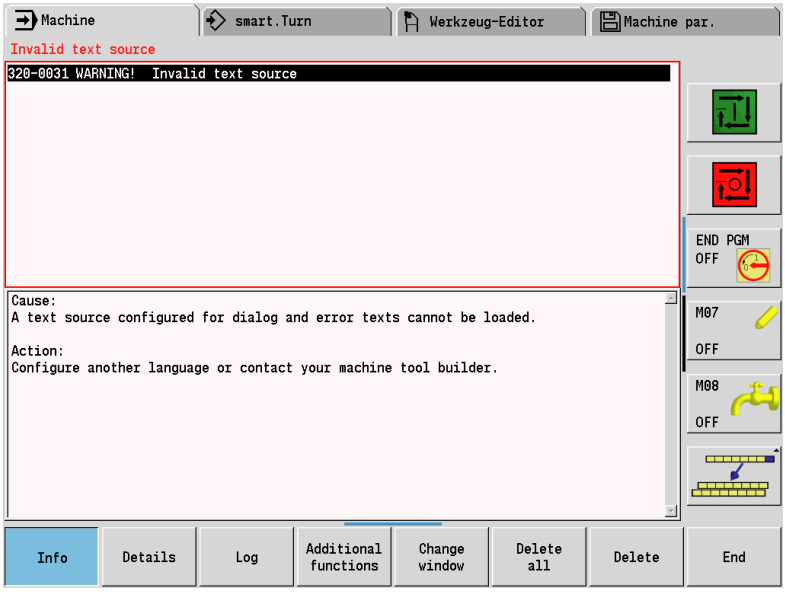
▶ Press the ERR key to call the error window

The error window contains the details of all errors that have occurred (see figure below, framed area).

To obtain information on the cause of error and the corrective action, proceed as follows (see figure):

Info

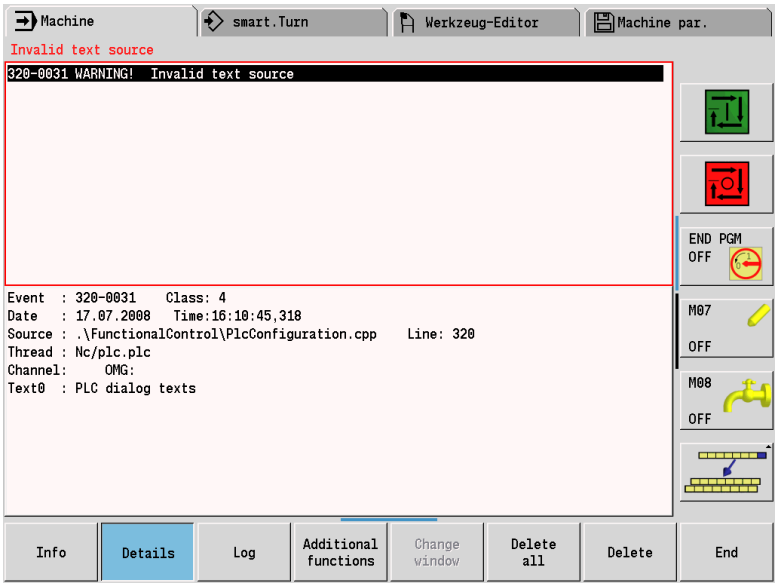
▶ Press the **Info** soft key



For further details regarding the software's internal error data, such as date, time, event class, line of the NC program, control program reporting the error, etc., proceed as follows (see figure):

DETAILS

▶ Press the **Details** soft key



Deleting errors

To delete an individual error:

▶ Position the cursor on the entry to be deleted:

DELETE

▶ Press the **DELETE** soft key.

To delete all errors contained in the error window:

DELETE ALL

▶ Press the **DELETE ALL** soft key

Information provided by the **error message:**

- Error number: Assigned by HEIDENHAIN or the machine tool builder
- Error class: Defines the control's reaction to this error (see table)
- Error text: Describes the error (in one or more lines) If the error occurs while an NC program is being run, the line of the NC program will also be indicated.

Overview of error handling in the control:

Error class	Reaction	Display	Log file entry	Acknowl-edgment	Error group
Ev_class_2	None	x			Warning
Ev_class_3	None		x		Warning
Ev_class_4	None	x	x		Warning
Ev_class_5	None	x	x	x	Error
Ev_class_6	Feed stop	x	x	x	Error
Ev_class_7	Program abortion	x	x	x	Error
Ev_class_8	Program aborts at stable position	x	x	x	Error
Ev_class_9	Emergency stop	x	x	x	Error
Ev_class_10	Reset	x	x	x	System error
Ev_class_11	NC stop	x	x		Error
Ev_class_12	NC stop	x	x	x	Error
Ev_class_13	Program abortion	x	x		Error
Ev_class_14	Reset – without output of error text ("Processor check error")	x	x	x	System error
Ev_class_15	Feed stop	x	x		Error
Ev_class_16	Emergency stop	x	x		Error
Ev_class_17	Display informational text	x	x		Info
Ev_class_18	None	x	x		Warning
Ev_class_19	Program abortion	x	x	x	Error

Ev_class_18 is used to report that service files were saved.

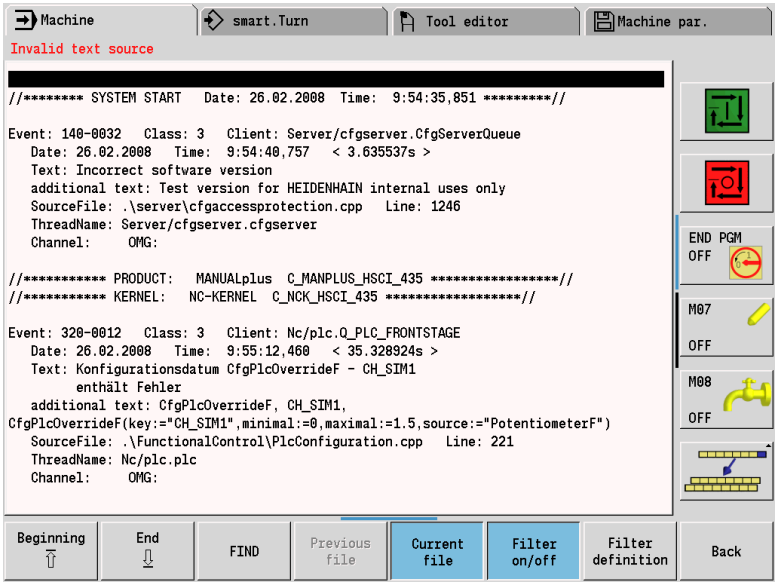
Error log file

The control saves all errors that occurred and the error information, including all details, in the error log file (see figure).

To call the error log file:

▶ Press the **LOG FILE** soft key

▶ Press the **Error Log File** soft key.



To move within the log file:

To the oldest entry:

▶ Press the **BEGIN** soft key

To the most recent entry:

▶ Press the **END** soft key

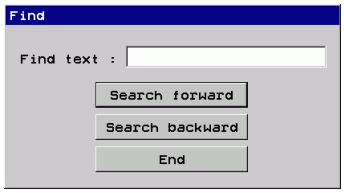
To view other log file entries:

▶ Move using the arrow keys (up arrow, down arrow, page up, page down)

To find a log file entry:

Call the "Find" dialog box:

- Press the **FIND** soft key



- Enter the search string
- Define the search direction

Current and previous error log file

The error log file uses two files, the **current file** and the **previous file.**

If the current file is full, the control switches the files. After converting the current file to the previous file, the control creates a new current file.

To switch between the current and the previous error log file:

- Press the **PREVIOUS FILE** soft key.

- Press the **CURRENT FILE** soft key.

Filter

Use a filter to limit the log file display to the following error groups:

- Information
- Warnings
- Errors
- System errors

In addition, you can select the following information:

- Date and time from which you want the log file contents to be displayed.
- Clients whose errors and error information are to be considered in the log file display.

Whether the **filter is taken into account** depends upon the setting of the **FILTER ON/OFF** soft key:

- Soft key active: Filter is taken into account
- Soft key not active: Filter is not taken into account

To set the filter:

▶ Press the **Filter definition** soft key

To display information:

▶ Press the **INFOS** soft key

To display warnings:

▶ Press the **WARNINGS** soft key

To display errors:

▶ Press the **ERROR** soft key

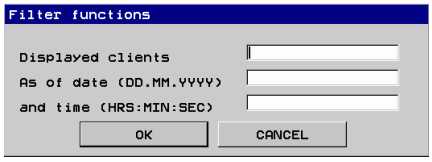
To display system errors:

▶ Press the **SYSTEM ERROR** soft key

To set the client and/or date and time:

Call the "Filter functions" dialog box

▶ Press the **FURTHER FILTER FUNCTIONS** soft key



▶ Enter the client and/or date and time

▶ Confirm your entry with **OK**

To **display** the **log file** under consideration of the new filter settings (prerequisite: the **FILTER ON/OFF** soft key is on):

▶ Press the **UPDATE** soft key.

Keystroke log file

The control saves all keystrokes and mouse events that occurred in the keystroke log file (see figure).

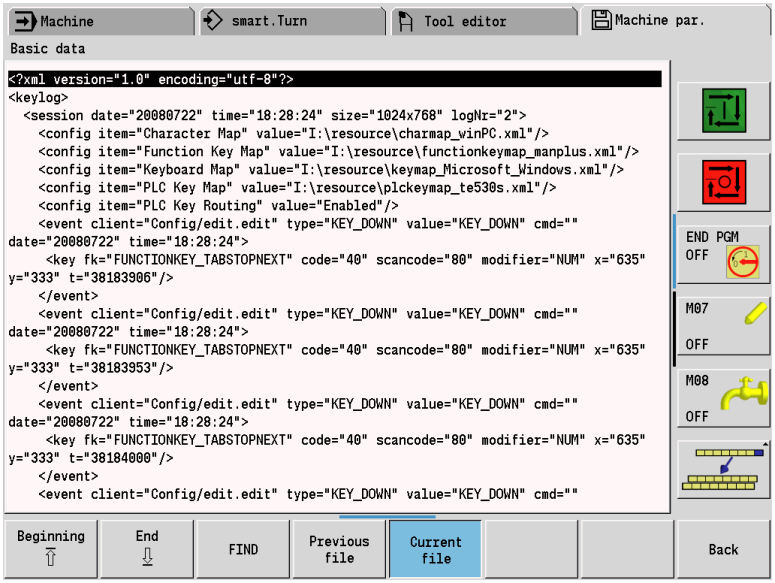
Call the keystroke log file from within the error system:

LOGFILE

▶ Press the **LOG FILE** soft key.

KEYSTROKE LOG FILE

▶ Press the **KEYSTROKE LOG** soft key.



To move within the log file:

BEGIN

To move to the oldest entry:

▶ Press the **BEGIN** soft key.

END

To move to the most recent entry:

▶ Press the **END** soft key

To view other log file entries:

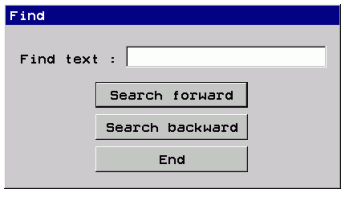
▶ Move using the arrow keys (up arrow, down arrow, page up, page down)

To find a log file entry:

FIND

Call the "Find" dialog box:

▶ Press the **FIND** soft key



▶ Enter the search string

▶ Define the search direction

Current and previous keystroke log file

The keystroke log file uses two files, the **current file** and the **previous file**:

If the current file is full, the control switches the files. The current file is converted to the previous file and the previous file to the current file. The contents of the previous file are deleted before new entries are made.

To switch between the current and the previous error log file:

▶ Press the **PREVIOUS FILE** soft key.

▶ Press the **CURRENT FILE** soft key.

Log

The accumulated keystrokes are now stored simultaneously with the control events in the log and are displayed in table view (see figure).
In order to be able to track machine operation or machine conditions systematically, detailed additional information is entered and stored simultaneously with all important log entries, such as keystrokes, errors, system errors or warnings.

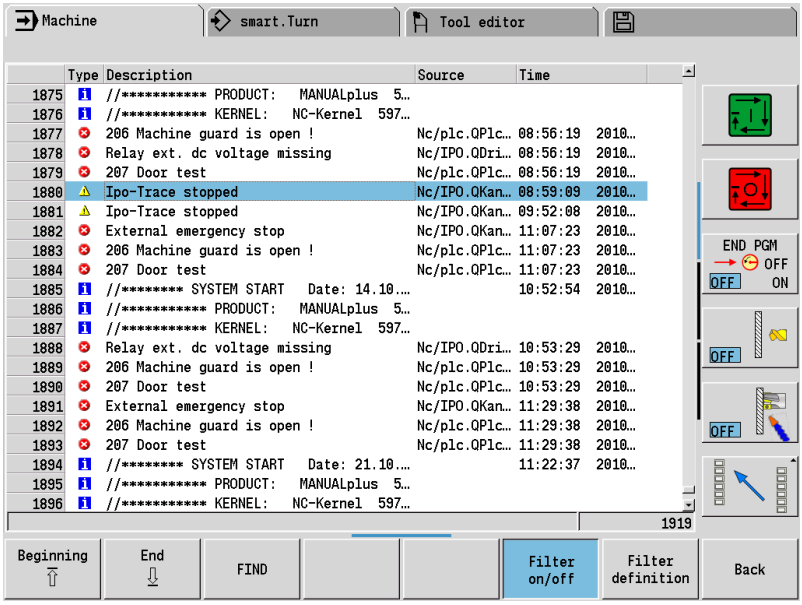
At least 4 weeks of control operation can be recorded with the log. The data is saved on the SYS partition and therefore cannot be changed by the user or OEM.

The log can be read with the programs TeleService or TNCremoNT and is part of the service files.

Call the log from within the error system:

▶ Press the **LOG FILE** soft key

▶ Press the **LOG** soft key



Moving within the log file:

To move to the oldest entry:

▶ Press the **BEGIN** soft key.

To move to the most recent entry:

▶ Press the **END** soft key

To view other log file entries:

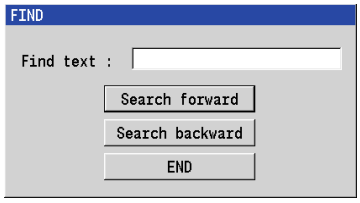
▶ Navigate in vertical direction by simply using the navigation keys

To find a log entry:

FIND

Call the "Find" dialog box:

- Press the **FIND** soft key



- Enter the search string
- Define the search direction

Filter

Use a filter to limit the log display to the following error groups:

- Information
- Warnings
- Errors
- System errors

In addition, you can select the following information:

- Date and time from which you want the log contents to be displayed.
- Clients whose errors and error information are to be considered in the log display.

Whether the **filter is taken into account** depends upon the setting of the **FILTER ON/OFF** soft key:

Filter ein/aus

- Soft key active: Filter is taken into account
- Soft key not active: Filter is not taken into account

To set the filter:

Filter definition

- Press the **Filter Definition** soft key

To display information:

INFOS

- Press the **INFOS** soft key

To display warnings:

WARNINGS

- Press the **WARNINGS** soft key

To display errors:

ERRORS

- Press the **ERROR** soft key

To display system errors:

SYSTEM ERRORS

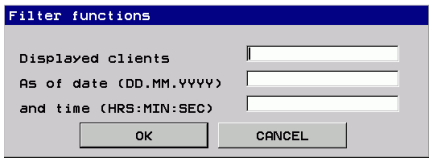
- Press the **SYSTEM ERROR** soft key

To set the client and/or date and time:

FURTHER FILTER FUNCTIONS

Call the "Filter functions" dialog box

- Press the **FURTHER FILTER FUNCTIONS** soft key



- Enter the client and/or date and time
- Confirm your entry with **OK**

To **display** the **log** under consideration of the new filter settings (prerequisite: the **FILTER ON/OFF** soft key is on):

UPDATE

- Press the **UPDATE** soft key.

Saving log files (service files)

You can save the error log file, keystroke log file, the log files of the IPO and PLC as well as the currently effective machine configuration and various other pieces of information on the hard disk of the control.

This function is particularly interesting if servicing becomes necessary and you want to transmit the log files and the control configuration to the HEIDENHAIN Service department.

The control automatically packs the data into a *.ZIP file.
Path: **TNC:\service*.zip**

To save log files:

- Call the error window by pressing the **ERR** key.

- Press the **LOG FILE** soft key.

- Press the **Service Files** soft key.

- The control automatically creates the *.ZIP file **TNC:\service*.zip.** ***** = number; if there is more than one service *.ZIP file on the control, the files are numbered in increasing order. The service1.zip file is always the most recent file.
  Up to 5 service ZIP files are saved (service1.zip to service5.zip).

Settings in the configuration editor	MP number
System Paths CfgPlcPath **errorTable** **errorText**	 102303 102304

PLC error messages are defined in the PET table (PLC Error Table). If the PLC detects an error, it is transferred to the error system by Module 9084, Module 9085 or by activating a marker defined in the PET table. The error system ensures that the error is displayed and processed. The PLC error messages are now displayed with the prefix PLC followed by the line number from the PET table (e.g. **PLC00239**).

Module 9086 is used to delete PLC error messages, and Module 9087 is used to interrogate the current status of the error message.

PET table (PLC error table)

- Enter the path and file name of the PET table in **MP_errorTable.**
- Enter the file name of the text file for PLC error messages in **MP_errorText.**

Note

A *.PET table is absolutely mandatory, since without it the PLC program cannot be compiled or activated.

Use the program "PLC-Text" to enter data in the PET table.

If a *.PET table contains more than 999 error messages, the excessive messages are ignored and the error message **PET table: Too many lines** appears.

MP_errorTable

PLC error message table
Available from NCK software version: 597 110-01.

Format: String
Input: Path and file name of the PET table, for example: %OEM%\table\
Default: –
Access: LEVEL2
Reaction: NOTHING

Priority of PLC error messages

You can enter a priority between 0 and 2 for the PLC error messages defined in the PET table. Priority 0 (error) is the highest priority, followed by priority 1 (warning) and priority 2 (info). PLC error messages triggering an EMERGENCY STOP receive the highest priority (independent of the priority from the PET table). Therefore, these error messages always appear at the first position in the error list. The error message **External EMERGENCY STOP** has a lower priority, but still a higher priority than the top PLC priority. This means that PLC error messages triggering an Emergency Stop always appear at the first position in the error list. They are followed by **External EMERGENCY STOP** and then by further PLC error messages, depending on their priority.

Error text file

Error texts are defined directly in the PET table (max. 32 characters; not language-sensitive) or in the error text file. In the error text file, you define the error text to be displayed as well as the information on the cause of error and corrective action.

Error text files are language-sensitive. The path for the error text file is permanently defined: %OEM%\plc\language\en (or another language abbreviation).

In MP_System/DisplaySettings/CfgDisplayLanguage/plcErrorLanguage, you define the language to be used.

You define the name of the error text files in MP_errorText.

MP_errorText

Text file for PLC error messages
Available from NCK software version: 597 110-01.

Format:	String
Input:	Example: **PLCErrorText.csv** The path **%OEM%\plc\language** is permanently defined. The last subdirectory is the language abbreviation for the respective conversational language, e.g. **en** for English.
Default:	–
Access:	LEVEL2
Reaction:	NOTHING

Structure of the PET table

The PLC error message table (*.PET) consists of the following columns, to which you can assign special attributes:

- **NR**
  Line number in the table. The modules select the PLC error message by assigning the line number.
- **ERROR**
  The error texts can be specified in the following manners:
  - Direct entry of the error text (max. 32 characters)
  - Line number of the PLC error text file (# <line no.>) defined in MP_errorText.
- **MARKER**
  The PLC error message can be activated without module call by setting the marker defined here. The marker is also set if the error message was activated through Module 9085. Enter the symbolic name of the marker to be set.
  Entry 0: No error marker

- **Error class:** The error class is defined in the following columns (See "Error status" on page 903). If none of these error classes is set in the PET table, NN_ChnErrorWarning is set.
  - **RESET**
    0: No NC reset upon activation of the error message (no system error).
    1: NC reset upon activation of the error message (system error). The PLC program stops.
  - **NC_STOP**
    0: No NC stop upon activation of the error message
    1: NC stop upon activation of the error message (NN_ChnErrorNcStop is set).
  - **NC_CANCEL**
    0: No NC stop with subsequent INTERNAL STOP upon activation of the error message
    1: NC stop with subsequent INTERNAL STOP upon activation of the error message (NN_ChnErrorCancel is set).
  - **F_STOP**
    0: Feed-rate enable is not influenced
    1: Feed-rate enable is reset upon activation of the error message (NN_ChnErrorFStop is set).
  - **EMER_STOP**
    0: No EMERGENCY STOP upon activation of the error message
    1: EMERGENCY STOP upon activation of the error message (NN_ChnErrorEmergencyStop is set).
- **CE**
  0: Error message can be deleted by the user.
  1: Error message cannot be deleted by the user.
- **PRIO**
  A priority of 0 to 2 can be entered for the error message, with priority 0 being the highest priority. If the PLC triggers more than one error at the same time, the errors with the highest priority are the first to be sent to the event server (error system).
- **WARN_LVL:** Not evaluated.

Structure of the error text file

In the error text file, there are four columns with the following meanings:

- **Reference number:** This reference is used in the PET table ("Error" column).
- **Error text:** Displayed error text.
- **Cause of error:** Text that is displayed under "Cause" after you have pressed the **Info** soft key.
- **Corrective action**: Text that the error system displays under "Action" after you have pressed the **Info** soft key

Module 9084 Display PLC error message with additional data

The module displays PLC error messages with additional data. You can insert placeholders (%s, %d, %f) at any position of the error texts. The placeholders are assigned the data from the module at run time. Only those place holders that are defined in the PLC error message will be replaced. %s is replaced by the string or the string content. The first occurrence of %d or %f in the PLC error message is replaced by the content of variable 1, and the second occurrence of %d or %f is replaced by the content of variable 2. %d is an integer, %f is a floating point number with three decimal places. Alternatively, you can define the number of decimal places with %.1f to %.6f.

If the module is called several times with the same line number of the *.PET table, the error message is entered only once in the queue. A maximum of 32 PLC error messages can be entered in the queue.

If an error marker is assigned in the PET table, it is set.

If the *.PET table or the line number is not found, the error message **PLC ERROR <line number>** appears.

Call:

```
PS      B/W/D/K   <line number of the *.PET table>
                  0 to 999: Line number
PS      B/W/D/K/S<Data for %s>
PS      B/W/D/K   <Data for %d or %f; variable 1>
PS      B/W/D/K   <Data for %d or %f; variable 2>
CM      9084
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	PLC error message with additional data displayed
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Line number missing
	8	Incorrect operating mode, compatibility error marker set
	23	Overflow of PLC error message queue

Module 9085 Display PLC error messages
The module transfers PLC error messages to the error system. The error message texts come directly from the compiled error table (.PET) or from the selected text file for PLC error messages. PLC error messages (except reset errors) can be deleted by Module 9086 or by the user. However, deletion can be disabled in the error table ("CE" column).

Up to 32 error messages can be placed in the queue.

If an error marker is assigned to the error, it is set.

System error: Is displayed without entry in the queue.

Error number –1: System error message **EMERGENCY STOP PLC** is displayed. This error message also occurs if no *.PET table was defined.

Error number not equal to –1 and no *.PET table selected:
System error message **PLC: ERROR TABLE MISSING**

Call:
PS B/W/D/K <Line number of the *.PET table>
0 to 999: Line number
–1: System error message **EMERGENCY STOP PLC**
CM 9085

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Error message displayed or in queue
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Line number missing
	8	Incorrect operating mode, compatibility error marker set
	23	Overflow of PLC error message queue, or too many error messages from string memory

Module 9086 Delete PLC error message

Use this module to erase all set PLC error messages or a specific error message. System errors cannot be deleted.

Call:

PS B/W/D/K <Line number of the *.PET table>
0 to 999: Line number
–1: Erase all PLC error messages

CM 9086

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Error message displayed or in queue
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Line number missing
	8	Incorrect operating mode, compatibility error marker set

Module 9087 Status of PLC error message

The module interrogates the status of a specific PLC error message, or the PLC error status in general. In addition, the number of the error message active on the screen and the total number of PLC error messages in the error list can be interrogated.

Call:

PS B/W/D/K <Line number of the *.PET table, status code>
0 to 999: Line number
–1: PLC error message, or general status
–2: Number of the active PLC error message
–3: Number of error messages in the *.PET table

CM 9087

PL B/W/D <Status/error code>
For code 0 to 999:
0: No error message with the number, or message deleted
–1: Line number does not exist
Bit 0 – PLC error message is displayed
Bit 1 – PLC error message in queue

For code –1:
0: No PLC error message
2: PLC error message in queue

For code –2:
≥ 0: Number of the displayed error
–1: No error in the *.PET table

For code –3:
≥ 0: Number of errors in the *.PET table

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Status information was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Invalid line number of status code

Entering data in log files

The error log file can be used by the PLC for diagnostic purposes.

Entering data **from the PLC** into the error log file:

- Use Module 9275 to write ASCII data into the error log file
- Use Module 9276 to write the contents of operands into the error log file

Note

Do not use Modules 9275 and 9276 in the PLC program as shipped. Instead, use them only for debugging. Otherwise the processing times could be increased and the hard disk could be written to unnecessarily.

Module 9275 Write ASCII data into the log

The module writes a character string from a PLC string or an immediate string into the error log file. The entry can be given a special identifier for fast finding or later editing.

A buffer of approx. 210 bytes is available for the data to be written (including the entry identification).

Call:

```
PS    B/W/D/K/S<Log entry>
                –1: No entry
PS    B/W/D/K/S<Log identifier>
                –1: No entry
PS    B/W/D/K  <Priority>
                0: Information
                1: Warning
                2: Error
CM    9275
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Entry was written
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Invalid priority
	2	Invalid string number or invalid immediate string
	12	No string end identifier
	20	Module was not called in a spawn job or submit job

Module 9276 Write operand contents into the log

The module writes the contents of PLC operands into the error log file. The entry can be given a special identifier for fast finding or later editing.

A buffer of approx. 210 bytes is available for the data to be written (including the entry identification).

The operands M/I/O/C/T are stored in binary format (e.g.110101), the operands B/W/D in hexadecimal format.

Call:

PS B/W/D/K <Identifier for operand name>
0: M (marker)
1: I (input)
2: O (output)
3: C (counter)
4: T (timer)
5: B (byte)
6: W (word)
7: D (double word)

PS B/W/D/K <Address of the first operand>

PS B/W/D/K <Number of operands>

PS B/W/D/K/S<Log identifier>
–1: No entry

PS B/W/D/K <Priority>
0: Information
1: Warning
2: Error

CM 9276

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Entry was written
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Invalid priority
	2	Invalid identifier for operand name
	3	Invalid first operand address
	4	Sum of first operand address and number of operands invalid
	5	Address is not a word/double-word address
	12	No string end identifier
	20	Module was not called in a spawn job or submit job
	36	Entry in the log was truncated after 210 characters

Settings in the configuration editor	MP number
System	
CfgDiagnosis	
traceDir	116604
osTraceFileSize	116601
tcpTraceFileSize	116602
ncTraceFileSize	116605
krnlTraceFileSize	116603

For expanded diagnostic purposes you can use the machine configuration to activate special logs for the following:

- Operating system diagnostics
- Network diagnostics
- NC software diagnostics
- NC kernel diagnostics

Note

Only HEIDENHAIN can evaluate the logs!

You activate log book creation through machine parameter by defining a maximum file size for the respective log file. Also, you have to configure a path in which the log files are saved. The optional parameters are defined below.

Note

The machine parameters described below change internal environment variables of the HEIDENHAIN operating system. Changing the parameters therefore causes the control to reboot!

In **MP_traceDir**, enter the target directory for the log files. If no path is entered under **MP_traceDir**, the control automatically saves the log files in the end-user partition (%USR%\) in the LOG directory.

MP_traceDir

	Target directory for log files Available from NCK software version: 597 110-04.
Format:	String
Input:	Directory path, for example: %OEM%\logfiles
Default:	No value, parameter optional
Access:	LEVEL1
Reaction:	RESET

Use parameter **MP_osTraceFileSize** to specify the file size for the log file of the HEIDENHAIN operating system. The file is saved in the subdirectory **OSTRACE** of the path specified under **MP_traceDir**. If the parameter is not part of the configuration, no log file is generated for the operating system. The **trace.act** file contains the current recordings, and the **trace.dmp** file has the data from the previous time the control was switched on.

MP_osTraceFileSize

	Maximum log file size for messages of the HEIDENHAIN operating system. Available from NCK software version: 597 110-04.
Format:	Numerical value
Input:	1 to 10 [MB] 0: Log inactive
Default:	No value, parameter optional (= function is inactive)
Access:	LEVEL1
Reaction:	RESET

Use parameter **MP_tcpTraceFileSize** to specify the file size for the log files of the network. Up to 10 files (**capture1** to **capture10**) are saved with recordings fn the network in the **TCPDUMP** subdirectory of the path specified under **MP_traceDir**.

MP_tcpTraceFileSize

	Maximum log file size for messages of the network Available from NCK software version: 597 110-04.
Format:	Numerical value
Input:	1 to 10 [MB] 0: Log inactive
Default:	No value, parameter optional (= function is inactive)
Access:	LEVEL1
Reaction:	RESET

Use parameter **MP_ncTraceFileSize** to specify the file size for the messages of the NC software. Up to 10 files (**capture1** to **capture10**) are saved with recordings of the NC software in the **NCDUMP** subdirectory of the path specified under **MP_traceDir**.

MP_ncTraceFileSize

	Maximum log file size for messages of the NC software. Available from NCK software version: 597 110-04.
Format:	Numerical value
Input:	1 to 10 [MB] 0: Log inactive
Default:	No value, parameter optional (= function is inactive)
Access:	LEVEL1
Reaction:	RESET

Use parameter **MP_krnlTraceFileSize** to specify the file size for the messages of the NC kernel. Up to 10 files (**capture1** to **capture10**) are saved with recordings on the kernel in the **KLOG** subdirectory of the path specified under MP_traceDir.

MP_krnlTraceFileSize

	Maximum log file size for messages of the NC kernel Available from NCK software version: 597 110-04.
Format:	Numerical value
Input:	1 to 10 [MB] 0: Log inactive
Default:	No value, parameter optional (= function is inactive)
Access:	LEVEL1
Reaction:	RESET

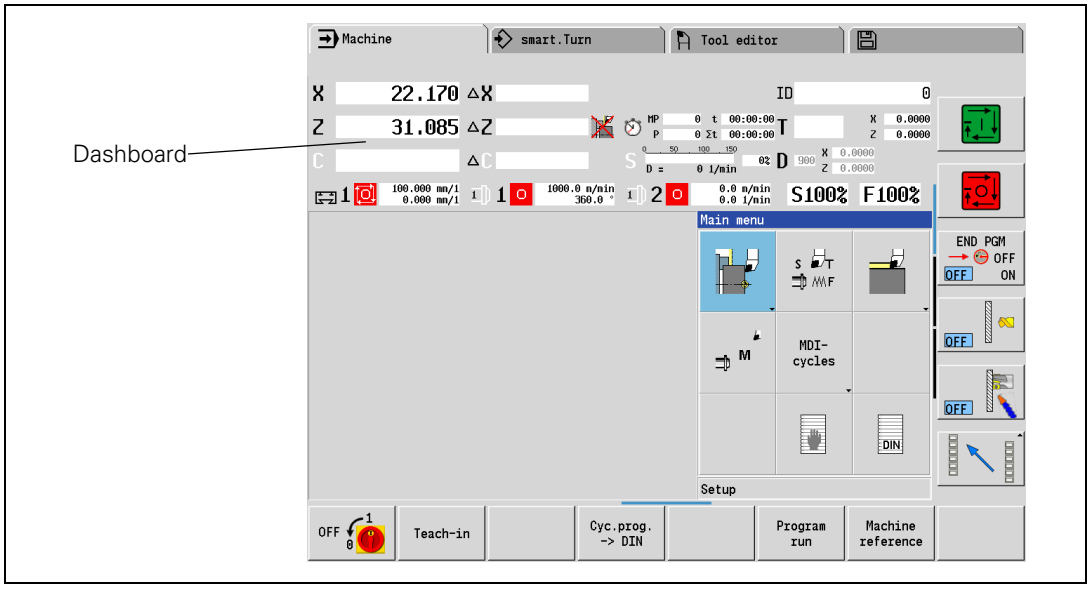
Suppressing the "Key non-functional" message

Settings in the configuration editor	MP number
System CfgConfigSettings **suppressUserMsg**	 106502

The parameter **MP_suppressUserMsg** is used to suppress the green "key non-functional" warning message. This warning is not recorded in the error log and is always issued when a key is pressed on the control that is not assigned to a function in the current operating situation.

MP_suppressUserMsg

	Do not display the **Key non-functional** error message Available from NCK software version: 597 110-04.
Format:	Selection menu
Selection:	**TRUE** The control does not output the "Key non-functional" error message. **FALSE** The control displays all error messages.
Default:	No value, parameter optional (= FALSE)
Access:	LEVEL3
Reaction:	NOTHING



The configurable machine display is referred to as the **dashboard** (see figure). The dashboard features 16 fields whose contents can be specified via machine parameter. You can create one or more dashboard for each operating mode of a machining channel. The corresponding dashboard is displayed when an operating mode is switched to. If more than one dashboard is defined for an operating mode, then there must be a user function in place for switching between dashboards.

The details of the displays, such as axis designations, number of decimal places, etc. are specified in the selection or configuration of the dashboard element. You specify in **MP_unitOfMeasure** (CfgUnitOfMeasure) whether the units are displayed in the inch or metric system.

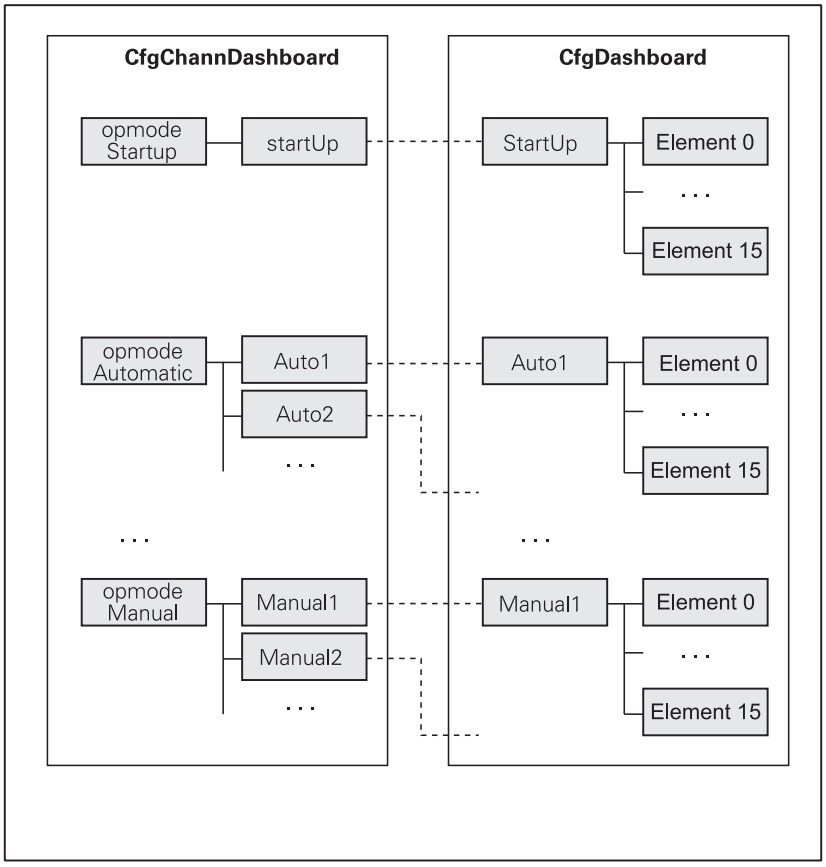
Sequence of the dashboard elements:

[0]	[1]	[2]	[3]
[4]	[5]	[6]	[7]
[8]	[9]	[10]	[11]
[12]	[13]	[14]	[15]

The dashboards are configured in the following steps:

- In the channel-dependent parameter object **CfgChannDashboard** you specify for each operating mode one or more key names for dashboard configurations.
- Configure the dashboard in the **CfgDashboard** parameter object. Under this key name you assign up to 16 dashboard elements for the layout of the dashboard. Youdo so by assigning key names for the dashboard elements.
- Specify the dashboard elements in the **CfgDashboardElemnt** parameter object. You assign an "image" to each key name of a dashboard element. Depending on the type of element, you gate the element to one or more axes or one or more machining channels.

The overview below shows the connections between the dashboard parameters.



Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannDashboard	
defaultDashboard	203401
opmodeStartup	203402
opmodeReference	203403
opmodeManual	203404
opmodeManualLarge	203408
opmodeMDI	203405
opmodeAutomatic	203406
opmodeAutomaticLarge	203409
opmodeSGTest	203407

The control displays the dashboard for the active operating mode. If no dashboard is defined for an operating mode, the default dashboard is displayed.

- Specify the **default dashboard** with the **MP_defaultDashboard** parameter
- If desired, assign one or more dashboards to an operating mode

Dashboard switchover

The soft key can be used to switch between the dashboards entered in **MP_opmodeManual** and **MP_opmodeAutomatic** depending on the operating mode. Up to 40 dashboards can be configured for the Manual and Automatic operating modes. If the last dashboard in the list is active, the first dashboard is jumped to. The prerequisite is that all dashboards entered in the list are also configured under **CfgDashboard**.

MP_defaultDashboard

	Default dashboard Available from NCK software version: 597 110-01.
Format:	Selection menu
Selection:	Key name of a dashboard configured under System/DisplaySettings/CfgDashboard
Default:	DB_DEFAULT
Access:	LEVEL3
Reaction:	RUN

MP_opmodeStartUp

Dashboard for start-up phase
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key name of a dashboard configured under System/DisplaySettings/CfgDashboard
Default: DB_STARTUP1
Access: LEVEL3
Reaction: RUN

MP_opmodeReference

Dashboard for the Reference operating mode
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key name of a dashboard configured under System/DisplaySettings/CfgDashboard
Default: DB_REFER1
Access: LEVEL3
Reaction: RUN

MP_opmodeManual

Dashboard for the Manual operating mode
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key name of a dashboard configured under System/DisplaySettings/CfgDashboard
Default: DB_MANUAL1
Access: LEVEL3
Reaction: RUN

MP_opmodeMDI

Dashboard for the MDI operating mode
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key name of a dashboard configured under System/DisplaySettings/CfgDashboard
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

MP_opmodeAutomatic

	Dashboard for the Automatic operating mode Available from NCK software version: 597 110-01.
Format:	Selection menu
Selection:	Key name of a dashboard configured under System/DisplaySettings/CfgDashboard
Default:	DB_AUTO1
Access:	LEVEL3
Reaction:	RUN

MP_opmodeSGTest

	Dashboard for the safety-related-test operating mode Available from NCK software version: 597 110-01.
Format:	Selection menu
Selection:	Key name of a dashboard configured under System/DisplaySettings/CfgDashboard
Default:	DB_SGTEST1
Access:	LEVEL3
Reaction:	RUN

Settings in the configuration editor	MP number
System	
DisplaySettings	
CfgDashboard	
[Key name of dashboard 1]	113200
[Key name of dashboard 2]	
.	
.	
.	
[Key name of dashboard n]	
CfgDashboardElemnt	
[Key name of the dashboard element]	
dashboardpicType	113101
attribut	113102
entityList	113103
CfgGlbDispSettings	
axesDisplayMode	604803

Configuring a dashboard

- In the **CfgDashboard** parameter object, specify the configuration for each defined dashboard. For each dashboard you enter the key names of up to 16 dashboard elements.

MP_[Keyname Dashboard]

Key name of the dashboard
Available from NCK software version: 597 110-01.

Format: Array [0...15]
Input: List of key names of dashboard elements configured under **CfgDashboardElemnt** in the sequence of arrangement.
Default: –
Access: LEVEL3
Reaction: RUN

The dashboard elements are aligned from left to right and from top to bottom. The index 0 defines the element at top left in the display.

Configuring a dashboard element

In the **CfgDashboardElemnt** parameter object you specify for each defined dashboard element the image, the attribute and the key names of the "associated" axis/axes or machining channel(s). "Associated" means that this display element shows values of this axis or machining channel (examples: position value of the axis, feed rate of the machining channel, speed of the spindle, etc.).

The following table shows the available dashboard elements.

- Select in **MP_dashboardpicType** the name of the dashboard element from the list.
- If desired, use **MP_attribut** to assign an attribute to the dashboard element.
- In **MP_entityList** you associate the axis/axes or the machining channel to be displayed in this dashboard element.

Configuring the axis display

MP_axesDisplayMode can be used to set the type of axis display. If the **Default** value is entered in **MP_axesDisplayMode**, the axis is displayed as defined in the attribute of the dashboard element. The **ActualValue**, **NominalValue**, **Lag** (following error) and **Distance** fields show the corresponding values.

MP_dashboardpicType

Image type of a dashboard element
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See table below
Default: ActualValue
Access: LEVEL3
Reaction: RUN

MP_attribut

Attribute of the dashboard element
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: See table below
Default: No value, parameter optional
Access: LEVEL3
Reaction: RUN

MP_entityList

Axis/Axes or channel associated with the dashboard element
Available from NCK software version: 597 110-01.

Format: Array [0...39]
Input: Key names from **MP_axisList** or **MP_channelList**
Default: –
Access: LEVEL3
Reaction: RUN

MP_axesDisplayMode

Configuring the type of axis display
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **Default**
Display defined as in the attribute
ActualValue
Display of the actual value
NominalValue
Display of the nominal value
Lag
Display of the following error
Distance
Display of the distance yet to go
Default: Default
Access: LEVEL3
Reaction: NOTHING

The following table lists the available dashboard elements and the associated attributes.

Image	Name	Display/Attribute
   Handwheel active  Clamping active    	ActualValue [DB_X1_POS, DB_Y1_POS, DB_Z1_POS, DB_W1_POS]	**Display:** Current position of an axis (actual-value display), handwheel and clamping are active ■ Letter designating the axis appears in black: Axis enabled by controller ■ Letter designating the axis appears in white: Axis disabled by controller **Attributes:** ■ Bit 0=0: Display the position of the tool tip ■ Bit 0=1: Display the IPO axis value—identified by "I" after the letter designating the axis. Tool lengths and datum shifts are not considered. ■ Bit 1=0: Display only if reference run has occurred ■ Bit 1=1: Display even if no reference run has occurred ■ Bit 2=1: Following error is displayed (code L) With the **General display settings -> Axis display** or **MP_axesDisplayMode** user parameter, the position display can be adapted, whereby the attribute settings are then without effect. **Selection:** ■ Default: No effect, attribute settings are without effect ■ ActualValue: Actual position (REF ACTL) with respect to the machine datum (code A) ■ NominalValue: Nominal position (REF NOML) with respect to the machine datum (code N) This value corresponds to the interpolator axis value (code I) if bit 0=1 ■ Lag: Following error (code L) ■ Distance: Distance to go (code D)

Image	Name	Display/Attribute
     	ActualCAxisValue [DB_C1]	**Display:** Current position of the C axis, handwheel and clamping are active ■ Letter designating the axis appears in black: Axis enabled by controller ■ Letter designating the axis appears in white: Axis disabled by controller **Attributes:** ■ Bit 0=0: Display the position ■ Bit 0=1: Display the IPO axis value—identified by "I" after the letter designating the axis. Tool lengths and datum shifts are not considered. ■ Bit 1=0: Display only if reference run has occurred ■ Bit 1=1: Display even if no reference run has occurred ■ Bit 2=1: Following error is displayed (code L) With the **General display settings -> Axis display** or **MP_axesDisplayMode** user parameter, the position display can be adapted, whereby the attribute settings are then without effect. **Selection:** ■ Default: No effect, attribute settings are without effect ■ ActualValue: Actual position (REF ACTL) with respect to the machine datum (code A) ■ NominalValue: Nominal position (REF NOML) with respect to the machine datum (code N) This value corresponds to the interpolator axis value (code I) if bit 0=1 ■ Lag: Following error (code L) ■ Distance: Distance to go (code D)

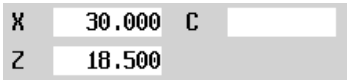
Image	Name	Display/Attribute
   Protection zone monitoring active  Protection zone monitoring inactive	ActualValueAnd DistanceToGo [DB_C1_DELTA,DB_X1_DELTA, DB_Z1_DELTA, DB_Y1_DELTA, DB_W1_DELTA]	**Display:** Distance-to-go in an axis and status of protection zone **Attributes:** ■ Bit 0=0: Display the programmed distance-to-go ■ Bit 0=1: Display the distance-to-go in the physical axis ■ Bit 1=0: Display only if reference run has occurred ■ Bit 1=1: Display even if no reference run has occurred ■ Bit 2=0: Do not display the status of the protection zone ■ Bit 2=1: Display the status of the protection zone (protection-zone monitoring is active/inactive)
  	AllAxes [DB_ALL_AXES1]	**Display:** Position values of up to four axes **Attributes:** ■ Bit 0=0: Display the position of the tool tip ■ Bit 0=1: Display IPO axis value—Letter designating the axis is highlighted in blue ■ Bit 1=0: Display only if reference run has occurred ■ Bit 1=1: Display even if no reference run has occurred The arrangement of the position values is determined by the sequence of the entityList entries. ■ [0]—Top left ■ [1]—Top right ■ [2]—Bottom left ■ [3]—Bottom right

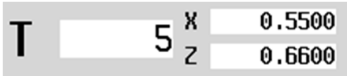
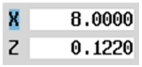
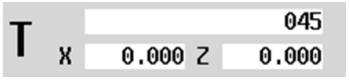
Image	Name	Display/Attribute
	ToolDisplayWith CompValues [DB_TOOL1]	**Display:** Tool pocket number and tool compensation values
 Code (blue) for driven tool		
 Code (blue) of mirrored tool holder		
 Code (blue) for displaying the special compensation DS		
	ToolDisplayWith Identification-Number [DB_TOOL_ID]	**Display:** Tool ID number and tool compensation values **Attributes:** ■ Bit 0=0: Display the tool ID number and tool compensation values ■ Bit 0=1: Increase the display size of the tool ID number—**No** display of tool compensation values! ■ Bit 1=1: Same as Bit 0=1, but with display of "ID" instead of "T"
 Code (blue) for driven tool		
 Code (blue) of mirrored tool holder		
 Increase the display size of the tool ID number		
 Attribute = 2: Increase the display size of the tool number with "ID" instead of "T"		

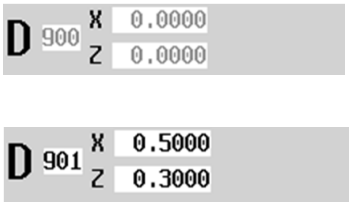
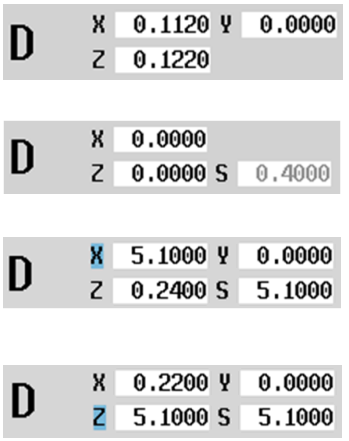
Image	Name	Display/Attribute
	AdditiveCompensation [DB_ADD_CORR1]	**Display:** Additive workpiece compensation for X, Z and Y If no workpiece compensation is active, D=900 and the compensation values=0.0000 are displayed in gray. If workpiece compensation is active, D=901 to 916 and the active compensation values are displayed in black.
	ActualToolData [DB_TOOL_CORR1]	**Display:** Tool compensation for X, Z, Y and, if required, special compensation S If a special compensation that is not active (G148 O=0) is assigned to a tool, the compensation value is displayed in gray. If special compensation S is activated with G148 O=1 (DX->DS) or O=2 (DZ->DS), the S value is displayed and the assigned axis letter is highlighted in blue.

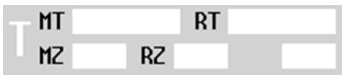
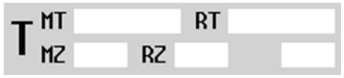
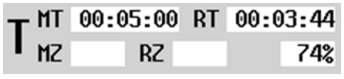
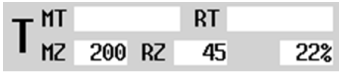
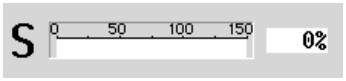
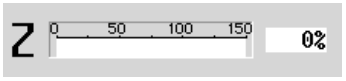
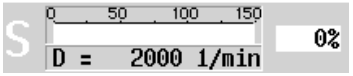
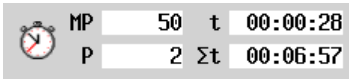
Image	Name	Display/Attribute
 Global tool life switch OFF (MP_lifeTime=Off): The letter T is displayed in white.  Global tool life switch ON (MP_lifeTime=On): The letter T is displayed in black (tool without tool life monitoring)  Tool life monitoring **according to time**  Tool life monitoring **according to part quantity**	ToolDisplay-WithToolLife-Information [DB_TOOL_LIFE1]	**Display:** Tool and tool life information ■ MT: Maximum tool life ■ RT: Current tool life ■ MZ: Maximum quantity ■ RZ: Current quantity ■ %: [current/maximum] · 100 The display is updated cyclically about every 30 seconds, as well as after tool change, program end or program cancellation.

Image	Name	Display/Attribute
 Attribute=0 (Bit 0=0)	SpindleAnd SpeedInformation [DB_S1, DB_S2]	**Display:** Spindle number, gear range, status (M3/M4/M5/M19/C axis), override, and rotational speed or stopping angle. ■ Spindle symbol displayed in black: Spindle enabling ■ Spindle symbol displayed in white: No spindle enabling **Attributes:** ■ Bit 0=0: Display the actual rotational speed + override ■ Bit 0=1: Display the actual rotational speed [1/min] and the programmed value ([m/min] at Vconst, or [1/min] at Nconst ■ Bit 1=1: Display the actual rotational speed, the programmed value and the override value ■ Bit 2=1: The current spindle position in [°] is displayed as actual value when the spindle is stopped (M5) If the value **TRUE** is entered in **MP_plcSpindleSelect**, the number of the selected spindle is given a blue background. For a C-axis with external spindle drive (e.g. S4 drives the main spindle S1 through a transmission), the spindle speed of S4 can now be displayed in the S1-element in the C-axis mode. For this purpose the spindle (S1), which is also the spindle to be displayed, must be assigned in the external C axis in **MP_relatedWpSpindle** under CfgCAxisProperties [C_n]. In addition, the physical axis of the external drive (S4) must be entered under CfgProgAxis [C_n] in **MP_relatedAxis**. Then the bit "AktAchsStatus2::CHANNEL_AXIS" (spindle in C-axis mode) is monitored in the spindle interface. In this case, the axis velocity of the external drive (S4) is shown in the spindle display (S1) as long as the C-axis mode is active.
 Attribute=1 (Bit 0=1)		
 Attribute=2 (Bit 1=1)		
 Attribute=4 (Bit 2=1)		
 Attribute=5 (Bit 0=1, Bit 2=1)		
 Display for positioning of spindle: Current spindle position [°]		
 Display of spindle states		
 Selected spindle on blue background if MP_plcSpindleSelect=TRUE		

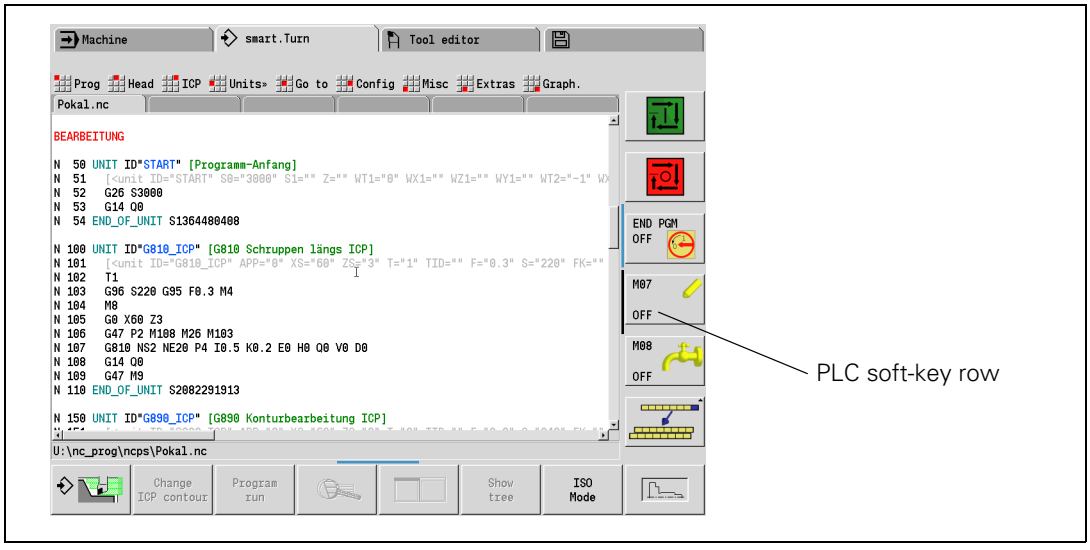
Image	Name	Display/Attribute
 Attribute=0 (Bit 0=0)  Attribute=1 (Bit 0=1)  Attribute=2 (Bit 1=1)  Cycle OFF	SlideAndFeed RateInformation [DB_CH1_ STATE]	**Display:** Slide number, slide status (cycle ON/OFF), feed rate and override. If the slide is idle, the programmed feed rate is displayed in gray. **Attributes:** ■ Bit 0=0: Display the actual feed rate and the override value ■ Bit 0=1: Display the programmed feed rate and the actual feed rate ■ Bit 1=1: Display the programmed feed rate, the actual feed rate and the override value
 Attribute=0 (Bit 0=0)  Attribute=2 (Bit 1=1)  Attribute=4 (Bit 2=1)	ChannelDisplay [DB_OVERRIDE]	**Display:** Override for F, S and R **Attributes:** ■ Bit 0=0: S at left, F or R at right ■ Bit 0=1: F or R at left, S at right ■ Bit 1=0: Display of feed override F ■ Bit 1=1: Display of rapid traverse override R ■ Bit 2=0: Display of feed override F ■ Bit 2=1: Display of rapid traverse override R + feed override F

Image	Name	Display/Attribute
 Utilization display of the spindle  Utilization display of an axis:  Attribute=2 (Bit 1=1) Utilization display showing the rotational speed limiting (only for spindles)	LoadDisplay [DB_LD_C1, DB_LD_S1, DB_LD_X1, DB_LD_Z1]	**Display:** Utilization of the drive of an axis (spindle, X, Z and C axes) **Attributes:** ■ Bit 0=0: Digital drive, display data are supplied by the CC ■ Bit 0=1: Analog drive, display data are supplied by the user PLC ■ Bit 1=0: Without display of rotational speed limiting ■ Bit 1=1: With display of the rotational speed limiting (only for spindles)
 Display of unit quantities and time per unit  Display of unit quantities	QuantityInformation AndTimePerUnit [DB_WPCT1] QuantityInformation [DB_WPC1]	**Display:** Unit quantities and time per unit MP: Default unit quantity P: Number of finished parts t: Time in current program Σt: Total time **Display:** Unit quantity MP: Default unit quantity P: Number of finished parts The quantity is incremented after each M30, M99 or M18 programmed counter pulse.
	BlankField [DB_EMPTY]	**Display:** Empty box

7.3 PLC Soft Keys

In the vertical soft-key row, you can display your own soft keys through the PLC in all operating modes. Use the PLCdesignNT TNC soft key editor to create the soft keys. The MenuDesign software enables you to define the menu structure of the vertical soft-key row. PLCdesignNT integrates the project file of MenuDesign in your PLC project. For more detailed information, please refer to the respective online help of the programs.

When a PLC soft key is pressed the NC enters the soft-key number in the PLC operand **NP_GenSoftkeyVert**. On the rising edge of the keystroke it enters the soft-key number; on the falling edge it enters –1. The PLC can enter –1 itself after recognizing the soft-key number.



7.4 Switching the Control On/Off

7.4.1 Powering up the control

Start sequence

The control is started up in several phases. The following is a summary of the process. The parts at which the OEM can directly influence the control start-up by making entries or macros are described comprehensively in the following.

Sequence of the control start-up:

- **Step 1: BIOS loading process**
  The BIOS saved on an EPROM of the MC main computer is loaded. BIOS means Basic Input Output System. The BIOS also tests the installed main memory and the individual peripheral components.
- **Step 2: HeROS boot process**
  After the BIOS is loaded the control starts the HEIDENHAIN HeROS operating system (HEIDENHAIN Real-time Operating System).
- **Step 3: Starting the NC software**
  HeROS starts the NC software. The start-up screen appears on the control monitor as well as a dialog box showing the number of the currently loaded process. During start-up the control reads the config data from the *.cfg files. If the configuration contains errors or, for example, new and therefore yet uninitialized parameters from a software update, the boot process is interrupted. The code-number dialog is shown and you have to enter the MP code number **95148** in order to open the config editor. Here all new or changed parameters are marked by a clearly recognizable red exclamation point. Use the **SAVE** soft key to save the changed configuration and exit the configuration editor. The the control continues the starting sequence of the NC software.
- **Step 4: Acknowledge the power interruption**
  If so configured with **MP_powerInterruptMsg**, see page 949, the control waits until you confirm the **Power interrupted** message by pressing the CE key. If the machine configuration is faulty, the power interruption can be acknowledged only after the configuration is corrected.
- **Step 5: Compile the PLC program**
  After the CE key is pressed, the control compiles the PLC program. The message **Compiling the PLC program** is displayed. If the PLC program cannot be successfully compiled, you can see the error in the status window of the PLC compiler in the editor of the PLC Programming mode of operation.
- **Step 6: Configuring the CC controller unit (only digital control)**
  The interpolator now configures the CC controller unit and starts the DSPs. If errors occur, they are shown in the error window (ERR key).
- **Step 7: Testing the EMERGENCY STOP**
  If the CC controller unit reports its readiness, the control starts the EMERGENCY STOP test. For more information on the process, see "Emergency stop monitoring" on page 680.
- **Step 8: Running the start-up cycle**
  The start-up cycle is run in total three times—once for the Program Run modes, once for the **Test Run** mode, and once for the **Programming** mode. During execution the control displays the message **Control is being initialized**. After the control start-up, the start-up cycle establishes a defined control condition.
- **Step 9: Traversing the reference marks (if configured)**
  Finally, the control shows the message **Traverse the reference points** to prompt you the move the axes over the reference marks. Now the control start-up process is complete. The **Traverse Reference Marks** mode cannot be exited until all axes have been successfully referenced.

Power interrupted message

Settings in the configuration editor	MP number
System DisplaySettings CfgStartupData **powerInterruptMsg**	 101501

In **MP_powerInterruptMsg,** you define the behavior during control start-up. You can choose between the following start-up procedures.

- The **`Power interrupted`** message is displayed during control start-up. Run-up is only continued after the message has been acknowledged.
- Startup is not interrupted.
  `Power interrupted` does not appear.

MP_powerInterruptMsg

Acknowledge the **`Power interrupted`** message
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **TRUE**
Run-up is only continued after the message has been acknowledged.
FALSE
The **`Power interrupted`** message does not appear.

Default: FALSE

Access: LEVEL3

Reaction: NOTHING

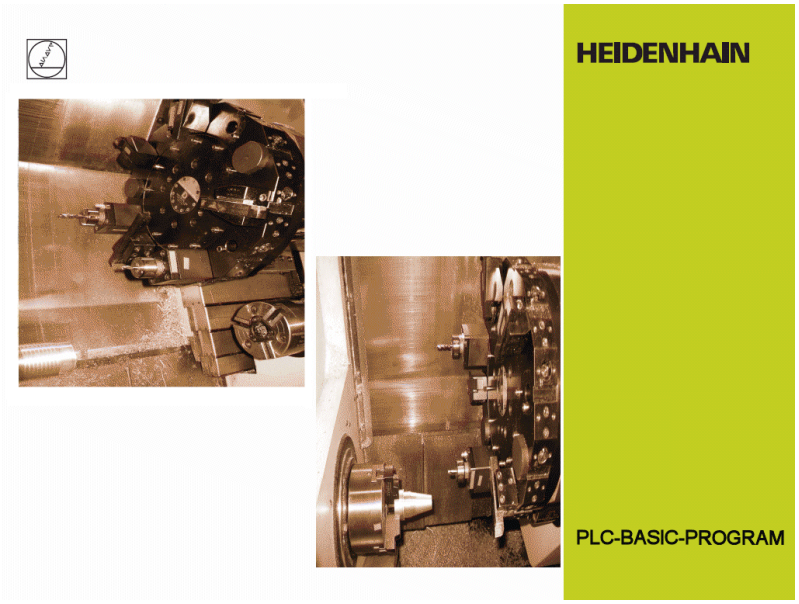
PLC operand / Description	Model
NN_GenCycleAfterPowerOn 1. PLC scan after power on	M
NN_GenCycleAfterPlcStop 1st PLC scan after PLC interruption	M
NN_GenCycleAfterReConfig 1st PLC scan after changing of the configuration data	M
NN_GenNcInitialized Control is being initialized (after start-up cycles)	M

Customer-specific logo

While the control is starting, a customer-specific company logo can be displayed instead of the HEIDENHAIN logo. The logo must be a bitmap file (*.BMP) with a picture size of 1024 x 768 pixels.

Path and file name of bitmap file:

- PLC:\resource\startup_1024x768.bmp



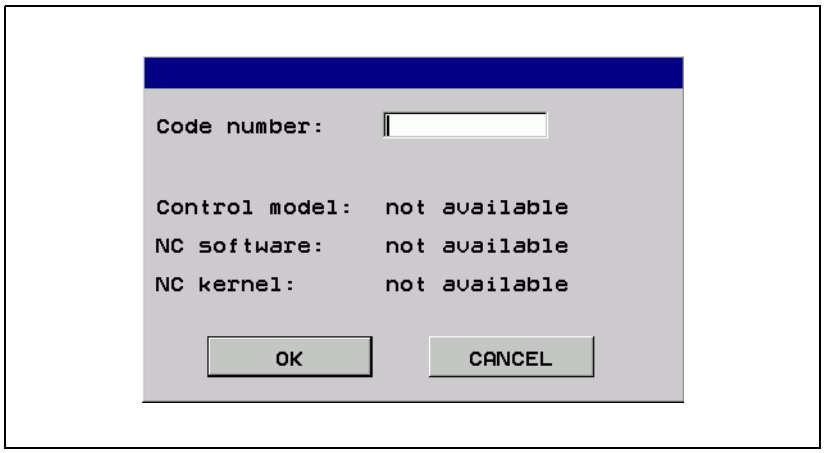
Starting the control with empty PLC partition

During startup, machine parameters are automatically transferred from the PLC partition. If the PLC partition does not contain any data, however, (e.g. loss of data, partition inadvertently erased or initial operation), the control cannot be started.

If this occurs, you can access the control via TNCremoNT. This enables you to restore a PLC partition that has been backed up before.

Example:

During start-up, the control detects that the required machine parameter files are not available on the PLC partition. The start-up process is interrupted. The following dialog box appears:



(The machine parameters for configuring the conversational language are also stored on the PLC partition. Since these parameters are not available at present, English is set as the default language.)

- Use TNCremoNT to establish a connection to the control (the last valid Ethernet configuration is effective)
- Copy the data you saved before, with TNCbackup for example, back to the PLC partition
- Shut down the control by pressing the **OFF** soft key.
- Restart the control.

The system should work properly again.

Control startup with faulty configuration

If configuration errors occur during control start-up, the **Error during start-up** message appears instead of **Power interrupted**.

Also, the error messages for the incorrect configuration data, which are triggered by the applications, are displayed individually.

Settings in the configuration editor	MP number
System	
DisplaySettings	
CfgShutDown	
shutdownOnConfig	101601
shutdownOnError	101602
shutdownOnUser	101603
shutdownOnOem	101604
maxTermTime	101605
powerOffDevice	101608
powerOffSlot	101609
powerOffPort	101606
powerOffDelay	101607

The control must be shut down before it can be switched off. During shutdown the ready signal of the servo drives is removed and the memory card (or hard disk) is put into sleep mode. The shutdown can be delayed.

There are various causes for shutdown. In the parameter object **CfgShutDown,** you define the behavior when the control is shut down, depending on the cause. A distinction is made between the following causes:

- **shutdownOnConfig**:
  Configuration data that cause a reset were changed.
- **shutdownOnError**:
  A severe error occurred.
- **shutdownOnUser**:
  The machine operator terminates control operation (by soft key).
- **shutdownOnOem**:
  The PLC program terminates control operation with Module 9279 or 9189.

MP_maxTermTime specifies the maximum permissible time for all processes to be concluded. With the parameter you can define a time window after which the control shut down must have been concluded.

After shutting down the control, you can set a PLC output (to switch off the machine, for example). Systems with HSCI interface require other settings than systems without HSCI interface.

Systems without HSCI interface:

- Use the **POWEROFF** setting for shutdown, or use Module 9279 – mode 2 to shut down the control.
- **MP_powerOffPort** defines whether a PLC output is to be set after shutdown.
  Define a PLC output (O0 to O31) in **MP_powerOffPort**.
- The setting of the PLC output can be delayed. If required, enter in **MP_powerOffDelay** the time by which the setting of the PLC output is to be delayed.

Systems with HSCI interface:

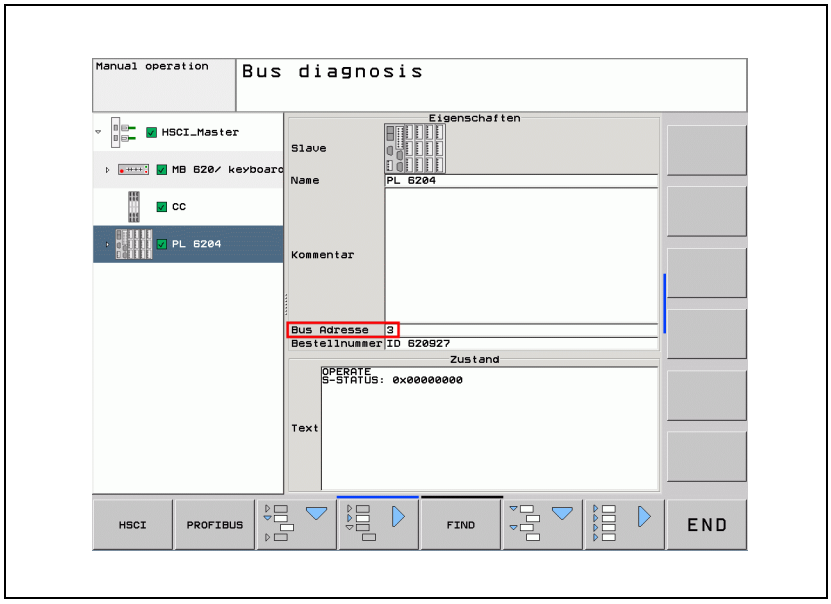
Note

The following PLC outputs can be set automatically after shutdown:

- Freely available PLC outputs at X9 of a PL 62xx (system PL)
- PLC outputs at X6 of a UEC 11x

The output terminals that can be used are listed in the tables on the following pages.

- Use the **POWEROFF** setting for shutdown, or use Module 9279 – mode 2 to shut down the control.
- Set the parameter **MP_powerOffDevice**:
  Every external PL in the HSCI system has a unique bus address. To define the PLC output to be set, you must specify the bus address of the affected system PL or the UEC 11x in parameter **MP_powerOffDevice**. The bus address can be determined with the BUS DIAGNOSIS, for example:



- Set the parameter **MP_powerOffSlot** = 0.
- Use **MP_powerOffPort** to define the desired output terminal.
  Use the tables on the following pages to determine the value for parameter **MP_powerOffPort**.

PL 62xx without Functional Safety, connection X9:

Terminal of the PLC output	Setting in MP_powerOffPort
X9.3b	3
X9.4a	0
X9.4b	4
X9.5a	1
X9.5b	5
X9.6a	2
X9.6b	6

UEC 11x without Functional Safety, connection X6:

Terminal of the PLC output	Setting in MP_powerOffPort
X6.1a	4
X6.1b	0
X6.2a	5
X6.2b	1
X6.3a	6
X6.3b	2
X6.4a	7
X6.4b	3
X6.5a	12
X6.5b	8
X6.6a	13
X6.6b	9
X6.7a	14
X6.7b	10
X6.8a	15
X6.8b	11
X6.9a	20
X6.9b	16
X6.10a	21
X6.10b	17
X6.11a	22
X6.11b	18
X6.12b	19

- The setting of the PLC output can be delayed. If required, enter in **MP_powerOffDelay** the time by which the setting of the PLC output is to be delayed.

Machine parameters

MP_shutdownOnConfig

Behavior when RESET configuration data is changed
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **RESTART**
The control is shut down and then restarted.
TERMINATE
The control is shut down, but the operating system remains active.
SHUTDOWN
The control and the operating system are shut down.
POWEROFF
The control and the operating system are shut down. If a PLC output is entered in **MP_powerOffPort**, it will be set after shutdown.

Default: RESTART

Access: LEVEL3

Reaction: NOTHING

MP_shutdownOnError

Behavior when RESET errors are acknowledged
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **RESTART**
The control is shut down and then restarted.
TERMINATE
The control is shut down, but the operating system remains active.
SHUTDOWN
The control and the operating system are shut down.
POWEROFF
The control and the operating system are shut down. If a PLC output is entered in **MP_powerOffPort**, it will be set after shutdown.

Default: RESTART

Access: LEVEL3

Reaction: NOTHING

MP_shutdownOnUser

Behavior during switch-off by soft key
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **RESTART**
The control is shut down and then restarted.
TERMINATE
The control is shut down, but the operating system remains active.
SHUTDOWN
The control and the operating system are shut down.
POWEROFF
The control and the operating system are shut down. If a PLC output is entered in **MP_powerOffPort**, it will be set after shutdown.

Default: TERMINATE

Access: LEVEL3

Reaction: NOTHING

MP_shutdownOnOem

Behavior when PLC module 9279 is called
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **RESTART**
The control is shut down and then restarted.
TERMINATE
The control is shut down, but the operating system remains active.
SHUTDOWN
The control and the operating system are shut down.
POWEROFF
The control and the operating system are shut down. If a PLC output is entered in **MP_powerOffPort**, it will be set after shutdown.

Default: TERMINATE

Access: LEVEL3

Reaction: NOTHING

After shutdown has been initiated, the control waits for the time defined in **MP_maxTermTime** before starting the shutdown.

MP_maxTermTime

Delay time until control is shut down
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to 1000 [s]
Default: 0
Access: LEVEL3
Reaction: NOTHING

Both optional parameters **MP_powerOffDevice** and **MP_powerOffSlot** are relevant only for HSCI systems (e.g. TNC 620). No values are to be entered for controls without HSCI.

For an HSCI system, enter the bus address of the HSCI PL in the parameter **MP_powerOffDevice**, e.g. the bus address of a PL 62xx or UEC 11x at which the control is to set a PLC output after shutting down.

MP_powerOffDevice

Bus address of the HSCI device
Available from NCK software version: 597 110-04.

Format: Numerical value
Input: 0 to 20
Default: No value, optional parameter
Access: LEVEL1
Reaction: NOTHING

For an HSCI system, enter in parameter **MP_powerOffSlot** the number of a slot of the HSCI device defined under **MP_powerOffDevice** at which a PLC output is to be set after shutdown. The system module of a PL 62xx or a UEC 11x has the number 0.

