SIEMENS

SINUMERIK

SINUMERIK 802D sl
Turning, milling, grinding, nibbling

Operating Instructions

Valid for

Control	Software version
SINUMERIK 802D sl G/N	1.4 SP7
SINUMERIK 802D sl T/M	1.4 SP7

11/2012

6FC5397-0CP10-8BA0

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Order number: 6FC5397-0CP10-8BA0
Ⓟ 08/2013 Technical data subject to change

Copyright © Siemens AG 2005 - 2012.
All rights reserved

Foreword

SINUMERIK documentation

The SINUMERIK documentation is organized in the following categories:

- General documentation
- User documentation
- Manufacturer/service documentation

Additional information

You can find information on the following topics at www.siemens.com/motioncontrol/docu:

- Ordering documentation/overview of documentation
- Additional links to download documents
- Using documentation online (find and search in manuals/information)

Please send any questions about the technical documentation (e.g. suggestions for improvement, corrections) to the following address:

docu.motioncontrol@siemens.com

My Documentation Manager (MDM)

Under the following link you will find information to individually compile OEM-specific machine documentation based on the Siemens content:

www.siemens.com/mdm

Training

For information about the range of training courses, refer under:

- www.siemens.com/sitrain

  SITRAIN - Siemens training for products, systems and solutions in automation technology
- www.siemens.com/sinutrain

  SinuTrain - training software for SINUMERIK

FAQs

You can find Frequently Asked Questions in the Service&Support pages under Product Support. http://support.automation.siemens.com

SINUMERIK

You can find information on SINUMERIK under the following link:

www.siemens.com/sinumerik

Target group

This manual is intended for use by planners, configuration engineers, technicians, installation personnel, programmers, commissioning personnel, operators, service and maintenance personnel

Benefits

The operating instructions impart knowledge about the components and allow the addressed target groups to properly and safely install, set up, test and commission the SINUMERIK 802D sl.

Standard scope

This documentation only describes the functionality of the standard version. Additions or revisions made by the machine tool manufacturer are documented by the machine tool manufacturer.

Other functions not described in this documentation might be executable in the control. However, no claim can be made regarding the availability of these functions when the equipment is first supplied or in the event of servicing.

For the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation, or maintenance.

Licensing provisions

The software SINUMERIK 802D sl is protected by national and international copyright laws and agreements. Unauthorized reproduction and distribution of this software or parts thereof is liable to prosecution. It will be prosecuted both according to criminal and civil law and may result in severe punishment or demands for compensation.

In the software SINUMERIK 802D sl, open source software is used. The licensing provisions for this software are located on the Toolbox CD and are to be observed accordingly.

Acceptance report

You can find a sample report for the acceptance of SINUMERIK 802D sl on the Internet at:

http://support.automation.siemens.com under the heading Current > Acceptance reports

Technical Support

You will find telephone numbers for other countries for technical support in the Internet under http://www.siemens.com/automation/service&support

EC Declaration of Conformity

The EC Declaration of Conformity for the EMC Directive can be found on the Internet at:

http://support.automation.siemens.com

Here, enter the number 15257461 as the search term or contact your local Siemens office.

Contents

1 Description

1.1 System overview

Overview

The CNC operator panel of the **SINUMERIK 802D sl** control system combines all CNC, PLC, HMI, and communication tasks in a single component. The maintenance-free hardware integrates the DRIVE-CLiQ interface for the drives and PROFIBUS interface for the I/O modules with the slimline operator panel, creating a ready-to-install unit (Panel Control Unit).

The **SINUMERIK 802D sl** can perform digital control (with SINAMICS S120) or analog control (with ADI4) of up to 6 axes.

Of the 6 axes, the following configurations are possible:

- A maximum of 5 NC axes and one PLC axis

  Up to 2 of these 5 NC axes can be configured as a spindle.

- A maximum of 3 NC axes and 3 PLC axes

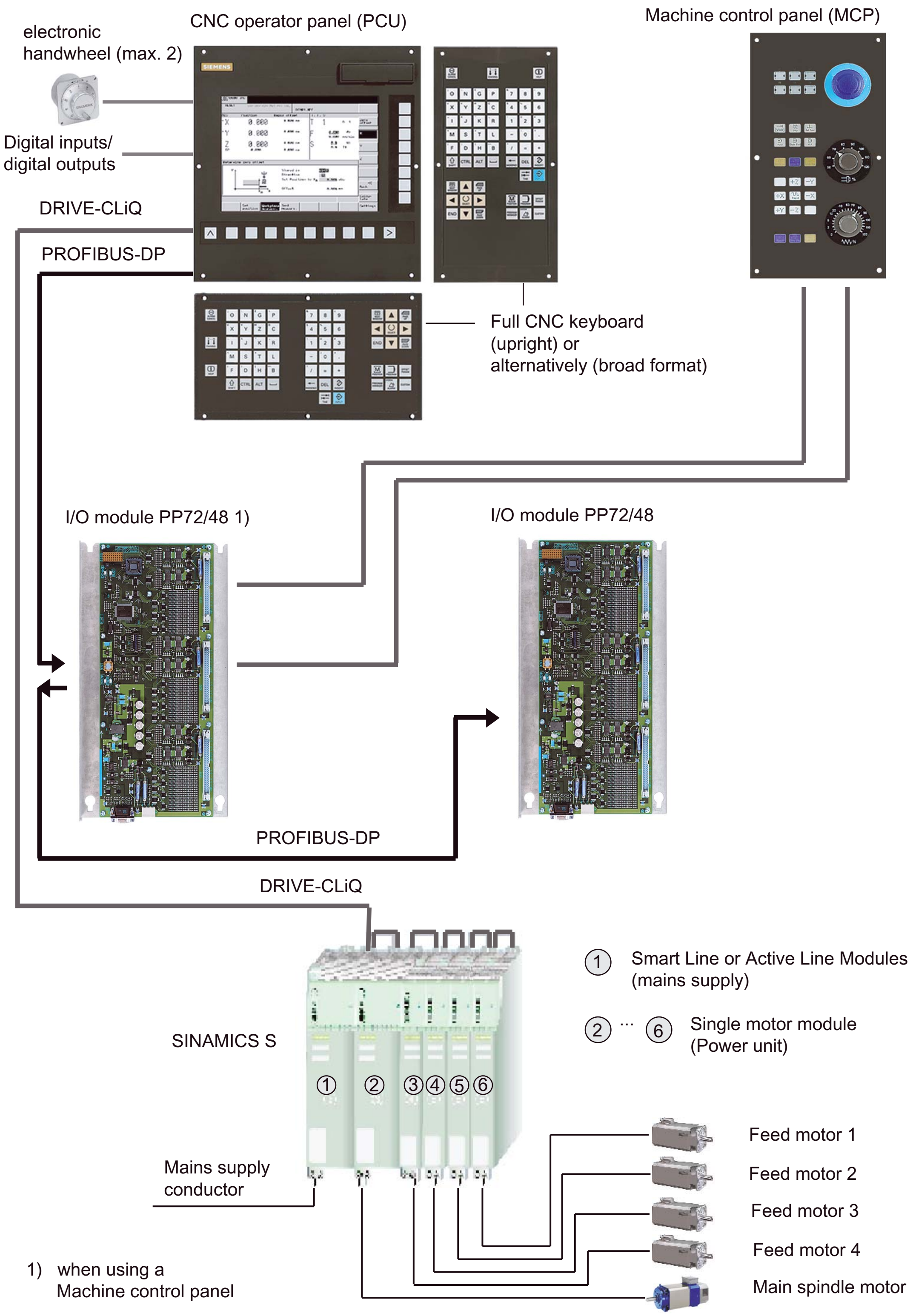


Figure 1-1 SINUMERIK 802D sl with SINAMICS S120 (example configuration)

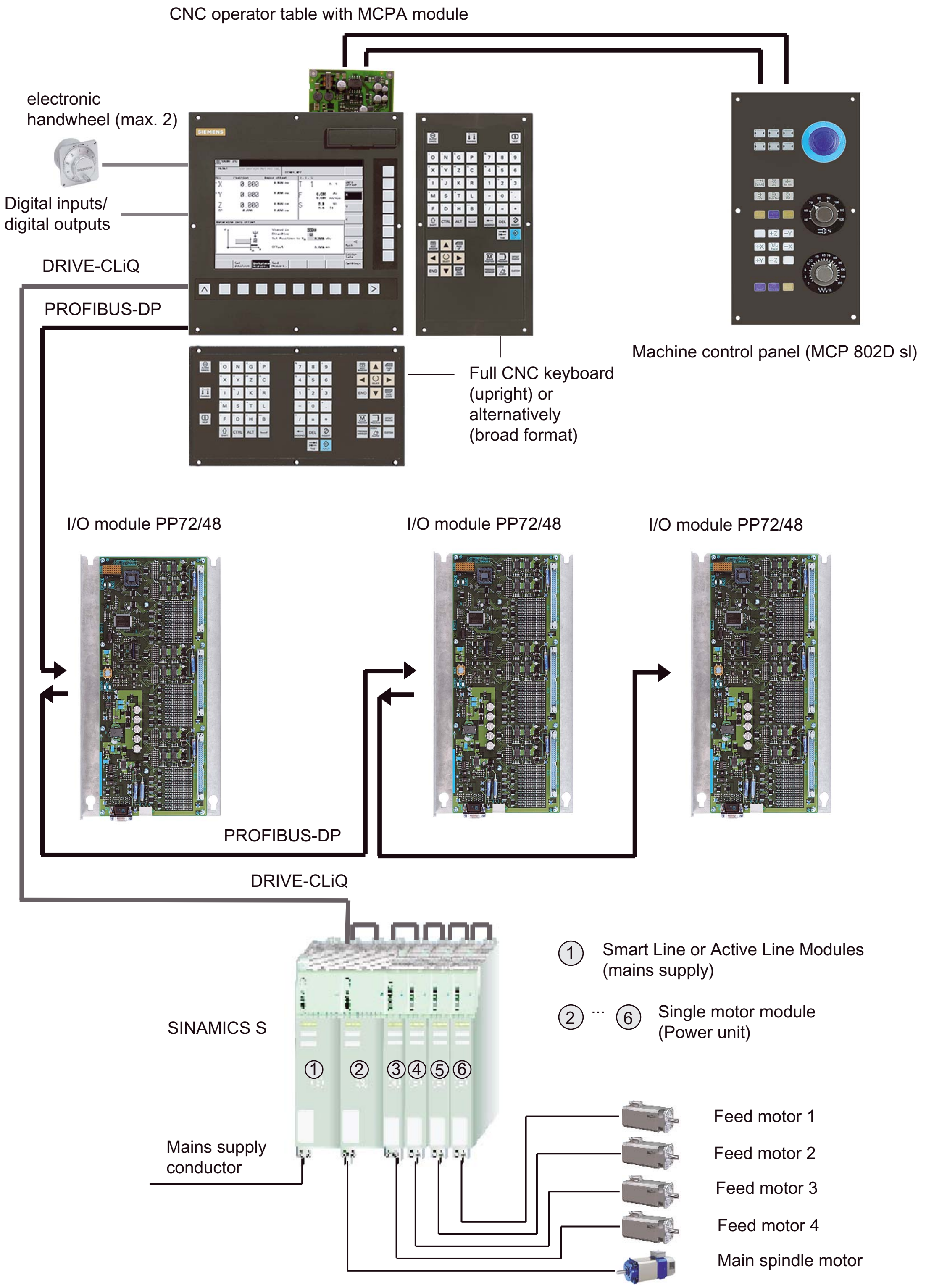


Figure 1-2 SINUMERIK 802D sl with MCPA module (example configuration)

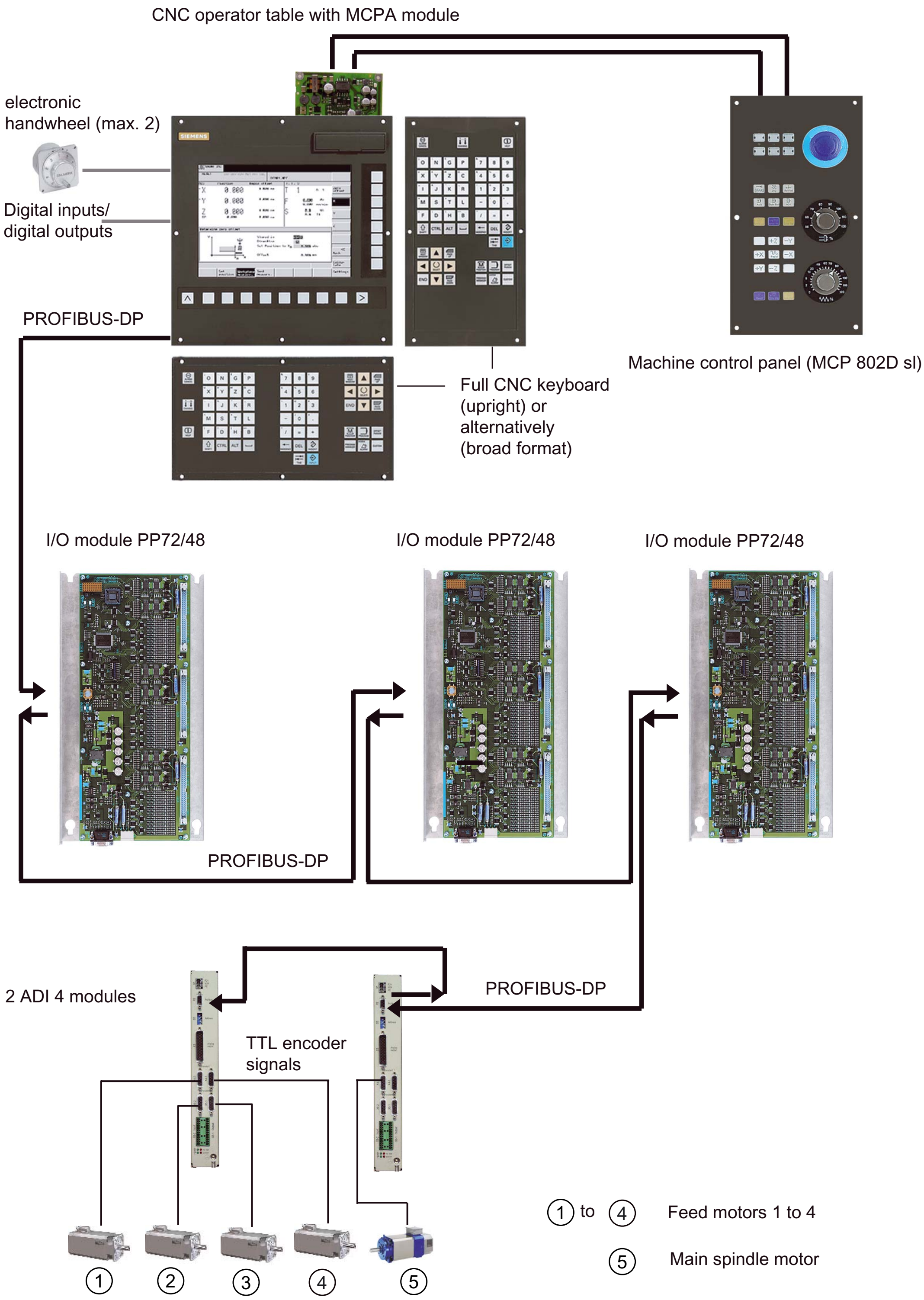


Figure 1-3 System overview with ADI4 (example configuration)

Components

The components of the SINUMERIK 802D sl control system are:

- CNC operator panel (PCU)
- Full CNC keyboard (vertical or horizontal format)
- MCP machine control panel

  Incorporates all keys and switches required for the operation of a machine The machine control panel is available in 2 versions:

  – Machine control panel MCP to connect via a PP 72/48 I/O module

  – Machine control panel MCP 802D sl to connect via an MCPA module
- MCPA module (hardware optional)

  The MCPA module is a supplemental/expansion module of the SINUMERIK 802D sl. It places the following resources at your disposal:

  – Analog output for ± 10 V (X701) for connecting an analog spindle

  – Interface for connecting an MCP 802D sl external machine control panel (X1, X2)

  – Interface for connecting inputs and outputs (1 bytes each) in the form of high-speed inputs/outputs.
- PP72/48 I/O module

  The PP72/48 I/O module is a user-friendly and low-cost module (without a separate housing) for connecting digital inputs/outputs within the framework of an automation system based on PROFIBUS-DP.

  The module has the following important features:

  – Peripherals interface for connecting an external machine control panel (MCP) or a terminal strip converter for the digital inputs and outputs (X111, X222, X333)

  – PROFIBUS-DP connection (max 12 Mbits/s)

  – 72 digital inputs and 48 digital outputs

  – On-board status display via four diagnostic LEDs

  To supply the module and the digital outputs, an external voltage source (+24 V DC) is required.
- ADI4 (analog drive interface for 4 axes)

  The ADI4 module facilitates an analog drive interface for 4 axes/spindles, with or without an incremental encoder.
- Drive units

  – SINAMICS S120

    Communication between the SINUMERIK 802D sl control system and the SINAMICS S 120 drive takes place via the DRIVE-CLiQ communication system (Drive Component Link with IQ).

- **DRIVE-CLiQ Hub Module DMC20**

  The DRIVE-CLiQ Hub Module DMC20 is used to implement point-to-point distribution of a DRIVE-CLiQ line and to enable direct measuring systems to be employed.

  The component is ideal for applications which require DRIVE-CLiQ nodes to be removed in groups, without interrupting the DRIVE-CLiQ line and, therefore, data exchange.

- **Sensor Module Cabinet-Mounted SMC10/SMC20/SMC30**

  The Sensor Module Cabinet-Mounted SMC10/SMC20/SMC30 evaluates sensor signals and transmits the speed, actual position value, rotor position and, if necessary, the motor temperature and home position via DRIVE-CLiQ to the Control Unit.

  The SMC10 is used to evaluate sensor signals from resolvers.

System software

The following system software is installed in the retentive internal memory of the PCU of each SINUMERIK 802D sl by default:

- **Boot software** - starts the system
- **Human Machine Interface (HMI) software** - realizes all operator functions
- **NCK software (NC Kernel)** - realizes all NC functions.
- **Programmable Logic Control (PLC) software** - executes the integrated PLC user program cyclically.
- **Drive software** - implements drive control

Toolbox

A tool box is delivered on CD ROM together with the appropriate system software.

The toolbox contains software tools for configuring the control system. It must be installed on your PC/PG.

The following software can be found in the Toolbox:

- Configuration data for the SINUMERIK 802D sl:
  - Setup files for the technologies
  - Cycle packages for the technologies
  - Retroloadable languages
- SIMATIC Automation License Manager

  The Automation License Manager is needed for managing license keys (e.g. for RCS802).
- RCS802 Commissioning and diagnostic tool (must be licensed for Ethernet and remote control function)

  This program can be used to transfer texts, user data and programs from the PC to the CNC operator panel (PCU) and vice versa.

- PLC 802 programming tool

  Tool to create PLC user program
- PLC user library

  PLC sample programs
- Startup Tool

  Tool for commissioning, in particular for drive optimization

  This tool is an export from HMI Advanced.

Note

The table of contents and notes for setup can be found in the siemense.txt file.

STARTER

The STARTER commissioning tool is provided on DVD along with the relevant system software.

References

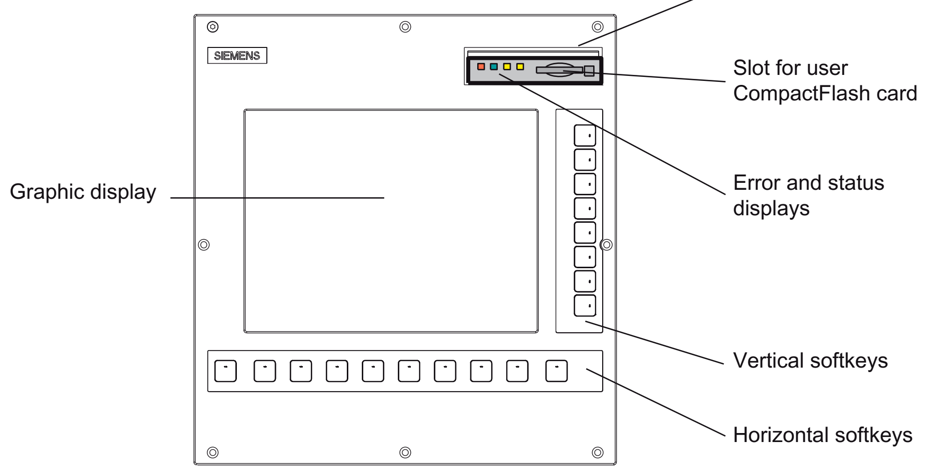
SINAMICS S120 Getting Started with the STARTER Commissioning Tool

1.2 Description of components

View

The illustration below shows the CNC operator panel (PCU) with its interfaces and front panel elements.

Front view of the CNC operator panel (PCU)



Rear view of the CNC operator panel (PCU)

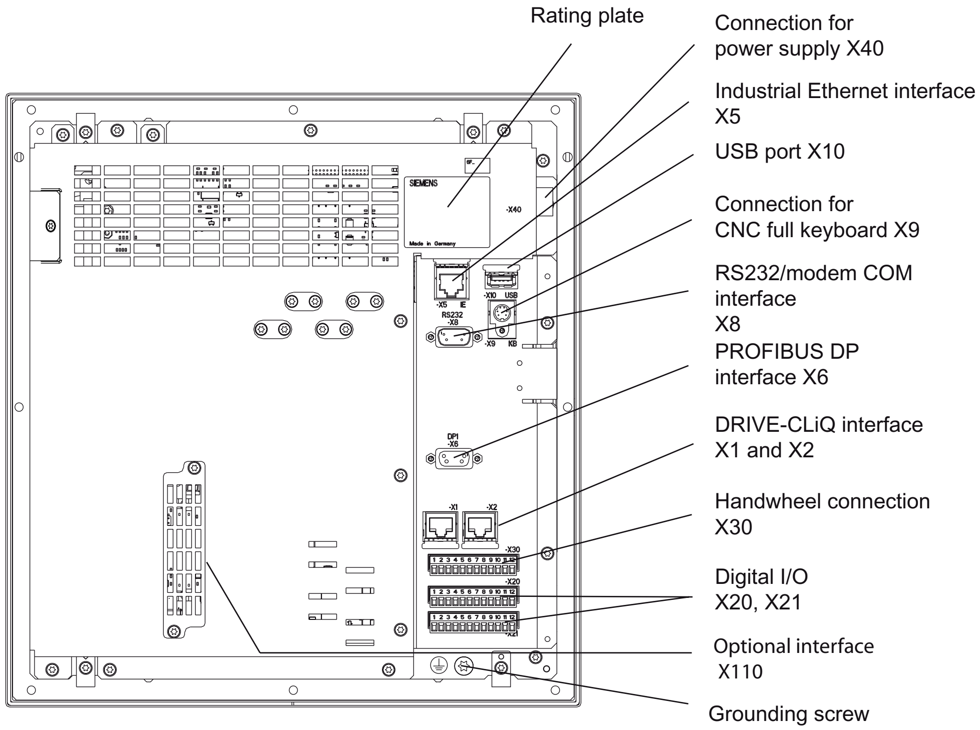


Figure 1-4 Position of the interfaces and front elements on the CNC operator panel (PCU)

CNC operator panel (PCU) interfaces

The PCU and its functions are described in the table below.

Table 1- 1 PCU interfaces

Interfaces	Function
Slot for the CompactFlash Card, front view	50-pin slot for user CompactFlash Card and 4 LEDs
Power supply connection X40	3-pin screw-type terminal connection for connecting the 24 V load power supply
Ethernet interface X5	8-pin RJ45 socket connector for connection to an Industrial Ethernet
USB port X10	4-pin USB port for connecting USB accessories
Full CNC keyboard connection X9	6-pin PS/2 socket for connecting the full CNC keyboard
RS232 COM port X8	9-pin Sub-D connector for connecting a PG/PC
PROFIBUS-DP interface X6	9-pin sub D socket for connection to PROFIBUS DP
DRIVE-CLiQ interface X1 and X2	8-pin RJ45 socket for connecting the SINAMICS S120 drive
Handwheel connection X30	12-pin screw-type male connector for connecting a max. of 2 handwheels
Digital inputs/digital outputs X20 and X21	12-pin screw-type male connector for connecting the digital inputs and outputs
Option interface X110	48-pin female connector for connecting the MCPA module

Error and status displays

See Chapter"Error and status displays" (Page 102)

2 interfaces

2.1 CNC operator panel interfaces

2.1.1 Slot for CompactFlash card

Overview

The slot for the optional user CompactFlash Card is located at the front behind the protective flap.

CompactFlash Card for user data

You can use the CompactFlash card, for example:

- For commissioning data
- For NC programs
- To carry out software updates
- For storing user data
- For saving set parameters

Notes on the CompactFlash Card for user data

- The SINUMERIK CNC supports the file systems FAT16 and FAT32 for CompactFlash cards. You may need to format the memory card if you want to use a memory card from another device or if you want to ensure the compatibility of the memory card with the SINUMERIK. However, formatting the memory card will permanently delete all data on it.
- Do not remove the memory card while it is being accessed. This can lead to damage of the memory card and the SINUMERIK as well as the data on the memory card.

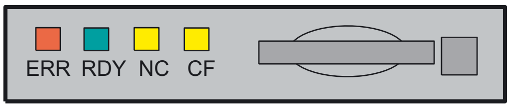


Figure 2-1 LEDs on the CNC operator panel (PCU)

Note

The yellow "CF" LED on the right flashes when the memory card is accessed.

- If you cannot use a memory card with the SINUMERIK, it is probably because the memory card is not formatted for the control system (e.g. Ext3 Linux file system), the memory card file system is faulty, or it is the wrong type of memory card.
- Insert the memory card carefully with the correct orientation into the memory card slot (observe indicators such as arrow or similar). This way you avoid mechanical damage to the memory card or the device.
- Only use memory cards that have been approved by Siemens for use with SINUMERIK. Even though the SINUMERIK keeps to the general industry standards for memory cards, it is possible that memory cards from some manufacturers will not function perfectly in this device or are not completely compatible with it (you can obtain information on compatibility from the memory card manufacturer or supplier).
- For the CompactFlash Card for user data of the SINUMERIK 802D sl, only the empty memory card 1 GB, Smart Modular, with order number 6FC5313-5AG00-0AA1 is permitted.

2.1.2 Ethernet interface

The following versions can be connected to the Ethernet interface X5 via an Industrial Ethernet network.

- Ethernet peer-to-peer to a PG/PC
- Ethernet network to a company network

Industrial Ethernet is a communication network with a transmission rate of 10/100 Mbit/s.

Cable type

Industrial Ethernet cable (CAT5):

- IE TP XP cord (crossed TP cable) for a max. length of 10 m (e.g. Ethernet peer-to-peer)
- IE FC TP for a max. length of 100 m

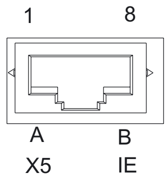
References

Catalog NC 802D sl, Catalog NC 61, Catalog IK PI

Female connector pin assignment

Designation: X5 (IE)
Type: 8-pin RJ45 socket

Table 2- 1 Pin assignment of female connector X5

Schematic view of the connector, mounting position and labeling	Pin	Name	Description
	1	TXP	Transmit data +
	2	TXN	Transmit data -
	3	RXP	Receive data +
	4	not assigned	-
	5	not assigned	-
	6	RXN	Receive data -
	7	not assigned	-
	8	not assigned	-
You can obtain additional information about the different cable systems for Ethernet from your SIEMENS contact.			

2.1.3 USB port

Note

A USB FlashDrive can be connected to this port.

The port uses driver SW 1.1.

The machine OEM can adapt this port (by extending the cable from the control system housing).

References

Catalog NC 802D sl, Catalog NC 61

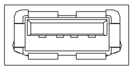
Note

The USB interface is **only** approved for a USB-FlashDrive.

Female connector pin assignment

Designation: X10(USB)
Type: 4 pin USB host

Table 2- 2 Female connector pin assignment X10

Schematic view of the connector, mounting position and labeling	Pin	Name	Description
X10 USB 	A1	P5_USB	5 V supply voltage
	A2	DN_USB	USB data - (Channel 0)
	A3	DP_USB	USB data + (Channel 0)
	A4	M	Ground

Note

The USB interface has a current carrying capacity of 0.5 A.

2.1.4 RS232 COM port

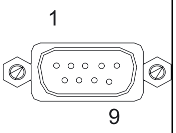
A PC/PG for data exchange with the CNC operator panel can be connected to male connector X8.

Connector pin assignment

Identifiers: X8 (RS232)
Type: 9-pin Sub-D terminal strip

Table 2- 3 Pin assignment of connector X8

Schematic view of the connector, mounting position and labeling	Pin	Name	Description German/English:	
	1	DCD	Received Line Signal Detector Carrier Detector	Data carrier detect
	2	RxD	Received Data	Received data
	3	TxD	Transmitted Data	Transmitted data
	4	DTR	Data Terminal Ready	Data Terminal Ready
	5	G	Ground	Ground
	6	DSR	Data Set Ready	Data Set Ready
	7	RTS	Request To Send	Transmission request
	8	CTS	Clear To Send	Ready to send

Schematic view of the connector, mounting position and labeling	Pin	Name	Description German/English:	
RS232 X8 	9	Not assigned	-	-

2.1.5 PROFIBUS-DP Interface

The CNC operator panel (PCU) communicates with the I/O modules via the PROFIBUS-DP interface.

The PROFIBUS DP protocol is used for communication.

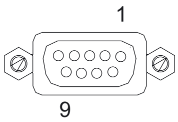
The baud rate of the PROFIBUS-DP interface is 12 Mbit/s; the baud rate cannot be changed. Converters for optical fiber cable (OLMs, OLPs) or repeaters are not permitted.

The CNC operator panel functions as a master.

Female connector pin assignment

Designation: X6 (DP1)
Type: 9-pin Sub-D socket connector

Table 2- 4 Pin assignment of socket X6

Schematic view of the female connector, mounting position and labeling	Pin	Name	Description
DP1 X6 	1	not assigned	-
	2	M24	
	3	B	Data input/output (RS485)
	4	RTS	Transmission request
	5	M5	5 V reference potential
	6	P5	5 V power supply 90 mA, short-circuit-proof
	7	P24	24V power supply (teleservice) 150mA, short-circuit-proof, not isolated
	8	A	Data input/output (RS485)
	9	Not assigned	-

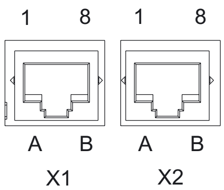
2.1.6 DRIVE-CLiQ interface

The CNC operator panel (PCU) can communicate with the "SINAMICS S" drive via the DRIVE-CLiQ interface.

Female connector pin assignment

Designation: **X1, X2**
Type: 8-pin RJ45 socket

Table 2- 5 Pin assignment of female connector X1 and X2

Schematic view of the female connector, mounting position and labeling	Pin	Name	Description
	1	TXP	Transmit data +
	2	TXN	Transmit data -
	3	RXP	Receive data +
	4	not assigned	-
	5	not assigned	-
	6	RXN	Receive data -
	7	not assigned	-
	8	not assigned	-
	A	not assigned	-
	B	not assigned	-
Blanking plate for DRIVE-CLiQ interface: Molex corp., order no. 85999-3255			

2.1.7 Handwheel connection

Up to 2 electronic handwheels can be connected to connector X30 on the CNC operator panel (PCU).