MP_powerOffSlot

Number of the slot on the HSCI device (PL or UEC)
Available from NCK software version: 597 110-04.

Format: Numerical value
Input: 0 to 8
Default: No value, parameter optional
Access: LEVEL1
Reaction: NOTHING

The entry in **MP_powerOffPort** has the following meaning:

- The corresponding PLC output is set if the requirements described above are fulfilled (shutdown with **POWEROFF** setting or shutdown with Module 9279 – mode 2).
- No entry: No PLC output is set.

MP_powerOffPort

	PLC output to be set after shutdown Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0 to 31: Corresponds to PLC outputs I0 to I31 for systems without HSCI For systems with HSCI, please see the device-specific table. No entry: Do not set PLC output.
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	NOTHING

After shutdown, the control waits for the time defined in **MP_powerOffDelay** before setting the PLC output.

MP_powerOffDelay

	Delay time until PLC output is set Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0 to 1000 [s]
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	NOTHING

Module 9189 Shut down the control

Module 9189 shuts down the control. The PLC is not executable after shutdown. The message windows, which appear during shutdown via soft key, do not appear.

Call:
CM 9189

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Control was shut down
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	20	Module was not called in a spawn job or submit job

Module 9279 Shut down control (configurable)

The module terminates control operation. The behavior during shutdown of the control is defined in a transfer parameter.

Constraints:

- After the module has been called, the file system updates the data on the memory card (or hard disk) and closes all files.
- The module cannot be called in the cyclic PLC program since all accesses to the hard disk by the PLC must be implemented in a submit or spawn job.
- The PLC is not executable after shutdown.
- If mode 1 is active, a control reset is triggered immediately after shutdown.
- The module call does not result in any outputs on the screen.

Call:

PS B/W/D/K <Mode>
0: Shut down the control
1: Shut down and restart the control
2: Shut down the control; then set the PLC output from MP_powerOffPort (if defined).

CM 9279

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Control reset was carried out
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	20	Module was not called in a spawn job or submit job

7.5 Keystroke Simulation

The MANUALplus 620 features two control panels:

- Control keyboard
- MB 620T machine operating panel (from HEIDENHAIN) or a machine operating panel from the OEM

The machine operating panel is connected over an HSCI cable to the MC 6110T.

The key code of the control keyboard is evaluated directly by the NC.

The machine operating panel has PLC inputs and outputs which you must evaluate in the PLC.

7.5.1 Control keyboard

The key code is displayed in **NP_GenKeyCode** while a key of the control keyboard is being pressed. See "Codes for keystroke simulation" on page 964.

PLC operand / Description	Type
NP_GenKeyCode Code of the depressed key	D

Note

The code of the key actually pressed is always saved in **NP_GenKeyCode**. Characters entered using the "cell-phone-style keypad" are not stored in this PLC operand. Instead, the code of the numeric key you pressed is displayed *n* times.

The following modules can influence keys and soft keys:

- Module 9180: Simulation of NC keys
- Module 9181: Disabling of individual keys
- Module 9182: Re-enabling of individual keys
- Module 9183: Disabling groups of NC keys
- Module 9184: Re-enabling of groups of NC keys

Module 9180 Keystroke simulation
The module simulates NC keys and soft keys. You transfer the code of the desired key.

If you transfer the code value zero, the number of occupied elements in the keystroke queue is returned. In this case there is no keystroke simulation.

Call:
PS B/W/D/K <Key code>
CM 9180
PL B/W/D <Number of occupied elements / error status>
0: Key code was transferred, key queue is empty
1 to 16 : Key code was not yet simulated, max. 16 entries in the keystroke queue are possible
–1: For error see NN_GenApiModuleErrorCode

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	NC key was simulated
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Transferred parameter > maximum value
	2	Transferred parameter invalid
	22	Keystroke queue overflow

Module 9181 Disable NC key by PLC
The module disables individual NC keys.

Call:
PS B/W/D/K <Key code>
CM 9181
PL B/W/D <Error status>
0: NC key disabled
–1: For error see NN_GenApiModuleErrorCode

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	NC key was disabled
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Transferred parameter > maximum value
	2	Transferred parameter invalid

Module 9182 Re-enable NC key by PLC
The module cancels the effect of Module 9181.

Call:
PS B/W/D/K <Key code>
CM 9182
PL B/W/D <Error status>
0: NC key enabled
–1: For error see NN_GenApiModuleErrorCode

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Disabling was canceled
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Transferred parameter > maximum value
	2	Transferred parameter invalid

Module 9183 Disable NC key groups by PLC
The module disables the specified key group. The table at the end of this section contains the assignment of the keys to the key groups.

The key-group codes are:

- 0: All keys
- 1: ASCII
- 2: Soft keys, Page Up/Down
- 3: Cursor, ENT, NOENT, DEL, END, GOTO
- 4: Numbers, algebraic signs, decimal point, actual position capture
- 5: Operating modes
- 6: Block opening keys

Call:
PS B/W/D/K <Key-group code>
CM 9183
PL B/W/D <Error status>
0: Group of NC keys disabled
–1: Transferred parameter > maximum value

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	The group of NC keys was disabled
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Transferred parameter invalid

Module 9184 Re-enable locked NC key groups by PLC

The module cancels the effect of Module 9183 for the given key group. The table at the end of this section contains the assignment of the keys to the key groups.

The key-group codes are:

- 0: All keys
- 1: ASCII
- 2: Soft keys, Page Up/Down
- 3: Cursor, ENT, NOENT, DEL, END, GOTO
- 4: Numbers, algebraic signs, decimal point, actual position capture
- 5: Operating modes
- 6: Block opening keys

Call:

PS	B/W/D/K	<Key-group code>
CM	9184	
PL	B/W/D	<Error status> 0: Group of NC keys enabled –1: Transferred parameter > maximum value

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Disabling was canceled
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Transferred parameter invalid

Codes for keystroke simulation

The following table takes into account all keys used on the MANUALplus 620. Please note, however, that not all the keys listed in the table must necessarily be available on your keyboard.

Code	Key	Group
$08	BACKSPACE	ASCII
$0A	ENT (Return)	ASCII
$1B	ESC	ASCII
$20	SPACE	ASCII
$21	!	ASCII
$22	‚Äû	ASCII
$23	#	ASCII
$24	$	ASCII
$25	%	ASCII
$26	&	ASCII
$28	(	ASCII
$29	)	ASCII
$2A	*	ASCII
$2B	+	ASCII
$2C	,	ASCII
$2D	–	ASCII
$2E	. (ASCII DOT)	ASCII
$2F	/	ASCII
$30	0	Numbers
$31	1	Numbers
$32	2	Numbers
$33	3	Numbers
$34	4	Numbers
$35	5	Numbers
$36	6	Numbers
$37	7	Numbers
$38	8	Numbers
$39	9	Numbers
$3A	:	ASCII
$3B	;	ASCII
$3C	<	ASCII
$3D	=	ASCII
$3E	>	ASCII
$3F	?	ASCII
$41	A	ASCII
$42	B	ASCII
$43	C	ASCII
$44	D	ASCII
$45	E	ASCII

Code	Key	Group
$160	Soft key 0	Vertical soft key
$161	Soft key 1	Vertical soft key
$162	Soft key 2	Vertical soft key
$163	Soft key 3	Vertical soft key
$164	Soft key 4	Vertical soft key
$165	Soft key 5	Vertical soft key
$17D	FNEXT (vertical)	Vertical soft key
$180	Soft key 0	Horiz. soft key
$181	Soft key 1	Horiz. soft key
$182	Soft key 2	Horiz. soft key
$183	Soft key 3	Horiz. soft key
$184	Soft key 4	Horiz. soft key
$185	Soft key 5	Horiz. soft key
$186	Soft key 6	Horiz. soft key
$187	Soft key 7	Horiz. soft key
$19C	FBACK	Horiz. soft key
$19D	FNEXT (horizontal)	Horiz. soft key
$19E	FNEXT-UP	Horiz. soft key
$1A0	C-UP	Cursor
$1A1	C-DOWN	Cursor
$1A2	C-LEFT	Cursor
$1A3	C-RIGHT	Cursor
$1A8	ENTER	Cursor
$1A9	NO-ENTER	Cursor
$1AB	DEL	Cursor
$1AC	END BLOCK	Cursor
$1AD	GOTO	Cursor
$1AE	CE	
$1BC	+/–	Numbers
$1BD	.	Numbers
$1C0	Machine	Operating mode
$1C4	DINplus	Operating mode
$1C5	Tool editor	Operating mode
$1C7	Machine para.	Operating mode
$1E9	Err	
$1EF	Calc	
Empty key = no code		
$1EE	I (Info)	
$1EB	Circle with three arrows	

Code	Key	Group
$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

Code	Key	Group
$1EC	Circle with two arrows	
$103	Smart-Up	
$104	Smart-Down	
$105	Smart	
$127	INS	ASCII
$128	Home	ASCII
$129	PgUp	ASCII
$12A	PgDn	ASCII
$133	END	ASCII
$134	DEL	ASCII

7.5.2 Machine operating panel

The machine operating panel provides PLC inputs and PLC outputs for evaluating its keys. Use the IOconfig PC software to link the inputs and outputs of the machine operating panel with the PLC operands.

PLC operand / Description	Type
PP_AxTraversePos Manual traverse in positive direction 0: Do not move axis 1: Move axis	M
PP_AxTraverseNeg Manual traverse in negative direction 0: Do not move axis 1: Move axis	M
PP_ChnRapidTraverseKey Rapid-traverse key 0: Rapid-traverse key not pressed 1: Rapid-traverse key pressed	M

7.6 Electronic Handwheel

The control supports:

- handwheels that are connected to the serial handwheel input X23 of the MC.
- handwheels with position encoders, which are connected to the position encoder inputs of the MC.

For information about mounting and electrical connection, see "Handwheel Input" on page 206 or see "Handwheels" on page 226.

You can connect the following handwheels to the handwheel input (X23) of your control:

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

For information about mounting and electrical connection, see "HRA 110 handwheel adapter" on page 210.

As an alternative, or additionally, you can connect the HR 180 handwheel to the position encoder inputs of the MC.

- Define the type of handwheel, or the connection of the handwheel to the control in **MP_type.** If the handwheel is connected to a position encoder input, the handwheel connection is described in the configuration object **CfgAxisHandwheel**.
- If an HRA 110 is used, the axis-specific description of the counting direction and the traverse per handwheel revolution is in the **CfgAxisHandwheel** configuration object.

7.6.1 Serial handwheel

Settings in the configuration editor	MP number
System	
CfgHandwheel	
type	100601
initValues	100602
incrPerRevol	100603
rasterPerRevol	100604
countDir	100605
sensitivity	100606
speedFactor	100607
feedFactor	100608
crossShortSafety	100609

Type of handwheel

MP_type

Handwheel
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
NONE
No handwheel connected.
HRNAX
HRA 110 handwheel adapter connected to X23
HR410
HR 410 connected to X23
HR332
HR 332 connected to X23
HR330
HR 330 connected to X23
HR130
HR 130 connected to X23

Default: NONE
Access: LEVEL3
Reaction: RUN

Handwheel initialization

You enter initialization values for serial handwheels in **MP_initValues.** During startup, the control transfers the initialization values to the handwheel.

Information about the initialization values:

- HR 410: see "HR 410 portable handwheel" on page 981
- HR 130: No initialization values are necessary for this model
- HRA 110 with HR 150: See "HR 150 panel-mounted handwheels with HRA 110 handwheel adapter" on page 983

MP_initValues

Initialization values for handwheel
Available from NCK software version: 597 110-01.

Format: Array [7]
Input: 0 to 255
Default: 0
Access: LEVEL3
Reaction: RUN

Evaluation of handwheel pulses

Define the evaluation of the handwheel pulses in the following parameters:

- **MP_incrPerRevol**: Number of increments per handwheel revolution
- **MP_rasterPerRevol**: Detent steps per handwheel revolution (only for handwheels with detent)
- **MP_countDir:** Direction for handwheel input

MP_incrPerRevol

Increments per handwheel revolution
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to max. value
0: Standard value for HEIDENHAIN handwheel (20 000 increments)
Default: 0
Access: LEVEL3
Reaction: RUN

MP_rasterPerRevol

Detent steps per handwheel revolution
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to max. value
0: Handwheel without detent
Default: 0
Access: LEVEL3
Reaction: RUN

MP_countDir

Counting direction for handwheel
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **Positive**
Positive counting direction
Negative
Negative counting direction
Default: Positive
Access: LEVEL3
Reaction: RUN

Threshold sensitivity

Shock or vibrations can cause a slight motion at the handwheel and produce an unintentional axis movement.

- In **MP_sensitivity,** enter a threshold sensitivity, in order to avoid unintentional movements.

MP_sensitivity

Sensitivity for electronic handwheel
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to 10 000 [pulses]
Default: 10 [pulses]
Access: LEVEL3
Reaction: RUN

Handwheel with axis-direction keys

For handwheels with axis-direction keys (e.g. HR 410), you define three feed rates in **MP_feedFactor**. These entries refer to the feed rate entered in **`CfgFeedLimits/manualFeed.`** In the parameter **MP_speedFactor** you define three transmission ratios for the distance traversed per handwheel revolution configured in **MP_distPerRevol.**

MP_speedFactor

Handwheel transmission ratio
Available from NCK software version: 597 110-02.

Format: Array [0...2]
Input: [0] = First transmission ratio in [%]
[2] = Second transmission ratio in [%]
[3] = Third transmission ratio in [%]
Distance per handwheel revolution:
Transmission ratio · **`CfgAxisHandwheel/distPerRevol`**
Default: [0] = 1
[1] = 10
[2] = 100
Access: LEVEL3
Reaction: RUN

MP_feedFactor

Manual feed rates in the Electronic Handwheel mode
Available from NCK software version: 597 110-01.

Format: Array [0...2]
Input: 0 to 100 [%]
[0] = First feed rate in [%]
[1] = Second feed rate in [%]
[2] = Third feed rate in [%]
Default: 1, 10 and 100 [%]
Access: LEVEL3
Reaction: RUN

Note

MP_feedFactor is only valid in the **`Electronic Handwheel`** operating mode, but then for all manual axis-direction keys including the manual axis-direction keys of the operating panel.

In **MP_crossShortSafety** you define whether a short-circuit test is to be performed on the permissive buttons.

MP_crossShortSafety

Short-circuit-proofed handwheel
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **On**
Short-circuit test on
Off
Short-circuit test off
Default: Off
Access: LEVEL3
Reaction: RUN

Locking the handwheel

Disable handwheel pulses for specific axes with **PP_AxHandwheelLocked.** Disable the handwheel pulses for all axes with **PP_GenHandwheelLocked.** If **PP_GenHandwheelLocked** is reset, axis-specific disabling of **PP_AxHandwheelLocked** is effective again.

PLC operand / Description	Model
PP_GenHandwheelLocked Disable handwheel motion 0: Enable handwheel movements for all axes 1: Disable handwheel movements for all axes	M
PP_AxHandwheelLocked Disable handwheel motion for specific axes 0: Enable handwheel movements for this axis 1: Disable handwheel movements for this axis	M

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name for the parameter set]	
CfgAxisHandwheel	
input	400201
hsciCcIndex	400210
countDir	400202
distPerRevol	400203
incrPerRevol	400204
rasterPerRevol	400205
encoderSignal	400206
encoderFreq	400207
encoderResistor	400208
decToSoftLimit	400209

Handwheel connection

Note

You define the handwheel parameters within the parameter set of an axis. This permanently assigns the handwheel to **this axis**. You can assign the handwheel to a different axis, for example, by changing to a different parameter set.

Define the handwheel connection in the following parameters:

- **MP_input:** Assignment of handwheel to position encoder input
- **MP_hsciCcIndex**: Index of the CC controller unit to which the handwheel is connected
- **MP_EncoderSignal:** 1-V_{PP} or 11-μA_{PP} signal
- **MP_EncoderFreq:** Maximum input frequency
- **MP_EncoderResistor:** Terminating resistor

MP_input

Handwheel connector at encoder input
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **none**
No handwheel connected to position encoder input
X01 to X06
Handwheel connected to X01 to X06 of the MC
X35 to X38
Handwheel connected to X35 to X38 of the MC
X201 to X210
Handwheel connected to X201 to X210 of the CC

Default: None

Access: LEVEL3

Reaction: NOTHING

MP_encoderSignal

Signal amplitude at position encoder input for handwheel
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **1 Vpp**
Input signal of encoder is 1 Vpp signal.
11 µA
Input signal of encoder is 11 µA signal.
This setting is not supported by the CC 61xx controller unit! If the value is set nevertheless, the control outputs the **0400: 11 uA not supported** error message.
TTL
Input signal of encoder is TTL signal.

Default: No value, parameter optional

Access: LEVEL3

Reaction: NOTHING

MP_encoderFreq

Input frequency of position encoder input for handwheel
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **Fast**
Input frequency is 350 kHz
Slow
Input frequency is 50 kHz

Default: No value, parameter optional

Access: LEVEL3

Reaction: NOTHING

MP_encoderResistor

Terminating resistor of position-encoder input for handwheel
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **Without**
Without terminating resistor
120 Ohm
With terminating resistor

Default: No value, parameter optional

Access: LEVEL3

Reaction: NOTHING

Evaluation of handwheel pulses

Define the evaluation of the handwheel pulses in the following parameters:

- **MP_countDir**: Counting direction of the handwheel for this axis. If a serial handwheel or an HRA 110 is connected to X23, the counting direction can be specified for each axis.
- **MP_distPerRevol**: Path of the axis per handwheel revolution. The value applies to all connected handwheels, including serial handwheels at X23. With this parameter a traverse per handwheel revolution for each axis can be entered for HR 150 handwheels connected to HR 110. The value is multiplied by the entry in **MP_speedFactor** (CfgHandwheel).
- **MP_incrPerRevol**: Number of increments per handwheel revolution
- **MP_rasterPerRevol**: Detent steps per handwheel revolution (only for handwheels with detent)

MP_countDir

Axis-dependent counting direction of the handwheel
Available from NCK software version: 597 110-02.

Format: Selection menu

Selection: The parameter applies both for the handwheel at the position input and for the serial handwheel connected to X23.

Positive
Counting direction is positive

Negative
Counting direction is negative

Default: Positive

Access: LEVEL3

Reaction: NOTHING

MP_distPerRevol

Axis traverse per handwheel revolution
Available from NCK software version: 597 110-02.

Format: Numerical value

Input: Distance in [mm] with up to 9 decimal places
Applies for **all** connected handwheels (handwheel at position input and at X23).
The value is multiplied by **MP_speedFactor** (CfgHandwheel).

Default: 1 [mm]

Access: LEVEL3

Reaction: NOTHING

MP_incrPerRevol

Increments per revolution of handwheel at encoder input
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: 0 to max. value
0: Standard value for HEIDENHAIN handwheel (20 000 increments)

Default: No value, parameter optional

Access: LEVEL3

Reaction: NOTHING

MP_rasterPerRevol

Detent steps per revolution of handwheel at encoder input
Available from NCK software version: 597 110-01.

Format: Array

Input: 0 to max. value
0: Handwheel without detent

Default: No value, parameter optional

Access: LEVEL3

Reaction: NOTHING

Handwheel positioning up to software limit switch

The optional parameter **MP_decToSoftLimit** is used to specify a braking ramp for handwheel-positioning of the axis to the software limit switch. Limit the maximum axis speed with **MP_decToSoftLimit** such that no overshoot of the axis occurs when the software limit switch is reached.

Proceed as follows to determine the value to be set for the **MP_decToSoftLimit** parameter:

- Record the momentary speed of the axis **v (N act)** with the integrated oscilloscope
- During measurement with the oscilloscope, quickly move the axis to the software limit switch with the axis handwheel.

The control uses the following formula for limiting the maximum possible speed on reaching the software limit switch:

$$Vmax = \sqrt{2 \cdot \text{Distance to SW limit switch} \cdot \text{MP\_decToSoftLimit}}$$

- Experiment with various values for the parameter **MP_decToSoftLimit** and compare the different oscilloscope results with each other.

MP_decToSoftLimit

	Brake ramp for handwheel motions to the software limit switch (MP_swLimitSwitchPos and MP_swLimitSwitchNeg) Available from NCK software version: 597 110-03.
Format:	Numerical value
Input:	0.000 000 000 to max. value [m/s^2] For rotary axes, the brake ramp is specified in [1000°/s^2]. 0: Value from **MP_maxAcceleration** is used
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	NOTHING

Settings in the configuration editor	MP number
System CfgHandwheel **speedFactor**	100607
Axes ParameterSets [Key name for the parameter set] CfgAxisHandwheel **distPerRevol**	400203

The distance covered by an axis per handwheel revolution depends on the traverse per handwheel revolution and the handwheel transmission ratio.

- In **MP_distPerRevol**, define the distance traversed per handwheel revolution for each axis moved by handwheel
- In **MP_speedFactor**, define the handwheel transmission for three transmission ratios

MP_distPerRevol

Traverse per handwheel revolution
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 000 001 to max. value [mm]
Default:	1 [mm]
Access:	LEVEL3
Reaction:	NOTHING

MP_speedFactor

Handwheel transmission ratio
Available from NCK software version: 597 110-01.

Format:	Array [0...2]
Input:	0 to 100 [%] [0] = First ratio in [%] [1] = Second ratio in [%] [2] = Third ratio in [%]
Default:	1, 10 and 100 [%]
Access:	LEVEL3
Reaction:	RUN

The transmission ratio is set either at the control panel or directly on the handwheel.

7.6.4 Assigning a handwheel to an axis

Module 9036 allows you to assign a handwheel connected to connection X23 to an axis (NC or PLC axis). The module also defines the transmission ratio.

Use Module 9035 to find the axis to which the handwheel is assigned.

Module 9036 Write NC status information
The module modifies status information from the NC. The status information to be modified is transferred by function number.

- **Select the handwheel axis** function: The handwheel connected to connector X23 of the MC is assigned to an axis.
- **Set the handwheel transmission ratio:** The handwheel transmission ratio is defined.

You can change the following status information:

Number of the function	Function	Value
0	Handwheel subdivision X	0 to 10: Subdivision factors
1	Handwheel subdivision Y	0 to 10: Subdivision factors
2	Handwheel subdivision Z	0 to 10: Subdivision factors
3	Handwheel subdivision IV	0 to 10: Subdivision factors
4	Handwheel subdivision V	0 to 10: Subdivision factors
5	Handwheel subdivision for all axes	0 to 10: Subdivision factors
6	Select the handwheel axis	Index from MP_axisList or -1: Deselect all axes
7	Set the handwheel transmission ratio	■ 0: Slow ■ 1: Normal ■ 2: Fast
8	Reserved	–
9	Reserved	–
10	Limit value for jog increment	0.0001 mm to 50 mm or –1: Cancel the limiting 2: New jog increment = minimum (programmed jog increment, previous limit value) and cancel limitation
11 to 19	Handwheel subdivision of axes 1 to 9	0 to 10: Subdivision factors

Note

On the MANUALplus 620, only function 6 "Select the handwheel axis" from the table above is used.

Constraints:

- Handwheel subdivision 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.
- Call codes 0 to 4 refer to the 5 axes assigned to the axis keys X/Y/Z/IV/V. Call codes 11 to 19 refer to the internal axis numeration.
- The value for the jog increment limitation is transferred in [1/10000 mm].
- If a jog limitation is entered in the inch mode, the limitation value [degrees] for rotary axes is calculated from the limitation value [mm] / 24.5.

Possible errors:

- The input parameter <Number of the function> does not refer to any overwritable status information in this software version.
- The transferred value is outside of the range valid for this status information.
- Entry of this status information is disabled, e.g. via the machine configuration.

Call:

PS	B/W/D/K	<Number of the function>
PS	B/W/D/K	<Value to be written>
CM	9036	
PL	B/W/D	<Error code> 0: Status written 1: Incorrect status code 2: Transferred value is out of range 3: Input disabled

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Status information was written
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Transferred value is out of range
	2	Incorrect number of the status information
	3	Transferred value is out of range
	6	Input disabled

Module 9035 Read NC status information

The module reads status information from the NC. The status information to be read is transferred by function number.

The following handwheel information can be read:

Transferred number		Return value
9	Handwheel axis	Finds the axis which is assigned to the handwheel connected to connection X23 of the MC. –1: None or more than one 0: X axis 1: Y axis 2: Z axis 3: IV axis 4: V axis 5: VI axis 6: VII axis etc.
10	Handwheel axis, bit-encoded	Bit 0: X axis Bit 1: Y axis Bit 2: Z axis Bit 3: IV axis Bits 4 to 13: Axes 5 to 14 (only available for PLC programs that work with API 1.0)
	Handwheel subdivision factor	0 to 10
11	X key	
12	Y key	
13	Z key	
14	IV key	
15	V key	
20	HR 410 speed	0: Slow 1: Medium 2: Fast
	Handwheel subdivision factor	0 to 10
31	Axis 1	
32	Axis 2	
33	Axis 3	
34	Axis 4	
35	Axis 5	
36	Axis 6	
37	Axis 7	
38	Axis 8	
39	Axis 9	

Call:
PS B/W/D/K <Number of the function>
CM 9035
PL B/W/D <Status information>

Error recognition:

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

7.6.5 HR 410 portable handwheel

- In **MP_type,** select HR 410.
- In **MP_initValues,** define whether the keys on the handwheel are evaluated by the NC or PLC.

Initialization values for HR 410 handwheel:

- **MP_initValues** [0]:
  - 0: Evaluation of the keys by NC, including LEDs, except for three function keys
  - 1: Evaluation of the keys by PLC
- Entry in **MP_initValues** [1 to 7]: Reserved

Keys evaluated by the NC: **MP_initValues [0] = 0**	Keys evaluated by the PLC: **MP_initValues [0] = 1**

X		IV
Y		V
Z		ISTWERT-ÜBER-NAHME
VORSCHUB KLEIN	VORSCHUB MITTEL	VORSCHUB GROß
–		+
O109 I173	O110 I174	O111 I175

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

The function keys I173 to I175 are always evaluated by the PLC.

Assignment of axis keys:

- When the keys are evaluated by the NC, the following assignment of axis keys applies (the key designations refer to the figure above):
  - X key: Index 0 from **MP_axisList**
  - Y key: Index 1 from **MP_axisList**
  - Z key: Index 2 from **MP_axisList**
  - IV key: Index 3 from **MP_axisList**
  - V key: Index 4 from **MP_axisList**
- When the keys are evaluated by the PLC, the handwheel axis is selected with Module 9036. This is the only case in which a PLC axis can be assigned to the handwheel with Module 9036.

Use the three feed rates from **MP_feedFactor** to influence the feed rate when you press an axis-direction key. You set the feed rate on the HR 410 (keys for low feed rate, medium feed rate, and high feed rate).

- In **MP_type**, select HRNAX
- Specify in **MP_initValues** the assignment of the third handwheel

Note

- Through the config object **CfgAxisHandwheel** you can specify for each axis the counting direction and the distance per handwheel revolution. To do so, open the config object **CfgAxisHandwheel** in the parameter set of the desired axis. Enter the values in **MP_countDir** and **MP_distPerRevol**.
- The parameters **MP_incrPerRevol**, **MP_rasterPerRevol** and **MP_countDir** (from System/CfgHandwheel) apply for all handwheels connected to the HRA 110. The values in CfgAxisHandwheel overwrite these global values.
- The handwheel inputs X1 and X2 are permanently assigned to the logical axes 0 and 1 (**MP_axisList**[0,1]). The assignment of the third handwheel (input X3) depends on the initialization values. It can either be permanently assigned to an axis or the axis can be set with the switch S2 (on the HRA 110).

Initialization of the handwheel adapter

Initialization values for HRA 110 handwheel adapter:

- MP_initValues [0] – assignment of the third handwheel:
  - 0:Switch position 1 – Handwheel is assigned to logical axis 2
    Switch position 2 – Handwheel is assigned to logical axis 3
    Switch position 3 – Handwheel is assigned to logical axis 4
  - 1:Switch position 1 – Handwheel is assigned to logical axis 0
    Switch position 2 – Handwheel is assigned to logical axis 1
    Switch position 3 – Handwheel is assigned to logical axis 2
    Switch position 4 – Handwheel is assigned to logical axis 3
    Switch position 5 – Handwheel is assigned to logical axis 4
  - 2:Switch position 3 – Handwheel is assigned to logical axis 2
    Switch position 4 – Handwheel is assigned to logical axis 3
    Switch position 5 – Handwheel is assigned to logical axis 4
- MP_initValues [1] – for permanent assignment of the third handwheel, the following applies:
  - 1: Logical axis 0
  - 2: Logical axis 1
  - 4: Logical axis 2
  - 8: Logical axis 3
  - 16: Logical axis 4
- MP_initValues [2] – assignment of the third handwheel:
  - 0: Axis selection switch as defined in MP_initValues[0]
  - 1: Permanently as defined in MP_initValues[1]
- Entry in MP_initValues [3 to 7]: Reserved

Switch position 1 on the HRA 110 corresponds to the left stop.

Assignment of switch positions to PLC inputs

The tables below list the assignments of switch positions of S1 and S2 to the PLC inputs I160 to I175.

The two switches work with a 0 V logic circuit.

Example: If switch S1 is in position 3, input I162 is logically 0, and the inputs I160, I161, I163 to I167 are logically 1.

Step switch 1

Switch position	PLC input
1 (at the left stop)	I160
2	I161
3	I162
4	I163
5	I164
6	I165
7	I166
8 (at the right stop)	I167

Step switch 2: Axis selection switch

Switch position	PLC input
1 (at the left stop)	I168
2	I169
3	I170
4	I171
5	I172
6	I173
7	I174
8 (at the right stop)	I175

7.7 Override

7.7.1 Override devices

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcOverrideDev	
[Key name of override device]	
source	104301
mop	104304
mode	104302
values	104303

The control supports the following override devices:

- Up to three potentiometers
- Group of keys (you can use two machine operating keys for setting the override value).

The input values of the override devices (potentiometer setting or key input) are evaluated as follows:

- **DISCRETE:** The control converts the input values into a maximum of 64 discrete override values.
- **LINEAR:** The control interpolates the input values linearly between the minimum and maximum override value.
- **CURVE:** The control uses a curve to convert the lower input values into override values. In this way you can achieve a finer resolution of the lower override values. Higher input values are again interpolated linearly.

Define the override devices as follows:

- Define the override device in **MP_source**
- In **MP_mop**, enter the key name of the machine operating panel on which the override source is located. This is only required for future software versions that support the connection of more than one machine operating panel.
- In **MP_mode**, define the evaluation of the override input
- In **MP_values**, define the discrete override values or the interpolation points of the curve

MP_source

Selection of configurable source for override values
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **OVR1**
Potentiometer 1
OVR2
Potentiometer 2
OVR3
Potentiometer 3
KEY
Group of keys
Default: OVR1
Access: LEVEL3
Reaction: RESET

MP_mop

Enter the key name of the machine operating panel on which the override source is located. This MP is only required for future software versions that support the connection of more than one machine operating panel.
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **MB**
Key name of the connected machine operating panel
Default: MB
Access: LEVEL3
Reaction: RESET

MP_mode

Evaluation of override values
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **DISCRETE**
The key inputs or potentiometer settings are converted in up to 25 discrete override values from MP_values.
LINEAR
The input from the override device is interpolated linearly between the minimum and maximum override value.
CURVE
The input values of the override device are converted using the curve defined in MP_values.
You define the curve with up to 64 interpolation points in MP_values. The input values of the override device are again interpolated linearly above the last interpolation point specified. If no curve is defined in MP_values, the control uses a standard curve.
Default: DISCRETE
Access: LEVEL3
Reaction: RESET

MP_values

Discrete values or interpolation points for curve
Available from NCK software version: 597 110-01.

Format: Array [0 to 63]

Input: 0.000 to 200.000 [%]
The meaning of the parameter depends on MP_mode:
MP_mode=DISCRETE: Enter override values for a maximum of 64 key inputs or potentiometer settings.
MP_mode=LINEAR:
MP_values has no function.
MP_mode=CURVE:
Define a curve with up to 64 interpolation points. The override values are taken from the curve. Linear interpolation is again effective above the last interpolation point specified.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcPeriphery	
overrideFullRatio	103406
overrideDelta	103407
overrideIntegDelta	103408

MP_overrideFullRatio allows you to compensate voltage losses.

MP_overrideFullRatio

Compensation for cable losses of the override potentiometer
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.5000 to 1.0000
Default: 0.5 (optional parameter)
Access: LEVEL2
Reaction: NOTHING

Use **MP_overrideDelta** and **MP_overrideIntegDelta** to influence the sensitivity of the override potentiometers. **MP_overrideDelta** suppresses short-term fluctuations and **MP_overrideIntegDelta** compensates the signal drift.

MP_overrideDelta

Compensation for thermal noise in override potentiometers
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.00010 to 0.10000
Default: 0.0001 (optional parameter)
Access: LEVEL2
Reaction: NOTHING

MP_overrideIntegDelta

Compensation for thermal noise in override potentiometers
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.00010 to 1.00000
Default: 0.0001 (optional parameter)
Access: LEVEL2
Reaction: NOTHING

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcOverrideS	
[Key name of spindle]	
minimal	104401
maximal	104402
source	104402

Speed override

▶ In the parameter object **CfgPlcOverrideS,** create a parameter block for each spindle to which a spindle speed override is to apply.

MP_minimal

Minimum value for override
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 to 100.000 [%]
Default:	0
Access:	LEVEL3
Reaction:	RESET

MP_maximal

Maximum value for override
Available from NCK software version: 597 110-01.

Format:	Numerical value
Input:	0.000 to 200.000 [%]
Default:	150
Access:	LEVEL3
Reaction:	RESET

MP_source

Source for override values
Available from NCK software version: 597 110-01.

Format:	Selection menu
Selection:	Key name for override device from CfgPlcOverrideDev
Default:	–
Access:	LEVEL3
Reaction:	RESET

The percentage adjusted with the speed override is entered by the NC in **NN_SpiOverrideInput** and **PP_SpiOverride.**

You can change the percentage through the PLC:

- Enter the desired percentage in **PP_SpiOverride.** The NC immediately takes over the new value

PLC operand / Description	Model
NN_SpiOverrideInput Speed override set [%]	D
PP_SpiOverride Speed override entered by the PLC [%]	D

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgPlcOverrideF	
minimal	201901
maximal	201902
source	201903

- In the channel-sensitive parameter object **CfgPlcOverrideF,** create a parameter block for each machining channel (slide) to which a feed rate override is to apply.

MP_minimal

Minimum value for override
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.00 to 100.00 [%]
Default: 0
Access: LEVEL3
Reaction: RESET

MP_maximal

Maximum value for override
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.00 to 200.00 [%]
Default: 150
Access: LEVEL3
Reaction: RESET

MP_source

Source for override values
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key name for override device from CfgPlcOverrideDev
Default: PotentiometerF
Access: LEVEL3
Reaction: RESET

Note

The feed rate override also applies to rapid traverse if the rapid traverse override is not active.

The percentage adjusted with the feed rate override is entered by the NC in **NN_ChnFeedOverrideInput** and **PP_ChnFeedOverride.**

You can change the percentage through the PLC:

- Enter the desired percentage in PP_ChnFeedOverride. The NC immediately takes over the new value

PLC operand / Description	Model
NN_ChnFeedOverrideInput Feed-rate override set [%]	D
PP_ChnFeedOverride Feed-rate override entered by the PLC [%]	D

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgPlcOverrideR	
minimal	202001
maximal	202002
source	202003

- In the channel-sensitive parameter object **CfgPlcOverrideR,** create a parameter set for each machining channel (slide) to which a rapid traverse override is to apply.

MP_minimal

Minimum value for override
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.00 to 100.00 [%]
Default: 0
Access: LEVEL3
Reaction: RESET

MP_maximal

Maximum value for override
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.00 to 200.00 [%]
Default: 150
Access: LEVEL3
Reaction: RESET

MP_source

Source for override values
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key name for override device from CfgPlcOverrideDev
Default: PotentiometerF
Access: LEVEL3
Reaction: RESET

You can change the percentage through the PLC:

- Enter the desired percentage in PP_ChnRapidFeedOverride. The NC immediately takes over the new value

PLC operand / Description	Model
NN_ChnRapidFeedOverrideInput Rapid traverse override set [%]	D
PP_ChnRapidFeedOverride Rapid traverse override entered by the PLC [%]	D

7.8.1 Diagnosis of the external PL

Module 9007 Read the diagnostic information of a PLC input/output unit
The module provides diagnostic information about the PL 510. To save computing time, refrain from continuously calling this module.

Call:
PS B/W/D/K <Number of the PLB 510 basic module (0 to 3)>
PS B/W/D/K <Number of the information>
0: Reserved
1: Reserved
2: Reserved
3: Total number of PL 510 on this MC
4: Reserved
5: Reserved
CM 9007
PL B/W/D/K <Diagnostic information>
0 to 4: Number of PL 510

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Diagnostic information was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Invalid code
	2	Invalid PL module
	24	Module was called in a spawn job or submit job
	51	Diagnostic function cannot be read because PL 510 system is running in modular mode.

Module 9137 Read diagnostic information of the PL 510

The module provides diagnostic information about the PL 510. To save computing time, refrain from continuously calling this module.

Call:

PS B/W/D/K <Number of the PLB 510 basic module (0 to 3)>
PS B/W/D/K <Number of the slot (0 to 3)>
PS B/W/D/K <Number of the information>
0: Possible mode of operation (PL modular mode)
1: Active mode of operation
2: Reserved
3: Reserved
4: Basic module code
5: Status of the basic module
6: Module model in the slot
7: Reserved
8: Reserved
9: Status of the module in the slot
10: Logical status of the outputs of a PLD 16-8
11: Short-circuit of the outputs of a PLD 16-8
12: Number of connected PL 510
CM 9137
PL W/D <Diagnosis information>
Information no. 0:
0: "PL 510" operating mode not possible (only "PL 4xxB" operating mode, without new functions of the PL 510)
1: "PL 510" operating mode possible
Information no. 1:
0: "PL 4xxB" operating mode active (without new functions of the PL 510)
1: "PL 510" operating mode active
Information no. 2:
0: No PLB 510
1: PLB 510
Information no. 3:
0..15: Hardware version
Information no. 4:
0..15: Hardware code
Information no. 5:
Bit 0 = 1: Power supply of the PLB 510 is OK
Bits 1 to 15: Reserved
Information no. 6:
0: No module in slot
1: Reserved
2: PLD 16-8
3: PLA 4-4 in slot
Information no. 7:
0..15: Hardware version (identifies the function status of the module)
Information no. 8:
0..15: Hardware code (identifies a hardware change state)

Information no. 9:
PLD 16-8 (module type 2):
Bit 0 = 1: Power supply outputs 0 to 3 are OK
Bit 1 = 1: Power supply outputs 4 to 7 are OK
Bit 2 = 1: Short circuit at an output
Bit 3 = 1: At least one output idle (< 300 mA)
Bits 4 to 6: No meaning
Bit 7 = 1: Output 7 is a programmable output (otherwise "control is ready")
Bits 8 to 31: No meaning
PLA 4-4 (module type 3):
Bit 0 = 1: Power supply of the inputs is OK
Bits 1 to 31: No meaning

Information no. 10:
Bit 0: Status of output 0 (PLD 16-8)
to
Bit 7: Status of output 7 (PLD 16-8)

Information no. 11:
Bit 0: Short circuit at output 0 (PLD 16-8)
to
Bit 7: Short circuit at output 7 (PLD 16-8)
Bit 8: Idle (< 300 mA) Out0 (PLD 16-8)
to
Bit 15: Idle (< 300 mA) Out7 (PLD 16-8)

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Diagnostic information was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Invalid code
	2	Invalid basic module number or slot number
	24	Module was called in a spawn job or submit job
	51	Function not possible since no PL 510 system is connected or the I/O module type is incorrect for the desired type of information.

Explanations:

- **Hardware version** – identifies the function status of the module. When a function that affects the software is changed, the code number identifying the version is increased by one. Modules with low version numbers cannot be replaced by modules with higher version numbers.
- **Hardware code** – identifies the hardware change state. The hardware changes do not affect the functions. It is not necessary to take the hardware code into account when a module is replaced.
- **Short circuit:** Short circuit codes (both the group signal as well as the output-specific messages) are modal. They are indicated by the error LED on the I/O module. In the event of a short circuit, the affected output is automatically reset. With Module 9139 you can withdraw the short-circuit code and then drive the output again.
- **No load:** The limit values for no-load code are minimum 20 mA and maximum 300 mA.

Module 9139 Monitoring functions for the PL 510 PLC input/output units
The short circuit of an output of the PLD 16-8 is indicated by an LED, and the output is reset. Short-circuit monitoring remains in place, and must therefore be reset with Module 9139.

To save computing time, refrain from repeatedly calling this module.

Call:
PS B/W/D/K <Function>
0: Reserved
1: Reserved
2: Reset short-circuit monitoring
CM 9139

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Short-circuit monitoring was reset
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Invalid code
	2	Invalid basic module number or slot number
	24	Module was called in a spawn job or submit job
	51	Function not possible or not a PL 510

7.8.2 24 V– switching input/outputs

In PLC addresses you can find the current states of the switching inputs and outputs.

For the current states of the inputs/outputs of the external PL:

- Read all inputs with Module 9002
- Update all outputs with Module 9005

With Module 9004 you can evaluate the rising or falling edge of the PLC inputs.

Note

Before the PLC program is converted, the PLC outputs are reset. In addition, the memory of the PLC outputs is reset.

During a loss of power (power fail), the control tries to reset the PLC outputs.

Settings in the configuration editor	MP number
System PLC CfgPlcDeferCutoff [Key name of the output] **port** **time**	 103601 103602

If all PLC outputs are switched off (e.g., during PLC program compilation or due to a PLC run-time error), the outputs defined via the configuration object **CfgPlcDeferCutoff** can be switched off delayed by a time between 0 and 5 seconds. The delay only affects outputs that cannot be switched off by an emergency stop, since for the outputs that can be shut off by an emergency stop, the 24 V supply is shut off immediately.

MP_port

	Number of the physical output port on the PL Available from NCK software version: 597 110-02.
Format:	Numerical value
Input:	0 to 31
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	NOTHING

MP_time

	Delay time during switch-off Available from NCK software version: 597 110-02.
Format:	Numerical value
Input:	0.1 to 5.0 [s]
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	NOTHING

Module 9002 Read the inputs of a PLC input/output unit

The module downloads the current states of the PLC input/output unit. These states can be read in the PLC addresses (process image). The contents of the PLC addresses remain unchanged until you call this module again.

For PL 510, inputs of empty slots are not read.

The program can be called only in the cyclic PLC program.

Call:
PS B/W/D/K <Number of the PLB 510 basic module (0 to 3)>
CM 9002

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Inputs were read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid PL number
	24	Module was called in a spawn job or submit job

Module 9004 Read the edges of PLC inputs

The module sets, upon falling or rising edges of the PLC inputs, specified edge markers or bits in the specified byte range. Changes in the inputs are recognized only if a change also occurs in the PLC addresses (see Module 9002).

Ensure that the specified edge markers or edge bytes are in an unoccupied area. The edge bytes are written beginning with the least significant bit. Superfluous bits are erased.

Call:

PS B/W/D/K <Number of the first PLC input>
PS B/W/D/K <Number of the first edge marker or edge byte>
PS B/W/D/K <Number of PLC inputs>
PS B/W/D/K <Edge evaluation>
0: Rising edge. Entry in edge marker
1: Falling edge. Entry in edge marker
2: Rising edge. Entry in edge byte
3: Falling edge. Entry in edge byte
CM 9004

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Outputs were set
	1	Invalid transfer parameter

Module 9005 Set the outputs of PLC input/output unit
The module overwrites the outputs of the PLC input/output unit with the values from the PLC addresses. The outputs are set or reset immediately at the time of module execution and remain in their condition until they are set or reset again by this module.

For the PL 510, the outputs of empty slots are not overwritten with values from the PLC addresses.

The program can be called only in the cyclic PLC program.

Call:
PS B/W/D/K <Number of the PLB 510 basic module (0 to 3)>
CM 9005

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Outputs were set
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid PL number
	24	Module was called in a spawn job or submit job

Settings in the configuration editor	MP number
System PLC CfgPlcPeriphery **pt100Discrete**	 103404

Transferring the analog inputs of the MC

±10 V– analog inputs and inputs for Pt 100 thermistors are available either on the MC unit or on the external PL (see Chapter 3, Mounting and Electrical Installation). Use Module 9003 to load the current temperatures of the Pt 100 thermistors or the values of the analog inputs.

To transfer the analog inputs and temperature inputs of the external PL, use Module 9138.

Module 9003 Transfer the analog input of the MC
This module reads the current value of the specified analog input, regardless of whether it is actually connected.

Value range ±10 V input: –10 V to +10 V, at a resolution of 10 mV
Value range Pt 100 input: 0 to 100 °C, at a resolution of 0.1 °C

The module can be called only in the cyclic PLC program.

Call:

PS	B/W/D/K	<Number of analog input> 0 to 63: Reserved 64 to 66: ±10 V input on connection X48 67 to 69: Pt 100 input on connection X48
CM	9003	
PL	W/D	<Analog value> No. 64 to 69: Natural number with the unit 0.01 V or 0.1 °C

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Input was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid PL number or invalid analog input number
	24	Module was called in a spawn job or submit job

In **MP_pt100Discrete** you define whether the values of the Pt 100 inputs are transferred immediately or with a change rate of 1 K/s. The disadvantage of transfer with a change rate of 1 K/s is that at high change rates it may take some time until the correct temperature has been transferred. For example, it would take 30 seconds to correctly read a temperature change of 30 K. An advantage of this, however, is a low sensitivity to disturbance: the temperature display will not jump back and forth between two values:

- If you wish to transfer the values of the Pt 100 inputs immediately, set **MP_pt100Discrete** = True
- If you wish to work with a change rate of 1 K/s, set **MP_pt100Discrete** = False.

MP_pt100Discrete

	Transfer of PT100 values Available from NCK software version: 597 110-01.
Format:	Selection menu
Selection:	**TRUE** Transfer value immediately **FALSE** Transfer value at 1 K/s
Default:	TRUE
Access:	LEVEL2
Reaction:	NOTHING

Transferring the analog inputs of the PL 510

Module 9138 Read analog input of the PL 510

The module transfers the current value of the given analog input of the PL 510.

Value range ±10 V input: –10 V to +10 V, at a resolution of 0.01 V
Value range Pt 100 input: 0 to 100 °C, at a resolution of 0.01 °C

To save computing time, refrain from repeatedly calling this module. The module can be called only in the cyclic PLC program.

Call:

PS	B/W/D/K	<Number of the PLB 510 basic module (0 to 3)>
PS	B/W/D/K	<Number of the slot (0 to 3)>
PS	B/W/D/K	<Number of the analog input (0 to 7)>
CM	9138	
PL	B/W	<Analog value> Analog inputs 0 to 3: Natural number (–1000 to +1000) in steps of 0.01 V Analog inputs 4 to 7: Natural number (0 to 10 000) in steps of 0.01 °C

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Input was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid basic module number or slot number
	24	Module was called in a spawn job or submit job
	51	Function not possible or not a PL 510 or PLA 4-4 analog module

7.8.4 Analog outputs

You can select the analog outputs 1 to 6 at connection X8.

Note

Every analog axis or analog spindle needs an analog output. These outputs are no longer available to the PLC.

Module 9130 Output analog voltage
The module places an analog voltage on an analog output. The voltage is output with a slight delay after the end of the PLC scan.

Call the module only once for each output per PLC scan!

Format: 1 mV
Voltages greater than +10 V or less than –10 V are limited to the respective maximum value.

Call:

PS	B/W/D/K	<Number of the analog output> 1 to 6: Analog outputs 1 to 6 (X8) 7 to 13: Analog outputs 7 to 13 (X9)
PS	B/W/D/K	<Analog voltage in mV>
CM	9130	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Analog voltage was output
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Invalid analog output
	2	Disabled analog output

7.9 Operating Times and System Times

7.9.1 Measuring operating times

Settings in the configuration editor	MP number
System	
CfgPlcOperTimes	
displayPlcTimes	105001
resetPlcTimes	105002
resetNcTimes	105003
textNumber	105004

The control can measure up to 11 operating times (operating hours counter) and store them in a file:

Operating time	Meaning
TNCTIME	Control on
MACHINETIME	Machine on (NC operating time)
PROGTIME	Program run (NC operating time)
PLCTIME0 to PLCTIME7	Freely definable PLC operating time

- In **MP_resetPlcTimes,** specify the PLC operating times you can reset with the code number **857282**
- In **MP_resetPlcTimes,** specify the NC operating times you can reset with the code number **857282**
- In **MP_displayPlcTimes,** define the operating times you want to display
- In **MP_textNumber,** define the dialog messages to be displayed for the individual PLC operating times

The time is measured in seconds and is updated every minute during the run time. When the control is switched off, no more than one minute is lost.

The NC starts and stops the NC operating hours counter (**`Control on`**, **`Machine on`** and **`Program run`**).

PLC operating hours counters 1 to 8:

- Start with Module 9190.
- Stop with Module 9191.

All operating times are saved during a hard-disk backup.

Use the following modules to evaluate or change the operating times:

- Module 9190: Start the PLC operating hours counter
- Module 9191: Stop the PLC operating hours counter
- Module 9192: Transfer the operating hours counter
- Module 9193: Set the operating hours counter
- Module 9194: Alarm when operating times are exceeded

Note

When measuring the operating time in the Program Run mode, the different operating mode groups are currently not distinguished.

MP_displayPlcTimes

Display PLC operating times
Available from NCK software version: 597 110-01.

Format: String
Input: Binary value
Bits 0 to 7 represent PLC operating times 1 to 8
0: Do not display
1: Display
Default: %11111111
Access: LEVEL2
Reaction: NOTHING

MP_resetPlcTimes

Reset PLC operating times with the code number
Available from NCK software version: 597 110-01.

Format: String
Input: Binary value
Bits 0 to 7 represent PLC operating times 1 to 8
0: Do not reset
1: Reset
Default: %00000000
Access: LEVEL2
Reaction: NOTHING

MP_resetNcTimes

Reset NC operating times with the code number
Available from NCK software version: 597 110-01.

Format: String
Input: Binary value
Bit 0: "Control on" operating time
Bit 1: "Machine on" operating time
Bit 2: "Program run" operating time
0: Do not reset
1: Reset
Default: %000
Access: LEVEL2
Reaction: NOTHING

The dialog texts for the PLC operating hours counter are defined in the text file %OEM%\plc\language\en (or languages other than English).

MP_textNumber

Dialogs for PLC operating times
Available from NCK software version: 597 110-01.

Format: Array [0...7]

Input: [0] corresponds to the text for PLC time 1, etc.
In the text file for PLC dialogs, indicate the line number of the dialog text (%OEM%\plc\language\en (or language other than English)).

Default: [0]: 16
[1]: 17
[2]: 18
[3]: 19
[4]: 20
[5]: 21
[6]: 22
[7]: 23

Access: LEVEL2

Reaction: NOTHING

Module 9190 Start the PLC operating hours counter

The module activates one or more operating hours counters. The control measures the time that elapses until activation is completed. Use Module 9191 to cancel activation. Bits 0 to 7 of the transfer value represent the PLC operating hours counters 1 to 8. For every bit set, the associated counter is started.

Call:

PS B/W/D/K <PLC operating time>
Bits 0 to 7 represent PLC operating times 1 to 8

CM 9190

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	PLC operating time started
	1	Incorrect parameter

Module 9191 Stop the PLC operating hours counter

The module stops one or more PLC operating hours counters. The time measured by the control is saved. Bits 0 to 7 of the transfer value represent the PLC operating hours counters 1 to 8. For every bit set, the associated counter is stopped.

Call:

PS B/W/D/K <PLC operating time>
Bits 0 to 7 represent PLC operating times 1 to 8

CM 9191

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	PLC operating time stopped
	1	Incorrect parameter

Module 9192 Transfer the operating hours counter
The module reads the given PLC or NC operating hours counter.

The current value is transferred in seconds. If the value is greater than 2 147 483 648 (approx. 69 years), a negative number will be transferred.

Call:

PS	B/W/D/K	<Number of the operating time> –3: **Control on** –2: **Machine on** –1: **Program run** 0 to 7: PLC operating time counter 1 to 8
CM	9192	
PL	B/W/D	<Current time [s]> –1: Error

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	PLC operating time was read
	1	Incorrect transfer value, or module was not called in a spawn or submit job

Module 9193 Set the operating hours counter
The module overwrites the given PLC or NC operating hours counter. The old value is lost irretrievably.

The value of the NC operating hours counters may only be changed in exceptional cases (e.g. when the control is exchanged).

The time for **Control on** cannot be overwritten.

Transfer all values greater than 2 147 483 648 (approx. 69 years) as negative numbers.

Call:

PS	B/W/D/K	<Number of the operating time> –3: **Control on** –2: **Machine on** –1: **Program run** 0 to 7: PLC operating times 1 to 8
PS	B/W/D/K	<New time [s]>
CM	9193	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Operating time was overwritten
	1	Incorrect transfer value, or module was not called in a spawn or submit job

Module 9194 Alarm when operating time exceeded

The module activates a monitoring function in the NC, which sets a PLC marker when the given maximum time for a PLC or NC operating hours counter is exceeded. The marker is set the first time the maximum time is exceeded, and then cyclically once per minute.

The marker can be delayed by max. 59 s the first time it is set. All values greater than 2 147 483 648 (approx. 69 years) must be transferred as negative numbers.

If you enter the value zero as the alarm threshold, the function is deactivated.

Call only in a submit job or spawn job.

Call:

PS	B/W/D/K	<Number of the operating time>
		–3: **Control on**
		–2: **Machine on**
		–1: **Program run**
		0 to 7: PLC operating times 1 to 8
PS	B/W/D/K	<Alarm threshold [s]>
PS	B/W/D/K	<Number of the alarm markers>
CM	9194	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Alarm function activated
	1	Incorrect transfer value, or module was not called in a spawn or submit job

Settings in the configuration editor	MP number
System CfgSystemTime **offsetToUTC**	 105201

System time management varies depending on the control system:

- Single-processor systems internally operate with UNIX system time. This is the number of seconds since 0:00 hours on January 1, 1970. The parameter **MP_offsetToUTC** defines the time difference between Universal Time (Greenwich time) and local time. It is the user's task to adjust between Daylight Saving Time and Standard Time. Daylight Saving Time or Standard Time can be set in **MP_offsetToUTC**.
- For dual-processor systems, the Windows operating system provides the system time. Windows automatically adjusts for Daylight Saving Time or Standard Time.

Use the following modules to transfer the system time:

- Module 9195: Transfer the real-time clock (UNIX system time)
- Module 9055: Convert time (binary) to formatted string (and consider **MP_offsetToUTC**)

- **Single-processor systems**: In **MP_offsetToUTC**, define the time difference between Universal Time and local time, taking into account Daylight Saving Time or Standard Time
- **Dual-processor systems**: Set **MP_offsetToUTC** = 0

MP_offsetToUTC

	Time difference to Universal Time Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	–12 to +14 [hours] with up to 2 decimal places 0: Universal Time (Greenwich Mean Time) 1: Central European Time (CET) 2: Central European Daylight Time (CEDT)
Default:	1
Access:	LEVEL1
Reaction:	NOTHING

Module 9195 Transfer the real-time clock

The module reads the time of the real-time clock. A double word is returned, which contains the number of seconds accumulated since 0:00 hours on January 1, 1970 (UNIX system time).

Call:

CM	9195	
PL	D	<System time> Number of seconds since 0:00 hours on January 1, 1970.

Module 9055 Convert time (binary) to formatted string

The module provides the date and time (local time) as an ASCII string with configurable format.

The module converts the binary UNIX system time (number of seconds since 0:00 hours on January 1, 1970) into an ASCII string, taking into account the time difference between local time and Universal Time (Greenwich time) defined in MP_offsetToUTC.

Call:

PS B/W/D/K <System time>
Number of seconds since 0:00 hours on January 1, 1970.
PS B/W/D/K <String number for the result>
PS B/W/D/K <Format>
0: DD.MM.YYYY hh:mm:ss
1: D.MM.YYYY h:mm:ss
2: D.MM.YYYY h:mm
3: D.MM.YY h:mm
4: YYYY-MM-DD- hh:mm:ss
5: YYYY-MM-DD- hh:mm
6: YYYY-MM-DD h:mm
7: YY-MM-DD- h:mm
8: DD.MM.YYYY
9: D.MM.YYYY
10: D.MM.YY
11: YYYY-MM-DD
12: YY-MM-DD
13: hh:mm:ss
14: h:mm:ss
15: h:mm
CM 9055

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	String was created
	1	Incorrect transfer value

7.10 Touch Probe

The following touch probes can be connected for workpiece measurement:

- TS 220: Touch-trigger probe with cable connection for workpiece setup and measuring during machining
- TS 440, TS 444, TS 640 and TS 740: Touch trigger probes with infrared transmission for workpiece setup and measurement during machining

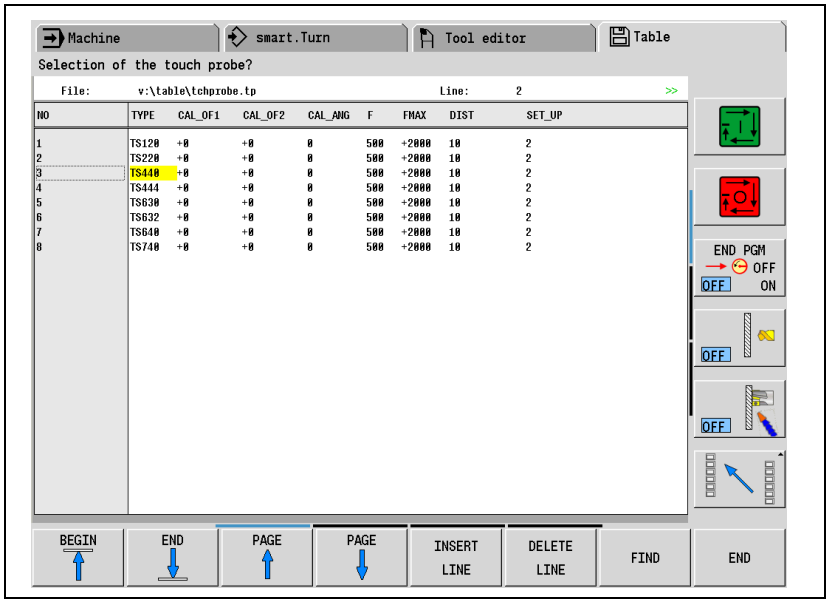
▶ Specify in the tool editor which touch probe is connected.

▶ Make sure that the spindle is locked during the measuring process.

Touch-probe table

In the touch-probe table, you configure the touch probe data. You can reach the touch-probe table through the file manager under **TNC:\table\tchprobe.tp**.

Enter the touch-probe data in the touch-probe table. All HEIDENHAIN touch probes are already preconfigured.



To make it possible to cover the widest possible range of applications, the touch-probe table offers numerous settings to enable you to determine the behavior common to all touch probe cycles:

Column	Description
NO	Number of the touch probe: Enter this number in the tool table (column: **TP_NO**) under the appropriate tool number
TYPE	Designation of the touch probes
CAL_OF1	Offset of the touch probe axis to the spindle axis for the reference axis
CAL_OF2	Offset of the touch probe axis to the spindle axis for the minor axis
F	Feed rate at which the control is to probe the workpiece.

Column	Description
FMAX	Feed rate at which the touch probe pre-positions, or is positioned between the measuring points
DIST	Maximum measuring range. If the stylus is not deflected within the defined path, the control outputs an error message.
SET_UP	Setup clearance for pre-positioning during probing cycles
F_PREPOS	Pre-positioning with speed from the column FMAX: FMAX_PROBE Pre-position at the machine's rapid traverse: FMAX_MACHINE
TRACK	Spindle orientation (the touch probe is oriented so that it is always probed by the same point on the touch-probe stylus tip)

Note

At this time, the lathe control evaluates only columns NO, TYPE and F of the touch probe table.

For more detailed information about tool measurement, please refer to the chapter "In-Process Measurement" of the User's Manual for your control.

Settings in the configuration editor	MP number
System	
ProbeSettings	
CfgToolMeasuring	
measuringType	604601
feed	604602
distance	604603
Axes	
Settings	
LinearAxis	
[Key name of the axis]	
CfgProbePosition	
positionProbePos	604701
positionProbeNeg	604702
maxMeasuringFeed	604703

There are two types of measurement available for measuring tools:

- Measurement with an optical gauge
- Measurement with a touch probe

Tools can only be measured in the **Machine** mode of operation. In the main menu, select the **Set T, S, F** dialog box and press the **Measure tool** soft key.

Measurement with an optical gauge

If you want to use an optical gauge for tool measurement, set **MP_measuringType** to **Optic**. Appropriate help graphics and soft keys for tool measurement by means of an optical gauge will then be displayed in the **Machine** mode of operation. Since the tool tip must be moved manually by the user to the cross hairs of the optical gauge, you do not need to enter any further settings in the machine parameters.

Measurement with a touch probe

If you want to use a touch probe for tool measurement, set **MP_measuringType** to **Probe**.

To measure a tool, position the tool in front of the touch probe and press the **NC start** key. The control then moves the tool automatically towards the touch probe and returns it to its initial position after measurement. A separate measurement is required for every axis direction. For more detailed information about tool measurement, please refer to the User's Manual of the control.

The expert program **_Measure.ncs**, which was tested in conjunction with a TT 140 tool touch probe from HEIDENHAIN, is available as a probing cycle. If touch probes other than the TT 140 are used, the expert program **_Measure.ncs** might need to be adapted by the OEM.

You configure the values for the measuring feed rate, the measuring range and the position of the touch probe as follows:

- In **MP_feed**, enter the value for the feed rate at which the tool is to approach the touch probe.
- In **MP_distance**, enter a value for the measuring range. The tool stops when it has traversed the measuring range without reaching the touch probe.
- Enter the position of the touch probe in **CfgProbePosition** for each individual axis. In **MP_positionProbePos**, define the position of the touch probe in the positive axis direction with respect to the machine datum, and in **MP_positionProbeNeg** in the negative axis direction.
- In addition, you can enter in **MP_maxMeasuringFeed** an axis-specific maximum feed rate for approaching the touch probe.

MP_measuringType

Type of tool measurement (optical gauge, touch probe)
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **None**
No tool measurement possible
Probe
Tool measurement with touch probe
Optic
Tool measurement with optical gauge
Default: -
Access: LEVEL3
Reaction: NOTHING

MP_feed

Feed rate for approaching the touch probe
Available from NCK software version: 597 110-04.

Format: Numerical value
Input: [mm/min]
Default: 0
Access: LEVEL1
Reaction: NOTHING

MP_distance

Measuring path
Available from NCK software version: 597 110-04.

Format: Numerical value
Input: [mm]
Default: 0
Access: LEVEL1
Reaction: NOTHING

MP_positionProbePos

Position of the touch probe in positive axis direction with respect to the machine datum
Available from NCK software version: 597 110-04.

Format: Numerical value
Input: [mm]
Default: 0
Access: LEVEL3
Reaction: RUN

MP_positionProbeNeg

Position of the touch probe in negative axis direction with respect to the machine datum
Available from NCK software version: 597 110-04.

Format: Numerical value
Input: [mm]
Default: 0
Access: LEVEL3
Reaction: RUN

MP_maxMeasuringFeed

Maximum permissible feed rate for approaching the touch probe
Available from NCK software version: 597 110-04.

Format: Numerical value
Input: [mm]
Default: 0
Access: LEVEL3
Reaction: RUN

7.11 Additional Parameters for Lathes

7.11.1 Coordinate system of the lathe

Settings in the configuration editor	MP number
System DisplaySettings CfgCoordSystem **coordSystem**	 114901

Specify in **MP_coordSystem** the coordinate system of the lathe. The support graphics and the position of the axes in the simulation are among the items influenced by the coordinate system.

MP_coordSystem

Definition of the coordinate system
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: See table below
Default: +X, +Z
Access: LEVEL3
Reaction: NOTHING

Meaning of the selection elements (the arrows point in the positive direction):

Selection	Coordinate system	
off or **+X, +Z**		Horizontal lathe; turning behind the center
–X, +Z		Horizontal lathe; turning in front of center
+Z, +X		Vertical lathe (vertical boring and turning mill); turning to the right of center

Settings in the configuration editor	MP number
Axes	
Settings	
LinearAxis	
[Key name of the spindle axis]	
CfgAxisProperties	
threadSafetyDist	300902
CfgProtectionZone	
limitPositive	301001
limitNegative	301002

If a G33 thread block is programmed without approach path, the control uses **MP_threadSafetyDist** as approach path. The approach path must suffice in order to attain the programmed feed rate.

MP_threadSafetyDist

Approach path for the thread start
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: Approach path in [mm] with up to 9 decimal places
Default: 3
Access: LEVEL3
Reaction: RESET

Protection zone monitoring

Settings in the configuration editor	MP number
Axes	
Settings	
LinearAxis	
[Key name of the spindle axis]	
CfgProtectionZone	
limitPositive	301001
limitNegative	301002
Aggregates	
CfgGlobalProperties	
protectionZoneOff	601805

Use the parameters **MP_limitPositive** and **MP_limitNegative** to specify the positive and negative values for the protection zone.

Set the **MP_protectionZone** parameter to TRUE in order to suppress protection-zone monitoring.

MP_limitPositive

Positive protection zone
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: Positive limit value for the protection zone in [mm] with up to 9 decimal places
If **MP_limitPositive** = 0 and **MP_limitNegative** = 0, then protection-zone monitoring is switched off.

Default: 999 999 999 [mm]

Access: LEVEL3

Reaction: RUN

MP_limitNegative

Negative protection zone
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: Negative limit value for the protection zone in [mm] with up to 9 decimal places
If **MP_limitPositive** = 0 and **MP_limitNegative** = 0, then protection-zone monitoring is switched off.

Default: -999 999 999 [mm]

Access: LEVEL3

Reaction: RUN

MP_protectionZone

Switch off protective zone monitoring
Available from NCK software version: 597 110-03.

Format: Selection menu

Selection: TRUE
Protective-zone monitoring inactive
FALSE
Protective-zone monitoring active

Default: TRUE

Access: LEVEL1

Reaction: RUN

7.11.3 Spindles

Settings in the configuration editor	MP number
Axes	
Settings	
SpindleAxis	
[Key name of the spindle axis]	
CfgMachineTable	
sysKinSimple	300703
basisTransKinSim	300704
sys	300701
basisTrans	300702
Aggregates	
General	
CfgGlobalProperties	
freezeVconst	601808

Subkinematics (as of NC SW 548 328-03)

In **MP_sysKinSimple**, enter the key name of the subkinematics for the tool spindle.

MP_sysKinSimple

Subkinematics of the tool spindle
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: Key names from Channels/Kinematics/CfgKinSimpleModel
Default: K_WP_S1
Access: LEVEL3
Reaction: RESET

The key name of the subkinematics for the machine base is to be entered in **MP_basisTransKinSim**.

MP_basisTransKinSim

Subkinematics of the machine base
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: Key names from Channels/Kinematics/CfgKinSimpleModel
Default: K_WP_S1_BASE
Access: LEVEL3
Reaction: RESET

Subkinematics (up to NC SW 548 328-02)

For main spindles you use the parameter **MP_sys** to specify to which machine-base system this spindle is assigned.

MP_sys

Assign the machine-base system to the spindle
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key names from Channels/Kinematics/CfgTrafoByDir
Default: KT_MT_S1
Access: LEVEL3
Reaction: RESET

In the **MP_basisTrans** parameter, specify the standard vector for datum shifts. The control shows a selection menu of all key names of the coordinate transformations available under CfgTrafoByDir.

MP_basisTrans

Standard vector for datum shifts
Available from NCK software version: 597 110-03.

Format: Selection menu
Selection: Key names from Channels/Kinematics/CfgTrafoByDir
Default: KT_MT_S1_BASISTRANS
Access: LEVEL3
Reaction: RESET

Freeze spindle speed for rapid traverse

With **MP_freezeVconst** you can prevent the spindle from changing its speed during constant surface speed Vconst according to the current diameter if there are several rapid traverse movements. The speed remains unchanged during the first rapid traverse movement after a feed rate and is not brought to the speed corresponding to the diameter at the target point until the last rapid traverse path before a feed rate path. This prevents unnecessary braking and acceleration of the spindle during several successive rapid traverse movements.

MP_freezeVconst

Freezing the spindle speed for rapid traverse movements and active constant surface speed.
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: **TRUE**
The spindle speed is held constant for rapid traverse movements and active constant surface speed.
FALSE
The spindle speed is always adjusted to the current diameter for rapid traverse movements and active constant surface speed.

Default: -

Access: LEVEL3

Reaction: RUN

Settings in the configuration editor	MP number
Axes	
Settings	
C axis	
[Key name of the C axis]	
CfgCAxisProperties	
blockBrake	300801
SpindlePrePosit	300802
relatedWpSpindle	300803
Aggregates	
General	
CfgAggregateKeys	
caxisKeys	600004

List all C axes of the machine in **MP_caxisKeys**. The control needs this information in order to distinguish C axes from the "C" rotary axis.

MP_caxisKeys

List of the key names of all C axes of the machine
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key names of all existing machine axes under PhysicalAxis.
Select the key names of the C axes and enter them here.
Default: –
Access: LEVEL3
Reaction: RESET

Enter in **MP_blockBrake** whether a shoe brake is present.

MP_blockBrake

Shoe brake present
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Shoe brake present
FALSE
Shoe brake not present
Default: FALSE
Access: LEVEL3
Reaction: RESET

Specify in the parameter **MP_spindlePrePosit** whether the spindle is to be pre-positioned upon M19. Enter the angle for pre-positioning in degrees [°].

MP_spindlePrePosit

	Spindle pre-positioning upon M19 Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0 to 99 [°] Angle in [°] to which the spindle is positioned before the C axis is positioned.
Default:	0 [°]
Access:	LEVEL3
Reaction:	RESET

In the optional parameter **MP_relatedWpSpindle** you specify the assigned workpiece spindle for C axes with separate drive.

MP_relatedWpSpindle

	Assigned workpiece spindle Available from NCK software version: 597 110-03.
Format:	Selection menu
Selection:	Key names of all existing machine axes under PhysicalAxis. Select the workpiece spindle assigned to the C axis. No entry = A workpiece spindle was assigned to the C axis through CfgProgAxis/relatedAxis.
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RUN

7.11.5 Tool carriers

The description of the tool carriers of a machine is set up as follows:

- **CfgAggregateKeys:**
  List of the tool carriers of the machine
- **CfgAssignAggregate:**
  Assign tool carrier to the machining channel
- **CfgTHDescription**:
  Description of the individual tool carriers
- **CfgToolMount:**
  Description of the tool holders of a tool carrier (See "Tool holders" on page 1031)

Settings in the configuration editor	MP number
Aggregates	
General	
CfgAggregateKeys	
toolHolderKeys	600001
ToolHolder	
[Key name of the tool carrier]	
CfgTHDescription	
ordinalNr	600203
type	600204
spindleNr	600207
maxSwivelPosition	600208
xDimToSlideRef	600210
zDimToSlideRef	600211
yDimToSlideRef	600212
CfgToolMountKeys	
toolMountKeys	600101
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgAssignAggregate	
assignToolHolder	203901

List in **MP_toolHolderKeys** all tool carriers of the machine. The control supports up to six tool carriers per machine.

MP_toolHolderKeys

List with the key names of all tool carriers of the machine
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key names of the tool carriers from the ToolHolder folder
Default: –
Access: LEVEL3
Reaction: RESET

Use **MP_assignToolHolder** to assign the tool carrier(s) to a machining channel. The control supports up to three tool carriers per machining channel.

MP_assignToolHolder

List with the key names of the tool carriers of this machining channel
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key names of the tool carriers from the ToolHolder folder
Default: –
Access: LEVEL3
Reaction: RESET

The following parameters describe the type of the tool carrier and its geometric position. In addition, the assigned spindle and the coolant circuits are declared if driven tools are intended.

MP_ordinalNr

Number of the tool carrier
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to 9
Default: 1
Access: LEVEL3
Reaction: RESET

MP_type

Type of tool carrier
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **None**
Turret
Multifix
(only one tool mount)
Default: MultiFix
Access: LEVEL3
Reaction: RESET

In the parameter **MP_spindleNr** you define the spindle for driven tools. During generation of the NC program, this parameter for G and M functions for driven tools is evaluated.

MP_spindleNr

(Code of the) spindle for driven tool
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **NoSpindle**
No driven tool
Spindle1
Spindle 1 for driven tool
Spindle2
Spindle 2 for driven tool
Spindle3
Spindle 3 for driven tool
Default: Spindle2
Access: LEVEL3
Reaction: RESET

For tool turrets, enter the number of swivel positions in **MP_maxSwivelPosition.** If a simple tool mount is used (example: multifix), enter "1"

MP_maxSwivelPosition

Number of turret swivel positions
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 1 to 99
Turret: Enter number of swivel positions (2 to 99)
Multifix: Enter 1
Default: 32
Access: LEVEL3
Reaction: RESET

The tool holders of a turret can be located on the left, on the right, and/or on the front. Therefore, the number of tool holders can be a multiple of the number of swivel positions.

MP_xDimToslideRef

X dimension for slide reference
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 to 100 000.000 [mm]
Distance from slide reference point to tool carrier reference point in [mm].
If the slide reference and the tool carrier reference are identical, the dimension = 0.
Default: 888.88 [mm]
Access: LEVEL3
Reaction: RESET

MP_zDimToslideRef

Z dimension for slide reference
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.0000 to 100 000.0000 [mm]
Distance from slide reference point to tool carrier reference point in [mm].
If the slide reference and the tool carrier reference are identical, the dimension = 0.
Default: 0
Access: LEVEL3
Reaction: RESET

MP_yDimToslideRef

Y dimension for slide reference
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.0000 to 100 000.0000 [mm]
Distance from slide reference point to tool carrier reference point in [mm].
If the slide reference and the tool carrier reference are identical, the dimension = 0.
Default: 0
Access: LEVEL3
Reaction: RESET

In **MP_toolMountKeys**, list all tool holders of the tool carrier. The tool holders are described in the **CfgToolMount** parameter object.

MP_toolMountKeys

List of the key names of all tool holders of this tool carrier
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: Key names from the ToolMount folder
Default: –
Access: LEVEL3
Reaction: RESET

7.11.6 Tool holders

The tool holders are assigned to a tool carrier in the **MP_toolMountKeys** parameter (See "Tool carriers" on page 1027).

The position and properties of each tool holder are described in **CfgToolMount.**

Settings in the configuration editor	MP number
Aggregates	
ToolMount	
[Key name of the tool holder]	
CfgToolMount	
mountPosWAPP	600401
freeTnr	600402
distCarrierRefX	600407
distCarrierRefZ	600408
distCarrierRefY	600409
correctionX	600410
correctionZ	600411
correctionY	600412
mirroringAxes	600416
convTblNr	600417

Define in **MP_mountPosWAPP** the number of the tool holder according to the "WAPP" principle. The **mountPosWAPP** attribute consists of the following parts:

- W = Number of the tool carrier (1 to 6)
- A = Holder number (0 to 3)
- PP = Swivel pocket (1 to 99)

MP_mountPosWAPP

	(W)APP designation of holder location Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	1 to 6999 Enter the number of this tool holder according to the WAPP principle. W = Tool carrier number (1 to 6) A = Holder position (0 to 3) PP = Swivel pocket of the turret
Default:	No value, parameter optional (initial value: 1001)
Access:	LEVEL3
Reaction:	RESET

Example: For holder 0 of swivel pocket 12 of tool carrier 1, the entry in **mountPosWAPP** is 1012.

Assign in **MP_freeTnr** a unique T number (1 to 899) to this tool holder. This T number is used in manual operation, in the NC program and in the turret assignment table (ToolAllo.tch) in order to activate the tool of this holder. Use the WAPP number if you do not define the T number.

The T number 0 is reserved by the control, and represents a tool whose geometric dimensions are 0. For reasons of clarity, the tools are normally numbered sequentially.

MP_freeTnr

	Free T number of the tool holder Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0: The free T number is not used 1 to 899: Each tool holder is assigned a unique number.
Default:	No value, parameter optional (initial value: 1)
Access:	LEVEL3
Reaction:	RESET

The tool holders are dimensioned relative to the tool carrier. The control adds the values from **MP_distCarrierRef*** and **MP_correction***.

MP_distCarrierRefX

	X dimension for tool carrier reference Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	–100 000.000 to 100 000.000 [mm] Enter the distance in millimeters [mm] from the tool holder datum to the tool carrier datum.
Default:	No value, parameter optional (initial value: 0)
Access:	LEVEL3
Reaction:	RESET

MP_distCarrierRefZ

	Z dimension for tool carrier reference Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	–100 000.000 to 100 000.000 [mm] Enter the distance in millimeters [mm] from the tool holder datum to the tool carrier datum.
Default:	No value, parameter optional (initial value: 0)
Access:	LEVEL3
Reaction:	RESET

MP_distCarrierRefY

Y dimension for tool carrier reference
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 to 100 000.000 [mm]
Enter the distance in millimeters [mm] from the tool holder datum to the tool carrier datum.
Default: No value, parameter optional (initial value: 0)
Access: LEVEL3
Reaction: RESET

MP_correctionX

Correction in X of the tool-holder position
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 to 100 000.000
Correction value in [mm]
Default: 0
Access: LEVEL3
Reaction: RESET

MP_correctionZ

Correction in Z of the tool-holder position
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 to 100 000.000
Correction value in [mm]
Default: 0
Access: LEVEL3
Reaction: RESET

MP_correctionY

Correction in Y of the tool-holder position
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 to 100 000.000
Correction value in [mm]
Default: 0
Access: LEVEL3
Reaction: RESET

Working with two tool carriers

The MANUALplus mirrors dimensions and converts operations if your machine is equipped with **two tool carriers** (example: one tool carrier in front of the workpiece, the other behind it).

The kinematics are based on the "standard tool carrier." You also define the **coordinate system** in **MP_worldCoorSystem** based on the standard tool carrier.

When entering the dimensions of tools for the "additional tool carrier," enter them as if this tool carrier were arranged in the "standard quadrant."

You program operations with the "additional tool carrier" as if it were arranged in the "standard quadrant." Because of the **MP_mirroringAxes** parameter, the MANUALplus knows the position of the tools. It therefore starts mirroring, and because of the **MP_convTblNr** parameter, converting, if the "additional tool carrier" is used.

Define in **MP_mirroringAxes** which axes, if any, are to be mirrored for this tool holder. If mirroring is not necessary, then the entry is omitted. The tool dimensions and traverse paths are mirrored.

MP_mirroringAxes

	Axes to be mirrored for this tool holder Available from NCK software version: 597 110-01.
Format:	Selection menu
Selection:	Key name of the axes to be mirrored
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

Define in **MP_convTblNr** the number of the conversion table to be activated when this tool holder is inserted (for conversions: See "Conversions" on page 1038).

MP_convTblNr

	Number of the conversion table to be activated Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0: Do not perform any conversion 1: Activate conversion table 1 (file: conv1.hc) 2: Activate conversion table 2 (file: conv2.hc) 3: Activate conversion table 3 (file: conv3.hc)
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcSymName	
stoppingAngle	116101
maxSpeedSpindle	116102
dbLoadDisplay	116103
readTsfData	116104
displayMode	116105
setToolPlace	116106

Transferring spindle data to the PLC

The two machine parameters **MP_stoppingAngle** and **MP_maxSpeedSpindle** are used to transfer information about the spindle to the PLC program. Use the dialog "Set T, S, F" to enter values for the stopping angle [°] and the maximum spindle speed [1/min]. The PLC can manipulate the two values once the dialog is closed.

MP_stoppingAngle

PLC operand for transfer of the spindle stopping angle
Available from NCK software version: 597110-01.

Format: String
Input: Max. 500 characters
Enter the name of a PLC operand (double word) with which the stopping angle of the spindle is transferred to the PLC.
Default: For HEIDENHAIN basic PLC program:
NP_DG_Transfer_Stopping_Angle
Access: LEVEL1
Reaction: NOTHING

MP_maxSpeedSpindle

PLC operand for transfer of the maximum spindle speed
Available from NCK software version: 597110-01.

Format: String
Input: Max. 500 characters
Enter the name of a PLC operand (double word) with which the maximum spindle speed is transferred to the PLC.
Default: For HEIDENHAIN basic PLC program:
NP_DG_Transfer_Speed_Limit
Access: LEVEL1
Reaction: NOTHING

Transferring data from the tool editor to the PLC

In the tool editor an integer value can be assigned to each tool via the "PLC" input field. The associated dialog opens after the **EDIT** soft key has been pressed, and is located on the second page of the dialog window. The PLC can read the entered values directly from the tool table. This makes it possible for the PLC to group the tools according to certain criteria, and manage them separately.

Load display for analog drives

The following attribute evaluation was added to the "LoadDisplay" dashboard element in order to be able to assign the motor data to a load display when analog axis drives and spindle drives are used.

- Bit 0 = 0 Display IPO data (as previously)
- Bit 0 = 1 Display PLC data (with analog drives)

In **MP_dbLoadDisplay**, enter the PLC program marker whose value is to be displayed in the dashboard by the load display.
This marker (called **PN_DG_LoadDisplay_Transfer**, for example) is defined as integer array [MAX_AXIS] in the PLC program.

PN_DG_Uebergabe_Auslastanzeige	Integer axis [MAX_AXIS]
Array with max. number of logical axes	(const long MAX_AXIS = 10)

The axis-specific integer values are displayed on a one-to-one basis (e.g. if the value is 55, "55%" is displayed). The maximum display value is 999%.

MP_dbLoadDisplay

PLC operand for dashboard load display
Available from NCK software version: 597 110-04.

Format: String
Input: Max. 500 characters
Enter the name of a PLC operand (double word) with which the load value of the respective axis or spindle is transferred to the load display of the dashboard.
Default: For HEIDENHAIN basic PLC program:
PN_DG_Transfer_utilization_display
Access: LEVEL3
Reaction: NOTHING

Transferring the spindle speed and feed rate data to the PLC

With the dialog "**Set T, S, F**" you can choose between constant surface speed [m/min] or constant rotational speed [1/min] for the spindle speed. For the feed rate you can choose between feed per revolution [mm/1] or feed per minute [mm/min]. The input values of these parameters are read by the PLC from the tables **ch_tsf.mch** and **sp_tsf.msp**, if **MP_readTsfData = TRUE** is set.

MP_readTsfData

PLC reads data from "Set T, S, F" dialog
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **TRUE**
The data on feed rate and spindle speed in the **Set T, S, F** dialog is read by the PLC from the **ch_tsf.mch** and **sp_tsf.msp** tables.
FALSE
The control always starts with feed per revolution and constant cutting speed.
Default:
Access: LEVEL3
Reaction: NOTHING

Transfer display mode to PLC

In **MP_displayMode**, enter a symbolic variable name in order to transfer the active display mode of the machine display (e.g. "manual operation" = 0 or "automatic mode" = 1) to the PLC. This makes it possible in the **Program Run** operating mode, for example, to switch the machine display ("Dashboard") to automatic mode already before activating cycle start.

MP_displayMode

Transfer display mode to PLC
Available from NCK software version: 597 110-04.

Format:	String
Input:	Max. 500 characters Enter the name of a PLC operand (double word) with which the current display mode is transferred to the PLC.
Default:	-
Access:	LEVEL3
Reaction:	NOTHING

Tool pocket preset by the PLC

With the machine parameter **MP_setToolPlace** under CfgPlcSymName, a symbolic variable name can be defined under which the PLC can preset a tool pocket. The user interface evaluates the request, inserts the tool in the tool pocket and displays the T number. As an acknowledgment, the defined tool is overwritten by the NC in the variable with (0). Then, as with a normal tool change, the tool call strobe is set and the requested T number is transferred. If the tool change cannot be performed, a (–1) is entered. At present the tool preset by the PLC is permitted only in machine mode.

MP_setToolPlace

Tool pocket preset by the PLC
Available from NCK software version: 597 110-04.

Format:	String
Input:	Max. 500 characters Symbolic variable name under which the PLC can define a tool pocket, which is then inserted by the user interface and displayed.
Default:	-
Access:	LEVEL3
Reaction:	NOTHING

7.11.8 Conversions

Conversions are required in order to mirror the working space. Example: Reversing the direction of rotation for circular arcs during machining in front of or behind the turning center, or when machining the rear face.

If the conversion table is active, the interpreter replaces a function listed in G_ORG/M_ORG with the corresponding function from G_CONV/ M_CONV.

The conversion lists are stored in files. They are configured by HEIDENHAIN. Conversion list 1 has the following structure:

NR	G_ORG	G_CONV	M_ORG	M_CONV
0	2	3	3	4
1	3	2	4	3
2	12	13	203	3
3	13	12	303	103
4	297	97	0	0
5	397	197	0	0

The conversion can only be used for DIN programs (not for cycle programs).

7.11.9 Global settings

Tool life management

Settings in the configuration editor	MP number
Aggregates General CfgGlobalProperties **lifeTime**	 601801

Use **MP_lifeTime** to switch tool-life monitoring for tool service age or workpiece quantity on or off.

MP_lifeTime

Activate/deactivate tool life monitoring
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **ON**
Monitoring on
OFF
Monitoring off
Default: ON
Access: LEVEL1
Reaction: RUN

Settings in the configuration editor	MP number
System	
DisplaySettings	
CfgGlbDispSettings	
plcSpindleSelect	604801
plcChannelSelect	604802

In order to define the spindle (e.g. spindle or driven tool) to be assigned the TSF dialog, the spindle can be selected by the PLC.
You can activate the spindle selection with **MP_plcSpindleSelect**. The selected spindle is then entered in a non-editable input field in the TSF dialog. When the spindle data is saved, it is assigned only to the selected spindle.

Spindle selection is implemented in the PLC program. The machine manufacturer defines whether a machine key or a soft key is used for switchover.

Channel selection using **MP_plcChannelSelect** has been prepared, but it currently has no function.

MP_plcSpindleSelect

Selection of spindle number by PLC
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: **TRUE**
Spindle-dependent input (speed, direction of rotation, etc.) refers to the spindle number selected by the PLC.
FALSE
Input refers to the spindle assigned to the tool carrier

Default: -

Access: LEVEL3

Reaction: NOTHING

MP_plcChannelSelect

Selection of channel number via PLC
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: **TRUE**
Channel-dependent input refers to the channel number selected by the PLC
FALSE
Input is possible for only one channel

Default: -

Access: LEVEL3

Reaction: NOTHING

To select the spindle and channel display

When two or more channels are supported, with Module 9480 you can control which channel provides the data to be displayed in the user interface.

With Module 9482 you can control the display of spindle data when two or more spindles are supported.

Module 9480 Select the channel display

On controls that support multiple channels, this module makes it possible to control the display of channel data. In addition, The user can enter information (e.g. through dialog menus) for a selected channel. The module selects the program channel for the selected operating panel. There is no explicit response time by which this request is accepted by the operating panel. The execution of this module can be monitored with Module 9481.

Call:

PS	B/W/D/K	<Operating panel> 0: At present only one user interface possible
PS	B/W/D/K	<Action> Bit 0: Display of channel data Bit 1: Entry of channel information
PS	B/W/D/K	<Channel index> 0: First channel 1: Second channel etc.
CM	9480	
PL	D	<Error code> 0 : New channel display selected 1: Addressed operating panel not in permitted range 2 : Addressed channel not in permitted range 3: Action not valid 4: Operating panel not designed for switchable channel display 5: Module not executed, as switchover already active for operating panel

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error (M4203)	0	No error
	1	See error code above

Module 9481 Find the channel display
The module provides the active and the selected channel display of an operating panel.

Call:

PS	B/W/D/K	<Operating panel> 0: At present only one user interface possible
CM	9481	
PL	D	<Index of the active channel used by the user interface> -1: No active channel present
PL	D	<Index of the selected channel that was preset by the PLC> -1: No selected channel present

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error (M4203)	0	No error
	1	Addressed operating panel not in permitted range
	2	Addressed operating panel not designed for switchable channel display

Module 9482 Select the spindle display
On controls that support multiple channels, this module makes it possible to control the display of spindle data. In addition, user entries (e.g. through dialog menus) for a selected spindle can also be enabled. The module selects the program spindle for the selected operating panel. There is no explicit response time by which this request is accepted by the operating panel. The execution of this module can be monitored with Module 9483.

Call:

PS	B/W/D/K	<Operating panel> 0: At present only one user interface possible
PS	B/W/D/K	<Action> Bit 0: Display of spindle data Bit 1: Entry of spindle information
PS	B/W/D/K	<Logical spindle number>
CM	9482	
PL	D	<Error code> 0: New spindle display selected 1: Addressed operating panel not in permitted range 2: Addressed spindle not in permitted range 3: Action not valid 4: Operating panel not designed for switchable channel display 5: Module not executed, as switchover already active for operating panel

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error (M4203)	0	No error
	1	See error code above

Module 9483 Find the spindle display
The module provides the active and the selected spindle display of an operating panel.

Call:

PS	B/W/D/K	<Operating panel> 0: At present only one user interface possible
CM	9483	
PL	D	<Logical spindle number actively used by the user interface> -1: No active spindle available
PL	D	<Logical spindle selected by the PLC> -1: No selected spindle available

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error (M4203)	0	No error
	1	Addressed operating panel not in permitted range
	2	Addressed operating panel not designed for switchable spindle display

Interpreter stop upon tool change

Settings in the configuration editor	MP number
Aggregates General CfgGlobalProperties **iStopT**	601802

The interpreter pre-interprets approx. 15 to 20 NC blocks. If variables are used for programming, then it must be ensured that the variable is assigned before the NC block is interpreted. This can be achieved with an **interpreter stop,** which interrupts the pre-interpretation.

Use the **MP_iStopT** parameter to specify whether an interpreter stop is triggered before the tool change.

MP_iStopT

	Interpreter stop upon tool change Available from NCK software version: 597 110-01.
Format:	Selection menu
Selection:	**TRUE** Interpreter stop before tool change **FALSE** No interpreter stop
Default:	FALSE
Access:	LEVEL3
Reaction:	RUN

Example for tool life management: During a tool change, the control checks the tool life of the tool. If "Interpreter stop at tool change" is active, the pre-interpretation is stopped and the actual usage time of the tool is taken into account. Without an interpreter stop, it is possible for some NC blocks to have been pre-interpreted but not yet performed. The usage time of the tool for these blocks is not taken into account during the tool change.

Tapping

Settings in the configuration editor	MP number
Aggregates General CfgGlobalProperties **threadDwell**	 601803

Use **MP_threadDwell** to halt the NC program for several seconds during tapping. This function is usually used during the setup procedure.

MP_threadDwell

	Dwell time for tapping Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	0 to 65535 [s] Dwell time in seconds [s] at the end point of the thread
Default:	0 [s]
Access:	LEVEL3
Reaction:	RUN

NC stop while tapping

Settings in the configuration editor	MP number
Aggregates General CfgGlobalProperties **threadLiftOff**	601804

Use **MP_threadLiftOff** to specify if a tapping process can be interrupted by an NC stop. During such an interruption, the threading tool retracts by the programmed distance.

MP_threadLiftOff

Liftoff out of the thread
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Upon NC stop in the thread the tool is lifted off by the programmed distance
FALSE
NC stop not permitted while in the thread.
Default: FALSE
Access: LEVEL3
Reaction: RUN

Activate handwheel in the thread

Settings in the configuration editor	MP number
Aggregates General CfgGlobalProperties **threadHandWheelOn**	601807

With **MP_threadHandWheelOn**, you activate the "Handwheel in thread" function, which makes it possible to compensate position and angular error of the linear and spindle axes. In **MP_threadHandWheelOn**, enter the value **TRUE**, to automatically insert the required G922 function in the thread cycles.

MP_threadHandWheelOn

Activation of the "handwheel in thread" function.
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: **TRUE**
The function G922 is automatically inserted in the thread cycles.
FALSE
The function G922 is not automatically inserted in the thread cycles.

Default: -

Access: LEVEL3

Reaction: RUN

7.11.10 Settings for cycles

Settings in the configuration editor	MP number
ProcessingData	
CfgGlobalTechPara	
safetyDistBlankOut	602005
safetyDistBlankIn	602006
safetyDistWorkpOut	602007
safetyDistWorkpIn	602008

You specify the safety clearances for cycle programming in the **CfgGlobalTechPara** parameter object. A distinction is made between machining of a workpiece blank and a prepared workpiece, as well as inside and outside machining. The control uses the global safety clearances if no safety clearances are defined in the cycle or DIN program.

MP_safetyDistBlankOut

Global safety clearance to the workpiece blank—outside
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0.000 to 100 000.000 [mm]
Default: 0
Access: LEVEL1
Reaction: RUN

MP_safetyDistBlankIn

Global safety clearance to the workpiece blank—inside
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0.000 to 100 000.000 [mm]
Default: 0
Access: LEVEL1
Reaction: RUN

MP_safetyDistWorkpOut

Global safety clearance to the prepared workpiece—outside
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0.000 to 100 000.000 [mm]
Default: 0
Access: LEVEL1
Reaction: RUN

MP_safetyDistWorkpIn

Global safety clearance to the prepared workpiece—inside
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0.000 to 100 000.000 [mm]
Default: 0
Access: LEVEL1
Reaction: RUN

Settings in the configuration editor	MP number
ProcessingData	
CfgGlobalTechPara	
DefaultG14	602009
DefaultCLT	602010
DefaultG60	602011
DefGlobG47P	602012
DefGlobG147SCT	602013
DefGlobG147SCK	602014
DefGlobOverMeasI	602015
DefGlobOverMeasK	602016

You specify the global settings for programming with smart.Turn in the **CfgGlobalTechPara** parameter object. The parameters defined here are used in the start unit. The global settings of the start unit are the default values for all further units.

MP_DefaultG14

Settings for "Tool change point GWW" in the start unit
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: **–1: No axis**
Default setting: Do not generate a G14
0: Simultaneously
Default setting: G14 Q0
1: First X, then Z
Default setting: G14 Q1
2: First Z, then X
Default setting: G14 Q2
3: Only X
Default setting: G14 Q3
4: Only Z
Default setting: G14 Q4

Default: 0

Access: LEVEL1

Reaction: RUN

MP_DefaultCLT

Settings for "Coolant CLT" in the start unit
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: **0: Without**
Default setting: Do not activate a coolant circuit
1: Circuit 1 on
Default setting: Activate coolant circuit 1
2: Circuit 2 on
Default setting: Activate coolant circuit 2

Default: 1

Access: LEVEL1

Reaction: RUN

MP_DefaultG60

Settings for "protective zone G60" in the start unit
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: 0: Active
Default setting: G60 Q0
1: Inactive
Default setting: G60 Q1

Default: 0

Access: LEVEL1

Reaction: RUN

MP_DefGlobG47P

Global "safety clearance G47" for the start unit. The "G47 P.." of the machining unit is generated with this safety clearance.
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: Safety clearance in [mm] with up to 9 decimal places

Default: 2 [mm]

Access: LEVEL1

Reaction: RUN

MP_DefGlobG147SCI

Global "Safety clearance plane SCI" for the start unit. The "I" of the "G147 I.. K.." call of the machining unit is generated with this safety clearance.
Available from NCK software version: 597 110-03.

Format: Numerical value

Input: Safety clearance in [mm] with up to 9 decimal places

Default: 2 [mm]

Access: LEVEL1

Reaction: RUN

MP_DefGlobG147SCK

Global "Safety clearance infeed direction SCK" for the start unit. The "K" of the "G147 I.. K.." call of the machining unit is generated with this safety clearance.
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: Safety clearance in [mm] with up to 9 decimal places
Default: 2 [mm]
Access: LEVEL1
Reaction: RUN

MP_DefGlobOverMeasI

Global "Oversize I (X direction)" for the start unit.This safety clearance is used for "Oversize I" during generation of the cycles for turning.
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: Oversize in [mm] with up to 9 decimal places
Default: 2 [mm]
Access: LEVEL1
Reaction: RUN

MP_DefGlobOverMeasK

Global "Oversize K (Z direction)" for the start unit.This safety clearance is used for "Oversize K" during generation of the cycles for turning.
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: Oversize in [mm] with up to 9 decimal places
Default: 2 [mm]
Access: LEVEL1
Reaction: RUN

Settings in the configuration editor	MP number
System	
Simulation	
CfgSimGeneral	
restartAtM99	114801
pathDelay	114802

If an NC program ends with M99, the control uses **MP_restartAtM99** to check whether the simulation of the NC program should be repeated. Application example: Continuous simulation for exhibitions, etc.

MP_restartAtM99

M99 causes a restart of the NC program
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **On**
If M99, then the NC program is simulated again.
Off
If M99, then the NC program is not simulated again.
Default: Off
Access: LEVEL1
Reaction: NOTHING

MP_pathDelay allows you to influence the speed of the simulation. After output of a traversed distance, the control waits for the time entered in "path delay."

MP_pathDelay

Path delay
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: Enter the delay [s] (smallest unit: 10 ms).
Default: 0
Access: LEVEL1
Reaction: NOTHING

Settings in the configuration editor	MP number
System	
Simulation	
CfgTimeDetGeneral	
toolChangeTime	115001
gearShiftingTime	115002
mFunTimeAllow	115003
ProcessingTime	
CfgmFunKeys	
mFunTimeKeys	115401
CfgTimeDetMfun	
[Key name of the M time allowance]	
mFun	115101
timeAllow	115102

The time calculation function of the simulation calculates the non-productive times on the basis of the parameters entered here. The time entered in **MP_mFunTimeAllow** is used for all M functions.

You can enter additional time allowances for specific M functions in the **ProcessingTime** parameter object.

MP_toolChangeTime

Time allowance for tool change
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 to 10 000 000.000
Time for the tool change in seconds [s].
Default: 0 [s]
Access: LEVEL1
Reaction: NOTHING

MP_gearShiftingTime

Time allowance for gear shifting
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 to 10 000 000.000
Time for gear shifting in [s].
Default: 0 [s]
Access: LEVEL1
Reaction: NOTHING

MP_mFunTimeAllow

General time allowance for M function
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 to 10 000 000.000
Time for performing M functions in [s].
Default: 0 [s]
Access: LEVEL1
Reaction: NOTHING

Proceed as follows to assign individual time allowances to M functions:

- Create a new key name for the M function's time allowance under **CfgTimeDetMfun**.
- Declare the M function in **MP_mFun** and the individual time allowance in **MP_timeAllow.** The simulation adds this individual time allowance to the time allowance from **MP_mFunTimeAllow**
- Enter the new key name for the time allowance to the list under **CfgmFunKeys/mFunTimeKeys.** The new key name is automatically offered at the very end of the list in the selection menu.

MP_mFunTimeKeys

Key names for M functions with specific time allowances
Format: Array [0...29]
Input: Key names for M functions with specific time allowances.
Default: –
Access: LEVEL1
Reaction: NOTHING

MP_mFun

M function with specific time allowance
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to 999
Number of the M function with specific time allowance.
Default: 0
Access: LEVEL1
Reaction: NOTHING

MP_timeAllow

Operating time of the M function
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 to 10 000 000.000 [s]
Operating time of the M function in seconds [s] for the simulation's time calculation.
Default: 0 [s]
Access: LEVEL1
Reaction: NOTHING

Simulation window, standard workpiece blank

Settings in the configuration editor	MP number
System	
Simulation	
CfgSimWindowSize	
zeroPosX	115201
zeroPosZ	115202
deltaX	115203
deltaZ	115204
CfgSimBlank	
outsideDiameter	115301
blankLength	115302
rightBlankEdge	115303
insideDiameter	115304

If no **workpiece blank** is programmed, the control works with the standard window size from the **CfgSimWindowSize** parameter object and the standard workpiece blank from the **CfgSimBlank** parameter object.

MP_zeroPosX

Zero position in X
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 to +100 000.000
Distance of the coordinate origin referenced to the lower window in [mm].
Default: -100 [mm]
Access: LEVEL1
Reaction: NOTHING

MP_zeroPosZ

Zero position in X
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: –100 000.000 to +100 000.000
Distance of the coordinate origin referenced to the left window in [mm].
Default: -150 [mm]
Access: LEVEL1
Reaction: NOTHING

MP_deltaX

Vertical expansion of the graphic window
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0.000 to 100 000.000
Vertical expansion of the graphic window in [mm].
Default: 200 [mm]
Access: LEVEL1
Reaction: NOTHING

MP_deltaZ

Horizontal expansion of the graphic window
Available from NCK software version: 597 110-01.
Format: Numerical value
Input: 0.000 to 100 000.000
Horizontal expansion of the graphic window in [mm].
Default: 200 [mm]
Access: LEVEL1
Reaction: NOTHING

MP_outsideDiameter

Outside diameter of the blank
Available from NCK software version: 597 110-01.
Format: Numerical value
Input: 0.000 to 100 000.000 [mm]
Outside diameter of the blank in [mm].
Default: 100 [mm]
Access: LEVEL1
Reaction: NOTHING

MP_blankLength

Workpiece blank length
Available from NCK software version: 597 110-01.
Format: Numerical value
Input: 0.000 to 100 000.000 [mm]
Total length of the blank in [mm].
Default: 80 [mm]
Access: LEVEL1
Reaction: NOTHING

MP_rightBlankEdge

Oversize of the workpiece blank
Available from NCK software version: 597 110-01.
Format: Numerical value
Input: –100 000.000 to +100 000.000
Oversize of the workpiece blank referenced to the workpiece datum in [mm].
Default: 0 [mm]
Access: LEVEL1
Reaction: NOTHING

MP_insideDiameter

Inside diameter of the blank
Available from NCK software version: 597 110-01.
Format: Numerical value
Input: For hollow cylinders: inside diameter [mm]
For solid workpieces: enter 0 [mm]
Default: 0 [mm]
Access: LEVEL1
Reaction: NOTHING

7.11.13 User parameters

Parameters that the operator can change without needing to enter a code number are called user parameters. These are used to:

- Set the unit of measure
- Set the behavior of the simulation
- Define global safety clearances
- Define global settings for the start unit of the smart.Turn editor
- Etc.

To select them in the **Organization** operating mode:

USER PARAMETER

▶ Press the soft key

Parameter settings

User parameters are a subgroup of the configuration parameters. They are described as part of the configuration parameters (see table).

User parameters	Config parameters	Page
Definition of unit of measure in effect for display	System/ DisplaySettings/ CfgUnitOfMeasure	879
General settings for automatic operation (switch tool-life monitoring off)	System/ CfgGlobalProperties	1039
Simulation – general settings	Simulation/ CfgSimGeneral	1050
Simulation – machining times for the NC functions in general	Simulation/ CfgTimeDetGeneral	1051
Simulation – operating times for M functions	Simulation/ CfgTimeDetMfun	1051
Simulation – specification of the (standard) window size	System/Simulation/ CfgSimWindowSize	1053
Simulation – specification of the (standard) blank size	System/Simulation/ CfgSimWindowSize	1053
General settings – safety clearances	ProcessingData/ CfgGlobalTechPara	1045
General settings – smart.Turn operating mode	ProcessingData/ CfgGlobalTechPara	1045

7.12 Configuration of the Lathe

The MANUALplus 620 is shipped with a modular standard configuration. You can adapt this standard configuration to your machine with very little effort. This chapter describes what configurations are possible and how they are set.

Based on the basic configuration, you can enter settings for the following components:

- Coordinate system of the lathe
- Linear axes
- Spindles, gear stages
- C axis, driven tool
- Y axis
- W axis
- Tool carriers

7.12.1 Coordinate system

Specify in **MP_coordSystem** the coordinate system of the lathe.

The MANUALplus 620 supports horizontal and vertical lathes. On horizontal lathes the tool carrier can be located in front of or behind the workpiece. The standard configuration describes a horizontal lathe with tool carrier behind the workpiece.

Settings in the configuration editor	MP number
System DisplaySettings CfgCoordSystem **coordSystem**	 114901

Selectable configurations:

+X, +Z Tool carrier behind the workpiece
+X, +Z Tool carrier in front of the workpiece
+Z, +X Vertical lathe: tool carrier to the right of the workpiece

7.12.2 Settings for linear axes

Counting direction of the axes

Check the counting directions of the actual and nominal values of the individual axes in the **MP_signCorrActualVal** and **MP_signCorrNominalVal** parameters.

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the axis]	
CfgAxisHardware	
signCorrActualVal	400001
signCorrNominalVal	400002

Reference positions of the axes

Then, starting from the positive quadrant, set the reference position of each axis via the **MP_refPosition** parameter.

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the axis]	
CfgReferencing	
refPosition	400403

Software limit switch

Set the software limit switches in the **MP_swLimitSwitchPos** and **MP_swLimitSwitchNeg** parameters.

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the axis]	
CfgPositionLimits	
swLimitSwitchPos	400501
swLimitSwitchNeg	400502

Traverse direction of the handwheels

For handwheels connected to the position inputs, check the traverse direction in the **MP_countDir** parameter.

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of the axis]	
CfgAxisHandwheel	
countDir	400202

Manual direction keys

Check the direction of motion of the manual direction keys in the **NP_MG_key_X_inv_direction** operand.

Settings in the configuration editor		MP number
System		
PLC		
CfgOemBool		
[NP_MG_key_X_inv_direction]		
value		
[0]:	**false**	104501.0
[NP_MG_key_Z_inv_direction]		
value		
[0]:	**false**	104501.0

7.12.3 Settings for spindles

Direction of spindle rotation

Check the direction of spindle rotation (M functions M3 and M4) in the **MP_signCorrActualVal**, **MP_signCorrNominalVal** and **MP_changeTurnDir** parameters (see "Direction of rotation for spindles with C axis" on page 434).

Settings in the configuration editor	MP number
Axes	
ParameterSets	
[Key name of spindle]	
CfgAxisHardware	
signCorrActualVal	400001
signCorrNominalVal	400002
CfgSpindle	
changeTurnDir	401509

Activating further gear stages

The standard configuration contains additional inactive parameters for further gear stages for the main spindle S1 **(PS1_1**, **PS1_2**, **PS1_3).** These parameter blocks were gated in **KeySynonym/CfgKeySynonym** to the parameter set for gear stage 0 **(PS1_0),** and are therefore identical as a default. You only have to change the parameters that differ in the individual gear stages. The parameter set **PS1_C1** was preconfigured for C-axis operation with spindle S1.

Make changes to the parameter sets in **Axes/ParameterSets.** The parameter sets describe the axis control response, the encoder connection, the encoder signals, etc.

Assign a parameter set to an axis by entering the key name of the parameter set in **MP_parList** (see "Assigning parameter sets" on page 408).

Settings in the configuration editor	MP number
Axes	
PhysicalAxis	
[Key name of the axis]	
CfgAxis	
parList	
[0]: PS1_0	Key.300107.0
[1]: PS1_1	Key.300107.1
...	

The switching between gear stages is controlled by the PLC. Transfer to the PLC a selection of the parameter sets listed in **MP_parList** by entering them in **MP_gearSpeed0.**

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcSStrobe	
[Key name of S strobe]	
parList	
gearSpeed0	
[0]: PS1_0	Key.104008.0
[1]: PS1_1	Key.104008.1
...	

The configuration object CfgFeedLimits of the respective parameter set defines the minimum and maximum spindle shaft speed for each gear stage. The list must be sorted in ascending order, with the smallest shaft speed at the top. Gear ranges are not supported if the list is missing or empty.

Configuration as external spindle

Settings in the configuration editor	MP number
Axes	
PhysicalAxis	
[Key name of the axis]	
CfgAxis	
axisHw	key.300104
axisMode	key.300105

In order to operate an externally driven tool (e.g.drilling machine via spindle S2) only by the PLC without an additional control loop, you can configure the driven tool as an external spindle.
To configure a spindle as an external spindle, enter the value **`None`** in **MP_axisHw**. You must also enter the value **`PlcControlled`** in **MP_axisMode**.

Note

For external spindles, you can only enter feed rate per minute (G94) and constant speed (G97) when entering the feed rate and rotational speed in the TSF menu.

7.12.4 Driven tool

A driven tool powered by spindle S2 is preconfigured in the standard configuration of the MANUALplus 620.

Configuration without driven tool

Remove spindle S2 from the configuration if your machine is operated without a driven tool.

- Remove spindle S2 from the **axisList** and **spindleIndices** lists of the **CfgAxes** entity.

Settings in the configuration editor	MP number
System	
CfgAxes	
axisList	
[0]: X1	100001.0
[1]: Z1	100001.1
[2]: S1	100001.2
[3]: S2	100001.3
spindleIndices	
[0]: S1	100002.0
[1]: S2	100002.1

- Remove kinematics that contain spindle S2 **(K1_CH1_S12_C1** and **K2_CH1_S12)**

When using the new kinematic model (as of NC SW 548 328-03):

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgKinList	
kinCompositeModels	
[0]: K1_CH1_S12_C1	Key.203001.0
[1]: K2_CH1_S12	Key.203001.1
[2]: K3_CH1_S1_C1	Key.203001.2
[3]: K4_CH1_S1	Key.203001.3

When using the old kinematic model (up to NC SW 548 328-02):

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannelAxes	
kinModels	
[0]: K1_CH1_S12_C1	Key.200306.0
[1]: K2_CH1_S12	Key.200306.1
[2]: K3_CH1_S1_C1	Key.200306.2
[3]: K4_CH1_S1	Key.200306.3

Configuration with driven tool

If you want to reinsert spindle S2, then you must undo the steps described in the section above.

Attention

When reinserting spindle S2 into your configuration, do not reverse the order of the steps described in the section above!

7.12.5 Settings for the C axis

In addition to a C axis driven by the tool spindle (spindle S1), the standard configuration of the MANUALplus 620 also has parameters for a separately driven C axis (spindle S4).

The standard configuration has a preconfiguration for a C axis driven by the tool spindle (spindle S1).

C axis driven via tool spindle

Removing the C axis: Make the following changes if your machine is to be operated without a C axis or with a separately driven C axis:

- Remove the C1 entry from the **CfgChannelAxes** entity in the **progAxis** list

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannelAxes	
progAxis	
[0]: X1	200301.0
[1]: Y1	200301.1
[2]: Z1	200301.2
[3]: C1	200301.3

- When using the new kinematic model (as of NC software level 548328-03), remove the associated C-axis kinematics (**K1_CH1_S12_C1** and **K3_CH1_S1_C1**) from the **CfgKinList** entity in the **kinCompositeModels** list.

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgKinList	
kinCompositeModels	
[0]: K1_CH1_S12_C1	203001.0
[1]: K2_CH1_S12	203001.1
[2]: K3_CH1_S1_C1	203001.2
[3]: K4_CH1_S1	203001.3

- When using the old kinematic model (up to NC software level 548328-02), remove the associated C-axis kinematics (**K1_CH1_S12_C1** and **K3_CH1_S1_C1**) from the **CfgChannelAxis** entity in the **kinModels** list.

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannelAxes	
kinModels	
[0]: K1_CH1_S12_C1	200306.0
[1]: K2_CH1_S12	200306.1
[2]: K3_CH1_S1_C1	200306.2
[3]: K4_CH1_S1	200306.3

- Remove the C1 entry from the **CfgAggregateKeys** entity in the **caxisKeys** list

Settings in the configuration editor	MP number
Aggregates	
General	
CfgAggregateKeys	
caxisKeys	
[0]: C1	600004.0

Inserting a C axis: In order to create a configuration with a C axis driven by the tool spindle, the steps described above must be undone.

Attention

When reinserting the C axis into your configuration, do not reverse the order of the steps described in the section above!

C axis with separate drive

The standard configuration features two spindles (tool spindle 1 and tool spindle 2). The C axis with separate drive is realized via spindle 4 (S4). Spindle S3 must be configured additionally to spindle S4. The parameters already exist in the axis data (PS3_0, PS4_0 and PS4_C1). Spindle S3 is marked as inactive in the axisMode parameter in the CfgAxis entity.

Make the following changes to active the C axis with separate drive via spindle S4:

- Insert spindles S3 and S4 in the **axisList** and **spindleIndices** lists of the **CfgAxes** entity.

Settings in the configuration editor	MP number
System	
CfgAxes	
axisList	
[0]: X1	100001.0
[1]: Z1	100001.1
[2]: S1	100001.2
[3]: S2	100001.3
[4]: S3	100001.4
[5]: S4	100001.5

Settings in the configuration editor	MP number
System	
CfgAxes	
spindleIndices	
[0]: S1	100002.0
[1]: S2	100002.1
[2]: S3	100002.2
[3]: S4	100002.3

- Assign the separate spindle (S4) to the C axis (C1) in **MP_relatedAxis**

Settings in the configuration editor	MP number
Axes	
CfgProgAxis	
[Key name of the C axis]	
relatedAxis: S4	Key.300005

- Assign the workpiece spindle (S1) to the C axis in **MP_relatedWpSpindle**

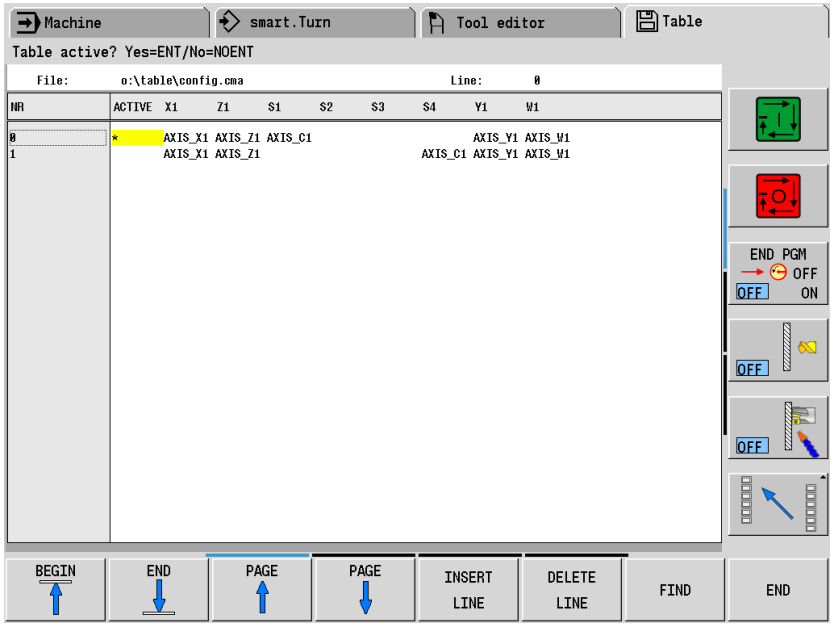
Settings in the configuration editor	MP number
Axes	
Settings	
C axis	
[Key name of the C axis]	
CfgCAxisProperties	
relatedWpSpindle: S1	Key.300803

- Deactivate spindle S3 with the parameter **MP_axisMode** in the entity **CfgAxis**

Settings in the configuration editor	MP number
Axes PhysicalAxis [Key name of the spindle axis] CfgAxis **axisMode: NotActive**	 Key.300105

Activate the appropriate axis compensation for spindle S4:

- Switch to the **Organization** mode of operation.
- Press the  soft key.
- Enter the code number **95148.**
- Press the **END** soft key.
- Shift the soft-key row to the right and press the **PGMMGT** soft key.
- On the **PLC:** drive, go to the **PLC:\table** directory and select the **config.cma** file.
- In the table that opens, activate the row in which the **AXIS_C1** entry for axis compensation is entered for spindle S4 instead of spindle S1. For a row to be active, the asterisk must be in the **ACTIVE** column at the beginning of the row.
- If the asterisk is not in the row for your configuration, use the arrow keys to move the cursor to the asterisk, and press the **TAB** key to remove it. Now use the cursor to mark the **ACTIVE** column of the desired row, and enter the asterisk by pressing the **ENTER** key.



7.12.6 Configuring the Y axis

In addition to the X and Z axes, you can optionally activate the Y axis that is perpendicular to these axes. If the Y axis is to be positioned at an angle not equal to 90° to the X or Z axis, the Y axis can also be configured as oblique axis.

The configuration data of the MANUALplus 620 contains preconfigured kinematics and axis data for the Y axis. In order to activate the Y axis, the configuration data must be modified as follows.

Activating the Y axis

Activate the Y axis: Make the following changes if your machine is to be operated with a Y axis:

- Remove the Y1 entry in the entity **CfgAxes** from the list **specCoordSysList**.

Settings in the configuration editor	MP number
System	
CfgAxes	
specCoordSysList	
[0]: Y1	100003.000

- Insert the entry Y1 in the list **axisList** of the entity **CfgAxes**.

Settings in the configuration editor	MP number
System	
CfgAxes	
axisList	
[0]: X1	100001.000
[1]: Z1	100001.001
[2]: Y1	100001.002
[3]: S1	100001.003
[4]: S2	100001.004

- Add the entry Y1 to the lists **refAxis** and **restoreAxis** of the entity **CfgChannelAxes** to define for the Y axis the reference sequence and the sequence for returning to the contour.

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannelAxes	
refAxis	
[0]: X1	200303.000
[1]: Y1	200303.001
[2]: Z1	200303.002
restoreAxis	
[0]: Z1	200305.000
[1]: Y1	200305.001
[2]: X1	200305.002

▶ To display the current position and distance-to-go of the Y axis in the dashboard, change the entries in the lists **opmodeStartup**, **opmodeReference**, **opmodeManual** and **opmodeAutomatic** of the entity **CfgChannDashboard** as follows.

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannDashboard	
opmodeStartup	
[0]: DB_STARTUP1_Y	203402.000
opmodeReference	
[0]: DB_REFER1_Y	203403.000
opmodeManual	
[0]: DB_MANUAL1_Y	203404.000
opmodeAutomatic	
[0]: DB_AUTO1_Y	203406.000

▶ Switch off the test mode of the Y axis by entering **MP_testMode** =FALSE. Please remember that the Y axis must be connected electrically if **MP_axisMode** = Active. In this case, you should also verify if a speed encoder input is assigned to the Y axis in **MP_speedEncoderInput**, and a nominal speed command output in **MP_pwmSignalOutput**.

Settings in the configuration editor	MP number
Axes	
PhysicalAxis	
[Y1]	
CfgAxis	
testMode	Y1.300106

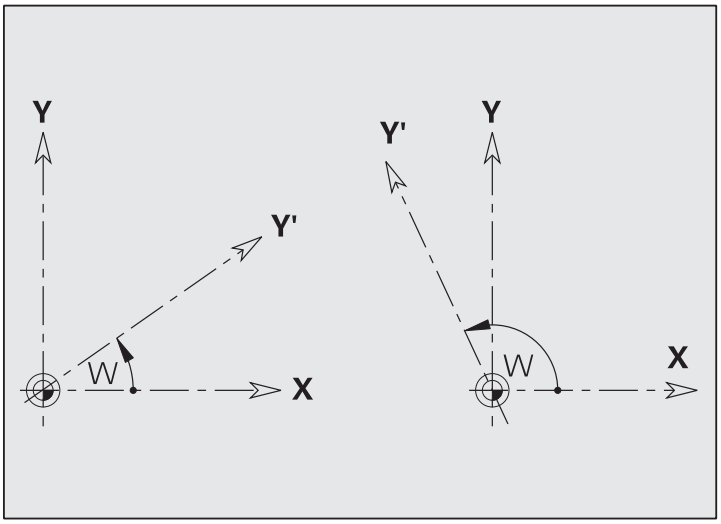
Y axis as oblique axis

The term **oblique axis** is used to describe an axis that is superposed on another axis, and is **not** perpendicular to this axis. "Superposed" means that the axis is connected to the base axis, and moves with the base axis.

The Y axis is usually used as an oblique axis. The following variants exist:

- Variant 1: The oblique axis Y' is superposed on the X axis, and is at an angle other than 90° to the X axis.
- Variant 2: The X axis is superposed on the oblique axis Y', and is at an angle other than 90° to the Y' axis.

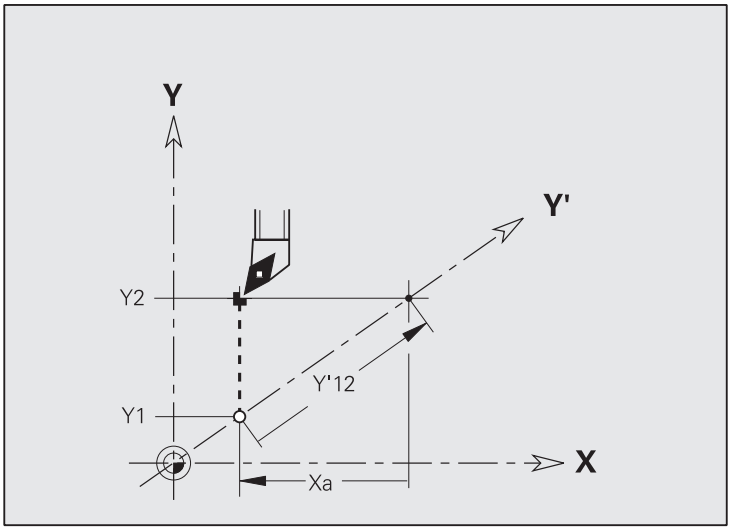
In both cases if the Y' axis is moved, then compensating movements must take place in the X axis.



Designations:

- **X:** X axis (coupling axis)
- **Y:** (virtual) Y axis
- **Y'**: Oblique axis
- **w:** Angle between coupling axis and oblique axis

Example:
The tool is traversed from position Y1 to Y2. The oblique axis moves along the path Y'12. At the same time, the X axis is moved along the path Xa in compensation (see the sketch).



In the kinematics of the Y axis, a rectangular coordinate system (w = 90°) was preconfigured, in which the Y axis is superposed on the X axis. If you want to design the Y axis as an oblique axis, you must correct the transformation of the Y axis (**KT_Y1**) according to the angle that is formed by the Y axis and the X axis.
If movement is in the positive Y direction, the compensating motion in the X axis is in the negative X direction for angles from 0° to 90°.

Settings for the new kinematic model (as of NC SW 548 328-03)

When using the new kinematic model, the following steps are necessary for activating the subkinematics for Y as oblique axis (for example C=30°).

- Delete subkinematics **K_XYZ_CH1** and rename **K_XYZ_CH1_Y30** to **K_XYZ_CH1**.

Settings in the configuration editor	MP number
Channels	
Kinematics	
CfgKinSimpleModel	
...	
K_XYZ_CH1	key.202800
...	
K_XYZ_CH1_Y30	key.202800

- For a configuration with turret and multifix, the mirrored subkinematics **K_XYZ_CH1_Y30_MIR** for the Y axis with oblique-axis coupling must be inserted instead of **K_XYZ_CH1_MIR** into the tool holder description (in the **kinModel** list).

Settings in the configuration editor	MP number
Aggregates ToolMount TH2_TM001_MULTI CfgToolMount kinModel **K_WP_S1_MIR_X** **K_XYZ_CH1_Y30_MIR**	 key.600419.000 key.600419.001

Settings for the old kinematic model (up to NC SW 548 328-02)

The following table illustrates the transformation settings required for angles of 30°, 45° and 60°. However, you can also specify any intermediate values for angle w.

w = 90° (preconfigured)	w = 30°	w = 45°	w = 60°	f(w)
Channels Kinematics CfgTrafoByDir KT_Y1 **zDir** **[0] : 1** **[1] : 0** **[2] : 0** **xDir** No change	 **0.866** **0** **0.5**	 **0.707** **0** **0.707**	 **0.5** **0** **0.866**	 **cos (w)** **sin (w)**
Channels Kinematics CfgTrafoByDir KT_TH1 **zDir** No change **xDir** **[0] : 0** **[1] : 1** **[2] : 0**	 **0** **0.866** **0.5**	 **0** **0.707** **0.707**	 **0** **0.5** **0.866**	 **cos (w)** **sin (w)**

Activating compensating movement in Manual mode

If the Y axis is designed as an oblique axis, the compensating motion of the axis coupled to the Y axis can also be activated in Manual mode of operation by using **MP_kinManualMode**.

Settings in the configuration editor	MP number
Channels ChannelSettings [CH_NC1] CfgChannelProperties **kinManualMode**	 203804

Reference dimensions and limit switches must be entered for each individual axis. The positions are then displayed in the rectangular coordinate system.

MP_kinManualMode

Switch kinematics off/on in Manual mode
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **On**
Kinematics active in Manual mode
Off
Kinematics not active in Manual mode
Default: -
Access: LEVEL3
Reaction: RUN

Reference run with the oblique axis

The oblique axis and coupling axis move independently of each other during reference run. When the oblique axis is traversed, the compensating motions only take place after all axes have determined the reference points.

Activating compensation value tables

Copies of the compensation value tables that were expanded for the Y axis are saved in the directory **tnc:\update\o\table** during the software update. These tables (axis_x1.com, axis_y1.com, ...) are not active.
In order to activate the compensation value tables expanded by HEIDENHAIN, use **TNCremoNT** to save the tables stored in **tnc:\table** and replace them with the tables contained in the directory **tnc:\update\o\table**. After this, previously existing compensation values must be updated accordingly.

Tool change with active Y axis

Settings in the configuration editor	MP number
Aggregates General CfgGlobalProperties **doProgAfterTCall**	 601806

Use **MP_doProgAfterTCall** to define whether a subprogram is to be run after a tool change so that the Y axis moves to position Y=0. The expert program **_tcall1.ncs** is used as subprogram by default (See "Expert programs" on page 1086).

MP_doProgAfterTCall

Run subprogram _tcall1.ncs after the tool change
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **TRUE**
The subprogram _tcall1.ncs is run after the tool change
FALSE
No subprogram is run after the tool change
Default: -
Access: LEVEL3
Reaction: RUN

Deactivating the Y axis

If you want to remove the Y axis from the configuration of your machine, you must follow the above-described procedure in reverse order.

7.12.7 Configuring the W axis

The W axis that is parallel to the Z axis can optionally be activated in addition to the Z axis. The W axis can be moved by the PLC, where the position value of the W axis is taken into account in the position value of the Z axis. Only the OEM-specific G codes (G600 to G699) can currently be used to program the W axis in the NC program.

The configuration data of the MANUALplus 620 contains preconfigured kinematics and axis data for the W axis. In order to activate the W axis, the configuration data must be modified as follows.

Activation of W axis

Make the following changes if your machine is to be operated with a W axis:

- Switch to the **Organization** mode of operation.
- Press the  soft key.
- Enter the code number **95148.**
- Press the **CONFIG DATA** soft key.
- Enter **W1** in the list **axisList** of the entity **CfgAxes**. For this, select the entry **Z1**, press the **MORE FUNCTIONS** soft key and then the **INSERT** soft key, and select **W1** in the pop-up window. Press the **OK** soft key to confirm your selection.

Settings in the configuration editor	MP number
System	
CfgAxes	
axisList	
[0]: X1	100001.000
[1]: Z1	100001.001
[2]: W1	100001.002
[3]: S1	100001.003
[4]: S2	100001.004

Settings for the new kinematic model (as of NC SW 548 328-03)

If you use the new kinematic model, replace the subkinematics **K_XYZ_CH1** with **K_XYZW_CH1** in all kinematic models (**K1_CH1_S12_C1** to **K4_CH1_S1**).

Settings in the configuration editor	MP number
Channels	
Kinematics	
CfgKinComposModel	
[Key name of the kinematics model]	
subKinList	
[0]: TOOL_TH1	Key.202901.000
[1]: K_XYZW_CH1	Key.202901.001
[2]: ...	Key.202901.002
[3]: ...	Key.202901.003

Settings for the old kinematic model (up to NC SW 548 328-02)

When using the old kinematic model, the existing kinematic models must be expanded by the W axis.

- In the entity **CfgKinModel**, you open the first kinematic model with the key name **K1_CH1_S12_C1**. Use the **INSERT** soft key to enter **W1** in the list **axesToolSide**. Please note that the sequence of the axes in this list must correspond to the design of your machine. Therefore, the W axis must usually be inserted at the beginning of the list.

Settings in the configuration editor	MP number
Channels	
Kinematics	
CfgKinModel	
[Key name of the kinematics model]	
axesToolSide	
[0]: W1	Key.200001.000
[1]: Z1	Key.200001.001
[2]: X1	Key.200001.002
[3]: Y1	Key.200001.003

Note

Please note that the sequence of the axes in the lists of the kinematic models must correspond to the actual design of your machine. The axes must be listed in the correct sequence starting from the machine base system to the tool.

- You also need to enter **KT_W1** in the list **trafoToolSide**. For this, select the entry **KT_Z1** and press the ENT key.
- Press the **COPY FIELD** soft key and then the **CANCEL** soft key.
- Move the cursor to the location where you want to insert the value and press the **INSERT** soft key.
- Then press the **PASTE FIELD** soft key to insert the entry **KT_Z1** in the input window.
- Change the designation **KT_Z1** to **KT_W1** by pressing the **GOTO** key to show the alphabetic keyboard.
- To confirm the change, press the **OK** soft key.
- Press the **OK** soft key again to enter the new value in the list.
- The entry **KT_W1** has been added to the dataFiles list as shown below.

Settings in the configuration editor	MP number
Channels	
Kinematics	
CfgKinModel	
[Key name of the kinematics model]	
trafoToolSide	
[0]: KT_W1	Key.200002.000
[1]: KT_Z1	Key.200002.001
[2]: KT_X1	Key.200002.002
[3]: KT_Y1	Key.200002.003

- You also need to enter **KT_W1** in the **trafoDirToolSide** list in the same manner.

Settings in the configuration editor	MP number
Channels	
Kinematics	
CfgKinModel	
[Key name of the kinematics model]	
trafoDirToolSide	
[0]: KT_W1	Key.200003.000
[1]: KT_Z1	Key.200003.001
[2]: KT_X1	Key.200003.002
[3]: KT_Y1	Key.200003.003

- Repeat the previous steps for all other kinematic models **K2_CH1_S12**, **K3_CH1_S1_C1** and **K4_CH1_S1** of the entity **CfgKinModel**.

Settings in the configuration editor	MP number
Channels	
Kinematics	
CfgKinModel	200000
K1_CH1_S12_C1	
K2_CH1_S12	
K3_CH1_S1_C1	
K4_CH1_S1	

In order to describe the coordinate transformations, you must define the direction vectors for the W axis. Since the W axis is parallel to the Z axis, you can copy the entry **KT_Z1** in the entity **CfgTrafoByDir**, change the designation to **KT_W1** and insert it again.

- Select the entry **KT_Z1** in the **CfgTrafoByDir** entity and press the **COPY** soft key.

- Enter the new key name **KT_W1** in the dialog window and select the file **PLC:\config\..\kin_ch1_s1.cfg** in the menu for selecting the memory file. Press the **OK** soft key to confirm your changes.

Settings in the configuration editor	MP number
Channels	
Kinematics	
CfgTrafoByDir	200100
KT_C1	key.200100
...	
KT_W1	key.200100
...	
KT_Z1	key.200100

General settings for the W axis

In the next step, you must include the W axis in the list of programmable axes.

- Select the entry **Z1** in the **progAxis** list of the **CfgChannelAxes** entity. Press the **MORE FUNCTIONS** soft key, then the **INSERT** soft key and select **W1** in the pop-up window. Press the **OK** soft key to confirm your selection.

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannelAxes	
progAxis	
[0]: X1	key.200301.000
[1]: Y1	key.200301.001
[2]: Z1	key.200301.002
[3]: W1	key.200301.003
[4]: C1	key.200301.004

- Add the entry W1 to the lists **refAxis** and **restoreAxis** of the entity **CfgChannelAxes** to define for the W axis the reference sequence and the sequence for returning to the contour.

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
[Key name of the machining channel]	
CfgChannelAxes	
refAxis	
[0]: Z1	200303.000
[1]: X1	200303.001
[2]: W1	200303.002
restoreAxis	
[0]: W1	200305.000
[1]: Z1	200305.001
[2]: X1	200305.002

- Verify that the value **`ParallelLinCoord`** is entered in **MP_progKind**.

Settings in the configuration editor	MP number
Axes	
CfgProgAxis	
[W1]	
progKind	300003

- In **MP_parAxComp**, enter the value **`Display`**. The position of the W axis is only taken into account for the principal axes in the three-axis kinematics starting with the subsequent NC block if the parameter Display is set. However, there is no compensating movement.

Settings in the configuration editor	MP number
Axes	
PhysicalAxis	
[W1]	
CfgAxisPropKin	
parAxComp	300205

Attention

If the W axis is used to move a parallel Z axis, the protection zone values for the Z axis need to be reset after every movement of the W axis.

In order to show the current position and distance-to-go for the W axis in the dashboard, supplement the **DB_AUTO1** list in the **CfgDashboard** entity as described below:

- Select the entries of the corresponding list items (7 and 11) and confirm each selection with the ENT key. Select the dashboard elements **DB_W1_POS** and **DB_W1_DELTA** in the selection menu.

Settings in the configuration editor	**MP number**
System	
DisplaySettings	
CfgDashboard	
DB_AUTO1	
[0]: DB_X1_POS	key.113201.000
...	
[7]: DB_W1_POS	key.113201.007
...	
[11]: DB_W1_DELTA	key.113201.011
...	

- Switch off the test mode of the W axis by entering **MP_testMode** =FALSE. Please remember that the W axis must be connected electrically if **MP_axisMode** = Active. In this case, you should also verify if a speed encoder input is assigned to the W axis in **MP_speedEncoderInput**, and a nominal speed command output in **MP_pwmSignalOutput**.

Settings in the configuration editor	**MP number**
Axes	
PhysicalAxis	
[W1]	
CfgAxis	
testMode	W1.300106

- Before exiting the configuration editor, save all the changes you have made and restart the control.

MP_parAxComp

Compensation of parallel linear axes
Available from NCK software version: 597 110-04.

Format: Selection menu

Selection: **Off**
The position of the axis is not taken into account for the axes in the 3-axis kinematics.
Display
The position of the axis is taken into account for the axes in the 3-axis kinematics (principal axes) with the subsequent NC block, but there is no compensating movement.
Move
The tool is kept in place by a simultaneous compensating movement of the axes in the 3-axis kinematics.

Default: -

Access: LEVEL3

Reaction: RESET

Activating compensation value tables

- Copies of the compensation value tables that were expanded for the W axis are saved in the directory **tnc:\update\o\table** during the software update. These tables (axis_x1.com, axis_w1.com, ...) are not active.
  In order to activate the compensation value tables expanded by HEIDENHAIN, use **TNCremoNT** to save the tables stored in **tnc:\table** and replace them with the tables contained in the directory **tnc:\update\o\table**. After this, previously existing compensation values must be updated accordingly.

Deactivating the W axis

If you want to remove the W axis from the configuration of your machine, you must follow the above-described procedure in reverse order.

7.12.8 Configuring the tool carrier

The following tool carriers are preconfigured in the basic configuration:

- **Multifix** quick change tool holder (factory default setting)
- **Tool turret**
- **Multifix and tool turret**

Depending on the three configurations named above, the following key names are assigned to the tool carriers in the control:

Machine configuration	Multifix	Tool turret	Multifix and tool turret
Multifix	TH1_MULTI	-	TH2_MULTI
Tool turret	-	TH1_REV	TH1_REV

Enter the settings described below in the configuration editor.

Multifix as tool carrier

The tool carrier TH1_MULTI and the tool holder TH1_TM001_MULTI are already configured in the standard configuration.

- In the two entities listed below, enter the key name **TH1_MULTI** for the multifix tool carrier.

Settings in the configuration editor	MP number
Channels ChannelSettings CH_NC1 CfgAssignAggregate assignToolHolder **[0]: TH1_MULTI**	203901.0
Aggregates General CfgAggregateKeys toolHolderKeys **[0]: TH1_MULTI**	600001.0

▶ Check the description, and add the dimensions of the tool carrier **TH1_MULTI** (See "Tool carriers" on page 1027) and the tool holder **TH1_TM001_MULTI** (See "Tool holders" on page 1031).

Settings in the configuration editor	MP number
Aggregates	
ToolHolder	
TH1_MULTI	
CfgTHDescription	600200
ordinalNr	600203
. . .	. . .
Aggregates	
ToolMount	
TH1_TM001_MULTI	
CfgToolMount	600400
mountPosWAPP	600401
. . .	. . .

Tool turret

The tool carrier TH1_REV is already configured in the standard configuration.

▶ In the two entities listed below, enter the key name **TH1_REV** for the tool turret.

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
CH_NC1	
CfgAssignAggregate	
assignToolHolder	
[0]: TH1_REV	203901.0
Aggregates	
General	
CfgAggregateKeys	
toolHolderKeys	
[0]: TH1_REV	600001.0

- Specify the number of swivel positions in **MP_maxSwivelPosition.**
- Check the description, and add the dimensions of the tool carrier **TH1_REV** (See "Tool carriers" on page 1027).

Settings in the configuration editor	MP number
Aggregates	
ToolHolder	
TH1_REV	
CfgTHDescription	600200
ordinalNr	600203
. . .	. . .
maxSwivelPosition	600208

In the standard configuration, three tool holders are configured for each of the 12 swivel pockets, meaning a total of 36 tool holders for the tool turret. A key name is assigned to each tool holder. The prepared key names include:

- the tool carrier (TH1)
- the number of the tool holder (TMx01 to TMx12)
- the number of the swivel pocket (TM0xx...TM2xx)
- the type of the tool carrier (MULTI or REV).

- Delete the key names of unused tool holders.

Settings in the configuration editor	MP number
Aggregates	
ToolHolder	
TH1_REV	
CfgToolMountKeys	
toolMountKeys	
[0]:TH1_TM001_REV	600101.0
...	...
[36]:TH1_TM212_REV	

The standard configuration includes 36 tool holders on the tool turret (TH1_TM001_REV .. TH1_TM212_REV).

▶ Check the descriptions, and add the dimensions of the **tool holders** used on your machine (See "Tool holders" on page 1031).

Settings in the configuration editor	MP number
Aggregates	
ToolMount	
TH1_TM001_REV	
CfgToolMount	600400
mountPosWAPP	600401
. . .	...
TH1_TM002_REV	
CfgToolMount	600400
mountPosWAPP	600401
. . .	...
. . .	

Multifix and tool turret

The tool carriers TH1_REV and TH2_MULTI are already configured in the standard configuration.

▶ In the two entities listed below, enter the key names **TH1_REV** for the tool turret and **TH2_MULTI** for the multifix tool carrier.

Settings in the configuration editor	MP number
Channels	
ChannelSettings	
CH_NC1	
CfgAssignAggregate	
assignToolHolder	
[0]: TH1_REV	203901.0
[0]: TH2_MULTI	203901.1
Aggregates	
General	
CfgAggregateKeys	
toolHolderKeys	
[0]: TH1_REV	600001.0
[0]: TH2_MULTI	600001.0

Note

For the turret-Multifix configuration it is assumed that the tool turret is in the positive quadrant as standard tool carrier.

▶ Specify the number of swivel pockets for the tool turret in **MP_maxSwivelPosition.**

▶ Check the description, and add the dimensions of the tool carriers **TH1_REV** and **TH2_MULTI** (See "Tool carriers" on page 1027).

Settings in the configuration editor	MP number
Aggregates	
ToolHolder	
TH1_REV	
CfgTHDescription	600200
ordinalNr	600203
. . .	. . .
maxSwivelPosition	600208
. . .	
TH2_MULTI	
CfgTHDescription	600200
ordinalNr	600203
. . .	. . .

The standard configuration includes 36 tool holders on the tool turret (TH1_TM001_REV .. TH1_TM212_REV). The key name TH2_TM001_MULTI is entered for the tool holder of the Multifix tool carrier.

- Delete the key names of unused tool holders of the tool turret.

Settings in the configuration editor	MP number
Aggregates	
ToolHolder	
TH1_REV	
CfgToolMountKeys	
toolMountKeys	
[0]:TH1_TM001_REV	600101.0
...	...
[36]:TH1_TM212_REV	
TH2_MULTI	
CfgToolMountKeys	
toolMountKeys	
[0]:TH2_TM001_MULTI	600101.0

The standard configuration includes 36 tool holders on the tool turret (TH1_TM001_REV .. TH1_TM212_REV) as well as the tool holder of the Multifix tool carrier (TH2_TM001_MULTI).

- Check the descriptions, and add the dimensions of the **tool holders** used on your machine of the tool turret and the Multifix tool carrier (See "Tool holders" on page 1031).

Settings in the configuration editor	MP number
Aggregates	
ToolMount	
TH1_TM001_REV	
CfgToolMount	600400
mountPosWAPP	600401
. . .	...
TH1_TM002_REV	
CfgToolMount	600400
mountPosWAPP	600401
. . .	...
. . .	
TH2_TM001_MULTI	
CfgToolMount	600400
mountPosWAPP	600401
. . .	...

Mirroring of the X axis is then activated for the Multifix with the **KT_MT_S1_MIRROR** and **KT_TH1_MIRROR** parameters. Additionally for C-axis operation, the C-axis position is rotated by 180° with parameter **MP_mirroringAxes = X1** (rotation of the C axis around Z). See "Working with two tool carriers"on page 1034.

Settings in the configuration editor	MP number
Aggregates	
ToolMount	
TH2_TM001_MULTI	
CfgToolMount	
coorTrafo	
[0]: KT_MT_S1_MIRROR	600415.0
[1]: KT_TH1_MIRROR	600415.1
mirroringAxes	
[0]: X1	600416.0

7.12.9 Expert programs

HEIDENHAIN provides numerous subprograms that realize useful functions for the operator. These "expert programs" (*.ncs) are used explicitly with M and G functions, or are internally called and performed with cycles. For example, the C axis can be switched on with the C-axis expert program via the M14 function, and switched off again via M15, without needing to make any changes to the control's configuration.

Expert programs (*.ncs)	Function
_c1_ein.ncs	Activate the C axis (M14)
_c1_aus.ncs	Deactivate the C axis (M15)
_G85.ncs	Undercut (G85)
_G87G88.ncs	Transition radius/chamfer (G87, G88)
_G500.ncs	Undercut type E according to EPL2/4 (G500)
_G501.ncs..._G599.ncs	Customer cycles
_g798.ncs	Helical slot milling (G798)
_g799.ncs	Thread milling with the C axis (G799)
_G800.ncs	Thread milling, lateral surface, with Y axis (G800)
_G806.ncs	Thread milling, front face, with Y axis (G806)
_CHAR.NCS	Engraving cycle: XC plane, front face (G801) ZC plane, lateral surface (G802) XY plane, front face, Y axis (G803) YZ plane, lateral surface, Y axis (G804)
_Measure.ncs	Touch probe cycle for tool measurement
_tcall1.ncs _tcall2.ncs	Positions the Y axis to Y=0 after every tool change if **MP_doProgAfterTCall = TRUE**. Identical to _tcall1.ncs, for a second slide.

The expert programs are based on the NC syntax, and can be edited with the help of a text editor or the smart.Turn editor. Since the programs are programmed using variables, they are written as generally as possible. For example, offsets are assigned to be able to also use the M14 function for configured subspindles. The main spindle is addressed with M014, the first subspindle with M114, the second subspindle with M214, etc.

For more information about the expert programs, refer to your control's User's Manual.