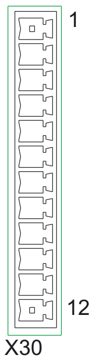
The handwheel must meet the following requirements:

Transmission procedure:	5 V square wave signals (TTL level or RS422)
Signals:	Track A as a true and negated signal (U_{a1}, U_{a1}) Track B as a true and negated signal (U_{a2}, U_{a2})
Max. output frequency:	500 kHz
Phase shift	of Track A to Track B: 90° ±30°
Supply:	5 V, max. 250 mA

Connector pin assignment

Designation: X30
Type: 12-pin connector

Table 2- 6 Pin assignment of connector X30

Schematic view of the connector	Pin	Name	Description
	1	3P5	5VDC supply voltage
	2	G	Ground
	3	1A	Track A, handwheel 1
	4	X1A	Track A_N, handwheel 1
	5	1B	Track B, handwheel 1
	6	X1B	Track B_N, handwheel 1
	7	3P5	5VDC supply voltage
	8	G	Ground
	9	2A	Track A, handwheel 2
	10	X2A	Track A_N, handwheel 2
	11	2B	Track B, handwheel 2
	12	X2B	Track B_N, handwheel 2

2.1.8 Digital inputs/outputs

The fast digital inputs and digital outputs at the X20 and X21 connectors are reserved for SINAMICS drives.

A maximum of 16 digital inputs or a maximum of 8 digital and 8 bidirectional inputs/outputs can be used.

See also

Connection overview for SINUMERIK 802D sl (Page 79)

Terminal assignment and block diagram

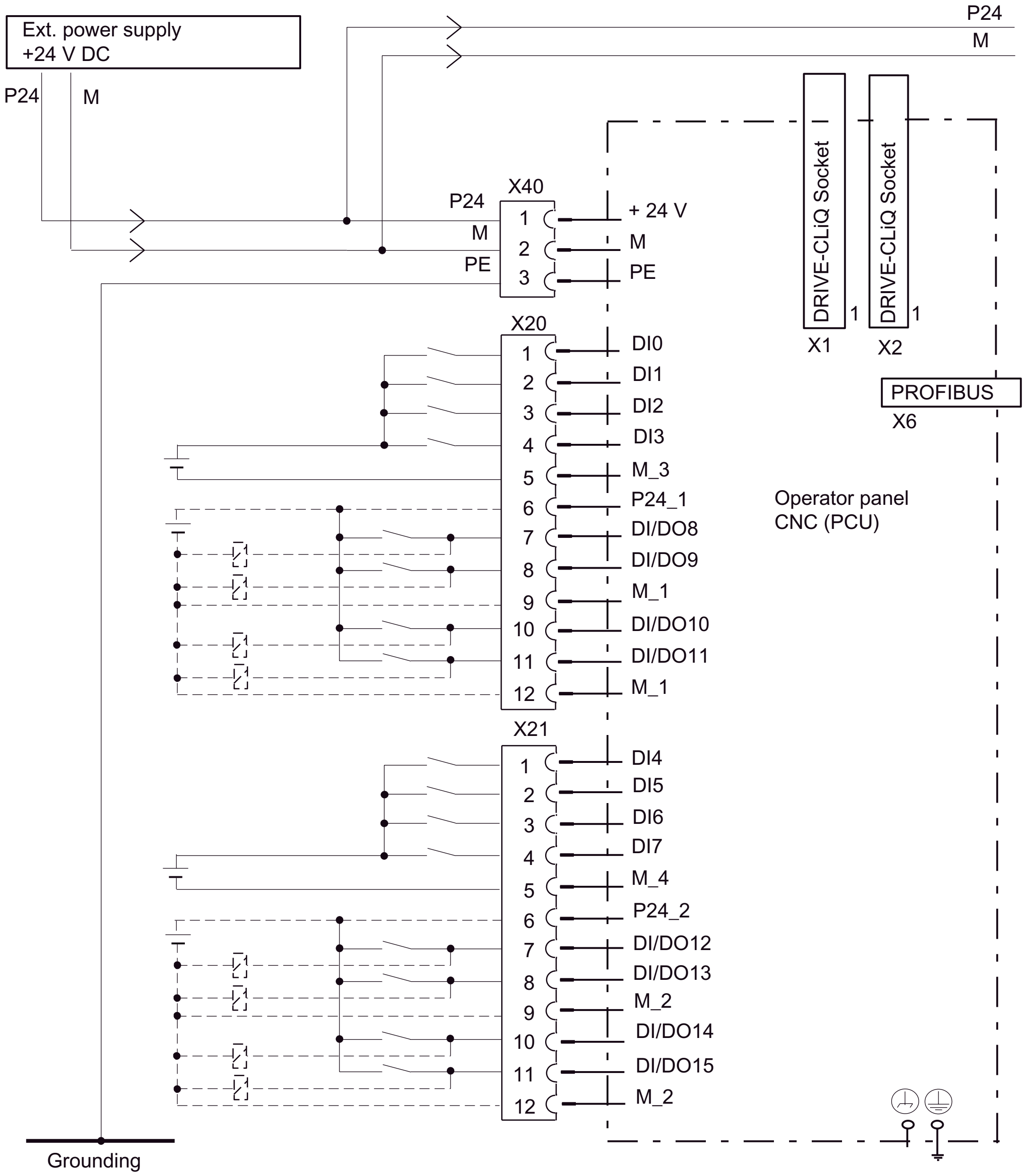
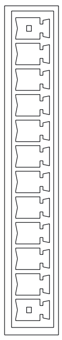
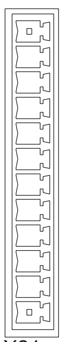


Figure 2-2 Connection example

Connector pin assignment

Designation: X20, X21
Type: 12-pin connector

Table 2- 7 Pin assignment of the connectors X20 and X21

Representation	Pin	Name	Description	Technical details
	1	DI0	Digital input 0	**Input:** Voltage: 24 V DC (20.4...28.8 V) Level: 0 signal: -3...5 V 1-signal: 11...30 V Input delay: 0 → 1 signal: 15 µs (typically 6) 1 → 0 signal: 150 µs (typically 40)
	2	DI1	Digital input 1	
	3	DI2	Digital input 2	
	4	DI3	Digital input 3	
	5	M_3	Ground for DI0...DI3	
	6	P24_1	24 V DC supply voltage for DI/DO8...DI/DO11 (required for digital outputs)	**As output:** max. output current: 1 signal: 5 mA...0.5 A Total current of all outputs: max. 2 A (in case of 50% simultaneity) Output delay: 0 → 1 signal: 500 µs (typically 150 µs) 1 → 0 signal: 500 µs (typ. 150 µs) each for RL = 60 Ohms switching frequency: 100 Hz (ohmic load) 2 Hz (inductive load) **As input:** Data, see connector X21
	7	DI/DO8	Digital input/digital output	
	8	DI/DO9	Digital input/digital output	
	9	M_1	Ground for DI/DO8...DI/DO11	
	10	DI/DO10	Digital input/digital output	
	11	DI/DO11	Digital input/digital output	
	12	M_1	Ground for DI/DO8...DI/DO11	
	1	DI4	Digital input 4	**Input:** Data, see connector X20
	2	DI5	Digital input 5	
	3	DI6	Digital input 6	
	4	DI7	Digital input 7	
	5	M_4	Ground for DI4...DI7	
	6	P24_2	24 V DC supply voltage for DI/DO12...DI/DO15 (required for digital outputs)	**As output:** Data, see connector X20 **As input:** Voltage: 24 V DC (20.4...28.8 V) Level: 0 signal: -3...5 V 1 signal: 11...30 V Input delay: 0 → 1 signal: 15 µs (typically 6) 1 → 0 signal: 150 µs (typically 40)
	7	DI/DO12	Digital input/digital output	
	8	DI/DO13	Digital input/digital output	
	9	M_2	Ground for DI/DO12...DI/DO15	

Representation	Pin	Name	Description	Technical details
	10	DI/DO14	Digital input/digital output	
	11	DI/DO15	Digital input/digital output	
	12	M_2	Ground for DI/DO12...DI/DO15	

⚠ DANGER

Separation according to EN60204-1

The 24 V power supply must be a functional extra low voltage supply with protective separation in accordance with EN60204-1, Section 6.4, PELV (with M ground).

Digital inputs (PCU)

The inputs comply with standard IEC 1131-2/DIN EN 61131-2, characteristic type 2 (24 V P switching). Switches or proximity encoders (2- or 3-wire encoders) can be connected.

Digital outputs (PCU)

The outputs comply with standard IEC 1131-2/DIN EN 61131-2 (24 V P switching).

See also

Setting the PROFIBUS addresses (Page 128)

2.2 MCP machine control panel interfaces

The following machine control panels are available:

- MCP 802D sl machine control panel
- MCP machine control panel

The table below describes which modules are used with each of these machine control panels whenever they are connected:

Table 2- 8 Connecting an MCP

Modules	MCP 802D sl	MCP
MCPA	Yes	No
PP72/48	No	Yes

2.2.1 Interfaces of the machine control panel MCP 802D sl

The illustration below shows the back of the machine control panel MCP 802D sl with its interfaces.

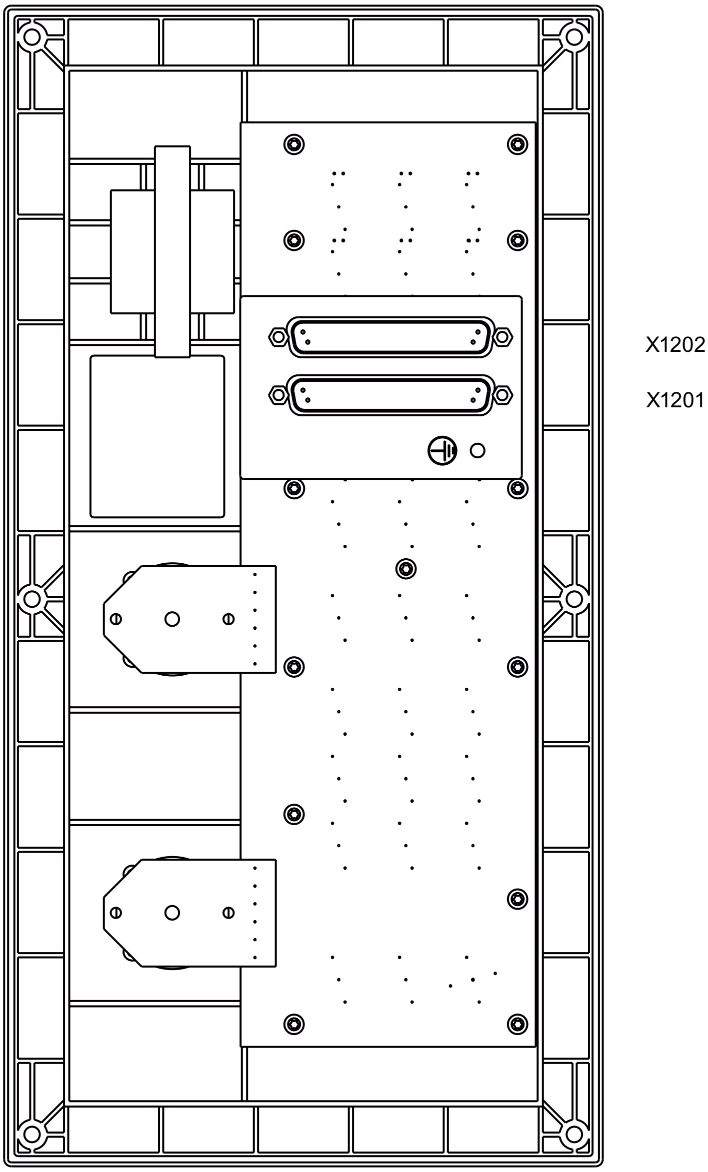


Figure 2-3 Interfaces at the MCP 802D sl

Table 2- 9 Interfaces

Interfaces	Function
Interface **X1201**	40-pin sub D connector for connecting the machine control panel MCP 802D sl to the MCPA module **X1**
Interface **X1202**	40-pin sub D connector for connecting the machine control panel MCP 802D sl to the MCPA module **X2**

Interface assignments

Designation: X1201, X1202
Type: 40-pin D-Sub connector

Table 2- 10 Pin assignment of the connectors X1201 and X1202

X1201					
Pin	Name	Description	Pin	Name	Description
1	KEY1	Input bit	2	KEY2	Input bit
3	KEY3	Input bit	4	KEY4	Input bit
5	KEY5	Input bit	6	KEY6	Input bit
7	KEY7	Input bit	8	KEY8	Input bit
9	GND		10	KEY9	Input bit
11	KEY10	Input bit	12	KEY11	Input bit
13	KEY12	Input bit	14	KEY13	Input bit
15	KEY14	Input bit	16	KEY15	Input bit
17	KEY16	Input bit	18	GND	
19	KEY17	Input bit	20	KEY18	Input bit
21	KEY19	Input bit	22	KEY20	Input bit
23	KEY21	Input bit	24	KEY22	Input bit
25	KEY23	Input bit	26	KEY24	Input bit
27	GND		28	LED1	Output bit
29	LED2	Output bit	30	LED3	Output bit
31	LED4	Output bit	32	LED5	Output bit
33	LED6	Output bit	34	not assigned	-
35	not assigned	-	36	GND	
37	not assigned	-	38	not assigned	-
39	not assigned	-	40	not assigned	-

X1202					
1	KEY25	Input bit	2	KEY26	Input bit
3	KEY27	Input bit	4	not assigned	-
5	not assigned	-	6	not assigned	-
7	not assigned	-	8	not assigned	-
9	GND		10	FEED_OV_A	Input bit
11	FEED_OV_B	Input bit	12	FEED_OV_C	Input bit
13	FEED_OV_D	Input bit	14	FEED_OV_E	Input bit
15	not assigned	-	16	not assigned	-
17	not assigned	-	18	GND	
19	SPINDLE_OV_A	Input bit	20	SPINDLE_OV_B	Input bit
21	SPINDLE_OV_C	Input bit	22	SPINDLE_OV_D	Input bit
23	SPINDLE_OV_E	Input bit	24	not assigned	-

X1202					
25	not assigned	-	26	not assigned	-
27	not assigned	-	28	not assigned	-
29	not assigned	-	30	not assigned	-
31	not assigned	-	32	not assigned	-
33	not assigned	-	34	not assigned	-
35	not assigned	-	36	GND	
37	not assigned	-	38	not assigned	-
39	not assigned	-	40	not assigned	-

2.2.2 Interfaces of the machine control panel MCP

The illustration below shows the back of the machine control panel MCP with its interfaces.

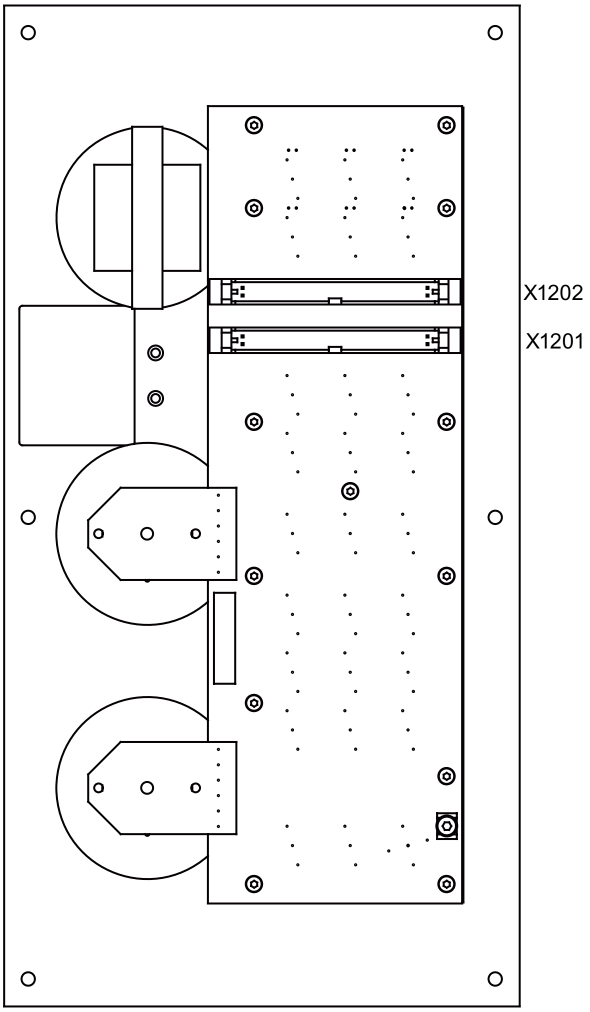


Figure 2-4 Interfaces on the MCP

Table 2- 11 Interfaces

Interfaces	Function
Interface X1201	50-pin ribbon-cable plug for connecting the machine control panel MCP to the PP module
Interface X1202	50-pin ribbon-cable plug for connecting the machine control panel MCP to the PP module

Interface assignments

Designation: X1201, X1202
Type: 50-pin ribbon cable connector

Table 2- 12 Pin assignment of the connectors X1201 and X1202

X1201					
Pin	Name	Description	Pin	Name	Description
1	GND		2	+24V	
3	KEY1	Input bit	4	KEY2	Input bit
5	KEY3	Input bit	6	KEY4	Input bit
7	KEY5	Input bit	8	KEY6	Input bit
9	KEY7	Input bit	10	KEY8	Input bit
11	KEY9	Input bit	12	KEY10	Input bit
13	KEY11	Input bit	14	KEY12	Input bit
15	KEY13	Input bit	16	KEY14	Input bit
17	KEY15	Input bit	18	KEY16	Input bit
19	KEY17	Input bit	20	KEY18	Input bit
21	KEY19	Input bit	22	KEY20	Input bit
23	KEY21	Input bit	24	KEY22	Input bit
25	KEY23	Input bit	26	KEY24	Input bit
27	not assigned	-	28	not assigned	-
29	not assigned	-	30	not assigned	-
31	LED1	Output bit	32	LED2	Output bit
33	LED3	Output bit	34	LED4	Output bit
35	LED5	Output bit	36	LED6	Output bit
37		Output bit	38		Output bit
39		Output bit	40		Output bit
41		Output bit	42		Output bit
43		Output bit	44		Output bit
45		Output bit	46		Output bit
47	24VDC	24 V DC	48	24VDC	24 V DC
49	24VDC	24 V DC	50	24VDC	24 V DC

X1202					
Pin	**Name**	**Description**	**Pin**	**Name**	**Description**
1	GND		2	+24V	
3	KEY25	Input bit	4	KEY26	Input bit
5	KEY27	Input bit	6		Input bit
7		Input bit	8		Input bit
9		Input bit	10		Input bit
11	Feed_OV_A	Input bit	12	Feed_OV_B	Input bit
13	Feed_OV_C	Input bit	14	Feed_OV_D	Input bit
15	Feed_OV_E	Input bit	16		Input bit
17		Input bit	18		Input bit
19	Sp-OV-A	Input bit	20	Sp-OV-B	Input bit
21	Sp-OV-C	Input bit	22	Sp-OV-D	Input bit
23	Sp-OV-E	Input bit	24		Input bit
25		Input bit	26		Input bit
27	not assigned	-	28	not assigned	-
29	not assigned	-	30	not assigned	-
31		Output bit	32		Output bit
33		Output bit	34		Output bit
35		Output bit	36		Output bit
37		Output bit	38		Output bit
39		Output bit	40		Output bit
41		Output bit	42		Output bit
43		Output bit	44		Output bit
45		Output bit	46		Output bit
47	24VDC	24 V DC	48	24VDC	24 V DC
49	24VDC	24 V DC	50	24VDC	24 V DC

2.3 MCPA module interfaces

Overview

The illustration below shows the MCPA module and its interfaces.

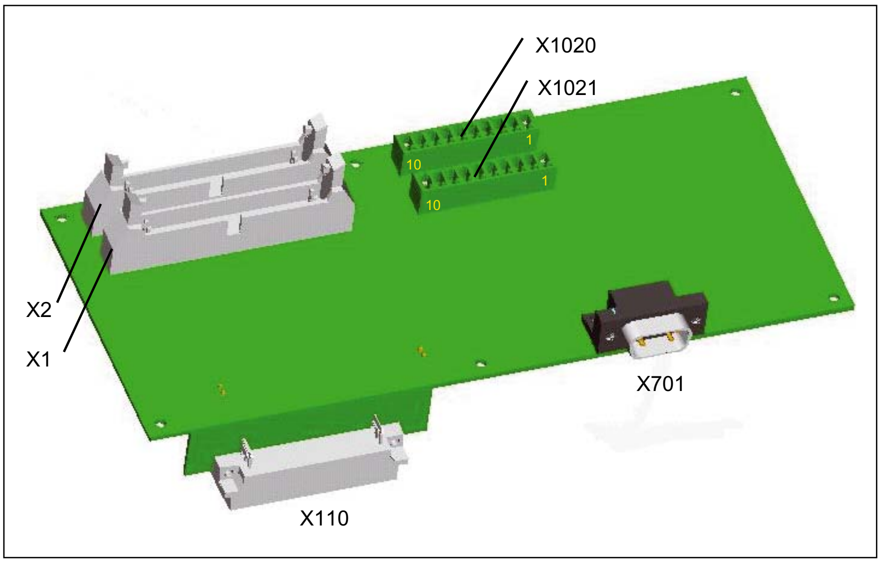


Figure 2-5 Position of the interfaces on the MCPA module

Table 2- 13 Interfaces

Interfaces	Function
X1 and X2	40-pin plug connectors for connecting the machine control panel MCP 802D sl
X1020 and X1021	10-pin plug connectors for connection of the power supply and of the high-speed digital inputs and outputs (I/O)
X701	9-pin D-Sub connector for connecting an analog spindle with directly mounted spindle actual-value encoder
X110	48-pin plug connectors for connecting the MCPA module to the PCU

Assignment of the interface to the MCP 802D sl

Designation: X1, X2
Type: 40-pin ribbon cable connector

Table 2- 14 Pin assignment of connectors X1 and X2

X1					
Pin	Name	Description	Pin	Name	Description
1	KEY1	Input bit	2	KEY2	Input bit
3	KEY3	Input bit	4	KEY4	Input bit
5	KEY5	Input bit	6	KEY6	Input bit

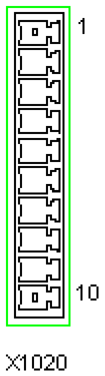
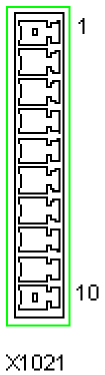
X1					
Pin	Name	Description	Pin	Name	Description
7	KEY7	Input bit	8	KEY8	Input bit
9	GND		10	KEY9	Input bit
11	KEY10	Input bit	12	KEY11	Input bit
13	KEY12	Input bit	14	KEY13	Input bit
15	KEY14	Input bit	16	KEY15	Input bit
17	KEY16	Input bit	18	GND	
19	KEY17	Input bit	20	KEY18	Input bit
21	KEY19	Input bit	22	KEY20	Input bit
23	KEY21	Input bit	24	KEY22	Input bit
25	KEY23	Input bit	26	KEY24	Input bit
27	GND		28	LED1	Output bit
29	LED2	Output bit	30	LED3	Output bit
31	LED4	Output bit	32	LED5	Output bit
33	LED6	Output bit	34	not assigned	-
35	not assigned	-	36	GND	
37	not assigned	-	38	not assigned	-
39	not assigned	-	40	not assigned	-

X2					
1	KEY25	Input bit	2	KEY26	Input bit
3	KEY27	Input bit	4	not assigned	-
5	not assigned	-	6	not assigned	-
7	not assigned	-	8	not assigned	-
9	GND		10	FEED_OV_A	Input bit
11	FEED_OV_B	Input bit	12	FEED_OV_C	Input bit
13	FEED_OV_D	Input bit	14	FEED_OV_E	Input bit
15	not assigned	-	16	not assigned	-
17	not assigned	-	18	GND	
19	SPINDLE_OV_A	Input bit	20	SPINDLE_OV_B	Input bit
21	SPINDLE_OV_C	Input bit	22	SPINDLE_OV_D	Input bit
23	SPINDLE_OV_E	Input bit	24	not assigned	-
25	not assigned	-	26	not assigned	-
27	not assigned	-	28	not assigned	-
29	not assigned	-	30	not assigned	-
31	not assigned	-	32	not assigned	-
33	not assigned	-	34	not assigned	-
35	not assigned	-	36	GND	
37	not assigned	-	38	not assigned	-
39	not assigned	-	40	not assigned	-

Assignment of the I/O interface connectors

Designation: **X1020, X1021**
Type: 10-pin connector

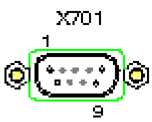
Table 2- 15 Pin assignment of the connectors X1020 and X1021

Representation	Pin	Name	Description	Technical details
	1			**Input:** Voltage: 24 V DC (20.4...28.8 V) Level: 0 signal: -3...5 V 1 signal: 11...30 V
	2	DI9	high-speed digital input 9	
	3	DI10	high-speed digital input 10	
	4	DI11	high-speed digital input 11	
	5	DI12	high-speed digital input 12	
	6	DI13	high-speed digital input 13	
	7	DI14	high-speed digital input 14	
	8	DI15	high-speed digital input 15	
	9	DI16	high-speed digital input 16	
	10	M	Frame ground	
	1	P24	24 V DC supply voltage	**As output:** Max. output current: 1 signal: 5 mA...0.5 A Total current of outputs: max. 2 A
	2	DO9	high-speed digital output 9	
	3	DO10	high-speed digital output 10	
	4	DO11	high-speed digital output 11	
	5	DO12	high-speed digital output 12	
	6	DO13	high-speed digital output 13	
	7	DO14	high-speed digital output 14	
	8	DO15	high-speed digital output 15	
	9	DO16	high-speed digital output 16	
	10	M	Frame ground	

Connector pin assignment (analog output to the drive)

Designation: **X701**
Type: 9-pin Sub-D terminal strip

Table 2- 16 Pin assignment of connector X701

Schematic view of the connector, mounting position and labeling	Pin	Name	Description German
	1	Analog OUT	Analog output with a signal level of ±10 V Resolution 11 bits + sign
	2	not assigned	-
	3	Uni-Dir2	Digital output for unipolar spindle +24 V
	4	Uni-Dir1	Digital output for unipolar spindle +24 V

Schematic view of the connector, mounting position and labeling	Pin	Name	Description German
	5	Enable 1-	Analog drive enable (contact: electrically isolated n.o. contact)
	6	Analog OUT	Analog output 0 V Reference signal
	7	not assigned	-
	8	not assigned	-
	9	Enable 2-	Analog drive enable (contact: electrically isolated n.o. contact)

Note

The controller enable for the analog drive should be connected via the relay contact (connector X701; pin 5 and pin 9) of the MCPA module. The controller enable signal is controlled via the PLC interface.

References:

SINUMERIK 802D sl List Manual; signals to the axis/spindle

SINUMERIK 802D sl Function Manual; Various Interface Signals (A2)

See also

Starting Up the PLC (Page 131)

2.4 PP72/48 I/O module interfaces

The diagrams below show the interfaces, the operator controls and status displays, (Page 102) and (in the example) the options for connecting to the I/O interface of the I/O module.

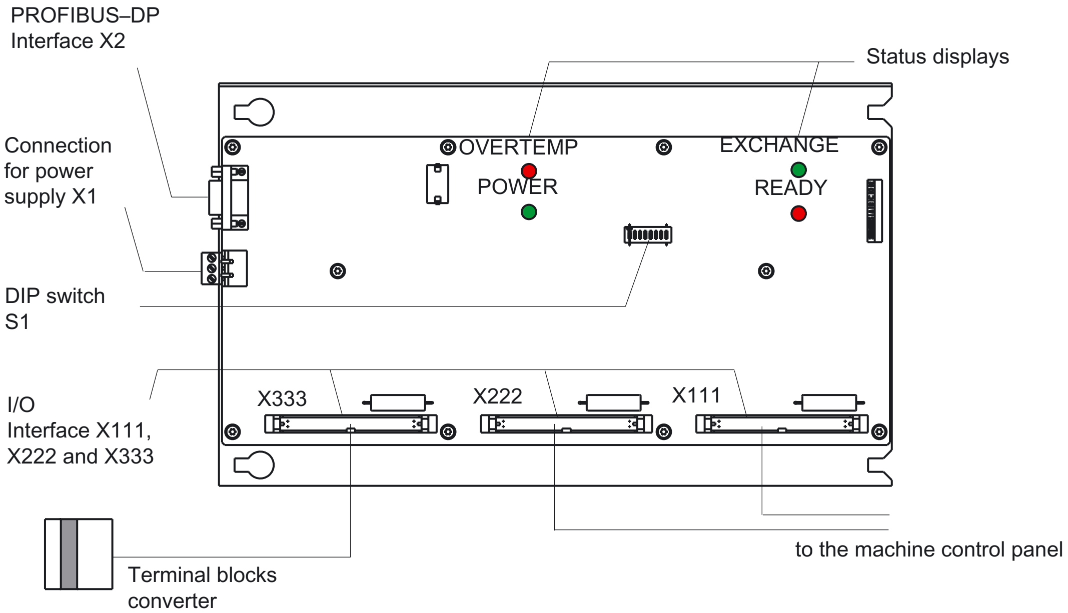


Figure 2-6 Position of the interfaces and status displays on the I/O module with connection to the MCP and a terminal strip converter

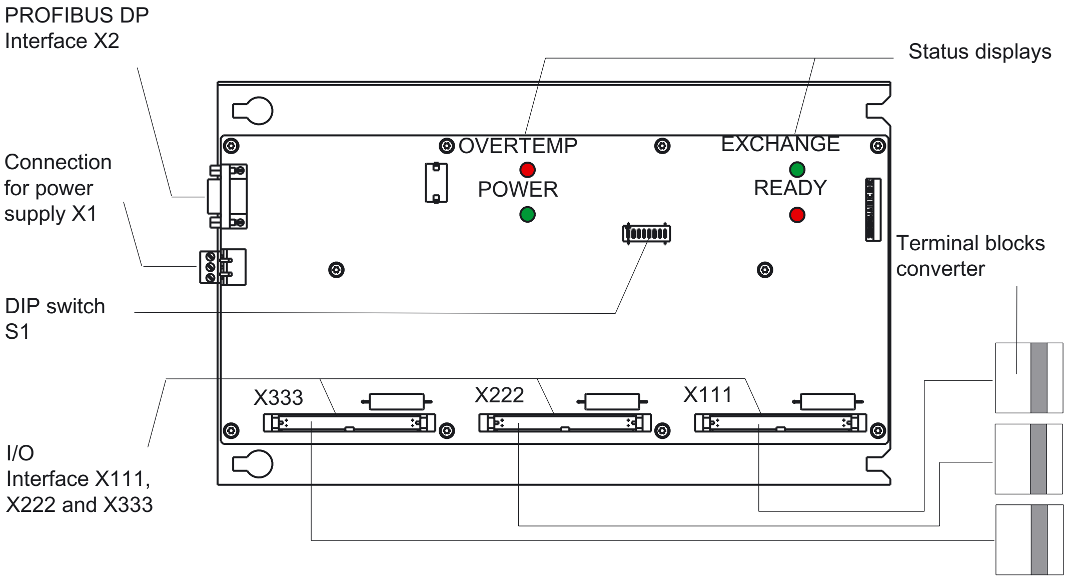


Figure 2-7 Position of the interfaces and status displays on the I/O module when connecting 3 terminal strip converters

PP 72/48 interfaces

The interfaces and control elements of the PP72/48 I/O module and their functions are described in the table below.

Table 2- 17 Interfaces

Interfaces	Function
PROFIBUS-DP interface	9-pin Sub-D socket **X2** for connection to the PROFIBUS-DP
Power supply connection	3-pin screw-type terminal connection **X1** for connecting the 24 V load power supply
I/O device interface	50-pin plug connectors **X111**, **X222**, **X333** for connecting the machine control panel or the terminal strip converters for the digital inputs/outputs
DIL switch	DIL switch **S1** for setting the PROFIBUS-DP address

PROFIBUS-DP interface (X2)

The **PROFIBUS DP** protocol is used for communication.

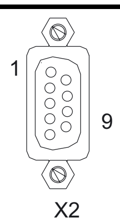
The baud rate of the PROFIBUS-DP interface is 12 Mbps.

The PP72/48 **I**/O module functions as a slave.

Female connector pin assignment

Designation: **X2**
Type: 9-pin Sub-D socket connector

Table 2- 18 Female connector X2 pin assignment

Schematic view of the connector	Pin	Name	Description
	1	not assigned	-
	2	not assigned	-
	3	B	Data input/output (RS485)
	4	RTS	Transmission request
	5	M5	5 V reference potential
	6	P5	5 V power supply 90 mA, short-circuit-proof
	7	not assigned	-
	8	A	Data input/output (RS485)
	9	not assigned	-

I/O interface

The following devices can be connected to the connectors X111, X222 and X333 (50-pin ribbon-cable plug):

- One machine control panel (MCP) and one terminal strip converter for digital inputs/digital outputs

  or

- Three terminal strip converters for digital inputs and digital outputs

The terminal strip converters are connected to the PP72/48 I/O module via ribbon cable. The individual wiring can be performed at the terminal strips according to your particular application.

Connector pin assignment

Designation: X111, X222, X333
Type: 50-pin ribbon cable connector

Table 2- 19 Pin assignment of the connectors X111, X222, X333

Pin	Name	Description	Pin	Name	Description
1	G	Ground	2	$P24OUT_{INT}$	24VDC, internal supply voltage for the inputs
3	DI m+0.0	Input bit	4	DI m+0.1	Input bit
5	DI m+0.2	Input bit	6	DI m+0.3	Input bit
7	DI m+0.4	Input bit	8	DI m+0.5	Input bit
9	DI m+0.6	Input bit	10	DI m+0.7	Input bit
11	DI m+1.0	Input bit	12	DI m+1.1	Input bit
13	DI m+1.2	Input bit	14	DI m+1.3	Input bit
15	DI m+1.4	Input bit	16	DI m+1.5	Input bit
17	DI m+1.6	Input bit	18	DI m+1.7	Input bit
19	DI m+2.0	Input bit	20	DI m+2.1	Input bit
21	DI m+2.2	Input bit	22	DI m+2.3	Input bit
23	DI m+2.4	Input bit	24	DI m+2.5	Input bit
25	DI m+2.6	Input bit	26	DI m+2.7	Input bit
27	not assigned	-	28	not assigned	-
29	not assigned	-	30	not assigned	-
31	DO n+0.0	Output bit	32	DO n+0.1	Output bit
33	DO n+0.2	Output bit	34	DO n+0.3	Output bit
35	DO n+0.4	Output bit	36	DO n+0.5	Output bit
37	DO n+0.6	Output bit	38	DO n+0.7	Output bit
39	DO n+1.0	Output bit	40	DO n+1.1	Output bit
41	DO n+1.2	Output bit	42	DO n+1.3	Output bit
43	DO n+1.4	Output bit	44	DO n+1.5	Output bit
45	DO n+1.6	Output bit	46	DO n+1.7	Output bit

Pin	Name	Description	Pin	Name	Description
47	DOCOMx[1)]	24VDC supply voltage for the outputs	48	DOCOMx[1)]	24VDC supply voltage for the outputs
49	DOCOMx[1)]		50	DOCOMx[1)]	
[1)]x = 1 for connector X111; x = 2 for connector X222; x = 3 for connector X333 m = 0 for connector X111; m = 3 for connector X222; m = 6 for connector X333 n = 0 for connector X111; n = 2 for connector X222; n = 4 for connector X333					

DANGER

The 24 V power supply must be protective extra low voltage in accordance with EN60204-1, Section 6.4, PELV (with M ground).

Note

The connecting cable between the power source, load current supply connection, and associated reference potential M must **not**exceed the maximum permissible length of 10 m.

Digital inputs

The diagram below shows the connector pin assignment for the digital inputs at connection X111 (example). Connectors X222 and X333 are assigned analogously.

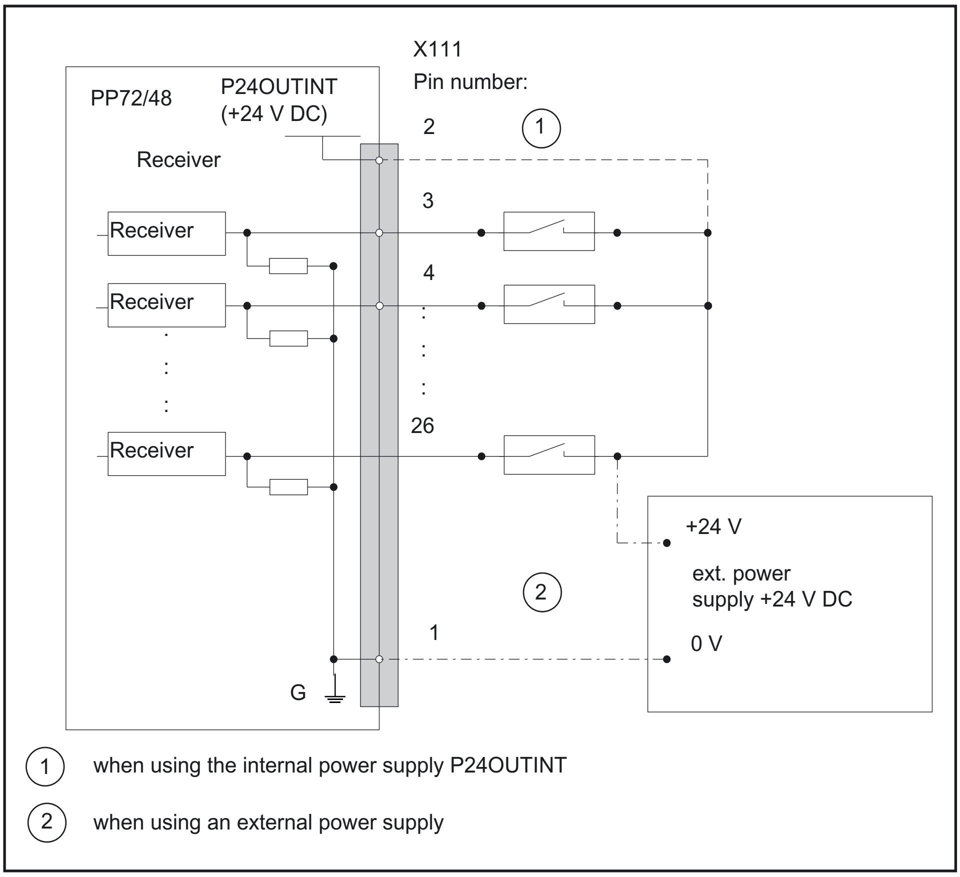


Figure 2-8 Terminal assignment for the digital inputs

Internal power supply (P24OUTINT)

The internal power supply for the digital inputs (X111, X222, X333: pin 2) is taken from the general power supply of module X1, pin 2 (P24).

> **⚠ CAUTION**
>
> Make sure that a max. current of I_{out} = 0.25 A at X111, X222, X333 on pin 2 is not exceeded. An exceeding of the maximum current might destroy the module.

External power supply

If an external power supply is used for the digital inputs, their reference ground must be connected to X111, X222, X333: Pin 1 (G).

X111, X222, X333: Pin 1 ($P24OUT_{INT}$) remains open.

Digital outputs

The diagram below shows the connector pin assignment for the digital outputs at connection X111 (example). Connectors X222 and X333 are assigned analogously.

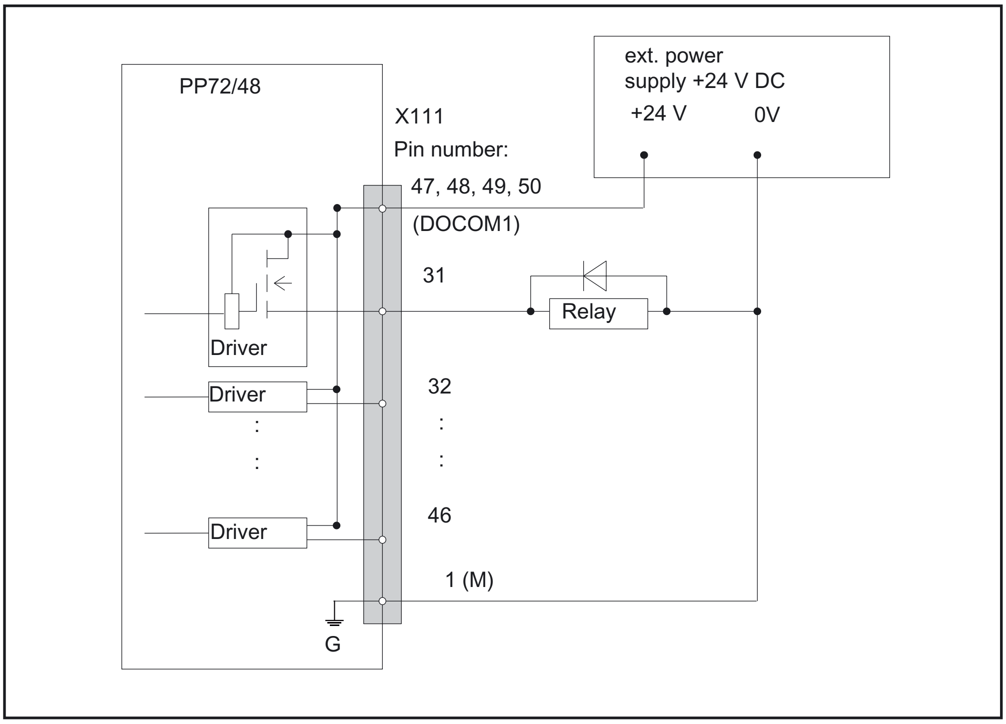


Figure 2-9 Terminal assignment for the digital outputs

To supply the digital outputs, an external 24VDC power supply must be connected to DOCOMx (X111, X222, X333: pins 47, 48, 49, 50).

The reference ground of the external power supply source must be connected to X111, X222, X333: Pin 1 (G).

⚠ CAUTION
It is the user's responsibility to ensure that the max. current consumption per DOCOMx pin (X111, X222, X333: pins 47 through 50) does **not** exceed 1 A. **It is imperative** to connect the 24 V power supply for the digital outputs for DOCOMx **to all four pins** (X111, X222, X333: **pins 47 through 50**).

⚠ DANGER
The 24 V power supply must be protective extra low voltage in accordance with EN60204-1, Section 6.4, PELV (with M ground).

2.5 ADI 4 module interfaces

2.5.1 Overview of connections

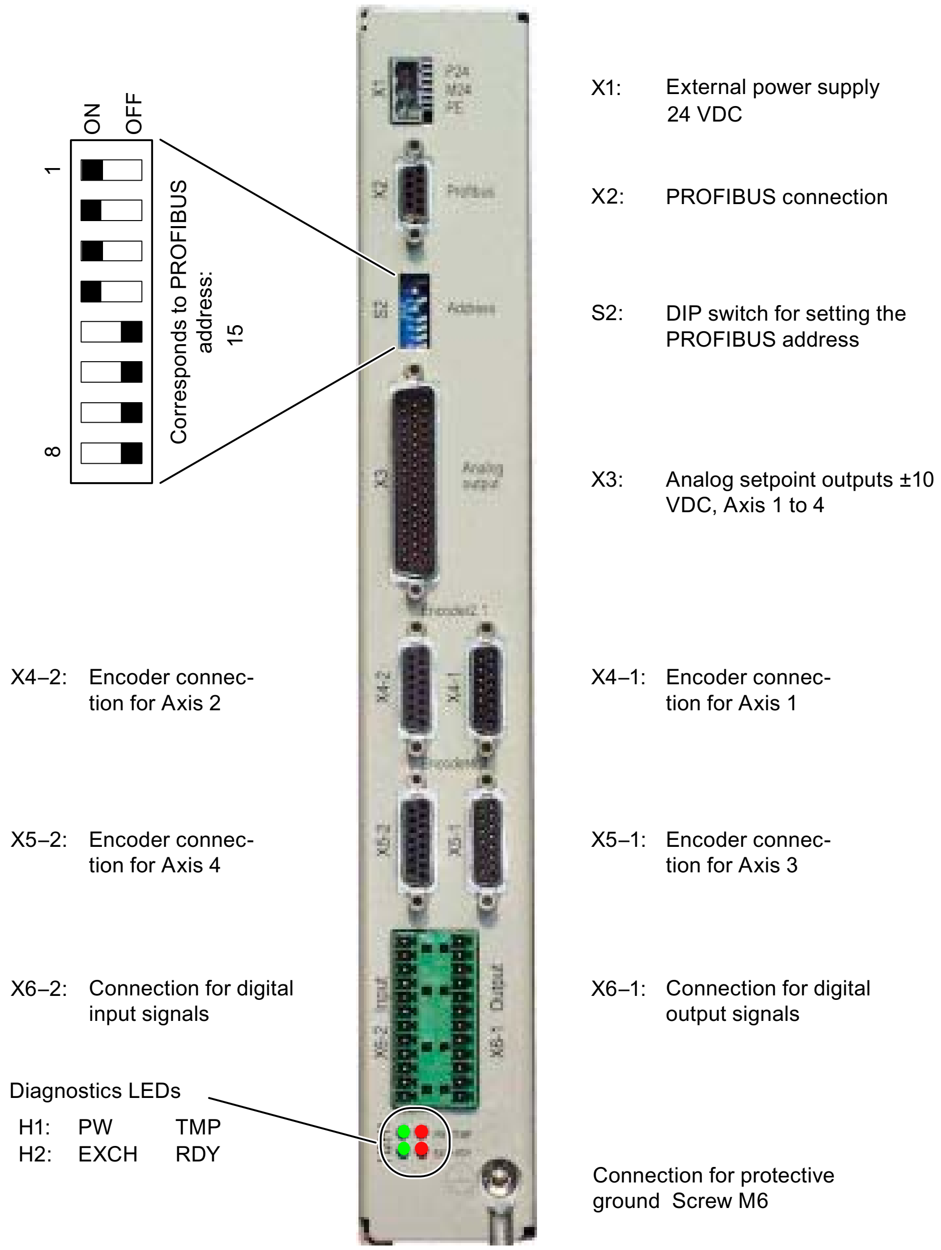


Figure 2-10 Overview of connections

2.5.2 Interface (X2): PROFIBUS DP

Connection

9-pin sub D socket

Pin assignment

Table 2- 20 Pin assignment: PROFIBUS DP (X2)

Pin	Designation	Type[1)]	Function
1	-	-	-
2	-	-	-
3	RxD/TxD-P	B	Receive/transmit data P (B line)
4	RTS	O	Request to Send
5	DGND	VO	Data reference potential (M5V)
6	VP	VO	Supply voltage plus (P5V)
7	-	-	-
8	RxD/TxD-N	B	Receive/transmit data N (A line)
9	-	-	-
[1)] VO: Voltage output O: Output B: Bidirectional			

Connectors

- 6ES7 972-0BA41-0XA0; cable outlet 35°, without PG connection socket
- 6ES7 972-0BB41-0XA0; cable outlet 35°, with PG connection socket

Cables

- 6XV1 830-0EH10; by the meter; without trailing capability
- 6XV1 830-3EH10; by the meter; with trailing capability

Other technical data

Maximum possible data rate: 12 Mbits/s

2.5.3 Interface (S2): PROFIBUS address

Setting

The PROFIBUS address of the ADI 4 DP slave can only be 15 or 16 for the 802D sl and is set using the S2 switch.

- PROFIBUS address 15: S2 switch, 1 to 4 set to ON
- PROFIBUS address 16: S2, only switch 5 set to ON

Table 2- 21 Meaning of switch S2

Switches	Meaning
1	PROFIBUS address: $2^0 = 1$
2	PROFIBUS address: $2^1 = 2$
3	PROFIBUS address: $2^2 = 4$
4	PROFIBUS address: $2^3 = 8$
5	PROFIBUS address: $2^4 = 16$
6	PROFIBUS address: $2^5 = 32$
7	PROFIBUS address: $2^6 = 64$
8	Not used

Note

A newly set PROFIBUS address will only come into effect after power OFF/ON.

2.5.4 Interface (H1/H2): Module status

The module status is displayed on the front of the module with four diagnostic LEDs.

Table 2- 22 Diagnostic LEDs (H1/H2)

Designation		Color	Description
H1	POWER	Green	Supply voltage LED = Off: Supply voltage not applied LED = On: Supply voltage is applied
	OVTEMP	Red	Overtemperature display LED = Off: Device temperature < overtemperature limit LED = On: Device temperature ≥ Overtemperature limit

Designation		Color	Description
H2	EXCHANGE	Green	Status: Message frame exchange with DP master LED = Off: No message frame exchange with DP master LED = On: Cyclic message frame exchange with DP master
	READY	Red	Ready status: Message frame exchange with DP master LED = Off: Not yet ready LED = On: Ready LED = Off and EXCHANGE = On: Message frame exchange active LED = flashing: Error occurred during message frame exchange

Reference

Manual SINUMERIK ADI 4 - Analog Drive Interface for 4 Axes

2.6 Interfaces of the DP/DP coupler

Note

You can find information on the DP/DP coupler in the "SIMATIC, DP/DP coupler" manual.

See also

Connecting the DP/DP coupler (Page 93)

2.7 Interfaces for supplementary components

2.7.1 Interfaces for DRIVE-CLiQ Hub Module DMC20

⚠ DANGER
The 50 mm clearances above and below the components must be observed.

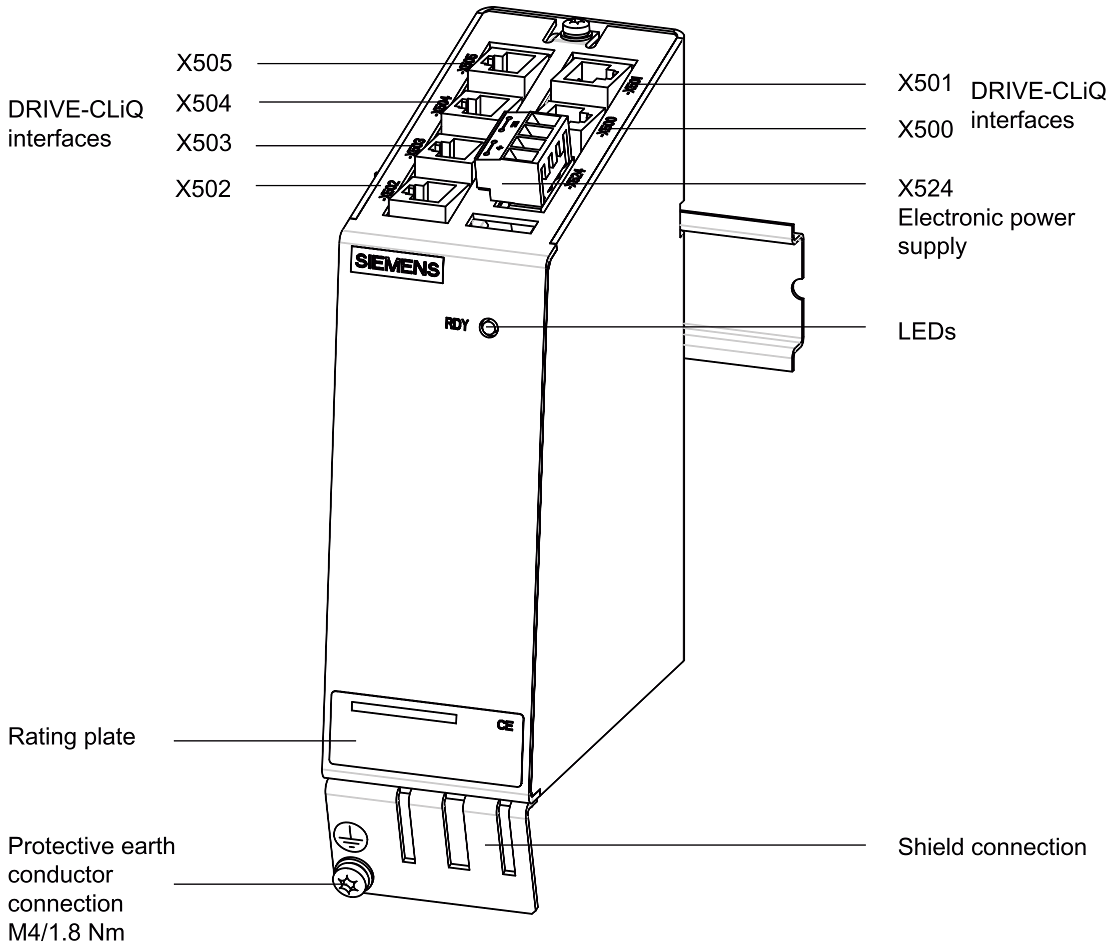
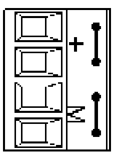


Figure 2-11 Interface description of the DMC20

See also Section Connecting the DRIVE-CLiQ Hub Module DMC20 (Page 96).

Table 2- 23 X524 terminals for the electronic power supply

	Terminal	Designation	Technical specifications
	+	Electronic power supply	24 DC (20.4 – 28.8)
	+	N. c.	
	M	Electronic ground	
	M	Electronic ground	
Max. connectable cross-section: 2.5 mm² Type: Screw terminal type 2			

Note

The two "+" and "M" terminals are jumpered in the connector. This ensures the supply voltage is looped through.

The current consumption increases by the value for the DRIVE-CLiQ node and digital outputs.

Table 2- 24 Significance of the LED on the DMC20

LED	Color	Status	Description
READY	-	Off	Electronics power supply outside permissible tolerance range.
	Green	Steady light	The component is ready for operation and cyclic DRIVE-CLiQ communication is taking place.
	Orange	Steady light	DRIVE-CLiQ communication is being established.
	Red	Steady light	At least one fault is present in this component.
	Green Red	Flashing 2 Hz	The firmware is being downloaded. Component recognition via LED is activated (po154).

2.7.2 Interfaces for Sensor Module Cabinet 10 (SMC10)

2.7.2.1 Safety Information

⚠ WARNING

The 50 mm clearances above and below the components must be observed.

Note

Only one encoder system may be connected per Sensor Module.

Note

There must be no electrical connection between the encoder system housing and the signal cables. Failure to comply may mean the system does not attain the required level of noise immunity.

⚠ CAUTION

Connecting cables to temperature sensors must always be installed with shielding. The cable shield must be connected to the chassis potential at both ends over a large surface area. Temperature sensor cables that are routed together with the motor cable must be twisted in pairs and shielded separately.

2.7.2.2 Overview

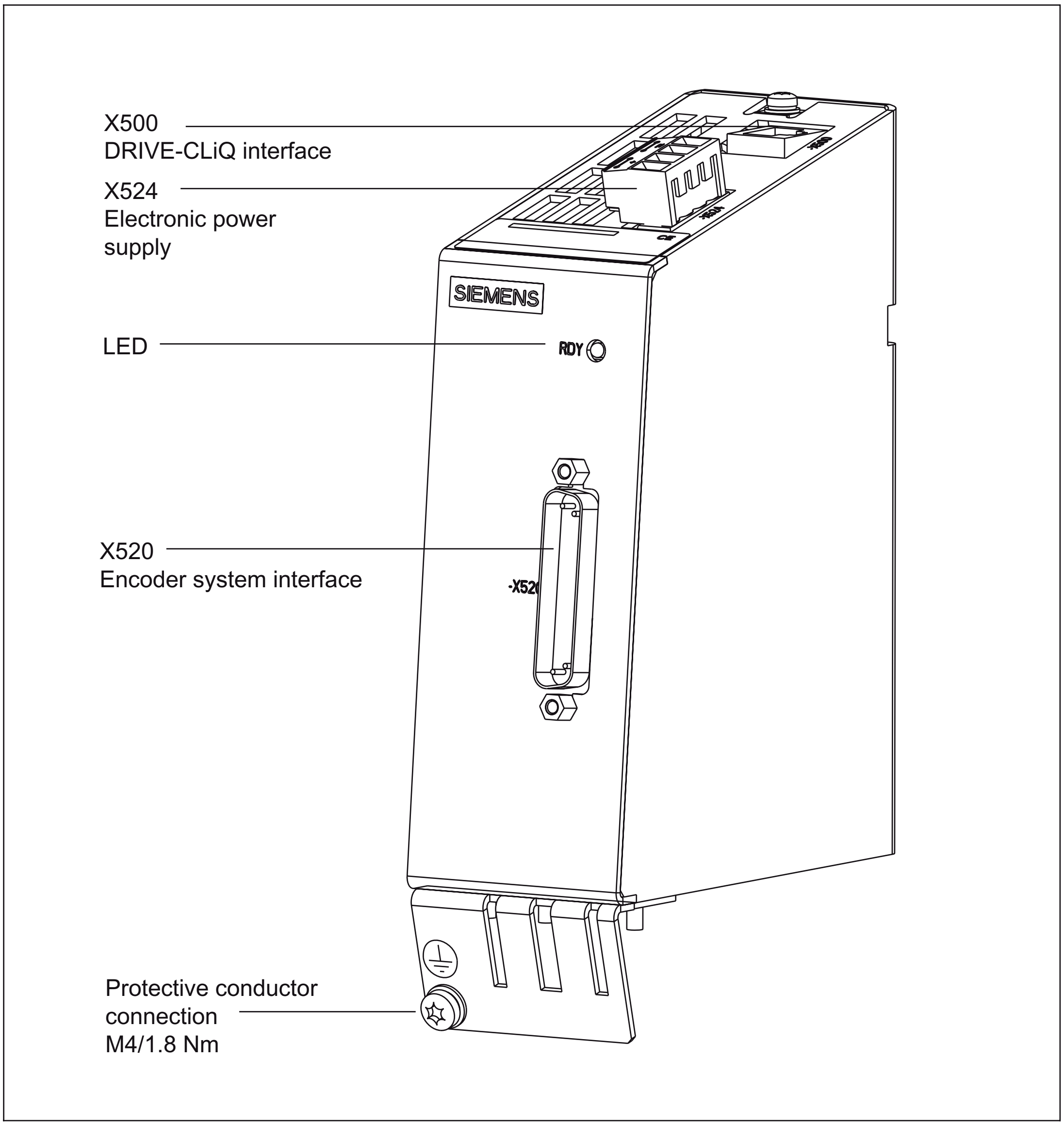
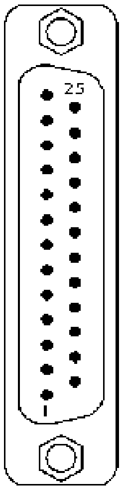


Figure 2-12 Interface description of the SMC10

2.7.2.3 X520 sensor system

Table 2- 25 Sensor interface X520

	Pin	Signal name	Technical data
	1	P encoder	Encoder power supply
	2	M encoder	Ground for encoder power supply
	3	A	Incremental signal A
	4	A*	Inverse incremental signal A
	5	Ground	Ground (for internal shield)
	6	B	Incremental signal B
	7	B*	Inverse incremental signal B
	8	Ground	Ground (for internal shield)
	9	Reserved, do not use	
	10	Clock	Clock EnDat interface, SSI clock[1)]
	11	Reserved, do not use	
	12	Clock*	Inverse clock, EnDat interface, Inverse SSI clock[1)]
	13	+ Temp	Motor temperature measurement KTY84-1C130 (KTY+) Temperature sensor connection KTY84-1C130 / PTC
	14	5 V Sense	Sense input encoder power supply
	15	Data	Data, EnDat interface, SSI data[1)]
	16	0 V Sense	Ground sense input encoder power supply
	17	R	Reference signal R
	18	R*	Inverse reference signal R
	19	C	Absolute track signal C
	20	C*	Inverse absolute value signal C
	21	D	Absolute track signal D
	22	D*	Inverse absolute track signal D
	23	Data*	Inverse data EnDat interface, Inverse SSI data[1)]
	24	Ground	Ground (for internal shield)
	25	- Temp	Motor temperature measurement KTY84-1C130 (KTY-) Temperature sensor connection KTY84-1C130 / PTC

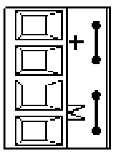
[1)] Only from Firmware 2.4 onwards

2.7.2.4 Safety notes regarding temperature sensor connection

> ⚠ **DANGER**
>
> **Risk of electric shock!**
>
> Temperature sensors should only be connected at the "+Temp" and "-Temp" terminals if they satisfy the safety isolation requirements of EN61800-5-1. If you are unable to guarantee safe electrical separation (e.g. for linear or third-party motors), a Sensor Module External (SME120 or SME125) will need to be used.
>
> Failure to comply carries the risk of electric shock.

2.7.2.5 Electronics power supply X524

Table 2- 26 Terminal block X524

	Terminal	Function	Technical data
	+	Electronic power supply	Voltage: 24 V (20.4 V – 28.8 V) Current consumption: max. 0.35 A Maximum current via jumper in connector: 20 A at 55°C
	+	Electronic power supply	
	M	Electronic ground	
	M	Electronic ground	
Max. connectable cross-section: 2.5 mm² Type: Screw terminal 2			

> **Note**
>
> The two "+" or "M" terminals are jumpered in the connector. This ensures that the supply voltage is looped through.

2.7.2.6 Description of the LEDs on the SMC10

Table 2- 27 Description of the LEDs on the SMC10

LED	Color	State	Technical specifications
	-	Off	Electronics power supply is missing or outside permissible tolerance range.
	Green	Steady light	The component is ready for operation and cyclic DRIVE-CLiQ communication is taking place.
	Orange	Steady light	DRIVE-CLiQ communication is being established.
RDY	Red	Steady light	This component has at least one fault. **Note:** LED is driven irrespective of the corresponding messages being reconfigured.

LED	Color	State	Technical specifications
	Green/ Red	Flashing 2 Hz	Firmware is being downloaded.
	Green/ Orange or Red/ Orange	Flashing 2 Hz	Component recognition via LED is activated (p0144) **Note:** Both options depend on the LED status when component recognition is activated via p0144 = 1.

2.7.3 Interfaces for Sensor Module Cabinet 20 (SMC20)

WARNING

The 50 mm clearances above and below the components must be observed.

Note

Only one measuring system can be connected to each Sensor Module.

Note

There may be no electrical connection between the measuring system housing and the measuring system electronics (this requirement is fulfilled for most encoder systems). If this is not carefully observed, then under certain circumstances the system will not be able to reach the required noise immunity (there is then a danger of equalization currents flowing through the electronics ground).

NOTICE

Connecting cables to temperature sensors must always be installed with shielding. The cable shield must be connected to the chassis potential at both ends over a large surface area. Temperature sensor cables that are routed together with the motor cable must be twisted in pairs and shielded separately.