If a software update is performed on the **SYS:** drive of the control, the current C-axis programs are copied to the **TNC:\nc_prog\ncps** directory. In order to protect any existing, and possibly modified, expert programs, these are automatically backed up in the ncps directory with a modified file name (extension: *_save). The new files all contain the current version of the default settings managed by HEIDENHAIN.

For more information about software updates, See "NC software exchange on the MANUALplus 620" on page 94.

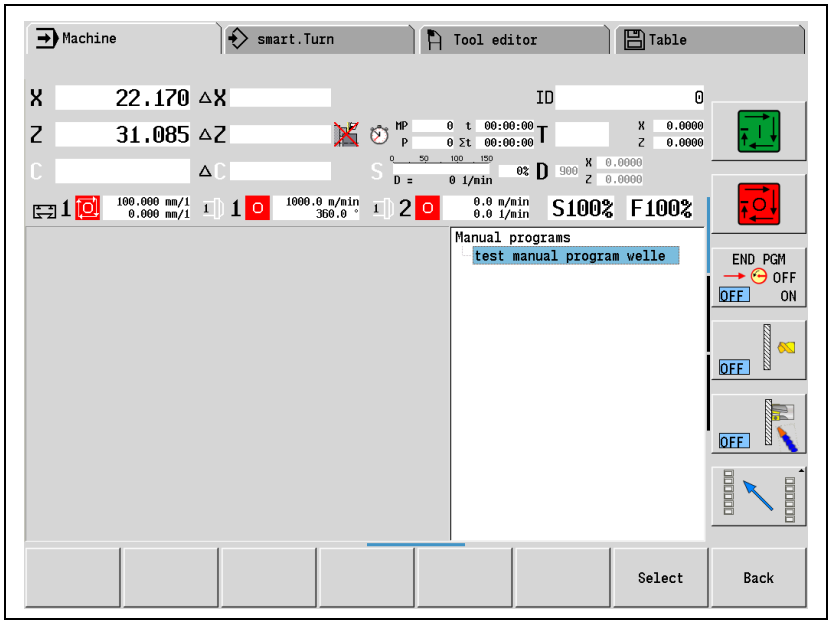
7.12.10 Manual programs

If the menu guidance of the user interface (3x3 menu) is expanded with **MP_extManualMode** (see "Expanded menu structure" on page 884), then in the **Machine** operating mode, under the **Manual program** menu item of the user interface, previously defined NC subprograms can be shown in a clearly structured list view, from which they can be opened.

In the file **PLC:\resource\ManualProg\ManualProg_en.str** you associate the program names shown in the list of manual programs with the NC subprograms stored under **TNC:\nc_prog\ncps**. Please note the following format for this:

"Sample program", "program name.nc"

In the file **ManualProg_en.str** you first enter in quotation marks any name for the program (this is how the NC subprogram will appear in the list of manual programs). Separated by a comma, you then enter in quotation marks the actual program name including the file extension .nc, as it is stored in the NCPS program folder.



8 PLC Programming

8.1 PLC Functions

The integrated PLC of the control contains its own text editor for creating the statement list for the PLC program. You enter PLC commands and comments using an optional USB keyboard. An even simpler way is to create your PLC program on a PC with the PLC development software **PLCdesignNT.** If you want to use **PLCdesignNT,** contact HEIDENHAIN.

The control supports you with the COMPILE function, which compiles the PLC program and checks it for logical errors, and the API DATA, TABLE, TRACE and WATCH LIST functions, with which you can check the status of the PLC operands.

The control starts the PLC cycles at the clock rate of the PLC cycle time. In **MP_plcCount**, you set the PLC cycle time to a multiple of the IPO clock. Use Module 9196 to determine the PLC cycle time setting.

Settings in the configuration editor	MP number
System	
CfgCycleTimes	
ipoCycle	100301
plcCount	100302

MP_ipoCycle

Cycle time of position controller (interpolation clock pulse)
Format: Selection menu
Selection: 3 ms
At present, only a cycle time of 3 ms for the position controller is supported. Therefore no other values can be selected.
Default: 3 ms
Access: LEVEL3
Reaction: RESET

MP_plcCount

PLC cycle time (Look Ahead cycle time)
Format: Numerical value
Input: 3 to 10 [· **MP_ipoCycle**]
The PLC and the Look Ahead run at a clock rate that is a multiple of the IPO clock (interpolation clock). The Look Ahead function is triggered exactly two IPO clock pulses after the PLC.
Default: 7
Access: LEVEL3
Reaction: RESET

Module 9196 Find the PLC cycle time
The PLC cycle time is determined in µs.

Call:
CM 9196
PL D <>PLC cycle time in [µs]>

8.1.1 The API 3.0 symbolic memory interface

The PLC programmer needs an interface for coupling his program to the functions and states of the NC part of the control. Often such an interface is known as API (**Application Programming Interface**). Three methods are used for HEIDENHAIN PLC programming:

- Integrated system subprograms (modules) are available: **PLC modules**
- On HEIDENHAIN NCK-based controls, a shared, structured data area is available, which the PLC programmer accesses with symbolic structure operands: **The API 3.0 symbolic interface.**
  This Technical Manual deals only with the symbolic memory interface.

If you are using the API 3.0 symbolic memory interface, symbolic operands are available, See "Overview of the PLC Operands" on page 378

Application

The following structures exist:

Group	Type of structure	Structure definitions
General data	ApiGen	PlcApiGeneral
Operating modes	ApiOmg	PlcApiOmg[OMG_COUNT]
Processing channels	ApiChn	PlcApiChannel[CHANNEL_COUNT]
Axes	ApiAxis	PlcApiAxis[AXIS_COUNT]
Spindles	ApiSpin	PlcApiSpindle[SPINDLE_COUNT]

Individual structure elements are accessed by indexes (except for the "General data" group).

Example of a PLC command line:

```
L PlcApiAxis[x].NN_AxInMotion
IFT
  ...
ENDI
```

Note

Further information about working with indexed structures is provided in the online help of PLCdesignNT.

ApiMarker.def definition file

HEIDENHAIN makes the **ApiMarker.def** file available to the PLC developer. As soon as the file is included in the PLC program via the **INCLUDE** command, the control uses the symbolic API 3.0. Collected in structures, the file contains all symbolic PLC operands.

Example:

```
* ================================================
#TYPEDEF PlcApiAxis
* ================================================
  internal                    D
  NN_AxLogNumber              D     * logical axis number
  NN_AxDriveReady             M     * drive is ready to work
  PP_AxDriveOnRequest         M     * drive on request
  NN_AxDriveOn                M     * drive is on
  .
  .
  .
```

Note

The **ApiMarker.def** file is located on the PLC partition of the control.

HEIDENHAIN releases a revised version of the **ApiMarker.def** file at irregular intervals. The most recent version of the ApiMarker.def file is automatically transferred to the control when the NC software is updated. The old **ApiMarker.def** file in the main directory of the PLC program will not be overwritten in order to ensure executability of the PLC program.

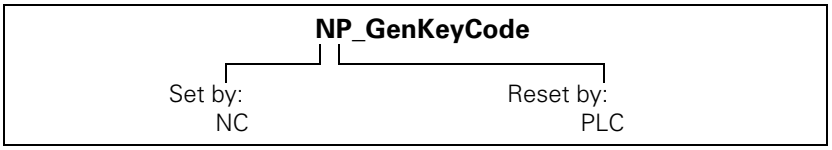
After an update of the NC software, you will find the current version of the file in the following directory of the control:

PLC:\proto\plc\ApiMarker.def

You can also find the file on the HESIS Web Including FileBase on the Internet at http://filebase.heidenhain.de.

Name convention for symbolic PLC operands

The first two letters at the beginning of the symbolic operands give information about the setting and resetting behavior:



Operand	Data direction	Setting or resetting behavior
NN_xxx	NC –> PLC	Set by NC, reset by NC
NP_xxx	NC –> PLC	Set by NC, reset by PLC
PP_xxx	PLC –> NC	Set by PLC, reset by PLC
PN_xxx	PLC –> NC	Set by PLC, reset by NC

For a list and description of all PLC operands See "Overview of the PLC Operands" on page 378.

Programming with the symbolic API

In the definition file **ApiMarker.def,** associated symbolic PLC operands are structured into five groups with the **#TYPEDEF** command:

- PlcApiGeneral
- PlcApiOmg
- PlcApiChannel
- PlcApiAxis
- PlcApiSpindle

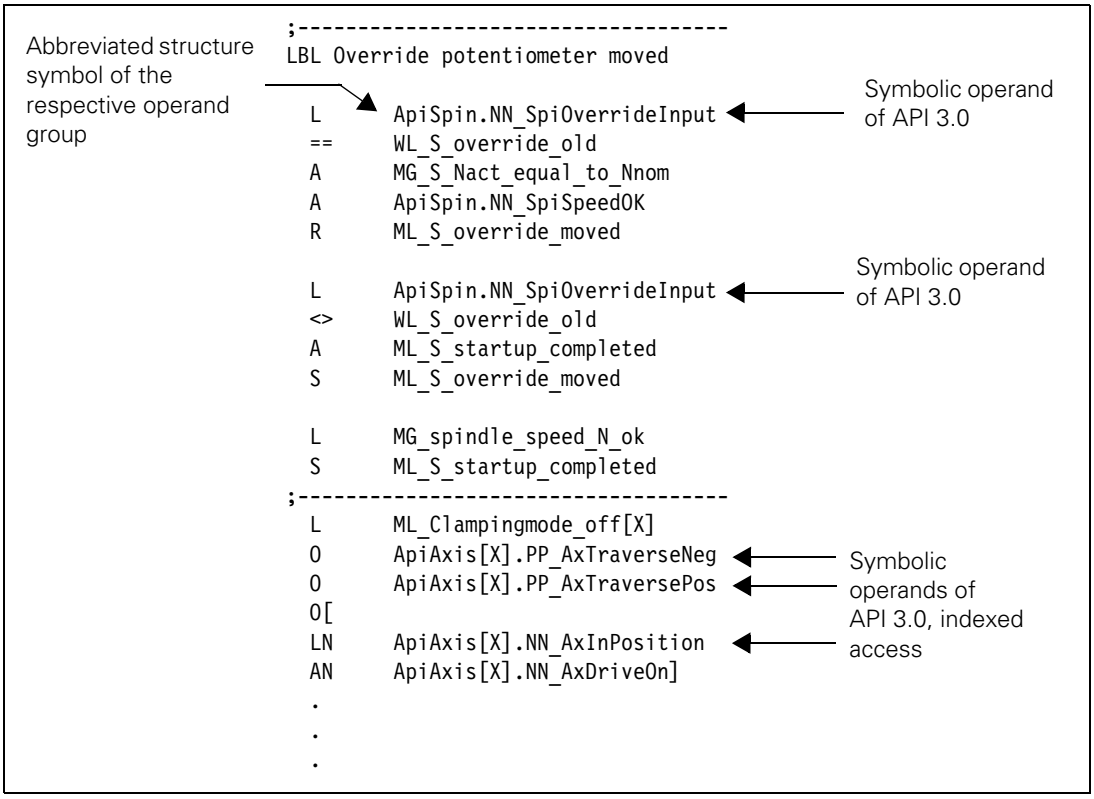
The **#TYPE** command assigns an abbreviated structure symbol to each of the five operand groups:

- PlcApiGeneral = **ApiGen**
- PlcApiOmg = **ApiOmg**
- PlcApiChannel = **ApiChn**
- PlcApiAxis = **ApiAxis**
- PlcApiSpindle = **ApiSpin**

You address individual elements of these five structures by entering the structure symbol, followed by a period as a separator, and then the name of the operand, e.g. **ApiSpin.NN_SpiOverrideInput**.

The structures ApiOMG, ApiChn and ApiAxis are accessed by indexes, e.g. PlcApiAxis[x].NN_AxInMotion.

Example of program:



Note

More information about programming with structures is provided in the online help of PLCdesignNT.

8.1.2 HEIDENHAIN PLC basic program

A PLC basic program for the control is available from HEIDENHAIN. This comprehensive PLC program serves as a basis for adapting the control to the requirements of the respective machine.

You need the software **PLCdesignNT** to adapt the PLC basic program.

Registered customers can download the current version of the PLC basic program from the "HESIS-Web Including Filebase" on the Internet (http://hesis.heidenhain.de).

The PLC basic program provides the following functions:

- Controlling all axes
- Positioning the axes after reference run
- Clamped axes
- Homing the axes, reference end positions
- Compensating the axis temperature
- Feed rate control
- Controlling and orienting the spindle
- Spindle brake
- Gear switching via M functions
- C axis via main drive
- C axis with separate drive
- Positioning of the tool turret with three-phase motor
- Servo turret (one drive for the turret wheel and the driven tool)
- Tool change for multifix tools
- PLC soft keys
- Displaying and managing PLC error messages
- Positioning the PLC axis (example)
- Hydraulic control
- Hydraulic chuck
- Sleeve 2, pressure
- Coupled tailstock
- Steady rest 2, pressure
- Turret control via Sauter switch module
- Electronic handwheels (with C or Y axis assignment)
- Controlling the coolant system
- Handling M functions
- Lubrication
- Chip conveyor
- Touch probes
- Controlling the doors

Note

If you have any questions regarding the PLC basic program, please contact the HEIDENHAIN PLC hotline.
Telephone: +49 / 8669 / 31-3102
E-mail: service.plc@heidenhain.de

8.1.3 Selecting the PLC programming mode of operation

Start the **PLC programming** mode of operation:

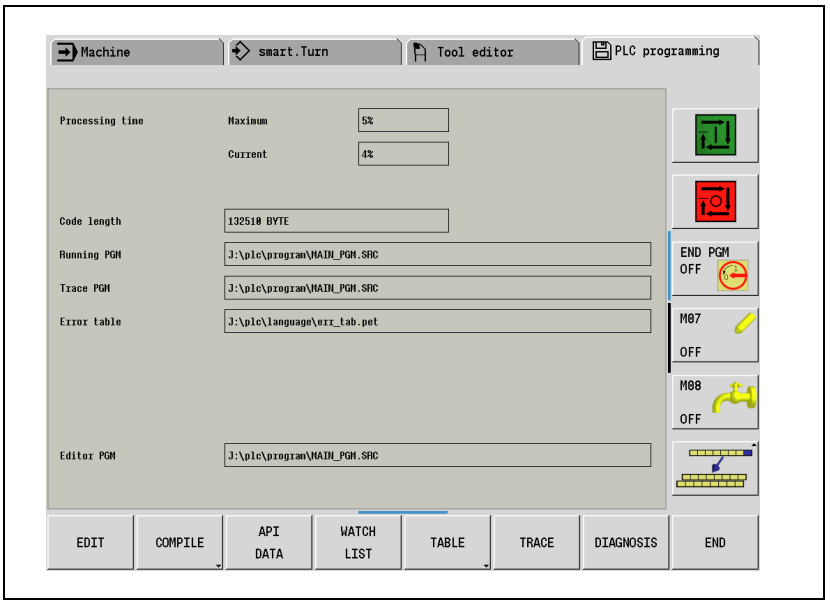
- Select the **Organization** mode
- Press the  soft key.
- Enter the code number 807667. Confirm with the **ENT** key or the **OK** button or soft key
  If you have already entered the code number, press the **PLC EDIT** soft key

Exit PLC mode:

- Press the **END** soft key or the **END** key.

8.1.4 PLC main menu

After you have entered the code number (or pressed the **PLC EDIT** soft key), the control displays the PLC main menu:



Processing time maximum:

Maximum run time of the PLC program.

The run time of the PLC cycle program is monitored. If the PLC cycle has not been completed two interpolator clock pulses before the PLC cycle time elapses, the control issues the error message **PLC: time out**.

Processing time current:
The time taken for the latest PLC scan in %.

The PLC processing time (time for a PLC cycle) is given as a percentage of the maximum time: 100% is the equivalent of a run time of 1 ms at a cycle time of 21 ms. Use the following formula to calculate the run time t_{run} [ms] as a function of the PLC cycle time t_{PLC} [ms] and the processing time $t_{calc.}$ [%]:

$$t_{run} = \frac{t_{PLC} \cdot t_{calc.}}{21}$$

Code length:
Length of the compiled sequential program in bytes.

Running PGM:
Name of the last compiled PLC program (program in process memory).

During switch-on, the control automatically compiles the program that was selected in process memory before switch-off.
The files only become active after they have been compiled.

Trace PGM:
Name of the last program or file opened with the **TRACE** function

Editor PGM:
Name of the program or file in the editor's main memory

PLC functions of the main menu

From the PLC main menu you can use soft keys to access the following PLC functions:

Soft key	Function	Page
EDIT	Edit the file located in RAM memory	1107
COMPILE	Compile the PLC program	1106
API DATA	Display the states of the symbolic operands (new programming interface)	1099
WATCH LIST	Display the states of the selected operands in a table	1100
TABLE	Display the logical states of the PLC operands (M/I/O/T/C/B/W/D)	1102
TRACE	Display the TRACE function	1104
DIAGNOSIS	Call the diagnostic functions	1110
END	Exit the **PLC programming** mode	–

8.1.5 File management

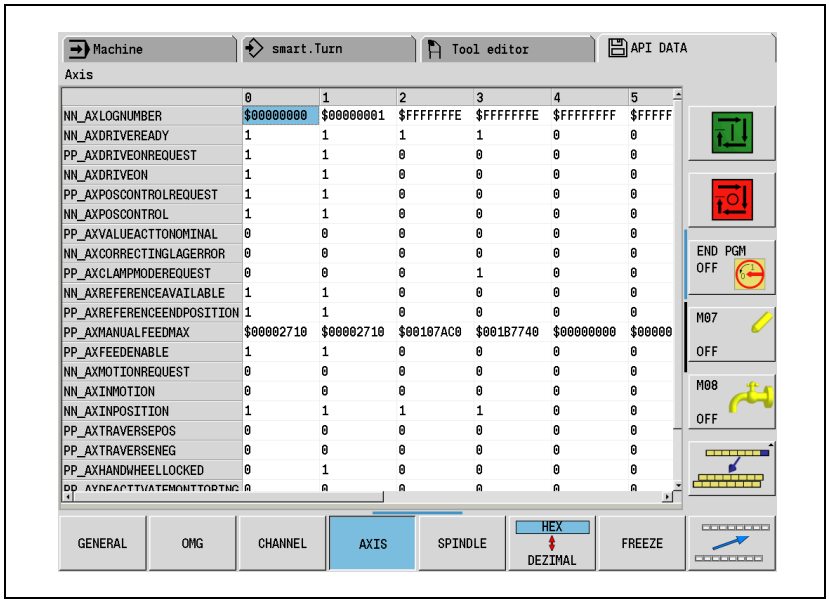
You call the file management by pressing the COMPILE PLC PGM soft key in the PLC programming mode of operation

File types displayed by the control when you press the **SELECT TYPE** soft key:

Soft key	Function
PLC	Show only PLC programs (*.PLC)
SRC	Show only PLC source files (*.SRC)
DEF	Show only PLC definition files (*.DEF)
MCG	Show only PLC machine configuration files (*.MCG)
PET	Show only PLC error tables (*.PET)
DEFAULT	Set a default display filter with the PLC file extensions *.PLC, *.SRC, *.DEF and *.PET.
DISPLAY FILTER	Specify a user-defined display filter. Enter "*.*" to display all files. If you enter more than one file extension, separate them with a semicolon.
	Return to previous menu.

8.1.6 The API DATA function

The API DATA function enables you to display the states or contents of the symbolic API markers and API double words on the control.



	0	1	2	3	4	5
NN_AXLOGNUMBER	$00000000	$00000001	$FFFFFFFE	$FFFFFFFE	$FFFFFFFF	$FFFFF
NN_AXDRIVEREADY	1	1	1	1	0	0
PP_AXDRIVEONREQUEST	1	1	0	0	0	0
NN_AXDRIVEON	1	1	0	0	0	0
PP_AXPOSCONTROLREQUEST	1	1	0	0	0	0
NN_AXPOSCONTROL	1	1	0	0	0	0
PP_AXVALUEACTTONOMINAL	0	0	0	0	0	0
NN_AXCORRECTINGLAGERROR	0	0	0	0	0	0
PP_AXCLAMPMODEREQUEST	0	0	0	1	0	0
NN_AXREFERENCEAVAILABLE	1	1	0	0	0	0
PP_AXREFERENCEENDPOSITION	1	1	0	0	0	0
PP_AXMANUALFEEDMAX	$00002710	$00002710	$00107AC0	$001B7740	$00000000	$00000
PP_AXFEEDENABLE	1	1	0	0	0	0
NN_AXMOTIONREQUEST	0	0	0	0	0	0
NN_AXINMOTION	0	0	0	0	0	0
NN_AXINPOSITION	1	1	1	1	0	0
PP_AXTRAVERSEPOS	0	0	0	0	0	0
PP_AXTRAVERSENEG	0	0	0	0	0	0
PP_AXHANDWHEELLOCKED	0	1	0	0	0	0

Soft keys for the API DATA function

Soft key	Function
GENERAL	Display the contents of general API markers
OMG	Display the contents of the API markers for machining groups
CHANNEL	Display channel-specific API data
AXIS	Display axis-specific API data
SPINDLE	Display the API markers that apply to the spindle
HEX DEZIMAL	Show operand contents in hexadecimal or decimal notation
FREEZE	Freeze the screen
	Return to previous menu.

8.1.7 The WATCH LIST function

The WATCH LIST function enables you to create a table providing a dynamic overview of the states of the selected PLC operands.

Meaning of the columns in the WATCH LIST:

- MODULE: **<Global>** for global symbolic operands or path with the name of the *.SRC file in which the operand is defined
- ADDR: Absolute address of the operand
- TYPE: Type (M for marker, B for byte, W for word, etc.) of the PLC operand
- VALUE: Contents of the operand
- COMMENT: Comment for the operand

Soft keys within the WATCH LIST function:

Soft key	Function
INSERT LINE	Insert a new line above the current line
DELETE LINE	Delete the current line
SYMBOL LIST	Display a selection list with all symbolic operands used in the active PLC program
WAHLEN	Mark the operand selected for transfer to the WATCH LIST. The operand is marked with an asterisk "*."
DELETE	Unmark the selected operand
DELETE ALL	Unmark all operands
INSERT	Load selected operands into the WATCH LIST
	Return to the previous menu
	Load or save the WATCH LIST
LOAD	Load the saved WATCH LIST selection from the memory card (*.WLT file)
STORE	Save the active WATCH LIST to the memory card
	Return to the previous menu
HEX DEZIMAL	Show operand contents in hexadecimal or decimal notation
FREEZE	Freeze the screen. Current changes are no longer displayed.
	Return to the PLC main menu

Display of symbolic operands in the WATCH LIST

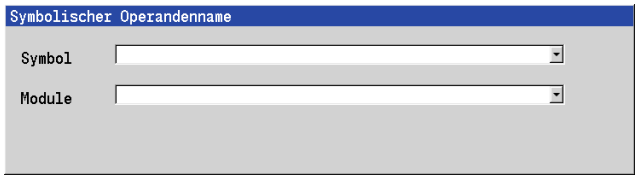
- Press the **WATCH LIST** soft key to call the menu of the WATCH LIST function.
- Press the **SYMBOL LIST** soft key to open a list box containing all global and local operands used in the PLC program.
- Use the arrow keys to move within the SYMBOL LIST. Press the right cursor key to open a tree structure. Press the left cursor key to close an open tree structure
- Use the arrow keys to select the desired operand and press the **SELECT** soft key to transfer it.
- Press the **INSERT** soft key to insert the selected operand.

Note

Operands can only be selected with the **SYMBOL LIST** soft key if you are working with the *.SRC source files of the PLC program on the control. Otherwise the error message **Selection list is empty** appears.

Display of operands in the WATCH LIST

- Press the **WATCH LIST** soft key to call the menu of the WATCH LIST function.
- Press the **INSERT LINE** soft key.
- The following dialog box appears:



Enter the symbolic operand name in the **Symbol** input box, or enter the module name in the **Module** input box. Press the GOTO key to view a history list of the operand and module names last entered.

- Confirm your entry with the OK soft key or button.

Internal process of the WATCH LIST function

If you are working with the source files (*.SRC), the control internally creates a complete symbol list from the *.MAP file, identifying the structures and arrays. In another step, the control resolves the structure elements and array elements, removes the nesting levels and internally creates a new list file. The generated information is displayed in a tree structure (SYMBOL LIST function). When you select and then insert symbols the first time, the file PLC:\TABLE\TMP.WLT is automatically created. This file is automatically saved when you exit the WATCH LIST function and loaded when you call the WATCH LIST again. You can save the active WATCH LIST under any desired name and then reload it. The control remembers the last active WATCH LIST and automatically loads it when you call the function again.

8.1.8 The TABLE function

From the PLC main menu, press the **TABLE** soft key to select the table of markers, inputs, outputs, counters, timers, bytes, words, double words, and strings. The states are displayed dynamically on the screen.

To select a certain operand:

- Use the arrow keys, or
- Press the GOTO key. A dialog box appears. Enter the number of the operand and confirm your entry with the **OK** soft key.

To set or reset markers, inputs, outputs, timers or counters:

- Press the **SET/RESET** soft key.

To change bytes, words, double words or strings:

- Use the arrow keys or the GOTO key to select the desired operand.
- Press the ENT key and enter the new value. Conclude your entry with the ENT key.

Soft keys within the TABLE function:

Soft key	Submenu	Function
M/I/O/T/C	Select Markers/Inputs/Outputs/Timers/Counters for **TABLE**.	
	M MARKER	Show a list of the markers
	I INPUT	Show a list of the inputs
	O OUTPUT	Show a list of the outputs
	T TIMER	Show a list of the timers
	C COUNTER	Show a list of the counters
	SET/ RESET	Set/reset the selected operand
	FREEZE	Freeze the screen. Current changes to PLC operands are no longer shown.
		Return to the PLC main menu

Soft key	Submenu	Function
B/W/D/S	Select Bytes/Words/Double words/Strings for **TABLE**	
	B BYTE	Show a list of the bytes
	W WORD	Show a list of the words
	D DOUBLE	Show a list of the double words
	S STRING	Show a list of the strings
	HEX DEZIMAL	Show operand contents in hexadecimal or decimal notation
	FREEZE	Freeze the screen. Current changes to PLC operands are no longer shown.
		Return to the PLC main menu
	Return to the PLC main menu	

8.1.9 The TRACE function

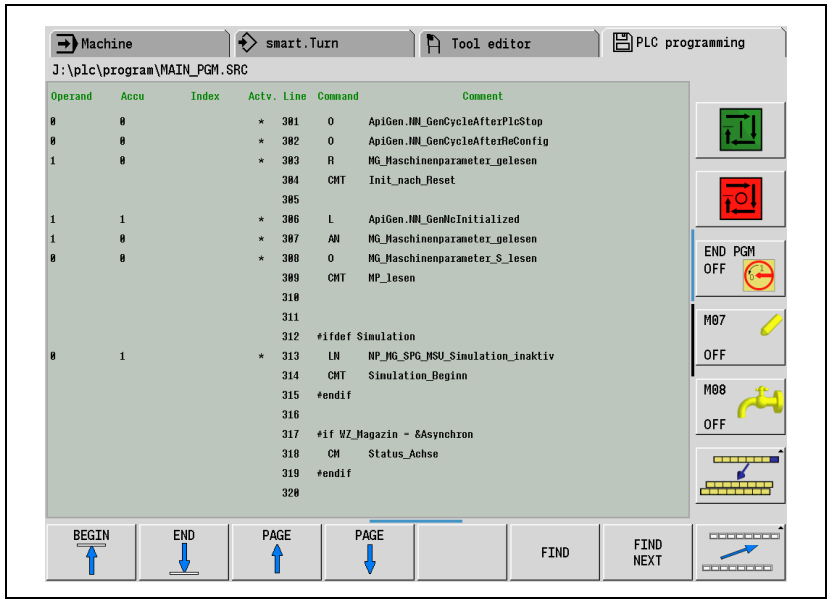
The TRACE function enables you to:

- Check the logical states of markers, inputs, outputs, timers, and counters.
- Check the content of bytes, words and double words.

From the PLC main menu, press the TRACE soft key to select the TRACE function. The control displays:

- The statement list (STL) of the selected PLC program
- For every program line, the content of the operand and the accumulator in HEX or decimal code (selectable by soft key)

The control identifies every cyclically executed command with an asterisk **(*)**. Use the arrow keys or the GOTO function to select the program section that the control should display on the screen. In addition, you can use the NAVIGATE soft key to call additional soft keys for navigating in the source text and a search function.
The PLC program to be displayed is selected with the **SELECT MODULE** soft key, and must be the currently active main program or a file integrated with USES.



Soft keys within the TRACE function:

Soft key	Submenu	Function
NAVIGATE	Open the submenu for navigating in the STL selected for the trace function	
	BEGIN	Jump to the beginning of the STL
	END	Jump to the end of the STL
	PAGE	Scroll back one page in the STL
	PAGE	Scroll forward one page in the STL
	FIND	Find text in the STL
	FIND NEXT	Find the next hit
		Return to previous menu.
ADD TO WATCH LIST	Take the PLC operands of the currently marked lint into the WATCH LIST	
HEX DEZIMAL	Show operand or accumulator contents in hexadecimal or decimal format	
SELECT MODULE	Open the file manager. The PLC program to be run in "Trace" mode can be selected.	
FREEZE	Freeze the screen. Current changes to PLC operands are no longer shown.	
	Return to the PLC main menu	

8.1.10 The COMPILE function

Compiling a completed PLC program transfers it to the process memory where it can then become active. The name of the compiled program then appears in the line **Running PGM** in the main menu.

Soft keys within the COMPILE function:

Soft key	Function
COMPILE ALL	Compile the current PLC program, current PLC error table, and current soft-key project file (**MP_mainPgm, MP_errorTable** and **MP_softkeyProject** entries in the **CfgPlcPath** machine parameter).
COMPILE PLC-MAIN PROGRAM	Only compile the current PLC program (entry **CfgPlcPath => MP_mainPgm**)
COMPILE PLC- ERRORTAB.	Only compile the current PLC error table (entry **CfgPlcPath => MP_errorTable**)
SELECT + COMPILE PLC PGM	Select and compile a PLC program
SELECT + COMPILE ERRORTAB.	Select and compile a PLC error table

To compile a PLC program:

- Press the **COMPILE PLC PGM** soft key for the control to switch to the program manager
- Use the arrow keys to select the PLC program to be compiled.
- Press the **SELECT** soft key.

Note

The compilation of very extensive PLC programs may take some time. Compilation is completed when the PLC main menu appears on the screen again and values are displayed under **Processing time.** If errors occur during the compilation of the program, the control displays a corresponding message in the PLC main menu.

8.1.11 The EDIT function

Pressing the **EDIT** soft key or selecting an editable file in the program manager opens the editor. PLC source texts (*.SRC, *.PLC), PLC definition files (*.DEF) and PLC error tables are editable files.

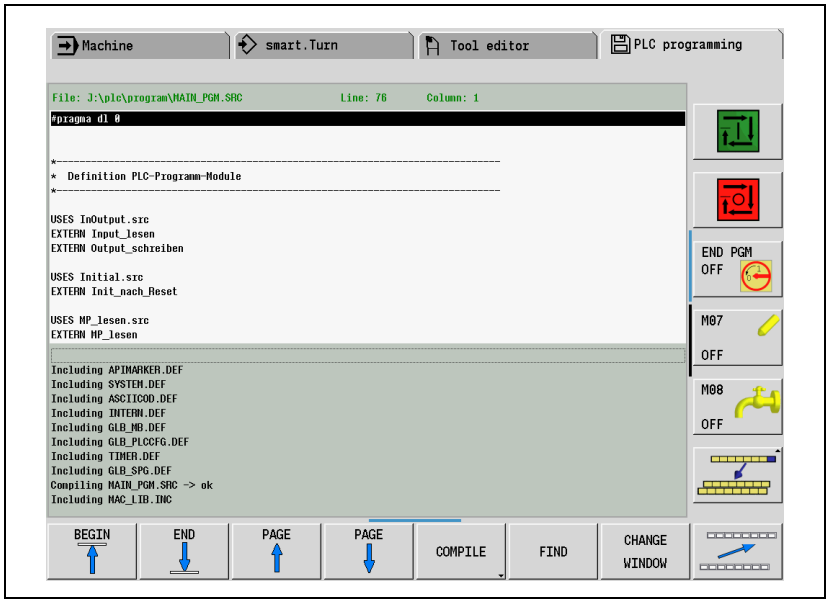
The feature content is like that of an ASCII editor. The cursor can be positioned in all directions and line breaks can be inserted. Tables are opened in a separate table editor.

The editor's display window takes up the entire screen and is divided into two sections only in case of an error. The upper part of the window is the workspace. In the lower third of the window, the PLC compiler displays status and error messages only in the event of an error.

If errors occurred during compilation of the PLC programs, you can use the soft keys ERROR and ERROR to jump between the individual errors in the editor window. Also, the errors are automatically marked with the cursor.

Note

The complete feature content of the editor only becomes available after you have connected an external USB keyboard.



Soft keys within the EDIT function:

Soft key	Submenu	Function
NAVIGATE	Opens the submenu for navigation in the editing window.	
	ANFANG	The cursor jumps to the beginning of the file in the active window.
	ENDE	The cursor jumps to the end of the file in the active window.
	SEITE	Page back in the open file by one page.
	SEITE	Page down in the open file by one page.
	FIND	Opens the dialog box of the search function.
		Return to previous menu.
COMPILE	Opens the submenu for selection of the COMPILE function.	
	COMPILE ALL	Compile the current PLC program, current PLC error table, and current soft-key project file (**MP_mainPgm, MP_errorTable** and **MP_softkeyProject** entries in the **CfgPlcPath** machine parameter).
	COMPILE PLC-MAIN PROGRAM	Only compile the current PLC program (entry **CfgPlcPath => MP_mainPgm**)
	COMPILE PLC-ERRORTAB.	Only compile the current PLC error table (entry **CfgPlcPath => MP_errorTable**)
	SELECT + COMPILE PLC PGM	Select and compile a PLC program
	SELECT + COMPILE ERRORTAB.	Select and compile a PLC error table
ERROR	Jump forward in the editing window to the next error	
ERROR	Jump back in the editing window to the previous error	

Soft key	Submenu	Function
PROJEKT DATEI WÄHLEN	Opens the status and error message window of the PLC compiler, even is there is no error pending.	
	ANFANG	The cursor jumps to the beginning of the file in the active window.
	ENDE	The cursor jumps to the end of the file in the active window.
	SEITE	Page back one page in the selected window
	SEITE	Page forward one page in the selected window
	AUSWÄHLEN	Opens the file highlighted by the cursor (only displayed if the cursor is located in the lower window)
	FENSTER WECHSELN	Switches between the upper part (workspace) and the lower part of the window (status messages of the PLC compiler).
		Return to previous menu.
	Return to the PLC main menu	

8.1.12 Diagnostic functions

The MANUALplus 620 features numerous diagnostic functions for finding errors.

To call the diagnostic functions:

- Select the **Organization** mode of operation.
- Enter the code number 95148.
- Press the **DIAGNOSIS** soft key.

The following diagnostic functions are available:

Soft key	Soft key	Soft key	Function
BUS DIAGNOSIS	After pressing this soft key, you can test various HSCI and Profibus settings, provided that you are using one of the two bus systems.		
	PROFIBUS	After pressing this soft key, you can test various Profibus settings, provided that you are using a Profibus system.	
	HSCI DIAGNOSIS	After pressing this soft key, you can test various HSCI settings, provided that you are using an HSCI system.	
DRIVE DIAGNOSIS	Various drive diagnosis functions can be selected after pressing this soft key. Before selecting the diagnostic function, under **Supply unit** you must select the power supply unit being used, so that the signals present are not interpreted as errors.		
	OSCI	The integrated oscilloscope is started.	
	The following soft key appears only if the **Power Interrupted** message was not acknowledged, and if the code number 688379 or 807667 was entered.		
	I CONTROL	The integrated oscilloscope for commissioning the current controller is opened.	
HEROS DIAGNOSIS	The file TNC:\herosdiagnose.txt is created after pressing this soft key. HEIDENHAIN uses this file for diagnosis of the operating system.		

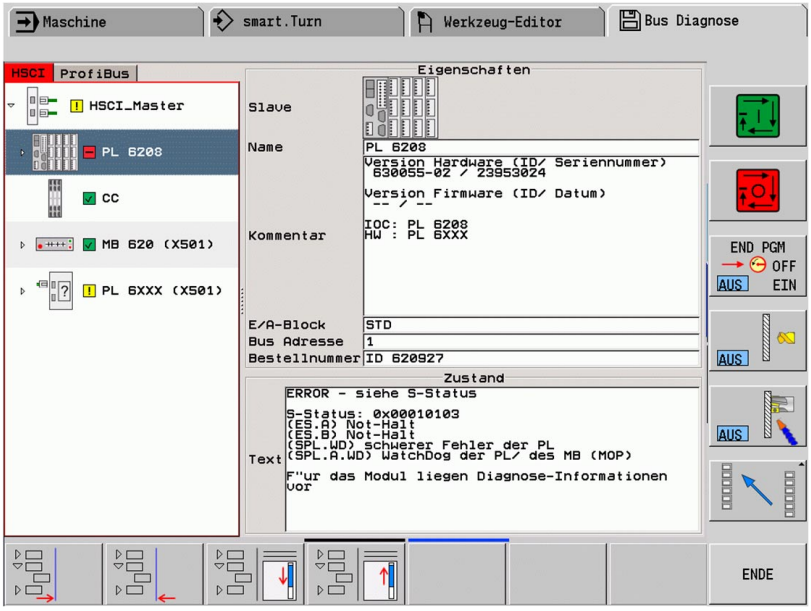
8.1.13 Bus diagnosis

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

You can access the functions as follows:

- Switch to the **Organization** mode of operation.
- Enter the code number 95148.
- Press the DIAGNOSIS soft key.
- Press the BUS DIAGNOSIS soft key.

Example screen:



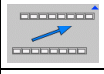
In Diagnosis mode, the structure of the HSCI/PROFIBUS system as well as the details of the HSCI/PROFIBUS components can be displayed.

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

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

Soft keys for HSCI/ PROFIBUS diagnostics

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

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

Navigation with the arrow keys of the operating panel:

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

Screen contents

"HSCI/PROFIBUS" window (at left)

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


OK

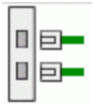
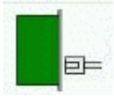
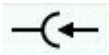

Error (further information in the text window)


Warning (further information in the text window)


Undefined state (further information in the text window)

"Attributes" window (at upper right)

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

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

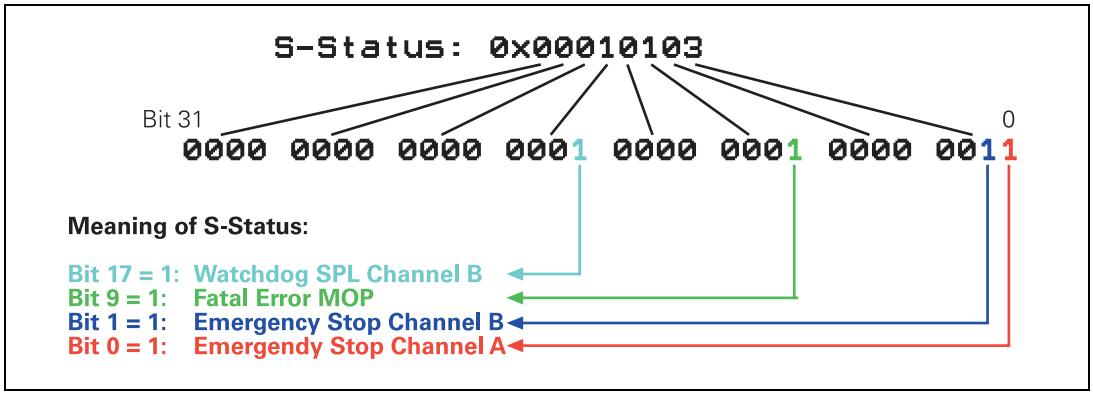
Field	Function	Formed from...
Comment	Name of the slave (for slaves and slave modules) or terminal description	■ Master entries: • "Configuration file" ■ Master/slave entries: • "Name" • "Hardware ID" • "Hardware serial number" • "Firmware ID" • "Firmware timestamp" • "Configuration file" ■ Terminal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

"State" window (at lower right)

Current status of selected HSCI/PROFIBUS component.

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

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



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

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

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

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

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

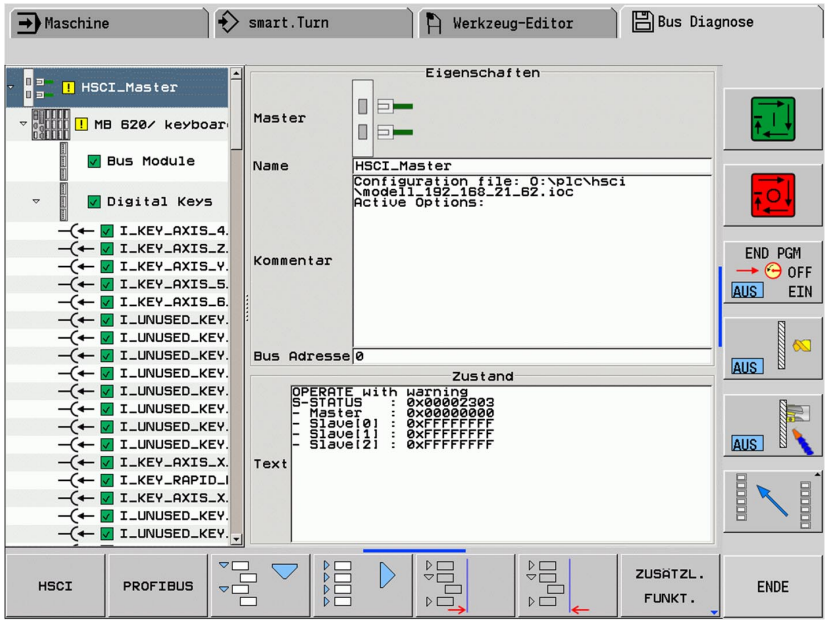
S status bit	Signal	Meaning
16	–SMOP.A.WD	SMOP watchdog, channel A
17	–SMOP.B.WD	Only in controls with Functional Safety (FS): SMOP watchdog, channel B
18	PGOOD.A	Voltage monitoring of channel A reports an error
19	PGOOD.B	Voltage monitoring of channel B reports an error
20	1	Reserved
21..31	1	Reserved

Overview of HSCI/ PROFIBUS I/O configuration

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

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

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



Settings in the configuration editor	MP number
System	
PLC	
CfgPlcPeriphery	
iocProject	103402
iocOption	103403

You can configure the PLC input/output systems with HSCI interface (PL 6xxx, UEC 11x, MB machine operating panel with connected HR handwheel) and the PROFIBUS components using the IOconfig PC software.

The components of an HSCI/PROFIBUS system are specified in a **project** with IOconfig and their arrangement is described. On the basis of this data, IOconfig generates the **IOC file** (*.IOC), which contains all relevant configuration data.

Then you transfer the IOC file to the control. Enter the name of the project file in the **MP_iocProject** parameter. With the next start-up, the IOC file is then read in and evaluated by the control.

Note

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

MP_iocProject

Configuration file for IOC hardware (HSCI and PROFIBUS components)
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 260 characters
Path and file name of the IOC project file, for example:
`%OEM%\plc\ioc\project.ioc`
Default: –
Access: LEVEL3
Reaction: RESET

Hardware components that are not required for every machine but are principally intended as maximum scope of upgrading for a specific machine series are called Options.

With the IOconfig PC software, the desired components are configured as options and specially identified in the IOC file. With the **MP_iocOption** machine parameter you specify which option(s) is/are active on the current machine:

MP_iocOption

	Option data for IOC hardware (HSCI, PROFIBUS) Available from NCK software version: 597 110-03.
Format:	Array [0...99]
Input:	Name(s) of IOC options – : By inserting "–" in front of the name of the option, it can be disabled.
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

8.3 Operands

8.3.1 Operanden-Übersicht

Die von HEIDENHAIN vordefinierten, symbolischen PLC-Operanden der symbolischen Programmierschnittstelle API 3.0 sind im Kapitel 5 aufgeführt (see "Overview of the PLC Operands" auf Seite 378).

Operand	Kurzbezeichnung	Adressbereich
Merker	M (Marker)	**M0 bis M9999** **M0 bis M999** frei; werden nur bei Eingabe der Schlüsselzahl 531210 gelöscht, nicht bei einem Reset (remanenter Bereich), der Bereich kann in der *.CFG-Datei des PLC-Compilers verkleiner werden **M1000 bis M3999** frei, werden bei Reset gelöscht **M4000 bis M5999** bei Verwendung des alten TNC-API reserviert für Schnittstelle NC/PLC. (M4800 bis M4999 werden vor dem ersten Durchlauf des PLC-Programms gelöscht, z. B. nach dem Compilieren oder Neustart) Bei Verwendung des neuen symbolischen API frei verwendbar. **M6000 bis M9999** frei, werden bei Reset gelöscht
Eingang	I (Input)	**I0 bis I999** Maximal 8 C-E/A-Systeme (PL 6xxx) sind im HSCI-System erlaubt, davon 1 System-PL PL 62xx oder 1 UEC 11x. Zuordnung der Eingänge zu symbolischen Operanden erfolgt über die PC-Software IOconfig.
Ausgang	O (Output)	**O0 bis O999** Maximal 8 externe PLC-E/A-Systeme (PL 6xxx) sind im HSCI-System erlaubt, davon 1 System-PL PL 62xx oder 1 UEC 11x. Zuordnung der Ausgänge zu symbolischen Operanden erfolgt über die PC-Software IOconfig.
Zähler	C (Counter)	Zähler starten: **C0 bis C47** Zähler-Inhalt: **C48 bis C95** Zählerimpuls-Freigabe: **C96 bis C143**
Timer	T (Timer)	Timer-Start: **T0 bis T47** Timer läuft: **T48 bis T95 und T96 bis T999**

Operand	Kurzbezeichnung	Adressbereich
Byte Wort Doppelwort	B (Byte) W (Word) D (Doubleword)	**B0 bis B9999** (8 Bit) **B0 bis B255** frei; abhängig von der Definition in der *.CFG-Datei des PLC-Compilers wird der definierte Bereich nur bei Eingabe der Schlüsselzahl 531210 gelöscht, nicht bei einem Reset (remanenter Bereich). Ist in der *.CFG-Datei kein Bereich definiert, ist B0 bis B127 der remanente Bereich. **B256 bis B2047** bei Verwendung des alten TNC-API reserviert für Schnittstelle NC/PLC. Bei Verwendung des neuen, symbolischen API frei verwendbar. **B2048 bis B9999** frei, werden bei Reset gelöscht
Konstante	K	**–2 147 483 647 bis +2 147 483 647**
String	S	**S0 bis S99**

Module 9405 Convert a symbolic operand into a numerical PLC operand
Module 9405 converts symbolic names of variables in character strings into the absolute addresses of the corresponding PLC operands. This module enables you to reduce the run time of your PLC program if you regularly read data from SQL tables.

The symbolic names must be contained within single quotes and follow a colon, e.g.
:'S_StringVariable'.

If this conversion already occurred once during the initialization of the PLC program, then the processing time is reduced for the subsequent module calls that replace the embedded variables with the momentary values. This affects modules 9440 and 9450, for example.

Example:

Two symbolic operands are to be converted. The operand **MG_W_TOOLNR** has the absolute address W1234 and the numerical value 5.
The operand **MG_W_POCKET** has the absolute address W3456 and the numerical value 19.

Output string:
UPDATE TOOL_P SET T = :'MG_W_TOOLNR' WHERE P = :'MG_W_POCKET'

After execution of Module 9405:
UPDATE TOOL_P SET T = :'W1234' WHERE P = :'W3456'

After execution of Module 9450:
UPDATE TOOL_P SET T = 5 WHERE P = 19

Call:
PS B/W/D/K/S<>String with symbolic name>
PS B/W/D/K/S<>Target for string with resolved symbols>
CM 9405
PL B/W/D <>Error number>
0: Module executed correctly
2: Parameter does not exist
3: Invalid address transferred
11: String could not be converted
12: String too long
15: Module was not called in a submit job

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Module executed successfully
	1	See above for errors

8.3.2 Operand addressing (byte, word, double word)

The memory for operands B (8 bits), W (16 bits), and D (32 bits) is only 8 bits wide. Since the operands can be 8, 16 or 32 bits wide, an overlap of the memory areas will occur, which you must take into account when addressing the memory.

Double word	Word	Byte	Memory	Word address	Double-word address
D0	W0	B0	8 bits	High byte	Highest byte
		B1	8 bits	Low byte	
	W2	B2	8 bits	High byte	
		B3	8 bits	Low byte	Lowest byte
D4	W4	B4	8 bits	High byte	
		B5	8 bits	Low byte	
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
D1020	W1020	B1020	8 bits	High byte	Highest byte
		B1021	8 bits	Low byte	
	NN_GenApiModuleErrorCode	B1022	8 bits	High byte	
		B1023	8 bits	Low byte	Lowest byte

For byte addressing, every address is accessible; for word addressing, every second address; and for double word addressing, every fourth from 0 to 4092. The address parameter indicates the high byte of the word address (W) and the highest byte of the double-word address (D).

Markers, timers and counters are addressed with the corresponding code letters M, T or C followed by the operand number (e.g. M500, T7, C18).

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcTimer	
[Key name of the timer]	
run	103503
start	103504
gate	103505
unit	103501
value	103502

Timers

The PLC has over 999 timers, which you control through special markers with the symbol **T**.

T0 to T47 are reserved as start markers for the timers, T48 to T999 are reserved as process markers.

The key name of the timer is freely selectable in the machine configuration. The process marker (e.g. T48) is defined with the optional machine parameter **MP_run.** If no value is entered in **MP_run,** the control uses the key name of the timer as process marker.

You must define the run times of the timers in machine parameter **MP_value.** As the time unit, under **MP_unit** you can choose between seconds and PLC cycles.

You start the first 48 timers by setting one of the timers with the process markers T0 to T47 for at most one PLC scan (otherwise the control restarts the timer with the negative edge for each additional scan). The control reserves the timer with the duration defined in machine parameter **MP_value** and sets the process marker until the defined duration has expired. A change of the default value for a PLC timer only becomes effective after a PLC program restart.

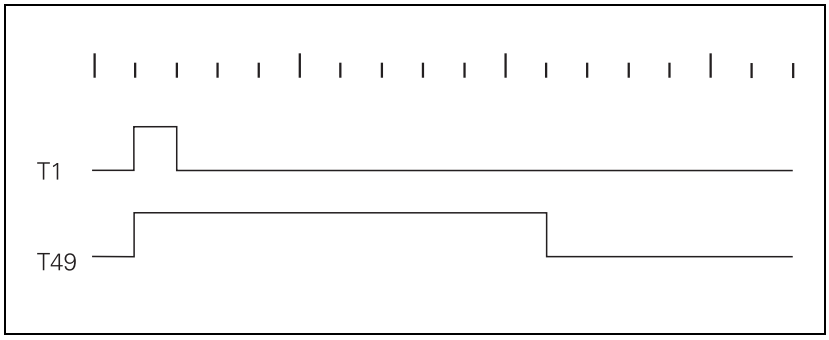
The optional machine parameters **MP_start** and **MP_gate** are not evaluated by the control. The PLCdesignNT development environment uses them for the automatic generation of PLC definition files.

You can also set and start timers T0 to T47 with Module 9006 (See "Module 9006: Set and start PLC timer" on page 1132).
Timers T96 to T999 can only be started through Module 9006.

Cyclic timers (> T96) can be defined and started with Module 9197 (See "Module 9197 Start cyclic timer" on page 1133). They are reset for one PLC cycle, and are then restarted automatically.

Example:

Start of timer 1
Run time in [timer] = 10 (PLC cycles)



Start timer	Timer is running	Machine parameter
T0	T48	timer[0]
T1	T49	timer[1]
T2	T50	timer[2]
T3	T51	timer[3]
T4	T52	timer[4]
T5	T53	timer[5]
T6	T54	timer[6]
T7	T55	timer[7]
T8	T56	timer[8]
T9	T57	timer[9]
T10	T58	timer[10]
T11	T59	timer[11]
T12	T60	timer[12]
T13	T61	timer[13]
T14	T62	timer[14]
T15	T63	timer[15]
T16	T64	timer[16]
T17	T65	timer[17]
T18	T66	timer[18]
T19	T67	timer[19]
T20	T68	timer[20]
T21	T69	timer[21]
T22	T70	timer[22]
T23	T71	timer[23]
T24	T72	timer[24]
T25	T73	timer[25]
T26	T74	timer[26]
T27	T75	timer[27]
T28	T76	timer[28]

Start timer	Timer is running	Machine parameter
T29	T77	timer[29]
T30	T78	timer[30]
T31	T79	timer[31]
T32	T80	timer[32]
T33	T81	timer[33]
T34	T82	timer[34]
T35	T83	timer[35]
T36	T84	timer[36]
T37	T85	timer[37]
T38	T86	timer[38]
T39	T87	timer[39]
T40	T88	timer[40]
T41	T89	timer[41]
T42	T90	timer[42]
T43	T91	timer[43]
T44	T92	timer[44]
T45	T93	timer[45]
T46	T94	timer[46]
T47	T95	timer[47]

Counters

The PLC has 48 counters, which you control via special markers with the symbol **C**.

C0 to C47 are reserved as start markers for the timers, C48 to C95 are reserved as process markers.

The key name of the counter is freely selectable in the machine configuration. The process marker (e.g. C48) is defined with the optional machine parameter **MP_run.** If no value is entered in **MP_run,** the control uses the key name of the counter as process marker.

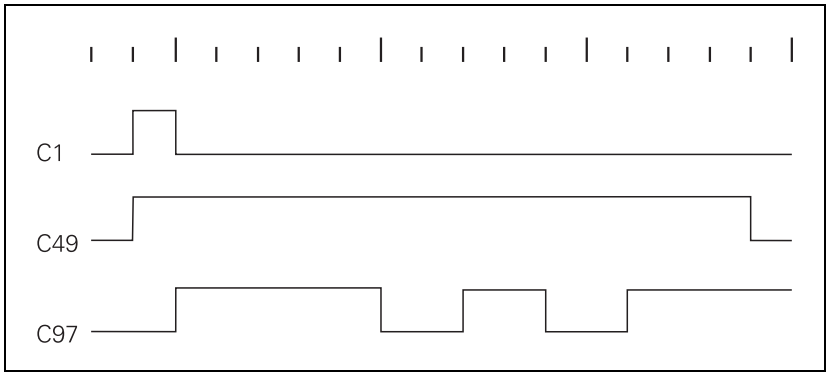
You must define the default values for the counters in machine parameter **MP_value.** As the time unit, under **MP_unit** you can choose between seconds and PLC cycles.

After you have set a marker from the C0 to C47 range, the control loads the counter with the value that is saved in machine parameter **MP_value.** As the time unit, under **MP_unit** you can choose between seconds and PLC cycles. The marker range C48 to C95 (process markers) indicates whether the counter has expired. Use markers C96 to C143 (enabling markers) to start and stop the counter.

The optional machine parameters **MP_start** and **MP_gate** are not evaluated by the control. The PLCdesignNT development environment uses them for the automatic generation of PLC definition files.

Example:

Logic diagram for counter C1
Counter default in [counter] = 10 (PLC cycles)



Start counter	Counter is running	Enable counter	Machine parameter
C0	C48	C96	counter[0]
C1	C49	C97	counter[1]
C2	C50	C98	counter[2]
C3	C51	C99	counter[3]
C4	C52	C100	counter[4]
C5	C53	C101	counter[5]
C6	C54	C102	counter[6]
C7	C55	C103	counter[7]
C8	C56	C104	counter[8]
C9	C57	C105	counter[9]
C10	C58	C106	counter[10]
C11	C59	C107	counter[11]
C12	C60	C108	counter[12]
C13	C61	C109	counter[13]
C14	C62	C110	counter[14]
C15	C63	C111	counter[15]
C16	C64	C112	counter[16]
C17	C65	C113	counter[17]
C18	C66	C114	counter[18]
C19	C67	C115	counter[19]
C20	C68	C116	counter[20]
C21	C69	C117	counter[21]
C22	C70	C118	counter[22]
C23	C71	C119	counter[23]
C24	C72	C120	counter[24]
C25	C73	C121	counter[25]
C26	C74	C122	counter[26]
C27	C75	C123	counter[27]
C28	C76	C124	counter[28]
C29	C77	C125	counter[29]
C30	C78	C126	counter[30]
C31	C79	C127	counter[31]
C32	C80	C128	counter[32]
C33	C81	C129	counter[33]
C34	C82	C130	counter[34]
C35	C83	C131	counter[35]
C36	C84	C132	counter[36]
C37	C85	C133	counter[37]
C38	C86	C134	counter[38]
C39	C87	C135	counter[39]

Start counter	Counter is running	Enable counter	Machine parameter
C40	C88	C136	counter[40]
C41	C89	C137	counter[41]
C42	C90	C138	counter[42]
C43	C91	C139	counter[43]
C44	C92	C140	counter[44]
C45	C93	C141	counter[45]
C46	C94	C142	counter[46]
C47	C95	C143	counter[47]

MP_run

Process marker for the timer or counter
Available from NCK software version: 597 110-03.

Format: String

Input: Max. 80 characters
Identifies the timer or counter (process marker), for which the respectively configured time applies.
Timer: T48 to T999
Counter: C48 to C95
If two or more times are configured, the parameter identifies the first timer or counter.

Default: No entry, value optional

Access: LEVEL3

Reaction: NOTHING

MP_start

Start marker for the timer or counter
Available from NCK software version: 597 110-03.

Format: String

Input: Max. 80 characters
Identifies the start marker for the timer or counter.
Timer: T0 to T47
Counter: C0 to C47
There is no start marker for timers T96 to T999. They can be started only through Module 9006.
This parameter is not evaluated by the control. The PLCdesignNT development environment uses it for the automatic generation of PLC definition files.

Default: No entry, value optional

Access: LEVEL3

Reaction: NOTHING

MP_gate

Release marker for counters
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 80 characters
Only for counters:
Identifies the enabling marker (gate marker) for the corresponding counter.
Gate markers are C96 to C143.
This parameter is not evaluated by the control. The PLCdesignNT development environment uses it for the automatic generation of PLC definition files.
Default: No entry, value optional
Access: LEVEL3
Reaction: NOTHING

MP_unit

Time unit for timers and counters
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **SECONDS**
Input in seconds
PLC_CYCLES
Input in number of PLC cycles
Default: SECONDS
Access: LEVEL3
Reaction: NOTHING

MP_value

Default values for PLC timers and counters
Available from NCK software version: 597 110-01.

Format: Array [1...99]
Input: Minimum value: 0
Maximum value: 1000 000
Decimal places: 3
Default: 0
Access: LEVEL3
Reaction: NOTHING

Management of timer addresses

The MANUALplus 620 references the timers and counters via symbolic names. The config object **CfgPlcTimer** is configured in the config editor for this:

- Parameter **MP_run** (103503):
  Name of the process marker (if the attribute is missing then the key is the name of the marker)
- Parameter **MP_start** (103504):
  Name of the start marker (timer to NR48 and counter)
- Parameter **MP_gate** (103505):
  Name of the gate marker (only counter)

The PLC compiler of the MANUALplus 620 can manage position and timer addresses itself (as with B/W/D and markers). The **#pragma range** pragma instruction must be entered in the source code for this.

These pragma statements also have the following effects:

- Global/static memory is reserved in rising order
- Support of two ranges each for /bn and /mn areas
- Compiler range /sn for strings

Syntax:

(Instruction for range reservation of the timers and counters. The range must be between 48 and 999 in each case.)

```
#define /tn <min value> <max value>
#define /cn <min value> <max value>
```

Definition of a timer started via a module:

```
#define <run timer name>  T
```

Definition of a timer started via a start timer:

```
#define /ts:<start timer name>  <run timer name>  T
```

Definition of a counter started via a module:

```
#define <run counter name>  C
```

Definition of a counter started via a start counter:

```
#define /cs:<start counter name> /cg:<counter gate name>
<run counter name>  C
```

Module 9006:Set and start PLC timer

Use Module 9006 to define the run time for a PLC timer and start the timer.
Constraints:

- If during a PLC scan a timer from T0 to T47 is set in the PLC program, and the same timer is activated through Module 9006, then the direct activation through T0 to T47 has priority regardless of whether the module is called before or after setting T0 to T47.
- Immediately after the module call, one of the markers T48 to T96 is set. T0 to T47 are not set.
- The control rounds the actual run time up to integral PLC cycle times.
- Cancel run time: Reset timers T48 to T999.

Call:

PS	B/W/D/K	<>Timer number> Input value: 0 to 999
PS	B/W/D/K	<>Run time> 0 to 1 000 000 000 [ms] –1: Run time from machine parameter
CM	9006	

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Timer started
	1	Error. See NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Invalid timer number or excessive run time
	2	Timer already assigned for cyclic timer
	3	Timer is started as cyclic timer (Module 9197)

Module 9197 Start cyclic timer

Module 9197 can define and start a timer > T96 as cyclic timer. After expiration of the defined time, the timer is reset for a PLC cycle and afterwards is automatically restarted.

- A cyclic timer can be stopped by setting the run time to 0.
- The programmed timer is set immediately after the module call.
- The run time is programmed in [ms].
- The control rounds the actual run time up to integral PLC cycle times.

Call:

PS	B/W/D/K	<>Timer number> 96 to 999
PS	B/W/D/K	<>Run time> 0 to 1 000 000 000 [ms] –1: Run time from machine configuration
CM	9197	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Timer started
	1	Error. See NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Excessive run time
	3	Invalid timer number

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcFastInput	
number	103701
significance	103702
operand	103703

With CfgPlcFastInput, you define the PLC inputs that are interrogated within the position controller cycle time (See "MP_plcCount" on page 1089) instead of within the defined PLC cycle time (See "MP_ipoCycle" on page 1089). In MP_number, enter the PLC input of the control that is to be used as a fast PLC input. You define the associated symbolic PLC operands set by the fast PLC input in **MP_operand.**

A maximum of five PLC inputs can be defined as fast PLC inputs. For this purpose, five input arrays [arrays 0...4] are available in the configuration editor of the control.

For the control to identify with certainty a signal change, the signal duration at the fast PLC input must last a minimum of 4 ms.

Attention

Only the PLC inputs of the control can be defined as fast PLC inputs, and not the inputs on a PL 510.

MP_number

Numerical designation for fast PLC inputs
Available from NCK software version: 597 110-01.

Format: Array [0...4]
Input: 0 to 31 [no. of the PLC input on the MC]
Default: 0
Access: LEVEL3
Reaction: RUN

MP_significance

Activation criterion for fast PLC inputs

Format: Selection menu

Selection: **lowActive**
Activate at LOW level
highActive
Activate at HIGH level
allEdges
Activate at both levels
disabled
Switched off

Default: disabled

Access: LEVEL3

Reaction: RUN

MP_operand

PLC operand for fast PLC inputs

Format: Array [0...4]

Input: A string with max. 24 characters can be entered for each array.
Example:
NN_FastInput_01

Default: NN_FastInput_01 to NN_FastInput_05

Access: LEVEL3

Reaction: RUN

8.4 Data Organization

8.4.1 Data organization on the CFR memory card

The **CFR** memory card of the MANUALplus 620 is divided into three partitions:

TNC partition

User-specific data such as NC programs, tool tables, and datum tables.

PLC partition

Your OEM-specific data such as system files, PLC programs, machine parameters, help files, PLC dialogs, PLC error tables, compensation value tables and OEM cycles. The PLC partition is visible only after you have entered the code number 807667.

As a machine tool builder, you are concerned primarily with the PLC partition.

Note

Define all your machine-specific settings with the configuration editor!

SYS partition

For HEIDENHAIN

System-specific files, such as system files, NC dialogs, HEIDENHAIN cycles, etc.
The partition SYS can neither be viewed nor selected.

Attention

Alterations to the system partition can impair proper function of the control!

Size of the partitions

The storage capacity of the CFR memory card is 1 GB.

Partition	Contents	Size
SYS:	System files	640 MB
PLC:	OEM files	50 MB
TNC:	User files	300 MB

Touch probe monitoring

The control now monitors the amount of memory available. The memory available on the CFR CompactFlash memory card as well as the RAM main memory are monitored. If warning thresholds are reached or exceeded, the error message **Too little free memory on <partition/memory>** is output. The control checks the memory cyclically. The warning message is repeated each cycle until you have made more memory available, such as by deleting files that are no longer needed. Press the ERR key and the INTERNAL INFO soft key to have the control display the affected partition, the warning threshold and the remaining memory available. If the **Not enough free RAM** error message is displayed, HEIDENHAIN recommends closing memory-intensive applications (such as the **Test Run** operating mode or editing graphics) until any active workpiece machining process is completed. Then restart the control.

Partition or memory	Warning threshold [MB]	Cycle time for checking [s]
TNC:\	1	30
PLC:\	1	60
Main memory (RAM)	20	30

8.4.2 Data organization on the hard disk

There are three different drives (partitions):

TNC partition

User-specific data such as NC programs, tool tables, and datum tables.

PLC partition

Your OEM-specific data such as system files, PLC programs, machine parameters, help files, PLC dialogs, PLC error tables, compensation value tables and OEM cycles. The PLC partition is visible only after you have entered the code number 807667.

As a machine tool builder, you are concerned primarily with the PLC partition.

Note

Define all your machine-specific settings with the configuration editor!

SYS partition

For HEIDENHAIN

System-specific files, such as system files, NC dialogs, HEIDENHAIN cycles, etc.
The partition SYS can neither be viewed nor selected.

Attention

Alterations to the system partition can impair proper function of the control!

Assignment to drives

Partition	Contents	Hardware code
SYS	System files	"R" drive
PLC	OEM files	"O" drive
TNC	User files	"V" drive

8.4.3 Compressing graphic files

Use the ZIP packer to compress graphic files (e.g. PLC soft keys, support graphics for cycles, or the startup screen) in order to save memory space. For example, the size of a 2-MB bitmap file (*.bmp, *.bmx) can be reduced to 40 KB with the ZIP packer. You have two basic possibilities:

Zipping individual files (e.g. the startup screen):

Compress the graphics file and give the ZIP file the following name:

[File name].[extension].zip

Example:
The startup screen **startup_1024x768.bmp** is compressed and saved as **startup_1024x768.bmp.zip** to **PLC:\resource.** Make sure to remove the original **startup_1024x768.bmp** file, since the control first searches for uncompressed files. If the uncompressed *.bmp does not exist, then the control automatically uses the ZIP archive whose file name contains the graphics file. The file is automatically unzipped and displayed at run-time.

Zipping entire directories (e.g. all support graphics)

You can compress all graphics files in a directory into one ZIP archive. You must use the directory name as the name of the ZIP archive. The original directory must be renamed or deleted.
Example:
The following cycle support graphics are contained in the **PLC:\oemcyc\hlp\nc** directory:

- **PLC:\oemcyc\hlp\nc\help1.bmp**
- **PLC:\oemcyc\hlp\nc\help2.bmp**
- **PLC:\oemcyc\hlp\nc\help3.bmp**

A ZIP archive with the file name **hlp.zip** is now created with the following contents:

- **nc\help1.bmp**
- **nc\help2.bmp**
- **nc\help3.bmp**

The ZIP file is saved in the **PLC:\oemcyc** directory, and the uncompressed original **PLC:\oemcyc\hlp\nc** directory is removed or renamed.
The path to the zipped graphics file **help1.bmp** from the example above is:

- **PLC:\oemcyc\hlp\nc\help1.bmp** or
- **PLC:\oemcyc\hlp.zip\nc\help1.bmp**

Settings in the configuration editor	MP number
System	
Paths	
CfgOemPath	
ncDir	102006
CfgUserPath	
ncDir	102201

Use the machine parameter **MP_ncDir** to define a list of drives and/or directories, which are displayed in the file manager of the control. The drives and paths are visible only if the user has the required access right.

In the factory default setting of the NC software, the three drives TNC:\, PLC:\ and SYS:\ are already configured through the environmental variables %USR%, %OEM% and %SYS%. The SYS partition is reserved exclusively for HEIDENHAIN and is not visible without the HEIDENHAIN daily password. The PLC:\ partition only becomes visible when you enter the PLC code number 807667.

You can add further drives or directories to the list in **MP_ncDir,** which are to be displayed in the folder tree after the file manager has been opened.

CfgOemPath:

MP_ncDir

List of drives and/or directories
Available from NCK software version: 597 110-01.

Format: Array [0...9]

Input: The drives and directories entered here are visible in the file manager, provided that you have the required access rights. The respective paths may contain NC programs or tables, for example, floppy disk drive directories, HDR or CFR directories, network drives, etc.

Default: [0]: %USR%\
[1]: %OEM%\
[2]: %SYS%\

Access: LEVEL3

Reaction: RESET

CfgUserPath:

MP_ncDir

	List of drives and/or directories Available from NCK software version: 597 110-01.
Format:	Array [0...9]
Input:	The drives and directories entered here are visible in the file manager, provided that you have the required access rights. The respective paths may contain NC programs or tables, for example, floppy disk drive directories, HDR or CFR directories, network drives, etc.
Default:	No value, parameter optional
Access:	LEVEL1
Reaction:	NOTHING

8.4.5 PLC system files

Paths and names for PLC system files and data are defined in the **plc.cfg** file.

Note

Make sure that you only make entries to the plc.cfg file by using the configuration editor.

Path for the plc.cfg file

Settings in the configuration editor	
CfgConfigDataFiles	
dataFiles	
...	
...	
[10]::	PLC:\config\plc.cfg
...	
...	

In the configuration editor (code number 95148), press the **CONFIG FILE LISTS** soft key to enter the path for the **plc.cfg** file.
In the CfgDataFiles object, under **dataFiles** you enter the paths of all *.cfg files required for the system (See "Machine Parameters" on page 245).
The default path for the plc.cfg file is PLC:\config\plc.cfg.
This means that the control looks for the necessary entries for PLC system data in this file in this path.

Paths and names for PLC programs and text files

Settings in the configuration editor	MP number
System	
Paths	
CfgPlcPath	
mainPgm	102301
pwmPgm	102302
errorTable	102303
errorText	102304
fn14ErrorText	102312
dialog	102305
softkeyProject	102306
compErrorTable	102307
compCfgFile	102308
events	102309
keymapFile	102310
magazineRules	102311

In the configuration editor, specify under **cfgPlcPath** which directories and names are used for storing PLC programs and files for PLC error messages.