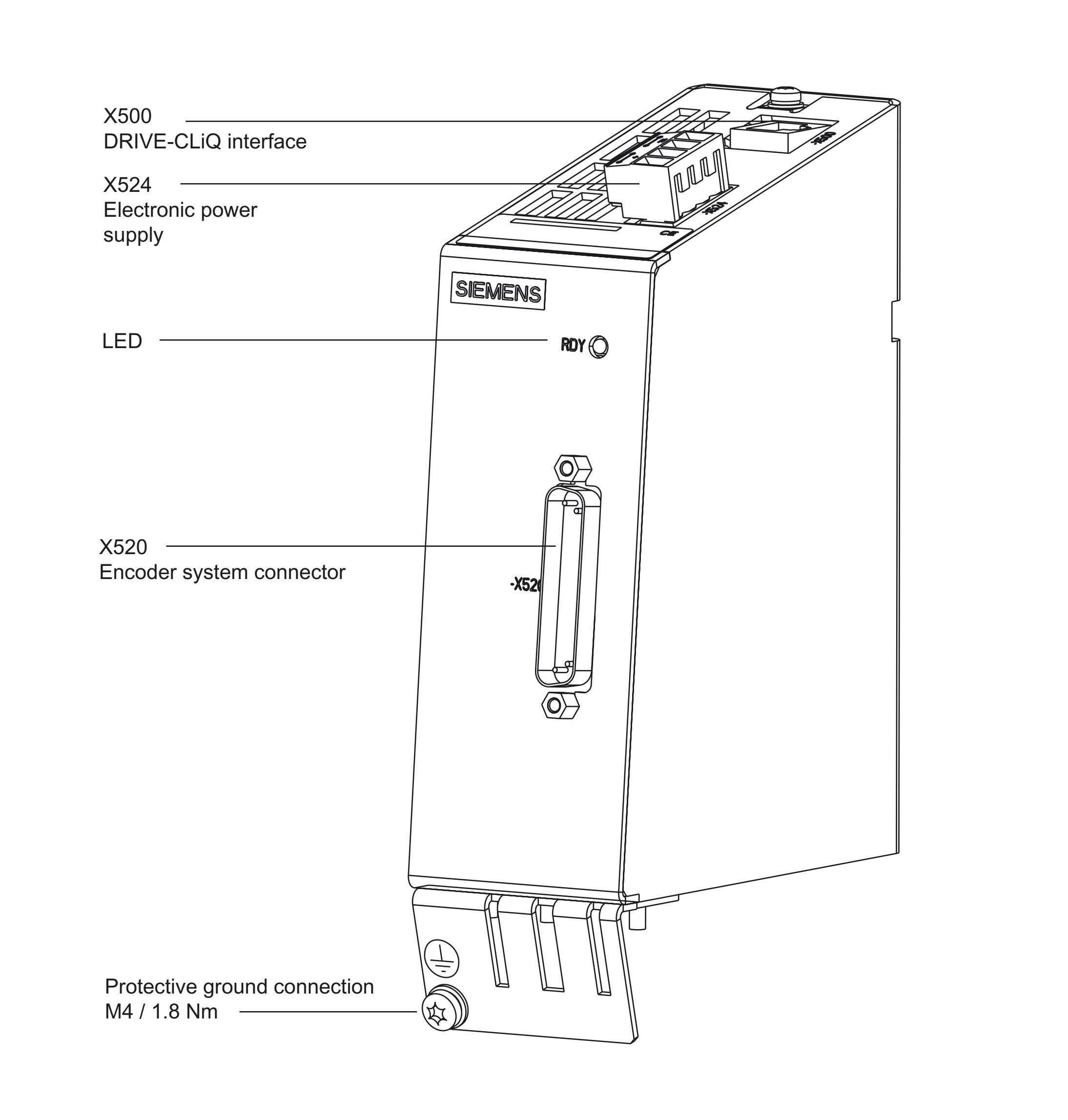
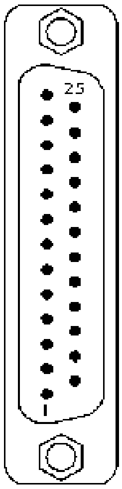


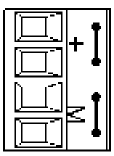
Figure 2-13 Interface description of the SMC20

Table 2- 28 Sensor interface X520

	Pin	Signal name	Technical data
	1	P encoder	Encoder power supply
	2	M encoder	Ground for encoder power supply
	3	A	Incremental signal A
	4	A*	Inverse incremental signal A
	5	Ground	Ground (for internal shield)
	6	B	Incremental signal B
	7	B*	Inverse incremental signal B
	8	Ground	Ground (for internal shield)
	9	Reserved, do not use	
	10	Clock	Clock EnDat interface, SSI clock[1)]
	11	Reserved, do not use	
	12	Clock*	Inverse clock, EnDat interface, Inverse SSI clock[1)]
	13	+ Temp	Motor temperature measurement KTY84-1C130 (KTY+) Temperature sensor connection KTY84-1C130 / PTC
	14	5 V Sense	Sense input encoder power supply
	15	Data	Data, EnDat interface, SSI data[1)]
	16	0 V Sense	Ground sense input encoder power supply
	17	R	Reference signal R
	18	R*	Inverse reference signal R
	19	C	Absolute track signal C
	20	C*	Inverse absolute value signal C
	21	D	Absolute track signal D
	22	D*	Inverse absolute track signal D
	23	Data*	Inverse data EnDat interface, Inverse SSI data[1)]
	24	Ground	Ground (for internal shield)
	25	- Temp	Motor temperature measurement KTY84-1C130 (KTY-) Temperature sensor connection KTY84-1C130 / PTC

[1)] Only from Firmware 2.4 onwards

Table 2- 29 Terminal block X524

	Terminal	Function	Technical data
	+	Electronic power supply	Voltage: 24 V (20.4 V – 28.8 V) Current consumption: max. 0.35 A Maximum current via jumper in connector: 20 A at 55°C
	+	Electronic power supply	
	M	Electronic ground	
	M	Electronic ground	
Max. connectable cross-section: 2.5 mm^2 Type: Screw terminal 2			

Note

The two "+" or "M" terminals are jumpered in the connector. This ensures that the supply voltage is looped through.

Table 2- 30 Description of the LEDs on the SMC20

LED	Color	State	Technical specifications
RDY	-	OFF	Electronics power supply is missing or outside permissible tolerance range.
	Green	Steady light	The component is ready for operation and cyclic DRIVE-CLiQ communication is taking place.
	Orange	Steady light	DRIVE-CLiQ communication is being established.
	Red	Steady light	At least one fault is present in this component. **Note:** LED is driven irrespective of the corresponding messages being reconfigured.
	Green/ Red	Flashing 2 Hz	Firmware is being downloaded.
	Green/ Orange or Red/ Orange	Flashing 2 Hz	Component recognition via LED is activated (p0144) **Note:** Both options depend on the LED status when component recognition is activated via p0144 = 1.

2.7.4 Interfaces for Sensor Module Cabinet 30 (SMC30)

⚠WARNING
The 50 mm clearances above and below the components must be observed.

Note

Only one measuring system can be connected to each Sensor Module.

Note

There may be no electrical connection between the measuring system housing and the measuring system electronics (this requirement is fulfilled for most encoder systems). If this is not carefully observed, then under certain circumstances the system will not be able to reach the required noise immunity (there is then a danger of equalization currents flowing through the electronics ground).

NOTICE
When the measuring system is connected via terminals, make sure that the cable shield is connected to the component.

NOTICE
Connecting cables to temperature sensors must always be installed with shielding. The cable shield must be connected to the chassis potential at both ends over a large surface area. Temperature sensor cables that are routed together with the motor cable must be twisted in pairs and shielded separately.

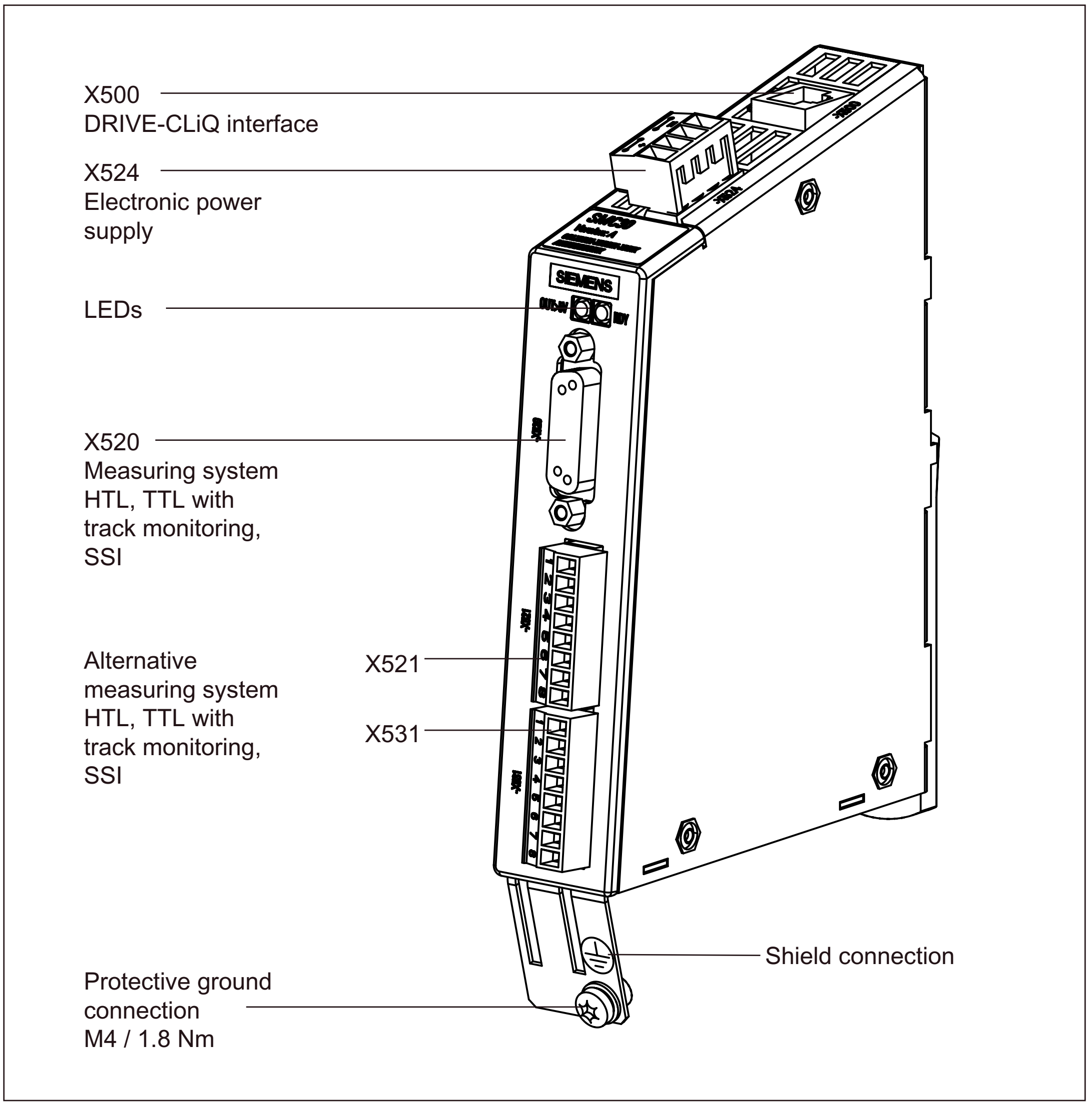


Figure 2-14 Interface description SMC30, 30 mm wide

As from order number 6SL3055-0AA00-5CA2

Connection example 1: HTL encoder, bipolar, with reference signal

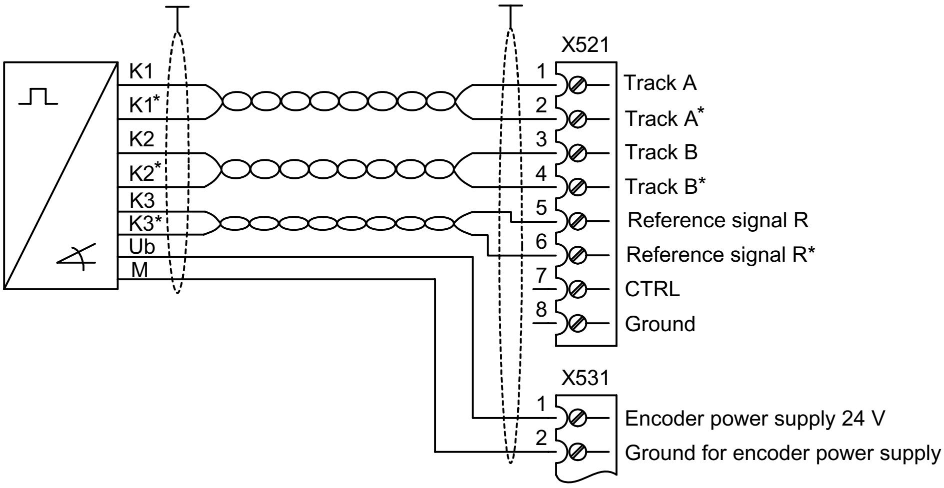


Figure 2-15 Connection example 1: HTL encoder, bipolar, with reference signal

Signal cables must be twisted in pairs in order to improve noise immunity against induced noise.

Connection example 2: HTL encoder, unipolar, with reference signal

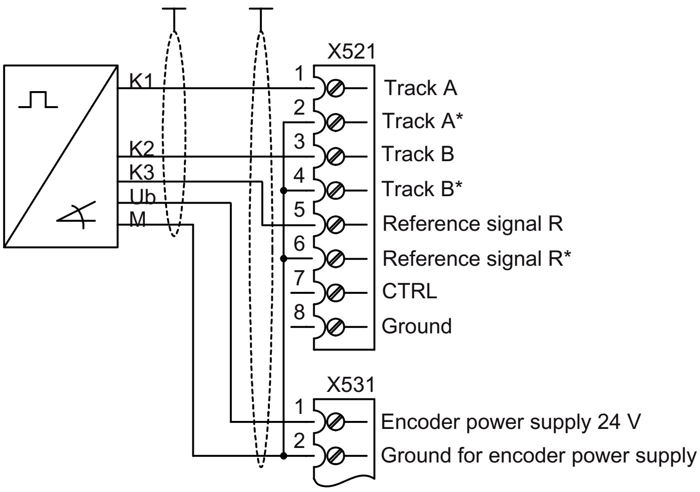


Figure 2-16 Connection example 2: HTL encoder, unipolar, with reference signal1)

1) Because the physical transmission media is more robust, the bipolar connection should always be used. The unipolar connection should only be used if the encoder type does not output push-pull signals.

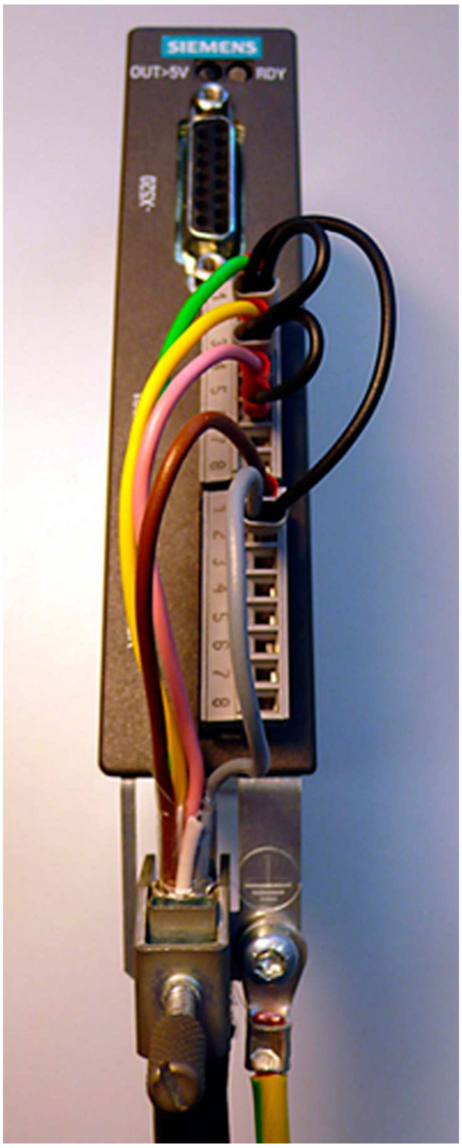
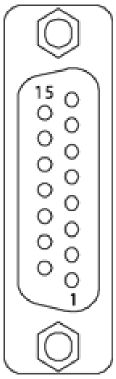


Photo of connection example 2: SMC30, 30 mm wide,
as from order number 6SL3055-0AA00-5CA2

Note: Diagram of the wire jumpers to connect unipolar HTL encoders with reference signal

Table 2- 31 Measuring system connection X520

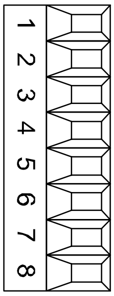
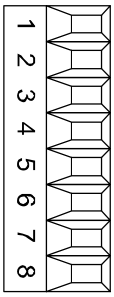
	Pin	Signal name	Technical data
	1	Reserved, do not use + Temp[2]	Motor temperature measurement KTY84-1C130 (KTY+) Temperature sensor connection KTY84-1C130 / PTC
	2	Clock	SSI clock[1]
	3	Clock*	Inverse SSI clock[1]
	4	P_Encoder 5 V / 24 V	Encoder power supply
	5	P_Encoder 5 V / 24 V	
	6	P_Sense	Sense input encoder power supply
	7	M_Encoder (M)	Ground for encoder power supply
	8	Reserved, do not use - Temp[2]	Motor temperature measurement KTY84-1C130 (KTY-) Temperature sensor connection KTY84-1C130 / PTC
	9	M_Sense	Ground sense input
	10	R	Reference signal R
	11	R*	Inverse reference signal R
	12	B*	Inverse incremental signal B
	13	B	Incremental signal B
	14	A* / data*	Inverse incremental signal A / inverse SSI data[1]
	15	A / data	Incremental signal A / SSI data[1]

[1] Only from Order No. 6SL3055-0AA00-5CA1 and Firmware 2.4

[2] Only from Order No. 6SL3055-0AA00-5CA2 and Firmware 2.5 SP1

NOTICE
The encoder power supply can be parameterized to 5 V or 24 V. The encoder may be destroyed if you enter the wrong parameters.

Table 2- 32 Measuring system connection X521 / X531

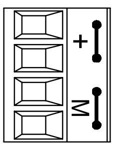
	Pin	Name	Technical data
X521	1	A	Incremental signal A
	2	A*	Inverse incremental signal A
	3	B	Incremental signal B
	4	B*	Inverse incremental signal B
	5	R	Reference signal R
	6	R*	Inverse reference signal R
	7	CTRL	Control signal
	8	M	Ground
	1	P_Encoder 5 V / 24 V	Encoder power supply
	2	M_Encoder	Ground for encoder power supply
X531	3	- Temp	Motor temperature measurement KTY84-1C130 (KTY-) Temperature sensor connection KTY84-1C130 / PTC
	4	+ Temp	Motor temperature measurement KTY84-1C130 (KTY+) Temperature sensor connection KTY84-1C130 / PTC
	5	Clock	SSI clock[2)]
	6	Clock*	Inverse SSI clock[2)]
	7	Data	SSI data[2)]
	8	Data*	Inverse SSI data[2)]
Max. connectable cross-section: 1.5 mm² When using unipolar HTL encoders, at the terminal block A*, B*, R* must be connected to (jumper) M_Encoder (X531)[1)].			

[1)] Because the physical transmission media is more robust, the bipolar connection should always be used. The unipolar connection should only be used if the encoder type does not output push-pull signals.

[2)] Only from Order No. 6SL3055-0AA00-5CA1 and Firmware 2.4

NOTICE
When the measuring system is connected via terminals, make sure that the cable shield is connected to the component. Refer to the Chapter "Electrical connection".

Table 2- 33 Terminal block X524

	Terminal	Function	Technical data
	+	Electronics power supply	Voltage: 24 V (20.4 V – 28.8 V) Current consumption: max. 0,55 A Maximum current via jumper in connector: 20 A at 55 °C
	+	Electronic power supply	
	M	Electronics ground	
	M	Electronics ground	
Max. connectable cross-section: 2.5 mm² Type: Screw terminal 2			

Note

The two "+" or "M" terminals are jumpered in the connector. This ensures that the supply voltage is looped through.

Table 2- 34 Description of the LEDs on the SMC30

LED	Color	State	Description
RDY	-	Off	Electronic power supply is missing or outside permissible tolerance range.
	Green	Steady light	The component is ready for operation and cyclic DRIVE-CLiQ communication is taking place.
	Orange	Steady light	DRIVE-CLiQ communication is being established.
	Red	Steady light	At least one fault is present in this component. **Note:** The LED is driven irrespective of the corresponding messages being reconfigured.
	Green/ Red	Flashing light 2 Hz	Firmware is being downloaded.
	Green/Orange or Red/Orange	Flashing light 2 Hz	Component recognition via LED is activated (p0144) **Note:** Both options depend on the LED status when component recognition is activated via p0144 = 1.
OUT > 5 V	-	Off	If the RDY LED is also off, the electronic power supply is either missing or outside the permissible tolerance range. Otherwise, measuring system power supply <= 5 V
	Orange	Steady light	Measuring system power supply > 5 V **Notice** It must be ensured that it is permissible to operate the connected encoder with a 24 V power supply. Operating an encoder intended for a 5-V connection on 24 V can irreparably damage the encoder electronics.

Cause and rectification of faults

The following reference contains further information about the cause and rectification of faults:

References

SINAMICS S120 Commissioning Manual

3 Application planning

3.1 Overview

Basic Rules

The present chapter describes various general rules for electrical design. You must observe these basic rules to ensure trouble-free operation.

Safety regulations

To ensure safe operation of your plant, realize the following measures and adapt them to your particular conditions:

- An EMERGENCY OFF concept in accordance with the applicable regulations

  (e.g. European standards EN 60204, EN 418, and similar)
- Additional measures for end position limiting of axes

  (e.g. hardware limit switches)
- Equipment and measures for protection of motors and power electronics in accordance with the SINAMICS Installation Guidelines.
- Also refer to Chapter "ESD Guidelines" in the Appendix of this manual.

References

For further information about EMC guidelines, we recommend the publication: EMC Guidelines, Configuration Manual

Standards and regulations

When connecting SINUMERIK 802D sl, please observe the relevant VDE guidelines, in particular VDE 0100 or VDE 0113 for disconnecting devices, short-circuit and overload protection.

3.2 General rules for operation of a SINUMERIK 802D sl

When integrating a SINUMERIK 802D sl into a plant, you must observe the following general rules.

Starting the plant after certain events

If ...	then ...
Startup after voltage drop or power failure,	dangerous operating states must be excluded. If necessary, force EMERGENCY-OFF.
Startup after releasing the EMERGENCY OFF device,	no uncontrolled or undefined start must occur.

Mains voltage

At ...	make sure that ...
Stationary plants or systems without all-pole line voltage disconnect switch	the building installation must be equipped with a power disconnect switch or a fuse.
load power supplies, power supply modules	the set range of the rated voltage complies with the local mains voltage.
all current circuits	deviation of the line voltage from the rated value must be within the permitted tolerance (refer to "Technical data of the installed components").

24VDC power supply

At ...	ensure ...
24 V supply	Safe (electrical) isolation of low voltage

Protection against external electrical interference

At ...	make sure that ...
all plants, installations and systems in which SINUMERIK is installed	the plant or system is connected to the protective conductor for diverting electromagnetic interference.
Supply, signal, and bus lines	The wiring arrangement and installation complies with EMC regulations.
Signal and bus lines	A cable or wire break must not give rise to undefined states in the plant or system.

3.3 Rules regarding current consumption and power loss of a cubicle arrangement

The power loss of all components used in a cabinet must not exceed the maximum amount that can be dissipated from the cabinet.

Note

When dimensioning the control cabinet, you must make sure that the permissible ambient temperature is not exceeded for the components installed, even if the outside temperatures are high.

For the current consumption and the power loss of the individual modules, please refer to Chapter "Technical Data" (Page 441).

4 Assembling

Overview

To install **SINUMERIK 802D sl**, first secure the individual components on the site of installation and then connect them with each other.

Open Equipment

The modules of **SINUMERIK 802D sl** are open-type equipment. This means that you may only install SINUMERIK 802D sl in enclosures, control cabinets or electrical service rooms. Access to these housings, cubicles or electrical service rooms must only be possible using a key or a tool Only trained or authorized personnel are allowed access to these enclosures, control cabinets or electrical operating rooms.

General procedure when installing SINUMERIK 802D sl

⚠ WARNING
Before installing or removing the components of the SINUMERIK 802D sl control system, make sure that the system is disconnected from the mains.

Note

When installing the control components, observe the dimensions given in Chapter "Dimension Drawings". The drilling patterns constitute the basis for preparing the mounting holes.

Note

Mounting the CNC operator panel (PCU), machine control panel and full CNC keyboard

The maximum permissible tightening torque for the fixing screws is 1.8 Nm and this value must not be exceeded.

Installing the CNC operator panel (PCU)

Install the CNC operator panel as shown in the relevant illustrations and diagrams in the chapter titled "Dimension drawings".

> ⚠ CAUTION
>
> If you do not have access to the back of the control system during installation, you must connect the CNC operator panel prior to installation. When doing so, note that connector X40 (power supply connection) and the conductors connected to it protrude beyond the mounting edge.
>
> When installing the CNC operator panel, do **not** pull off the connector and take care **not to damage** the cables.

Installing the machine control panel

Install the machine control panel as shown on the relevant illustrations and diagrams in Chapter "Dimension Drawings".

Installing the full CNC keyboard

You can install the full CNC keyboard either next to the operator panel or beneath the CNC operator panel. Observe the specifications in the illustrations in Chapter "Dimension Drawings".

Installing the PP72/48 I/O module

The module must be installed according to EN 60204. Dimension drawing of the module, see Chapter "Dimension Drawings".

Installing the SINAMICS S120 drive

For information regarding the **SINAMICS S120** drive system (design, connection, planning, dimensioning, configuring, etc.)

References

SINAMICS S120 Equipment Manuals

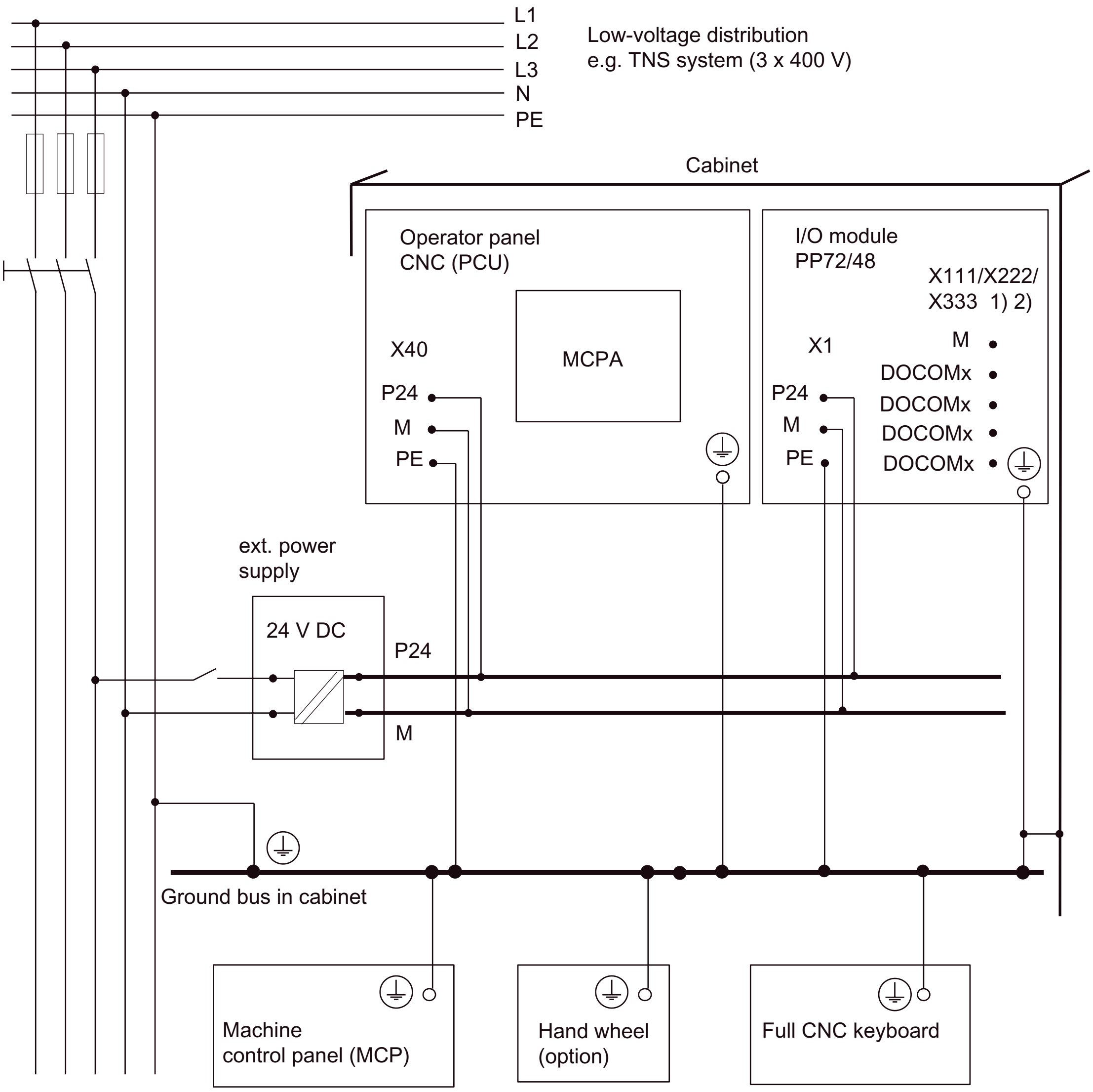
See also

Dimensional Drawings (Page 455)

Connecting

5

5.1 Overall design of the SINUMERIK 802D sl



1) For use of an external power supply for digital inputs, see Chapter "Interfaces of PP 72/48".

2) The load power supply is configured by the user.

Figure 5-1 Possibility of supplying the modules via a grounded infeed

5.2 Connecting the protective conductor for the individual components

⚠ CAUTION
The individual components shown in the Fig. "Possibility of Supplying Modules" require connection to a protective conductor. The individual components must be connected to the central grounding point. Make always sure that a low-resistance connection is provided to the protective conductor.

Minimum cross-section of the cable to the protective conductor: 10 mm^2

Whereas all remaining components are grounded via a grounding screw, the PP72/48 I/O module must be connected directly to the central grounding point via the mounting plate (installation acc. to EN 60204). If no grounding can be provided via the mounting plate, it **must** be connected to the central grounding point via an additional line (cross-section ≥ 10 mm^2).

5.3 Connection overview for SINUMERIK 802D sl

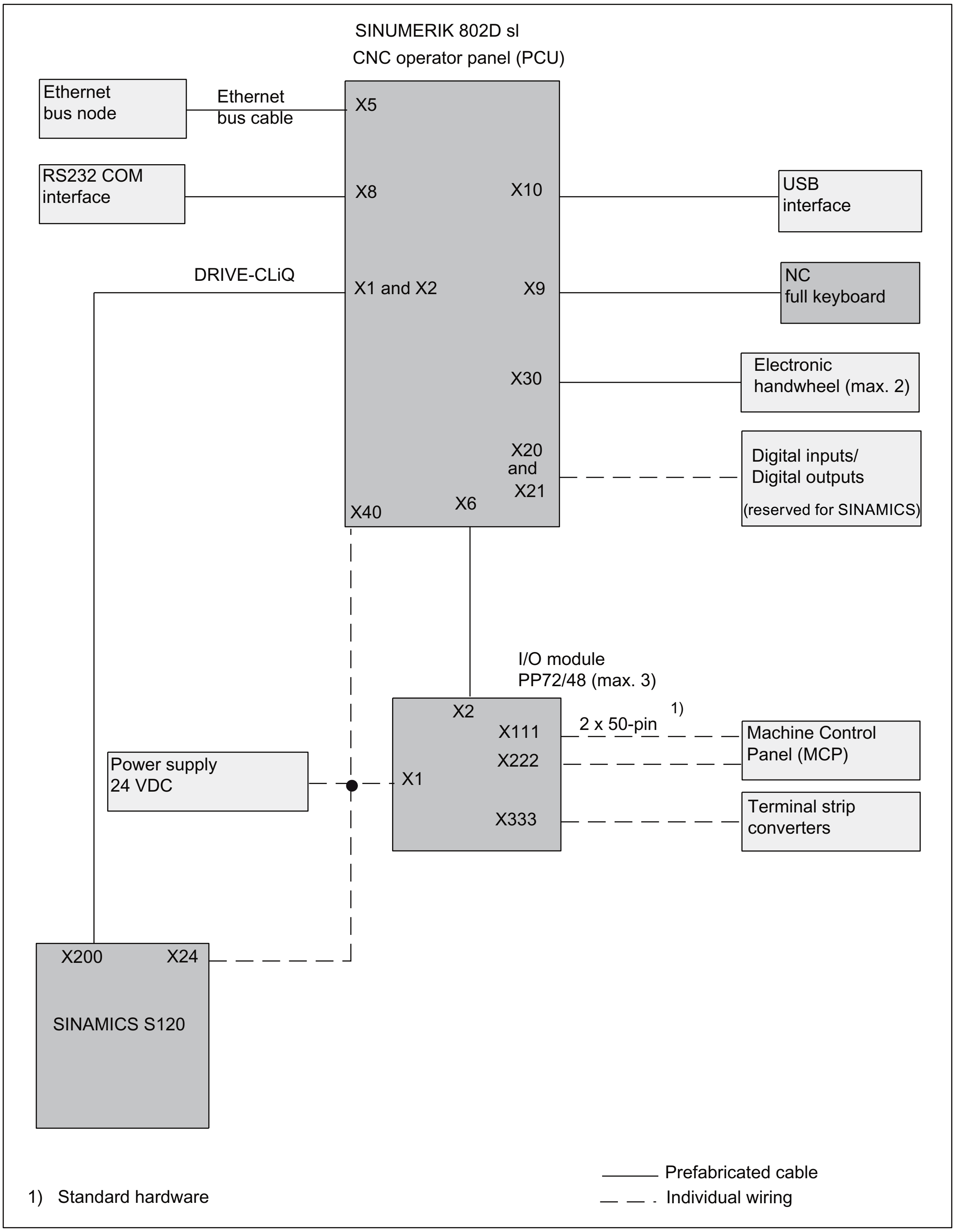


Figure 5-2 Connection overview without MCPA module

X20/X21

The fast digital inputs and digital outputs at the X20 and X21 connectors are reserved for SINAMICS.

References

For detailed descriptions, see "PLC_appl_Programm.pdf" in the "...\Programming Tool PLC\Lib802Dsl" Toolbox installation directory.

Note

Connect the lines as shown in the "Connection overview without MCPA module" illustration.

The preassembled cable sets from Siemens provide optimum interference immunity.

References

For information regarding the cables (cable designations, connector types, etc.), see:

Catalog NC 802D sl, Catalog NC 61

For information regarding PROFIBUS-DP and Industrial Ethernet, see:

Catalog IK PI

5.4 Connecting the MCPA module

The MCPA module is connected to the PCU via X110. The ribbon cable (length 0.6 m, part of the MCP 802D sl delivery) is used to connect the machine control panel MCP 802D sl.

For this, the following connectors from the MCPA module and the machine control panel MCP 802D sl are connected:

MCPA	MCP 802D sl
X1	X1201
X2	X1202

Note

When digital outputs are connected at connector X1021 or an analog spindle is connected at connector X701, connector X1021 (PIN1 24 V; PIN10 0 V) must be used for the MCPA module's power supply.

Note

The variable assignment of the machine control panel is described in the PLC user interface.

References

SINUMERIK 802D sl Function Manual

PLC user library V01.07.00 of SINUMERIK 802D sl

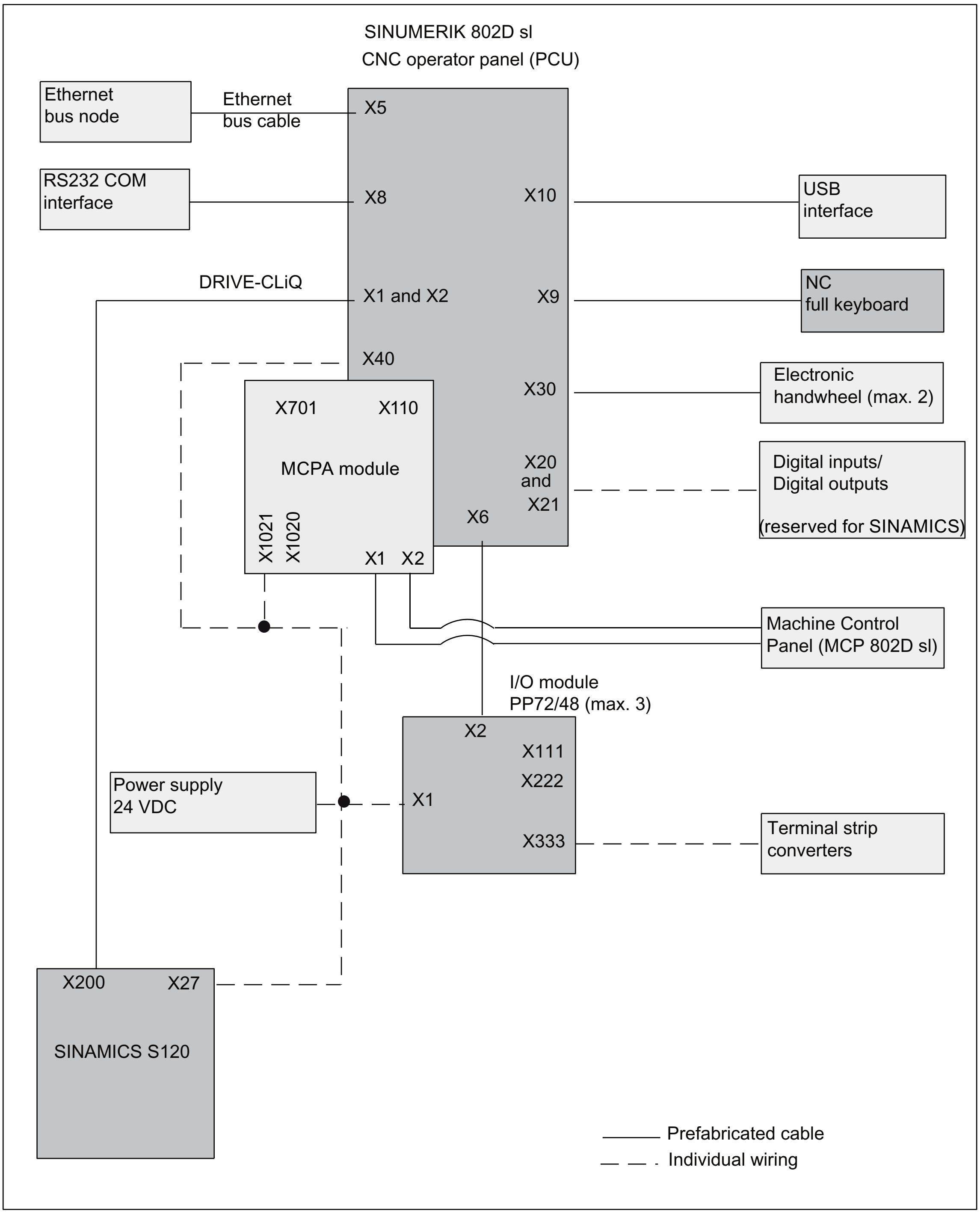


Figure 5-3 Connection overview with MCPA module

References

For detailed descriptions, see "PLC_appl_Programm.pdf" in the "...\Programming Tool PLC\Lib802Dsl" Toolbox installation directory.

5.5 Connecting the high-speed digital inputs/outputs at the MCPA module

The high-speed digital inputs/outputs are connected via interfaces X1020 and X1021 of the MCPA module.

See also

MCPA module interfaces (Page 40)

5.6 Connecting the power supply

The required 24 V DC load power supply must be connected to the following connectors:

- On screw terminal block X40 of the operator panel CNC
- On screw terminal block X1 of the I/O module PP72/48

Features of the load power supply

⚠ DANGER
The 24 V DC protective extra-low voltage **must** be generated as a protective extra-low voltage with safe electrical isolation (to IEC 204-1, Section 6.4, PELV) and grounded by the user (provide a PELV M signal connection to the central grounding point of the system).

Table 5- 1 Electrical parameters of the load power supply for the CNC operator panel (X40) and for the PP72/48 I/O module (X1)

Parameters	Values	Conditions
Voltage range mean value	20.4...28.8 V	
Ripple	3.6 Vpp	
Non-periodic overvoltage	35 V	500 ms duration 50 s recovery time
Rated current consumption • CNC operator panel (PCU) • PP72/48 I/O module	typically 1 A -	

Parameters	Values	Conditions
Starting current • CNC operator panel (PCU) • PP72/48 I/O module	2.6A -	
Power consumption • CNC operator panel (PCU) • PP72/48 I/O module	max. 50 W max. 11 W	

Table 5- 2 Pin assignment of the screw-terminal blocks X40 (on the PCU) and X1 (on the I/O module)

Terminal	Signal	Description
1	P24	24 V DC
2	M	Ground
3	PE	Protective earth

Note

Make sure that the connecting cable between the voltage source and the load power supply connection does not exceed a maximum length of 10 m (PP72/48 I/O module only).

Connecting the mains lines

⚠WARNING
Before connecting the modules, first disconnect the equipment from the mains!

For connecting the power supply, use flexible lines with a line cross-section of at least 1 mm^2.

Use end sleeves to connect the lines.

Remove the insulation from the cable end, assemble the end sleeve, insert the cable end (with end sleeve) into the screw terminal connection and tighten the fastening screw.

Insert the screw terminal with cables into connector X40 on the CNC operator panel (PCU).

Reverse polarity protection

With correct connection and the power supply turned on, the LEDs "RDY" (PCU) and "POWER" (PP72/48) are lit in green.

Note

In the event of polarity reversal, the control system will not work. However, a built-in reverse polarity protection will protect the electronics against damage.

Fuse

If the control is defective, an internal fuse protects the electronics from consequential damage (e.g., fire). In this case, the entire control system must be replaced.

5.7 Connecting the full CNC keyboard to the CNC (PCU) operator panel

The connection cable for connecting the full CNC keyboard to the CNC operator panel (PCU) is supplied together with the full CNC keyboard. Connect socket X9 on the CNC operator panel (PCU) to the PS/2 socket on the rear of the full CNC keyboard.

Reference

Catalog NC 802D sl, Catalog NC 61

5.8 Connecting the Ethernet interface

Connect the Ethernet connection cable to the CNC operator panel, female connector X5. Make sure that the connector locks into position when connecting.

See also

Ethernet interface (Page 26)

5.9 Connecting the RS232 COM port

Insert the Sub-D sockets into connector X8 on the CNC operator panel and into the connector on the PG/PC. Lock the connector into position using the knurled screws.

Note

Use only shielded lines twisted in pairs; the shield must be connected to the metal or metalized connector casing on the side of the control system.

The cable set offered as accessories provides maximum interference immunity.

Connection diagram

The diagram below shows the pin assignment of the interconnecting cable between the CNC operator panel and a PG/PC with 9-pin or 25-pin socket connector.

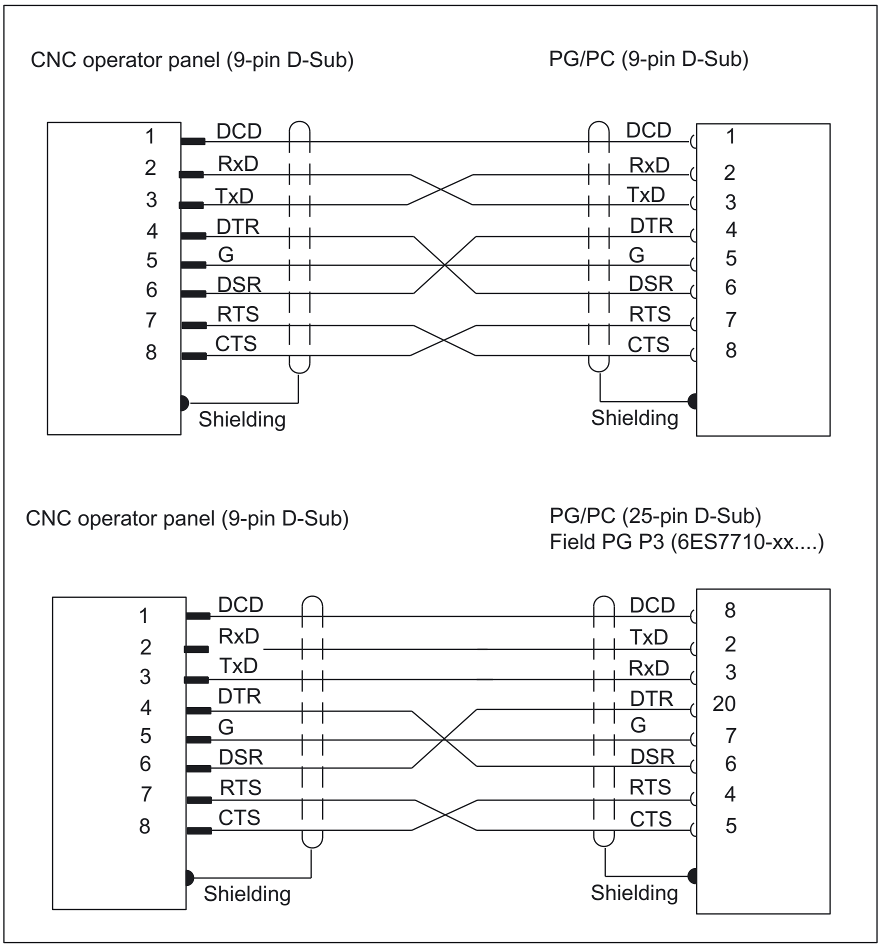


Figure 5-4 Connection diagram for interconnecting the CNC operator panel and the PG/PC

5.10 Connecting PROFIBUS

PROFIBUS is used to interconnect all the nodes. The PCU is the master; ADI 4 and PP 72/48 are slaves.

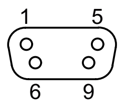
The baud rate for the clock-synchronized PROFIBUS is permanently preset to 12 Mbaud and cannot be changed. Converters for fiber-optic cable (OLMs, OLPs) or repeaters are not permitted.

PROFIBUS interface

Socket type: 9-pin D-Sub socket

Max. cable length: 100 m at 12 Mbaud

Table 5- 3 Socket pin assignment

Pin	Signal	Meaning	
1	Shield		
2	Reserved		
3	RxD/TxD-P	Receive/transmit data plus, B cable (red)	
4	CNTR-P	Control signal for repeater (direction control)	
5	DGND	Data transmission potential (ground to 5 V)	
6	VP	Supply voltage for the terminating resistors P (P5V)	
7	Reserved		
8	RxD/TxD-N	Receive/transmit data plus, A cable (green)	
9	CNTR-N	Repeater control signal (direction control)	

Note

Only use the recommended PROFIBUS connectors. These are designed in such a way that the continuing PB branch becomes disconnected when a terminating resistor is connected.

PB-Master = PCU should be located at the start of the PB line.

Please ensure terminating resistors are only placed at the first and last nodes.

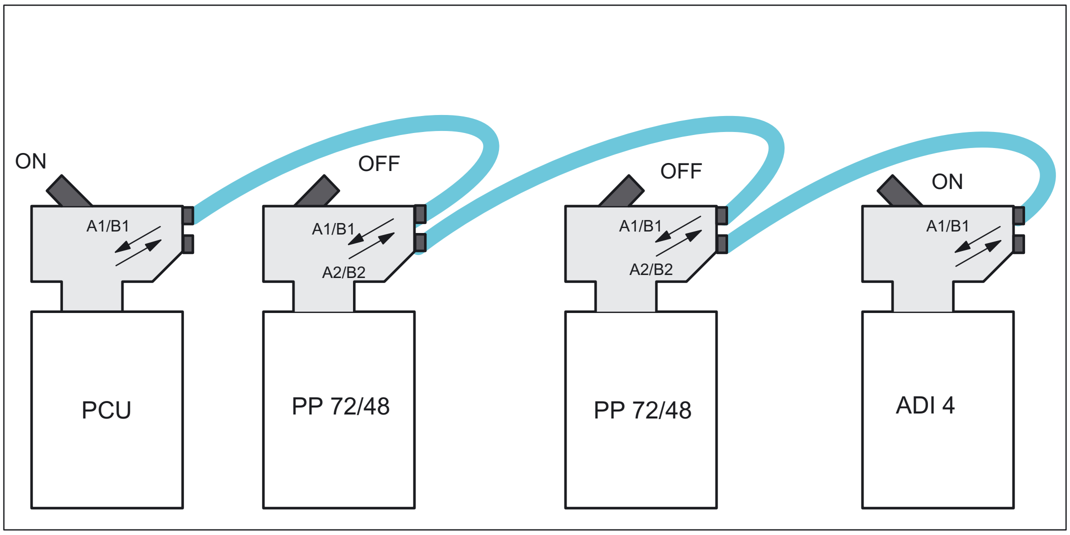


Figure 5-5 Basic arrangement of a PROFIBUS line

5.11 Connecting a modem

Connecting a modem

Insert the Sub-D sockets into connector X8 on the CNC operator panel and into the connector on the PG/PC. Lock the connector in position using the knurled screws.

See RS232 COM port (Page 28) Connecting the RS232 COM port (Page 86)

Modem types

The following modems can be connected:

- All external modems with the following features:
  - V24 terminal
  - AT command set that can be directly addressed via the V24 interface (no soft modems)
- ISDN modem
- Analog modem
- ISDN/analog combination box or cell phones (hardware modem with AT command set)

Other features of the RS232/modem interface

- Data formats: 8N1 or 8E1
- V24 baud rates: 9,600; 19,200; 38,400; 57,600; and/or 115,200 bps

Special features

- If you are using an ISDN connection, the terminal with which you are communicating must also support ISDN or you will need to use an ISDN/analog combination box, which can connect to both normal ISDN (digital) lines and analog lines.

  Accordingly, the inverse applies to modems: either a hybrid or an analog device is required at the other end of the connection.

- When an ISDN box is connected to larger telephone systems (e.g. HICOM), the full range of ISDN functions will generally not be available, despite the telephones having identical connectors and functionality (there is no S0 bus).

  For this reason, it is not possible to establish a direct connection to this type of box.

  Where HICOM systems are concerned, it is possible to use a telephone with an ISDN adapter plug, for example.

  Once configured correctly, a full range of ISDN functions will then be available for this device.

- If the communication peer takes the form of a mobile phone (hardware modem with AT command set required), whether it supports digital or analog connections (or both) will depend primarily on the make of phone.

  Furthermore, it should be noted, for example, that some mobile phone network operators (and contracts) only permit outgoing data links to be established as standard. In such cases, you will need to reserve/take out a contract for a second number for incoming "data exchanges".

- You must check the following:
  - The AT command set of the modem being used; it may be necessary to change the default values set in the controller.
  - The checkback signals OK, CONNECT, NO CARRIER, RING
  - The automatic call pick-up function (if the device cannot be used to pick up calls automatically, the software mode must be set, as well as the default AT command ("ATA") for picking up calls)
  - ESC sequence, hang-up sequence, and MSN on ISDN

5.12 Connecting the PP72/48 I/O module

PNO design guidelines

For electrical PROFIBUS networks, please also refer to the PROFIBUS-DP/FMS design guidelines issued by the PROFIBUS user organization. These contain important information about cable routing and the commissioning of PROFIBUS networks.

Publisher: PROFIBUS-Nutzerorganisation e.V.
Haid-und-Neu-Strasse 7
76131 Karlsruhe
Tel: +49 721 / 9658 590
Fax: +49 721 / 9658 589
Internet: http://www.profibus.com

Guideline, order no. 2.112

Bus node

The following bus nodes can be connected via the PROFIBUS-DP interface:

- CNC operator panel (always master)
- PP72/48 I/O module (slave)

Bus connector and bus cable

The PROFIBUS cable is a two-core, stranded and shielded cable which must not be twisted, stretched or squeezed.

References

For more information regarding the bus connector, the bus cable and the cable length, please refer to:

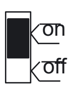
Catalog NC 802D sl

Connecting the Bus Connector

To connect the bus connector, proceed as follows:

1. Plug the bus connector into the module.
2. Screw the bus connector firmly into place.
3. If the bus connector is at the start or end of the PROFIBUS-DP connection, you must connect the terminating resistor on the connector (switch position "ON").

Terminating resistor
Turned on



Terminating resistor
not enabled

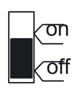


Figure 5-6 Bus connector terminating resistor switched on and off

⚠ WARNING
A bus segment must always be terminated on both ends; otherwise, the data traffic at the bus could be disturbed. Please make sure during startup and normal operation that power is always supplied to stations where the terminating resistor is active. The terminator is without effect if the last station to which a bus connector is connected is dead, since the bus connector is powered from the station.

Networking example

The diagram below shows a networking example for SINUMERIK 802D sl with two PP72/48 I/O modules.

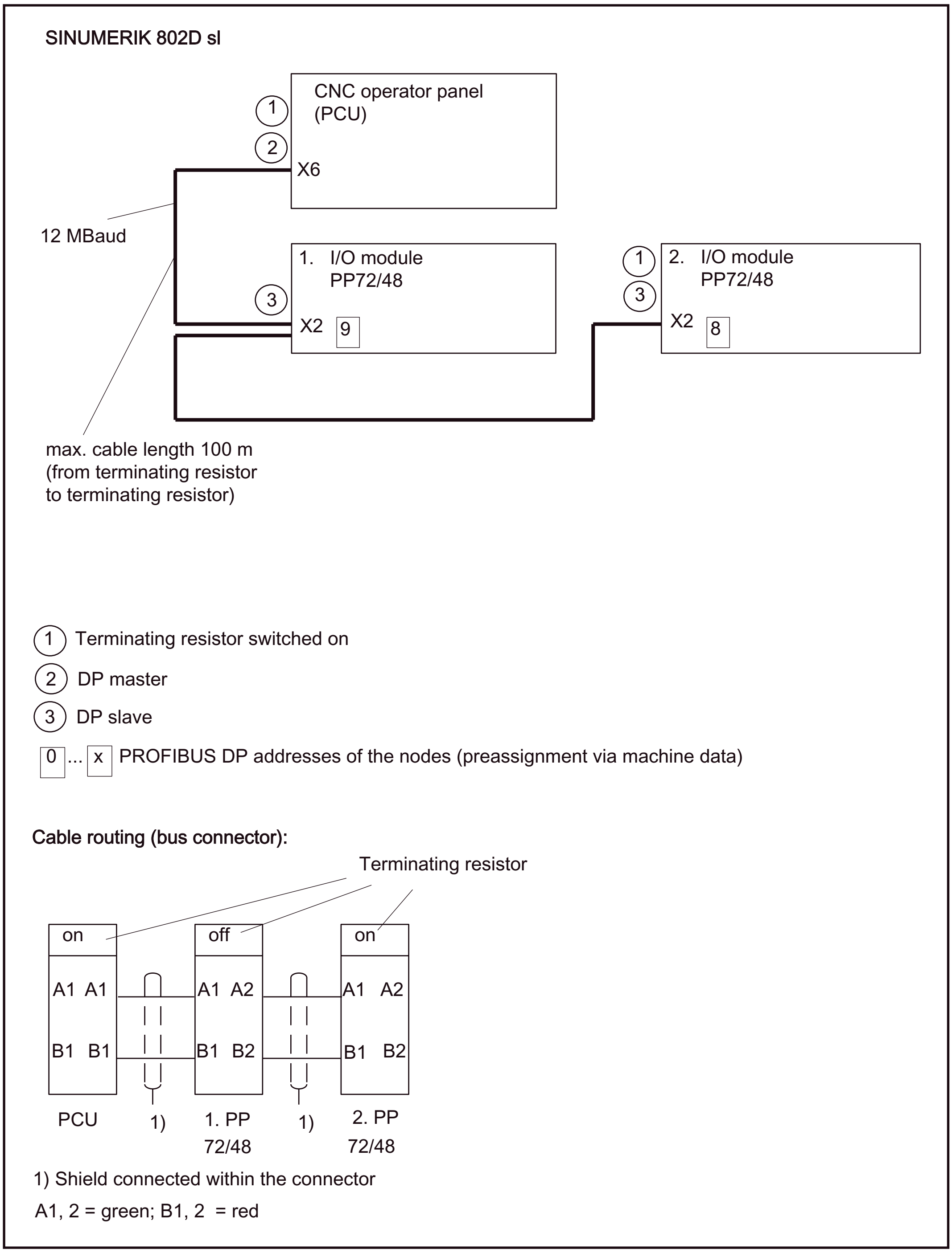


Figure 5-7 Networking example

5.13 Connecting an ADI4 module

References

The data necessary for connecting ADI 4 modules can be found in Manual ADI 4 - Analog Drive Interface for 4 Axes.

Note

For the configuration, load the ADI4.ini from the toolbox (in directory ..\special).

See also

Analog axis/spindle with TTL encoder via ADI4 (Page 141)

5.14 Connecting the DP/DP coupler

Cross-control PLC data interface

The DP/DP coupler is used to link two PROFIBUS DP networks together and to transfer data from the master of one network to the master of the other network.

For SINUMERIK 802D sl, 16 bytes are available for receiving and 16 bytes are available for sending. For more information, see the "SIMATIC DP/DP coupler" manual.

Note

DP/DP coupler as of Version B2
Order number: 6ES7158-0AD01-0XA0

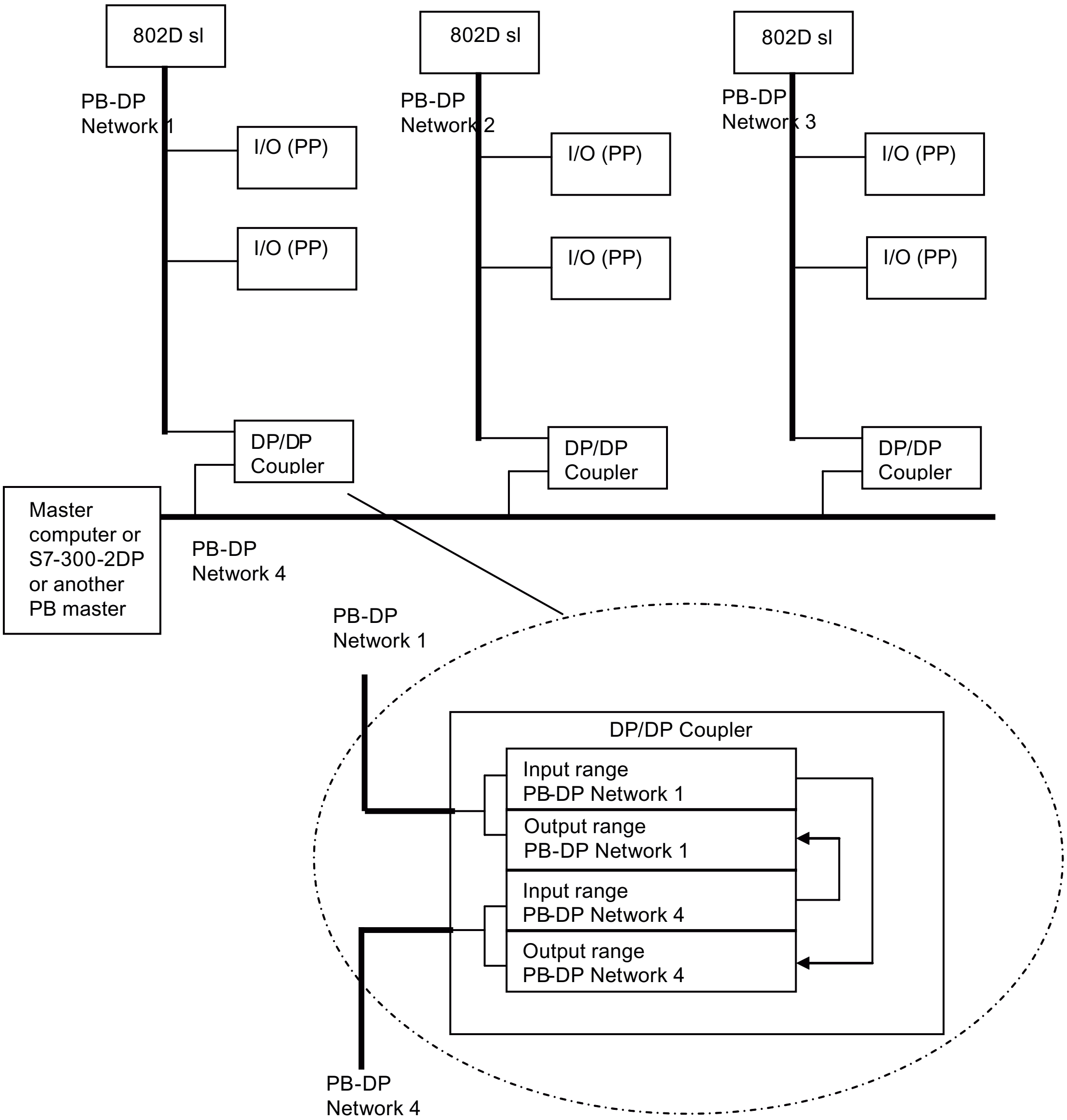


Figure 5-8 Using DP-DP coupler (example)

5.15 Connecting the SINAMICS drive to the DRIVE-CLiQ interface

Connect socket X1 or X2 on the CNC operator panel to socket X200 on the drive using the DRIVE-CLiQ signal line.

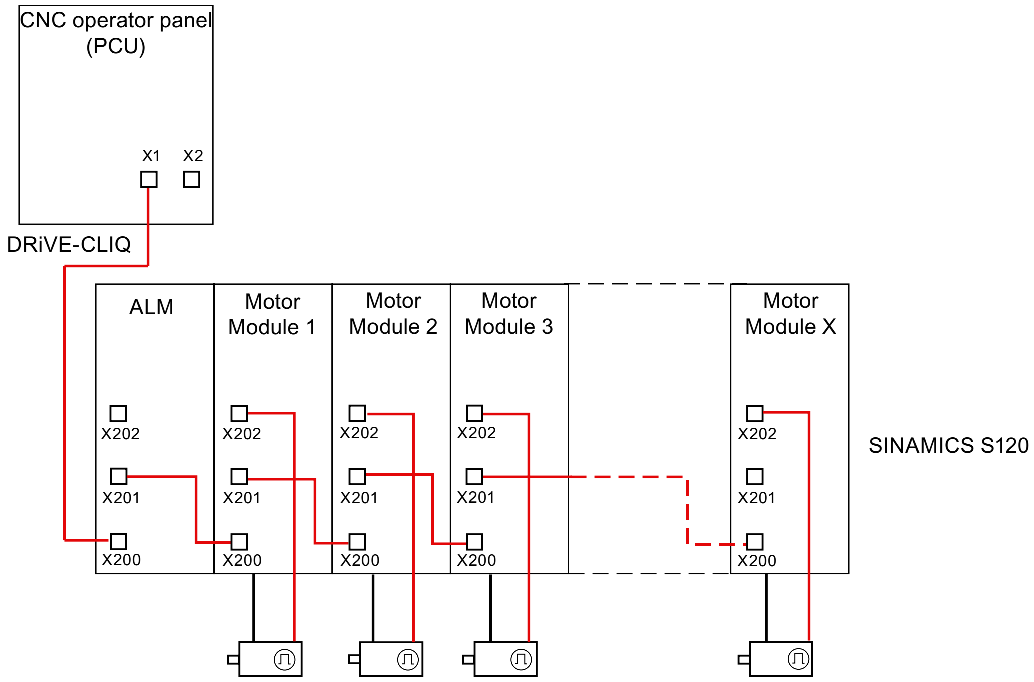


Figure 5-9 Connection with ALM (Active Line Module) and DRIVE-CLiQ

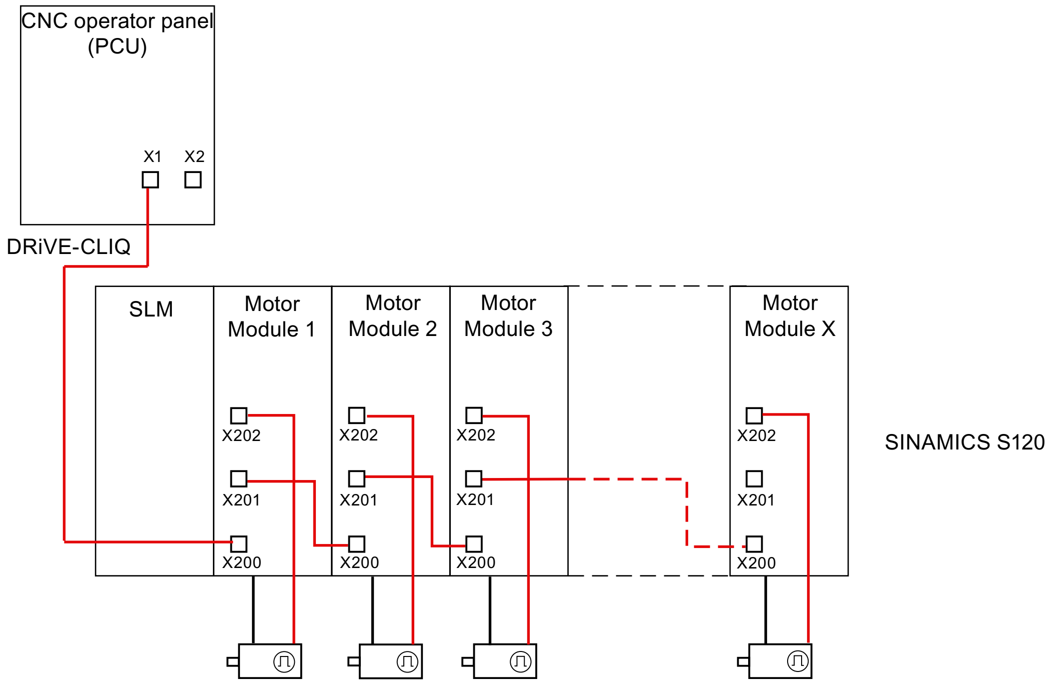


Figure 5-10 Connection with SLM (Smart Line Module) and DRIVE-CLiQ

With SMI motors (integrated measuring system interface), the connection is provided from the motor directly to X202 via the DRIVE-CLiQ line. In the case of direct measuring systems, the measuring system must be connected via an SMCxx module (where xx denotes the type of measuring system: e.g. SMC20 with incremental encoder or SMC30 with TTL encoder).