The following path and file names must be specified:

- **MP_mainPgm:**
  The path and name of the PLC main program (*.src or *.plc). All necessary program modules (*.src) are called from this program as required. The program modules must be in the same directory as the PLC main program.
- **MP_pwmPgm**:
  Path and name of the PLC program for commissioning the machine (see Current Controller Adjustment). This PLC program is used as an alternative if the machine parameter **MP_currentControlAdjust** is set to "on."
- **MP_errorTable:**
  Path and file name for the PET table (file name is plc_err.pet); PLC error table with references to error message texts (plc_err.a) and attributes for the control's behavior when an error occurs (created with the PLCtext software); see PLCdesignNT User's Manual.
- **MP_errorText:**
  File name for error message texts (e.g. **`ErrorText.csv`**); Only the file name is entered – the language-sensitive text files must be saved in language-specific directories which cannot be changed.
  - German texts in **%OEM%\plc\language\de**ErrorText.csv
  - English texts in **%OEM%\plc\language\en**ErrorText.csv
  - Czech texts in **%OEM%\plc\language\cs**ErrorText.csv
  - French texts in **%OEM%\plc\language\fr**ErrorText.csv
  - Italian texts in **%OEM%\plc\language\it**ErrorText.csv
  - Spanish texts in **%OEM%\plc\language\es**ErrorText.csv
  - Portuguese texts in **%OEM%\plc\language\pt**ErrorText.csv
  - Swedish texts in **%OEM%\plc\language\sv**ErrorText.csv
  - Danish texts in **%OEM%\plc\language\da**ErrorText.csv
  - Finnish texts in **%OEM%\plc\language\fi**ErrorText.csv
  - Dutch texts in **%OEM%\plc\language\nl**ErrorText.csv
  - Polish texts in **%OEM%\plc\language\pl**ErrorText.csv
  - Hungarian texts in **%OEM%\plc\language\hu**ErrorText.csv
  - Russian texts in **%OEM%\plc\language\ru**ErrorText.csv
  - Chinese texts in **%OEM%\plc\language\zh**ErrorText.csv
  - Chinese (Traditional) texts in **%OEM%\plc\language\zh-tw**ErrorText.csv
  - Slovenian texts in **%OEM%\plc\language\sl**ErrorText.csv
  - Slovak texts in **%OEM%\plc\language\sk**ErrorText.csv
  - Latvian texts in **%OEM%\plc\language\lv**ErrorText.csv
  - Norwegian texts in **%OEM%\plc\language\no**ErrorText.csv
  - Korean texts in **%OEM%\plc\language\ko**ErrorText.csv
  - Estonian texts in **%OEM%\plc\language\et**ErrorText.csv
  - Turkish texts in **%OEM%\plc\language\tr**ErrorText.csv
  - Romanian texts in **%OEM%\plc\language\ro**ErrorText.csv
  - Lithuanian texts in **%OEM%\plc\language\lt**ErrorText.csv
- **MP_dialog**:
  File name for PLC dialogs (e.g. Dialog.csv); Only the file name is entered – the language-sensitive text files must be saved in language-specific directories which cannot be changed. (see error message texts)
- **MP_softkeyProject**:
  Path and name of project file for PLC soft keys;
- **MP_compErrorTable**:
  Preset by HEIDENHAIN and should not be changed.
  Path and file name of the error table of the PLC compiler;

- **MP_compCfgFile**:
  Path and name of the configuration file for the PLC compiler;
- **MP_events**: (not currently evaluated!)
  Path and file name of the PLC event list (spawn processes)
- **MP_keymapFile**:
  Preset by HEIDENHAIN and should not be changed.
  Path and name of the configuration file for the keyboard mapping.
- **MP_magazineRules:** (optional)
  Path and name of a file with magazine rules for managing the tool memory.
  Example: **%OEM%\plc\tchrules.tcr**

MP_mainPgm

	Path and file name of the PLC main program
Format:	String
Input:	Max. 260 characters Path and file name of the PLC main program Example: **%OEM%\PLC\main.src**
Default:	–
Access:	LEVEL3
Reaction:	NOTHING

MP_pwmPgm

	Path and file name of the PLC commissioning program
Format:	String
Input:	File name and path of the PLC program for commissioning the current controller. This PLC program is compiled and used as an alternative if the machine parameter **MP_currentControlAdjust** (System/CfgHardware) is set to "on." Example: **%OEM%\PLC\SetUp.plc**
Default:	–
Access:	LEVEL3
Reaction:	NOTHING

MP_errorTable

	Path and file name of the PET table
Format:	String
Input:	Max. 260 characters Path and file name of the PLC error table Example: **%OEM%\PLC\LANGUAGE\ERR_TAB.PET**
Default:	–
Access:	LEVEL3
Reaction:	NOTHING

MP_errorText

	Name of the text file for PLC error texts
Format:	String
Input:	Max. 260 characters Name (without path) of the text file for PLC error texts. The language-sensitive text files must be saved in language-specific directories that cannot be changed. Example: **ErrorText.csv**
Default:	–
Access:	LEVEL3
Reaction:	NOTHING

MP_dialog

Name of the text file for PLC dialogs
Format: String
Input: Max. 260 characters
Name (without path) of the text file for PLC error texts.
The language-sensitive text files must be saved in language-specific directories that cannot be changed.
Example: **dialog.csv**
Default: –
Access: LEVEL3
Reaction: NOTHING

MP_softkeyProject

Path and name of the project file for PLC soft keys
Format: String
Input: Max. 260 characters
Path and name of the file
Example: **%OEM%\BasisPgm\Softkeys\softkeys.xrs**
Default: –
Access: LEVEL3
Reaction: NOTHING

MP_compErrorTable

Path and file name of the error table of the PLC compiler
(Value preset by HEIDENHAIN and should not be changed!)
Format: String
Input: Max. 260 characters
Path and name of the file
Default: **%SYS%\config\plccomp.ert**
Access: LEVEL3
Reaction: NOTHING

Note

Be sure not to change the default value in **MP_compErrorTable,** or proper operation of the PLC compiler can no longer be guaranteed.

MP_compCfgFile

Path and name of the configuration file for the PLC compiler

Format: String
Input: Max. 260 characters
Path and name of the file
Example: **%OEM%\BasisPgm\Programm\OEM.cfg**
Default: –
Access: LEVEL3
Reaction: NOTHING

MP_keymapFile

Path and name of the configuration file for the keyboard mapping.
(Value preset by HEIDENHAIN and should not be changed!)

Format: String
Input: Max. 260 characters
Path and name of the file
Default: **%SYS%\resource\plckeymap_te530s.xml**
Access: LEVEL3
Reaction: NOTHING

Note

Be sure not to change the default value in **MP_keymapFile,** or proper operation of the control can no longer be guaranteed.

MP_magazineRules

Path and name of an ASCII file (*.TCR) with magazine rules
Available from NCK software version: 597 110-02.

Format: String
Input: Max. 260 characters
Path and name of the file
Example: **%OEM%\plc\tchrules.tcr**
Default: Optional input
Access: LEVEL3
Reaction: NOTHING

Settings in the configuration editor	MP number
System	
Paths	
CfgOemPath	
cycleMainTreeFile	102003
cycleSubTreeFiles	102004
oemCycle	102005

Use the machine parameters **MP_cycleMainTreeFile**, **MP_cycleSubTreeFiles** and **MP_oemCycle** to inform the control about the location where the CycleDesign cycle files are stored and about the path of the OEM cycles.

MP_cycleMainTreeFile

Path and name for the OEM cycle file (.CDF)
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 260 characters
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_cycleSubTreeFiles

List of paths/names of user cycle files (.CDF)
Available from NCK software version: 597 110-01.

Format: Array [0...9]
Input: String of max. 260 characters
The soft-key rows of these files are appended to the soft-key rows of the system or OEM cycle files.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_oemCycle

Path for OEM cycles
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 260 characters
Default: %OEM%\oemcyc
Access: LEVEL3
Reaction: RESET

PLC program version

Settings in the configuration editor	MP number
System Versions CfgPlcVersion **plcVersion**	 107301

Enter the version number of the PLC program in **MP_plcVersion**, which should be displayed on the screen of the control in the code number dialog.

MP_plcVersion

PLC software version; displayed version of the PLC program
Format: String
Input: Max. 32 characters
PLC program version
Example: **MANUALplus 620_BasisPgm_v1.5**
Default: –
Access: LEVEL3
Reaction: NOTHING

8.5 M Functions (M Strobe)

In the control you can program miscellaneous functions, also known as M functions. The code of an M function is transferred to the PLC before or after execution of the NC block.

Certain M functions are reserved for the NC or have fixed meanings for the NC. The other M functions are freely available and are evaluated by the PLC.

The structure of the M Functions menu in the DIN/ISO mode of the **smart.Turn** Programming mode can be adapted to the available M functions.

M functions are channel-sensitive. M functions are configured in two steps:

- In the channel-sensitive parameter **MP_mStrobes,** you define a key name for each M function used in this machining channel. In this way, you assign the M functions to this machining channel.
- In the parameter object **System/PLC/CfgPlcMStrobe,** you configure the M function.

8.5.1 Assigning M functions to the machining channels

Settings in the configuration editor	MP number
Channel	
ChannelSettings	
[Key name of the channel]	
CfgPlcStrobes	
mStrobes	201601
unitOfMeasure	201605

MP_mStrobes

M strobe descriptions of this machining channel
Format: Array [0...99]
Input: Key name of the M strobes used in this machining channel
Default: –
Access: LEVEL3
Reaction: RESET

In the marker defined in **MP_unitOfMeasure**, the PLC is informed of the unit of measure used in the NC program when M, S, T or alias strobes are executed.

MP_unitOfMeasure

Symbolic name or number of the PLC marker for the unit of measure of the NC program
Format: String
Input: Name of the PLC marker, which informs the PLC of the unit of measure of the NC program to be run.
PLC marker = 1: Inches
PLC marker = 0: Metric system
Default: Value optional
Access: LEVEL3
Reaction: RESET

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcMStrobe	
[Key name of M strobe]	
min	103901
max	103902
signal	103903
acknowledge	103904
code	103905
data	103906
revoke	103907
split	103908
track	103914
singular	103910
blockEnd	103911
blockSearch	103912
sync	103913
macro	103914

Transfer of the M function

If there is more than one M function number with the same transfer parameters, the M function numbers can be defined in a parameter object.

- Multiple M codes in the parameter object:
  - **MP_min:** Code of the first M function
  - **MP_max:** Greatest M code of the group
- One M code in the parameter object:
  - **MP_min:** M code of the parameter object
  - **MP_max:** No entry

If **MP_code** is defined, the NC transfers the programmed M code in the PLC word marker defined in **MP_code** and other data of the M function in the PLC word marker defined in **MP_data.**

MP_min

Code of the first M function

Format: Numerical value

Input: 0 to 9999

First M code described in the parameter object. The properties described in this parameter object apply for the specified M functions.

Default: 0

Access: LEVEL3

Reaction: RESET

MP_max

Code of the last M function

Format: Numerical value

Input: 0 to 9999

Code of the last (greatest) M function described in the parameter object.

No entry: The properties of the parameter object apply only to the M function defined in **MP_min.**

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

Transfer and acknowledgment of M function

In the **Program Run, Full Sequence** and **Program Run, Single Block** operating modes, the next NC block is not run until the PLC has acknowledged execution of the M function.

There are two possibilities for transferring the M strobe to the PLC and for acknowledgment by the PLC:

- Transfer and acknowledgment with **MP_signal**
  - Transfer of the M strobe: The PLC marker defined in **MP_signal** is set
  - Acknowledgment of the M strobe: The PLC marker defined in **MP_signal** is reset.
- Transfer with **MP_signal** and acknowledgment with **MP_acknowledge:**
  - Transfer of the M strobe: The PLC marker defined in **MP_signal** is set
  - Acknowledgment of the M strobe: The PLC marker defined in **MP_acknowledge** is set.

HEIDENHAIN recommends that you only use **MP_signal** for transfer and acknowledgment.

If **MP_signal** and **MP_acknowledge** are not defined, the data of the M strobe are saved without synchronization with the PLC program. The output is immediately acknowledged.

MP_signal

Symbolic name or number of the PLC marker that is set when the function is decoded.

Format: String

Input: Max. 80 characters
If you have not entered a value in the parameter **MP_acknowledge,** resetting this marker means an acknowledgment of the strobe.
No entry: The data connected with the output of the strobe is saved without synchronization with the PLC program and the output is immediately acknowledged.

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

MP_acknowledge

Symbolic name or number of the PLC marker that is set for acknowledging the strobe.
Available from NCK software version: 597 110-02.

Format: String

Input: Max. 80 characters
No entry: The strobe is reset with the PLC marker entered in **MP_signal.**

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

MP_code

Symbolic name or number of the PLC word marker for the M code

Format: String

Input: Max. 80 characters
Name of the PLC word marker in which the M code is transferred to the PLC
Example: **DG_M_Function_M10**

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

MP_data

Symbolic name or number of the PLC word marker for additional data

Format: String

Input: Max. 80 characters
Symbolic name or number of the PLC word marker in which the additional data of the M function is transmitted to the PLC.
The field size determined from the symbol defines up to how many data can be saved in the programmed sequence.
If an absolute number is given, only one value is saved.
If no value is entered, no data can be passed to the PLC.

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

MP_revoke

Numbers of M functions whose effect will be canceled by the output of the strobe.
Available from NCK software version: 597 110-02.

Format: Array [0...19]

Input: 0 to 9999
In the list, enter the numbers of the functions whose effect will be canceled when this code is output.
Example – Configuration of function M5: Calling function M5 cancels the effect of functions M3 and M4. In this example, you would enter the key names of the functions M3 and M4.
The parameter is effective for updating the status of the function in the status display and during block scan.
Note:
S outputs for the same spindle as well as T outputs and T2 outputs always cancel each other. However, in some cases they may also cancel other M functions. M functions cannot cancel the effects of S, T and T2 outputs.
If specific codes cancel each other, they cannot be combined in a strobe.

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

MP_split

The M function is split into the specified M functions
Available from NCK software version: 597 110-02.

Format: Array [0...1] (max. two M functions)

Input: The currently active M function can be split into the two specified M functions. You can cancel it in part by canceling one of the specified M functions. The currently active M function is split up if part of it is canceled by the **MP_revoke** parameter.
Example:
Mxx = Spindle ON clockwise (M3) + coolant ON (M8)
If function Mxx is active and an M5 is collected during the block scan, for example, the function M3 contained in Mxx is canceled. Only function M8 remains active. In this case, enter the key names "M3" and "M8" into the parameter.

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

MP_group

After a block scan, the collected M functions are implemented in the order of their group IDs (starting from the lowest).
Available from NCK software version: 597110-03.

Format: Numerical value

Default: Value optional

Access: LEVEL3

Reaction: RESET

MP_track

Automatic update of the active M functions
Available from NCK software version: 597 110-03.

Format: Selection menu

Selection: Normally the PLC program has to use the Module 9088 to implement the status of the active M function. Set this parameter to TRUE if the PLC runtime system is to implement the status automatically.
TRUE
The status of the active M function is implemented automatically, depending on the **MP_trackMState** parameter (CfgPlcOptions).
FALSE
The status of the active M function must be implemented by calling Module 9088.
No value:
The status of the active M function is implemented automatically, depending on the **MP_trackMState** parameter (CfgPlcOptions).

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

In **MP_singular**, M codes to be output in a separate strobe are defined.

MP_singular

Function is output in a separate strobe

Format: Selection menu

Selection: Use the parameter **MP_singular** to prevent the function from being output in a strobe together with other functions.
If several M, S, T functions are programmed in an NC block, the strobes can be set in parallel by the NC. With this parameter, you can exclude the current function from this parallel collection and simultaneous transmission in one strobe.
TRUE
Function is output in a separate strobe. Combined output is prevented.
FALSE
Function can be combined with other functions

Default: TRUE

Access: LEVEL3

Reaction: RESET

Execution at the beginning or end of block

In **MP_blockEnd,** you define whether the M function is to be executed at the beginning or end of block.

MP_blockEnd

Function output at block end

Format: Selection menu

Selection: **TRUE**
Function is executed at block end.
FALSE
Function is executed at beginning of block.

Default: FALSE

Access: LEVEL3

Reaction: RESET

Output of M code during block scan

Use the parameter **MP_blockSearch** to define whether the M code should also be output during the block scan.

MP_blockSearch

Function output also during block scan
Available from NCK software version: 597 110-02.

Format: Selection menu

Selection: **TRUE**
Function is also output during the block scan.
FALSE
Function is not output during the block scan. The function is collected and restored.

Default: FALSE

Access: LEVEL3

Reaction: RESET

Synchronization of the M function with the NC program

MP_sync defines the synchronization of the M function with NC program run.

MP_sync

Synchronization of function with the NC
Format: Selection menu
Selection: **SYNC_EXEC**
The function is synchronized with program run. The output of movement by the interpolator is stopped; then the function is executed.
Please note:
After this function has been executed, the look-ahead calculation continues using the position values that were active **before** execution of the function. This setting is not suitable for M functions executing PLC positioning movements, for example!
SYNC_CALC
The function is synchronized with program calculation. The interpretation of the NC program is stopped and the path is calculated internally; then the function is executed. The M function is completely executed. After successful execution has been acknowledged, calculation continues with new position values.
ASYNC
The function is output without synchronization.
Default: SYNC_EXEC
Access: LEVEL3
Reaction: RESET

Calling an NC macro with an M function

An NC subprogram can be executed instead of transferring an M function to the PLC. The path and name of the NC subprogram are entered in the parameter **MP_macro.**

M functions that call an NC subprogram are not sent to the PLC.

MP_macro

Call of an NC subprogram instead of the M function
Format: String
Input: Max. 260 characters
Path and name of the NC subprogram
Example: **`%OEM%\programs\MMacro10.ncs`**
Default: No entry, value optional
Access: LEVEL3
Reaction: RESET

Note

A maximum of six NC programs can be nested (subprograms, cycles, macros).

Adjust the M function menu

In the DIN/ISO mode of the **smart.Turn** Programming mode you can insert M functions directly into the NC program by means of an M Function menu (M menu). The structure and menu texts of the M menu are defined in language-sensitive files *.str under**PLC:\resource\MFktMenu** and can be adjusted to the available M functions.

Note

As the language of the M menu is not switched when the language is changed during run time, the control must be re-started in such a case.

A menu entry comprises:

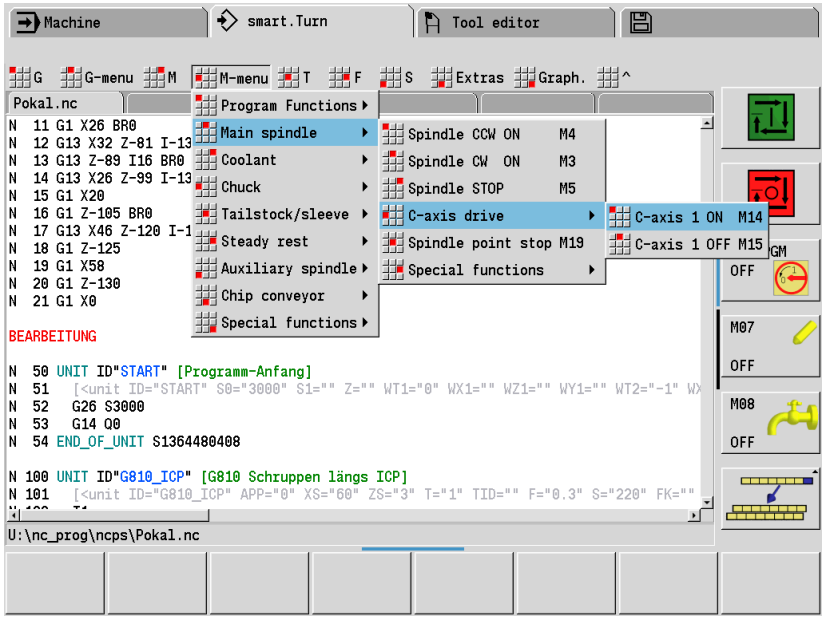
- **Menu position:** 3-digit number
  - 1. number: 1st menu level (1st pull-down menu)
    The number 1 is assigned to "Program Functions"—please use the numbers 2 to 9
  - 2. number: 2nd menu level—up to 9 entries under each entry of the 1st menu level
  - 3. number: 3rd menu level—up to 9 entries under each entry of the 2nd menu level
- **M number:** appears after the menu item has been selected
  - Further menu level follows: the M number has no significance
  - Negative M number: The menu item is visible, but not selectable
- **Menu text:** Menu item text (enclosed in quotation marks "....").
- **Comment:**
  - Follows the closing quotation mark of the menu text
  - Use a semicolon (";") to start the comment lines in the first column.
- Line feeds can be inserted at any location.

An excerpt of the **MFktMenu_de.str** file is shown below. The M menu resulting from this menu description is depicted in the figure following it.

Excerpt from a menu description (MFktMenu_de.str):

```
200,      0, "Main spindle"
210,      4, "Spindle CCW ON          M4"
220,      3, "Spindle CW ON           M3"
230,      5, "Spindle STOP            M5"
240,      0, "C axis drive"
241,     14, "C axis 1 ON             M14"
242,     15, "C axis 1 OFF            M15"
250,     19, "Spindle point stop      M19"
260,      0, "Special functions"
261,     12, "Index ON                M12"   Spindle brake ON
262,     13, "Index OFF               M13"   Spindle brake off
;
300,       0, "Coolant"
310,     108, "Circulation 1 ON       M108"
320,     107, "Circulation 2 ON       M107"
330,     109, "All OFF                M109"
340,      50, "Chuck jaw flushing ON    M50"
350,      51, "Chuck jaw flushing OFF   M51"
360,      49, "Lubrication pulse ON     M49"
;
400,       0, "Chuck"
...
```

This description creates the following menu:



8.5.3 Overview of M Functions of the MANUALplus 620

The M functions listed in the table below are executed/evaluated by the NC and then forwarded to the PLC for further execution. These M functions cannot be used for any other purposes.

M code	Function
M0	Programmed stop
M1	Selectable stop
M7	Coolant
M8	Coolant
M9	Coolant
M12	Spindle brake ON
M13	Spindle brake OFF
M14	Position C axis (assigned to spindle 1)
M15	Move C axis out (assigned to spindle 1)
M30	Program end without restart
M41 to M44	Gear ranges
M91	Program end without spindle stop (for retraction program during tool inspection)
M97	Multilateral synchronization (not with MANUALplus 620)
M99	Program end with restart
M114	Position C axis (assigned to spindle 2)
M115	Move C axis out (assigned to spindle 2)
M214	Position C axis (assigned to spindle 3)
M215	Move C axis out (assigned to spindle 3)
M314	Position C axis (assigned to spindle 4)
M315	Move C axis out (assigned to spindle 4)

The M codes listed in the following table "usually" have the described meaning.

M code	Function
M3	CW spindle rotation for spindle/screw 1
M4	CCW spindle rotation for spindle/screw 1
M5	Spindle stop for spindle/screw 1
M19	Position control for spindle/screw 1
M33	Open the spindle chuck while it is running
M36	Open the spindle chuck
M37	Close the spindle chuck
M103	CW spindle rotation for spindle/screw 2
M104	CCW spindle rotation for spindle/screw 2
M105	Spindle stop for spindle/screw 2
M119	Position control for spindle/screw 2
M203	CW spindle rotation for spindle/screw 3
M204	CCW spindle rotation for spindle/screw 3
M205	Spindle stop for spindle/screw 3
M219	Position control for spindle/screw 3
M303	CW spindle rotation for spindle/screw 4
M304	CCW spindle rotation for spindle/screw 4
M305	Spindle stop for spindle/screw 4
M319	Position control for spindle/screw 4

8.6 S Function (S Strobe)

The S function is channel-sensitive. S functions are configured in two steps:

- In the channel-sensitive parameter **MP_sStrobes,** you define a key name for the S function. In this way, you assign the S functions to a machining channel.
- In the parameter object **System/PLC/CfgPlcSStrobe,** you configure the S function.

Note

If the rotational speed is programmed within a G function, the S strobe will not be used. Then the rotational speed or constant cutting speed will be transmitted to the PLC in an M strobe.

Note

On the MANUALplus 620, the S strobe is used exclusively for parameter set switchover of the spindle gear ranges.

8.6.1 Assigning S functions to the machining channels

Settings in the configuration editor	MP number
Channel	
ChannelSettings	
[Key name of the channel]	
CfgPlcStrobes	
sStrobe	201602
unitOfMeasure	201605

MP_sStrobe

S strobe description of this machining channel
Format: Selection menu
Selection: Key name of the S strobe used in this machining channel
Default: –
Access: LEVEL3
Reaction: RESET

In the marker defined in **MP_unitOfMeasure**, the PLC is informed of the unit of measure used in the NC program when M, S, T or alias strobes are executed (See "MP_unitOfMeasure" on page 1148).

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcSStrobe	
[Key name of S strobe]	
type	104017
condition	104001
signal	104002
acknowledge	104003
spindleSpeed	104004
badSpeed	104005
cuttingSpeed	104018
spindleMode	104006
gearCode	104007
gearSpeed0	104008
gearSpeed1	104009
gearStop	104010
sCode	104011
revoke	104012
singular	104013
blockSearch	104014
sync	104015
syncGear	104016

Transfer and acknowledgment of S function

Use parameter **MP_condition** to define the conditions for the output of the S strobe.

There are two possibilities for transferring the S strobe to the PLC and for acknowledgment by the PLC:

- Transfer and acknowledgment with **MP_signal**
  - Transfer of the S strobe: The PLC marker defined in **MP_signal** is set.
  - Acknowledgment of the S strobe: The PLC marker defined in **MP_signal** is reset.
- Transfer with **MP_signal** and acknowledgment with **MP_acknowledge:**
  - Transfer of the S strobe: The PLC marker defined in **MP_signal** is set.
  - Acknowledgment of the S strobe: The PLC marker defined in **MP_acknowledge** is set.

HEIDENHAIN recommends that you only use **MP_signal** for transfer and acknowledgment.

If **MP_signal** and **MP_acknowledge** are not defined, the data of the S strobe is saved without synchronization with the PLC program. The output is immediately acknowledged.

MP_condition

Condition for sending the strobe to the PLC
Available from NCK software version: 597 110-02.

Format: Selection menu

Selection: **COND_ALWAYS**
Strobe is output with every programmed S code
COND_ST
Strobe is output only if spindle speed changes
COND_GP
Strobe is output only if gear range changes
If the parameter is missing from the configuration, the strobe is always output.

Default: COND_ST

Access: LEVEL3

Reaction: RESET

MP_signal

Symbolic name or number of the PLC marker that is set when the function is decoded.

Format: String

Input: Max. 80 characters
If you have not entered a value in the parameter **MP_acknowledge,** resetting this marker means an acknowledgment of the strobe.
No entry: The data connected with the output of the strobe is saved without synchronization with the PLC program and the output is immediately acknowledged.

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

MP_acknowledge

Symbolic name or number of the PLC marker that is set for acknowledging the strobe.
Available from NCK software version: 597 110-02.

Format: String
Input: Max. 80 characters
No entry: The strobe is reset with the PLC marker entered in **MP_signal.**
Default: No entry, value optional
Access: LEVEL3
Reaction: RESET

When the spindle speed or mode of rotation is changed in the **T, S, F menu** or the NC program, a strobe signal is generated and communicated to the PLC via the operand defined in **MP_signal**. At the same time the NC transmits the spindle speed in the PLC word defined in **MP_spindleSpeed** and the constant cutting speed in the PLC word defined in **MP_cuttingSpeed**. Depending on the value entered in the PLC word defined in **MP_spindleMode** for the spindle rotation, the PLC evaluates either **MP_spindleSpeed** (with constant spindle speed = 96) or **MP_cuttingSpeed** (with constant cutting speed = = 97).
If there are several spindles in an NC channel, you must enter in **MP_type** which spindle the S strobe is to refer to.
The PLC checks the programmed spindle speed in addition. The result of this check is saved in the PLC marker defined in **MP_badSpeed.** If the marker is set, the spindle speed is outside the permissible range. If **MP_badSpeed** is not defined, the spindle speed is not checked.

MP_type

If there are several spindles in an NC channel, you must enter in MP_type which spindle the S strobe is to refer to.
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0: S strobe refers to the main spindle
1: S strobe refers to spindle 1 (driven tool)
2: S strobe refers to spindle 2 (driven tool)
3: S strobe refers to opposing spindle
Default: 0
Access: LEVEL3
Reaction: RESET

MP_spindleSpeed

Symbolic name or number of the PLC word for transmitting the spindle speed
Available from NCK software version: 597 110-01.

Format: String
Input: Name of the PLC word in which the spindle speed is transmitted to the PLC
If the parameter is missing, the spindle speed cannot be read as a numerical value.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_badSpeed

Symbolic name or number of the PLC marker for impermissible spindle speeds
Available from NCK software version: 597 110-01.

Format: String
Input: Name of the PLC marker which is set if the spindle speed is outside the permissible range.
No entry: The spindle speed is not monitored.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_cuttingSpeed

Symbolic name or number of the PLC word for transmitting the constant cutting speed
Available from NCK software version: 597 110-01.

Format: String
Input: Name of the PLC word in which the cutting speed is transmitted to the PLC
If the parameter is missing, the constant cutting speed cannot be read as a numerical value.
Default:
Access: LEVEL3
Reaction: RESET

MP_spindleMode

Symbolic name or number of the PLC word for transmission of the spindle mode of rotation (96 for constant spindle speed, 97 for constant cutting speed). Depending on the value of the PLC operand defined in MP_spindleMode, the PLC evaluates either the content of MP_spindleSpeed or MP_cuttingSpeed.
Available from NCK software version: 597 110-01.

Format: String

Input: Name of the PLC word in which the spindle mode of rotation is transmitted to the PLC
If the parameter is missing, the constant cutting speed cannot be read as a numerical value.

Default:

Access: LEVEL3

Reaction: RESET

In **MP_singular,** the S codes that must be output in a separate strobe and must not be combined with other codes are defined.

MP_singular

Function is output in a separate strobe

Format: Selection menu

Selection: Use the parameter **MP_singular** to prevent the function from being output in a strobe together with other functions.
If several M, S, T functions are programmed in an NC block, the strobes can be set in parallel by the NC. With this parameter, you can exclude the current function from this parallel collection and simultaneous transmission in one strobe.

TRUE
Function is output in a separate strobe. Combined output is prevented.

FALSE
Function can be combined with other functions

Default: TRUE

Access: LEVEL3

Reaction: RESET

MP_revoke

Numbers of functions whose effect will be canceled by the output of the strobe.
Available from NCK software version: 597 110-02.

Format: Array [0...19]

Input: 0 to 9999
In the list, enter the numbers of the functions whose effect will be canceled when this code is output.
Example – Configuration of function M5: Calling function M5 cancels the effect of functions M3 and M4. In this example, you would enter the key names of the functions M3 and M4.
The parameter is effective for updating the status of the function in the status display and during block scan.
Note:
S outputs for the same spindle as well as T outputs and T2 outputs always cancel each other. However, in some cases they may also cancel other M functions. M functions cannot cancel the effects of S, T and T2 outputs.
If specific codes cancel each other, they cannot be combined in a strobe.

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

Parameter **MP_blockSearch** is used to define the behavior of the S function during block scan:

MP_blockSearch

S function output during block scan
Available from NCK software version: 597 110-02.

Format: Selection menu

Selection: **TRUE**
S function is also output during block scan.
FALSE
S function is not output during block scan The function is collected and restored.

Default: FALSE

Access: LEVEL3

Reaction: RESET

Gear shifting

You control the gear shifting through PLC outputs. The gear range is determined by the **MP_nominalSpeed** of each gear-range parameter set. The output of the gear range is defined in **MP_condition.** Use **MP_gearStop** to define if the spindle speed should automatically be reduced to 0 when shifting between gears.

- Configure a separate spindle parameter set for each gear stage.
- Use **MP_gearCode** to define a word marker, in which the gear range is transmitted to the PLC during decoding.

Note

Use the "KeySynonym" function to create new parameter sets rapidly and easily. (in the configuration editor under **KeySynonym** -> **CfgKeySynonym**). Only the first parameter set must be fully defined. All further parameters sets are "gated" to the first set **(MP_relatedTo),** so you only have to describe the differing parameters. For more information, please refer to „The KeySynonym Function" on page 289.

- In the list parameter **MP_gearSpeed0**, enter the names of spindle parameters sets for the operating mode 0. These parameter sets define the gear ranges for operating mode 0 if it is selected with PLC Module 9163, see "Switching the operating modes" auf Seite 706.
- Proceed in the same manner for operating mode 1 and enter the parameter sets in **MP_gearSpeed1.**

Each of the lists must be sorted in ascending order, with the smallest gear shaft speed at the top. Gear ranges are not supported if the list is missing or empty.

- Use the **CfgFeedLimits** configuration datum of a parameter set to define the minimum and maximum spindle shaft speed for each gear range.
- Use the parameter **MP_nominalSpeed (CfgFeedLimits)** to define the rated speed for each gear range.
- For digitally controlled spindles:
  Use the parameter **MP_gearShiftSpeed** to define the gear shifting speed for each gear range.
- For analog controlled spindles:
  Use the parameter **MP_gearShiftVoltage** to define for each gear range the nominal-voltage output for gear shifting.

The control selects the necessary gear range based on this shaft speed. The minimum and maximum shaft speeds of the individual gear ranges (**MP_minFeed** and **MP_maxFeed** parameters) may overlap.

MP_gearCode

Symbolic name or number of the PLC word for the gear range
Available from NCK software version: 597 110-01.

Format: String
Input: Name of the PLC word in which the gear range is transmitted to the PLC
No entry: No gear range is transmitted
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_gearSpeed0

Key names of parameter sets for gear ranges (operating mode 0)
Available from NCK software version: 597 110-02.

Format: Array [0...99]
Input: List with key names for spindle parameter sets for operating mode 0.
Default: –
Access: LEVEL3
Reaction: RESET

MP_gearSpeed1

Key names of parameter sets for gear ranges (operating mode 1)
Available from NCK software version: 597 110-02.

Format: Array
Input: List with key names for spindle parameter sets for operating mode 1.
Default: –
Access: LEVEL3
Reaction: RESET

MP_gearStop

Switch off the spindle when the gear range is changed
(cf. iTNC 530: MP3030 bit#1)
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: **TRUE**
If a strobe for changing the gear range is output, the control automatically switches the spindle off.
FALSE
If a strobe for changing the gear range is output, the spindle is not switched off.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_nominalSpeed

Rated speed for the gear range
Available from NCK software version: 597 110-02.

Format: Numerical value

Input: Shaft speed [rpm]
Enter the greatest programmable shaft speed at which this spindle parameter set is to be used. If a shaft speed greater than the given one is programmed, the next higher gear range is switched to.

Default: No value, parameter optional

Access: LEVEL3

Reaction: RUN

MP_sync

Synchronization of function with the NC

Format: Selection menu

Selection: **SYNC_EXEC**
The function is synchronized with program run. The output of movement by the interpolator is stopped; then the function is executed.
Please note:
After this function has been executed, the look-ahead calculation continues using the position values that were active **before** execution of the function. This setting is not suitable for M functions executing PLC positioning movements, for example.

SYNC_CALC
The function is synchronized with program calculation. The interpretation of the NC program is stopped and the path is calculated internally; then the function is executed. The M function is completely executed. After successful execution has been acknowledged, calculation continues with new position values.

ASYNC
The function is output without synchronization.

Default: SYNC_EXEC

Access: LEVEL3

Reaction: RESET

MP_syncGear

Synchronization of function with the NC if the gear speed changes

Format: Selection menu

Selection: **SYNC_EXEC**

The function is synchronized with program run. The output of movement by the interpolator is stopped; then the function is executed.

Please note:

After this function has been executed, the look-ahead calculation continues using the position values that were active **before** execution of the function. This setting is not suitable for M functions executing PLC positioning movements, for example.

SYNC_CALC

The function is synchronized with program calculation. The interpretation of the NC program is stopped and the path is calculated internally; then the function is executed. The M function is completely executed. After successful execution has been acknowledged, calculation continues with new position values.

ASYNC

The function is output without synchronization.

If the parameter is missing, synchronization is as given by the parameter **MP_sync.**

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

MP_gearShiftSpeed

Speed for gear shifting

Available from NCK software version: 597 110-03.

Format: Numerical value

Input: RPM [1/min] with up to 9 decimal places

Default: No value, parameter optional

Access: LEVEL3

Reaction: RUN

MP_gearShiftVoltage

Nominal voltage value output for gear shifting

Available from NCK software version: 597 110-03.

Format: Numerical value

Input: Voltage in volts [V] with up to 9 decimal places

Default: No value, parameter optional

Access: LEVEL3

Reaction: RUN

Coded output of spindle speed

Use the parameter **MP_sCode** to configure a word marker for the coded output of the spindle speed (S code). You must output the speed code to the spindle drive through PLC outputs.

If the speed code is changed, the NC sets the S strobe. If you acknowledge the S code with the marker defined in **MP_acknowledge,** the NC program is continued and the S strobe is reset by the NC.

If required, the programmed spindle speed is rounded off to the next standard value by the NC and given in S code as per DIN 66025 (see the S-code table below).

Speeds of 0 to 9000 min^{-1} are possible.

- Use the machine parameter **MP_sCode** to define a symbolic name or the number of a word marker, in which the S code of the spindle speed is transmitted to the PLC.

MP_sCode

	Word marker for the coded output of the spindle speed Available from NCK software version: 597 110-03.
Format:	String
Input:	Max. 80 characters Enter the symbolic name (API 3.0) or the number of a word marker (API 1.0), to which the S code of the spindle speed is written. Example: **NP_WG_S_Code** (API 3.0) **W258** (API 1.0)
Default:	No value, parameter optional
Access:	LEVEL3
Reaction:	RESET

S code table

S code	min^{-1}
S 00	0
S 01	0,112
S 02	0,125
S 03	0,14
S 04	0,16
S 05	0,18
S 06	0,2
S 07	0,224
S 08	0,25
S 09	0,28
S 10	0,315
S 11	0,355
S 12	0,4
S 13	0,45
S 14	0,5
S 15	0,56
S 16	0,63
S 17	0,71
S 18	0,8
S 19	0,9
S 20	1
S 21	1,12
S 22	1,25
S 23	1,4
S 24	1,6
S 25	1,8
S 26	2
S 27	2,24
S 28	2,5
S 29	2,8
S 30	3,15
S 31	3,55
S 32	4
S 33	4,5
S 34	5
S 35	5,6
S 36	6,3
S 37	7,1
S 38	8
S 39	9
S 40	10

S code	min^{-1}
S 41	11,2
S 42	12,5
S 43	14
S 44	16
S 45	18
S 46	20
S 47	22,4
S 48	25
S 49	28
S 50	31,5
S 51	35,5
S 52	40
S 53	45
S 54	50
S 55	56
S 56	63
S 57	71
S 58	80
S 59	90
S 60	100
S 61	112
S 62	125
S 63	140
S 64	160
S 65	180
S 66	200
S 67	224
S 68	250
S 69	280
S 70	315
S 71	355
S 72	400
S 73	450
S 74	500
S 75	560
S 76	630
S 77	710
S 78	800
S 79	900
S 80	1000
S 81	1120

S code	min^{-1}
S 83	1400
S 84	1600
S 85	1800
S 86	2000
S 87	2240
S 88	2500
S 89	2800
S 90	3150
S 91	3550
S 92	4000
S 93	4500
S 94	5000
S 95	5600
S 96	6300
S 97	7100
S 98	8000
S 99	9000

8.7 T Functions (T Strobe)

T functions are channel-sensitive. T functions are configured in two steps:

- In the channel-sensitive parameter **MP_tStrobes,** you define key names for the T functions. In this way, you assign the T functions to the machining channel.
- You configure the T functions in the parameter object **System/PLC/ CfgPlcTStrobe.**

8.7.1 Assigning T functions to the machining channels

Settings in the configuration editor	MP number
Channel	
ChannelSettings	
[Key name of the channel]	
CfgPlcStrobes	
tStrobes	201603
unitOfMeasure	201605

MP_tStrobes

T strobe description of this machining channel

Format:	Array [0..18]
Input:	Key name of the T strobes used in this machining channel
Default:	[0]: ToolCall0 [1]: ToolCall [2]: ToolDef
Access:	LEVEL3
Reaction:	RESET

In the marker defined in **MP_unitOfMeasure**, the PLC is informed of the unit of measure used in the NC program when M, S, T or alias strobes are executed (See "MP_unitOfMeasure" on page 1148).

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcTStrobe	
[Key name of S strobe]	
type	104101
condition	104102
signal	104103
acknowledge	104104
toolNumber	104105
toolIndex	104106
toolMagazine	104107
pocketNumber	104108
unloadTool	104109
externalTool	104110
internalTool	104111
specialTool	104112
revoke	104113
singular	104114
blockSearch	104115
sync	104116

Transfer and acknowledgment of T function

MP_type specifies the type of tool call.

Use the parameter **MP_condition** to define the conditions for the output of the T strobe.

There are two possibilities for transferring the T strobe to the PLC and for acknowledgment by the PLC:

- Transfer and acknowledgment with **MP_signal**
  - Transfer of the S strobe: The PLC marker defined in **MP_signal** is set.
  - Acknowledgment of the S strobe: The PLC marker defined in **MP_signal** is reset.
- Transfer with **MP_signal** and acknowledgment with **MP_acknowledge:**
  - Transfer of the S strobe: The PLC marker defined in **MP_signal** is set.
  - Acknowledgment of the S strobe: The PLC marker defined in **MP_acknowledge** is set.

HEIDENHAIN recommends that you only use **MP_signal** for transfer and acknowledgment.

If **MP_signal** and **MP_acknowledge** are not defined, the data of the S strobe is saved without synchronization with the PLC program. The output is immediately acknowledged.

The NC transfers the other data for the tool call in the PLC words defined in the following parameters:

- MP_toolNumber: Tool number (as specified in MP_Aggregates/ToolMount/ ...)
- MP_toolIndex: Number of the cutting edge if tools with several cutting edges are used
- MP_toolMagazine: Reserved
- MP_pocketNumber: Reserved

Machine parameters

MP_type

Type of T function
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **T0**
Remove tool from spindle
T1
Insert tool in spindle
T2
Prepare the next tool change
Default: –
Access: LEVEL3
Reaction: RESET

Parameter **MP_condition** specifies, whether a tool number transferred with a T strobe is accompanied by a transfer of the strobe to the PLC program.

However, the configuration for a T1 strobe dominates the configuration for a T0 strobe if a real exchange (T1 and T0 together) of the tool takes place.

MP_condition

Condition for sending the strobe to the PLC
Available from NCK software version: 597 110-02.

Format: Selection menu
Selection: **COND_ALWAYS**
Strobe is output with every programmed T code
COND_ST
Strobe is output only if tool number changes
COND_GP
Strobe is output only if pocket number changes
If the parameter is missing from the configuration, the strobe is always output.
Default: COND_ST
Access: LEVEL3
Reaction: RESET

MP_signal

Symbolic name or number of the PLC marker that is set when the function is decoded.

Format: String

Input: Max. 80 characters
If you have not entered a value in the parameter **MP_acknowledge,** resetting this marker means an acknowledgment of the strobe.
No entry: The data connected with the output of the strobe is saved without synchronization with the PLC program and the output is immediately acknowledged.

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

MP_acknowledge

Symbolic name or number of the PLC marker that is set for acknowledging the strobe.
Available from NCK software version: 597 110-02.

Format: String

Input: Max. 80 characters
No entry: The strobe is reset with the PLC marker entered in **MP_signal.**

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

MP_toolNumber

Symbolic name or number of the PLC word for transmitting the tool number
Available from NCK software version: 597 110-01.

Format: String

Input: Max. 80 characters
Name of the PLC word in which the tool number is transmitted to the PLC

Default: No value, parameter optional

Access: LEVEL3

Reaction: RESET

MP_toolIndex

Symbolic name or number of the PLC word for transmitting the tool index
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 80 characters
Name of the PLC word in which the tool index is transmitted to the PLC
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_toolMagazine

Symbolic name or number of the PLC word for transmitting the magazine number of the tool
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 80 characters
Name of the PLC word in which the magazine number of the tool is transmitted to the PLC
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_pocketNumber

Symbolic name or number of the PLC word for transmitting the pocket number of the tool
Available from NCK software version: 597 110-01.

Format: String
Input: Name of the PLC word in which the pocket number of the tool is transmitted to the PLC
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_unloadTool

Name or number of a marker that is set during decoding if no tool is loaded.
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 80 characters
The definition of this marker is not necessary if the T0 and T1 strobes are otherwise distinguished.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_externalTool

Name or number of a marker that is set during decoding if a tool is loaded that is not in the magazine.
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 80 characters
The definition of this marker is not necessary if the magazine and pocket numbers are evaluated elsewhere or are irrelevant.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_internalTool

Name or number of a marker that is set during decoding if a tool is loaded that is in the magazine.
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 80 characters
The definition of this marker is not necessary if the magazine and pocket numbers are evaluated elsewhere or are irrelevant.
Default: No value, parameter optional
Access: LEVEL3
Reaction: RESET

MP_specialTool

Name or number of a marker that is set during decoding if a special tool is loaded.
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 80 characters
Default: The definition of this marker is not necessary if no special tools are used.
Access: LEVEL3
Reaction: RESET

In **MP_singular,** the S codes that must be output in a separate strobe and must not be combined with other codes are defined.

MP_singular

Function is output in a separate strobe

Format: Selection menu

Selection: Use the parameter **MP_singular** to prevent the function from being output in a strobe together with other functions.
If several M, S, T functions are programmed in an NC block, the strobes can be set in parallel by the NC. With this parameter, you can exclude the current function from this parallel collection and simultaneous transmission in one strobe.
TRUE
Function is output in a separate strobe. Combined output is prevented.
FALSE
Function can be combined with other functions

Default: TRUE

Access: LEVEL3

Reaction: RESET

MP_revoke

Numbers of functions whose effect will be canceled by the output of the strobe.
Available from NCK software version: 597 110-02.

Format: Array [0...19]

Input: 0 to 9999
In the list, enter the numbers of the functions whose effect will be canceled when this code is output.
Example – Configuration of function M5: Calling function M5 cancels the effect of functions M3 and M4. In this example, you would enter the key names of the functions M3 and M4.
The parameter is effective for updating the status of the function in the status display and during block scan.
Note:
S outputs for the same spindle as well as T outputs and T2 outputs always cancel each other. However, in some cases they may also cancel other M functions. M functions cannot cancel the effects of S, T and T2 outputs.
If specific codes cancel each other, they cannot be combined in a strobe.

Default: No entry, value optional

Access: LEVEL3

Reaction: RESET

Parameter **MP_blockSearch** is used to define the behavior of the S function during block scan:

MP_blockSearch

S function output during block scan
Available from NCK software version: 597 110-02.
Format: Selection menu
Selection: **TRUE**
S function is also output during block scan.
FALSE
S function is not output during block scan The function is collected and restored.
Default: FALSE
Access: LEVEL3
Reaction: RESET

MP_sync

Synchronization of function with the NC
Format: Selection menu
Selection: **SYNC_EXEC**
The function is synchronized with program run. The output of movement by the interpolator is stopped; then the function is executed.
Please note:
After this function has been executed, the look-ahead calculation continues using the position values that were active **before** execution of the function. This setting is not suitable for M functions executing PLC positioning movements, for example.
SYNC_CALC
The function is synchronized with program calculation. The interpretation of the NC program is stopped and the path is calculated internally; then the function is executed. The M function is completely executed. After successful execution has been acknowledged, calculation continues with new position values.
ASYNC
The function is output without synchronization.
Default: SYNC_EXEC
Access: LEVEL3
Reaction: RESET

8.8 Alias Functions (Alias Strobe)

Alias functions are used to map control-specific functions on M functions.

In **CfgPlcStrobeAlias,** you can define channel-sensitive, but control-specific, functions. Alias functions are configured in two steps:

- In the channel-sensitive parameter **MP_aliasStrobes,** you define key names for the functions. In this way, you assign the functions to the machining channel.
- In the parameter object **System/PLC/CfgPlcStrobeAlias,** you configure the functions.

Note

The alias functions are mapped on M functions. In the parameter object **CfgPlcMStrobe,** you define the M functions used.

8.8.1 Assigning alias functions to the machining channels

Settings in the configuration editor	MP number
Channel	
ChannelSettings	
[Key name of the channel]	
CfgPlcStrobes	
aliasStrobes	201604
unitOfMeasure	201605

MP_aliasStrobes

List of implemented strobes in this channel
Available from NCK software version: 597 110-01.

Format: Array [0..18]

Input: Key names of the alias strobes used in this machining channel for the reproduction of control-dependent functions on a uniform M-function transfer to the PLC program.
Example: **GFUNC_CH1**

Default: –

Access: LEVEL3

Reaction: RESET

In the marker defined in **MP_unitOfMeasure**, the PLC is informed of the unit of measure used in the NC program when M, S, T or alias strobes are executed (See "MP_unitOfMeasure" on page 1148).

Settings in the configuration editor	MP number
System	
PLC	
CfgPlcStrobeAlias	
[Key name of alias strobe]	
type	104201
mCode	104202
mOffset	104203

You define data exchange between the NC and the PLC in the parameter object **CfgPlcStrobeAlias.** The organization of this data exchange varies depending on the control and is described in more detail in the Chapter „Data Transfer NC => PLC, PLC => NC" on page 1251.

MP_type specifies the type of call. In **MP_mCode,** you specify the M function on which the alias function is to be mapped.

MP_type

Type of alias function
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **FN19 (not with MANUALplus 620)**
Two values are transmitted synchronously from the NC program to the PLC.
FN29 (not with MANUALplus 620)
Up to eight values are transmitted asynchronously from the NC program to the PLC.
CYCLE13 (not with MANUALplus 620)
Define spindle position for M19
TCHPROBE
Call measuring cycles
GFUNCTION
G functions are transmitted to the PLC. The G functions, including the function parameters, are transferred in an M strobe.

Default: FN19

Access: LEVEL3

Reaction: RESET

MP_mCode

Number of the M function
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to 9999
Number of the M function for which the control-dependent function is mapped.
Default: 0
Access: LEVEL3
Reaction: RESET

If the GFUNCTION type is defined and **MP_mOffset** = TRUE, the difference between the M functions and G functions is indicated by the entry **MP_mCode** for the M strobe:

- Number in M strobe **<** Entry**MP_mOffset:** M function is defined
- Number in M strobe **>** Entry **MP_mOffset:** G function is defined

MP_mOffset

Transferred M code is offset
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
The first numerical value transferred is used as an offset and entered in the parameter MP_min of the associated M function. The remaining numerical values are written to the double word entered in MP_data.
FALSE
No offset is used. The control always issues the M function given under MP_min. Both transferred numerical values are written to the array of double words entered in MP_data.
Default: FALSE
Access: LEVEL3
Reaction: RESET

8.9 User-Defined Cycles

OEM cycles (G5xx)

With the MANUALplus 620 you can realize OEM cycles. The G functions G500 to G590 are intended for this purpose. When such G functions are called, corresponding subprograms named "_G5xx.ncs" are used, which are saved in the **TNC:\nc_prog\ncps** folder.
The following must be noted for the creation of OEM cycles:

- Dialogs are defined in the **PLC:\resource\formdlg\g_oem.fdxml** file
- The dialog texts must be written in a selected language directly in the dialog description. Language-sensitive calls via a TextID are currently not yet possible
- The dialogs can be combined with images that are independent of any orientation. A separate image list containing all images of this function is defined for every G function at the beginning of the **g_oem.fdxml** file. The image list also includes the paths to the storage locations of every image. In the default setting, the images are saved under **PLC:\resource\formdlg\MpE-dit\common_gfkt\..**
- The formats already defined in the **PLC:\resource\formdlg\pformats.xml** file can be used for the parameters
- The following designations can be used for parameters G500...G590: _A, _B, _C, _D, _E, _F, _H, _I, _J, _K, _O, _P, _Q, _R, _S, _T, _U, _V, _W, _X, _Y, _Z
- It must be considered that the transfer value for #_X is transferred divided by 2, as it usually is a diameter value.
- The following G5xx programs are predefined:
  G593 – Feed rate per tooth for spindle 6
  G595 – Feed rate per revolution for spindle 6
  G596 – Constant cutting speed for spindle 6
  G595 – Rotational speed for spindle 6

PLC-G functions (G6xx)

In addition to the OEM cycles (G5xx), the PLC-G functions G602 to G699 can be defined with the MANUALplus 620. Unlike OEM cycles, the PLC-G functions do not require subprograms but are processed directly in the PLC.

The same rules as for the OEM cycles (see above) apply to the creation of dialogs with images.

The G functions G600 and G601 are reserved for tool preselection and tools from the magazine. The following parameters can be defined for G602...G699: _X, _Y, _Z, _H, _Q, _C, _F, _S, _T

8.10 Tables

Different types of tables are managed in the control, such as motor tables, datum tables and tool tables.

Every table has its own structure. This means that the number of columns, the column designations, the dialogs for column entries, etc. vary depending on the table.

Each table type also has certain characteristics. A table type is identified by its file extension, e.g. ".T" for tool tables. Therefore, tables with the same extension have the same characteristics.

Note

Tables of different types, meaning tables with different extensions, are not compatible with each other. This means that you cannot copy tables from one extension to another, or simply change extensions.

Table characteristics are set in the configuration editor. The different table characteristics are defined in the machine-parameter object **CfgTableProperties,** and the column characteristics in the object **CfgColumnDescription.**

8.10.1 Table Types of the MANUALplus 620

Different types of tables for different tasks are saved in the control as a standard feature. The functions of the individual tables, broken down by partitions, are described in the following.

User tables

User tables (**TNC:\table**) primarily contain the parameters for tool management.

- The table **add_cor1.hac** contains the tool-independent compensation values D901...D916. These compensation values are added to the active wear compensation values of the tools and are activated with G149. Switch-off is done with G149 D900, program cancelation or program end.
- The table **techdata.hte** contains the technology data based on the criteria of material, cutting material and machining mode. In its standard version, you can store the cutting data for 9 workpiece-material/tool-material combinations in the technology database. Each workpiece-material/tool-material combination includes the cutting speed, the main and secondary feed rates, and the infeed for 16machining modes.
- The turret assignment table **ToolAllo.tch** depends on the machine configuration. HEIDENHAIN provides a configuration for a turret with 3*12 pockets plus Multifix, which can also be used for smaller systems (e.g. turret with 1*8 pockets). If the free T numbers in the tool carrier configuration changes, the turret assignment table also needs to be adjusted. The tools for the individual turret pockets can be selected from the tool table **toolturn.htt** simply by pressing a key.
- In the datum table **zero_po1.hzp** the datums for all axes of a channel relative to the workpiece spindles and C axis are entered.

Table name	Group	Meaning
add_cor1.hac	USR	Additive compensations (D corrections)
mach_dim.hmd		Machine dimensions (not for MANUALplus 620)
techdata.hte		Technology data
ToolAllo.tch		Turret assignment table
toolturn.htt		Tool table
to_hold.hld		Tool holder table (not MANUALplus 620)
to_text.mxt		Tool texts
zero_po1.hzp		Datum table

OEM tables

OEM tables are stored in the **PLC:\table** directory.

- The **axis_c1.com**, **axis_x1.com** and **axis_z1.com** tables contain parameters for axis compensation and must be assigned explicitly to the individual axes in the **config.cma** table.
- Machining data is saved in the two following tables: The **ch_pro.mch** table contains the values for the feed per minute and revolution (G94, G95) and the **sp_pro.msp** table the values for the angle upon spindle stop (M19), for constant surface speed (G96), constant spindle speed (G97) and the speed limitation (G26). The data is entered into the table by the MMI and activated again after start-up of the control. The data is taken over in automatic mode in NC Stop condition.
- If motors are used whose parameters are not contained in the standard table (SYS:\table\MOTOR.MOT, write-protected), these can be entered in the table **MOTOR_OEM.MOT**.
- The PLC error message table **plctestpgm.pet** is on the control for test purposes only. In shipping condition it is replaced by a freely selectable table or the **err_tab.pet** error message table of the HEIDENHAIN PLC basic program (See "PET table (PLC error table)" on page 919).
- To be able to access SQL tables from the NC program via a table number, the SQL tables are assigned numbers in the **SqlRef.hsh** table.

Table name	Group	Meaning
axis_c1.com	OEM	Axis compensation for C Axis
axis_x1.com		Axis compensation for X axis
axis_z1.com		Axis compensation for Z axis
config.cma		Compensation value table for all axes
ch_pro.mch		Current values for G94 and G95
conv_oem.hco		OEM conversion list for M Functions
cool_cir.hcc		Coolant (M functions for switching on and off a maximum of 8 coolant circuits)
MOTOR_OEM.MOT		Motor data (table is empty; default table on PLC:\table)
PlcTestPgm.pet		Test PLC error message table
pos_corr.hpc		Position compensations (not MANUALplus 620)
sp_pro.msp		Current values for M19, G96, G97 and G26
SqlRef.hsh		Table numbering for table accesses from the NC program

System tables

The write-protected directory **SYS:\table** contains the control's system tables.

- In the **conv1.hc** table, converted G and M functions are assigned to the original G and M functions (e. G. for rear-side machining). The conversion list is assigned to a tool holder by entering the number of the conversion list in **MP_convTblNr** (e.g. MP_convTblNr = 1 for conv1.hc).
- The **inverter.inv** table contains important parameters of the power modules (See "Entries in the power module table (inverter.inv)" on page 751).

Note

The power module table of older HEIDENHAIN contouring controls, **motor.amp**, is no longer supported by the control!

- **JhNcError.pet** error message table
- The HEIDENHAIN standard motor table, **MOTOR.MOT**, contains important motor parameters (See "Entries in the motor table (motor.mot)" on page 751).

Table name	Group	Meaning
conv1.hc	SYS	Heidenhain conversion list for M and G functions
inverter.inv		Power module table
JhNcError.pet		PLC error message list
MOTOR.MOT		Motor data
simu.hsi		Colors, chucking equipment (not MANUALplus 620)

Settings in the configuration editor	MP number
System	
Path	
CfgTablePath	
path	102501
CfgOemPath	
oemTable	102001
ProgramManager	
CfgFileType	
unitOfMeasure	102901
standardEditor	102902
fileSize	102903
alternateEditor	102904
softkeyIcon	102905
softkeyIconVariant	102906
protect	102907
TableSettings	
CfgTableProperties	
[Key name of the table]	
columnKeys	105501
primaryKey	105502
Columns	
[Key name of the table column]	
CfgColumnDescription	
width	105601
unit	105602
initial	105603
minimum	105604
maximum	105605
charset	105606
unique	105607
readonly	105608
unitIsInch	105609
CfgColumnText	
dialogText	
dialogRes	105701
text	105701.501
info	105701.502
softkeyIcon	105701.503
iconVariant	105702
choice	
value	
dialog	
dialogRes	105703
text	105704
info	105705
lockValue	
value	105705.401
dialog	105705.402

Proceed as follows to create a new table type:

- Define a new file extension (System/ProgrammManager/CfgFileType).
- Create a table configuration (System/TableSettings/CfgTableProperties).
- Create a column configuration (System/TableSettings/Columns).
- Create a new table using the file manager.
- Insert rows into the table with the table editor.

Specifying the file extension

The extension determines the type of table. The following constraints apply:

- Maximum length of three characters
- Only numbers or capital letters are permitted (this means no blank spaces or special characters).
- Do not use extensions that are already used for other tables or files. In the configuration editor, under System/ProgramManager/CfgFileType and System/TableSettings/CfgTableProperties check whether the desired extension appears. The desired extension should not already appear here.

The new file extension must be entered in the configuration editor so that a table with this extension can be opened by the table editor:

- Enter the code number 95148.
- Press the CONFIG DATA soft key.
- Select the object **System/ProgramManager/CfgFileType.**
- Press the INSERT soft key.
- Select **MP_unitOfMeasure.**
  If you want to be able to choose if the table or file is to contain values in mm or in inches (specified by soft key when opening a new file), enter "UNIT_MMINCH."
- Select **MP_standardEditor.**
  Select the input value TABLE EDITOR.
- Activate the parameters **MP_alternateEditor** and **MP_fileSize** as optional parameters.
  MP_alternateEditor defines an alternative editor for files whose size exceeds the size defined in **MP_fileSize.**
- Activate the optional parameters **MP_softkeyIcon** and **MP_softkeyIconVariant.**
  An icon can be used instead of a text in the table editor if, in addition, the parameter **MP_choice** (TableSettings/Columns/CfgColumnText) is defined.
- Optionally, use **MP_protect** to protect the file type from editing.
  Disables filtering or editing of a file type.
- Save the information with the END or SAVE soft key.

MP_unitOfMeasure

Alternative unit of measure for file/table
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **UNIT_INDEPENDENT**
Input without unit of measure
UNIT_MM
Input in mm
UNIT_INCH
Input in inches
UNIT_MMINCH
Input in mm or inches
Default: UNIT_INDEPENDENT
Access: LEVEL3
Reaction: NOTHING

MP_standardEditor

Editor for file/table
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TEXT EDITOR**
Opens the text editor of the control when a file is selected.
PROGRAM EDITOR
Opens the NC program editor of the control when a file is selected.
TABLE EDITOR
Opens the table editor of the control when a file is selected.
HELP VIEWER
Opens the HTML HELP VIEWER (Mozilla) of the control when a file is selected.
PATTERN EDITOR
Opens the point-pattern editor of the control when a file is selected.
Default: TEXT EDITOR
Access: LEVEL3
Reaction: NOTHING

MP_fileSize

File size from which the alternate editor (MP_alternateEditor) is used
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: Value in kilobytes [KB] e.g. 100
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

MP_alternateEditor

Alternate editor used for files starting from the size in MP_fileSize
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TEXT EDITOR**
ASCII editor of the control
PROGRAM EDITOR
NC program editor of the control
TABLE EDITOR
Table editor of the control
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

MP_softkeyIcon

Path / file name of a soft-key icon
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 500 characters
The path and file name of an icon may be used instead of a text. The icon is used for the generic soft key in the table editor if, in addition, the parameter **MP_choice** (TableSettings/Columns/CfgColumnText) is defined.
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

MP_softkeyIconVariant

Number of a soft-key-icon variant
Available from NCK software version: 597 110-01.

Format: Numerical value
Input: 0 to 100
A variant number can be entered in addition to the path and the file name defined in **MP_softkeyIcon.**
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

MP_protect

Lock specific file types
Available from NCK software version: 597 110-04.

Format: Selection menu
Selection: **LOCK_OFF:**
File type not locked
LOCK_SOFTKEY
Lock the **SELECT TYPE** soft key for selection of the file type
LOCK_FILETYPE
Lock file types from editing
LOCK_ALL
Lock the **SELECT TYPE** soft key for selection of the file type and lock the file type to prevent editing
Default: LOCK_OFF
Access: LEVEL3
Reaction: NOTHING

Table description

Specify in the configuration editor which columns are used in a table.

- Enter the code number 95148.
- Press the CONFIG DATA soft key.
- Select the object **System/TableSettings/CfgTableProperties.**
- Press the INSERT soft key.
- Enter the extension of the new table;
  Specify the file (memory file) in which the configuration data are to be saved (normally PLC:\config\oemtable.cfg)
- Confirm with OK.
- Select **MP_columnKeys.**
- Enter the first column name in the field [0];
  The name is in the following format: Tableextension.Columnname.
  The **column name** must not have any other periods, commas or special characters. Hyphens are permitted.
  For columns used with the same meaning in different tables, only the column name needs to be given. Examples of such columns are the columns predefined by HEIDENHAIN, such as **NAME** or **NR**.
  The table extension must always be entered in uppercase letters.
- Use the INSERT soft key to create more columns according to the above procedure.
- Confirm with OK.
- Select **MP_primaryKey** and assign the column name (name from **MP_columnKeys**).
- Confirm with OK.
- Press the SAVE soft key to save the data.

MP_columnKeys

	List of column names Available from NCK software version: 597 110-01.
Format:	Array [0...]
Input:	xxx.xxx Name of the column in uppercase letters in the format <table extension>.<column name>; Max. 20 characters
Default:	-
Access:	LEVEL3
Reaction:	NOTHING

MP_primaryKey

Name of the column, based upon which the data is sorted in ascending order
Available from NCK software version: 597 110-01.

Format: String
Input: <Column name>
Max. 18 characters
Name of the column must also be entered in **MP_columnKeys.**
Default: -
Access: LEVEL3
Reaction: NOTHING

MP_foreignKey

Specify a character string of the type <column name><blank><referential action> for each list item.

Format: Array [1...79]
Input: Valid values for <referential action> are NO ACTION, RESTRICT, SET NULL, SET DEFAULT and INHERIT.
Max. 40 characters
Default: -
Access: LEVEL3
Reaction: NOTHING

MP_modificationKey

Enter the key name of the column in which the timestamp is to be entered if the line is modified.

Format: String
Input: <Table extension>.<column name>;
Max. 18 characters
Default: -
Access: LEVEL3
Reaction: NOTHING

Column description Description of the individual columns takes place in the configuration editor.

- Enter the code number 95148.
- Press the CONFIG DATA soft key.
- Select the object **System/TableSettings/Columns.**
- Press the INSERT soft key.
- Enter the name of the new column (see column name under table description).
- Select the object **System/TableSettings/Columns/ CfgColumnDescription.**
- Press the INSERT soft key; confirm the suggestion with OK.
- Select **MP_width;** enter the max. column width (number of characters).
- Select **MP_unit;** enter the unit of measurement.
- Select **MP_initial** (optional parameter); enter the default value.
- Select **MP_minimum;** enter the minimum value for numerical values.
- Select **MP_maximum;** enter the maximum value for numerical values.

Note

It is essential that you enter the minimum and maximum values!

- Select **MP_charset** (optional parameters); defines the number of permissible characters. If the number is not defined, all characters are allowed.
- Select **MP_unique** (optional parameter); insert this parameter if the column is to contain unambiguous values. If the attribute is not defined, the same values may appear more than once in different rows.
- Select **MP_readonly;** protect data from access; The column representing the "primaryKey" of the table should be write-protected (set MP_readonly to TRUE).
- Select **MP_ unitIsInch;** specify the unit of measurement.
- Select **MP_choice** (optional);
  A list of value/text pairs can be defined here. Only these texts are then available in the table via a selection list (in the **Programming** mode of operation, the COLUMN NAME soft key opens the selection list). This stands for a value, which is then displayed. The text for the selection list can be entered directly (MP_text), or you can enter a link to a dialog table (MP_dialogRes).
- Select **MP_value** (optional); enter the value for MP_dialog.
- Select **MP_dialog** (optional);
- **MP_dialogRes** (optional); leave the attribute empty if the text is not to be language-sensitive.
- Select **MP_text** (optional); enter the text for MP_value.
  (keep MP_width in mind)
- Select **MP_lockValue** (optional);
  If the value entered in the column equals the value given here, the text in MP_text is displayed. It can no longer be edited. This way editing can be disabled depending on the value.

- Select **MP_value** (optional); enter the value for MP_dialog.
- Select **MP_dialog** (optional);
- (MP_dialogRes in preparation);
- Select **MP_text** (optional); enter the text for MP_value.
  (keep MP_width in mind)
- Select **MP_CfgColumnText**; enter the dialog text for the columns.
- Carry out the above procedure for all new columns, and save the information with the SAVE soft key.

The following machine parameters are used for defining the columns.

MP_width

	Column width Available from NCK software version: 597 110-01.
Format:	Numerical value
Input:	2 to 50 (column width of max. 50 characters)
Default:	2
Access:	LEVEL3
Reaction:	NOTHING

MP_unit

Unit of measurement for column entries
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection:
TEXT
Text entry
SIGN
Algebraic sign + or –
BIN
Binary number
DEC
Decimal, positive, whole number
(cardinal number)
HEX
Hexadecimal number
INT
Whole number
LENGTH
Length
FEED
Feed rate (mm/min or 0.1 ipm)
IFEED
Feed rate (mm/min or ipm)
FEED_CUT
Cutting speed in m/min or feet/min
FEED_ROT
Feed rate in mm/revolution or inch/revolution
FLOAT
Floating-point number
BOOL
Logical value
INDEX
Index with subindices
TSTAMP
Time/Date

Default: TEXT
Access: LEVEL3
Reaction: NOTHING

MP_initial

Value automatically entered in a column when a new table is created (optional).
Available from NCK software version: 597 110-01.

Format: String
Input: Max. 50 characters
NULL: No default value. This column may be left blank.
Value: Default value. When a new row is inserted, this value is assigned as a default to the column.
If a default value other than NULL is given, then a valid value must always be entered in the column.
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

MP_minimum

Smallest permissible input value
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: Max. 50 characters
e.g.: –99999.9999
The minimum value is considered only for the columns with numerical values or logical values. It defines the smallest permissible numerical input value or the text representing the logical value FALSE. For values of the FLOAT, FEED, IFEED or LENGTH data types, the given number of decimal places determines the number of decimal places used for values in this column, e.g. 0.001 means 3 decimal places.

Default: No value, parameter optional

Access: LEVEL3

Reaction: NOTHING

MP_maximum

Largest permissible input value
Available from NCK software version: 597 110-01.

Format: Numerical value

Input: Max. 50 characters
e.g.: 99999.9999
The maximum value is considered only for the columns with numerical values or logical values. It defines the largest permissible numerical input value or the text representing the logical value TRUE. For values of the FLOAT, FEED, IFEED or LENGTH data types, the given number of decimal places determines the number of decimal places used for values in this column, e.g. 300 000 means 3 decimal places.

Default: No value, parameter optional

Access: LEVEL3

Reaction: NOTHING

MP_charset

Permissible number of characters for text columns (optional)
Available from NCK software version: 597 110-01.

Format: String

Input: Max. 224 characters
The number of permissible characters is evaluated only for text columns. If this parameter is not defined, all characters are allowed; otherwise, only the characters listed here are allowed.

Default: No value, parameter optional

Access: LEVEL3

Reaction: NOTHING

MP_unique

Defines whether only unambiguous values are allowed in the column (optional)
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **FALSE**
Values may occur more than once
TRUE
Only unambiguous values allowed
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

MP_readonly

Write protection on column entry
If the attribute is set to TRUE, the value assigned when inserting the line cannot be changed. If the attribute is not set or set to FALSE, values may be overwritten.
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **FALSE**
Values may be overwritten
TRUE
Values are write-protected
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

MP_unitIsInch

Values in inches (optional)
If lengths and feed rates are to be specified in the column in a definite unit of measure, enter TRUE here for values in inches and FALSE for values in mm. If the attribute is not set, the unit of measure is taken from the corresponding table.
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **FALSE**
Column entry in mm
TRUE
Column entry in inches
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

If, in the table editor, you want to display language-sensitive texts for columns, you must insert a CfgColumnText data object. However, this object is not absolutely necessary. If the object is missing, the column name is shown in the dialog line in the table editor. When inserting, the same column name must be given as in CfgColumnDescripton.

Settings in the configuration editor	MP number
System TableSettings CfgTablePrototype [Key name of the table extension] **path** **enableReset**	 106101 106102

A prototype table can be defined for every table type. If the machine operator creates a new table in the file manager and a prototype of the table is available, the properties of the prototype are automatically assigned to the new table. The prototype table is copied and serves as a basis for the new table.

HEIDENHAIN recommends using the name "prototype" as file name for the prototype. The file extension depends on the respective table type. The name of a prototype file for the pocket table would be **prototype.tch**.

The prototype-table files are saved in the **PLC:\proto\table** directory on the PLC: partition of the control.

Table prototypes are useful, for example, for the automatic assignment of default values to a new pocket table created in the file manager.

Use **MP_path** to define the path of the table prototype:

MP_path

	Path/name for the prototype of a table type Available from NCK software version: 597 110-02.
Format:	String
Input:	Max. 80 characters HEIDENHAIN recommends: Storing the prototypes in the PLC:\proto\table directory on the PLC: partition. There you will also find the prototype of the pocket table preconfigured by HEIDENHAIN.
Default:	For pocket table: **%OEM%\proto\table\prototype.tch**
Access:	LEVEL3
Reaction:	RUN

Use the parameter MP_enableReset to define whether the current table may be overwritten by the prototype. You must set this parameter to **TRUE** if you want to reset the pocket table with the **RESET POCKET TABLE** soft key, i.e. if you want to overwrite it with the prototype.