5.16 Connecting the DRIVE-CLiQ Hub Module DMC20

Connect socket X1 or X2 on the CNC operator panel (PCU) to socket X500 on the DRIVE-CLiQ Hub Module DMC20 using the DRIVE-CLiQ signal line.

Using sockets X501 to X505 on the DRIVE-CLiQ Hub Module DMC20, you can establish a point-to-point connection of motors with direct measuring systems to the DRIVE-CLiQ system, via the SMC20 module.

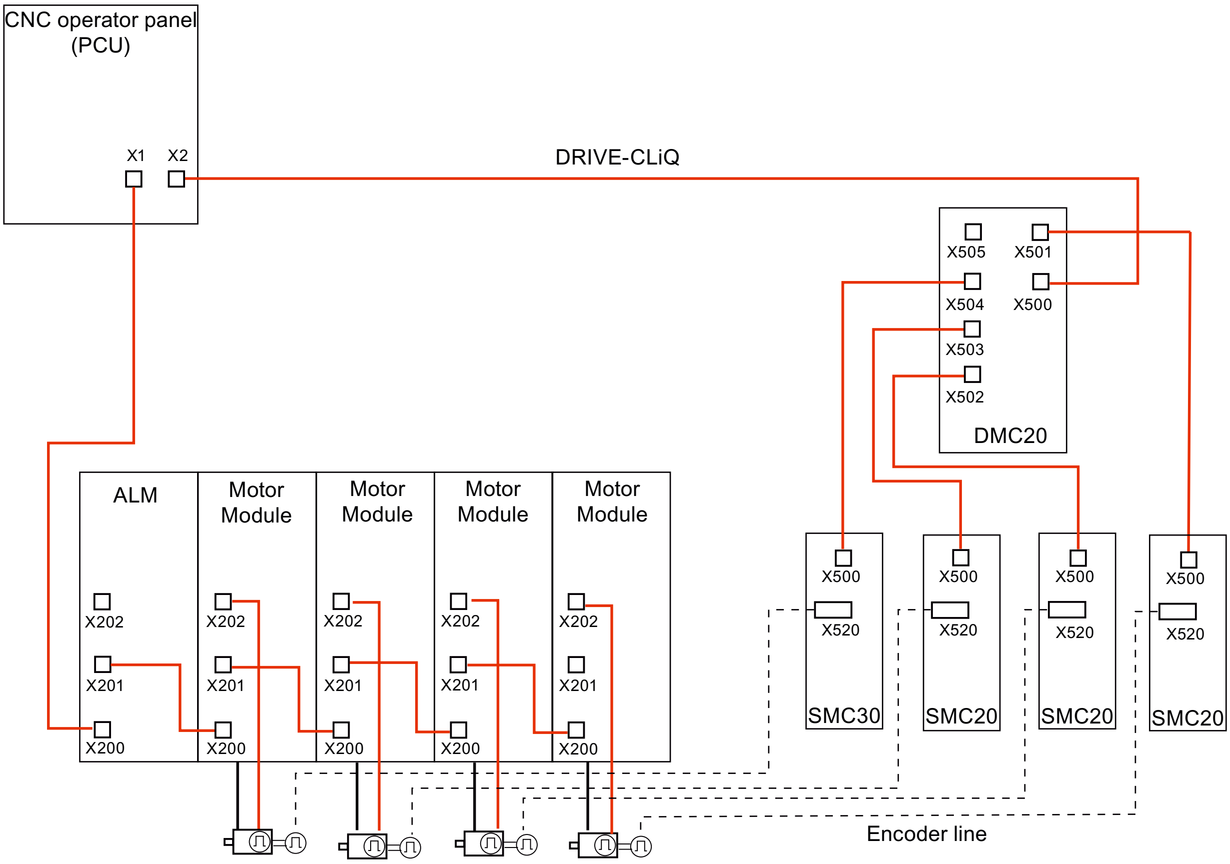


Figure 5-11 Connecting the DMC20

5.17 Connecting an analog spindle

The setpoint input of the analog spindle is output via the X701 interface on the MCPA module. It is possible to set the analog output to be unipolar.

The analog spindle is set up using machine data in the controller.

Connection of a directly mounted spindle actual-value encoder (TTL)

The TTL encoder requires an SMC30 module (DRIVE-CLiQ component). The encoder is connected via the SMC30 module interface X520 (encoder connection: TTL encoder with open-circuit monitoring).

NOTICE
The encoder power supply can be parameterized to 5 V or 24 V. The encoder may be destroyed if you enter the wrong parameters. Parameter location: Machine data of the component (drive object) that the SMC30 module uses to communicate with SINAMICS Encoder configuration: P404 bit20 set = 5 V P404 bit21 set = 24 V

See also

MCPA module interfaces (Page 40)

Interfaces for Sensor Module Cabinet 30 (SMC30) (Page 63)

5.18 Connecting digital inputs/outputs to the PCU

Connection cables

To connect the digital inputs and outputs, you should use flexible lines with a cross-section of at least 0.5 mm^2 .

If you only connect one line per connection, end sleeves are not absolutely necessary.

If you connect two lines per connection, lines with a cross-section between 0.25 and 0.75 mm^2 with end sleeve must be used.

Fasten the lines to the supplied screw terminals and insert the terminals into connectors X20 and X21 on the CNC operator panel.

Note

To ensure optimum interference immunity when connecting probes or BEROs, shielded lines must be used.

The max. line length is 30 m.

5.19 Connecting digital inputs/digital outputs to the PP72/48 I/O module

You can use the I/O interfaces X111, X222, X333 as digital inputs or digital outputs. To this end, fasten the insulation displacement connectors to the flat ribbon cable and route it from the I/O module to the terminal strip converter. The individual wiring can be performed on the terminal strip converters.

Remove the insulation from the cable end, insert the cable end (with end sleeve) into the screw terminal connection and tighten the fastening screw.

5.20 Connecting the machine control panel to the PP72/48 I/O module

Use two flat ribbon cables to connect the machine control panel (X1201 and X1202) to the PP72/48 I/O module (e.g. X111 and X222).

References

Catalog NC 802D sl, Catalog NC 61

Note

More information regarding the machine control panel and the pin assignment of the connectors X1201 and X1202 can be found in Section Interfaces of the machine control panel MCP 802D sl (Page 35).

5.21 Connecting shielded cables via the shield connection (PCU)

The shield of shielded signal lines must be connected to ground. The connection to ground is achieved by connecting the shield connection directly to the housing.

Shield connection

2 EMC shield clips provide the shield connection and are part of the scope of delivery of the CNC operator panel.

Mounting the shield connection

1. Remove the isolation from the shield as appropriate for the size of the EMC shield clip.
2. Place the shield on the housing in the appropriate place (see Figure below).
3. Screw the EMC shield clips onto the housing.

   Ensure firm fit of the cable on the housing.
4. For mechanical strain relief of the lines and cables, you can use the cable clamp or the EMC shield clips (see figure below).

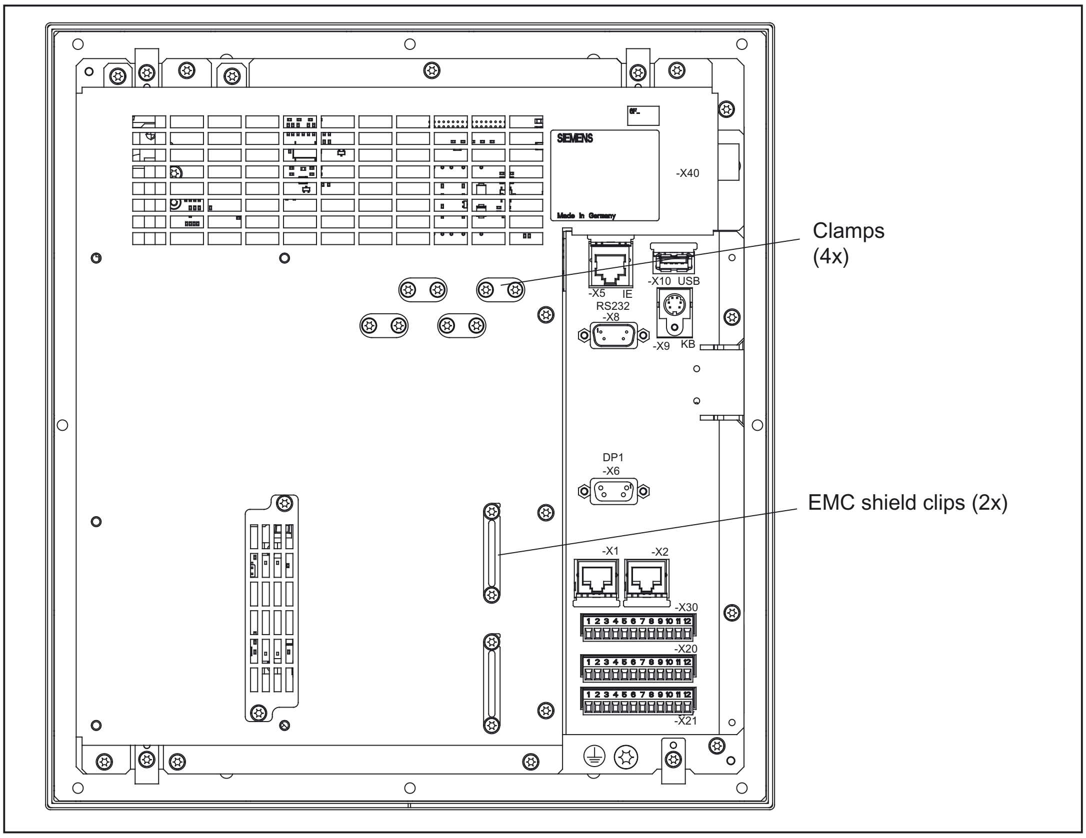


Figure 5-12 Connecting and securing shielded lines via the shield connection

Operation (hardware) 6

6.1 Control and display elements

Operator control elements

The defined functions are called up via the horizontal and vertical softkeys. For a description, please refer to this manual:

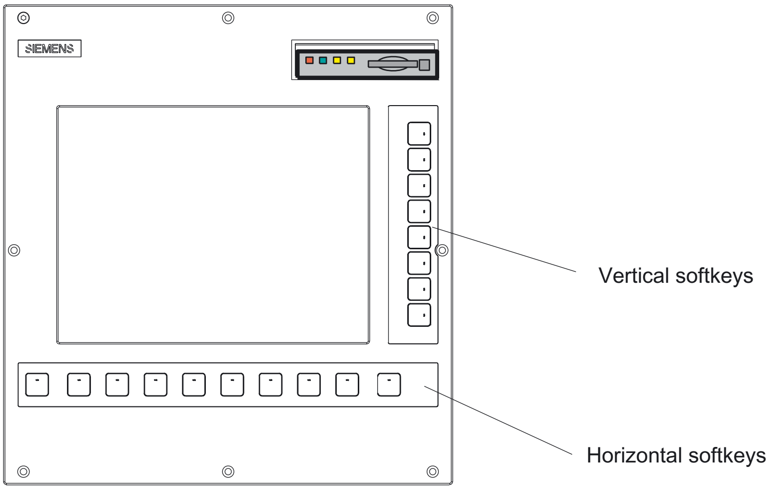
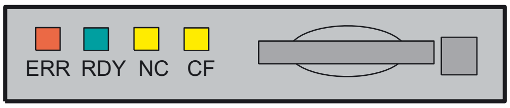


Figure 6-1 CNC operator panel

6.2 Error and status displays

LED displays on the CNC operator panel (PCU)

The following LEDs are installed on the CNC operator panel.



The individual LEDs and their functions are described in the table below.

Table 6- 1 Status and error displays

LED	Significance
ERR (red)	Serious error, remedy through power OFF/ON
RDY (green)	Ready for operation
NC (yellow)	Signoflife monitoring
CF (yellow)	Reading from/writing to CF card

References

You can find information on error description in the SINUMERIK 802D sl Diagnostics Manual

LED displays on the PP 72/48 I/O module

The following LEDs are installed on the I/O module:

Table 6- 2 Status displays

LED	Meaning
POWER (green)	Power supply of the electronic equipment ready for operation
READY (red)	I/O module ready for operation; but no cyclic data exchange with DP Master is performed
EXCHANGE (green)	I/O module ready for operation; cyclic data exchange with DP Master is performed
OVTEMP (red)	Overtemperature indication

7 Commissioning (general)

7.1 Initial commissioning (IBN)

Commissioning requirements

- You will need the following:
  - SINUMERIK 802D sl User Documentation
  - SINUMERIK 802D sl Description of Functions
  - SINUMERIK 802D sl Parameter Manual
  - A PC for commissioning and data backup
  - Tools installed from the Toolbox CD:
    RCS802 commissioning and diagnostics tool
    PLC802 programming tool
    802D sl configuration data
    PLC user library
    Commissioning software for SINAMICS S120 (for optimizing the drive)
    Adobe Acrobat Reader
- The mechanical and electrical installation of the system must be completed.

Commissioning sequence

To commission the SINUMERIK 802D sl, proceed as follows:

1. Check that the PCU boots.
2. Set the password
3. Load the language version and, if applicable, the language files.
4. Load technology
5. Set the general machine data
6. Set the PROFIBUS addresses
7. PLC commissioning
8. Drive commissioning
9. Set the axis/spindle-specific machine data
   - Match the encoder to the axis / spindle
   - Match the setpoint to the axis / spindle
10. Perform a dry run for the axes and for the spindle
11. Drive optimization

12. Complete the startup; perform a data backup.
13. Load user cycle masks

7.2 Access levels

Protection levels

The SINUMERIK 802D sl provides a concept of protection levels for enabling data areas. The various access authorizations control the protection levels 0 to 7 whereby **0** is the highest and **7** the lowest level.

The control system is delivered with default passwords for protection level 1 to 3.

Table 7- 1 Access level concept

Protection level	Locked by	range
0		Siemens, reserved
1	Password: SUNRISE (default)	Expert mode
2	Password: EVENING (default)	Machine manufacturer
3	Password: CUSTOMER (default)	Authorized operator, setter
4 to 7	No password/deleted password and user interface from PLC → NCK	Authorized operator, setter or appropriate graduations as desired

In the menus listed below the input and modification of data depends on the protection level set:

- Tool offsets
- Work offsets
- Setting data
- RS232 settings
- Program creation / program correction

The protection levels can be set for these function areas using the display machine data (USER_CLASS...)

Protection levels 1 ... 3

The protection levels 1 to 3 require a password. Passwords can be changed after activation. If they are no longer recognized, a reinitialization must be carried out (power up with default machine data). This will reset all passwords to their defaults according to the software release you have acquired.

The password remains set until it is reset with the "Reset password" softkey. **POWER ON** will **not**reset the password.

Protection levels 4 ... 7

Protection level 7 is set automatically if no password is set and no protection level interface signal is set. The protection levels 4 to 7 can be set from the PLC user program even without a password by setting the bits in the user interface.

Note

Setting of the access levels is described in the Programming and Operating Manual.

7.3 RCS802 tool

7.3.1 Interfaces and functions of the RCS802 tool

With the RCS802 tool (Remote Control System), you have a tool for your PC that will support you in your daily work with SINUMERIK 802D sl.

You establish the connection between the control system and the RCS802 tool on the PC using the following interfaces:

Interfaces

Table 7- 2 Interfaces

Interfaces	SINUMERIK 802D sl	RCS802 on PC
RS232	Is available for all product versions.	Are available.
Peer-to-peer Ethernet	Is available for all product versions.	Are available.
Ethernet network	Only available for SINUMERIK 802D sl pro.	Function that requires a license

Functions of the RCS802 tool with license key

Note

Importing the license key

You will only obtain the full functionality of the RCS802 tool after importing the license key RCS802.

Table 7- 3 Functions of the RCS802 tool that require a license

Function	RCS802 tool without license key			RCS802 tool with license key		
	RS232	P2P	Ethernet	RS232	P2P	Ethernet
Managing projects	Yes	Yes	Yes	Yes	Yes	Yes
Data exchange with SINUMERIK 802D sl	Yes	Yes	No	Yes	Yes	Yes
Commissioning SINUMERIK 802D sl	Yes	Yes	No	Yes	Yes	Yes
Setting-up a share drive	No	Yes *	Yes	No	Yes	Yes
Remote control	No	No	No	Yes	Yes	Yes
Screen shot	No	No	No	Yes	Yes	Yes

As user, use "peertopeer" with password "$PEER§PEER".

7.3.2 Setting the connections on the RCS802 tool

RCS802 tool

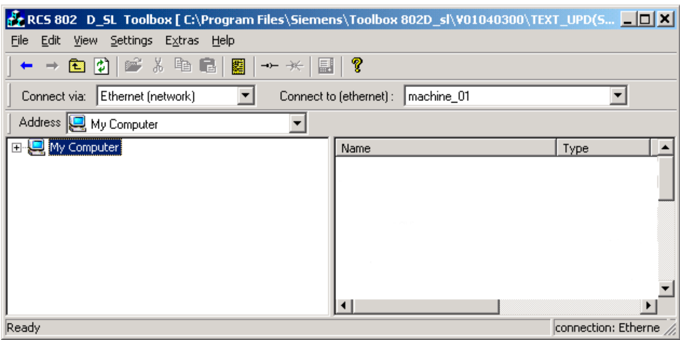


Figure 7-1 Explorer window of the RCS802 tool

After starting the RCS802 tool, you will be in OFFLINE mode. In this mode you only manage files on your PC.

In the ONLINE mode, the directory **Control 802** is also available. This directory makes data exchange with the control system possible. In addition, a remote control function is provided for process monitoring.

The ONLINE connections from the PC to the control are parameterized/activated via the "Setting" > "Connection" menu items in the "Connection Settings" dialog box.

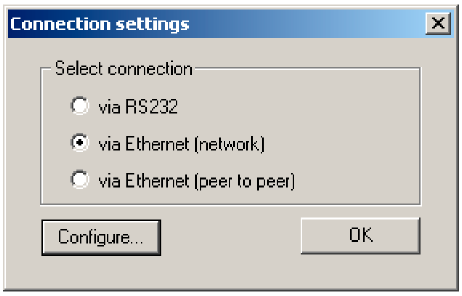
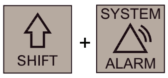


Figure 7-2 Connection Settings

Note

The RCS802 tool includes a detailed online help function. Refer to this help menu for further details e.g. establishing a connection, project management etc.

7.3.3 Establishing an RS232 connection to the control



You are now in the <SYSTEM> operating area.

PLC

Press the "PLC" softkey.

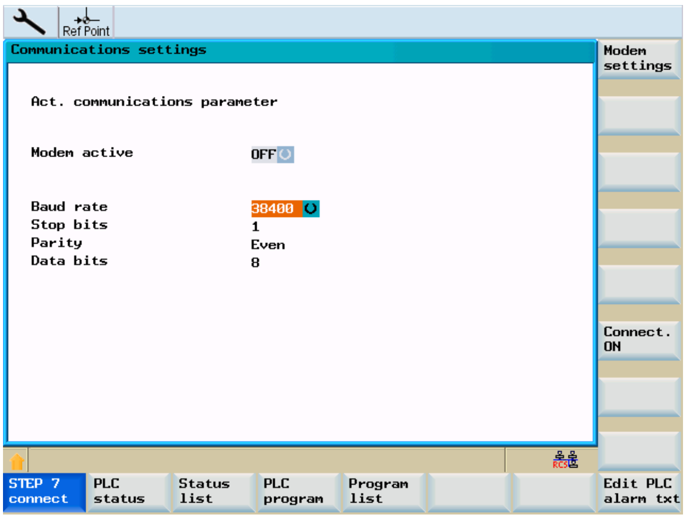


Figure 7-3 Communication settings RS232

Connect to STEP 7

Set the parameters for communication in the "STEP 7 Connect" dialog.

Connect. on

Activate the RS232 connection with the "Connect. ON" softkey.

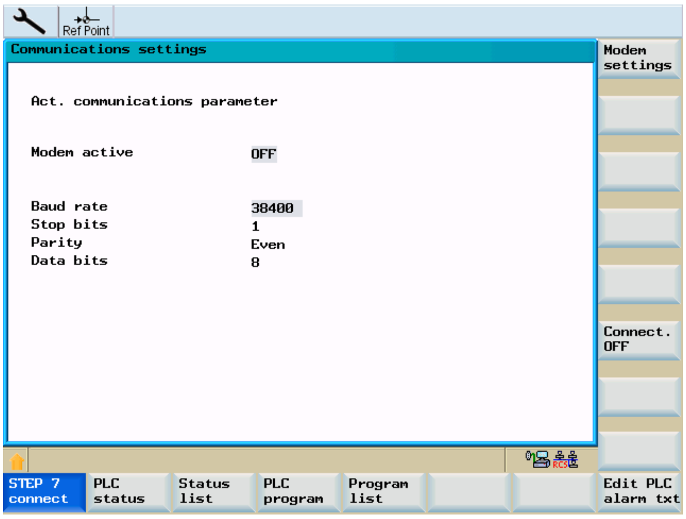
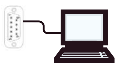


Figure 7-4 RS232 connection active

No modifications to the settings are possible in this state.

The softkey label changes to "Connect. OFF".



In the lower right corner of the screen, the icon shows that the connection to PC via the RS232 interface is active.

7.3.4 Network connections

7.3.4.1 Working on the basis of a network connection

The remote access (access to the control system from a PC or from a network) to the control system is disabled by default.

After a local user logs on at the PC, the **RCS tool** provides the following functions:

- Commissioning functions
- Data transfer (transfer of part programs)
- Remote control for the control system

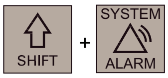
To grant access to a part of the file system, first share the relevant directories with other users.

Note

If you share directories with other users, the authorized network nodes are granted access to the shared files in the control system. Depending on the sharing option, the user can modify or delete files.

7.3.4.2 User log in - RCS log in

For the Ethernet connections, you must first log on to the control system as a user.



RCS log-in

In the <SYSTEM> operating area, select the "RCS Connect" softkey. The user log-in input screen will appear.

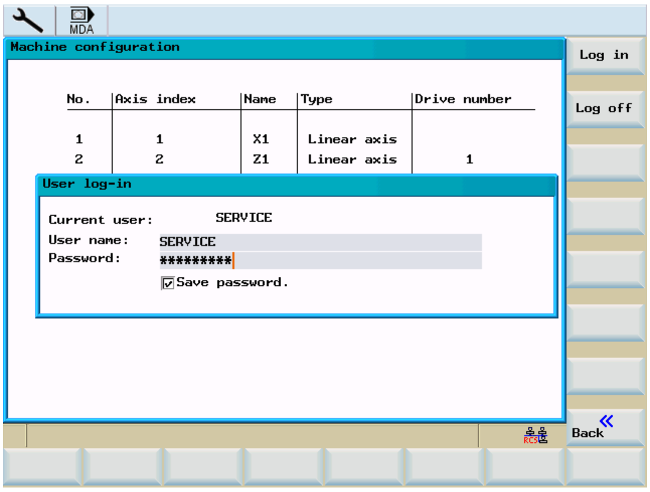


Figure 7-5 User log-in

Logon

Type user name and password into the appropriate input fields and select the "Log in" softkey to confirm your input.

After successful log-in, the user name is displayed in the **Current user** line.

Select the "Back" softkey to close the dialog box.

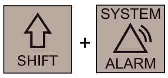
Note

This log-in simultaneously serves for user identification for remote connections.

Logoff

Press the "Log off" softkey. This will log out the current user, all user-specific settings are saved, and any enables already granted are canceled.

7.3.4.3 Establishing a peer-to-peer Ethernet connection to the control



You are now in the <SYSTEM> operating area.

Service display

Press the softkeys "Service display" >"Service control".

Service control

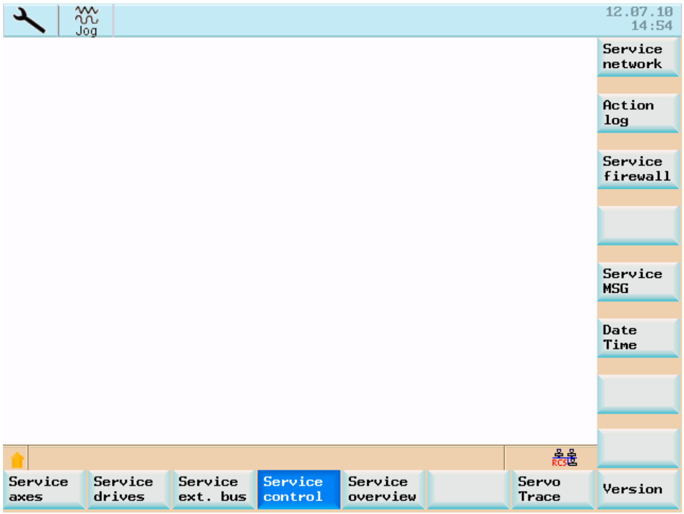


Figure 7-6 "Service control"

Service network

Press "Service network".

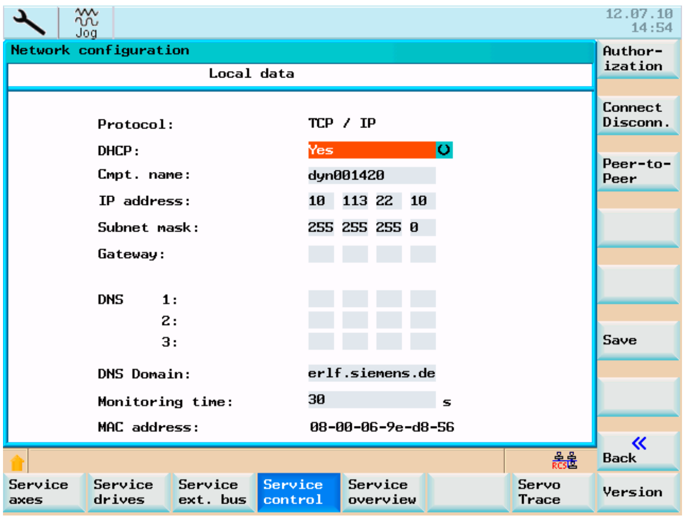


Figure 7-7 "Network configuration" main screen

Peer-to-peer

Press the "Peer-to-peer" softkey.

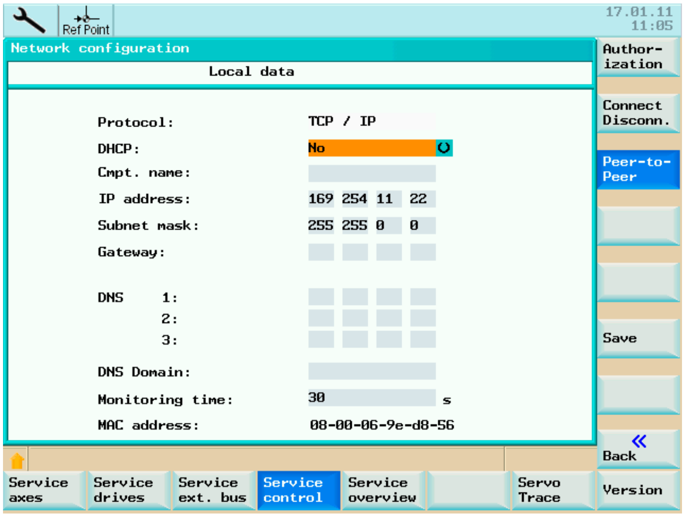


Figure 7-8 "Peer-to-peer"

The following message is shown on the HMI:

"Connection is set up"

- IP Address: 169.254.11.22
- Subnet mask: 255.255.0.0

Note

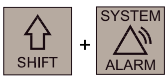
The IP address and subnet mask shown are fixed values.

These values cannot be changed.

Peer-to-peer

Using the "Peer-to-peer" softkey you can cancel the Ethernet peer-to-peer connection.

7.3.4.4 Operating sequence to establish an Ethernet connection to the control



- You are now in the <SYSTEM> operating area.

Service display

Service control

- Press the softkeys "Service display" >"Service control".

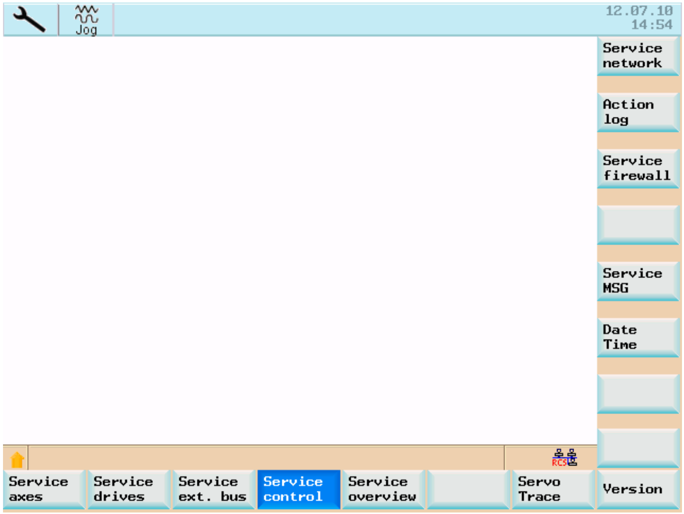


Figure 7-9 "Service control"

Service network

Press the softkey "Service network" (only available for SINUMERIK 802D sl pro), to enter the network configuration.

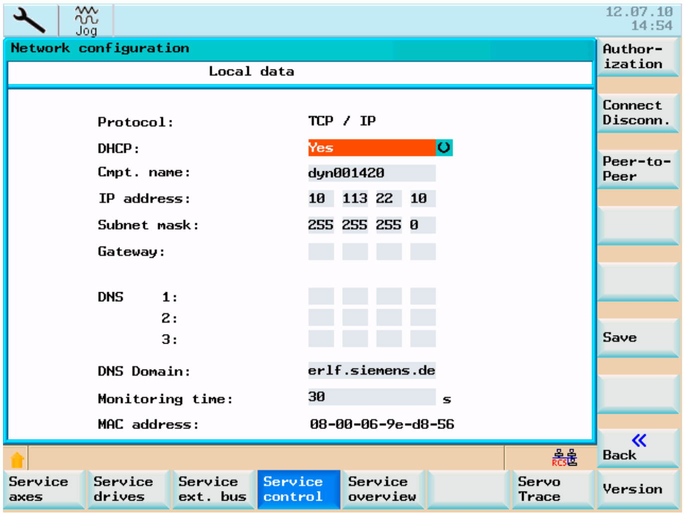


Figure 7-10 Network configuration

In this view you can enter the IP address.

Save

The Ethernet connection is established with "Save".

References

SINUMERIK 802D sl Programming and Operating Manual; Network Operation

Initial start-up 8

8.1 Turning on and booting the control system

Procedure

- Check the system visually for:
  - Correct mechanical installation with secure electrical connections
  - Supply voltage
  - connection of shielding and grounding.
- Connect the control system (booting in the normal mode)

Booting in the normal mode

When the control system is turned on, the boot sequence is displayed on the control system with all its individual phases. Once the start screen of the user interface has appeared, the booting sequence is completed.

Booting of the control in commissioning mode

After POWER ON and once the operating system has been started, the words "SINUMERIK Solution line" are displayed filling the whole screen. Once these words disappear, press the <SELECT> key.

The "Startup Menu" is displayed. Use the arrow key to select an appropriate power-up/startup mode and press <INPUT> to confirm.

If there is no password set, the following modes are available:

- "Normal startup"

  If this option is chosen, the control system will boot with the last machine data set and the previously loaded programs.
- "Reload saved user data"

  The user data (machine data, programs, etc.) that was backed up to the flash memory are accepted as the current data and the boot up is carried out.
- "Software update"

  In this case, the control system will not boot at all. The software can only be updated if a CompactFlash card with a software update is inserted in the slot for the CompactFlash card.

If there is a password set, the following modes are available:

- "Normal startup"
- "Reload saved user data"

- "Startup with default data" (only displayed if protection level 1 or 2 is set)

  If this option is chosen, the control system will boot with default machine data.
- "PLC stop"

  Select PLC Stop while the control system is booting if PLC Stop can not be triggered via the user interface any more.
- "PLC overall reset/default PLC program"

  All PLC variables are reset, a NOP (no operation) program is loaded.
- "HMI startup with default data"

  The HMI will power up with default display machine data.
- "Remove drive data"

  The drive machine data is reset and the factory setting is loaded.
- "Remove drive data / default data / startup with default data 2"

  The drive machine data is reset and the default data is loaded.
- "Software update"

  In this case, the control system will not boot at all. The software can only be updated if a CompactFlash card with a software update is inserted in the slot for the CompactFlash card.

8.2 Language setting and file management

As standard, the control system contains the following files for:

- User interface language (e.g. English, German, Simplified Chinese)
- Alarm texts
- Help texts

The language can be, e.g., English, German, Simplified Chinese.

In addition to these files, the RCS802 tool can be used to expand/modify user-specific files. The files are then transferred to the control system using the RCS802 tool.

Use the RCS802 tool to create a project and then edit the relevant files in it.

8.2.1 Creating and Editing Projects

Requirements

The RCS802 tool and the toolbox are installed on the PC/PG.

Operating sequence to create a project

1. Start RCS802 on the PC.
2. Select the Toolbox version in the menu bar under "Settings" > "Toolbox" > "Select Version And Project" and press the "Project" button (1).
3. In the menu screen (2) press the "New" button. The window "Create new project" is opened (3).
4. Enter a name for the new project.
5. Add the languages you require.
6. Click "OK" to confirm your input.

   The system creates the new project and displays it in the project overview (4).

Note

To activate the selected project, click "OK" in the project overview (4).

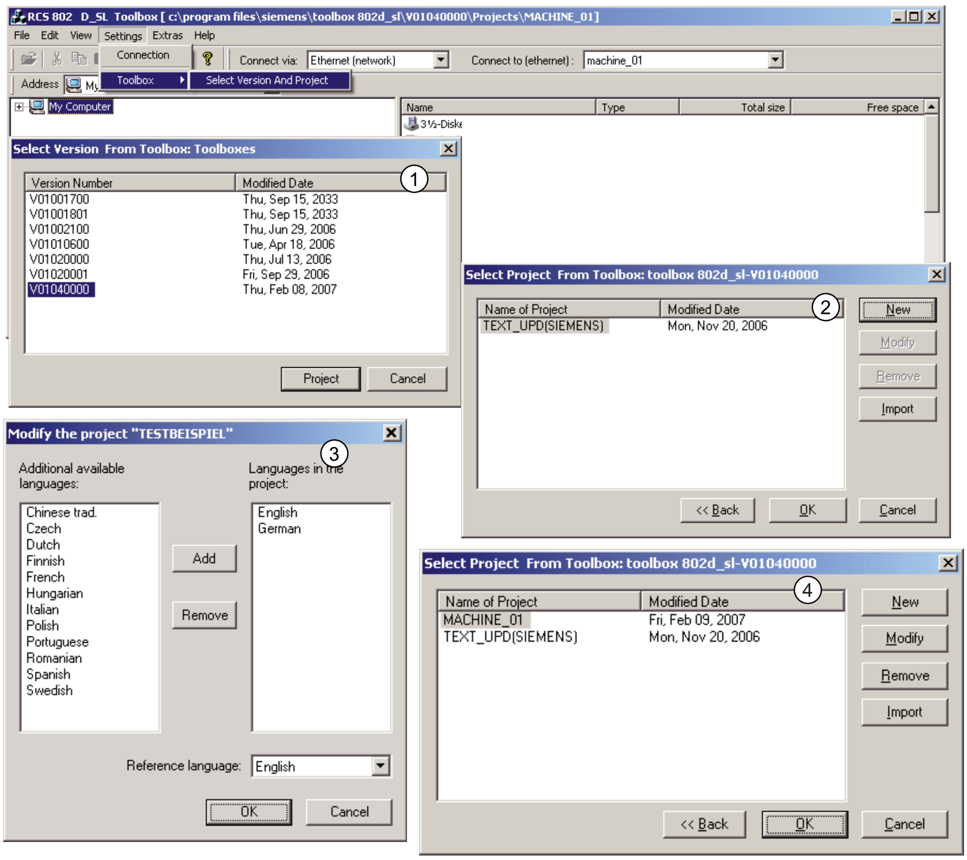


Figure 8-1 Creating a project

Operating sequence to edit a project

1. Start RCS802 on the PC.
2. Select the Toolbox version in the menu under "Settings" > "Toolbox" > "Select Version And Project" and press the "Project" button (1).
3. Select the project to be edited in the project overview (4) and press "Modify".

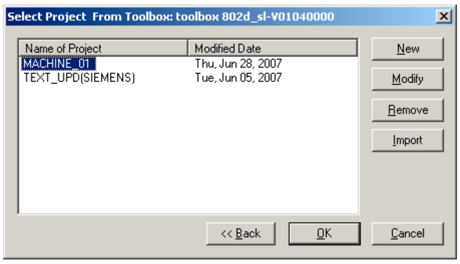


Figure 8-2 Dialog box "Select Project"

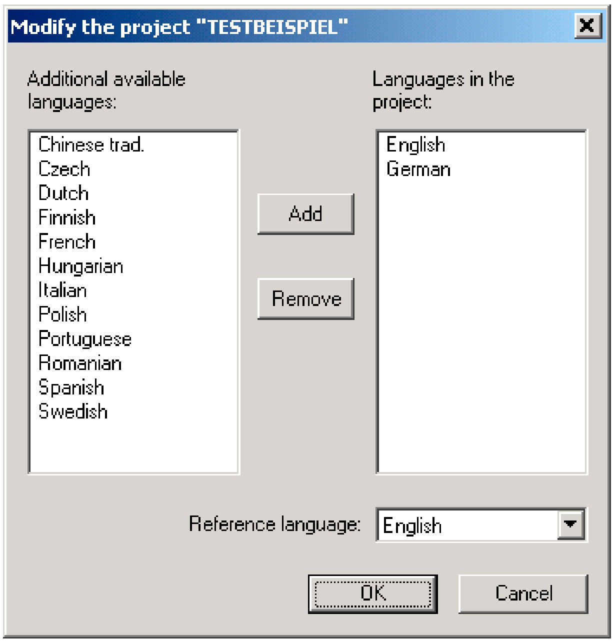


Figure 8-3 Editing a project

4. In the subsequent menu display, you can add languages to or remove languages from the project.

8.2.2 Managing the HMI online help

8.2.2.1 Overview, managing the HMI online help

To edit the HMI online help, in the main screen menu click "Extras" > "Toolbox Manager" > "Generate Helpsystem".

Generate Helptext

The languages of the active toolbox project and their help files are displayed in the dialog box "Generate Helptext".

You can extend or remove existing help files and create new help files.

Further, the help system can be generated and loaded into the control.

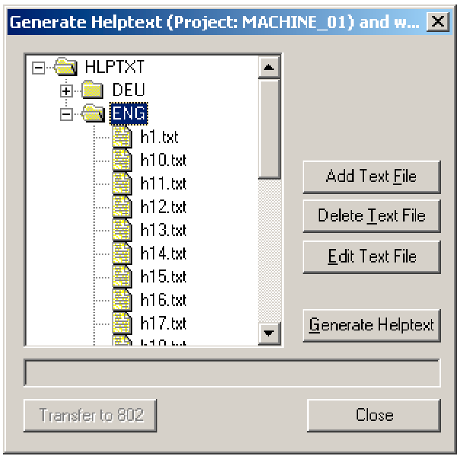


Figure 8-4 Dialog box "Generate Helptext"

Supplementing own help texts

A new help file is created in the selected language and opened with a text editor by clicking on "Add Text File". Add your new help subjects. Save the changes before closing the text editor.

Changing existing help texts

In the dialog box "Generate Helptext", select the file to be changed and click on "Edit Text File". The help file is opened in a text editor. Make the changes and save these before closing the text editor.

8.2.2.2 Transferring help texts to the SINUMERIK 802D sl

Operating sequence

1. Ensure that the RCS802 is connected to the SINUMERIK 802D sl.
2. Select the language to be transferred.
3. In the "Generate Helptext" dialog box, click "Generate Helptext".
4. To start transfer of the help texts, click on "Transfer to 802".

   It can take several seconds to update the HMI.
5. If the dialog box "Confirm overwriting of files" is displayed, acknowledge this by clicking on "Yes".

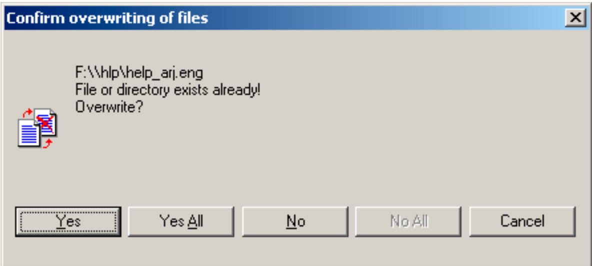


Figure 8-5 Dialog box "Confirm overwriting of files"

6. After the help texts have been transferred, a prompt will ask if you would like to open a "logfile". Click "Yes".

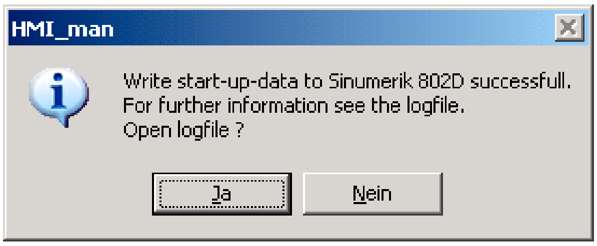


Figure 8-6 Open logfile prompt

7. You can see which help texts have been transferred. Click "OK".

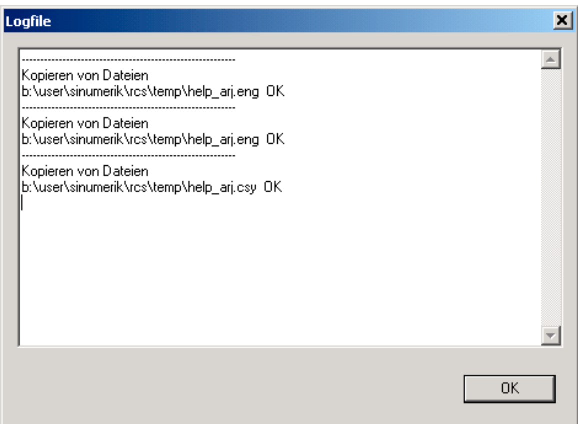


Figure 8-7 Logfile

8. After the HMI online help has been successfully updated, acknowledge the dialog box "Write help-text file to SINUMERIK 802D successful" with "OK".

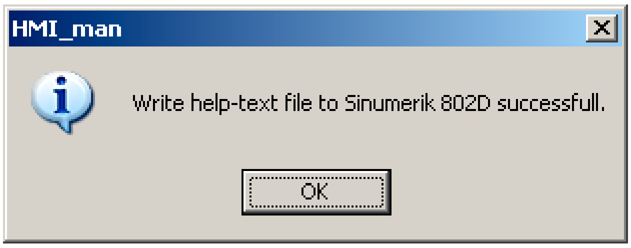


Figure 8-8 Dialog box "Write help-text file to SINUMERIK 802D successful"

8.3 Setting the technology

Note

The SINUMERIK 802D sl is delivered with default machine data. In the next step, the appropriate setup file **must** be loaded from the toolbox into the control system.

The following technology can be configured using setup files:

- Turning
- Milling
- Cylindrical grinding
- Surface grinding
- Nibbling

From the installed toolbox, the setup file relevant for the technology is to be used in conjunction with the control system variant (value, plus, pro).

The setup file must be loaded during the first commissioning after booting of the control system, but prior to the general configuration.

Note

Please always observe the readme file supplied with the "Toolbox". It provides up-to-date information.

Note

The file sgud*.ini is available in the toolbox directory "Techno". This file can be downloaded if needed in the application. The GUD definitions are activated at the end of the loading procedure. Here the following values are deleted:

- Global user data (GUD)
- Tool compensation data
- Protection zones
- R parameters
- Work offsets/FRAME
- Leadscrew error compensation

Before loading, back up the data on the control as described in External data backup (Page 425)(and following pages) and reload it afterwards!

Turning configuration

- setup_T.arc

  Contains the complete setup for the turning technology, including standard cycles

- setTra_T.arc

  Contains the complete setup for the turning technology, including standard cycles with a second spindle for the options TRANSMIT, TRACYL and the use of milling cycles.
  Note: Only load one of the two setup files.

- trafo_T.ini

  Text file - only for the turning technology. These settings are supplemented by the turning technology with machine data for the second spindle and for the options TRANSMIT/TRACYL.

  Note: If you use milling cycles, load the cycles.spf file into the control system.

- trafo_Mx.ini

  Text file for milling applications on the lathe. These settings are supplemented by the lathing technology with machine data for the second spindle and for the option TRACYL.

  – trafo_MA.ini: For rotary axis A
  – trafo_MB.ini: For rotary axis B
  – trafo_MC.ini: For rotary axis C

  Note

  You can find the trafo_MA.ini, trafo_MB.ini, and trafo_MC.ini files in the toolbox under the path: .../Toolbox / 802D_sl/V.../Techno/Milling/Config_Siemens.

- setISO_T.arc

  Binary file for switching from SIEMENS mode to ISO mode turning (B code) with simulation axes and spindle

- isoTra_T.arc

  Binary file for switching from SIEMENS mode to ISO mode turning (B code) with second spindle

- ISO_A_T.ini

  Text file for switching from ISO mode B code to ISO mode A code

- ISO_C_T.ini

  Text file for switching from ISO mode B code to ISO mode C code

- ISO_B_T.ini

  Text file for switching back from ISO mode A code or C code to B code

- turnG22.ini

  Text file for switching on the function "STORED STROKE CHECK FUNCTION".

- setup_Tplus_MM.arc and setup_Tpro_MM.arc

  Configuration files for Manual Machine Plus Turning

Milling configuration

- setup_M.arc

  Contains the complete setup for the milling technology, including standard cycles

- setISO_M.arc

  Binary file for switching from SIEMENS mode to ISO mode milling with simulation axes and spindle

- ISOG70_M.ini (to use G70/G71 for INCH/METRIC)

  Text file allowing the function G70/G71 to be used for INCH/METRIC switchover in ISO mode milling as well

- millG22.ini

  Text file for switching on the function "STORED STROKE CHECK FUNCTION"
- mold.ini

  Contains standard presettings for mold making applications (only for SINUMERIK 802D sl pro)

Cylindrical grinding configuration:

- setup_G_C.arc

  Contains the complete setup for the external cylindrical grinding technology with Cartesian X-Z axes, including standard cycles
- setup_G_C_inc.arc

  Contains the complete setup for the external cylindrical grinding technology with non-Cartesian X-Z axes (inclined axis with fixed angle), including standard cycles

Surface grinding configuration

- setup_G_S.arc

  Contains the complete setup for the surface grinding technology, including standard cycles

Nibbling configuration

- setup_N.arc

  Contains the complete setup for the nibbling technology with mechanically coupled die
- setup_N_MC.arc

  Contains the complete setup for the nibbling technology with die coupled via servo axis

Configuration ADI 4

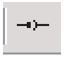
- adi4.ini (in directory ..\SPECIAL)

  Machine data for setting up analog setpoint output via ADI4.

Sequence of operations

- Make the connection between the PG/PC and the control system (CNC operator panel).
- Turn on the control system and wait until the control system has completed its booting sequence without errors. In the <SYSTEM> operating area, set the password to protection level 2 or higher:

- Start RCS802 on the PG/PC and use the following button to make the connection between the PG/PC and the control system:

- Copy/paste the icon for the setup file to the "Data" folder on drive A of the 802D sl.

  If the toolbox has been installed in the default directory, the setup files can be found, for example, under

  C:\Program Files\Siemens\Toolbox 802D_sl\V01xxyyzz\TECHNO\[Technology]\CONFIG_xx\[Version].
- During transfer, the NCK is automatically powered up several times, and the HMI is restarted once the transfer is complete.
- The SINUMERIK 802D sl is now preset to the required technology.

8.4 Activate "User function" softkey

A softkey is reserved for integrating user dialogs in the operating areas <POSITION>, <OFFSET PARAMETER> and <SYSTEM>.

The softkey displays, as default, the "User function" text.

The softkey is automatically activated when loading script files, and is linked with the corresponding script. The script files are in the following directory:

F:\appl\

Changing a softkey text

You can change the softkey text for the operating areas <POSITION>, <OFFSET PARAMETER> and <SYSTEM>.

Enter your own text in the "<menu>" tag after the "caption" attribute.

Example

```
<menu>
 <menu name="main" caption="my%softkey>
 ...
 ...
</menu>
```

<POSITION> operating area

The following files are available for the <POSITION> operating area:

File name	Operating mode	Softkey	Menu name
ma_jog.xml	JOG	Horizontal softkey 6	main
ma_mda.xml	MAD	Horizontal softkey 1	main
ma_auto.xml	AUTO	Horizontal softkey 6	main

<OFFSET PARAMETER> operating area

The following file is available for the <OFFSET PARAMETER> operating area:

File name	Softkey	Menu name
pa.xml	Horizontal softkey 4	main

<SYSTEM> operating area

The following file is available for the <SYSTEM> operating area:

File names	Softkey	Menu name
dg.xml	Horizontal softkey 8	main

Example

To exit the user function, use the "close_from" tag.

```
<DialogGui>
<menu name = "main">
 <open_form name = "main_pa_add_screen_form" />
 <softkey_ok>
 <!-- Close form has to be used to return to the standard operating area. -->
 <close_form />
 </softkey_ok>
</menu>
<form name = "main_pa_add_screen_form" xpos ="12" ypos="100" width="500" height="240">
 <init>
 <caption>additional screen parameter (pa.xml)</caption>
 <data_access type="true" />
 <control name = "c1" xpos = "322" ypos = "34" refvar="nck/Channel/Parameter/R[0]" hotlink="true" />
 <control name = "c2" xpos = "322" ypos = "54" refvar="nck/Channel/Parameter/R[1]" hotlink="true" />
 <control name = "c3" xpos = "322" ypos = "74" refvar="nck/Channel/Parameter/R[2]" hotlink="true" />
 <control name = "c4" xpos = "322" ypos = "94" refvar="nck/Channel/Parameter/R[3]" hotlink="true" />
```

```
<control name = "c5" xpos = "322" ypos = "114" refvar="nck/Channel/Parameter/R[4]" hotlink="true" />
<control name = "c6" xpos = "322" ypos = "134" refvar="nck/Channel/Parameter/R[5]" hotlink="true" />
<control name = "c7" xpos = "322" ypos = "154" refvar="nck/Channel/Parameter/R[6]" hotlink="true" />
<control name = "c7" xpos = "322" ypos = "154" refvar="nck/Channel/Parameter/R[6]" hotlink="true" />
<control name = "c8" xpos = "322" ypos = "174" refvar="nck/Channel/Parameter/R[7]" hotlink="true" />
<control name = "c9" xpos = "322" ypos = "194" refvar="nck/Channel/Parameter/R[8]" hotlink="true" />
<control name = "c10" xpos = "322" ypos = "214" refvar="nck/Channel/Parameter/R[9]" hotlink="true" />
</init>
<paint>
<text xpos = "23" ypos = "34">R parameter 0</text>
<text xpos = "23" ypos = "54">R parameter 1</text>
<text xpos = "23" ypos = "74">R parameter 2</text>
<text xpos = "23" ypos = "94">R parameter 3</text>
<text xpos = "23" ypos = "114">R parameter 4</text>
<text xpos = "23" ypos = "134">R parameter 5</text>
<text xpos = "23" ypos = "154">R parameter 6</text>
<text xpos = "23" ypos = "174">R parameter 7</text>
<text xpos = "23" ypos = "194">R parameter 8</text>
<text xpos = "23" ypos = "214">R parameter 9</text>
</paint>
</form>
</DialogGui>
```

8.5 Input of the machine data

Overview

The most important machine data of the individual subareas are listed here to assist you. The detailed description of the machine data and interface signals is given in the Parameter Manual with cross-references to the section on function description.

Note

The machine data is preset by loading the technology files in such a manner that a change to the values is only required in exceptional cases.

Entering the machine data (MD)

Before you can enter the machine data, the password for protection level 2 must be set.

Use the relevant softkey to select the following machine data areas and to change the machine data if necessary:

- General machine data MD10000 to 19999
- Channel machine data MD20000 to 29999
- Axis machine data MD30000 to 39999
- Display machine data MD1 to 999
- Setting data
  - General setting data
  - Channel-specific setting data
  - Axis-specific setting data
- Drive machine data r0001 to r9999 (read-only)

  p0001 to p9999 (read/write)

The data you have entered are written to the data memory immediately. An exception is the drive machine data. To save the drive machine data permanently, set the parameter p971 in the SINAMICS for each individual drive object separately or set p977 of the CU_I to "1" and wait until it is reset automatically. If you forget to save the data, the old data is effective again after the next drive reset.

The machine data is activated depending on the machine data property "Activated".

8.6 Activating the high-speed digital inputs/digital outputs

Setting the Machine Data

The high-speed digital inputs/outputs are activated using the following machine data:

MD10350 $MN_FAST_DIG_NUM_INPUTS
MD10360 $MN_FAST_IO_DIG_NUM_OUTPUTS
MD10366 $MN_HW_ASSIGN_DIG_FASTIN[0]
MD10368 $MN_HW_ASSIGN_DIG_FASTOUT[0]

Example

The example describes the use of NCK inputs/outputs ($A_IN[9-16]; $A_OUT[9-16]) via MCPA.

The machine data MD10350 and MD10360 will be or are already set to 2. The machine data MD10366 and MD10368 must be input as follows (MCPA module as optional module 1 of the local bus):

4. Byte: 00 Segment number for LOCALBUS
3. Byte: 01 Module number (MCPA)
2. Byte: 01 Interface-module number
1. Byte: 01 I/O byte number

Table 8- 1 Example of machine data setting

Machine data		Value	Explanation
MD10350	$MN_FAST_IO_DIG_NUM_INPUTS	2	Number of input bytes *)
MD10360	$MN_FAST_IO_DIG_NUM_OUTPUTS	0x2	Number of output bytes *)
MD10366	$MN_HW_ASSIGN_DIG_FASTIN[0]	0x00 01 01 01	Hardware assignment MCPA
MD10368	$MN_HW_ASSIGN_DIG_FASTOUT[0]	0x00 01 01 01	Hardware assignment MCPA

*) The first I/O byte is reserved for SINAMICS.

The example shows that a total of 2 bytes for inputs/outputs are available.

A value of 2 for the input byte (analog output byte) means:

- 1x SINAMICS input/output byte
- 1x customer-specific input/output byte

As the MCPA inputs/outputs are always assigned to permanent PCU I/O addresses within part programs, they can be addressed bit-by-bit on a customer-specific basis as follows:

- Inputs: $A_IN[9] ... $A_IN[16]
- Outputs: $A_OUT[9]...$A_OUT[16]

```
...
N100 R1= $A_IN[9]        ; Reading digital input 1 of the MCPA module
N200 $A_OUT[16] = 1      ; writing a 1 on the last digital output of the MCPA module
N300 R2=$A_OUT[16]       ; Reading of the output Bit8
```

The SINAMICS inputs/outputs are addressed via the following bits:

- Inputs: $A_IN[1]...$A_IN[8]
- Outputs: $A_OUT[1]...$A_OUT[8]

Note

Where PCU digital inputs/outputs are being used in a component configuration (without SINAMICS drive modules), NCK inputs/outputs ($A_IN[1-8]; $A_OUT[1-8]) can only be used under certain circumstances.

In this case please contact the hotline (see Preface "Technical Support").

8.7 Setting the PROFIBUS addresses

Each bus node must be uniquely identified by means of the assignment of a PROFIBUS-DP address on the PROFIBUS-DP. Each PROFIBUS-DP address must be assigned only once on the bus.

Table 8- 2 Setting the PROFIBUS address

MD11240[2]	PROFIBUS node (slave)	PROFIBUS address
0	PP module 1	9
	PP module 2	8
	PP module 3	7
	Additional preconfigured PROFIBUS node: DP-DP coupler	6

PCU

The PCU is the master on the PROFIBUS. The address cannot be changed

PP 72/48

The PP 72/48 I/O module is the slave on the PROFIBUS. max. three PP modules can be connected. The PROFIBUS-DP address is set using DIL switch S1 (on the PP module). Use a screwdriver to set the PROFIBUS-DP address. It results from adding the switches that are in the "ON" position (left).

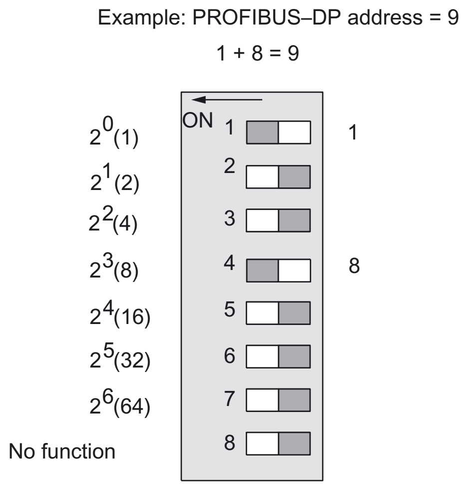


Figure 8-9 Setting the PROFIBUS-DP address on PP72/48

Table 8- 3 Setting the PROFIBUS address on the PP72/48

PROFIBUS address	DIL switch S1 (PP module)
9 (default setting) (PP module 1)	1 + 4 = ON 2 + 3 + 5 + 6 + 7 + 8 = OFF
8 (PP module 2)	4 = ON 1 + 2 + 3 + 5 + 6 + 7 + 8 = OFF
7 (PP module 3)	1 + 2 + 3 = ON 4 + 5 + 6 + 7 + 8 = OFF

Note

You will need to shut down and restart in order to apply the new PROFIBUS node address.

Using three PP72/48 I/O modules

If three PP72/48 I/O modules are used, the assignment of the input/output bytes will be as follows:

Table 8- 4 Assignment of the input/output bytes

1. PP 72/48 I/O module, PROFIBUS-DP address 9			
Connector	X111	X222	X333
Input byte	0...2	3...5	6...8
Output Byte	0...1	2...3	4...5
2. PP 72/48 I/O module, PROFIBUS-DP address 8			
Connector	X111	X222	X333
Input byte	9...11	12...14	15...17
Output Byte	6...7	8...9	10...11
3. PP 72/48 I/O module, PROFIBUS-DP address 7			
Connector	X111	X222	X333
Input byte	18...20	21...23	24...26
Output Byte	12...13	14...15	16...17

DP/DP coupler

The DP/DP coupler is the slave on the PROFIBUS. One DP/DP coupler can be connected at the most. The PROFIBUS-DP addresses are set via the DIL switch on the DP/DP coupler. Use a screwdriver to set the PROFIBUS-DP address. It results from adding the switches that are in the "ON" position.

PROFIBUS-DP address 6 must be set at the DP/DP coupler on the part of the SINUMERIK 802D sl (network 1 = DP1 – switch 2+4 on). The second address (network 2 = DP2) is user defined.

On the part of the 802D, first 16-byte digital inputs and then 16-byte digital outputs are parameterized by the fixed parameterization in the DP coupler. In the 802D sl, these are assigned to input bytes 27 - 42 or output bytes 18 - 33.

Thus on the second side of the DP/DP coupler, you must mirror that by first setting the 16-byte digital outputs and then the 16-byte digital inputs. You are free to decide on which input/output bytes you place these in your control system. (see the SIMATIC DP/DP Coupler Manual)

You can test the function briefly in SINUMERIK 802D sl under System/PLC Status.

IB 27 B _ _ _ _ _ _ _ _

QB 18 B _ _ _ _ _ _ _ _

Table 8- 5 Setting the PROFIBUS address on the DP/DP coupler

PROFIBUS address	DIL switch DP1 (SINUMERIK 802D sl) network 1
6	2 + 4 = ON

PROFIBUS address	DIL switch DP2 (SINUMERIK 802D sl) network 2
can be freely selected.	can be freely selected.

If the DP/DP coupler is used, the assignment of the input/output bytes will be as follows:

Table 8- 6 Assignment of the input/output bytes

DP/DP coupler network 1, PROFIBUS-DP address 6	
Input bytes	27 ... 42 (16 byte)
Output bytes	18 ... 33 (16 byte)

Changing the PROFIBUS-DP address

You can change the PROFIBUS-DP address setting at any time. However, if you do change the PROFIBUS-DP address setting, you will need to disconnect and reconnect the 24 V DC power supply in order to apply it.

See also

Digital inputs/outputs (Page 31)

8.8 Starting Up the PLC

After the PROFIBUS addresses have been set up, a PLC user program is executable. This PLC user program is needed for further commissioning. This program is loaded using the programming tool.

See also

Starting Up the PLC (Page 295)

Programming Tool PLC802 (Page 296)

PLC application Download/Upload/Copy/Compare (Page 327)

8.9 Startup of drives (SINAMICS)

The SINUMERIK 802D sl is used to commission the SINAMICS S120 drives via the HMI.

Note

Views

Various items of drive data relating to the Control Unit, Active Line Module, Motor Module, etc., drive objects are preassigned using a special mechanism that forms part of first startup in accordance with the requirements of a SINUMERIK 802D sl deviating from the SINAMICS standard.

The parameters affected are listed in file "SINAMICS_Delta_sichten.txt". which is included in the Toolbox.

Among other things, this affects the following machine data:

```
SERVO:

Message frame 116 is preassigned to P922

2000 is preassigned to P857

P951 is set to HiddenMacro (should not be displayed)

The value range for P1520 is restricted to between 0 and 10,000,000

The value range for P1521 is restricted to between -10,000,000 and 0

0 is preassigned to P1780

1 is preassigned to P2038

P8750 is set to Factory access level


7841 (fault number) is preassigned to P2100[0], and 3 is preassigned to P2101[0]
(therefore, OFF3 for fault 7841)
```

See also

Commissioning the drives via HMI (Page 331)

8.10 Set the axis/spindle-specific machine data.

Setpoint/actual value marshaling

Axis machine data MD30130 $MA_CTRLOUT_TYPE can be used to switch the setpoint output between simulation and SINAMICS drive. MD30240 $MA_ENC_TYPE can be used to do the same for the actual-value input.

Table 8- 7 Setpoint/actual value marshaling

Machine data	Simulation	Normal operation
MD30130	Value = 0 Simulation	Value = 1 In this case, the setpoint signals are output via Profibus.
MD30240	Value = 0 Simulation	Value = 1 (INCR) or 4 (EnDat) In this case, the actual values are read in via Profibus.

Note

For simulation, MD30130 and MD30240 must be set to a value of "0".

To enable the relevant NC axis to assign its setpoint to the appropriate SINAMICS drive, while ensuring that the actual values are returned from this SINAMICS drive, parameters must be assigned for the machine data MD30110 $MA_CTRLOUT_MODULE_NR and MD30220 $MA_ENC_MODULE_NR.

The following applies for the maximum configuration of 6 axes with Active Line Module:

Table 8- 8 maximum configuration

Axis	Drive number MD30110 MD30220	SINAMICS object no.
SP	1	3
X1	2	4
Y1	3	5
Z1	4	6
A1	5	7
PLC axis	6	8

If this setting does not match the order in the drive group (the order of the DRIVE-CLiQ connections corresponds to the order of the SINAMICS object no., here: 1. CU, 2. ALM, 3. Spindle, 4. X1 axis, 5. Y1 axis, 6. Z1 axis, 7. A1 axis, 8. PLC axis), the data must be adapted accordingly.