MP_enableReset

	Reset current table?
	Available from NCK software version: 597 110-02.
Format:	Selection menu
Selection:	TRUE
	Table may be overwritten with the prototype
	FALSE
	Table must not be overwritten with the prototype
Default:	For pocket table: TRUE
Access:	LEVEL3
Reaction:	RUN

8.10.4 Defining the path for OEM tables

Use the machine parameter MP_oemTable to define the path for OEM tables. OEM tables include tables such as the *.CMA and *.COM compensation-value tables for axis-error compensation or the *.MOT OEM motor table. With **MP_oemTable**, you inform the control about the location where the OEM tables are stored. The default setting %OEM%\table is entered in the parameter.

MP_oemTable

	Path for OEM tables Available from NCK software version: 597 110-01.
Format:	String
Input:	Max. 260 characters
Default:	%OEM%\table
Access:	LEVEL3
Reaction:	RESET

8.10.5 Symbolic names for tables

For access via SQL commands, tables are identified with a symbolic name and a file name including the path for the table characteristics.
Direct use of these paths, such as from cycles, has the disadvantage that if the drives or paths are changed, or if other tables are selected, the cycles must be changed.

In order to avoid this disadvantage, symbolic table names are used. These table names are place holders for the actual table name and path. When accessing a table, the control replaces the symbolic table name with the real table path and name.

Symbolic table names are saved in the control's configuration data in the **CfgTablePath** object (new key name). A symbolic table name should consist only of capital letters.

The logic names do not have to be in any certain format. Any name can refer to any table or table type.

The table being referenced does not even have to exist at the time that the logic table name is given. It can also be created afterwards.

Proceed as follows for specifying a symbolic table name:

- Enter the code number 95148.
- Press the **CONFIG DATA** soft key.
- Select the **System/Paths/CfgTablePath** object
- Press the **INSERT** soft key.
- Enter the symbolic table name (key name) and the storage file (e.g. the file **PLC:\config\oemtable.cfg**) and confirm with OK.
- Define the storage location of the table in **MP_path**.
- Press the **SAVE** soft key to save the data.

MP_path

	Path for tables Available from NCK software version: 597 110-01.
Format:	String
Input:	Max. 80 characters Path/name consisting of device name, up to 6 directories, file name and extension Example: **%USR%\table\tool.t**
Default:	–
Access:	LEVEL3
Reaction:	NOTHING

8.10.6 Editing tables via the PLC

You can also read tables and overwrite individual fields via PLC modules.

Note

The following modules must be called in a submit job or spawn job.
When entering the column names, pay attention to the case of the letters (whether they are small or capital).

Access via PLC module to the tables of the system partition is read-only!

Module 9240 Open a file
The module opens the file for access via the PLC. The "file handle" is created. This is a number which must be given for each subsequent access (such as in another PLC module).

Files should not be kept open unnecessarily, since they cannot be erased by the file manager during this time.
Ending a process (EM in a submit job) also closes all files opened by the process. The same applies if a process is canceled by a CAN instruction or by a renewed compiling of the PLC program.

The file handle must be saved in a double word.

Up to eight files may be open at once. However, the file can only be accessed by the process that opened it (SUBMIT job or SPAWN job). A file can also be opened more than once. If you want to prevent the file from being opened by more than one process, use the "lock file" mode.
To maintain a high processing speed, the file should be opened with the "BUFFERED" option for reading and writing ASCII texts. In this mode a part of the file is buffered in the main memory. This mode is not permitted for tables.

To append data to an existing file, set bit 0 = 1 (reading and writing) **and** bit 2 = 0 (record oriented).

Call:

PS	B/W/D/K	<>Mode> Bit#0 = 0: Read only Bit#0=1: Read and write Bit#1=0: Do not lock file Bit#1 = 1: Lock file Bit#2 = 0: Record oriented (for tables) Bit#2 = 1: Buffered (for ASCII files)
PS	B/W/D/K/S	<>String with file name> Complete path, file name and extension
CM	9240	
PL	D	<>File handle> Number for use in other modules –1: Error code in NN_GenApiModuleErrorCode

Error recognition:

Marker	Value	Meaning
NN_GenApiModule ErrorCode	1	Impermissible mode
	3	Incorrect string number
	7	File could not be opened
	20	Module was not called in a spawn or submit job

Module 9241 Close a file

With this module you close a file that has been opened with Module 9240. You must close the file in the process (submit job or spawn job) in which you opened it.

Call:

PS D <>File handle>
Number from Module 9240

CM 9241

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	File was closed
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Incorrect file handle
	20	Module was not called in a spawn or submit job

Module 9242 Positioning in a file

With this module you change the position of the cursor in a file opened with Module 9240. The new position is provided as result from Module 9242.
If the file was opened in the "record oriented" mode (tables), the cursor is positioned line by line.
If the file was opened in the "buffered" mode, the cursor is positioned character by character.
If you indicate a position before the beginning or after the end of the file, the cursor is positioned to the beginning or end of the file, respectively. The addressing of the new position is relative to the beginning or end of the file, or to the current position. You can interrogate the current position by transferring the position value zero relative to the current position.

Call only in a submit job or spawn job.

Call:

PS	D	<>File handle> Number from Module 9240
PS	B/W/D/K	<>Desired position>
PS	B/W/D/K	<>Mode> 0: Position relative to the file beginning 1: Position relative to the current position 2: Position relative to the file end
CM	9242	
PL	B/W/D/K	<>New position> –1: Error code in NN_GenApiModuleErrorCode

Error recognition:

Marker	Value	Meaning
NN_GenApiModule ErrorCode	1	Impermissible mode
	2	Incorrect file handle
	7	File system error
	20	Module was not called in a spawn job or submit job

Module 9243 Read from an ASCII file line by line
To read from a table, use Module 9245.

The module reads a line from the ASCII file opened with Module 9240,

and writes it to a PLC string.
Different processing times will result depending on whether you opened the file with the "buffered" option (buffered is faster).

The module reads up to a line break (line feed, '\n'), with a maximum of 127 characters. The line break is not saved in the result string, but is counted for the number of characters read.

The result is undefined when reading non-ASCII-coded files.

A certain amount of binary data is saved in the target string, but cannot be used.

Call:

PS	D	<>File handle> Number from Module 9240
PS	B/W/D/K	<>String number with result> 0 to 7
CM	9243	
PL	B/W/D	<>Number of read bytes> >0: Line has been read 0: File end has been reached –1: Error code in NN_GenApiModuleErrorCode

Error recognition:

Marker	Value	Meaning
NN_GenApiModule ErrorCode	2	Incorrect file handle
	3	Incorrect string number
	7	File system error
	20	Module was not called in a spawn job or submit job

Module 9244 Write to an ASCII file line by line

To write to a table, use Module 9246.

The module writes a line from a PLC string to an ASCII file already opened by Module 9240 in "buffered" mode.

If file is opened in "buffered" mode:

- Processing time is shorter.
- Files are saved to the hard disk only if more than 512 bytes are overwritten in several calls, or if the file is closed.
- The number of data specified in the transfer string is overwritten.

If file is opened in "record oriented" mode:

- Processing time is longer.
- The data is immediately saved to the hard disk.
- Exactly one line is overwritten. If there is a difference in length, the subsequent data is displaced by the difference.

Call:

PS	D	<>File handle> Number from Module 9240
PS	B/W/D/K/S	<>String number, source data> 0 to 7
CM	9244	
PL	B/W/D	<>Number of written bytes (including LF)> –1: Error code in NN_GenApiModuleErrorCode

Error recognition:

Marker	Value	Meaning
NN_GenApiModule ErrorCode	2	Incorrect file handle
	3	Incorrect string number
	7	File system error
	20	Module was not called in a spawn job or submit job

Module 9245 Read a field from a table

The module reads a data field from a table opened before by Module 9240 in "record-oriented" mode into a string. The data field is addressed by the field name and the line number.

To maintain a high processing speed, multiple lines should be read in ascending order.

Pay attention to the upper/lower case of field names.

If an error occurs, the content of the target string is undefined.

The module provides the contents as a string.

Call:

PS	D	<>File handle> from Module 9240
PS	B/W/D/K	<>Line> 0 to 65 535
PS	B/W/D/K/S	<>String number, column name> 0 to 15
PS	B/W/D/K/S	<>String number, result> 0 to 15
CM	9245	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Field was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	7	The table could not be read from
	20	Module was not called in a spawn job or submit job
	29	The opened file is not a table (extension .TAB, .P)
	30	Column name not found

Module 9246 Write to a field in a table

The module writes a string to a data field in a table opened by Module 9240 in "record-oriented" mode. The data field is addressed by the field name and the line number.

To maintain a high processing speed, multiple lines should be written in ascending order.

Pay attention to the upper/lower case of field names.

The field defined by the column name and line number is overwritten.

The module transfers a string.

Call:

PS	D	<>File handle> from Module 9240
PS	B/W/D/K	<>Line> 0 to 65 535
PS	B/W/D/K/S	<>String number, column name> 0 to 15
PS	B/W/D/K/S	<>String number, contents to be written> 0 to 15
CM	9246	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Field was written to
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	6	Table is write-protected
	7	Not a numerical field (Module 9256)
	11	The transferred value cannot be saved to the addressed field. Incorrect format
	20	Module was not called in a spawn or submit job
	29	Opened file is not of the .TAB or .P type.
	30	Column name not found

Module 9247 Search for a condition in a table

In a table opened by Module 9240 in "record-oriented" mode, the function searches for a data record which fulfills one or more conditions. The conditions are formulated with a subgroup of the System Query Language (SQL) database language.

Pay attention to the case of the letters (whether they are small or capital) in the commands and column names.
If you indicate a starting line, the module can search for several suitable field entries.

Permissible SQL commands:

Command	Meaning
+,–,*,/	Arithmetic operators
NOT, AND, OR	Logical operators
<,>, <=, >=, ==, <>	Comparisons
LIKE ´abc´	Text comparison
LIKE ´_abc%´	Partial string
()	Parentheses
MIN(column name)	Minimal value from the column
MAX(column name)	Maximum value from the column

Example:

Search a table for the line with the NC program 1.nc and the set datum X=–10.
String contents:

WHERE (PAL/PGM LIKE´PGM´) AND (NAME LIKE´1.nc´) AND (X==–10)

Call:

PS	D	<>File handle> from Module 9240
PS	B/W/D/K	<>Starting line> 0 to 65 535
PS	B/W/D/K/S	<>String number of condition or string with condition> 0 to 7
CM	9247	
PL	B/W/D	<>Line that fulfills the condition> –1: Error code in NN_GenApiModuleErrorCode

Error recognition:

Marker	Value	Meaning
NN_GenApiModule ErrorCode	1	Start line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	7	Module could not be read from the table
	20	Module was not called in a spawn job or submit job
	29	Incorrect file format
	30	Column name not found
	31	Syntax error in the transferred condition
	32	No data record found that fulfills the condition

Module 9249 Read and reset "errno"

This function reads the error status "errno" of the operating and file system, and resets this status to 0. This status can be used for more accurate determination of the errors in certain modules (e.g. 9240, 9242, 9243, 9244).

Constraints:

- "errno" always contains the code of the most recent error. The variable is only cleared with Module 9249.
- "errno" is only valid within a PLC process (SUBMIT job), and is separately present for each process.
- The definitions of the C programming language (UNIX compatible) as well as specific expansions by HEIDENHAIN are valid for the contents of "errno." A separate documentation exists for this.
- In order to use "errno" to see if an error has occurred, the module must be called before a program sequence in order to clear the contents of "errno."

Call:

```
CM    9249
PL    W/D      <>errno>
```

Module 9250 Starting the editor for sections of a table

In **Machine** mode, the module starts a table editor on the screen, which is used to edit the given rows and columns of a table. For this purpose a temporary copy of the file is created, which can be checked with Modules 9240, 9241, 9245 and 9247 before the edited data is transferred to the original file with Module 9251.

Constraints:

- The complete path or a symbolic name must be entered for the file to be edited.
- All fields to be edited must be entered in the order that they are to appear on the screen. Do not enter the field with the line number: it appears automatically. The field names must be separated from each other by blank spaces.
- An empty string may also be given for the fields to be edited. In this case all fields from the original file are assumed.
- The module creates the file SSYS:\TEMP\PLCTABED.TDB, which only exists for the time of the editing process. It can be read by the PLC, but must be closed before the editor is ended with Module 9251.
- If the PLC program is translated again while the editor is open, the editor is closed without updating the original file.
- If the "END" key or soft key is pressed while the editor is active, the PLC marker NN_GenTableEditEnd is set. The NC does not end the editor automatically. The PLC checks the entry because of this marker, and if necessary, ends the editor with Module 9251.
- Marker NN_GenTableEditEnd is deleted in advance when the module is called. It can also be reset by the PLC program.
- If –1 is entered for the last line to be edited, the file is edited to the end of the file.
- If all lines and columns are being edited, you can choose whether deletion or insertion of lines is permitted. There is also an option for editing the original file directly. However, Module 9251 cannot then be used to undo the changes.
- The line numbering begins with line 0.

Call:

PS B/W/D/K/S<>File name>
(string number 0..n or constant string)

PS B/W/D/K/S<>Fields to be edited>
(string number 0..n or constant string)

PS B/W/D/K <>First line to be edited>
0 to 65535

PS B/W/D/K <>Last line to be edited>
0 to 65535

PS B/W/D/K <>Additional parameters>
Mode:
Bit #0 =1: Lines can be inserted and deleted
(only for all lines and columns)
Bit #1 =1: Edit the original file
(only for all lines and columns)
Bit 2 =1: Show as form (otherwise table)
Bit 3 = 1: Write-protection of the file to be opened is respected

CM 9250

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Editor was opened
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	– Invalid values for the strings for file name and field name were transferred – The range defined by the entries for first and last line is not logical. Possibilities include start>end, or start after file end, or the given file is an empty document, or the start and/or end contain negative values. – Values for additional parameters were given, which is not permitted in this context, or options which are only intended for complete files were activated for a partial file (not all lines/columns)
	6	A write-protected file cannot be opened with transferred write-protection (bit 3).
	7	The given file cannot be read from, or the temporary file cannot be created.
	8	The module was called during an NC operating mode in which the table editor cannot be started.
	20	The module was called from the cyclic program part.
	28	A file has already been opened by the PLC for editing
	29	The given file does not exist or is not of the type .TAB or .P.
	30	The given field names do not exist in the given file.
	36	An invalid file name or file type was programmed.

Module 9251 Exit table editor of the PLC

The module ends the PLC's table editor, which had been started with Module 9250. Select whether the changed data are to be assumed in the original file, or whether the editor is closed without updating.

Constraints:

- The editor must have been started with Module 9250, and the file SYS:\TEMP\PLCTABED.TDB must have been created. The contents of this file are inserted into the original file given in the call for Module 9250.
- Only the lines and columns contained in the temporary file are inserted. All other lines and columns in the original file remain unchanged.
- The entered values are not checked. The PLC can automatically read and check the temporary file beforehand.
- The temporary file must not be open by the PLC at the time the module is called, because it cannot be deleted then. Calling Module 9250 again could then fail.

Call:
PS B/W/D/K <>Store changes>
(0 = no, 1 = yes)
CM 9251

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Editor was exited
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	3	An incorrect value was entered for the "Store changes" parameter.
	6	The changes could not be stored in the original file. See "errno" for the cause.
	20	The module was called from the cyclic program part.
	28	No editor had been opened with Module 9250.

Module 9252 Position the cursor in the PLC table editor.
The module moves the input field in the PLC table editor to a certain line and column.

Constraints:

- The PLC table editor must have been opened with Module 9250.
- The specified field must be contained in the group of fields to be edited from Module 9250.
- The line for positioning must be defined relative to the created temporary file, starting from line 0, meaning that it might be necessary to deduct a start line defined with Module 9250 from the line in the original file.

Call:
PS B/W/D/K/S<>Field name>
(string number 0..3 or constant string)
PS B/W/D/K <>Line>
(relative to the temporary file
CM 9252

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Cursor was set
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	The given line is not contained in the file being edited.
	3	An invalid value was transferred for the string of the field name.
	20	The module was called from the cyclic program part.
	28	No editor had been opened with Module 9250.
	30	The given field name does not exist in the given file.

Module 9255 Read a field from a table

The module reads a data field from a table opened by Module 9240 in "record-oriented" mode as an integer value. The data field is addressed by the field name and the line number.
To maintain a high processing speed, multiple lines should be read in ascending order.

Pay attention to the upper/lower case of field names.

If an error occurs, the number value of the result is undefined.

The function can only be used on fields containing numerical values.

If digits can be entered after the decimal point in the selected field, the numerical value is standardized to the last digit after the decimal point, meaning that for n digits after the decimal point, the value is multiplied by 10^n.

Call:

PS	D	<>File handle> from Module 9240
PS	B/W/D/K	<>Line> 0 to 65 535
PS	B/W/D/K/S	<>String number, column name> 0 to 15
CM	9255	
PL	B/W/D	<>Result>

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Field was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	7	The table could not be read from
	20	Module was not called in a spawn job or submit job
	29	The opened file is not a table (extension .TAB, .P)
	30	Column name not found

Module 9256 Write to a field in a table

The module writes an integer value to a data field in a table opened by Module 9240 in "record-oriented" mode. The data field is addressed by the field name and the line number.

The field defined by the column name and line number is overwritten.

Constraints:

- The file must not have been opened by Module 9240 in "buffered" mode.
- Only files of the type .TAB (freely definable tables) and .P (pallet tables) are permitted.
- To maintain a high processing speed, multiple lines should be written in ascending order.
- Pay attention to the upper/lower case of field names.
- Only fields defined for numerical values can be written to.
- If digits can be entered after the decimal point in the selected field, the numerical value is standardized to the last digit after the decimal point, meaning that for n digits after the decimal point, the value is divided by 10^n.

Call:

PS	D	<>File handle> from Module 9240
PS	B/W/D/K	<>Line> 0 to 65 535
PS	B/W/D/K/S	<>String number, column name> 0 to 15
PS	B/W/D/K	<>Numerical value to be written>
CM	9256	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Field was written to
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Line does not exist in table
	2	Incorrect "file handle" or table was opened in "buffered" mode
	3	Impermissible string numbers
	6	Table is write-protected
	7	Not a numerical field (Module 9256)
	11	The transferred value cannot be saved to the addressed field. Incorrect format
	20	Module was not called in a spawn or submit job
	29	Opened file is not of the .TAB or .P type.
	30	Column name not found

8.10.7 Access to tables via SQL commands

The NC software accesses the tables via a server. Accesses from the NC program, the PLC program and the editor are synchronized with each other, and locked against each other. This server is controlled with SQL commands.

The server is based on a transaction model which performs read or write operations for more than one row or column in such a way that they cannot be interrupted by other read or write operations.

A transaction consists of the following steps

- Selecting data
- Reading data
- Editing data (if required)
- Confirming or rejecting the changes

Selecting the data

To select data, use the SQL statement **SELECT.** The SELECT statement finds the columns of the rows of a table, which satisfy the entered **WHERE** condition.

The table is usually designated by a synonym. These synonyms are configuration data which can also be created, assigned and deleted via the SQL server. In addition, you can specify a table by using its path name, provided that you enclose the path name in single quotation marks.

The result of the query is a result set which is created and managed by the SQL server. The SQL server assigns a **handle** to the result set, which enables you to identify the result set for reading/editing data and committing the transaction. The handle is the result of the query, which is visible in the NC program. The value 0 indicates an invalid handle, i.e. it was not possible to create a result set for that query. If no rows that satisfy the specified condition are found, an empty result set is created and assigned a valid handle.

The ORDER BY function sorts the rows in the result set according to the values of a column in increasing or decreasing order. The row numbers in the result set are listed in increasing order and are not related to the row numbers in the table file.

You can create a dynamic query, for example, by replacing the comparison value in the **WHERE** condition with a reference to a Q parameter. To do this, program a colon (:) instead of the value and enclose the Q parameter in single quotation marks (e.g. 'Q2'). During interpretation this expression is replaced by the current content of the Q parameter.

The **FOR UPDATE** function locks the selected rows for the duration of the transaction. As a result, third parties can only read the data, but cannot edit them. However, this query cannot be executed if it is to access data that have already been locked.

Reading data

In order to read the selected data, use **SQL BIND** to bind the data of a column to a parameter that accesses the data from the NC program.

The data can be bound to a Q parameter (e.g. **SQL BIND Q5 'TOOL.L'**) or directly to a system datum of the interpreter (e.g. **SQL SYSBIND ID350 NR52 IDX 1 'PROBE.L'**). When a row is read, the content of the respective columns is stored in the bound parameter. To cancel the binding, it must be programmed again without specifying a column. The bindings are globally effective and must be canceled explicitly. The bindings must be unambiguous. The attempt to bind more than one parameter to a column will fail.

Use SQL FETCH to read the data of the result row by row. The result set is identified by the indicated **handle.** It is therefore possible to read various result sets alternately. An index in the range from 0 to n can be defined for the result-set row to be read. If no index is specified, the first row is read. The result of the function is 0 if data was read and stored in the bound parameters. The result of the function does not equal 0 if no data was read.

Editing data

If you want to edit data, you first have to edit the data in the bound parameters. The **SQL UPDATE** command, copies the data into the result set which is identified by the **handle.** An index in the range from 0 to n can be defined for the result-set row to be written. If no index is specified, the first row is edited. The result of the function is 0 if the data have correctly been transferred to the result set. The result of the function is not 0 if an error occurred during transfer.

Afterwards the changes are only visible within the edited result set. If you reread the edited row, the edited data are shown; if you enter a new query and reread the row, the original data will be shown. Third parties will still see the original data. The data is not transferred to the table until the transaction has been concluded. Thus it is possible to edit more than one row in a consistent manner in one transaction.

Concluding the transaction

Be sure to conclude every transaction. This way the resources assigned in the SQL server are released for the result set.
After concluding the transaction with **SQL COMMIT,** all changes are transferred from the result set into the table.
To cancel all changes, conclude the transaction with **SQL ROLLBACK**.
Once transferred, these changes cannot be undone. After the transaction has been completed, the rows locked during the selection are unlocked.
The transaction to be completed is identified by the **handle.** After the transaction has been completed successfully, the **handle** becomes invalid and cannot be used for accessing data any longer.
The result of the **SQL COMMIT** function will not be 0 if the edited data could not be transferred into the table file. This happens when edited rows were not locked during the selection and are locked by third parties at the time of commitment. The SQL server first checks whether all edited rows may be rewritten before it transfers the first change. The isolation of accesses ensures that the effects of the transaction are invisible to others until the transaction is committed. It may, however, happen that errors occur while a file is being accessed.

The **SQL ROLLBACK** function enables you to remove all rows, except for the indicated row, from the result set by defining an index. As a result, the changes made to the removed rows are canceled. This is especially useful if you first select and, as a consequence, lock more than one row, but then decide that you want to edit only one row. The other rows can then immediately be released. The handle remains effective until the transaction has been completed for all rows. If no index or an invalid index has been defined, the entire transaction is completed. The result of the function does not equal 0 if an error has occurred.

BNF notation

The following specifies the individual syntax elements in BNF notation. The individual NC blocks are listed in alphabetical order.

A specified syntax element is identified by a name in italics.

Individual characters are enclosed in single quotation marks; entire code words are plain text and are not enclosed in single quotation marks.

A single simple expression is enclosed in square brackets [ ], whereas a multiple expression is enclosed in braces { }. Two expressions separated by the "|" symbol indicate alternative options.

Expressions may contain letters, numbers, and symbols.

- uppercase := 'A'...'Z'
- lowercase := 'a'...'z'
- digit := '0'...'9'
- index := digit { digit }
- number := ['+' | '-'] digit { digit } ['.' digit { digit }
- literal := '"' { uppercase | lowercase | digit | symbol } '"'
- name := uppercase { uppercase | lowercase | digit | '$' | '#' | '_' }

Context elements

The following elements can only occur within the context of an NC block.

Q parameters

A Q parameter can be addressed directly or indirectly via another Q parameter.

- q-number := Q index
- q-reference := Q REF Q index
- q-parameter := q-number | q-reference

SQL parameters

Q parameters and system parameters can be bound to the column name of a table.

Definition

- system-group := ID index
- system-number := NR index
- system-index := IDX index
- system-parameter := system-group system-number system-index
- sql-parameter := q-parameter | system-parameter

SQL-HANDLE

The SQL handle identifies the result set of a previous SQL query. Only values assigned by the SQL server are valid handles.
The value 0 identifies an invalid handle.

Definition

- sql-handle := HANDLE q-number

SQL index

The SQL index identifies the row from the result set. The indices start at 0 and are listed in increasing order.
If no index is specified, the first row from the result set is automatically transferred.

Definition

- sql-index := INDEX (index | q-number)

SQL column

The name of the table and of the column to be bound is to be enclosed in double quotation marks.

Definition

- table-name := name
- column-name := name
- sql-column := '"' table-name '.' column-name '"'

SQL statement

Enclose the statement to be executed in double quotation marks. A reference to a Q parameter can be used within
a statement. Enclose the Q parameter in single
quotes after a colon. The interpreter replaces this sequence with the value of the Q parameter.

Definition

- sql-replacement:= ':' ''' q-parameter '''
- sql-statement:= literal

Example:

```
SQL Q5 " SELECT L, R FROM TOOL WHERE N = :'Q2' "
```

SQL

The NC block SQL defines an SQL statement to be executed. The SQL HANDLE, which will enable you to access the data at a later date, is stored in the specified parameter. It is valid until the transaction has been committed or canceled for all rows of the result set.

Definition

- sql-execute := SQL q-parameter sql-statement

Example:

```
SQL Q5 " SELECT L, R FROM TOOL WHERE N = :'Q2' "
```

SQL BIND

The NC block SQL BIND binds a Q parameter to a column of a table. If you redefine the NC block without specifying a column, the binding will be canceled.
Otherwise, the binding remains in effect until the current subprogram or cycle is completed.

Definition

- sql-bind:= SQL BIND q-parameter [sql-column]

Example:
SQL BIND Q63 "TCHPROBE.OFFS0"

SQL SYSBIND

The NC block SQL SYSBIND binds a system parameter to a column of a table. If you redefine the NC block without specifying a column, the binding will be canceled. As with SQL BIND, if not otherwise specified, the binding remains in effect until the subprogram or cycle is completed.

Definition

- sql-bind := SQL SYSBIND system-parameter [sql-column]

Example:

```
SQL SYSBIND Q63 "ID50 NR1 IDX1"
```

SQL FETCH

The NC block SQL FETCH reads a row from the result set of an SQL query and assigns the data to the bound parameters. If the values in the table are expressed in inches, lengths and feed rates are converted into millimeters during the reading process. The values in the bound parameters are always assumed to be metric. As with FN18, this also applies if the current program is entered in inches. If no index has been specified, the first row of the result set is transferred. The specified Q parameter is assigned a return code. If the command has been completed successfully, the Q parameter is assigned a zero. If not, it is assigned a one.

Definition

- sql-fetch:= SQL FETCH q-parameter sql-handle [sql-index]

Example:

```
SQL FETCH Q80 HANDLE Q5 INDEX0
```

SQL UPDATE

The NC block SQL UPDATE assigns the data from the bound parameters to the corresponding rows or columns of the table. If the values in the table are expressed in inches, lengths and feed rates are converted into millimeters before the assignment process. The values in the bound parameters are always assumed to be metric. As with FN17, this also applies if the current program is entered in inches.
The specified Q parameter is assigned a return code. If the command has been completed successfully, the Q parameter is assigned a zero. If not, it is assigned a one.

Definition

- sql-update := SQL UPDATE q-parameter sql-handle [sql-index]

Example:

```
SQL UPDATE Q80 HANDLE Q5 INDEX0
```

SQL COMMIT

The NC block SQL COMMIT cancels locks on table rows or table columns. Edited table data are permanently transferred through SQL COMMIT.
The specified Q parameter is assigned a return code. If the command has been completed successfully, the Q parameter is assigned a zero. If not, it is assigned a one.

Definition

- sql-commit := SQL COMMIT q-parameter sql-handle

Example:

```
SQL COMMIT Q80 HANDLE Q5
```

SQL ROLLBACK

The NC block SQL ROLLBACK undoes a transaction. In particular, the lock on rows in an SQL statement "SELECT ... FOR UPDATE" is canceled.
The specified Q parameter is assigned a return code. If the command has been completed successfully, the Q parameter is assigned a zero. If not, it is assigned a one.
If required, you can specify in the index the row for which the transaction is to take effect.

Definition

- sql-rollback := SQL ROLLBACK q-parameter sql-handle [sql-index]

Example:

```
SQL ROLLBACK Q80 HANDLE Q5
```

Command options for SELECT and UPDATE

Command options allow you to define conditions, sorting sequences and locks that modify the effect of a command.

WHERE

The WHERE option limits the effect of a command to the rows of a table which satisfy the specified condition.

Definition

- where-option:= WHERE condition

ORDER BY

The ORDER BY option influences the sequence of rows in the result set. At present, it is only possible to sort by column (default ASC).

Definition

- order-option:= ORDER BY column [ASC | DESC]

FOR UPDATE

The FOR UPDATE option already locks the rows during selection (pessimistic locking).
Without the FOR UPDATE option, the selected rows are not locked until the COMMIT command is executed (optimistic locking).

Definition

- update-option:= FOR UPDATE

FOR NOTIFICATION

The FOR NOTIFICATION option monitors the table for changes. FOR NOTIFICATION provides the client with a result containing information on the change.

Definition

- lock-option := FOR UPDATE | FOR NOTIFICATION

SQL commands

SELECT

In a SELECT statement a list of the columns to be selected and the table preceded by the keyword FROM must be specified. In addition, it may contain a condition with the keyword WHERE, a sorting sequence with the keyword ORDER BY and a command for pessimistic locking with the keyword FOR UPDATE.

Definition

- select-list := '*' | column-list
- select-option:= [where-option] [order-option] [lock-option]
- select-statement := SELECT select-list FROM table select-option

Examples:

```
SELECT * FROM TOOL WHERE RT == 5 AND LOCK <> 1 ORDER BY TIME
```

```
SELECT TIME FROM TOOL WHERE NR==7 FOR UPDATE
```

```
SELECT L,R,R2 FROM 'OEM:\TOOL.T' WHERE NAME LIKE 'T1999'
```

UPDATE

In an UPDATE statement the table and the columns to be edited preceded by the keyword SET must be specified. Furthermore, it may contain a condition with the keyword WHERE. If the WHERE condition is not specified, all rows are edited.

Definition

- assignment := column '=' expression
- update-list := assignment { ',' assignment }
- update-option:= [where-option]
- update-statement := UPDATE table SET update-list update-option

Examples:

```
UPDATE TOOL SET LOCK = 1 WHERE RT == 5 AND LOCK <> 1
```

```
UPDATE TOOL SET TIME = 0, LOCK = 0
```

```
UPDATE 'OEM:\TOOL.T' SET TIME = MAXTIME WHERE NAME LIKE 'T1999'
```

INSERT

In an INSERT statement the table and the values to be set which are to be enclosed within parentheses, separated by commas and preceded by the keyword VALUES must be specified. Be sure to assign all columns. The INSERT command appends a new row to the table. It is not possible to insert a new row between two rows.

Definition

- insert-list := '(' expression-list ')'
- insert-statement:= INSERT INTO table VALUES insert-list

Examples:

INSERT INTO TOOL VALUES (9,1,'T2000',0,1000)

INSERT INTO 'OEM:\TOOL.T' VALUES (9,1,'T2000',0,1000)

RENAME TABLE

The name of a table file is changed. If a logical table name is specified, the file identified by the name will be edited. Make sure that the name of the target file does not already exist. The name stored internally will be edited accordingly.

With this command, the previous table is copied into a new table. Then the previous table is deleted. This command allows you to move a table to another directory.

Definition

- rename-table-statement:= RENAME TABLE table TO table

Examples:

RENAME TABLE TOOL TO 'OEM:\TOOL.T'

RENAME TABLE 'OEM:\TOOL.T' TO 'OEM:\WERKZEUG.T'

CREATE TABLE

A CREATE TABLE statement creates a new table (new table file). Make sure that the specified table name does not already exist. The names of the columns to be inserted are given as a list. The properties of the columns are read from the configuration. A column configuration for each column name must be available for this. The column width is determined from the width configured for a column or from the length of the column name, depending on which width is larger.

If you enter an asterisk * instead of the list of column names, all the columns defined in the configuration for this type of table (table extension) will be used.

The ASINCH option is used to specify whether a table is to contain values in inches. As a result, inch-sensitive columns (defined in the column configuration) are created as inch columns in the table. If this option is not specified, the respective columns are created as millimeter columns.

With the LOCALCONFIG option, the properties of the table are stored locally in the table.

Definition

- create-list := '*' | column-list
- create-table-statement:= CREATE TABLE table '(' create-list ')' [ASINCH] [LOCALCONFIG]

Examples:

```
CREATE TABLE TOOL(L,R,R2,DL,DR,DR2,TL,RT,TIME1,TIME2,
                  CUR_TIME,PLC) ASINCH

CREATE TABLE 'OEM:\TOOL.T' (L,R,R2,DL,DR,DR2,TL,RT,TIME1,TIME2,
                            CUR_TIME,PLC)

CREATE TABLE *
```

ALTER TABLE

ALTER TABLE modifies the properties of a table. This option enables you to add or delete columns, and to modify the properties of columns. When you add columns or modify the properties of columns, the new properties are read from the configuration. When column properties are modified, the values are not modified. If the width of the new column is larger than the width of the previous one, the column is extended to the new width. If the width of the new column is smaller, however, the column is not changed, so as to avoid any loss of data.

A new column is inserted in the table at the position which is determined from the columns defined for the respective type of table (table extension) in the configuration. The sequence of columns in the table is derived from the configuration data.

Definition

- alter-table-options:= ADD | MODIFY | DROP
- alter-table-statement:= ALTER TABLE table alter-table-options '('column-list ')'

Examples:

```
ALTER TABLE TOOL ADD (DOC)

ALTER TABLE 'OEM:\TOOL.T' MODIFY (L)
```

COPY TABLE

COPY TABLE copies the table into a new table. Make sure that the name of the target file does not already exist. The name stored internally will be modified accordingly. It is possible to specify logical table names.

Definition

- copy-table-statement:= COPY TABLE table TO table

Examples:

```
COPY TABLE TOOL TO 'OEM:\TOOL.T'
COPY TABLE 'OEM:\TOOL.T' TO 'OEM:\WERKZEUG.T'
```

DROP TABLE

DROP TABLE deletes an existing table file. If a logical table name is specified, the file identified by the name will be deleted.

Definition

- drop-table-statement:= DROP TABLE table

Examples:

```
DROP TABLE TOOL
DROP TABLE 'OEM:\TOOL.T'
```

RENAME COLUMN

RENAME COLUMN changes the name of an existing column. The properties of the column are not changed. If the configuration of the table is stored locally in the table, the name of the column is also changed in the respective configuration data.

Definition

- rename-column-statement:= RENAME COLUMN table '(' column-list ')' TO '(' column-list ')'

Example:

```
RENAME COLUMN TOOL ( DR2 ) TO ( DIR )
```

CREATE SYNONYM

CREATE SYNONYM creates a new logical table name. It is not necessary that the table file identified by the logical name already exists.

Definition

- create-synonym-statement:= CREATE SYNONYM table-name FOR table-literal

Example:

```
CREATE SYNONYM OUTIL FOR 'OEM:\TOOL.T'
```

ALTER SYNONYM ALTER SYNONYM assigns another table file to the logical name.

Definition

- alter-synonym-statement:= ALTER SYNONYM table-name TO table-literal

Examples:

```
ALTER SYNONYM TOOL TO 'OEM:\WERKZEUG.T'
```

DROP SYNONYM DROP SYNONYM removes a logical name. The table file identified by the logical name will not be removed.

Definition

- drop-synonym-statement:= DROP SYNONYM table-name

Examples:

```
DROP SYNONYM OUTIL
```

Application example for SQL commands

Read data from table

To determine the positions in a measuring cycle, calibration data of a touch probe are to be transferred from the tchprobe.tp table.
The center offsets are indicated in the CAL_OF1 and CAL_OF2 columns.

- BIND links the Q parameters with column names:
  ```
  SQL BIND Q63 "TCHPROBE.CAL_OF1"
  SQL BIND Q64 "TCHPROBE.CAL_OF2"
  ```
- SELECT chooses those columns from the table (TNC:\table\tchprobe.tp) that are to be assigned to the active touch probe (WHERE ACTNR==1):
  ```
  SQL Q5 "SELECT CAL_OF1,CAL_OF2 FROM 'TNC:\table\tchprobe.tp' WHERE
  ACTNR==1"
  ```
- FETCH reads a row from the result set (HANDLE Q5) and assigns the data to the bound parameters. With INDEX0, the first row of the selected data is read:
  ```
  SQL FETCH Q80 HANDLE Q5 INDEX0
  ```
- Safety check (Q80 is equal to zero?)
- The data is assigned to Q parameters:
  ```
  Q43 = Q63
  Q44 = Q64
  ```
- ROLLBACK releases the selected data:
  ```
  SQL ROLLBACK Q80 HANDLE Q5
  ```
- Safety check (Q80 is equal to zero?)
- Following that, the bindings are undone:
  ```
  SQL BIND Q63
  SQL BIND Q64
  ```

Writing data to a table

The triggering touch probe was newly calibrated. The Q parameters Q891, Q798 and Q799 contain the values determined for radius, center offset in the principal axis and center offset in the secondary axis.
The following SQL commands update the table entries with the calibration data.

- BIND links the Q parameters with column names:
  ```
  SQL BIND Q891 "TOOL.R0"
  SQL BIND Q798 "TCHPROBE.CAL_OF1"
  SQL BIND Q799 "TCHPROBE.CAL_OF2"
  ```
- SELECT chooses those columns from the table (TNC:\table\tchprobe.tp) that are to be assigned to the active touch probe (WHERE ACTNR==1):
  ```
  SQL Q5 "SELECT R0,CAL_OF1,CAL_OF2 FROM 'TNC:\table\tchprobe.tp'
  WHERE ACTNR==1"
  ```
- FETCH reads a row from the result set (HANDLE Q5) and assigns the data to the bound parameters. With INDEX0, the first row of the selected data is read:
  ```
  SQL FETCH Q80 HANDLE Q5 INDEX0
  ```
- Safety check (Q80 is equal to zero?)
- UPDATE writes the values from the bound Q parameters into the table:
  ```
  SQL UPDATE Q80 HANDLE Q5 INDEX0
  ```
- Safety check (Q80 is equal to zero?)
- COMMIT makes the changes to the table permanent:
  ```
  SQL COMMIT Q80 HANDLE Q5
  ```
- Safety check (Q80 is equal to zero?)
- Following that, the bindings are undone:
  ```
  SQL BIND Q891
  SQL BIND Q798
  SQL BIND Q799
  ```

Note

The SQL commands FETCH, UPDATE, COMMIT and ROLLBACK assign a return code to the Q parameters.
If the command has been completed successfully, the Q parameter is assigned a zero. If not, it is assigned a one.
You should always make a safety check after these commands.

Module 9440 Open a transaction

Module 9440 executes the SELECT statement that is given to the module. For a description of the supported queries, See "SQL commands" on page 1225. If the statement was executed successfully, a transaction is opened and its handle is returned. This handle can be used to read data from a machine table (tool table, for example), or to change data in the table. The cursor is placed on the first record of the result set. In order to save changes to the tables, the transaction must be concluded and closed with Module 9441 after changing the records.

If the statement contains a string **:'Bnnn'** or **:'Wnnn'** or **:'Dnnn'** (with nnn reading as number from 0 to the maximal number of BYTEs, WORDs or DWORDs respectively), this string is replaced by the integer value found in the associated PLC data.

Constraints:

- The module can only be executed within a submit job.
- No more than 10 transactions may be open at the same time.

Possible errors:

- The module was not called in a submit job
- The statement is syntactically not correct
- The table given does not exist, is not accessible or is fully or partially locked
- The columns given do not exist within the table
- No records were selected

Call:

PL	B/W/D/K/S<>Valid SQL statement>	
CM	9440	
PL	D	<>Transaction handle>
PL	B/W/D	<>Error number> For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Transaction was successfully opened
	1	Error. See <Error number>

Module 9441 Conclude and close a transaction

Module 9441 concludes a transaction. The module tries to write all buffered changes to the selected machine table. If the action is successful, the transaction is closed and the transaction handle is invalidated. Otherwise, the transaction remains open. In order to conclude the transaction despite this, correct the modifications so that no constraints are violated. If the changes are not successful, the buffered changes cannot be saved (conclusion with Module 9451).

Constraints:

- The module can only be executed within a submit job.

Possible errors:

- The module was not called in a submit job
- No transaction was opened for the given handle
- At least one modification made violates a uniqueness constraint defined for a column
- At least one modification made violates a foreign key constraint defined for a column

Call:

PS	B/W/D/K	<>Transaction handle>
CM	9441	
PL	B/W/D	<>Error number> For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Transaction was successfully closed
	1	Error. See <Error number>

Module 9442 Seek a record in the result set

Module 9442 positions the cursor on the record, defined by the record number, in the result set. If the given record number does not identify the desired record unambiguously, the cursor is then placed on the first or last record (depending on the value of the record number given). The first record is addressed by the record number 0.

Constraints:

- The module can only be executed within a submit job.
- There must already be a result set before Module 9442 can be used.

Possible errors:

- The module was not called in a submit job
- No transaction was opened for the given handle
- The record number exceeds the number of selected records.
- The statement did not lock the selected records and the record was deleted by another statement

Call:

PS	B/W/D/K	<>Transaction handle>
PS	B/W/D/K	<>Record number>
CM	9442	
PL	B/W/D	<>Error number>

For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Record was found
	1	Error. See <Error number>

Module 9443 Fetch a record from the result set

Module 9443 reads a record (line) from a table and saves it in a string. There must already be a transaction open whose transaction handle is given to the module. The values are returned as a comma separated list.

Empty fields are output as two successive commas (**,,**).

A decimal point is always used for data types **REAL**, **LENGTH**, and **FEED**. Values of the data types **SIGN**, **BOOL** and **TEXT** are enclosed in single quotes (**'**).

Constraints:

- The module can only be executed within a submit job.
- There must already be a result set before Module 9443 can be used.

Possible errors:

- The module was not called in a submit job
- An invalid string address is given to the module
- No transaction was opened for the given handle
- No record was picked
- The length of the string exceeds the maximal string length

Call:

PS	B/W/D/K	<>Transaction handle>
PS	B/W/D/K	<>String address in which the TNC saves the record>
CM	9443	
PL	B/W/D	<>Error number>

For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Record was read and copied
	1	Error. See <Error number>

Module 9444 Change a record in the result set

Module 9444 reads a string and saves it in the current record. The current record is the one in which the cursor is located at present. If the cursor is at the end of the result set, a new entry is added. There must already be a transaction open whose transaction handle is given to the module. The values must be given in the string as a comma separated list and in the appropriate table format.

Empty fields are output as two successive commas (**...,,...**) or via the keyword NULL (**...,NULL,...**).

A decimal point is always used for data types **REAL**, **LENGTH**, and **FEED**. Values of the data types **SIGN**, **BOOL** and **TEXT** are enclosed in single quotes (**'**).

The modifications are buffered until the transaction is committed.

Constraints:

- The module can only be executed within a submit job.
- There must already be a result set before Module 9444 can be used.

Possible errors:

- The module was not called in a submit job
- An invalid string address is given to the module
- At least one transferred value is outside the valid range
- At least one transferred value is syntactically incorrect

Call:

PS	B/W/D/K	<>Transaction handle>
PS	B/W/D/K	<>String address>
CM	9444	
PL	B/W/D	<>Error number>

For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Record was updated and inserted
	1	Error. See <Error number>

Module 9445 Read a single value from a table

Module 9445 reads a single value from a table cell and copies this value to a string. The cell content is selected via a SQL statement. This SQL statement is given to the module beforehand.

If the statement contains a string **:'Bnnn'** or **:'Wnnn'** or **:'Dnnn'** (with nnn reading as number from 0 to the maximal number of BYTEs, WORDs or DWORDs respectively), this string is replaced by the integer value found in the associated PLC data.

A decimal point is always used for data types **REAL**, **LENGTH**, and **FEED**. Values of the data types **SIGN**, **BOOL** and **TEXT** are enclosed in single quotes (**'**).

Constraints:

- The module can only be executed within a submit job.

Possible errors:

- The module was not called in a submit job
- An invalid string address is given to the module
- The statement is syntactically not correct
- The table given does not exist or is not accessible
- The columns given do not exist within the table
- More than one column was named in the statement
- No record or more than one record were selected by the statement

Call:

PS	B/W/D/K	<>Valid SQL statement>
PS	B/W/D/K	<>String address for the read value>
CM	9445	
PL	B/W/D	<>Error number> For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Value was read
	1	Error. See <Error number>

Module 9447 Delete record from result set

Module 9447 deletes the current record (table line). The current record is the one in which the cursor is located at present. There must already be a transaction open whose transaction handle is given to the module. The modification is buffered until the transaction is committed.

Constraints:

- The module can only be executed within a submit job.
- There must already be a result set before Module 9447 can be used.

Possible errors:

- The module was not called in a submit job
- No transaction was opened for the given handle

Call:

PS	B/W/D/K	<>Transaction handle>
CM	9447	
PL	B/W/D	<>Error number> For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Description was successfully loaded
	1	Error. See <Error number>

Module 9448 Load a column description

Module 9448 loads the description of one or more columns into a cache. The column is specified by its qualified name in the form table.column. If a wildcard "*" is given as the column (i.e. a qualified name in the form table.*****), the description of all columns of the given table is loaded. The description is used by various modules for the conversion to or from binary data. Required descriptions that were not found in the cache are automatically loaded during the execution of these modules. However, the execution time of these modules can be improved, if descriptions are cached.

Constraints:

- The module can only be executed within a submit job.

Possible errors:

- The referenced table does not exist.
- The referenced column does not exist.

Call:

PS	B/W/D/K	<>String address with the qualified column name>
CM	9448	
PL	B/W/D	<>Error number>
		For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Description was successfully loaded
	1	Error. See <Error number>

Module 9449 Extract a value from a comma separated list

Module 9449 extracts a value from a comma-separated list of values.

Constraints:

- The module can only be executed within a submit job.

Possible errors:

- The module was not called in a submit job
- The index for the value exceeds the number of values in the string

Call:

PS	B/W/D/K	<>String address for the list of values>
PS	B/W/D/K	<>Index of the value to extract>
PS	B/W/D/K	<>String address for the extracted value>
CM	9449	
PL	B/W/D	<>Error number>

For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Value was successfully extracted
	1	Error. See <Error number>

Module 9450 Execute an SQL statement

Module 9450 executes the SQL statement that is given to the module. For a description of the supported SQL statements, See "SQL commands" on page 1225. This module may not be used to open a transaction (such as via a SELECT statement).

If the statement contains a string **:'Bnnn'** or **:'Wnnn'** or **:'Dnnn'** (with nnn reading as number from 0 to the maximal number of BYTEs, WORDs or DWORDs respectively), this string is replaced by the integer value found in the associated PLC data.

Constraints:

- The module can only be executed within a submit job.

Possible errors:

- The module was not called in a submit job
- The statement is syntactically not correct
- The table given does not exist, is not accessible or is fully or partially locked
- The columns given do not exist within the table

Call:

PS B/W/D/K/S<>Valid SQL statement>
CM 9450
PL B/W/D <>Error number>
For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Statement was successfully executed
	1	Error. See <Error number>

Module 9451 Roll back and close a transaction

Module 9451 does not save all buffered modifications of a table during a transaction to the table. The transaction is closed and the transaction handle is invalidated.

Constraints:

- The module can only be executed within a submit job.

Possible errors:

- The module was not called in a submit job
- No transaction was opened for the given handle

Call:

PS	B/W/D/K	<>Transaction handle>
CM	9451	
PL	B/W/D	<>Error number>

For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Transaction was successfully closed
	1	Error. See <Error number>

Module 9452 Seek next record in the result set of a query

Module 9452 positions the cursor to the next entry in the result set.

Constraints:

- The module can only be executed within a submit job.
- There must already be a result set before Module 9452 can be used.

Possible errors:

- The module was not called in a submit job
- No transaction was opened for the given handle
- The last record in the result set has been reached
- The statement did not lock the selected records and the record was deleted by another statement

Call:

PS	B/W/D/K	<>Transaction handle>
CM	9452	
PL	B/W/D	<>Error number> For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Record was found
	1	Error. See <Error number>

Module 9453 Fetch binary data from the result set of a query

Module 9453 reads a record from a table and converts the data in the selected columns to binary values. There must already be a transaction open whose transaction handle is given to the module. The column values are copied to a number of successive DWORDs. The index of the first DWORD and the number of DWORDS are given to the module.

Data in a column of the type

- SIGN are converted to 0 or –1.
- BOOL are converted to 0 (FALSE) or +1 (TRUE).
- INDEX are each converted to a DWORD.
- REAL are converted to a DWORD by shifting the decimal separator to the right according to the maximum number of decimal places.
  For example, if the value 10.5 is in the table, Module 9453 supplies the value 105000.
- LENGTH and FEED are converted to a DWORD.
- Measurements in INCH are converted to metric units.

Constraints:

- The module can only be executed within a submit job.
- There must already be a result set before Module 9453 can be used.

Possible errors:

- The module was not called in a submit job
- No transaction was opened for the given handle
- No record was fetched
- The given number of DWORDs does not match the number of values
- The given range of DWORDs does not fit into the available memory

Call:

PS	B/W/D/K	<>Transaction handle>
PS	B/W/D/K	<>Index of the first DWORD>
PS	B/W/D/K	<>Number of the DWORDs to store>
CM	9453	
PL	B/W/D	<>Error number>

For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Data were read and converted
	1	Error. See <Error number>

Module 9454 Update binary data in the result set of a query

Module 9454 reads binary data from a number of successive DWORDs. These data are used to update the current record in the result set. If the cursor is at the end of the result set, a new record is added. There must already be a transaction open whose transaction handle is given to the module. The values are read from a number of successive DWORDs. The index of the first DWORD and the number of DWORDs are given to the module.

Data in a column of the type

- SIGN are converted to 0 or –1.
- BOOL are converted to 0 (FALSE) or +1 (TRUE).
- INDEX are each converted to a DWORD.
- REAL are converted to a DWORD by shifting the decimal separator to the right according to the maximum number of places.
  For example, if the value 10.5 is in the table, Module 9453 supplies the value 105000.
- LENGTH and FEED are converted to a DWORD.
- Measurements in INCH are converted to metric units.

The modifications to the table are buffered until the transaction is committed.

Constraints:

- The module can only be executed within a submit job.
- There must already be a query result before Module 9454 can be used.

Possible errors:

- The module was not called in a submit job
- No transaction was opened for the given handle
- The given number of DWORDS does not match the number of values
- The given range does not fit into the available memory
- At least one transferred value is outside the valid range

Call:

PS	B/W/D/K	<>Transaction handle>
PS	B/W/D/K	<>Index of the first DWORD in which the control reads the values>
PS	B/W/D/K	<>Number of DWORDs with values>
CM	9453	
PL	B/W/D	<>Error number> For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Data were updated or inserted
	1	Error. See <Error number>

Module 9455 Read a single numeric value from a table

Module 9455 reads a single value from a table and converts it to a binary value. The value is chosen via a given SELECT statement.

If the statement contains a string :'Bnnn' or :'Wnnn' or :'Dnnn' (with nnn reading as number from 0 to the maximal number of BYTEs, WORDs or DWORDs respectively) , this string is replaced by the integer value found in the corresponding PLC data.

Data in a column of the type

- SIGN are converted to 0 or –1.
- BOOL are converted to 0 (FALSE) or +1 (TRUE).
- INDEX are each converted to a DWORD.
- REAL are converted to a DWORD by shifting the decimal separator to the right according to the maximum number of places.
  For example, if the value 10.5 is in the table, Module 9453 supplies the value 105000.
- LENGTH and FEED are converted to a DWORD.
- Measurements in INCH are converted to metric units.

Constraints:

- The module can only be executed within a submit job.

Possible errors:

- The module was not called in a submit job
- An invalid string address is given to the module
- The given table does not exist or is not accessible
- The column given does not exist in the table
- The column does not hold a numerical value
- More than one column was named in the statement
- No record or more than one record were selected by the statement

Call:

PS	B/W/D/K	<>Valid SQL statement>
CM	9455	
PS	B/W/D	<>Element value>
PL	B/W/D	<>Error number>

For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Value was read
	1	Error. See <Error number>

Module 9458 Unload a column description

Module 9458 removes the description of one or more columns from the cache. The column is specified by its qualified name in the form table.column. If a wildcard '*' is given as the column (i.e. a qualified name in the form table.*), the description of all columns of the given table is unloaded. In order to save memory, descriptions should be unloaded if they are no longer used. They must be unloaded, if a different table file is used with the same table name.

Constraints:

- The module can only be executed within a submit job.

Possible errors:

- The column given to the module was not found in the cache

Call:

PS	B/W/D/K	<>String address with the qualified column name>
CM	9458	
PL	B/W/D	<>Error number>

For the meaning of the error number, See "Return codes of PLC Modules 9440-9459 (error stack)" on page 1249.

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Description was successfully unloaded
	1	Error. See <Error number>

Module 9459 Change or insert a value in a comma separated list

Module 9459 changes or inserts a value in a comma separated list of values. If the index is less than the number of values, the value in the list is replaced. If the index is equal to the number of values, the value is appended.

Constraints:

- The module can only be executed within a submit job.

Possible errors:

- The module was not called in a submit job
- The index for the value exceeds the number of values in the string

Call:

```
PS      B/W/D/K   <>String address for the list of values>
PS      B/W/D/K   <>Index of the value to insert>
PS      B/W/D/K   <>String address for the extracted value>
CM      9459
PL      B/W/D     <>Error number>
                  For the meaning of the error number, See "Return codes of
                     PLC Modules 9440-9459 (error stack)" on page 1249.
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Value was successfully updated or inserted
	1	Error. See <Error number>

Return codes of PLC Modules 9440-9459 (error stack)

The following errors can occur when running Modules 9440 to 9459, and are returned to the error stack:

Value (error stack)	Meaning
0	Module executed successfully
1	Parameter out of range
2	Parameter not defined
3	Invalid address programmed
4	Address too high or block too long
5	Specified address is not a double word address
11	String could not be converted
12	String too long
15	Module was not called in a submit job
16	No connection with SQL server established
17	Invalid transaction handle was programmed
20	Syntax of the statement is incorrect
25	Table file not found
26	Table file cannot be accessed
27	New file already exists
30	Table header is invalid
31	Configuration message is invalid
32	Table type has not been configured
33	Table contains no columns
34	Unexpected end of table
35	Table has already been opened
36	Table is write-protected
40	Column description is invalid
41	Column type was not configured
42	Column defined several times
43	Column already exists in table
44	Columns do not exist in table or record
45	No column given with the statement
50	Symbolic name already exists
51	Symbolic name does not exist
52	Symbolic name cannot be accessed
55	Index name already exists
56	Index name does not exist
57	Index must not be created
60	Data record already locked
61	Data record already deleted
62	Invalid length of a record
63	Index for result set too large
70	Invalid default value
71	Invalid value type

Value (error stack)	Meaning
72	Invalid number of values
73	Given value is not unique
75	Assigned value must not be null
76	Assigned value is invalid
77	Assigned value is too long
78	Assigned value is out of range
79	Assigned value already exists
80	Primary key must not be dropped or renamed
81	Primary key must not be updated
82	Primary key must not be set null
85	Action violates the referential integrity
86	Referential action conflicts with statement
90	Function not yet implemented
91	Internal (software) error

8.11 Data Transfer NC => PLC, PLC => NC

8.11.1 Introduction

Functions that are to be executed by the PLC are transferred from the NC to the PLC. The job (M, G or T function), the job parameter and the acknowledgment by the PLC are transmitted in strobes.

- M functions: Jobs from the NC to the PLC and acknowledgment by the PLC are transmitted in M strobes (See "M Functions (M Strobe)" on page 1148).
- Special G functions that are executed by the PLC must be configured correspondingly. This configuration is described below. Jobs from the NC to the PLC and acknowledgment by the PLC are transmitted in M strobes (See "M Functions (M Strobe)" on page 1148).
- T functions are executed by the PLC. Jobs from the NC to the PLC and acknowledgment by the PLC are transmitted in strobes (See "T Functions (T Strobe)" on page 1173).
- S functions are executed by the NC. However, an S strobe must be defined. To do so, define a dummy S strobe in the parameter object CfgPlcSStrobe (See "S Function (S Strobe)" on page 1160).

Settings in the configuration editor		MP number
System		
PLC		
CfgPlcMStrobe:	1148	
[Key name of M strobe]		
CfgPlcStrobeAlias		
[Key name of alias strobe]		
type:	GFUNCTION	104201
mCode:	1000	104202
mOffset:	TRUE	104203
NCchannel		
ChannelSettings		
[Key name of the machining channel]		
CfgPlcStrobes		
mStrobes		201601
aliasStrobes		201604

The NC transfers M functions to the PLC via M strobes. If G functions are transferred to the PLC, the following configuration is required:

- Channel-dependent parameter object
  NCchannel/ChannelSettings/key name of the channel/CfgPlcStrobes:
  - mStrobes: in addition to the key names of the M functions, also list the key names of the G functions to be transferred to the PLC
  - aliasStrobes: define the key name of the alias strobes
- Parameter object
  System/PLC/CfgPlcMStrobe:
  Use the key names to define the M and G functions and their transfer parameters. For G functions to be transferred to the PLC, the offset (here: 1000) is added to the number of the G function.
- Parameter object
  System/PLC/CfgPlcStrobeAlias:
  Use the key name to assign the type GFUNCTION to the alias strobe and define the offset.

With the parameters of the alias strobe, you make the following definitions:

- MP_type=GFUNCTION: (defined) G functions are transferred to the PLC
- MP_mCode=1000 and MP_mOffset=TRUE: Specifies that an offset (here. 1000) is added to the code of the G functions
  - Codes < 1000 are M functions
  - Codes > 1000 are G functions

On the basis of the entries in **CfgPlcMStrobe**, the NC checks which G functions are transferred to the PLC. Transfer is activated with the M strobe (See "M Functions (M Strobe)" on page 1148).

The following applies to data transfer (parameter of the M or G function):

- M and G functions (except G600...G699): The address of the operand containing the parameter is in MP_data
- G functions G600...G699: The operand specified in MP_data contains the address of a data field with the following structure:

Transfer date = parameter of the G function	Progr. max. value	Multiplier of the NC	Max. value transferred	Value if not programmed
"X"	9999.999	1000	9 999 999	10 000 000
"Z"	9999.999	1000	9 999 999	10 000 000
"Y"	9999.999	1000	9 999 999	10 000 000
"C"	9999.999	1000	9 999 999	10 000 000
"F"	9999.999	1000	9 999 999	10 000 000
"S"	9999	1	9 999	FFFF hex
"T"	9999	1	9 999	FFFF hex
"H"	9999	1	9 999	FFFF hex
"Q"	9999	1	9 999	FFFF hex

The REAL values listed in the NC program for the parameters X, Z, Y, C and F are multiplied by 1000 and transferred as DWORD (double word) values. The S, T, H and Q parameters are transferred as WORD values.

Overview of G functions

The machine tool builder defines the meaning of the G600 functions. The G functions that are also made available in the NC must be considered:

G code	G functions that are given to the PLC
Gx26	Speed limitation relative to the x spindle
Gx95	Feed rate per revolution relative to the x spindle
Gx96	Constant surface speed relative to the x spindle
Gx97	Constant speed relative to the x spindle

Note

The user should define the spindle speed and the speed limitation after system start-up to ensure correct values.

Settings in the configuration editor	MP number
System	
PLC	
CfgOemBool	
[Key name of arbitrary parameter]	
value	104501
ignorePlc	104502
CfgOemInt	
[Key name of arbitrary parameter]	
value	104601
ignorePlc	104602
CfgOemString	
[Key name of arbitrary parameter]	
value	104901
ignorePlc	104902
CfgOemPosition	
[Key name of arbitrary parameter]	
value	104701
ignorePlc	104702

Freely definable machine parameters are available for data transmission to the PLC. The control saves the contents of the machine parameters in PLC words.

In the machine parameters you can save, for example, values for PLC positioning movements and datum shifts, feed rates for PLC positioning movements or codes for the enabling of certain PLC functions or user texts for OEM cycles. You must evaluate the transferred characters and numerical values in your PLC program.

The freely definable machine parameters are divided into three groups:

- **CfgOemBool:**
  User parameters with logical values (TRUE, FALSE or ON, OFF)
- **CfgOemInt:**
  User parameters with integer values (whole numbers)
- CfgOemString
  User parameters with text data (lists with max. 9 strings, character length max. 100)
- **CfgOemPosition:**
  User parameters with fixed decimal values (position values)

Each of the freely definable parameters is in a subfolder (key name). The key name also specifies the name of the machine parameter. You specify the value of the parameter in the **value** subfolder.

If you do not want to copy a parameter to the PLC run-time system, the optional machine parameter **MP_ignorePlc** must be inserted and set to TRUE.

MP_value

List of user parameter values
Available from NCK software version: 597 110-01.

Format: Array
Input: Value of the user parameter
CfgOemBool: Logical values (Boolean)
CfgOemInt: Whole numbers (integer values)
CfgOemString: User text, e.g. for OEM cycles with max. 100 characters.
CfgOemPosition: Fixed-point values (position)
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

MP_ignorePlc

Do not copy user parameter to the PLC run-time system
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: TRUE
The parameter value is not copied to the PLC run-time system
FALSE
The parameter value is copied to the PLC run-time system
Default: No value, parameter optional
Access: LEVEL3
Reaction: NOTHING

User parameters

You can give the machine operator access via user parameters to the machine parameters that you define yourself, See "User Parameters" on page 277.

8.12 Program Creation

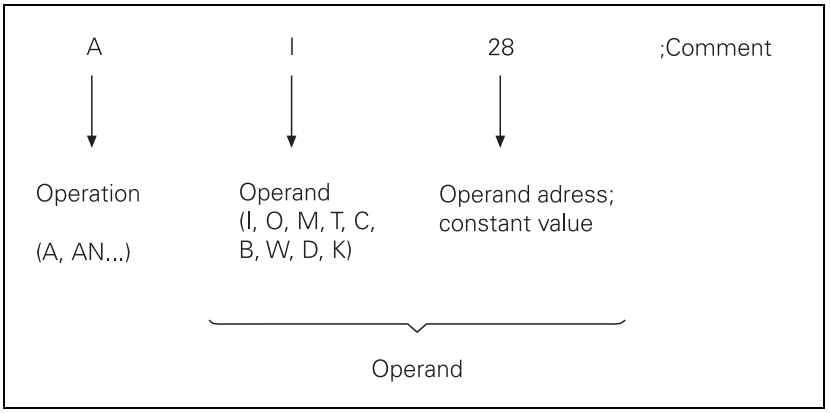
8.12.1 ASCII editor

The integrated editor enables you to create and edit the PLC program and all other necessary files right at the control through the ASCII keyboard. You will find a complete description of the editor, including the associated soft keys, in the User's Manual for the control.

8.12.2 Program format

Command

A command is the smallest unit of a PLC program. It consists of the operation part and the operand part.



The operation describes the function to be executed. It defines how the operand is to be processed by the TNC. The operand shows what is to be operated with. It consists of the operand abbreviation and a parameter (address). With the PLC commands you can combine (gate), delete and load register and memory contents, both with bit and word processing. For word processing, you can address memory contents with a length of 8 bits (byte), 16 bits (word) or 32 bits (double word).

8.12.3 Program structure

To make it easier to maintain and expand your PLC program, you should give it a modular structure. Modular means that you write a separate program module for each function. You can then call the individual modules from the main program. You should interrogate improper functioning of the machine in the PLC program and indicate such malfunctions on the screen with plain-language error messages.

Module 9019 Size of the processing stack

To debug functions, you can use Module 9019 to interrogate the contents of the processing stack. The function answers with the number of the bytes that lie on the processing stack of the PLC at the moment. If the processing stack is empty, the PLC run-time system returns the value zero. A byte, word or double word occupies four bytes on the stack; a marker, input, output, timer or counter occupies two bytes.

Call:

```
CM    9019
PL    B/W/D    <>Number of bytes on processing stack>
```

8.13 Command Set

8.13.1 Overview

The following table provides an overview of all commands explained in this chapter:

Group of functions	Syntax	Function
Loading and saving commands		
	L	Load
	LN	Load NOT
	L–	Load two's complement
	LB	Load BYTE
	LW	Load WORD
	LD	Load DOUBLE WORD
	=	Assignment
	B=	Assign BYTE
	W=	Assign WORD
	D=	Assign DOUBLE WORD
	=N	Assign NOT
	=–	Assign two's complement
Setting commands		
	S	Set
	R	Reset
	SN	Set NOT
	RN	Reset NOT
Logical operations		
	A	And
	AN	And NOT
	O	Or
	ON	Or NOT
	XO	Exclusive OR
	XON	Exclusive OR NOT
Arithmetic commands		
	+	Addition
	–	Subtraction
	x	Multiplication
	/	Division
	MOD	Remainder

Group of functions	Syntax	Function
Increment		
	INC	Increment operand
	INCW	Increment word accumulator
	INCX	Increment index register
Decrement		
	DEC	Decrement operand
	DECW	Decrement word accumulator
	DECX	Decrement index register
Comparisons		
	==	Equal to
	<	Less than
	>	Greater than
	<=	Less than or equal to
	>=	Greater than or equal to
	<>	Not equal to
Parenthetical expressions in logical operations		
	A[ ]	And [ ]
	AN[ ]	And NOT [ ]
	O[ ]	Or [ ]
	ON[ ]	Or NOT [ ]
	XO[ ]	Exclusive OR [ ]
	XON[ ]	Exclusive OR NOT [ ]
Parenthetical expressions with arithmetical instructions		
	+[ ]	Addition [ ]
	–[ ]	Subtraction [ ]
	x[ ]	Multiplication [ ]
	/[ ]	Division [ ]
	MOD[ ]	Remainder [ ]
Parenthetical expressions in comparisons		
	==[ ]	Equal to [ ]
	<[ ]	Less than [ ]
	>[ ]	Greater than [ ]
	<=[ ]	Less than or equal to [ ]
	>=[ ]	Greater than or equal to [ ]
	<>[ ]	Not equal to [ ]
Shifting commands		
	<<	Shift left
	>>	Shift right
Bit commands		
	BS	Bit set
	BC	Bit clear
	BT	Bit test

Group of functions	Syntax	Function
Stack operations		
	PS	Push data onto the data stack
	PL	Pull data from the data stack
	PSL	Push logic accumulator onto the data stack
	PSW	Push word accumulator onto the data stack
	PLL	Pull logic accumulator from the data stack
	PLW	Pull word accumulator from the data stack
Jump commands		
	JP	Unconditional jump
	JPT	Jump if logic accumulator = 1
	JPF	Jump if logic accumulator = 0
	CM	Call module
	CMT	Call module if logic accumulator = 1
	CMF	Call module if logic accumulator = 0
	EM	End of module, program end
	EMT	End of module if logic accumulator = 1
	EMF	End of module if logic accumulator = 0
	LBL	Label

8.13.2 LOAD (L)

Logic processing with the LOAD command

Syntax: L (LOAD)

Operands: M, I, O, T, C

Action:

Load the value of the addressed operand into the logic accumulator. Always use the L command at the beginning of a logic chain in order to be able to gate the operand in the following program sequence.

Example:

Gate the inputs I4 and I5 with AND, and assign the result to output O2.
Initial state:

Input	I4	= 1
Input	I5	= 0
Output	O2	= ?

Function	STL	Logic accumulator	Operand content
Load the operand content into the logic accumulator.	L I4	Logic accumulator = 1	
Gate the content of the logic accumulator and input I5 with AND.	A I5		0
Assign the gating result to output O2.	= O2		0

Word processing with the LOAD command

Syntax: L (LOAD)

Operands: B, W, D, K

Action:

Load the value of the addressed operand, or of a constant, into the word accumulator. If necessary, the accumulator is supplemented with the correct algebraic sign. In contrast to logical operations, you must always begin a sequence of word gating operations with an L command. You cannot replace the L command with a logical gating instruction.

Example:

Gate a constant and byte B5 with AND, and assign the result to byte B8.
Initial state:

Constant	54	= 36 (hex)
Byte	B5	= 2A (hex)
Output	B8	= ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K+54	36	
Gate the contents of the word accumulator and byte B5 with AND.	A B5		2A
Assign the gating result to byte B8.	= B8		22

8.13.3 LOAD NOT (LN)

Logic processing with the LOAD NOT command

Syntax: LN (LOAD NOT)

Operands: M, I, O, T, C

Action:

Load the one's complement of the addressed operand into the logic accumulator. Always use the L command at the beginning of a logic chain in order to be able to gate the operand in the following program sequence.

Example:

Gate the inverted logical state of inputs I4 and I5 with AND, and assign the result to output O2.
Initial state:
Input I4 = 0
Input I5 = 1
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the inverted operand content into the logic accumulator.	LN I4	0	
Gate the content of the logic accumulator and input I5 with AND.	A I5		1
Assign the gating result to output O2.	= O2		1

Word processing with the LOAD NOT command

Syntax: LN (LOAD NOT)

Operands: B, W, D, K

Action:

Load the complement of the addressed operand, or of a constant, into the word accumulator. If necessary, the accumulator is supplemented with the correct algebraic sign. In contrast to logical operations, you must always begin a sequence of word gating operations with an L command. You cannot replace the L command with a logical gating instruction.

Example:

Gate the complement of byte B6 and byte B5 with AND, and assign the result to byte B8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = B6 (hex)
Byte B8 = ?

Function	STL	Accumulator content	Operand content
Invert byte B6, and load into the word accu.	LN B6	2A	
Gate the contents of the word accumulator and byte B5 with AND.	A B5		B6
Assign the gating result to byte B8.	= B8		22

8.13.4 LOAD TWO'S COMPLEMENT (L–)

Syntax: L– (LOAD MINUS)

Operands: B, W, D, K

Action:

Load the two's complement of the addressed operand, or of a constant, into the word accumulator. If necessary, the control fills the accumulator with the correct algebraic sign. The two's complement allows negative numbers to be stored, i.e., a number loaded with the L command appears in the accumulator with an inverted sign. This command can be used only with word processing.

Example:

Negate the content of byte B5 and then add it to the content of byte B6. Assign the result to byte B8.
Initial state:
Byte B5 = 15 (dec)
Byte B6 = 20 (dec)
Byte B8 = ?

Function	STL	Accumulator content	Operand content
Load byte B5 into the word accumulator, invert the algebraic sign.	L– B5	–15	+15
Add the contents of the word accumulator and byte B6.	+B6	+5	+20
Assign the gating result to byte B8.	= B8	+5	+5

8.13.5 LOAD BYTE (LB)

Syntax: LB (LOAD BYTE)

Operands: M, I, O, T, C

Action:

Copy 8 markers, inputs, outputs, timers or counters with ascending numbering into the word accumulator. Each operand occupies one bit in the accumulator. The control saves the given operand address as LSB in the accumulator, the given address + 1 as LSB + 1 and so on. The last (8th) operand becomes the MSB! If necessary, the control fills the accumulator with the correct algebraic sign.

Example:

A pure-binary coded value is read through inputs I3 to I10 and saved in byte B8 in order to process it later.
Initial state:

Input	I3	= 1	Input	I7	= 0
Input	I4	= 1	Input	I8	= 1
Input	I5	= 1	Input	I9	= 1
Input	I6	= 0	Input	I10	= 0

Function	**STL**	**Accumulator content**								**Operand content**							
		7	6	5	4	3	2	1	0	I10	I9	I8	I7	I6	I5	I4	I3
Load inputs I3 to I10 into the accumulator (bit 0 to bit 7).	LB I3	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1
										7	6	5	4	3	2	1	0
Assign accumulator contents to byte 8.	= B8	1	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0

8.13.6 LOAD WORD (LW)

Syntax: LW (LOAD WORD)

Operands: M, I, O, T, C

Action:

Copy 16 markers, inputs, outputs, timer or counters with ascending numbering into the word accumulator. Each operand occupies one bit in the accumulator. The control saves the given operand address as LSB in the accumulator, the given address + 1 as LSB + 1 and so on. The last (16th) operand becomes the MSB! If necessary, the control fills the accumulator with the correct algebraic sign.

Example:

See example command LB. Use command LW in the same way as LB. However, the control processes 16 operands.

8.13.7 LOAD DOUBLE WORD (LD)

Syntax: LD (LOAD DOUBLE WORD)

Operands: M, I, O, T, C

Action:

Copy 32 markers, inputs, outputs, timers or counters with ascending numbering into the word accumulator. Each operand occupies one bit in the accumulator. The control saves the given operand address as LSB in the accumulator, the given address + 1 as LSB + 1 and so on. The last (32th) operand becomes the MSB! If necessary, the control fills the accumulator with the correct algebraic sign.

Example:

See example command LB. Use command LD in the same way as LB. However, the control processes 32 operands.

8.13.8 ASSIGN (=)

Logic processing with the ASSIGN command

Syntax: = (STORE)

Operands: M, I, O, T, C

Action:

Assign the content of the logic accumulator to the addressed operand. Use the = command only at the end of a sequence of logical gating operations in order to transfer a gating result to a logic operand. This command can be used several times in succession (see example).

Example:

Gate the inputs I4 and I5 with AND, and assign the result to outputs O2 and O5.
Initial state:

Input	I4	= 1
Input	I5	= 0
Output	O2	= ?
Output	O5	= ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of the logic accumulator and input I5 with AND.	A I5	0	0
Assign the gating result to output O2.	= O2	0	0
Assign the gating result to output O5.	= O5	0	0

Word processing with the ASSIGN command

Syntax: = (STORE)

Operands: B, W, D

Action:

Assign the content of the word accumulator to the addressed operand. Unlike bit processing, in word processing you can also use the = command within a sequence of word-gating operations. This command can be used several times in succession.

Example:

Gate a constant and byte B5 with AND, and assign the result to byte B8 and byte B10.
Initial state:

Constant	54	= 36 (hex)
Byte	B5	= 2A (hex)
Byte	B8	= ?
Byte	B10	= ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K+54	36	
Assign the contents of the word accumulator to byte B8.	= B8	36	36
Gate the contents of the word accumulator and byte B5 with AND.	A B5	22	2A
Assign the gating result to byte B8.	= B8	22	22
Assign the gating result to byte B10.	= B10	22	22

8.13.9 ASSIGN BYTE (B=)

Syntax: B= (STORE BYTE)

Operands: M, I, O, T, C

Action:

Assign 8 bits from the word accumulator to markers, inputs, outputs, timers or counters with ascending numbering. Every bit occupies an operand. The control assigns the LSB in the accumulator to the operand address specified in the command, the specified address +1 as LSB +1 etc. The last (8th) operand is assigned the MSB.

Example:

See example command W=. Use command B= in the same way as W=. However, the control processes 8 operands.

8.13.10 ASSIGN WORD (W=)

Syntax: W= (STORE WORD)

Operands: M, I, O, T, C

Action:

Assign 16 bits from the word accumulator to markers, inputs, outputs, timers or counters with ascending numbering. Every bit occupies an operand. The control assigns the LSB in the accumulator to the operand address specified in the command, the specified address +1 as LSB +1 etc. The last (16th) operand is assigned the MSB.

Example:

Transfer a certain bit pattern, located in word W8, to the output addresses O1 to O16.
Initial state:
Word W8 = 36FF (hex)

Function	STL	Accumulator content	Operand content
Load content of word W8 into the word accumulator.	L W8	36FF	
			O16 ... O1
Assign accumulator content to outputs O5 to O20.	W= O1	36FF	0 1 1 0 1 1 0 1 1 1 1 1 1 1 1

8.13.11 ASSIGN DOUBLE WORD (D=)

Syntax: D= (STORE DOUBLE WORD)

Operands: M, I, O, T, C

Action:

Assign 32 bits from the word accumulator to markers, inputs, outputs, timers or counters with ascending numbering. Every bit occupies an operand. The control assigns the LSB in the accumulator to the operand address specified in the command, the specified address +1 as LSB +1 etc. The last (32nd) operand is assigned the MSB.

Example:

See example command W=. Use command D= in the same way as W=. However, the control processes 32 operands.

8.13.12 ASSIGN NOT (=N)

Logic processing

Syntax: =N (STORE NOT)

Operands: M, I, O, T, C

Action:

Assign the complement of the logic accumulator to the addressed operand. For an example, see the ASSIGN (=) command.

Word processing

Syntax: =N (STORE NOT)

Operands: B, W, D

Action:

Assign the complement of the word accumulator to the addressed operand. For an example, see the ASSIGN (=) command.

8.13.13 ASSIGN TWO'S COMPLEMENT (=–)

Syntax: =– (STORE MINUS)

Operands: B, W, D

Action:

Assign the TWO'S COMPLEMENT of the word accumulator to the addressed operand. For an example, see the ASSIGN (=) command.

8.13.14 SET (S)

Syntax: S (SET)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 1, then set the addressed operand to 1, otherwise do not change it. Use the S command at the end of a sequence of logical gating operations in order to influence an operand, depending on the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 1, then set output O2 and marker M500.
Initial state:

Input	I4	= 1
Input	I5	= 0
Output	O2	= ?
Marker	M500	= ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of the logic accumulator and input I5 with OR.	O I5	1	0
Since the result of the operation is 1, set output O2.	S O2	1	1
Since the result of the operation is 1, set marker M500.	S M500	1	1

8.13.15 RESET (R)

Syntax: R (RESET)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 1, then set the addressed operand to 0, otherwise do not change it. Use the R command at the end of a sequence of logical gating operations in order to influence an operand, depending on the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 1, then reset output O2 and marker M500.
Initial state:

Input	I4	= 1
Input	I5	= 0
Output	O2	= ?
Marker	M500	= ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of the logic accumulator and input I5 with OR.	O I5	1	0
Since the result of the operation is 1, reset output O2.	R O2	1	0
Since the result of the operation is 1, reset marker M500.	R M500	1	0

8.13.16 SET NOT (SN)

Syntax: SN (SET NOT)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 0, then set the addressed operand to 1, otherwise do not change it. Use the SN command at the end of a sequence of logical gating operations in order to influence an operand depending on the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 0, then set output O2 and marker M500.
Initial state:

Input	I4	= 0
Input	I5	= 0
Output	O2	= ?
Marker	M500	= ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	0	0
Gate the content of the logic accumulator and input I5 with OR.	O I5	0	0
Since the result of the operation is 0, set output O2.	SN O2	0	1
Since the result of the operation is 0, set marker M500.	SN M500	0	1

8.13.17 RESET NOT (RN)

Syntax: RN (RESET NOT)

Operands: M, I, O, T, C

Action:

If the logic accumulator = 0, then set the addressed operand to 0, otherwise do not change it. Use the RN command at the end of a sequence of logical gating operations in order to influence an operand depending on the result of gating. This command can be used several times in succession (see example).

Example:

Gate input I4 and I5 with OR. If the gating result is 0, then reset output O2 and marker M500.
Initial state:
Input I4 = 0
Input I5 = 0
Output O2 = ?
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	0	0
Gate the content of the logic accumulator and input I5 with OR.	O I5	0	0
Since the result of the operation is 0, reset output O2.	RN O2	0	0
Since the result of the operation is 0, reset marker M500.	RN M500	0	0

8.13.18 AND (A)

Logic processing with the AND command

Syntax: A (AND)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an L command, i.e., the logical state of the operand is loaded into the logic accumulator. This is to ensure compatibility with the TNC 355, which does not have the special L command. In PLC programs, a sequence of logical gating operations should always be started with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with AND. The control saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with AND, and assign the result to output O2.
Initial state:

Input	I4	= 1
Input	I5	= 0
Output	O2	= ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of the logic accumulator and input I5 with AND.	A I5	0	1
Assign the gating result to output O2.	= O2	0	0

Word processing with the AND command

Syntax: A (AND)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with AND. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The control saves the result of the operation in the word accumulator.

Example:

Gate the contents of byte B5 and byte B6 with AND, and assign the result to byte B8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = 36 (hex)
Byte B8 = ?

Function	STL	Accumulator content	Operand content
Load byte B6 into the word accumulator.	L B6	2A	2A
Gate the contents of the word accumulator and byte B5 with AND.	A B5	22	36
Assign the gating result to byte B8.	= B8	22	22

8.13.19 AND NOT (AN)

Logic processing with the AND NOT command

Syntax: AN (AND NOT)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an LN command, i.e., the logical state of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with AND NOT. The control saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with AND NOT, and assign the result to output O2.
Initial state:
Input I4 = 1
Input I5 = 1
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of logic accumulator and input I5 with AND NOT.	AN I5	1	1
Assign the gating result to output O2.	= O2	1	1

Word processing with the AND NOT command

Syntax: AN (AND NOT)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with AND NOT. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The control saves the result of the operation in the word accumulator.

Example:

Gate the content of words W4 and W6 with AND NOT, and assign the result to word W8.
Initial state:
Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load W6 into the word accumulator.	L W6	3C36	3C36
Gate the contents of word accumulator and word W4 with AND NOT.	AN W4	814	36AA
Assign the gating result to word W8.	= W8	814	814

Logic processing with the OR command

Syntax: O (OR)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an L command, i.e., the logical state of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with OR. The control saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with OR, and assign the result to output O2.
Initial state:
Input I4 = 0
Input I5 = 1
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	0	0
Gate the content of the logic accumulator and input I5 with OR.	O I5	1	1
Assign the gating result to output O2.	= O2	1	1

Word processing with the OR command

Syntax: O (OR)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with OR. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The control saves the result of the operation in the word accumulator.

Example:

Gate the content of byte B5 and byte B6 with OR, and assign the result to word W8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load byte B6 into the word accumulator.	L B6	36	36
Gate the contents of the word accumulator and byte B5 with OR.	O B5	3E	2A
Assign the gating result to word W8.	= W8	3E	3E

8.13.21 OR NOT (ON)

Logic processing with the OR NOT command

Syntax: ON (OR NOT)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an LN command, i.e., the complement of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with OR NOT. The control saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with OR NOT, and assign the result to output O2.
Initial state:
Input I4 = 0
Input I5 = 0
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	0	0
Gate the content of logic accumulator and input I5 with OR NOT.	ON I5	1	0
Assign the gating result to output O2.	= O2	1	1

Word processing with the OR NOT command

Syntax: ON (OR NOT)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with OR NOT. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The control saves the result of the operation in the word accumulator.

Example:

Gate the content of words W4 and W6 with OR NOT, and assign the result to word W8.
Initial state:
Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load W6 into the word accumulator.	L W6	3C36	3C36
Gate the contents of word accumulator and word W4 with OR NOT.	ON W4	814	36AA
Assign the gating result to word W8.	= W8	814	814

8.13.22 EXCLUSIVE OR (XO)

Logic processing with the EXCLUSIVE OR command

Syntax: XO (EXCLUSIVE OR)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an L command, i.e., the logical state of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with EXCLUSIVE OR. The control saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and I5 with EXCLUSIVE OR, and assign the result to output O2.
Initial state:
Input I4 = 1
Input I5 = 1
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I4	1	1
Gate the content of logic accumulator and input I5 with EXCLUSIVE OR.	XO I5	0	1
Assign the gating result to output O2.	= O2	0	0

Word processing with the EXCLUSIVE OR command

Syntax: XO (EXCLUSIVE OR)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with EXCLUSIVE OR. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The control saves the result of the operation in the word accumulator.

Example:

Gate the contents of byte B5 and byte B6 with EXCLUSIVE OR, and assign the result to word W8.
Initial state:
Byte B5 = 2A (hex)
Byte B6 = 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load byte B6 into the word accumulator.	L B6	36	36
Gate the contents of the word accumulator and byte B5 with EXCLUSIVE OR.	XO B5	1C	2A
Assign the gating result to word W8.	= W8	1C	1C

8.13.23 EXCLUSIVE OR NOT (XON)

Logic processing with the EXCLUSIVE OR NOT command

Syntax: XON (EXCLUSIVE OR NOT)

Operands: M, I, O, T, C

Action:

- At the beginning of a logic sequence, this command functions like an LN command, i.e., the logical state of the operand is loaded into the logic accumulator. You should always begin a sequence of logical gating operations with a load command (see L, LN, L–).
- Within a logic sequence, gate the content of the logic accumulator and the logical state of the operand with EXCLUSIVE OR NOT. The control saves the result of the operation in the logic accumulator.

Example:

Gate the inputs I4 and marker M500 with EXCLUSIVE OR NOT, and assign the result to output O2.
Initial state:
Input I4 = 0
Marker M500 = 0
Output O2 = ?

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L M500	0	0
Gate the content of logic accumulator and input I4 with EXCLUSIVE OR NOT.	XON I4	1	0
Assign the gating result to output O2.	= O2	1	1

Word processing with the EXCLUSIVE OR NOT command

Syntax: XON (EXCLUSIVE OR NOT)

Operands: B, W, D, K

Action:

Gate the contents of the word accumulator and the operand with EXCLUSIVE OR NOT. In accordance with the different data widths of the operands (B = 8 bits; W = 16 bits; D = K = 32 bits), 8, 16 or 32 bits, respectively, are influenced in the accumulator. Thus, bit 0 of the accumulator is gated with bit 0 of the operand, bit 1 of the accumulator with bit 1 of the operand, etc. The control saves the result of the operation in the word accumulator.

Example:

Gate the content of words W4 and W6 with EXCLUSIVE OR NOT, and assign the result to word W8.
Initial state:
Word W4 = 36 AA (hex)
Word W6 = 3C 36 (hex)
Word W8 = ?

Function	STL	Accumulator content	Operand content
Load W6 into the word accumulator.	L W6	3C36	3C36
Gate the contents of word accumulator and word W4 with EXCLUSIVE OR NOT.	XON W4	FFFFF563	36AA
Assign the gating result to word W8.	= W8	FFFFF563	FFFFF563

8.13.24 ADDITION (+)

Syntax: + (PLUS)

Operands: B, W, D, K

Action:

The control extends the operand to the width of the accumulator (32 bits) and then adds the content of the operand to the content of the word accumulator. The result of the operation is stored in the word accumulator where you can process it further.

Example:

Add the constant and the number saved in word W6, then assign the result to double word D8.
Initial state:
Constant = 10 000 (dec)
Word W6 = 200 (dec)
Double word D8 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K10000	10000	
Add the content of the word accumulator and word W6.	+ W6	10200	200
Assign the result to double word D8.	= D8	10200	10200

8.13.25 SUBTRACTION (–)

Syntax: – (MINUS)

Operands: B, W, D, K

Action:

The control extends the operand to the width of the accumulator (32 bits) and then subtracts the content of the operand from the content of the word accumulator. The result of the operation is stored in the word accumulator where you can process it further.

Example:

Subtract the number saved in word W6 from the constant, and then assign the result to double word D8.
Initial state:

Constant		= 10 000 (dec)
Word	W6	= 200 (dec)
Double word	D8	= ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K10000	10000	
Subtract word W6 from the content of the word accumulator.	– W6	9800	9800
Assign the result to double word D8.	= D8	9800	9800

8.13.26 MULTIPLICATION (X)

Syntax: x (MULTIPLY)

Operands: B, W, D, K

Action:

The control extends the operand to the width of the accumulator (32 bits) and then multiplies the content of the operand by the content of the word accumulator. The result of the operation is stored in the word accumulator where you can process it further. If the control cannot execute the multiplication correctly, it then sets marker M4200, otherwise it resets it.

Example:

Multiply the constant by the number saved in word W6, then assign the result to double word D8.
Initial state:
Constant = 100 (dec)
Word W6 = 20 (dec)
Double word D8 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K100	100	
Multiply the content of the word accumulator by word W6.	x W6	2000	20
Assign the result to double word D8.	= D8	2000	2000

8.13.27 DIVISION (/)

Syntax: / (DIVIDE)

Operands: B, W, D, K

Action:

The control extends the operand to the width of the accumulator (32 bits) and then divides the content of the word accumulator by the content of the operand. The result of the operation is stored in the word accumulator where you can process it further. If the control cannot execute the division correctly, it then sets marker M4201, otherwise it resets it.

Example:

Divide the constant by the number saved in word W6, then assign the result to double word D8.
Initial state:
Constant = 100 (dec)
Word W6 = 20 (dec)
Double word D8 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K100	100	
Divide the content of the word accumulator by word W6	/ W6	5	20
Assign the result to double word D8.	= D8	5	5

8.13.28 REMAINDER (MOD)

Syntax: MOD (MODULO)

Operands: B, W, D, K

Action:

The control extends the operand to the width of the accumulator (32 bits) and then calculates the remainder resulting from the division of the content of the word accumulator by the content of the operand. The remainder is stored in the word accumulator where you can process it. If the control cannot execute the MOD command correctly, it sets marker M4202, otherwise it resets it.

Example:

Divide the number saved in word W6 by the constant, then calculate the REMAINDER and assign the result to double word D8.
Initial state:

Word	W6	= 50 (dec)
Constant		= 15 (dec)
Double word	D8	= ?

Function	STL	Accumulator content	Operand content
Load W6 into the word accumulator.	L W6	50	50
Divide the content of the word accumulator by a constant, then save the integral REMAINDER in the word accumulator.	MOD K15	11	15
Assign the REMAINDER to double word D8.	= D8	11	11

8.13.29 INCREMENT (INC)

INCREMENT operand

Syntax: INC (INCREMENT)

Operands: B, W, D

Action:

Increase the content of the addressed operand by one.

INCREMENT word accumulator

Syntax: INCW (INCREMENT WORD)

Operands: None

Action:

Increase the content of the word accumulator by one.

INCREMENT index register

Syntax: INCX (INCREMENT INDEX)

Operands: None

Action:

Increase the content of the index register by one.

8.13.30 DECREMENT (DEC)

DECREMENT operand

Syntax: DEC (DECREMENT)

Operands: B, W, D

Action:

Decrease the content of the addressed operand by one.

DECREMENT word accumulator

Syntax: DECW (DECREMENT WORD)

Operands: None

Action:

Decrease the content of the word accumulator by one.

DECREMENT index register

Syntax: DECX (DECREMENT INDEX)

Operands: None

Action:

Decrease the content of the index register by one.

8.13.31 EQUAL TO (==)

Syntax: == (EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator and the operand are equal, the condition is true and the control sets the logic accumulator to 1. If they are not equal, the logic accumulator is set to 0. The comparison takes place over the number of bits corresponding to the operand:
where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M300 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Compare the content of the word accumulator with the operand content D8; if not equal, set the logic accumulator to 0.	== D8	0	15000
Assign the result to marker M500.	= M500	0	0

8.13.32 LESS THAN (<)

Syntax: < (LESS THAN)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator is less than the operand, the condition is true and the control sets the logic accumulator to 1. If the word accumulator is greater than or equal to the operand, it sets the logic accumulator to 0. The comparison takes place over the number of bits in the operand:
where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator < operand; if not, set logic accumulator to 0.	< D8	0	15000
Assign the result to marker M500.	= M500	0	0

8.13.33 GREATER THAN (>)

Syntax: > (GREATER THAN)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator is greater than the operand, the condition is true and the control sets the logic accumulator to 1. If the word accumulator is less than or equal to the operand, it sets the logic accumulator to 0. The comparison takes place over the number of bits in the operand: where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator > operand; if so, set logic accumulator to 1.	> D8	1	15000
Assign the result to marker M500.	= M500	1	1

8.13.34 LESS THAN OR EQUAL TO (<=)

Syntax: <= (LESS EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator is less than or equal to the operand, the condition is true and the control sets the logic accumulator to 1. If the word accumulator is greater than the operand, it sets the logic accumulator to 0. The comparison takes place over the number of bits in the operand:
where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator <= operand; if not, set logic accumulator to 0.	<= D8	0	15000
Assign the result to marker M500.	= M500	0	0

8.13.35 GREATER THAN OR EQUAL TO (>=)

Syntax: >= (GREATER EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator is greater than or equal to the operand, the condition is true and the control sets the logic accumulator to 1. If the word accumulator is less than the operand, it sets the logic accumulator to 0. The comparison takes place over the number of bits in the operand: where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator >= operand; if so, set logic accumulator to 1.	>= D8	1	15000
Assign the result to marker M500.	= M500	1	1

8.13.36 NOT EQUAL (<>)

Syntax: <> (NOT EQUAL)

Operands: B, W, D, K

Action:

This command sets off a direct transition from word to logical processing. Compare the content of the word accumulator with the content of the addressed operand. If the word accumulator and the operand are not equal, the condition is true and the control sets the logic accumulator to 1. If they are equal, the logic accumulator is set to 0. The comparison takes place over the number of bits corresponding to the operand:
where B = 8 bits, W = 16 bits, and D = K = 32 bits.

Example:

Compare a constant with the content of double word D8, and assign the result to marker M500.
Initial state:
Constant = 16 000 (dec)
Double word D8 = 15 000 (dec)
Marker M500 = ?

Function	STL	Accumulator content	Operand content
Load the constant into the word accumulator.	L K16000	16000	
Check whether word accumulator <> operand; if so, set logic accumulator to 1.	<> D8	1	15000
Assign the result to marker M500.	= M500	1	1

8.13.37 AND [ ] (A[ ])

Syntax: A[ ] (AND [ ])

Operands: None

Action:

The use of parentheses enables you to alter the sequence of processing logical commands in a statement list. The opening-parenthesis command puts the content of the accumulator onto the program stack. If you address the logic accumulator in the last command before an opening-parenthesis statement, the control loads the content of the logic accumulator onto the program stack. If you address the word accumulator, the control loads the contents of the word accumulator. The "close-parenthesis" statement gates the buffered value from the program stack with the content of the logic accumulator or the word accumulator, depending on which accumulator was addressed prior to the "open-parenthesis" instruction. The control assigns the result of the gating operation to the corresponding accumulator. Maximum nesting depth: 16 parentheses.

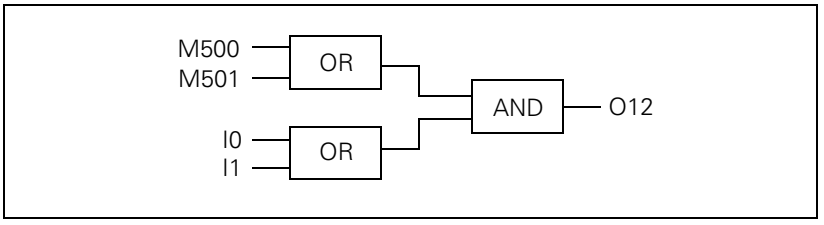
Please note:

The sequence of function is the same for word processing; however, the control writes the complete word accumulator onto the program stack.

Example:

Example for the commands AND [ ], AND NOT [ ], OR [ ], OR NOT [ ], EXCLUSIVE OR [ ], EXCLUSIVE OR NOT [ ]:

Use parentheses to develop a statement list in accordance with the following logic circuit diagram:



Initial state:

Marker	M500	= 0
Input	I0	= 0
Marker	M501	= 1
Input	I1	= 1
Output	O12	= ?

Function	STL	Accumulator content	Operand content
Load marker M500 into the logic accumulator.	L M500	0	0
Gate logic accumulator with marker M501.	O M501	1	1
Opening parenthesis: Buffer the accumulator content onto the program stack.	A[		
Load the state of input I0 into the logic accumulator.	L I0	0	0
Gate the logic accumulator with the state of input I1.	O I1	1	1
Closing parenthesis: Gate the accumulator content with the program stack (A[, O[...).	]		
Assign the result of the total operation to output O12.	= O12	1	1

8.13.38 AND NOT [ ] (AN[ ])

Syntax: AN[ ] (AND NOT [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

8.13.39 OR [ ] (O[ ])

Syntax: O[ ] (OR [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

8.13.40 OR NOT [ ] (ON[ ])

Syntax: ON[ ] (OR NOT [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

8.13.41 EXCLUSIVE OR [ ] (XO[ ])

Syntax: XO[ ] (EXCL: OR [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

8.13.42 EXCLUSIVE OR NOT [ ] (XON[ ])

Syntax: XON[ ] (EXCL: OR NOT [ ])

Operands: None

Action:

See example of command A[ ] (AND [ ])

8.13.43 ADDITION [ ] (+[ ])

Syntax: +[ ] (PLUS[ ])

Operands: None

Action:

Use parentheses together with arithmetical commands **only** for word processing. By using parentheses you can change the sequence of processing in a statement list. The opening-parenthesis command puts the content of the word accumulator onto the program stack. This clears the accumulator for calculation of intermediate results. The closing-parenthesis command gates the buffered value from the program stack with the content of the word accumulator. The control saves the result in the accumulator again. Maximum nesting depth: 16 parentheses. If an error occurs during calculation, the control sets the marker M4201.

Example:

Example for the commands ADD [ ], SUBTRACT [ ], MULTIPLY [ ], DIVIDE [ ], REMAINDER [ ].

Divide a constant by double word D36, add the result to double word D12, and assign the result to double word D100.
Initial state:
Constant = 1000 (dec)
Double word D12 = 15000 (dec)
Double wordD36 = 100 (dec)
Double word D100 = ?

Function	STL	Accumulator content	Operand content
Load the double word D12 into the word accumulator.	L D12	15000	15000
Opening parenthesis: Buffer the accumulator content onto the program stack.	+[		
Load the constant K1000 into the word accumulator.	L K1000	1000	
Divide the word accumulator by the content of the double word D36.	/ D36	10	100
Closing parenthesis: Gate the accumulator content with the program stack (+[, –[.....).	]		
Assign the result of the total operation to double word D100.	= D100	15010	15010

8.13.44 SUBTRACT [ ] (–[ ])

Syntax: –[ ] (MINUS –[ ])

Operands: None

Action:

See example of command ADD [ ]

8.13.45 MULTIPLY [ ] (x[ ])

Syntax: x[ ] (MULTIPLY [ ])

Operands: None

Action:

See example of command ADD [ ]

8.13.46 DIVIDE [ ] (/[ ])

Syntax: /[ ] (DIVIDE [ ])

Operands: None

Action:

See example of command ADD [ ]

8.13.47 REMAINDER [ ] (MOD[ ])

Syntax: MOD[ ] (MODULO [ ])

Operands: None

Action:

See example of command ADD [ ]

8.13.48 EQUAL TO [ ] (==[ ])

Syntax: ==[ ] (EQUAL[ ])

Operands: None

Action:

By using parentheses you can change the sequence of processing comparative commands in a statement list. The opening-parenthesis command puts the content of the word accumulator onto the program stack. This clears the accumulator for calculation of intermediate results.

The closing-parenthesis command gates the buffered value from the program stack with the content of the word accumulator. The control saves the result in the accumulator again. Maximum nesting depth: 16 parentheses.

Comparative commands cause a direct transition from word to logical processing. If the specified comparative condition is true, the control sets the logic accumulator to 1; if the condition is not fulfilled, it sets it to 0.

See next page for example.

Example:

Multiply a constant by double word D36, compare the result to double word D12, and assign the result to output O15.
Initial state:
Constant = 1000 (dec)
Double word D12 = 15000 (dec)
Double word D36 = 10 (dec)
Output O15 = ?

Function	STL	Accumulator content	Operand content
Load the double word D12 into the word accumulator.	L D12	15000	15000
Opening parenthesis: Buffer the accumulator content onto the program stack.	== [		
Load the constant into the word accumulator.	L K1000	1000	
Multiply the content of the word accumulator by double word D36.	x D36	10000	10
Closing parenthesis: Gate the accumulator content with the program stack (==[, >=[...); if condition not fulfilled, set logic accumulator to 0.	]		
Assign the result to output O15.	= O15	0	0

8.13.49 LESS THAN [ ] (<[ ])

Syntax: <[ ] (LESS THAN [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

8.13.50 GREATER THAN [ ] (>[ ])

Syntax: >[ ] (GREATER THAN [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

8.13.51 LESS THAN OR EQUAL TO [ ] (<=[ ])

Syntax: <=[ ] (LESS EQUAL [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

8.13.52 GREATER THAN OR EQUAL TO [ ] (>=[ ])

Syntax: >=[ ] (GREATER EQUAL [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

8.13.53 NOT EQUAL [ ] (<>[ ])

Syntax: <>[ ] (NOT EQUAL [ ])

Operands: None

Action:

See example of command EQUAL TO [ ]

8.13.54 SHIFT LEFT (<<)

Syntax: << (SHIFT LEFT)

Operands: B, W, D, K

Action:

A SHIFT LEFT statement multiplies the content of the word accumulator by two. This is done by simply shifting the bits in the accumulator by one place to the left. The result must lie in the range of -2 147 483 648 to +2 147 483 647, otherwise the accumulator contains an undefined value. You define the number of shifts through the operand. The control fills the right end of the accumulator with zeros.

This statement is one of the arithmetic commands because it includes the sign bit. For this reason, and to save time, you should not use this command to isolate bits.

Example:

Shift the content of double word D8 four times to the left, then assign it to double word D12.
Initial state:
Double word D8 = 3E 80 (hex)
Double word D12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E80	3E80
Shift the content of the word accumulator to the left by the number of bits that are specified in the operand.	<< K+1	7D00	
	<< K+1	FA00	
	<< K+1	1F400	
	<< K+1	3E800	
Assign the result to double word D12.	= D12	3E800	3E800

Instead of using the << K+1 command four times, simply use the << K+4 command.

8.13.55 SHIFT RIGHT (>>)

Syntax: >> (SHIFT RIGHT)

Operands: B, W, D, K

Action:

A SHIFT RIGHT statement divides the content of the word accumulator by two. This is done by simply shifting the bits by one place to the right. You define the number of shifts through the operand. The bits that the control shifts to the right out of the accumulator are then lost. The control extends the left side of the accumulator with the correct sign.

This statement is one of the arithmetic commands because it includes the sign bit. For this reason, and to save time, you should not use this command to isolate bits.

Example:

Shift the content of double word D8 four times to the right, then assign it to double word 12.
Initial state:
Double word D8 = 3E 80 (hex)
Double word D12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E80	3E80
Shift the content of the word accumulator to the right by the number of bits that are specified in the operand.	>> K+1	1F40	
	>> K+1	FA0	
	>> K+1	7D0	
	>> K+1	3E8	
Assign the result to double word D12.	= D12	3E8	3E8

Instead of using the >> K+1 command four times, simply use the >> K+4 command.

8.13.56 BIT SET (BS)

Syntax: BS (BIT SET)

Operands: B, W, D, K, X

Action:

With the BIT SET command you can set each bit in the accumulator to 1. The corresponding bits are selected (addressed) by the content of the specified operand or by a constant. As to the bit numbering, bit 0 = LSB and bit 31 = MSB. For operand contents greater than 32, the control uses the operand value modulo 32, i.e. the integral remainder of the result of the operand value divided by 32.

Example:

Load double word D8 into the accumulator, set bit 0 of the accumulator to 1, and save the result in double word D12.
Initial state:
Double word D8 = 3E 80 (hex)
Double word D12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E80	3E80
Set the bit specified in the operand to 1.	BS K+0	3E81	
Assign the result to double word D12.	= D12	3E81	3E81

8.13.57 BIT CLEAR (BC)

Syntax: BC (BIT CLEAR)

Operands: B, W, D, K, X

Action:

The BIT CLEAR command is used to set each bit in the accumulator to 0. The corresponding bits are selected (addressed) by the content of the specified operand or by a constant. As to the bit numbering, bit 0 = LSB and bit 31 = MSB. For operand contents greater than 32, the control uses the operand value modulo 32, i.e. the integral remainder of the result of the operand value divided by 32.

Example:

Load double word D8 into the accumulator, set bit 0 of the accumulator to 0, and save the result in double word D12.
Initial state:
Double word D8 = 3E 81 (hex)
Double word D12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E81	3E81
Set the bit specified in the operand to 0.	BC K+0	3E80	
Assign the result to double word D12.	= D12	3E80	3E80

8.13.58 BIT TEST (BT)

Syntax: BT (BIT TEST)

Operands: B, W, D, K, X

Action:

With the BIT TEST command, you can interrogate the status of each bit in the accumulator. With the BT command there is a direct transition from word to logic processing, i.e. the control checks the state of a bit in the word accumulator and then sets the logic accumulator. If the interrogated bit = 1, the control sets the logic accumulator to 1; otherwise it sets it to 0. The corresponding bits are selected (addressed) by the content of the specified operand or by a constant. As to the bit numbering, bit 0 = LSB and bit 31 = MSB. For operand contents greater than 32, the control uses the operand value modulo 32, i.e. the integral remainder of the result of the operand value divided by 32.

Example:

Load the double word D8 into the accumulator and assign the logical state of bit 0 to output O12.
Initial state:
Double word D8 = 3E 81 (hex)
Output O12 = ?

Function	STL	Accumulator content	Operand content
Load the double word D8 into the word accumulator	L D8	3E81	3E81
Check the state of the bit specified in the operand.	BT K+0	1	
Assign the result to output O12.	= O12	1	1

8.13.59 PUSH DATA ONTO THE DATA STACK (PS)

Logic processing with the PS command

Syntax: PS (PUSH)

Operands: M, I, O, T, C

Action:

The PS command enables you to buffer data. To do this, the control loads the addressed operand onto the data stack. Because the data stack has a width of 32 bits, you must write to it with a minimum width of one word. The control copies the operand value into bit 7 of the data stack's current address. The vacant bits of the occupied memory remain undefined or unused. In the event of a stack overflow, the control outputs an error message.

Data stack [bit]															
31 ... 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x ...	x	x	x	x	x	x	L	x	x	x	x	x	x	x

Example:

See PSW command.

Word processing with the PS command

Syntax: PS (PUSH)

Operands: B, W, D, K

Action:

The PS command enables you to buffer data. The control copies the addressed operand value into the current address of the data stack. During the word processing, the control copies two words per PS command onto the data stack and extends the operand—in accordance with the MSB—with the correct algebraic sign. In the event of a stack overflow, the control displays an error message.

Data stack for byte, word, double word and constant [bit]			
31 24	23 16	15 8	7 0
x x x x x x x x	x x x x x x x x	x x x x x x x x	B B B B B B B B
x x x x x x x x	x x x x x x x x	W W W W W W W W	W W W W W W W W
D D D D D D D D	D D D D D D D D	D D D D D D D D	D D D D D D D D
K K K K K K K K	K K K K K K K K	K K K K K K K K	K K K K K K K K

Example:

See PSW command.

8.13.60 PULL DATA FROM THE DATA STACK (PL)

Logic processing with the PL command

Syntax: PL (PULL)

Operands: M, I, O, T, C

Action:

The PL command is the counterpart to the PS command. Data that has been buffered with the PUSH command can be taken from the data stack by using the PULL command. The control copies bit 7 of the data stack's current address into the addressed operand. If the stack is empty, the control displays an error message.

Example:

See PSW command.

Word processing with the PL command

Syntax: PL (PULL)

Operands: B, W, D, K

Action:

The PL command is the counterpart to the PS command. Data that has been buffered with the PUSH command can be taken from the data stack by using the PULL command. During the word processing, the control copies with the PL command two words of the current data stack address into the addressed memory area. If the stack is empty, the control displays an error message.

Example:

See PSW command.

8.13.61 PUSH LOGIC ACCUMULATOR ONTO THE DATA STACK (PSL)

Syntax: PSL (PUSH LOGICACCU)

Operands: None

Action:

The PSL command enables you to buffer the logic accumulator. With the PSL command, the control copies the logic accumulator onto the data stack. Because the data stack has a width of 32 bits, you must write to it with a minimum width of one word. The control copies the operand value into bit 7 of the data stack's current address. The vacant bits of the occupied memory remain undefined or unused. In the event of a stack overflow, the control outputs an error message.

Data stack [bit]															
31 ... 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x ...	x	x	x	x	x	x	L	x	x	x	x	x	x	x

Example:

See PSW command.

8.13.62 PUSH WORD ACCUMULATOR ONTO THE DATA STACK (PSW)

Syntax: PSW (PUSH WORDACCU)

Operands: None

Action:

The PSW command enables you to buffer the word accumulator. With the PSW command, the control copies the word accumulator onto the data stack. The content of the word accumulator (32 bits) occupies two words on the data stack. In the event of a stack overflow, the control displays an error message.

Example:

Since the sequence is the same for all stack operations, this example also applies to the commands PS, PL, PSW, PLL, PLW. The difference between the individual operations lies merely in the transferred data width.

Call Module 15 at a certain place in the program. After returning to the main program, restore the original accumulator content. Accumulator content before the module call: 1A 44 3E 18

Function	STL	Accumulator content	Data stack
Buffer the word accumulator in the data stack	PSW	1A443E18	1A443E18
Call subroutine 15	CM 15		
Restore data stack into word accumulator.	PLW	1A443E18	1A443E18

8.13.63 PULL LOGIC ACCUMULATOR FROM THE DATA STACK (PLL)

Syntax: PLL (PULL LOGICACCU)

Operands: None

Action:

The PLL command is the counterpart to the PSL command. Data that has been buffered with the PUSH command can be restored from the data stack by using the PULL command. The control copies bit 7 of the data stack's current address into the logic accumulator. If the stack is empty, the control displays an error message.

Example:

See PSW command.

8.13.64 PULL WORD ACCUMULATOR FROM THE DATA STACK (PLW)

Syntax: PLW (PULL WORDACCU)

Operands: None

Action:

The PLW command is the counterpart to the PSW command. Data that has been buffered with the PUSH command can be restored from the data stack by using the PULL command. During the word processing, the control copies with the PLW command two words of the current data stack address into the word accumulator. If the stack is empty, the control displays an error message.

Example:

See PSW command.

8.13.65 UNCONDITIONAL JUMP (JP)

Syntax: JP (JUMP)

Operands: Label (LBL)

Action:

After a JP command, the control jumps to the label that you have entered and resumes the program from there. The JP command interrupts a logic sequence.

Example:

See JPT command.

8.13.66 JUMP IF LOGIC ACCUMULATOR = 1 (JPT)

Syntax: JPT (JUMP IF TRUE)

Operands: Label (LBL)

Action:

The JPT command is a conditional jump command. If the logic accumulator = 1, the control resumes the program at the label that you have entered. If the logic accumulator = 0, the control does not jump. The JPT command interrupts a logic sequence.

Example:

This example also applies to the JP and JPF commands.

Depending on the state of the input I5, skip a certain program section.
Initial state:
Input I5 = 1

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I5	1	1
If logic accumulator =1, jump to LBL 10.	JPT 10	1	
Skip the function.	L I3		
Skip the function.	O M500		
Skip the function.	= 020		
Label	LBL 10		
Resume the program run.	L M100	0	0

8.13.67 JUMP IF LOGIC ACCUMULATOR = 0 (JPF)

Syntax: JPT (JUMP IF FALSE)

Operands: Label (LBL)

Action:

The JPF command is a conditional jump command. If the logic accumulator = 0, the control resumes the program at the label that you have entered. If the logic accumulator = 1, the control does not jump. The JPF command interrupts a logic sequence.

Example:

See JPT command.

8.13.68 CALL MODULE (CM)

Syntax: CM (CALL MODULE)

Operands: Label (LBL)

Action:

After a CM command, the control calls the module that begins at the label that you have entered. Modules are independent subroutines that must be ended with the command EM. You can call modules as often as you wish from different places in your program. The CM command interrupts a logic sequence.

Example:

See CMF command.

8.13.69 CALL MODULE IF LOGIC ACCUMULATOR = 1 (CMT)

Syntax: CMT (CALL MODULE IF TRUE)

Operands: Label (LBL)

Action:

The CMT command is a conditional module call. If the logic accumulator = 1, the control calls the module that begins at the label you have entered. If the logic accumulator = 0, the control does not call the module. The CMT command interrupts a logic sequence.

Example:

See CMF command.

8.13.70 CALL MODULE IF LOGIC ACCUMULATOR = 0 (CMF)

Syntax: CMF (CALL MODULE IF FALSE)

Operands: Label (LBL)

Action:

The CMF command is a conditional module call. If the logic accumulator = 0, the control calls the module that begins at the label you have entered. If the logic accumulator = 1, the control does not call the module. The CMF command interrupts a logic sequence.

Example:

This example also applies to the CM and CMT commands.

Depending on the state of the input I5, call the Module 10. Initial state:
Input I5 = 0

Function	STL	Accumulator content	Operand content
Load the operand content into the logic accumulator.	L I5	0	0
If logic accumulator =0, jump to LBL 10.	CMF 10	0	
Resume main program after module execution.	L M100	1	1
	⋮		
End of main program	EM		
Label: Beginning of module.	LBL 10		
Statement in the module.	L I3	0	0
Statement in the module.	O M500	1	1
Statement in the module.	= O20	1	1
End of module, resume the main program with the command L M100.	EM		

8.13.71 END OF MODULE, END OF PROGRAM (EM)

Syntax: EM (END OF MODULE)

Operands: None

Action:

You must end each program or subroutine (module) with the EM command. An EM command at the end or within a module causes a return jump to the module call (CM, CMT, CMF). The control then resumes the program with the statement that follows the module call. The control interprets the EM command as program end. The control can reach the subsequent program instructions only through a jump command.

8.13.72 END OF MODULE IF LOGIC ACCUMULATOR = 1 (EMT)

Syntax: EMT (END OF MODULE IF TRUE)

Operands: None

Action:

An EMT command causes a return jump to the module call (CM, CMT, CMF) only if the logic accumulator = 1.

8.13.73 END OF MODULE IF LOGIC ACCUMULATOR = 0 (EMF)

Syntax: EMF (END OF MODULE IF FALSE)

Operands: None

Action:

An EMF command causes a return jump to the module call (CM, CMT, CMF) only if the logic accumulator = 0.

8.13.74 LABEL (LBL)

Syntax: LBL (LABEL)

Operands: ASCII name; maximum length: 32 characters

Action:

The label defines a program location as an entry point for the JP and CM commands. You can define up to 1000 jump labels per file. The ASCII name of the label may be up to 32 characters long. However, the control evaluates only the first 16 characters.

For importing global labels, see EXTERN statement.

8.14 INDEX Register (X Register)

You can use the index register for:

- Data transfer
- Buffering results
- Indexed addressing of operands

The index register is 32 bits wide.

You can use the X register anywhere in the program. The control does not check whether the current content is valid. Exception: For indexed write-accesses, the control checks whether the permissible address range is exceeded.

Example: = B100[X]

When the permissible address range is exceeded, the control issues the error message **320-0420 PLC: index range incorrect.** After restarting the control, you must not acknowledge the **POWER INTERRUPTED** message. Switch to the PLC editor, where you will be shown the error line.

Note

At the beginning of the PLC cycle, the control sets the index register to 0. Assign the index register a defined value before using it in your program.

The following addresses are valid:

- Mn[X]
- In[X]
- On[X]
- Cn[X]
- Tn[X]Operand number = n+X
- Bn[X]Operand number = n+X
- Wn[X]Operand number = n+2*X
- Dn[X]Operand number = n+4*X
- BTX Content of index register = operand
- BCX Content of index register = operand
- BSX Content of index register = operand
- Sn[X]String number = n+X
- S#Dn[X]Dialog text number = n+X
- S#En[X]Error text number = n+X
- S#An[X]ASCII code +X
- Sn^XSubstring from X-th character of the n-th string

The types K, and K$ cannot be indexed.

Note

If you address S#Dn[X] or S#En[X], the control loads the sequence <SUB>Dnnn or <SUB>Ennn in the string accumulator, where nnn is the modified string number.

Commands for operating the index register

The following commands are available for exchanging data between the word accumulator and index register, or between the stack and index register:

- LX (Load index to accu)Index register – word accumulator
- =X (Store accumulator to index)Word accumulator – index register
- PSX (Push index register)Index register – stack
- PLX (Pull index register)Stack – index register
- INCX(Increment index register)
- DECX(Decrement index register)

8.15 Commands for String Processing

String processing enables you to use the PLC program to generate and manipulate any texts. Use Module 9082 to display these texts in the PLC window of the screen, and delete them with Module 9080. A string accumulator as well as 100 string memories (S0 to S99), in each of which you can save a maximum of 128 characters, are provided in the control for string processing:

String accumulator (characters)	
1	128
x x x x x x x x x x x x x x	x x x x x x x x x x x x x x

String memory (characters)		
	1	128
S0	x x x x x x x x x x x x x x	x x x x x x x x x x x x x x
...	x x x x x x x x x x x x x x	x x x x x x x x x x x x x x

Example

String accumulator (characters)	
1	128
C O O L A N T O N	

String accumulator and string memory are volatile, which means that they are erased by the control when power is switched off. The operand "S" is available for string processing. You can use the operand "S" with different arguments.

Operand declaration

The "S" operands are to be used only for string processing. You can target the following addresses with the various arguments:

- Addressing string memory: After the operand designation, specify the number of the desired memory (S0 to S99).
- Symbolic operands (B/W/D operands) can now be used for indexed access to the string operands "S" or the PLC error and dialog files.
  Examples:
```
#define  offset  D100
...
L S2[offset]
= S8
...
```
  Or
```
S#D900[NP_W1022_Module_error_status]
```
- Address part of a string: Use the address Sn^X (see INDEX Register). The control addresses the substring beginning with the X-th character of the specified string.
  Effective immediately, addressing with symbolic operands is also possible: Sn^symbolic operand (B/W/D operand).
- Immediate string: You can also enter a string directly in the PLC program. The text string, which may contain a maximum of 128 characters, must be indicated by quotation marks.
  Example: **"Coolant 1 on"**
- Texts from the PLC error message file or from the PLC dialog file: By specifying the line number you can read texts from the active error message file or dialog file: **PLC-ERROR: S#Exx xx**: Line number from the PLC error message file (0 to 999)
  PLC-DIALOG: S#Dxx xx: Line number from the PLC dialog file (0 to 999).
  Enter the string #Exx or #Dxx in the argument <arg> of the string command. The control then saves a 5-byte-long string <SUB> E0xx or <SUB> D0xx (<SUB> = ASCII <SUB>) in the accumulator. Instead of this string, the control reads the line xx of the active error message or dialog file on the screen.
- Enter an ASCII character in the string. Define the ASCII character through its code: S#Axxx

Logical comparisons during string processing

Use the following procedure to compare two strings, depending on the argument:

- If you compare string memories or immediate strings, the control checks both strings character by character. After the first character that does not fulfill the condition of comparison, the control resets the logic accumulator. The control does not check the remaining characters. During a comparison, the control always uses the significance of the characters from the ASCII table. This results, for example, in:
  A < B
  AA > A
- If you have entered PLC error messages or PLC dialog texts in the argument, the control compares the position in the error-message file or dialog file (0 to 999), but not the actual text as in an immediate string.

The processing times depend on the length of the strings.

8.16 LOAD String (L)

Syntax: L (LOAD)

Operands: S <arg>

Action:

Load the string accumulator. The string that the control is to load is selected through the argument <arg> after the operand designation. See also "Operand declaration."

Example:

See OVWR command.

8.17 ADD String (+)

Syntax: + (PLUS)

Operands: S <arg>

Action:

Append another string to a string in the string accumulator. The string that the control is to load is selected through the argument <arg> after the operand designation. See also "Operand declaration." The resulting string must not be longer than 128 characters.

Example:

See OVWR command.

8.18 STORE String (=)

Syntax: = (STORE)

Operands: S <arg>

Action:

Assign the content of the string accumulator to the string memory. The memory into which the control is to copy the string is selected through the argument <arg> after the operand designation. Permissible arguments: 0 to 15 (String memory S0 to S99). See also "Operand declaration."

Example:

See OVWR command.

8.19 OVERWRITE String (OVWR)

Syntax: OVWR (OVERWRITE)

Operands: S <arg>

Action:

Save the string from the string accumulator in a string memory. This command differs from the = command in that the control does not transfer the "string end" character along with it. In this way you can overwrite the beginning of a string that is already in the string memory. The memory into which the control is to copy the string is selected through the argument <arg> after the operand designation. Permissible arguments: 0 to 99 (string memory S0 to S99). See also "Operand declaration."

Example:

This example also applies to the string commands L, + and =.

Add a string from the string memory S0 to an immediate string. The result is to overwrite the contents of string memory S1. Initial state:

Immediate string = **HYDRAULICS**
String memory S0 = **OIL**
String memory S1 = **COOLANT MISSING**

String memory (characters)	
	1 128
S0	**O I L**
S1	**C O O L A N T M I S S I N G**
...	...

Function	**STL**	**String accumulator (characters)**
		1 128
Load the immediate string into the string accumulator	L S "HYDRAUL."	**O I L**
Add content of string memory S0 to string accumulator.	+ S0	**H Y D R A U L . O I L**
Overwrite content of string memory S1 with content of string accumulator.	OVWR S1	**H Y D R A U L . O I L**

Final status:

String memory (characters)	
	1 128
S0	**O I L**
S1	**H Y D R A U L . O I L M I S S I N G**
...	...

8.20 EQUAL TO Command for String Processing (==)

Syntax: == (EQUAL)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator and the operand are equal, the condition is true and the control sets the logic accumulator to 1. If they are not equal, the control sets the logic accumulator to 0.

Example:

See command <>.

8.21 LESS THAN Command for String Processing (<)

Syntax: < (LESS THAN)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is less than the operand, the condition is true and the control sets the logic accumulator to 1. If the string accumulator is greater than or equal to the operand, it sets the logic accumulator to 0.

Example:

See command <>.

8.22 GREATER THAN Command for String Processing (>)

Syntax: > (GREATER THAN)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is greater than the operand, the condition is true and the control sets the logic accumulator to 1. If the string accumulator is less than or equal to the operand, it sets the logic accumulator to 0.

Example:

See command <>.

8.23 LESS THAN OR EQUAL TO Command for String Processing (<=)

Syntax: <= (LESS EQUAL)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is less than or equal to the operand, the condition is true and the control sets the logic accumulator to 1. If the string accumulator is greater than the operand, it sets the logic accumulator to 0.

Example:

See command <>.

8.24 GREATER THAN OR EQUAL TO Command for String Processing (>=)

Syntax: >= (GREATER EQUAL)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is greater than or equal to the operand, the condition is true and the control sets the logic accumulator to 1. If the string accumulator is less than the operand, it sets the logic accumulator to 0.

Example:

See command <>.

8.25 NOT EQUAL Command for String Processing (<>)

Syntax: <> (NOT EQUAL)

Operands: S <arg>

Action:

This command sets off a direct transition from string to logical processing. Compare the content of the string accumulator with the string in the argument. If the string accumulator is not equal to the operand, the condition is true and the control sets the logic accumulator to 1. If the string accumulator is equal to the operand, it sets the logic accumulator to 0.

Example:

This example of string processing also applies to the commands =, <, >, <=, >=, <>.

Compare the immediate string with the content of the string memory S0. Depending on the result, call Module 50.
Initial state:
String memory S0 = SPINDLE 2
Immediate string = SPINDLE 1

String memory (characters)		
	1	128
S0	**S P I N D L E 2**	
...	...	

Function	**STL**	**String accu. (characters), or logic accu.**
		1 128
Load the immediate string into the string accumulator	L S "SPINDLE 1"	S P I N D L E 1
Gate the content of string memory S0 with content of string accumulator (=, <, >, >=, ...)	<> S0	S P I N D L E 2
If the condition is fulfilled, set logic accumulator to 1 and call the module.	CMT 50	Logic accumulator = 1

8.26 Modules for String Processing

Module 9070 Copy a number from a string

The control searches a selectable string memory (S0 to S99) for a numerical value. The control copies the first numerical value found as a string into another selectable string memory. The control does not check whether a conflict arises between the source and target string. It may overwrite the source string (even then, however, the function of the module is ensured). The control recognizes unsigned and signed numbers, with and without decimal places. Both the period and comma are permitted as decimal point. The control returns the position (in characters) of the first character after the found number in the string memory to be searched.

Call:

PS	K/B/W/D	<>Address of the string memory to be searched>
PS	K/B/W/D	<>Address of the string memory for the found number>
CM	9070	
PL	B/W/D	<>Offset end of numerical string in the searched string memory>

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Number copied
	1	Error. See NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Invalid address of the source string or target string
	11	No number, no string end, or number string has a length of more than 79 characters

Example

```
L S"X-POS.:123"
= S0
PS K+0
PS K+1
CM 9070
PL W520
```

String memory (characters)						Data stack [bit]
	1	...	10	...	128	
S0	X - P O S . : 1 2 3					
S1	1 2 3					10
...	...					

Module 9071 Find the string length
The control finds the length of the string in a selectable string memory (S0 to S99).

Call:
PS K/B/W/D/S<>String no. or string>
CM 9071
PL B/W/D <>Length of the string>

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	String length found
	1	Error. See NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid immediate strings, address of the source or target string is out of range (S0 to S99), string memory was searched but no string end was found

Module 9072 Copy a byte block into a string
The control copies a byte block from a PLC word memory into a PLC string (S0 to S99). The control does not check whether the byte block consists of valid ASCII characters. The module always copies the entire programmed length of the byte block, regardless of any string-end code (0x00) in the byte block. The control automatically sets a string end code (0x00) after the last copied byte. If there are any ASCII special characters in the copied byte block, the contents of the string may not be displayed in the PLC diagnosis correctly.

Call:
PS K/B/W/D <>Start address of byte block>
PS K/B/W/D <>Length of byte block>
PS K/B/W/D <>PLC string>
CM 9072

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Byte block copied into PLC string (S0 to S99)
	1	Error. See NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Invalid start address of the programmed byte block
	2	Invalid length of the programmed byte block (max. 127 characters)
	4	Invalid sum of the start address and the programmed block length
	11	Invalid target string

Module 9073 Copy a string into a byte block

The control copies a PLC string into the word range of the PLC. The control does not check whether the string consists of valid ASCII characters. The programmed length of the string is always copied, regardless of any end-of-string identifiers (0x00). If there are any ASCII special characters or an end-of-string identifier (0x00) in the copied string, the contents of the string will not be displayed correctly in the PLC diagnostics.

Call:

```
PS    K/B/W/D  <>Target address of byte block>
PS    K/B/W/D  <>Length of byte block>
PS    K/B/W/D  <>PLC string>
CM    9073
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	PLC string copied into byte block
	1	Error. See NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Invalid target address of the programmed byte block
	2	Invalid length of the programmed byte block (up to 127 characters permitted)
	4	Invalid sum of the target address and the programmed block length
	11	Invalid source string

8.27 Submit programs

Submit programs are subroutines that the PLC submits to the NC for processing. This allows you to solve problems that are very processor-intensive, require program loops, or must wait for external results. It is a prerequisite that these programs are not restricted to a definite time frame. Depending on the processor load, the control allocates a certain processing power to a submit program. You start submit programs from the PLC program. They can access all data memory areas (M/B/W/D) as the main program can. To prevent problems, ensure that data processed by the PLC program is clearly separated from data processed by the submit program. You can place up to eight submit programs in a queue (submit queue). Each submit program receives an "identifier" (a number between 1 and 255, assigned by the NC), which the control enters in the word accumulator. With this identifier and the REPLY function you can then interrogate whether the program is in the queue, is being processed, or has already been processed.

The control processes the submit programs in the sequence in which they were entered in the queue. If an error occurs during the execution of the submit programs, the control can set a symbolic PLC operand previously defined.

You can define any names for the symbolic operands displaying errors during the execution of submit programs. The operands are assigned to the respective error events in the configuration file of the PLC compiler (usually **PLCCOMP.CFG**).

The following arithmetical errors can be displayed:

Entry in PLCCOMP.CFG	Description
MULERROR	Overflow during multiplication
DIVERROR	Division by 0
MODERROR	Incorrectly executed modulo

For configuring an arbitrary symbolic marker in the configuration editor, See "Data transfer machine parameters => PLC" on page 1254.

The assignment of the error events to a symbolic marker in the **PLCCOMP.CFG** may look like this, for example:

```
MULERROR = MG_Overflow_during_multiplication
DIVERROR = MG_Division_by_0
MODERROR = MG_MODULO_executed_incorrectly
```

Also, the operands must be integrated in your PLC project:

```
#define /s MG_Overflow_during_multiplication      M
#define /s MG_Division_by_0                       M
#define /s MG_MODULO_executed_incorrectly         M
```

The control lists these markers separately in the submit job. This means that the same markers can be edited simultaneously in the PLC run program without changing the original markers. No exact times can be stated for the commands for managing the submit queue.

8.28 Calling the Submit Program (SUBM)

Syntax: SUBM (SUBMIT)

Operands: Label (LBL)

Action:

Assign an identifier (1 to 255) to a labeled subroutine and put it in the queue. At the same time, the control writes the assigned number in the word accumulator. If programs are already entered in the submit queue, the control does not run the addressed program until the programs before it are finished. A submission to the queue may only take place from a PLC program. A SUBM command in a submit program is not possible.

If there is no room in the queue, or if you program the SUBM command in a submit program (nesting), the control assigns the value "0" to the word accumulator.

Example:

See CAN command.

8.29 Interrogating the Status of a Submit Program (RPLY)

Syntax: RPLY (REPLY)

Operands: B/W

Action:

Interrogate the status of the submit program with the specified identifier. You must have already stored the identifier in a byte or word when you call the submit program. With the RPLY command and the defined memory address (byte or word containing the identifier) the control transfers one of the following processing states to the word accumulator:

- Word accumulator = 0: Program complete/not in the queue
- Word accumulator = 1: Program running
- Word accumulator = 2: Program in the queue

Example:

See CAN command.

8.30 Canceling a Submit Program (CAN)

Syntax: CAN (CANCEL)

Operands: B/W

Action:

Cancel a submit program with the specified identifier during processing, or remove it from the queue. You must have already stored the identifier in a byte or word when you call the submit program. After you have canceled the program, the control immediately starts the next submit program from the queue. The following PLC modules cannot be canceled at just any location with CANCEL:

- PLC module for access to screen (908X)
- PLC module for reading NC files (909X)

For these modules, you must check with the RPLY command whether the CAN command may be executed.

Example:

This example also applies to the SUBM and RPLY commands.

Depending on input I10, submit the subroutine with the label LBL 300 to the NC for processing. In addition, check the processing status of the subroutine in the main program with the RPLY command, and cancel it with the CAN command, depending on input I11.

Function	**STL**
Load the content of input I10 into the logic accumulator	L I10
If logic accumulator =0, jump to LBL 100.	JPF 100
Interrogate the status of the submit program and load it into the word accumulator.	RPLY B128
If the word accumulator is not equal to 0, i.e., the submit program has already been transferred to the NC for processing, set the logic accumulator to 1.	<> K+0
If logic accumulator =1, jump to LBL 100.	JPT 100
Call submit program 300.	SUBM 300
Save the identifier of the submit program in byte 128.	= B128
Label	LBL 100
Load the state of input I11 into the logic accumulator	L I11
If logic accumulator = 0, jump to LBL 110 (skip the program cancellation).	JPF 110
Cancel the submit program.	CAN B128
Label	LBL 110
	⋮
End of main program	EM
Label: Beginning of the submit program	LBL 300
	⋮
End of the submit program	EM

Always insert submit programs, like any module, at the end of the main program. In this case, the content of the submit program could be a display in the PLC window that is realizable through permanently assigned PLC modules.

8.31 Cooperative Multitasking

You can run several processes in the PLC with cooperative multitasking. Unlike genuine multitasking, with cooperative multitasking information and tasks are exchanged only at places that you define. Cooperative multitasking permits up to 16 parallel PLC processes and the submit queue. In a program that you have started with SUBM, you can use commands for changing tasks and controlling events (Module 926x). You should additionally insert a task change between the individual jobs in the submit queue, so that the control can execute parallel processes by the end of a job at the latest. The cyclic PLC main program does not participate in cooperative multitasking, but interrupts a submit job and the parallel processes at whatever their current stage is.

8.31.1 Starting a parallel process (SPAWN)

Syntax: SPAWN <label>

Operands: D

Action:

In the specified double word, the control returns the identifier, see page 1323. The control returns –1 if no process could be started. You can call the spawn command only in a submit job or in another spawn process (maximum of eight parallel processes are permitted). If such a process ends with EM, the control removes it from the memory, and the memory space is again available.

8.31.2 Control of events

The parallel processes can make events available to one another. This saves processing time otherwise spent in the constant interrogating of operating states by the individual processes. A special feature of event control is the waiting period, during which the process can "sleep" for a programmed time. With this function you can repeat program sections in a slow time grid, for example for display or monitoring functions.

Process monitor

In the PLC programming mode you can use the MONITOR and PROCESS MONITOR soft keys to open a status screen in which the control displays all parallel processes, including the process for the submit queue. In a time interval, which can be set with the "+" and "–" soft keys, the iTNC displays:

- The name of the process (**TASKNAME**)
- The current status of the process (**STATE**)
  - Executable (**SCHED**)
  - Running (**RUN+**)
  - Waiting for event (**EVWAIT**)
  - Waiting for time period (**TMWAIT**)
  - AND-gating of the bits in the event mask (**AND**)
  - OR-gating of the bits in the event mask (**OR**)
- The event mask (**EVMASK**)
- The PLC module letting the process wait (**MOD**)
- How often the process has changed contexts in the last time interval (**SCHED**).
- How much CPU time the processor has used from the defined time interval (**CPU(ms)**). The control also shows the distribution of CPU time in a bar chart (**RATIO**).

Module 9260 Receive events and wait for events

Call the module only in a submit job or spawn job. The module enables a spawn job or submit job to interrogate or wait for the occurrence of one or more events. At the same time, the module triggers a change in context.

If you transfer the value zero for the event mask, the control returns all set events without deleting them. Otherwise, in a call with a waiting period, the control returns all the requested events and deletes them. For a call without a waiting period, the control returns and deletes the events only if the condition is met.

If the events are OR-gated, the control returns and deletes only the set events. You can specify the events to be deleted by calling without a waiting period and with an OR gate.

Event bits 16 to 31 are reserved for the operating system:

- Bit 16: BREAK, cancels a function. Setting and reading is permitted. If you transmit this event, the control cancels access to interfaces and the network!
- Bit 17: Reserved, do not use
- Bit 18: Reserved, do not use
- Bit 19: QUIT, acknowledgment of a request. Use this bit only in the immediate context of a request.
- Bit 20 to bit 31: Reserved, do not use

Call:

```
PS      B/W/D/K   <>Wait>
                  0 = Do not wait
                  –1 = Wait
PS      B/W/D/K   <>AND/OR>
                  0 = OR-gated, otherwise AND-gated
PS      B/W/D/K   <>Event mask>
                  0 = Available events
CM      9260
PS      B/W/D/K   <>Events>
                  Read events
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Event has been read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Incorrect transfer value for <Wait> parameter
	20	Module was not called in a spawn or submit job

Module 9261 Send events

This module is used to send events to a spawn or submit job and then interrogate them with Module 9260. You can call the module in the cyclic program section, in submit jobs and in spawn processes. The control addresses the receiver through the identifier that the spawn command has returned. The submit queue is addressed through the identifier $80000000 (not through the identifier returned by the SUBM command!). The control always assigns the events that you send to the submit queue to the job that is running at the time of arrival. If they are not read by this job, they remain for the next one. If you wish the receiver process to start immediately, after Module 9261 you must also call Module 9262 to enable a change of context.

Event bits 16 to 31 are reserved for the operating system (see Module 9260).

Call:

PS	D/K	<>Identifier> Identifier from the spawn command of the receiver K$80000000 = submit queue
PS	B/W/D/K	<>Events> Events to be triggered, bit encoded
CM	9261	

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Event has been sent
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	30	Incorrect identifier

Module 9262 Context change between spawn processes
You can call Module 9262 only in a submit job or spawn job. The module switches the context to another PLC process or submit queue if such a process exists and is not waiting for an event or for the expiration of a dwell time.

Call:
CM 9262

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Context was changed
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	30	Module was not called in a spawn or submit job

Module 9263 Interrupt a spawn process for a defined time
You can call Module 9263 only in a submit job or spawn job. The module interrupts the calling process for at least the specified time. If other processes or the submit queue are ready to run, the control changes the context to one of these processes. The waiting period is interpreted as an unsigned number, so that negative values result in very long waiting periods.

Call:
PS B/W/D/K <>Waiting period in ms>
CM 9263

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Delay is active
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	30	Module was not called in a spawn or submit job

Module 9264 Wait for a condition

The module enables a spawn process or submit job to wait for a specific condition. This module call replaces the procedure of the event list used by earlier HEIDENHAIN contouring controls.

You can call Module 9264 only in a submit job or spawn job.

The condition is a logical expression in accordance with the C language convention with the following syntax:

<memory operand>[<comparison operator><value>]

If at least one bit in the event mask is set, the process is continued immediately, and the event signalizes that the condition has been fulfilled. Module 9260 must wait for this. If the event mask equals zero, the process is paused until the condition is fulfilled. Only then is the module call ended.

The following relational operators may be used as conditions:

- == : Equal to
- <> : Not equal to
- < : Less than
- > : Greater than
- <= : Less than or equal to
- >= : Greater than or equal

If you do not specify a relational operator, the condition is fulfilled if the content of the memory operand is not equal to 0.

Constraints:

- If the value 0 is transferred for the event mask, the job waits for the condition to be fulfilled. There is always a change in context when another spawn process or the PLC queue is executable. If there are several other executable PLC processes, the CPU is allocated to the process that has been waiting for allocation the longest. If there is no other executable PLC process and the wait condition is not fulfilled, the NC software is allocated the CPU time that is not required.
- Module 9260 can be used to check if the condition has been fulfilled or to wait for the condition to be fulfilled if a value that is not equal to 0 is transferred for the event mask. Calling the Module 9264 does not cause a change in context in this case.

Call:

PS B/W/D/K/S<>Condition>
e.g. "ML_TestMemory[0] = 1"
Syntax corresponds to the NC syntax from **FN20: WAIT FOR:** (see the User's Manual).
Following conditions are permissible:
== : Equal to
<> : Not equal to
< : Less than
> : Greater than
<= : Less than or equal to
>= : Greater than or equal

PS B/W/D/K <>Event bit mask>
0 : Process is paused until condition is fulfilled
Bit#1 to bit#15: Reserved, do not use
Bit#16: BREAK – causes cancellation of a function.
Can be set and read. Access to interfaces and networks is aborted by sending this event in the case of waiting states!
Bit#17: Reserved, do not use
Bit#18: Reserved, do not use
Bit#19: QUIT – acknowledgment of a request.
Quitting a request may only be used in direct context with a request. Therefore: Delete QUIT event, set request to another process, wait for QUIT event.
Bit#20..#31: Reserved, do not use

CM 9264

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Condition is waited for
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Syntax error in the condition
	3	Address for string outside value range
	20	Module was not called in a spawn or submit job

8.32 Constants Field (KF)

You can use the constants-field data type to access one of several constants, defined in tabular form, depending on the value of the index register X. You address it with KF <Name>[X], where <Name> is a label indicating the beginning of the constants field. Constants fields must be introduced with the label KFIELD <Name>. This is followed by any quantity (other than zero) of constants and the end label ENDK. Constants fields can only be programmed where the program has previously been concluded with an EM or JP statement. The name of constants fields corresponds to the rules for naming labels.

Addressing

Types of addresses:

- L KF <Name> [X], with X ≥ 0:
  The control transfers the value of the constant defined by X in the constants field <Name>.
- L KF <Name> [X], with X = –1:
  The control transfers the length of the constants field <Name>.
- L KF <Name>:
  The control transfers the absolute address of the constants field <Name>. This is only worthwhile in conjunction with modules (e.g. Module 9200). You can also use this addressing in a constants field.

Example:

Function	STL
Access value field with X = [0 to 3].	L KF VAL_FIELD [X]
Assign one of the constants to word W0.	= W0
End of main program	EM
Define the constants field. Constant to be loaded with X = 0 Constant to be loaded with X = 3 End of the constants field.	KFIELD VAL_FIELD K+10 K+1 K$ABC K–100000 ENDK

The control checks the access to constants fields in the same way as the write access for indexed operands. X can assume only positive values from 0 to <Length of constants field –1>.

8.33 Program Structures

To design an easily understandable program, divide it into program sequences. Use labels (LBL) as well as conditional and unconditional jumps. If you use structured statements, the compiler creates the labels and jump commands. Remember that using these labels and jump commands reduces the number of available labels accordingly. You can nest structured statements in up to 16 levels. It is not possible to share levels.

Example:

Correct program structure	Incorrect program structure
IFT	**IFT**
⋮	⋮
WHILEF	**WHILEF**
⋮	⋮
ENDW	**ENDI**
⋮	⋮
ENDI	**ENDW**

The statements IFT, IFF, WHILET, WHILEF, ENDW, UNTILT and UNTILF require a valid gating result in the logic accumulator. They conclude the sequence of gating operations. The statements ELSE, ENDI and REPEAT require that all previous operations sequences have been concluded.

8.33.1 IF ... ELSE ... ENDI structure

The IF ... ELSE ... ENDI structure permits the alternative processing of two program branches depending on the value in the logic accumulator. The ELSE branch is not mandatory. The following commands are available:

- IFT (IF LOGIC ACCU TRUE):
  Following code only if logic accumulator = 1
- IFF (IF LOGIC ACCU FALSE):
  Following code only if logic accumulator = 0
- ELSE (ELSE):
  Following code only if IF is not fulfilled
- ENDI (END OF IF STRUCTURE):
  End of the IF structure

Function	STL
Load input I0 into the logic accumulator	L I0
Run the following code if logic accumulator = 1	IFT
Program code for I0 = 1	⋮
Run the following code if logic accumulator = 0; command can be omitted	ELSE
Program code for I0 = 0, can be omitted	⋮
End of the conditional processing	ENDI

8.33.2 REPEAT ... UNTIL structure

The REPEAT ... UNTIL structure repeats a program sequence until a condition is fulfilled. Under no circumstances can you wait with this structure in the cyclic PLC program for the occurrence of an external event! The following commands are available:

- REPEAT (REPEAT):
  Repeat the program sequence from here.
- UNTILT (UNTIL TRUE):
  Repeat the sequence until the logic accumulator = 1.
- UNTILF (UNTIL FALSE):
  Repeat the sequence until the logic accumulator = 0.

The control runs a REPEAT ... UNTIL loop at least once!

Function	STL
Assign the content of the logic accumulator to marker 100; conclusion of the previous commands	= M100
Repeat the following program code	REPEAT
Program code to be run	⋮
Load the index register	L X
Check the index register	>= K100
Repeat until X >= 100	UNTILT

8.33.3 WHILE ... ENDW structure

The WHILE ... ENDW structure repeats a program sequence if a condition is fulfilled. Under no circumstances can you wait with this structure in the cyclic PLC program for the occurrence of an external event! The following commands are available:

- WHILET (WHILE TRUE):
  Run the sequence if logic accumulator = 1.
- WHILEF (WHILE FALSE):
  Run the sequence if logic accumulator = 0.
- ENDW (END WHILE):
  End of the program sequence, return to the beginning

The control runs a WHILE ... ENDW loop only if at the beginning the WHILE condition is fulfilled. Before the ENDW statement you must reproduce the condition for execution. For the WHILE ... ENDW structure the control generates two internal labels. The condition can also be produced in a way different from before the WHILE statement!

Function	**STL**
	⋮
Load marker 100 into the logic accumulator; create condition for 1st WHILE scan.	L M100
Run the following code if logic accumulator = 1	WHILET
Program code for logic accumulator = 1	⋮
Produce the condition of repeated execution: Load marker 101 in the logic accumulator and gate the content of marker M102 with AND.	L M101 A M102
Jump back to the WHILE request	ENDW

8.34 CASE branch

Indexed module call (CASE)

Syntax: CASE (CASE OF)

Operands: B/W

Action:

Selects a certain subroutine from a list of module calls (CM). These CM commands must follow the CASE statement immediately and are numbered internally in ascending order from 0 to a maximum of 127. The content of the operand (B, W) addresses the desired module. Subsequent entries in the jump table (CM) must have addresses at least four bytes higher than the previous entry.

Example:

See ENDC command.

End of indexed module call (ENDC)

Syntax: ENDC (ENDCASE)

Operands: None

Action:

Use the ENDC command in connection with the CASE command. It must come immediately after the list of CM commands.

Example:

Function	STL
Case command and operand; the internal address of the desired module must be saved in the operand	CASE B150
Call module if operand content = 0 Internal addressing from 0 to max. 127	CM 100
Call module if operand content = 1	CM 200
Call module if operand content = 2	CM 201
Call module if operand content = 3	CM 202
Call module if operand content = 4	CM 203
Call module if operand content = 5	CM 204
Call module if operand content = 6	CM 300
End of the CASE statement	ENDC

8.35 Linking Files

You can store the source code of the PLC program in several files. To manage these files, use the following commands:

- USES
- GLOBAL
- EXTERN

These statements must be located at the beginning of your PLC program—i.e., before the first PLC command. With the USES statement you link another file into the program. The GLOBAL statement supplies a label from its own file for an entry that can be used by all other files. The EXTERN statement provides a label that is defined in another file and is identified there with GLOBAL. You can then call this label from the active file. You can dramatically improve the transparency of your program by dividing your source code by function into individual groups and then save these groups in individual files. The number of labels is not limited. You can link up to 256 files to one program. The total size is only limited by the available memory. If the memory is exceeded the error message **`System memory overflow`** appears. With multiple files, the main program in the directory must have the "M" status flag. This can be done in RAM by using the PLC program function "COMPILE" once and choosing the main program in the file window.

8.36 USES Statement (USES)

Syntax: USES <file name>

Operands: None

Action:

You can use the USES statement in the main program to link other files. Files that are linked with USES can themselves also use the statement to link further files. It is also permissible to use the USES statement to link a single file to several other files. The code for this file is generated only once. The USES statement requires a file name as an argument. The USES statement only links a file; it does not run the file's program code. The USES statement cannot be compared with a CM statement. The linked files must therefore contain individual modules that you can then call with the CM statement.

Example:

USES PLCMOD1
USES EPRUPG
USES RAMPLC

Linking of files:

Function	STL
Main program	PLCMAIN.PLC
Link the file for spindle control.	USES SPINDLE.PLC
Link the file for tool change.	USES TCHANGE.PLC
Program code	⋮

Function	STL
File for spindle control	SPINDLE.PLC
Integrate file with general subroutines.	USES PLCUPG.PLC
Program code	⋮

Function	STL
File for tool change	TCHANGE.PLC
Integrate file with general subroutines.	USES PLCUPG.PLC
Program code	⋮

Function	STL
File with general subroutines	PLCUPG.PLC
Program code	⋮

8.37 GLOBAL Statement (GLOBAL)

Syntax: GLOBAL <Label, declaration beyond the file boundary>

Operands: None

Action:

There is no limit to the number of labels in each file linked with USES. To enable a module that was defined in a file to be called from another file, you must declare the module to be global. This is done by entering the GLOBAL statement at the beginning of the file. You can set labels globally only if they are defined with LBL (and not with KFIELD!) later on in the program.

The main program must not contain any GLOBAL definitions. A single label cannot be declared global by more than one module. However, a name that is declared global in file A can be used again locally in file B. The number of labels is not limited.

8.38 EXTERN Statement (EXTERN)

Syntax: EXTERN <Label, a module from another file can now be called with a CM command>

Operands: None

Action:

To enable a label in one file to access modules that other files have declared as GLOBAL, you must declare the label with EXTERN. You must write the EXTERN statement at the beginning of the file. In the program code you can then jump to this label with the commands CM, CMT and CMF.

The following functions are not permitted with external labels:

- JP, JPF, JPT
- Access to a constants field
- Linking a CM statement in a CASE branch

The name of the external label cannot be used again as a local label in the same file.

8.39 PLC Modules

A number of PLC modules are available for PLC functions that are very difficult or even impossible to perform with PLC commands alone. You will find descriptions of these modules under the corresponding functions. (See "Overview of Modules" on page373.)

If the control processes a module incorrectly, it sets the marker **NN_GenApiModuleError**. You can evaluate this marker for displaying an error message.

8.39.1 Markers, bytes, words, and double words

Module 9000/9001 Copy in the marker or word range
Modules 9000 (markers) and 9001 (byte/word/double) copy a block with a certain number of markers or bytes, beginning with the start address, to the specified target address. For Module 9001 the length should always be defined in bytes.

Constraints: The control copies sequentially, beginning with the first memory cell. Therefore the function is not ensured if the source block and the target block overlap and the source block begins at a lower address than the target block. In this case the control overwrites the overlapping part of the source block before the copying process.

Call:

```
PS    B/W/D/K  <>Number of the 1st marker in source block>
PS    B/W/D/K  <>Number of the 1st marker in target block>
PS    B/W/D/K  <>Length of block in markers>
CM    9000
PS    B/W/D/K  <>Number of the 1st word in source block>
PS    B/W/D/K  <>Number of the 1st word in target block>
PS    B/W/D/K  <>Length of block in markers>
CM    9001
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Markers, bytes, words, or double words were copied
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Operand address invalid
	2	Address too high or block too long
	4	Programmed source or target block too long

Module 9010/9011/9012 Read in the word range

From the specified location in the word memory the control reads a byte, word or double word and returns it as an output quantity to the stack. Indexed reading is possible by specifying a variable as designation of the memory location.

Call:

```
PS    B/W/D/K   <>Address of the byte to be read>
CM    9010      ; READ BYTE
PL    B         <>Target address for byte that was read>

PS    B/W/D/K   <>Address of the word to be read>
CM    9011      ; READ WORD
PL    B         <>Target address for word that was read>

PS    B/W/D/K   <>Address of the double word to be read>
CM    9012      ; READ DOUBLE WORD
PL    B         <>Target address for double word that was read>
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Byte was read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	3	Invalid address was programmed
	5	Module 9011: Specified address is not a word address Module 9012: Specified address is not a double word address

Example of Module 9010

Initial state:

```
Byte  B10   = 35 (address)
Byte  B35   = 80 (byte to be read)
Byte  B100  = ?
```

Function	STL	Accumulator content (dec)	Data stack (dec)
Save the address (B10) of the byte to be read from the word accumulator in the data stack	PS B10	35	35
Read byte B35 and save in the data stack	CM 9010		80
Save data stack in byte B100.	PL B100	80	80

Module 9020/9021/9022 Write in the word range

The control writes the specified byte, word or double word to the defined location in the word memory. Indexed writing is possible by specifying a variable as designation of the memory location.

Call:

```
PS    B/W/D/K  <>Address of the byte to be written>
PS    B/W/D/K  <>Byte to be written>
CM    9020     ; WRITE BYTE TO ADDRESS

PS    B/W/D/K  <>Address of the word to be written>
PS    B/W/D/K  <>Word to be written>
CM    9021     ; WRITE WORD TO ADDRESS

PS    B/W/D/K  <>Address of the double word to be written>
PS    B/W/D/K  <>Double word to be written>
CM    9022     ; WRITE DOUBLE WORD TO ADDRESS
```

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Byte was written
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	3	Invalid address was programmed
	5	Module 9021: Specified address is not a word address Module 9022: Specified address is not a double word address

Example of Module 9020

Initial state:

```
Byte  B10   = 35 (address)
Byte  B100  = 120 (byte to be written)
Byte  B35   = ?
```

Function	STL	Accumulator content (dec)	Data stack (dec)
Save the address (B10) of the byte to be written from the word accumulator in the data stack	PS B10	35	35
Save byte B100 from the word accumulator in the data stack	PS B100	120	120
Write data stack to byte B35	CM 9020	120	

Module 9050 Conversion from binary →ASCII

Module 9050 converts a binary numerical value consisting of a mantissa and exponent to base 10 into an ASCII-coded decimal number and saves it as a string in the specified address. The exponent refers to the least significant place of the number. The control detects a negative number when the mantissa corresponds to a negative number in the notation as a two's complement. The control sets an algebraic sign only before negative numbers. The control does not convert trailing zeros after the decimal point or leading zeros before the decimal point. The control writes the string left-aligned in the string address that you specify.

Constraints:
The decimal character is defined by Machine Parameter MP7280 as a comma (MP7280 = 0) or a period (MP7280 = 1).

Call:

PS	B/W/D/K	<>Mantissa of the number to be converted>
PS	B/W/D/K	<>Exponent to base 10 of the value>
PS	B/W/D/K	<>String address in which the control saves the ASCII-coded decimal number>
CM	9050	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Number was converted
	1	For error code see NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid string address or invalid exponent

Module 9051 Conversion from binary →ASCII

Module 9051 converts a binary numerical value into an ASCII-coded decimal number in the specified format and saves it as a string in the specified address. The number is interpreted as a two's complement. For algebraically unsigned notation, the control converts the absolute amount of the number without putting a sign before the string. For algebraically signed notation, the control sets an algebraic sign ("+" or "–") in front of the string in any event. For notation in inches, the number is divided by 25.4 before conversion. If the number has more decimal places than the total that you have specified for the number of places before and after the decimal point, then the control omits the most highly significant decimal places. In right-aligned notation leading zeros before the decimal point are replaced by blanks; in left-aligned notation they are suppressed. Trailing zeroes after the decimal point are always converted.

Constraints:
The decimal character is defined by Machine Parameter MP7280 as a comma (MP7280 = 0) or a period (MP7280 = 1).

Call:

PS B/W/D/K <>Numerical value to be converted>
PS B/W/D/K <>Display modes, bit-encoded>

Bit #1/#0: Format
00: Sign and number left-aligned
01: Sign left-aligned, number right-aligned
10: Sign and number right-aligned
11: Not permitted
Bit #2: Display converted to INCH
Bit #3: Display with sign

PS B/W/D/K <>Number of places after the decimal point>
PS B/W/D/K <>Number of places before the decimal point>
PS B/W/D/K <>String address in which the control saves the ASCII-coded decimal number>
CM 9051

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Number was converted
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Invalid string address, invalid display mode or invalid number of places before or after the decimal point

Module 9052 Conversion of decimal string →Decimal number with an exponent

Module 9052 converts an ASCII-coded decimal number (possibly with decimal places) into a signed number and an exponent to the base of 10. You must assign the ASCII-coded decimal number to one of the string memories. If the number has no algebraic sign, the control interprets it as a positive number and accepts both a point and a comma as decimal character. If the full extent of the mantissa cannot be represented in a double word, then the last places are omitted and the exponent is corrected accordingly. If possible, the control adjusts the exponent so that it corresponds to the ASCII notation.

Call:

PS	B/W/D/K	<>String address in which the ASCII-coded decimal number is saved>
CM	9052	
PL	B/W/D	<>Numerical value>
PL	B/W/D	<>Exponent to the base of 10 of the value>

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Number was converted
	1	For error code see NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid string address or string contains none or too many characters

Module 9053 Conversion from binary → ASCII/hexadecimal

Module 9053 converts blocks of binary values from the word-marker range into a string of ASCII-coded hexadecimal numbers. The control reads the specified number of bytes from the word address that you have specified and converts it to a hexadecimally coded ASCII string. Each byte produces two characters in the string memory.

Call:

PS	B/W/D/K	<>Word address from which the binary values are saved>
PS	B/W/D/K	<>String address in which the control saves the hexadecimal numbers>
PS	B/W/D/K	<>Number of data bytes>
CM	9053	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Number was converted
	1	For error code see NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Too many data bytes
	2	Invalid string address
	4	Invalid word address

Module 9054 Conversion from ASCII/hexadecimal →binary

Module 9054 converts strings of ASCII-coded hexadecimal values into a block of binary values in the word-marker range. The string in the specified string memory is interpreted as a sequence of ASCII-coded hexadecimal numbers and converted into a block of corresponding binary bytes. Two ASCII characters produce one binary byte. The control saves the binary block beginning at the specified address in the word-marker range.

Call:

PS B/W/D/K <>String address in which the hexadecimal value is saved>
PS B/W/D/K <>Word address from which the control saves the binary values>
CM 9054

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Number was converted
	1	For error code see NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	2	Invalid string address
	11	Invalid word address
		Odd number of characters in the string or a character that cannot be interpreted as hexadecimal

Example

Initial state:
SO = "63"
BO = 99

Function	STL	String accu. (characters), data stack [bits]
Push string address S0 onto the data stack	PS K+0	63
Push word address B0 onto the data stack	PS B0	99
Conversion of the two ASCII characters 6 and 3 to the binary number 99	CM 9054	01100011

9 Data Interfaces

9.1 Introduction

In addition to their Central Processing Unit (CPU), computer systems usually include various peripheral devices.

A CPU is, for example:

- PC
- Control

Peripheral devices include:

- Printers
- Scanners
- External storage devices, such as hard disks, floppy-disk drives or USB memory sticks.
- Other computer systems

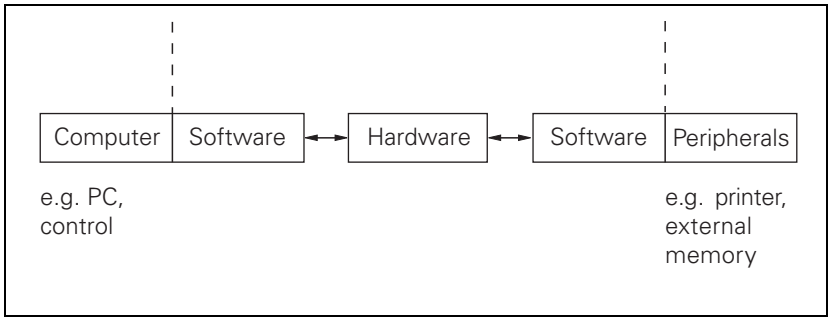
A data interface makes it possible for the CPU and its peripheral devices to communicate.

The interfaces, which consist of physical links between the computer system and the peripherals, need a transmission line and appropriate software in order to transfer data between the individual units.

Standard interfaces include:

- Ethernet
- USB 1.1
- RS-232-C/V.24

The relationship between hardware and software, which fully defines an interface, is illustrated by the following diagram:



The hardware in the diagram covers all the physical components, such as

- Circuit design
- Pin layout
- Electrical characteristics

The software is the operating software, which includes, for example, the drivers for the output modules.

9.2 The Ethernet Interface

You can connect the control with your plant's intranet or use a transposed cable to connect directly with a PC. The data transfer rate is dependent on the amount of traffic at the time on the net. For information on the pin layout, see Chapter 3, "Mounting and Electrical Installation."

X26: Ethernet interface RJ45 port (10BaseT)

Maximum cable length:
Unshielded: 100 m
Shielded: 400 m

Network topology: Star configuration

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

The control requires an NFS server (Network File System) or a Windows PC (SMB = Server Message Block) as the remote station. It must operate according to the TCP/IP protocol principle.

OSI 7-layer model		**Control**
7	Application layer	NFS, SMB
6	Presentation layer	
5	Communications layer	
4	Transport layer	TCP protocol
3	Network layer	IP protocol
2	Data link layer	Ethernet card
1	Physical layer	

Before networking, the TNC must be properly configured. Please discuss the required settings with your network supervisor.

9.3 HSCI interface

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

The following features characterize the HSCI connection:

- Based on standard 100BaseT Ethernet hardware
- Telegrams of the HSCI connection are not compatible with the Ethernet
- Line structure
- No collisions
- Data-transfer cycle (HSCI cycle): 3 ms
- Jitter less than 1 ns
- Only one master in the system (MC), all other devices are HSCI slaves
- Slave-to-slave communication possible
- HSCI master must initiate every data transfer
- Realization with FPGAs

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

The following applies to the assignment of the bus address:

- The bus address is the result of the device's position in the bus
- The master (MC) always has the bus address 0
- The bus addresses of the slaves result from their position in the bus:
  - First device after the master (MC): Bus address 1
  - Second device after the master (MC): Bus address 2
  - etc.
- The bus address 0xFF is used as multicast address

The following applies to the assignment of the device-type address:

Device-type address:	Device type
000000	MC
000001	Inputs/outputs of the MC
000010	CC
000011	Inputs/outputs of the CC (STO ports)
000100	PL 6xxx
000101	PL 6xxx interface for Profibus
000110	PL 6xxx S and MB xxx S from channel A
000111	PL 6xxx S and MB xxx S from channel B
001000	MB xxx
111111	Multicast to all device types
001011 ... 111110	Reserved

9.4 The USB Interface of the Control (USB 1.1)

The USB interface is a standard serial interface.
(USB = Universal Serial Bus)

USB 1.1 provides a maximum data transfer rate of 12 Mbps.

Various USB block devices, such as keyboard, mouse, external hard disks, and USB memory sticks, can be connected to the control via the USB interface (X141, X142).

Note

If USB components require more than 0.5 A, a separate power supply becomes necessary for these components. One possibility is the USB hub from HEIDENHAIN (368 735-01).

The USB interface features the "hot-plug capability." This means that you can connect USB devices to the USB interface and remove them, without having to shut down and then restart the control.

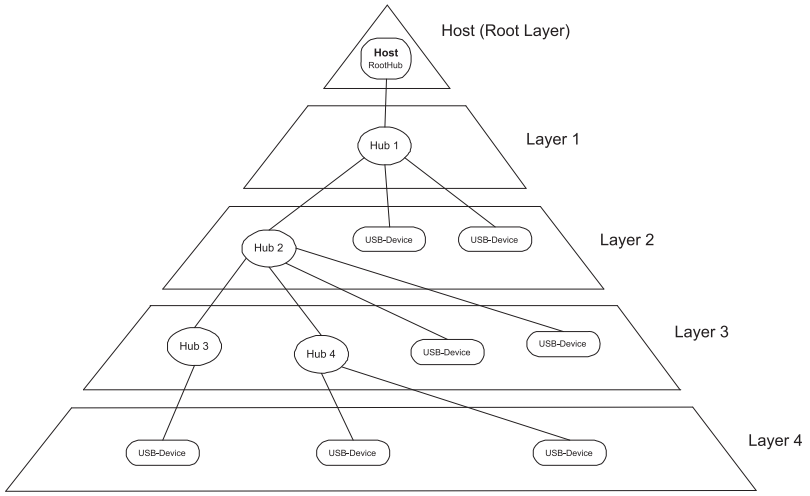
Transmission distance without hub: Up to 6 meters

Note

For greater transmission distances, you must use a USB hub after every six meters in order to amplify the signal. You can use more than one hub for one transmission distance. USB cables with a length of up to 30 meters (with 5 integrated USB hubs) are available from HEIDENHAIN under ID 624 775-xx.

Structure

The USB interface connects the USB peripheral devices with the USB host. The topology of a USB connection may consist of several levels arranged in a star configuration. Every level consists of a USB hub to which other USB devices or hubs are connected in a star configuration. A maximum of 127 USB devices can be connected to a USB host in this way.



Topology of the USB interface

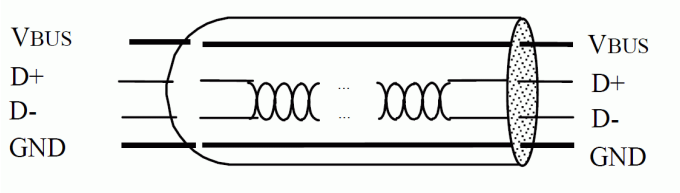
Functionality and signal designations

USB uses packet-based communication over two differential data lines. This reduces radiation and increases transmission reliability. USB provides significantly higher data transfer rates than other external interfaces (parallel / Centronics, serial / RS-232, RS-422):

- USB full speed of up to 12 Mbps
- USB low speed of up to 1.5 Mbps

Conventional interfaces, such as the RS-232, are more suitable for time-critical applications because they are not based on packets, which reduce the transfer rate (in case of packets with only a few bytes) or delay transmission (when collecting bytes for filling a packet).

Only four wires are required in a USB cable. Two for a power supply of 5 V (with max. 500 mA / 2.5 W) and two for data transmission:



USB devices on the control

The USB interface of the control allows for convenient and fast exchange of data. You can connect USB block devices, such as memory sticks, hard disks, CD-ROM drives, to your control via the USB interface without having to reboot the system. The data media can be accessed immediately after connection.

The control supports the following USB block devices:

- Floppy disk drives with the FAT/VFAT file system
- Memory sticks with the FAT/VFAT file system
- Hard disks with the FAT/VFAT file system
- CD-ROM drives

The control does not support USB devices with other file systems (e.g. NTFS). If you try to connect such devices, the control will issue an error message.

Note

It should basically be possible to connect all USB block devices with the above-mentioned file system to the control. If you nevertheless encounter problems, please contact HEIDENHAIN.

USB devices tested by HEIDENHAIN

A variety of USB storage media from different manufacturers is available on the market. It may happen that a USB device is not identified correctly by the control. The USB devices listed in the table below were tested by HEIDENHAIN for proper functioning in conjunction with the control (numerous other USB devices are supported by the control, but you should test them for proper functioning on the control before using them):

USB device	Manufacturer	Model designation	VendorID	ProductID	Revision
Floppy disk drive	TEAC	TEAC FD-05PUW	0644	0000	0.00
Floppy disk drive	TEAC	TEAC FD-05PUB	0644	0000	0.00
CD-ROM drive	TEAC	USB CD-ROM 210 PU	0644	1000	1.33
CD-ROM drive	FREECOM	USB2-IDE Controller	07ab	fc02	11.10
Hard disk	UNKNOWN	USB to IDE Converter	05e3	0702	0.02
Memory stick	TrekStor	USB MiniStick	0c76	0007	1.00
Memory stick	QDI	UNKNOWN	0c76	0005	1.00
Memory stick	Transcend	TS512MJFLASH	058f	9380	1.00
Memory stick	Transcend	Flash Disk	0ea0	2168	2.00
Memory stick	Generic	Mass Storage Device	058f	9384	1.05

9.5 The Serial Interface of the Control

9.5.1 RS-232-C/V.24 interface

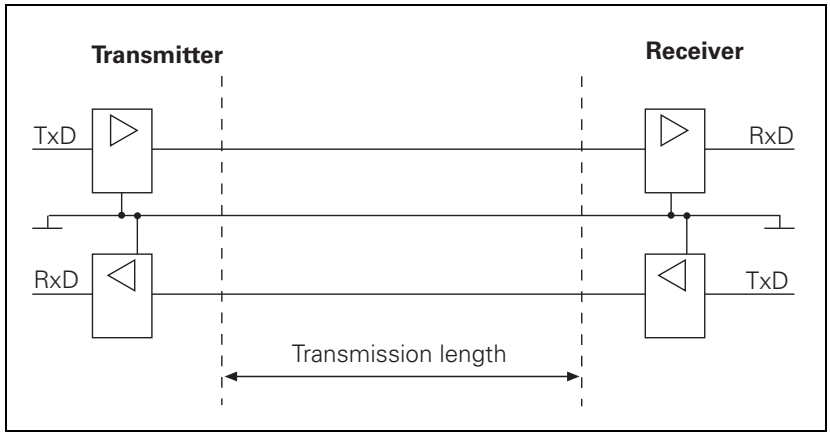
With RS-232-C/V.24, data transfer is executed asynchronously, with a start bit before each character and one or two stop bits after each character.

Transmission distance: up to 20 m

Hardware

The physical connection between two RS-232-C/V.24 interfaces is an asymmetrical line, i.e. the common ground connection between transmitter and receiver is used as a return wire.

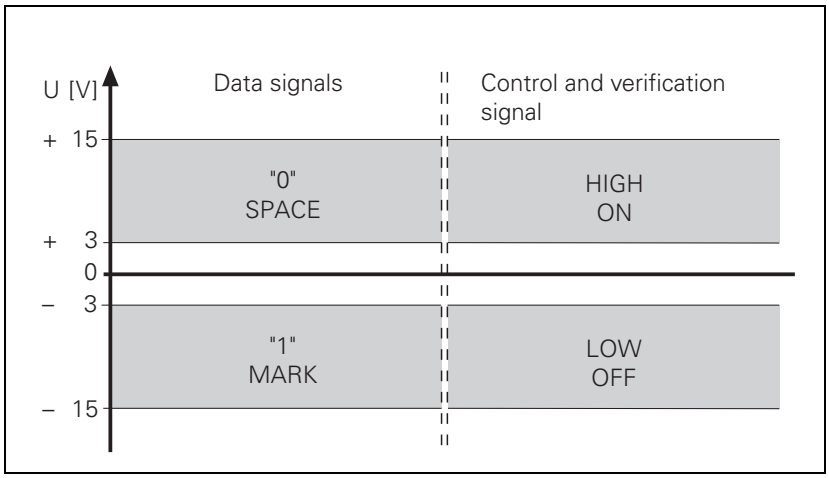
Physical connections:



Signal levels

The levels of the individual signal lines differ:

- Data lines: The data signals are defined as being logical zero (SPACE) over the range +3 V to +15 V and logical one (MARK) over the range –3 V to –15 V.
- Control and verification lines: These signals are defined as being ON (High) over the range +3 V to +15 V and as OFF (Low) over the range –3 V to –15 V.



Note

For all signals: The voltage range from –3 V to +3 V cannot be evaluated.

Signal designations

One must differentiate between the following types of lines and their signals:

- Data lines:
  - TxD Transmitted data
  - RxD Received data
- Control and signal lines:
  - DCD (Data Carrier Detect):
    Received signal level. The receiver signals that the information it has received lies within the defined level. The DCD signal is not used by the control. The control delivers no signal from this pin.
  - DTR (Data Terminal Ready):
    Control is ready / not ready for operation (e.g. the receiving buffer is full, the signal DTR indicates "LOW").
  - DSR (Data Set Ready):
    Peripheral device ready / not ready for service.
  - RTS (Request to Send):
    Switch transmission unit on. The control wishes to transmit data.
  - CTS (Clear to Send):
    Readiness for transmission. The peripheral wishes to transmit data.
- Ground conductors (lines for power supply):
  - Chassis GND:
    Housing connection
  - Signal GND:
    0 V lines for all signals

Pin layout

Keep in mind that there might be a difference between the pin layout of the control and the adapter block.

9.5.2 RS-422/V.11 interface

The RS-422/V.11 serial interface is suitable for data transfer rates up to 10 Mbps.

The interface modules of the control can transfer data at up to 115 200 bps.

Transmission distance: over 1 kilometer

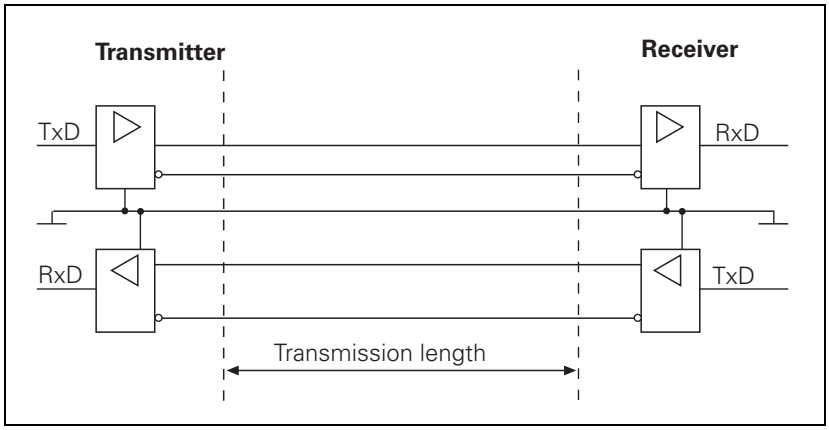
Hardware

The interface works symmetrically, using two signal lines. At the receiver, the difference in voltage of the two lines is evaluated.

Advantage:

- Large transmission distances are possible
- High data transfer rates

Physical connections:



Signal levels

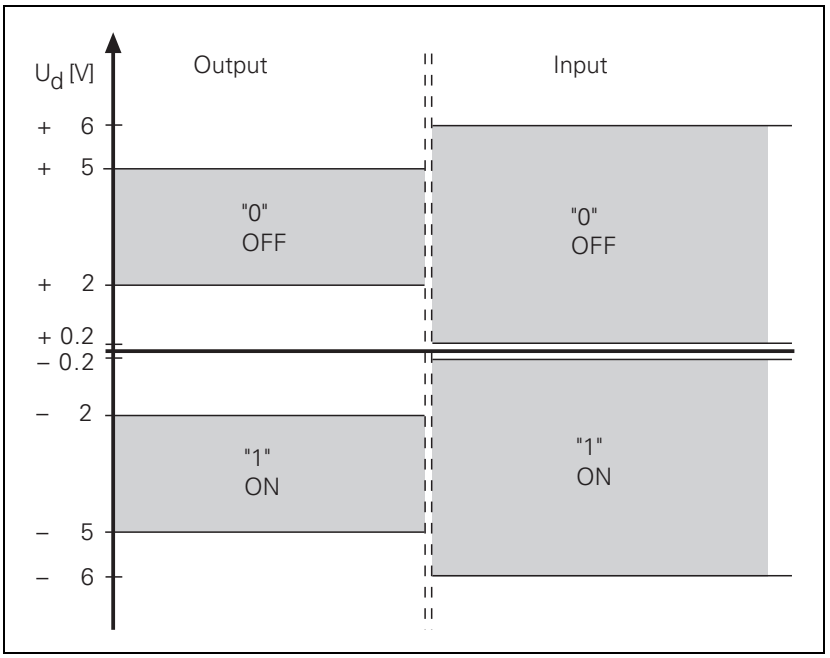
The signals are both transmitted and received as differential voltage.

A positive differential voltage corresponds to logical zero (OFF).

A negative differential voltage corresponds to logical one (ON).

V_{dmin} = 2 V and V_{dmax} = 5 V

The control unit detects the differential voltages between U_{dmin} = 0.2 and U_{dmax} = 6 V as a logically defined level.



Signal designations

The following signals are transmitted as differential signals:

Signals	Signal designation	
Data signals	TxD, $\overline{\text{TxD}}$	RxD, $\overline{\text{RxD}}$
Control and message signals	RTS	CTS
	DSR	DTR

The protective ground connects the transmitter and receiver housings.

GND is the differential voltage reference conductor.

These signals perform the same functions as those on the RS-232-C/V.24 interface.

Pin layout

The control and the adapter block have the same pin layout.

9.6 Configuring the Serial Interface

9.6.1 Control characters

Overview of control characters specific to HEIDENHAIN

Character	Description	Description
SOH	Start of Header	Identifies the beginning of the data transfer header. The character string contains the program number and information about the type of program and the transfer mode.
STX	Start of Text	Identifies the beginning of a program block.
ETB	End of Text Block	Terminates a data transfer block. The character that follows (BCC) is used for data checking.
DC1	XON	Starts the transfer of data.
DC3	XOFF	Stops the transfer of data.
ETX	End of Text	Transmitted at the end of a program.
EOT	End of Transmission	Terminates the data transfer and establishes the idle state. This character is transmitted by the control at the end of a program input and to the external device in the event of an error.
ACK	Acknowledgment	Transmitted by the receiver when a data block has been transferred without error.
NAK	Negative Acknowledgment	Transmitted by the receiver when a data block has been transferred with an error. The transmitter must re-transmit the data block.

Settings in the configuration editor	MP number
System	
Network	
Serial	
CfgSerialPorts	
activeRs232	106601
interfaceRs232	106602
interfaceRs422	106604
interfacePlc	106605
[0]: Keyname Interface PLC 0	106605.0
[1]: Keyname Interface PLC 1	106605.1
[2]: Keyname Interface PLC 2	106605.2
baudRateLsv2	106606
CfgSerialInterface	
[Key names of the interface	
paramet	106701
ers]	106702
baudRate	106703
protocol	106704
dataBits	106705
parity	106706
StopBits	106707
flowControl	106708
fileSystem	106709
bccAvoidCtrlChar	106710
rtsLow	
noEotAfterEtx	

Specifying and selecting interface parameters

You have the possibility of managing multiple parameter sets for the serial interface at the same time by using the configuration editor. The various key names in the CfgSerialInterface parameter object are used to distinguish between the various interface configurations. You assign the settings to the interface by entering the key name in the **MP_interfaceRs232** or **MP_interfaceRs422** parameter. This enables you to change quickly between different settings, for example if you frequently connect peripheral devices with different interface parameters. You configure the interface parameters in the **CfgSerialInterface** configuration object. Under each key name, the properties of a serial port are defined.

In addition, up to three different parameter sets are available via the PLC when using the serial interface. They are assigned by key names in the parameter **MP_interfacePlc.** If no key name has been defined, the parameter set configured under **MP_interfaceRs232** or **MP_interfaceRs422** is automatically used by the PLC (Module 9100) for assignment of the interface.

MP_activeRs232

Cannot be used for the MANUALplus 620!
Enable the RS-232 interface in the program manager
Available from NCK software version: 597 110-03.

Format: Selection menu

Selection: **TRUE**
The RS-232 interface is enabled in the program manager and shown as a drive icon **(RS232:).**
FALSE
The RS-232 interface cannot be accessed via the program manager.

Default: No entry, value optional

Access: LEVEL3

Reaction: NOTHING

MP_interfaceRs232

Key name of the data record for the RS-232 interface
Available from NCK software version: 597 110-01.

Format: String

Input: Max. 18 characters
Define the default parameter set for the serial RS-232 interface here. The "Default" data record is selected by default. But you can use any desired designation. The specified record must be contained in **MP_CfgSerialInterface.** The record is not effective if another record was activated by the PLC.

Default: No entry, value optional

Access: LEVEL2

Reaction: NOTHING

MP_interfaceRs422

Key name of the data record for the RS-422 interface
Available from NCK software version: 597 110-01.

Format: String

Input: Max. 18 characters
Define the default parameter set for the serial RS-422 interface here. The "Default" data record is selected by default. But you can use any desired designation. The specified record must be contained in **MP_CfgSerialInterface.** The data record is not effective if another record was activated by the PLC.

Default: No entry, value optional

Access: LEVEL2

Reaction: NOTHING

MP_interfacePlc

Key names of the data records for interface access by the PLC
Available from NCK software version: 597 110-01.

Format: Array [0...2]

Input: A string of max. 18 characters
Here you can enter a maximum of three different key names for interface accesses by the PLC. If no parameter set is specified, the control automatically uses the default parameter set defined in **MP_CfgSerialInterface.**

Default: [0]: PLC

Access: LEVEL3

Reaction: NOTHING

Defining the LSV2 baud rate

MP_baudRateLsv2

Data transfer rate for LSV2 communication in baud
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: Use a selection menu to define the transfer rate for the LSV2 communication. Minimum value is 110 baud, maximum value 115200 baud.

Default: BAUD_57600

Access: LEVEL2

Reaction: NOTHING

Creating parameter sets, configuring interface ports

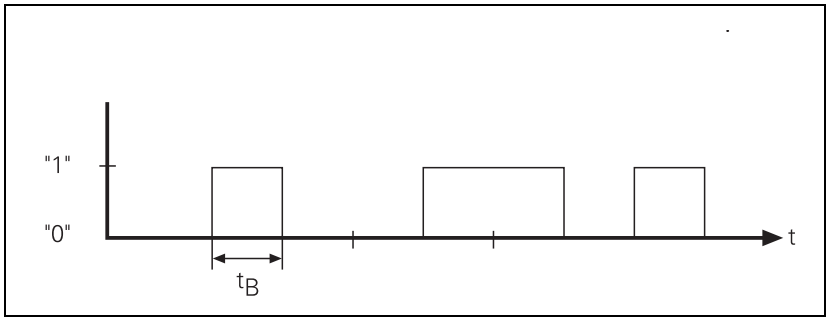
Use the **MP_CfgSerialInterface** configuration object to manage the individual parameter sets for the serial interface. Each parameter set is identified by a key name, and contains the properties of the respective connection. In **MP_CfgSerialPorts** you define which of the parameter sets is active, see page 1362. The interface settings to be defined are described below.

Data transfer rate: Baud rate

The data transfer rate is given in baud (bits per second).
Common transfer rates are:

110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 baud

The time taken to transmit one bit (t_B) can be calculated from the transfer rate:



$$t_B = \frac{1}{\text{transfer rate (bit/s)}}$$

For example, a transfer rate of 19 200 bps will have a bit duration of t_B = 52.083 µs.

$$t_B = \frac{1}{19200\ \text{(bit/s)}} = 52{,}083\ \mu s$$

The number of characters transmitted per second can be calculated from the transfer rate and the transmission format:

$$\text{characters transmitted per second} = \frac{\text{transfer rate (bit/s)}}{\text{number of bit per chacacters}}$$

Example:

With a transmission format of one start bit, 7 data bits, two stop bits and a data transfer rate of 300 bps, exactly 30 characters per second will be transmitted.

$$\text{characters per second} = \frac{300\ \text{(bit/s)}}{1 + 7 + 2} = 30$$

MP_baudRate

Data transfer rate for LSV2 communication in baud
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: Use a selection menu to define the transfer rate for the data transmission. Minimum value is 110 baud, maximum value 115200 baud.

Default: BAUD_9600

Access: LEVEL2

Reaction: NOTHING

Communications protocol

The protocol of a serial connection means the controlling of the data flow by feeding reserved ASCII characters into the data stream. Define the communications protocol of the interface in **MP_protocol.**

For an overview of all communications protocols available on the control, see page 1373.

MP_protocol

Communications protocol
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **STANDARD**
Standard data transfer. Data transferred line-by-line
BLOCKWISE
Package-based data transfer, "ACK/NAK" protocol. Blockwise data transfer is controlled by the control characters ACK (Acknowledge) and NAK (Not Acknowledge).
RAW_DATA
Data transferred without protocol. Transfer of characters without control characters. Protocol intended for transfer of data of the PLC.

Default: STANDARD

Access: LEVEL2

Reaction: NOTHING

Note

Here, the BLOCKWISE setting designates a form of data transfer where data is transmitted in blocks. This is not to be confused with the blockwise data reception and simultaneous blockwise processing by older TNC contouring controls. Blockwise reception of an NC program and simultaneous machining of the program is not possible!

Word length Data bits

Define whether a character is transmitted with 7 or 8 data bits.

MP_dataBits

Data bits in each transferred character
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **7 bits**
7 bits are transferred for each character transferred
8 bits
8 bits are transferred for each character transferred
Default: 8 bits
Access: LEVEL2
Reaction: NOTHING

Transmission reliability: Parity bit

The parity bit helps the receiver to detect transmission errors.
The parity bit can take three different forms:

- No parity (NONE): There is no error detection.
- Even parity (EVEN): The transmitter counts bits with a value of one. If the number is odd, the parity bit is set to one, otherwise it is cleared to zero. The sum of set data bits and the parity bit is therefore always even. Upon receiving a word, the receiver counts all of the set bits, including the parity bit. If the count is odd, there is a transmission error.
- Odd parity (ODD): The parity bit is chosen by the transmitter so that the total number of all the set bits is odd. An error will thus be detected if the receiver observes an even number of set bits in its evaluation.

Example: The letter "z" corresponds to the bit sequence: 1 1 1 1 0 1 0

Parity bit:

- With even parity = 1
- With odd parity = 0

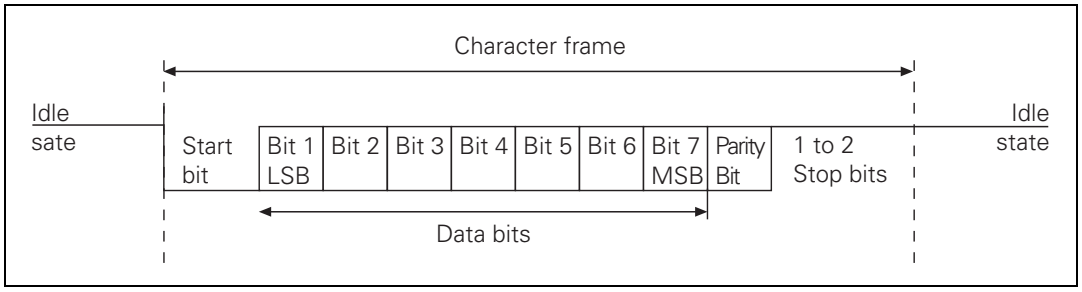
MP_parity

Specifies the type of parity checking
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **NONE**
No parity formation
EVEN
Even parity
ODD
Odd parity
Default: NONE
Access: LEVEL2
Reaction: NOTHING

Synchronization: Stop bits

The start bit and one or two stop bits enable the receiver to synchronize to every transmitted character during serial data transmission.



One start bit is sent before each character. In **MP_stopBits,** you determine the number of stop bits sent at the end of a character:

MP_stopBits

Number of stop bits
Available from NCK software version: 597 110-01.

Format:	Selection menu
Selection:	1 stop bit 1 stop bit is appended after each transferred character. 2 stop bits
Default:	1 stop bit
Access:	LEVEL2
Reaction:	NOTHING

Data transfer check: Handshaking

By handshaking, two devices control data transfer between them. A distinction is made between software handshaking and hardware handshaking.

You can choose either of the two procedures:

Hardware handshaking

Data transfer is controlled by electrical signals. Information, such as Clear to Send (CTS), Request to Send (RTS), "Start transmission" and "Stop transmission" is passed on by the hardware.

Example:

When a computer is to transmit a character, it checks the CTS signal line to see whether it is active (ON). If it is, the character is transmitted.

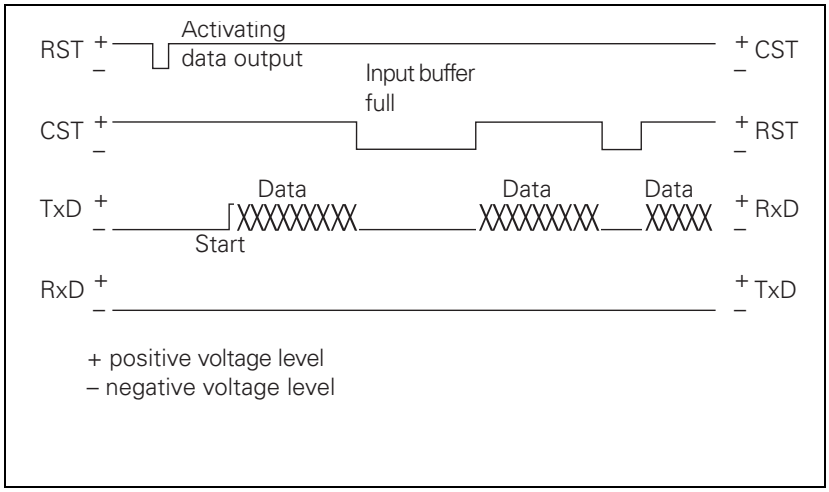
Hardware handshaking requires

- the data lines TXD and RXD (transmitted and received data)
- the RTS control line (switching on transmitting unit)
- the CTS signal line (Clear to Send)
- a ground connection

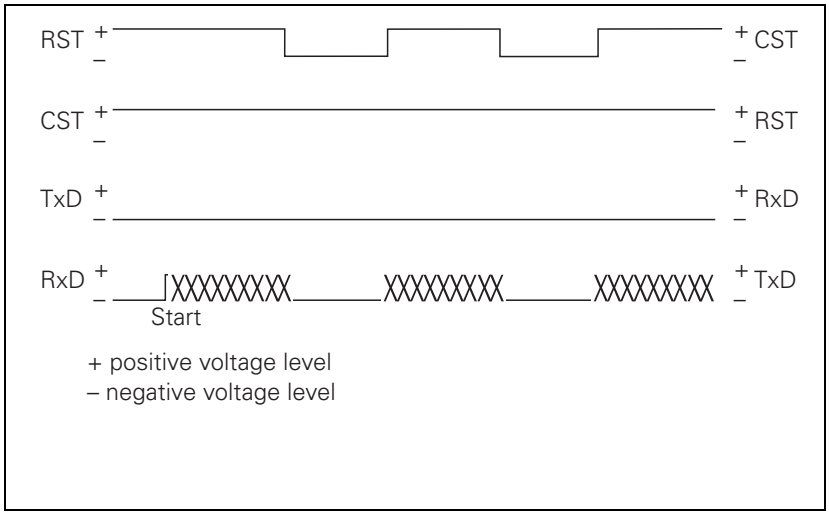
The DTR and DSR signals indicate the operational status of the LE and peripheral device:

- DTR: Interrogated by peripheral; it is logical one if LE is ready for operation.
- DSR: Interrogated by LE.
  - LOW level means: external data input/output not ready.
  - HIGH level means: external data input/output ready.

■ Data output from the control to EXT
When the receiving buffer is full, the external device resets the RTS signal. The control detects that the peripheral unit receiving buffer is full at its CTS input:



■ Data input from EXT to the control
When the receiving buffer is full, the control removes the RTS signal. This is detected by the peripheral device at its CTS input:



Software handshaking

Control of data transfer is achieved by control characters transmitted via the data line.

In **MP_flowControl,** you define whether the control stops transfer from an external device with control character <DC3>. Transfer is then resumed with character <DC1>. (XON/XOFF method)

If transfer is stopped with the control character <DC3>, up to three more characters can be stored; any further incoming characters are lost. Software handshaking is normally recommended when interfaces are connected to an external device.

MP_flowControl

Handshaking: Type of data-flow checking
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **NONE**
No data-flow checking; handshaking not active
RTS_CTS
Hardware handshaking; transfer is stopped with RTS active
XON_XOFF
Software handshaking; transfer is stopped with DC3 (XOFF) active

Default: RTS_CTS

Access: LEVEL2

Reaction: NOTHING

Defining the file system

In **MP_fileSystem,** define the file system for data transmission over the serial interface. This machine parameter is not mandatory. Remove the parameter from the configuration if you do not need a special file system.

Select one of two different file systems from the selection menu.

MP_fileSystem

Define the file system for file operation via the serial interface
Available from NCK software version: 597 110-01.

Format: Selection menu

Selection: **EXT**
Minimum file system for external devices. Corresponds to the EXT1 and EXT2 modes of earlier TNC controls. Use these settings if you are using printers, punches, or non-HEIDENHAIN data transfer software.
FE1
Use this setting for communication with the external HEIDENHAIN FE 401 B or FE 401 floppy disk unit as of software 230 626-03, or for communication with the "TNCserver" PC software from HEIDENHAIN.

Default: EXT

Access: LEVEL2

Reaction: NOTHING

Block Check Character (BCC)

The BCC is a block check character. The BCC is added to a transfer block to simplify error recognition. During the horizontal parity check and the cyclic block check, block check characters are calculated and added to every transferred data block. The appended BCC is compared with a second BCC that is calculated by the receiver. The comparison determines whether the transmission was executed without error.

Use **MP_bccAvoidCtrlChar** to ensure that the BCC is not interpreted as a control character.

On the control, numbers less than $20 are defined as control characters. If calculation of the BCC produces a number less than $20, then a blank space is sent in addition immediately before <ETB>. The BCC will consequently always be greater than $20 and cannot therefore be interpreted as a control character.

Note

You can remove this parameter from your configuration if you do not use blockwise data transfer.

MP_bccAvoidCtrlChar

Block Check Character (BCC) is not a control character
Available from NCK software version: 597 110-01.

Format: Selection menu
Selection: **TRUE**
Ensures that the check sum does not correspond to a control character
FALSE
Function not active
Default: FALSE
Access: LEVEL2
Reaction: NOTHING

Status of the RTS line

When using RTS/CTS hardware handshaking, two control lines in the RS-232 cable are used to signalize readiness to send data to or receive data from the other side. If the control wants to send data to the peripheral device, and the device is ready to receive data, then the RTS line is set to HIGH. The peripheral device sets the CTS line to HIGH in order to signalize its readiness to receive data from the control.

In the idle state, the control normally provides a HIGH level on the RTS line. You can change the idle state of the RTS line from HIGH to LOW level by setting the optional **MP_rtsLow** parameter.

MP_rtsLow

Idle state of the RTS line
Available from NCK software version: 597 110-03.

Format: Selection menu
Selection: **TRUE**
The idle state of the RTS line is logical LOW
FALSE
(default) The idle state of the RTS line is logical HIGH
Default: No entry, value optional
Access: LEVEL2
Reaction: NOTHING

Behavior after receipt of ETX

With the optional parameter **MP_noEotAfterEtx** you define the behavior of the control after reception of the ETX control character. ETX signalizes the end of input for software handshaking. Normally the control automatically sends an EOT control character after reception of an ETX control character (EOT= End Of Transmission). Set the **MP_noEotAfterEtx** parameter to the value TRUE in order to deactivate transmission of the EOT character.

MP_noEotAfterEtx

Defines the behavior after reception of an ETX control character
Available from NCK software version: 597 110-03.

Format: Selection menu
Selection: **TRUE**
No EOT control character is sent after reception of an ETX control character.
FALSE
(default) The control sends an EOT control character after reception of an ETX control character.
Default: Value optional, no entry
Access: LEVEL2
Reaction: NOTHING

9.7 Data Transmission Protocols

9.7.1 Standard communications protocol

General information

To set the standard communications protocol:

▶ See "MP_protocol" on page 1366.

When outputting a file, the <NUL> character is sent exactly 50 times at the start of file. When reading in, however, the control unit ignores this character, regardless of how often the peripheral sends the <NUL> character before the file.

The program blocks are not checked for correctness but are transmitted one after the other.

If you wish to signal an error to the control in the standard communications protocol, you must send the following sequence of instructions:
<ESC><1><Error number>

If the receiver's data buffer is full, the transmission can be stopped and resumed in one of two ways:

- Software handshaking
  - Stop transfer by sending the character <DC3> (XOFF)
  - Continue by transmitting the character <DC1> (XON)
- Hardware handshaking
  - By suitable levels on the control and message lines RTS and CTS of interfaces RS-232-C/V.24 or RS-422/V.11

Twelve characters before the receiving buffer is full, the control transmits the character <DC3> to the transmitter in order to terminate transmission.

Example: Protocol for conversational NC program

<NUL><NUL><NUL><NUL> 50 times

0 BEGIN PGM 1 MM<CR><LF> 1st program block

1 TOOL DEF 1 L+0 R+3<CR><LF> 2nd program block

26 END PGM 1 MM <CR><LF> End of program

... ...

<ETX><EOT> Close the data transmission menu

Example of software handshake

Control to peripheral device	Peripheral device to control
12 Z + 2 FMAX<CR><LF>	Receiving buffer full: <DC3> Receiving buffer ready again: <DC1>
13 Z -10 FMAX<CR><LF>	...

Output selected file

The serial interface uses software handshake.

The control outputs all of the program lines in sequence.

The peripheral device can:

- Stop transmission with <DC3>
- Resume transmission with <DC1>

Control to peripheral device	Peripheral device to control
<NUL> <NUL> <NUL>... 1. line of file <CR> <LF>... 5. line of file <CR> <LF>...	 Transmission stop: <DC3> Resume transmission: <DC1>
6. line of file <CR> <LF>... Last line of file <CR> <LF>	...

Load selected file

The serial interface uses software handshake.

To transfer a file from a peripheral device:

▶ Enter the file name in the control.

The control can:

- Stop transmission with <DC3>
- Resume transmission with <DC1>

Control to peripheral device	Peripheral device to control
100.H "START" <DC1>	 <NUL><NUL> 1. line PGM100<CR><LF> ... Last line PGM 100 <CR><LF><ETX>
<EOT>	

If the file name in the first line and the name indicated in the control are not identical, the control downloads each block and searches for the correct file name. If the END PGM block has been downloaded, and the selected name has not been found, the control stops transfer without an error message:

▶ In this case, terminate transfer with the END key.

9.7.2 Communications protocol with block check character

This protocol is specific to HEIDENHAIN and operates with its own control characters and an additional data check feature when transmitting.

The protocol is active during blockwise data transfer in FE1 mode, see "MP_fileSystem" auf Seite 1370.

In FE1 mode, a command sequence is output at the beginning to request the contents directory from the peripheral device.

Header

When a file is transferred, the first block — called the header — consists of the following characters:
<SOH><K><Name><M><ETB><BCC><DC1>

Character	Meaning
<SOH>	Identifies the beginning of the header
<K>	File code
<Name>	File name
<M>	Data transfer mode (E = input, A = output)
<ETB>	Identifies the end of the header
<BCC>	Block Check Character
<DC1>	XON

Block Check Character (BCC)

In addition to checking the parity of the individual characters, the parity of the complete transferred block is also checked. The BCC always rounds the individual bits of the transferred characters in a data transfer block to even parity.

Example of BCC generation:

In this example, program 15, which has been written in HEIDENHAIN plain-language text (H), is input through the data interface (E).

Character	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SOH	0	0	0	0	0	0	1
H	1	0	0	1	0	0	0
1	0	1	1	0	0	0	1
5	0	1	1	0	1	0	1
E	1	0	0	0	1	0	1
ETB	0	0	1	0	1	1	1
BCC	**0**	**0**	**1**	**1**	**1**	**1**	**1**

A parity bit is also generated for the BCC. With even parity, the parity bit in this example is assigned the value 1.

At the end of every block, the receiver checks whether it has been transferred correctly.

To do this, the receiver computes a BCC from the received block and compares it with the received BCC. If the received BCC and the computed BCC are identical, the receiver transmits the character <ACK> for positive acknowledgment. If the two BCCs are not identical, the data block was not transmitted correctly. The receiver transmits the character <NAK> for negative acknowledgment. The block must be re-transmitted. This process is repeated up to 15 times.

If the header is acknowledged with <ACK>, the first file block can be transmitted:

The beginning of a file block is identified by the control character <STX>. The remaining control characters in this block are identical with the control characters in the header. If this block is acknowledged by <ACK>, then the next program block is transmitted. If <NAK> is transmitted, the same block has to be re-transmitted, etc. Once the last program block has been acknowledged by <ACK>, the transmission is terminated by the characters <ETX> (end of text) and <EOT> (end of transmission).

Handshaking

The character <DC1> (XON) follows the BCC. This character is required by many devices to explicitly request the transmission once again from the transmitter.

The <DC1> character is not required for reading in a file in the BCC format.

The transmitter waits and only resumes data transmission when the receiver has transmitted a positive (ACK) or negative (NAK) acknowledgment to indicate that the receiving buffer is ready.

To disable transmission of the character <DC1>:

- Set **MP_flowControl** to NONE.

Example:

A file with the name PPP is to be transferred to a peripheral device (e.g. HEIDENHAIN FE 401 Floppy Disk Unit).

Control to peripheral device	Peripheral device to control
<SOH><L>PPP<A><ETB>BCC	ACK
<STX> "1st line"<ETB>BCC	<ACK>
...	...
<STX> "10th line"<ETB>BCC	<NAK>
<STX> "10th line"<ETB>BCC	<ACK>
<STX> "11th line"<ETB>BCC	<ACK>
...	...
<STX> "last line"<ETB>BCC	<ACK>
<ETX><EOT>	

Report an error to the control

FE1 mode is set.

If an error occurs at a peripheral device, the following block must be sent to the control:

<SOH><Error text><ETB>BCC

Peripheral device to control	Control to peripheral device
<SOH> "Error"<ETB>BCC	<ACK><EOT>

The error message received will be displayed on the control. To continue

▶ Press the CE key.

Request external directory

FE1 mode is set.

In FE1 mode the following Escape sequence is sent to request the external directory:

<DC3><ESC><DC1><0><SP><D><CR><LF>

The control expects the following input to this request:

xxxxxx<Code letters><Sectors><Name><P[1)]><CR><LF>
[1)] P = Protected (optional)

The first four lines, each ending in <CR><LF>, are ignored. In subsequent lines ending with <CR><LF>, the program name and, after any number of blank characters, the number of sectors are stored.

If the character combination <FREE:> is detected, only a number—the number of free sectors—will be transferred.

The control requests the complete directory. The directory is saved and the files of the selected type are displayed.

The peripheral device ends transmission with <EXT>. The control responds with <EOT>.

Output selected file

Control to peripheral device	Peripheral device to control
<SOH><K>Name<A><ETB>BCC	<SOH><K>Name<A><ETB>BCC
<STX> "1st line"<ETB>BCC	<ACK>
...	...
<STX> "last line"<ETB>BCC <DC1>	<ACK>
<ETX><EOT>	

Output marked files

Marked files are output in the same protocol as for outputting the selected files. After each file, the control characters <EXT><EOT> are sent to the peripheral device.

Load selected file

To download a file from an external storage device, the control transmits a header with the corresponding file name.

Control to peripheral device	Peripheral device to control
<SOH><K>Name<E><ETB>BCC <DC1>	<ACK> <STX>"1st line"<ETB>BCC<DC1>
<ACK>	...
...	<STX> "last line"<ETB>BCC <DC1>
<ACK>	<ETX>
<EOT>	

9.7.3 LSV2 transmission protocol

The LSV2 protocol is a data transfer protocol for the two-way transfer of commands and data.

The data is transferred in blocks — so-called telegrams — into which the data is split up.

The following functions are possible:

- Data transfer
- File management, such as deleting, copying and renaming files
- Changing, creating and deleting paths
- Remote operation of the control functions. The TNC screen appears on the computer monitor. All functions can be executed from the computer.
- Real DNC operation. Starting and stopping the machine from the PC
- Diagnosis of control error messages and keystrokes for service purposes. The last 1000 events are stored in the control.

HEIDENHAIN offers two LSV2 software packages. Please contact HEIDENHAIN for further information.

9.8 Saving and Loading Files

The table lists all the files that can be saved to external memory devices and loaded from them.

File	File extension
MANUALplus 620 cycle programs	.gmz
NC program, DIN/ISO	.nc, .ncs
Tool table	.htt
Datum table	.hzp
Machine parameters	.CFG
Compensation-value table	.COM
Compensation-value assignment	.CMA
PLC program	.PLC
PLC source code files	.SRC
Text file	
Pocket table	.TCH
Help files	
Point table	
PLC error table	.PET
Cutting-data table	
Freely definable tables	
Motor table	.MOT
Motor table (servo amplifiers)	.INV
Error file	.log
OEM cycles	
Oscilloscope recordings	.SCO

To write to or read from machine parameter files, compensation tables or PLC files, you must enter the correct code numbers. For data transmission with the **TNCremoNT** PC software from HEIDENHAIN the identifier of a file has no significance. The files are saved on the PC with the same extension as on the control.

9.9 Configuring the Control for TeleService 2.0

The TeleService 2.0 software for PCs offers numerous possibilities for remote maintenance and remote diagnosis of the MANUALplus 620. Along with the transmission of the screen content, all soft keys of the control can be operated remotely, for example.

A virtual screen keyboard (for MANUALplus as of TeleService 2.1) can be displayed, making remote key entry possible. In addition, comprehensive diagnostic possibilities, such as outputting of control logs, are available.

The machine operator can initiate a SERVICE REQUEST via soft key, as well as establish an automatic connection between the control and TeleService. In the **Transfer** mode, press the **TeleService** soft key after configuring the machine parameters described below. Please contact HEIDENHAIN if you would like to use TeleService 2.0.

For TeleService 2.0, you must connect the control to the network. For information, please refer to the "Transfer" chapter in the User's Manual. Then you can access the control with TeleService via the host name or IP address.

The following machine parameters are available for configuration of the service request:

Settings in the configuration editor	MP number
System	
Network	
CfgServiceRequest	
name	114601
host	114602
port	114603
content	114604
period	114605
timeout	114606
serverIp	114607
serverIpMask	114608

MP_name

Logical name of the service host
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 500 characters
The name identifies the remote service host and can be displayed on the user interface of the control.
Default: No value, parameter optional
Access: LEVEL2
Reaction: NOTHING

MP_host

Network address or host name of the remote service host
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 500 characters
Enter the network address as four decimal numbers separated by periods (IP address in dotted-decimal notation) or as a fully qualified host name (e.g.:
192.168.10.51
\HOME\DATA\SERVICE
Default: No value, parameter optional
Access: LEVEL2
Reaction: NOTHING

MP_port

Port number of the remote service host
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: Initial value: 19001
The control sends the service request report to the port entered here
Default: No value, parameter optional
Access: LEVEL2
Reaction: NOTHING

MP_content

Contents of the message to the remote service host
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 500 characters
The contents serve to identify the machine, e.g. machine model and serial number
Default: No value, parameter optional
Access: LEVEL2
Reaction: NOTHING

MP_period

Duration of repeated transmission of the message to the service host
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0 to 10 [s]
Transmission of the report is repeated during the specified period of time in seconds until the function is activated or the time entered in the **MP_timeOut** parameter is exceeded.
Default: No value, parameter optional
Access: LEVEL2
Reaction: NOTHING

MP_timeout

Timeout in minutes for transmission of the message to the remote service host
Available from NCK software version: 597 110-03.

Format: Numerical value
Input: 0 to 15 [min]
Default: No value, parameter optional
Access: LEVEL2
Reaction: NOTHING

MP_serverIp

Network address or host name of the server
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 500 characters
The data traffic of the specified server(s) is checked. The default value is the server given in the **MP_host** parameter.
Default: No value, parameter optional
Access: LEVEL2
Reaction: NOTHING

MP_serverIpMask

Subnet mask of the server network
Available from NCK software version: 597 110-03.

Format: String
Input: Max. 500 characters
Enter the subnet mask of the server as four decimal numbers separated by periods (IP address in dotted-decimal notation).
Default: No value, parameter optional
Access: LEVEL2
Reaction: NOTHING

9.10 The Transfer Mode of Operation

The Transfer mode is used for data backup and data exchange with other IT systems. NC program files, parameter or tool files are transferred.

The data is transmitted through Ethernet-based networks or via the USB interface. The control supports the networks provided by WINDOWS. The control uses the "dual-window concept". This means you see the files of your own system in the left and the files of the other side in the right half of the screen.

- You send data from the control or receive data from the other side. This is always initiated by the control.
- Use **TNCremoNT** if you want to send files from a PC to the control or "collect" them from the control. **TNCremoNT** is available free of charge from HEIDENHAIN.

The following transfer functions are available:

- **Programs** - Transmitting and receiving files
- **Back up parameters** Creating, transmitting and receiving
- **Restore Parameters**: - Reloading the parameter backup files
- **Backup tools**: Creating, transmitting and receiving
- **Restore Tools** - Reloading the tool backup files
- **Service** - Creating and transmitting service data
- **Data Backup** - Backing up **all** data in a project folder
- **Free external** - Freely selecting program files on a USB storage device
- **Miscellan. functions** - Importing the cycle programs and DIN programs of the MANUALplus 4110 and DIN PLUS programs of the CNC PILOT 4290

Note

For detailed information and descriptions of the Transfer mode of operation of MANUALplus 620, refer to the User's Manual for the control.

9.11 Data Transfer by PLC

9.11.1 PLC Modules

With the following PLC modules you can control the data interfaces from the PLC:

- Modules 9100 and 9101: Assign/release the data interfaces
- Module 9102: Interrogate the status of the interface
- Modules 9103 and 9104: Transmit and receive a string from the string memory. The transmit and receive buffers for the PLC are 128 characters long. Since every STRING ends with an END character, a STRING can only be up to 127 characters long.
- Modules 9105 and 9106: Transfer a block of binary values (bytes) from the word memory
- Module 9107: Read bytes from the receiving buffer without erasing the buffer
- Modules 9112 and 9113: Send or receive ASCII characters via the data interface

Strings and binary data are transmitted using ASCII characters. Example: Transfer of a block of binary data

Address	Value	ASCII character
.	.	.
B126	11111010	$FA
.	10000001	$81
.	.	
.	.	
.	.	

When transferring binary data starting from the address B126, the ASCII characters <F> <A> <8> <1> etc. are transmitted in sequence from the word memory through the interface. Each byte contains two ASCII characters. The transmitting and receiving buffers each hold 63 bytes.

Module 9100 Assign data interface
Module 9100 assigns a serial interface to the PLC and configures the transfer parameters. They initialize the interface, thereby erasing any errors that may have occurred. The interface is then ready to receive.

Once assigned to the PLC, the interface is disabled for use by the input/output program of the user interface.
Input/output program of the user interface is locked. The assignment is canceled when the PLC program is recompiled.

Can only be called in a submit job or spawn job!

Call:
PS B/W/D/K <Interface>
0: RS232
1: RS422
PS B/W/D/K <Transfer parameters>
0: Entry from interfacePlc[0] is used
1: Entry from interfacePlc[1] is used
2: Entry from interfacePlc[2] is used
CM 9100

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Interface was configured for the PLC and assigned
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Incorrect interface or incorrect transfer parameter
	13	No connection
	14	Interface busy or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9101 Release data interface
Module 9101 cancels the assignment of an interface to the PLC. The receive mode of the interface is canceled.

Can only be called in a submit job or spawn job!

Call:
PS B/D/W/K <Interface>
0: RS232
1: RS422
CM 9101

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Interface enabled
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Incorrect interface
	14	Interface not assigned
	20	Module was not called in a spawn or submit job

Module 9102 Status of data interface

Module 9102 reads the status information about an interface in bit-coded form.

The information "interface ready" is updated when the interface is assigned to the PLC or NC. If the interface is not assigned, the module reads the last valid status.

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
CM	9102	
PL	B/W/D	<Interface status> –1: Error code in NN_GenApiModuleErrorCode Bit 0: Interface is assigned Bit 1: Interface is assigned to the PLC Bit 2: Interface is ready Bit 3: Transmit buffer is empty Bit 4: Error during transmission Bit 5: Receive buffer is full Bit 6: Error in reception Bit 7: ETX was received (not ready to receive) Bit 8: Internal buffer from Module 9113 contains characters

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Status read
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Incorrect interface

Module 9103 Transmit string through data interface
You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9103 transmits a string from a string memory through one of the two interfaces. Links to the PLC error file and PLC dialog file are deleted.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	K/B/W/D	<Number of source string in the string buffer>
CM	9103	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	String was transmitted
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Incorrect interface or incorrect string number
	12	No string end found
	13	Interface not ready
	14	Interface not assigned
	15	Transmit buffer not empty
	20	Module was not called in a spawn job or submit job

Module 9104 Receive string through data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9104 reads a string from the receive buffer of a serial interface in a string memory and resets the receive buffer.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	K/B/W/D	<Number of the string in the string buffer>
CM	9104	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	String was received
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Incorrect interface or incorrect string number
	12	String too long
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9105 Transmit binary data through data interface
You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9105 transmits a block of binary values from the word memory of the PLC to one of the two interfaces. The transfer is in the form of ASCII-coded hexadecimal values. Every byte in the source block makes two ASCII characters at the interface.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	K/B/W/D	<Number of the first byte in the binary block>
PS	K/B/W/D	<Length of the binary block (0 to 63)>
CM	9105	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Data was transmitted
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Incorrect interface or incorrect byte number or block too long
	4	Block outside value range
	13	Interface not ready or no connection
	14	Interface not assigned
	15	Transmit buffer not empty
	20	Module was not called in a spawn or submit job

Module 9106 Receive binary data through data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9106 reads a block of binary values from one of the two interfaces into the word memory of the PLC. The transfer is in the form of ASCII-coded hexadecimal values. Every two ASCII characters from the serial interface make one byte in the binary block.

The length of the read binary block is returned as the initial variable.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	K/B/W/D	<Number of the first byte in the binary block>
CM	9106	
PL	B/W/D	<Length of binary block in bytes> –1: Incorrect module call

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Data was received
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	1	Incorrect interface or incorrect byte number or block too long
	4	Block outside value range
	11	Odd number of characters or illegal character
	12	String too long
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9107 Read from receiving buffer
You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9107 reads two ASCII characters from the receive buffer to one of the two interfaces and codes them to a binary value.

You can specify an offset that corresponds to the position of the byte to be read in a binary block read by Module 9106. The contents of the receiving buffer are retained and can be read by Modules 9104 and 9106.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	B/W/D/K	<Offset of byte to be read in binary block>
CM	9107	
PL	B/W/D	<Binary value read>

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Receiving buffer was read
	1	Error code in NN_GenApiModuleErrorCode ()
NN_GenApiModule ErrorCode	1	Incorrect interface or incorrect byte number
	11	Illegal character
	12	String too long or offset too large
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job

Module 9111 Receive a message via LSV2

Module 9111 reads a message (double word or string) that has been received from a host computer connected by LSV2 protocol.

The message must be transmitted from the host by the LSV2 command "M_PC<msg.l>".

Call:

PS	B/W/D/K	<Data type> 0: Binary data double word 1: String
PS	B/W/D/K	<Target address> With binary: Number of the double word With string: Number of the string
CM	9111	
PL	B/W/D	<Error code> 0: Message was read 1: No connection to host 2: No message of this type in receiving buffer 3: Incorrect data type (not 0 or 1) 4: Incorrect target address

Error recognition:

Marker	Value	Meaning
NN_GenApiModuleError	0	Message was received
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModuleErrorCode	2	Incorrect data type
	4	No double word address, or incorrect string number
	11	String too long
	13	No connection
	15	Transmit buffer not empty
	16	Receiving buffer empty

Module 9112 Transmit ASCII characters via data interface
You must first assign the interface to the PLC and initialize it with Module 9100. Module 9112 transmits a single ASCII character.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
PS	W/D/K	<ASCII code [0 to 255]>
CM	9112	

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Character was transmitted
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Incorrect interface
	13	Interface not ready or no connection
	14	Interface not assigned
	15	Transmit buffer not empty
	20	Module was not called in a spawn job or submit job

Module 9113 Receive ASCII characters via data interface

You must first assign the interface to the PLC and initialize it with Module 9100.

Module 9113 reads a single ASCII character from the receiving buffer of a serial interface and resets the receiving buffer.

If there is more than one character in the receiving buffer, the first is returned and the others are stored in a special buffer.

You can interrogate the current state with Module 9102, bit 8.

As long as data remains in the buffer, no further characters are collected from the interface.

Note

Store the result in a word at least so that the values to 255 will be recognized.

Can only be called in a submit job or spawn job!

Call:

PS	B/W/D/K	<Interface> 0: RS232 1: RS422
CM	9113	
PL	W/D	<ASCII character read [0 to 255 = ASCII characters; –1 = error>

Error recognition:

Marker	Value	Meaning
NN_GenApiModule Error	0	Character was received
	1	Error code in NN_GenApiModuleErrorCode
NN_GenApiModule ErrorCode	1	Incorrect interface
	12	String too long
	13	Interface not ready or no connection
	14	Interface not assigned
	16	Receiving buffer empty
	18	Transmission error or input/output not ready
	20	Module was not called in a spawn job or submit job
	37	Receiving queue is full

Numerics

A

B

C

C

D

E

F

G

H

I

J

K

L

M

M

N

O

P

M

R

S

S

T

U

V

W

X

HEIDENHAIN

DR. JOHANNES HEIDENHAIN GmbH
Dr.-Johannes-Heidenhain-Straße 5
83301 Traunreut, Germany
☏ +49 8669 31-0
FAX +49 8669 5061
E-mail: info@heidenhain.de

Technical support FAX +49 8669 32-1000
Measuring systems ☏ +49 8669 31-3104
E-mail: service.ms-support@heidenhain.de
TNC support ☏ +49 8669 31-3101
E-mail: service.nc-support@heidenhain.de
NC programming ☏ +49 8669 31-3103
E-mail: service.nc-pgm@heidenhain.de
PLC programming ☏ +49 8669 31-3102
E-mail: service.plc@heidenhain.de
Lathe controls ☏ +49 8669 31-3105
E-mail: service.lathe-support@heidenhain.de

www.heidenhain.de

747 323-20 · Ver00 · PDF · 11/2010