Example 1:

Milling machine with three axes and one spindle.

- The technology data block (setup_M.arc) has been loaded.
- The bus configuration has been selected with MD11240[2] = 0.
- Axis machine data MD30110 $MA_CTRLOUT_MODULE_NR and MD30220 $MA_ENC_MODULE_NR are adapted as follows.

Milling machine

Table 8- 9 Adapting the axis machine data for the milling machine

Axis	Drive number MD30110 MD30220	Sinamics object no.
X1	2	4
Y1	3	5
Z1	4	6
SP	1	3

- Set the PB addresses and object no. of the drives as specified in the table above. Since the 5th axis (A1) is not used, MD20070 $MC_AXCONF_MACHAX_USED[4]=0 must be parameterized. This will remove the axis from the configuration of the NC.

Example 2:

Lathe with two axes and one spindle/two spindles.

- The technology data block (setup_T.arc) has been loaded.
- The bus configuration has been selected with MD11240[2] = 0.
- Axis machine data MD30110 $MA_CTRLOUT_MODULE_NR and MD30220 $MA_ENC_MODULE_NR are adapted as follows.

Table 8- 10 Adapting the axis machine data

Axis	Drive number MD30110 MD30220	Sinamics object no.
X1	2	4
Z1	3	5
SP	1	3
A1	4	6

- Set the PB addresses and object no. of the drives as specified in the table above. Since the 5th axis (A1) is not used, MD20070 $MC_AXCONF_MACHAX_USED[4]=0 must be parameterized. This will remove the axis from the configuration of the NC.

8.10.1 Default settings of the axis machine data for feed axes

The following machine data list summarizes all default data or their recommended settings with SINAMICS S120 drives connected.

Once they have been set, the axes are ready to traverse, and only a fine adjustment (reference point approach, software limit switches, position controller optimization, speed feedforward control, lead error compensation,...) must be performed.

References

SINUMERIK 802D sl Function Manual

Note

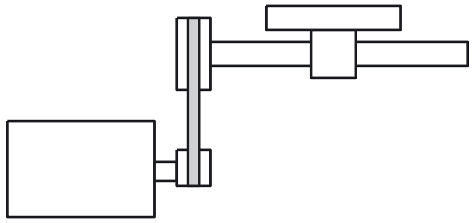
For feed axes, only parameter set 1 = index [0] is used. Index [1] ... [5] must only be parameterized when using the parameter set changeover function, G331 "Rigid tapping", or G33.

Table 8- 11 Default settings of the axis machine data for feed axes

MD	Name	Default value	Unit	Remarks
31030	$MA_LEADSCREW_PITCH	10	mm	Leadscrew of the ballscrew
31050 31060	$MA_DRIVE_AX_RATIO_DENOM $MA_DRIVE_AX_RATIO_NUMERA	1 1		Load gear transmission ratio Revolutions of the ballscrew Motor revolutions
32000	$MA_MAX_AX_VELO	10000	mm/min	Maximum axis velocity
32300	$MA_MAX_AX_ACCEL	1	m/s^2	Maximum axis acceleration
34200	$MA_ENC_REFP_MODE	1		1: Incremental encoder Motor order no: 1Fx6xxx-xxxxx-xAxx 0: EnDat encoder Motor order no: 1Fx6xxx-xxxxx-xExx
36200	$MA_AX_VELO_LIMIT	11500	mm/min	Threshold value for velocity monitoring; setting rule: MD36200 = 1.15 x MD32000

Example:

Motor with incremental encoder
Gear ratio: 1:2
Leadscrew pitch 5 mm
Max. axis speed 12 m/min
Max. axis acceleration 1.5 m/s2
Machine data settings:
MD31030 = 5
MD31050 = 1
MD31060 = 2
MD32000 = 12000
MD32300 = 1.5
MD36200 = 13800



The axis can now be traversed. The direction of movement can be reversed with MD32100 $MA_AX_MOTION_DIR = 1 or –1 (without affecting the control direction of the position control).

8.10.2 Default settings of the axis machine data for the spindle

With SINUMERIK 802D sl, the spindle is a subfunction of the entire axis functionality. The machine data of the spindle can, therefore, be found among the axis machine data (MD35xxx).

For this reason, data must also be entered for a spindle; this data has already been described in conjunction with the start-up of feed axes.

The following variants are offered for the spindle drive:

- Digital spindle drive with motor encoder
- Digital spindle drive with motor encoder and directly mounted encoder
- Analog spindle drive with directly mounted encoder
- Analog spindle without encoder

Note

For spindles without gearbox stage changeover, only gearbox stage 1 = index [1] is taken into account;
index [1] ... [5] must only be parameterized when using the gearbox stage changeover function.

Table 8- 12 Default settings of the axis machine data for the spindle

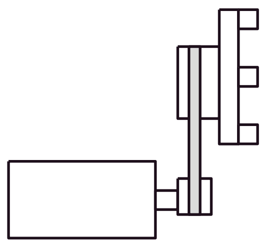
MD	Name	Default value	Unit	Remarks
30200	$MA_NUM_ENCS	1		0: spindle without speed actual-value encoder (AM mode = operation without encoder) 1: spindle with speed actual-value encoder integrated into the motor (1PH7 motor)
 31050 31060	 $MA_DRIVE_AX_RATIO_DENOM[1] $MA_DRIVE_AX_RATIO_NUMERA[1]	 1 1		**Load gear transmission ratio** Load revolutions Motor revolutions
35100	$MA_SPIND_VELO_LIMIT	10000	rpm	Maximum spindle speed
35130	$MA_GEAR_STEP_MAX_VELO_LIMIT[1]	500	rpm	Max. speed in gear stage 1
35200	$MA_GEAR_STEP_SPEEDCTRL_ACCEL[1]	30	rev/s^2	Acceleration in the speed control mode
36200	$MA_AX_VELO_LIMIT	6	mm/min, rpm	Velocity monitoring threshold value

Digital spindle drive with spindle actual-value encoder integrated into the motor

Parameterize the machine data listed in the previous table.

Example:

Motor with incremental encoder
Gear ratio: 1:2
max. spindle speed 9000 rpm
max. spindle acceleration 60 rev/s2
Machine data settings:
MD30200 = 1
MD31050 = 1
MD31060 = 2
MD35100 = 9000
MD35130 = 9000
MD35200 = 60
MD36200 = 9900



For the spindle, it can be necessary to adapt the following additional machine data.

Table 8- 13 Additional machine data

MD	Name	Default value	Unit	Recommendation/remark
34000	$MA_REFP_CAM_IS_ACTIVE	1		0: without reference point cam
34060	$MA_REFP_MAX_MARKER_DIST	20	Degr.	720_ = two spindle revolutions
34110	$MA_REFP_CYCLE_NR	1 ... 5		0: The spindle is not involved in channel-specific referencing.
35300	$MA_SPIND_POSCTRL_VELO	500	rpm	
36000	$MA_STOP_LIMIT_COARSE	0.04	Degr.	0.4
36010	$MA_STOP_LIMIT_FINE	0.01	Degr.	0.1
36030	$MA_STANDSTILL_POS_TOL	0.2	Degr.	1
36060	$MA_STANDSTILL_VELO_TOL	0.0139	Rpm	1 (interface signal "Axis/spindle stopped" V390x0001.4)
36400	$MA_CONTOUR_TOL	1	Degr.	3

Digital spindle drive with motor encoder and directly mounted spindle actual-value encoder (TTL)

Operating a second measuring system requires the following machine data settings.

Table 8- 14 Machine data to be set

MD	Name	Value	Recommendation/remark
30220	$MA_ENC_MODULE_NR[0]	3	Here you must enter a module number to which the measuring system is connected for the second measuring system (e.g., "3").
30230	$MA_ENC_INPUT_NR[0]	2	DRIVE-CLIQ slot on the module to which the second measuring system has been connected
32110	$MA_SENC_FEEDBACK_POL[0]	-1	If necessary, swap the counting direction

Analog spindle with spindle actual-value encoder

Hardware prerequisite:
The MCPA module must be available. (see Chapter "Connecting an analog spindle")

Table 8- 15 Machine data settings for analog spindle

Machine data		Value	Explanation
MD30100	$MA_CTRLOUT_SEGMENT_NR	0	Addressing local segment (onboard)
MD30110	$MA_CTRLOUT_MODULE_NR	1	Module No. 1

Machine data		Value	Explanation
MD30120	$MA_CTRLOUT_NR	1	Output No. 1
MD30130	$MA_CTRLOUT_TYPE	1	real standard output
MD30134	$MA_IS_UNIPOLAR_OUTPUT	0	0: bipolar; >0: Unipolar
MD32250	$MA_RATED_OUTVAL	100	100% control (10 V)
MD32260	$MA_RATED_VELO	3300	cause this speed
MD30230	$MA_ENC_INPUT_NR	2	Input No. 2 (2nd encoder)

Since the MCPA module does not have an encoder connection, it is only possible to use an encoder if it is available as a 2nd encoder of a SINAMICS axis. This 2nd encoder must be configured within SINAMICS in such a manner that it is contained in the message frame and is thus available to the controller.

Analog spindle without spindle actual-value encoder

For an analog spindle without an encoder, the same machine data apply as for an analog spindle with a directly mounted encoder, but MD30240 must be set to zero.

Analog spindle (unipolar) for SINUMERIK 802D sl

MD30134 = 1 unidirectional D/A value type "1"

MD32100 = 1 positive assignment, no inversion

MD32100 = -1 positive assignment, inversion

Machine data		Direction of spindle rotation	Voltage	Setpoint display	VB38020004
30134 = 1	32100 = 1	Spindle CW	>0	-	Bit 6 = 1
		Spindle CCW	>0	+	Bit 7 = 1
30134 = 1	32100 = -1	Spindle CW	>0	-	Bit 6 = 1
		Spindle CCW	>0	+	Bit 7 = 1

MD30134 = 2 unidirectional D/A value type "2"

MD32100 = 1 positive assignment, no inversion

MD32100 = -1 positive assignment, inversion

Machine data		Direction of spindle rotation	Voltage	Setpoint display	VB38020004
30134 = 2	32100 = 1	Spindle CW	>0	-	Bit 6 = 1
		Spindle CCW	>0	+	Bit 7 = 1
30134 = 2	32100 = -1	Spindle CW	>0	-	Bit 6 = 1
		Spindle CCW	>0	+	Bit 7 = 1

Example: Configuration for 3 axes with an analog spindle

Output of an analog setpoint for a spindle drive for connecting a converter (e.g., MICROMASTER). With software release 01.01, connection is possible via the MCPA module.

- SINAMICS S120:

  ALM; 1-axis module; 1-axis module; 1-axis module
- MICROMASTER:

  MM440

Table 8- 16 Machine data for the example

MD	Name	X	Y	Z	SP	Remarks
30100	$MA_CTRLOUT_SEGMENT_NR	5	5	5	0	local bus segment for analog
30110	$MA_CTRLOUT_MODULE_NR	2	3	4	1	Module order
30120	$MA_CTRLOUT_NR	1	1	1	1	Setpoint output on drive module/module
30130	$MA_CTRLOUT_TYPE	1	1	1	1	Output value of the setpoint values
30134	$MA_IS_UNIPOLAR_OUTPUT	0	0	0	0	The setpoint output is unipolar
30200	$MA_NUM_ENCS	1	1	1	1	Number of encoders
30220	$MA_ENC_MODULE_NR	2	3	4	3	Transport module (the SMC30 module is connected to the axis module of the Y axis)
30230	$MA_ENC_INPUT_NR	1	1	1	2	Input on the drive module (X202)
30240	$MA_ENC_TYPE	4	1	1	1	Encoder kind
32250	$MA_RATED_OUTVAL (spindle)	80				Rated output voltage 8 V at $U_{max/min}$
32260	$MA_RATED_VELO (spindle)	3200				Rated motor speed at 8 V

Thereafter, set the following machine data:

Table 8- 17 Additional machine data

MD	Name	Default value	Recommendation/remark
11240[2]	$MN_PROFIBUS_SDB_NUMBER	0	
13060[2]	$MN_DRIVE_TELEGRAM_TYPE	116	Standard message frame type for Profibus–DP

If the transport module is the 3rd axis (example: Y).

Table 8- 18 Additional machine data

MD	Name	Default value	Unit	Recommendation/Remark
34000	$MA_REFP_CAM_IS_ACTIVE	1		0: without reference point cam
34060	$MA_REFP_MAX_MARKER_DIST	20	Degr.	720_ = two spindle revolutions
34110	$MA_REFP_CYCLE_NR	1 ... 5		0: The spindle is not involved in channel-specific referencing.
35300	$MA_SPIND_POSCTRL_VELO	500	Rpm	
36000	$MA_STOP_LIMIT_COARSE	0.04	Degr.	0.4
36010	$MA_STOP_LIMIT_FINE	0.01	Degr.	0.1
36030	$MA_STANDSTILL_POS_TOL	0.2	Degr.	1
36060	$MA_STANDSTILL_VELO_TOL	0.0139	Rpm	1 (interface signal "Axis/spindle stopped" V390x0001.4)
36400	$MA_CONTOUR_TOL	1	Degr.	3

8.10.3 Analog axis/spindle with TTL encoder via ADI4

Properties

An ADI4 module (Analog Drive Interface for 4 Axes) is an interface module which can be used to operate up to four drives with an analog setpoint interface on the equidistant PROFIBUS-DP.

Communication between the controller and the ADI4 takes place via an ADI4-specific message frame type which, in addition to digital input/output data, also contains a message frame type (standard message frame 3) for each drive, specified according to a PROFIDrive profile.

As part of cyclic DP communication, the actual drive values (encoder values) are transferred from the ADI4 module to the controller via the PROFIBUS-DP, and the speed setpoints calculated by the controller are transferred to the ADI4 module.

The transferred speed setpoints are then output from the ADI4 module to the drives as analog values.

SINUMERIK 802D sl with ADI4 module

Up to 4 drives with an analog setpoint interface can be controlled by an ADI 4. A maximum of six 5 V TTL square wave signal encoders, both rotary and linear, can be connected to the SINUMERIK 802D sl using up to two ADI 4 modules.

Note

If one ADI 4 is used, all axes must be operated on an analog basis. Mixed operation involving both ADI 4 and SINAMICS is not possible.

The fixed assignment of the axes to the connectable measuring systems is selected via MD11240[0] $MN_ PROFIBUS_SDB_NUMBER.

The following tables in the "Configuration" section describe the configuration of the assignment of the connectable TTL encoders:

- For setting MD11240[0]=1 -> Table "MD11240[0]=1"
- For setting MD11240[0]=2 -> Table "MD11240[0]=2".

References

Product manual "ADI4 - Analog drive interface for 4 axes"

Parameterization

The DRIVE READY signals for every connected axis need to be connected to X6-2 pins 8 to 11 on the ADI 4.

To operate an ADI 4 module on a SINUMERIK 802D sl, it is necessary to load adi4.ini into the control system. (see directory...\Siemens\Toolbox 802D_sl\Vxxxxxxxx\Special\adi4.ini)

The following machine data must also be checked for each connected axis:

- MD30110[0] $MA_ CTRLOUT_MODULE_NR

  Setpoint: Drive/module number
- MD30220[0] $MA_ ENC_MODULE_NR

  Actual value: Drive module/measuring circuit number
- MD30240[0] $MA_ ENC_TYPE

  Actual value: Encoder type
- MD31020[0] $MA_ ENC_RESOL

  Encoder pulses per revolution
- MD32250[0] $MA_ RATED_VALUE
- MD32260[0] $MA_ RATED_VELO
- MD36700 $MA_DRIFT_ENABLE
- MD32110[0] $MA_ ENC_FEEDBACK_POL

Connectable measuring systems

TTL incremental encoder (for the increments, see the following tables "MD 11240[0]=1", " MD 11240[0]=2"), difference transmission with 5 V- square-wave signals (RS422 standard).

The signal period must be converted into a PPR count in order to establish which linear measuring systems can be used.

Example

Set the following machine data for a linear measuring system:

Table 8- 19 Machine data

Requirement	Machine data	Value
Linear measuring system	MD31000 $MA_ENC_IS_LINEAR	1
A grid division of 20 µm and quintuple EXE result in increment signals of 20 µm/5 = 4 µm resolution.	MD31010 $MA_ENC_GRID_POINT_DIST(0)	0.004
Leadscrew of 10 mm/rev	MD31030 $MA_LEADSCREW_PITCH	10

Depending on the leadscrew of the relevant machine axis, the following applies:

For a traverse path of 10 mm (with 1 ballscrew revolution), a spacing period of 4 µm results in a value of 2,500 pulses.

$$\frac{\text{MD31030 \$MA\_LEADSCREW\_PITCH}}{\text{MD31010 \$MA\_ENC\_GRID\_POINT\_DIST}} = \frac{10}{0.004} = 2500$$

Figure 8-10 Number of pulses

Once MD 11240[0] = 1 has been set, axes 1, 2, or 3 (every 2,500 increments) can be used.

Configuration

The following tables show the fixed assignment to the connectable TTL encoders.

Table 8- 20 MD11240[0]=1

PROFIBUS address	16			
Axis	1. Axis	2. Axis	3. Axis	4. Axis
Encoder lines	2500	2500	2500	1024
MD30110[0]	7	8	9	10
MD30220[0]	7	8	9	10
Connection of encoder ADI4	X4.1	X4.2	X5.1	X5.2

PROFIBUS address	15			
Axis	1. Axis	2. Axis	3. Axis	4. Axis
Encoder lines	1024	18000	9000	2500
MD30110[0]	11	12	13	14
MD30220[0]	11	12	13	14
Connection of encoder ADI4	X4.1	X4.2	X5.1	X5.2

Table 8- 21 MD11240[0]=2

PROFIBUS address	16			
Axis	1. Axis	2. Axis	3. Axis	4. Axis
Encoder lines	2048	2048	2048	1024
MD30110[0]	7	8	9	10
MD30220[0]	7	8	9	10
Connection of encoder ADI4	X4.1	X4.2	X5.1	X5.2

PROFIBUS address	15			
Axis	1. Axis	2. Axis	3. Axis	4. Axis
Encoder lines	1024	18000	9000	2048
MD30110[0]	11	12	13	14
MD30220[0]	11	12	13	14
Connection of encoder ADI4	X4.1	X4.2	X5.1	X5.2

Example

An analog spindle with an incremental rotary encoder is to be operated on a turning machine with the following values:

1. Encoder lines -> 2,500 increments/revolution
2. Max. spindle speed -> 9,000 revolutions/minute

The following values must be entered in the machine data for a spindle on the control system:

Table 8- 22 Machine data for a spindle

Machine data	Name	Value	Remarks
MD11240	$MN_PROFIBUS_SDB_NUMBER[0]	1	
MD30110	$MA_CTRLOUT_MODULE_NR[0]	9	

Machine data	Name	Value	Remarks
MD30220	$MA_ENC_MODULE_NR[0]	9	
MD30240	$MA_ENC_TYPE[0]	1	1:= Incremental encoder
MD31020	$MA_ENC_RESOL[0]	2500	Encoder pulses for rotary encoder
MD32250	$MA_RATED_VALUE[0,AX3]	100	
MD32260	$MA_RATED_VELO[0,AX3]	9000	
MD36700	$MA_DRIFT_ENABLE[AX3]	1	
MD32110	$MA_ENC_FEEDBACK_POL[0]	0:= Standard -1:= Inverted	

Note

The spindle on the ADI 4 only functions if a measuring system is connected.

See also

Connecting an ADI4 module (Page 93)

8.10.4 PLC-controlled axis

Axes can be controlled via the PLC's NCK interface V380x3000/V390x3000. The following functions are supported:

- Positioning axis
- Indexing axis

References

SINUMERIK 802D sl Function Manual, Positioning Axes

PLC user library in Toolbox

Prerequisites

The following set machine data are required to define a permanently assigned PLC axis:

Table 8- 23 Machine data for a permanently assigned PLC axis

Mach. data	Machine data	Value	Comments
General MD	MD19100 NUM_AXES_IN_SYSTEM	5	This machine data is displayed in "Expert mode". Standard assignment = 5 With maximum configuration = 6
Axis MD	MD30460 $MA_BASE_FUNCTION_MASK	20	The axis is a permanently assigned PLC axis. The axis can, however, be jogged and referenced. The axis cannot be assigned to the NC program This property is displayed from the NCK to the PLC in the V390x0011.7 "PLC axis permanently assigned" signal.
Channel MD	MD20070 [x $MC_AXCONF_MACHAX_USE D]	6	Directly after the parameterized NC axes (see the example below with a turning machine with 2 axes, a spindle and a PLC axis)

Example: Turning machine with 2 axes, a spindle and a PLC axis

The axis number of the PLC axis (usually 6) must be entered in the channel-specific machine data MD20070 $MC_AXCONF_MACHAX_USED after the NC axis for which parameters have been assigned.

Example: Lathe with 2 axes and 1 spindle

Table 8- 24 Adapting the axis number

Axis	MD20070
X1	[0]=1
Z1	[1]=2
SP	[2]=3
PLC	[3]=6

Prerequisite for PLC axis rotary axis functionality

Table 8- 25 Additional axis machine data for PLC axis rotary axis functionality

Machine data	Value
MD30300 $MA_IS_ROT_AX	1
MD30310 $MA_ROT_IS_MODUL	1
MD30320 $MA_DISPLAY_IS_MODULO	1

8.10.5 Completion of the commissioning of the axes/spindle

The general commissioning of the axes/spindle is completed. A fine optimization must still be carried out.

8.11 Completing the commissioning

After the start-up by the machine manufacturer has been completed, it is recommended to carry out a data backup prior to delivery to the end customer:

1. Perform internal data backup (at least protection level 3 required):
   - In the <SYSTEM> operating level, press the "Save paramet." softkey.
   - The drive machine data is saved automatically after the commissioning with HMI.
2. Carry out external data backup on customer CF card (see Chapter "Data backup and series commissioning")
3. Reset the access level:
   - Press the "Reset password" softkey.

8.12 Managing user files

8.12.1 Overview, managing user files

The following user files can be managed with the RCS802 tool:

- PLC user alarm texts (file alcu.txt)
- Cycle alarm texts (file alc.txt)
- Cycle mask texts and cycle messages (file aluc.txt)
- Cycles (*.spf)
- NC programs (*.mpf)

- HMI control files cov.com and sc.com
- Texts for user DLL
- Texts for user dialogs

User files can be saved as HMI start-up archive and directly loaded into the control or from the control into the toolbox project.

Operating sequence

1. In the main screen menu click "Extras" > "Toolbox Manager" > "Select OEM".

   All languages of the active toolbox project and their available user text files are displayed in the dialog box "Select OEM" under the "Alarmtexts" tab.

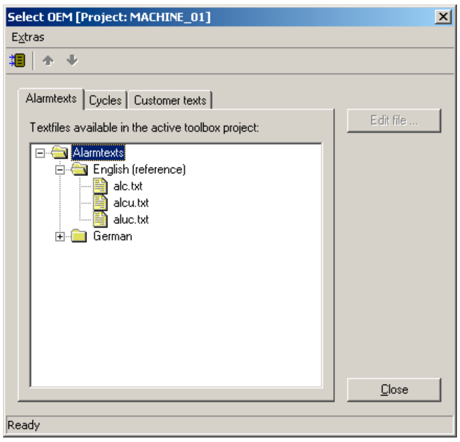


Figure 8-11 Dialog box "Select OEM - Alarmtexts"

2. All user files on the subject of cycles from the active toolbox project are listed under the "Cycles" tab.

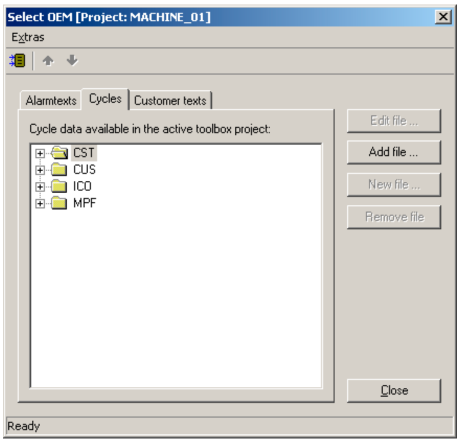


Figure 8-12 Dialog box "Select OEM - Cycles"

3. The texts for user DLLs are administered in the "Customer texts" tab.

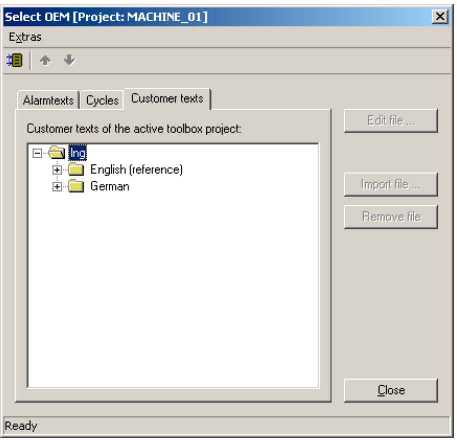


Figure 8-13 Dialog box "Select OEM - Customer texts"

See also

Editing/creating user cycle masks (Page 156)

8.12.2 Creating the cycle screen texts and softkey labels

Operating sequence

1. Select the main menu "Extras" > "Toolbox Manager" > "Select OEM".

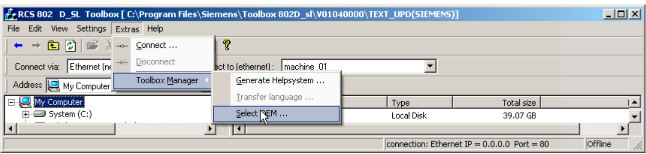


Figure 8-14 Sequence to select the new toolbox project

The languages available in the toolbox project are displayed with their respective text files under the "Alarmtexts" tab:

– alc.txt:

  Cycle alarm texts can be added, modified or removed.

– alcu.txt:

  Only the pre-defined PLC alarm texts can be modified.

– aluc.txt:

Customer texts for the cycle screens, cycle messages and softkeys can be added, modified or removed.

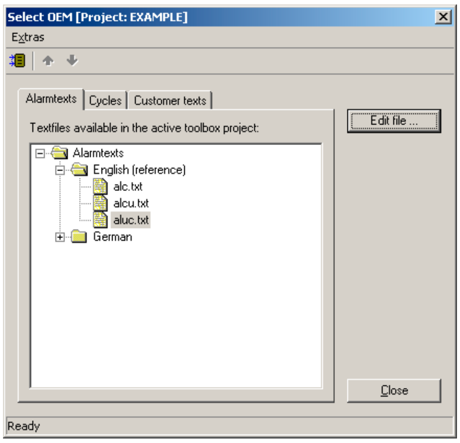


Figure 8-15 Display of text files in the "Select OEM" dialog box

2. Select, for example, the file "aluc.txt" and click "Edit file...".

   An input screen opens where you can edit the text.

   Depending on the selected user text file, the dialog box can be displayed differently:

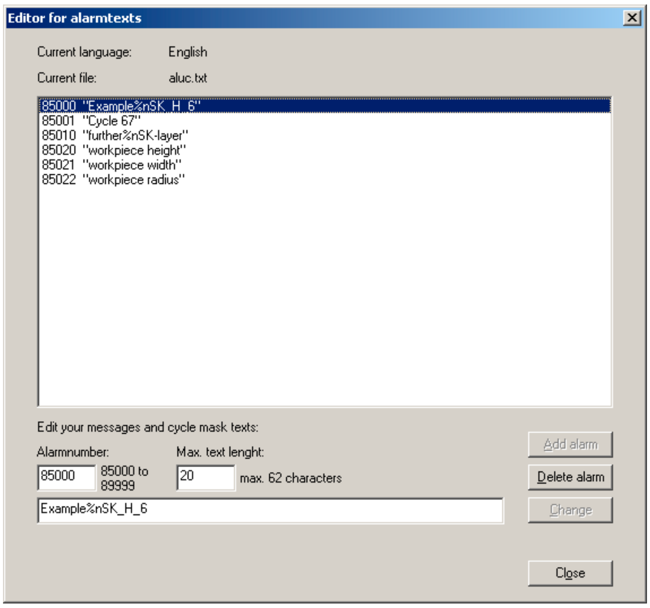


Figure 8-16 Screen for editing texts

3. You can now carry out the following functions:

   – In the "Alarmnumber" field enter the alarm number for the text.

   – The maximum number of available characters is displayed in the "Max. text length" field. A maximum of 2 lines with 9 characters each are available for softkey texts. The character string "%n" ("Example%nSK_H_6") is used as line separator. Single-line softkey texts do not need the separator.

– Enter the alarm text in the text field. Click "Add alarm" to add the new alarm to the file.
– The "Delete alarm" button deletes a selected alarm text.
– To change an existing text, select it from the list.

  You can change the alarm number, the text length and the text.

  Click "Change" to apply the change to the file.
– Once you have finished editing the texts, click "Close" to close the dialog box.

  A prompt appears. To save the change, click "Yes".

Note

The texts are called via $Alarmnumber in the files cov.comEditing the new softkeys (Page 166) and sc.com Editing the user cycle screens (Page 171).

8.12.3 Text color of the alarms

Text color of the alarms

The following notation can be used to control the text color of the alarms:

The alarm text starts with "#[Cxxxx" and ends with "#[Cyyyy".

Start: #[Cxxxx

End: #[Cyyyy

- xxxx -> required color
- yyyy -> default color (BLACK)

Note

At the end of the color-coded alarm, always reset the color change.

Example

"#[CREDSpindle running.#[CBLACKDo not open door!"

The following text colors are possible for alarms:

Note

Use the text color selection functionality for the PLC user alarms ONLY!

Table 8- 26 Text colors for alarms

Coding	Color
WHITE	White
BLACK	Black
GREY_1	Gray shade 1
GREY_2	Gray shade 2
YELLOW	Yellow
RED	Red
BLUE	Blue
ORANGE	Orange
PETROL	Petrol
SIM_BLUE	Siemens Blue
GREEN	Green

8.12.4 Saving user files as HMI start-up archive

Operating sequence

Proceed as follows to create an HMI start-up archive from a selection of user files:

1. Start the dialog "Save files as Startup archive (HMI)" from the menu item "Extras" > "Save files as archive…" or by clicking the following icon in the dialog "Select OEM".

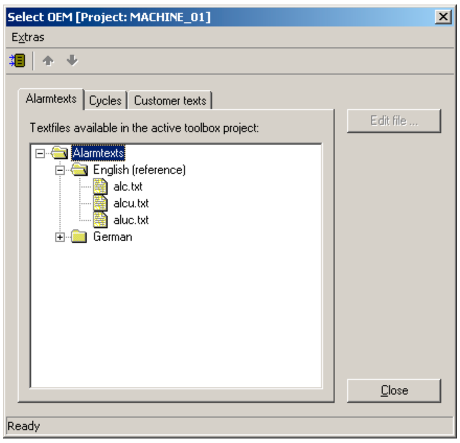


Figure 8-17 Dialog box "Select OEM - Alarmtexts"

2. In the lefthand directory tree, select the user files that should be included in the HMI start-up archive and insert these to the selection list on the righthand side by clicking on "Add". Click on "Save" to confirm the selection.

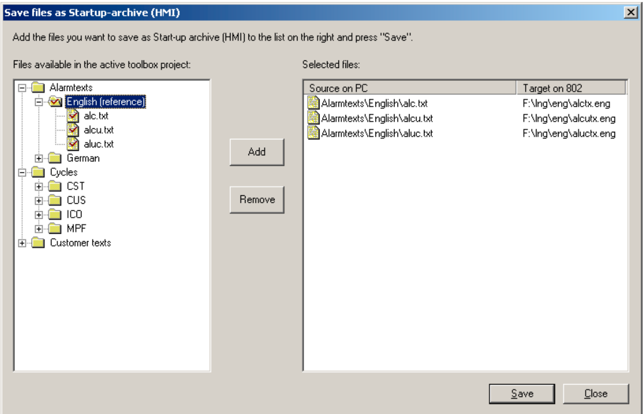


Figure 8-18 Dialog box "Save files as Start-up archive (HMI)"

3. Select where the files should be saved and enter the file names. Click "Save" to save the HMI start-up archive.

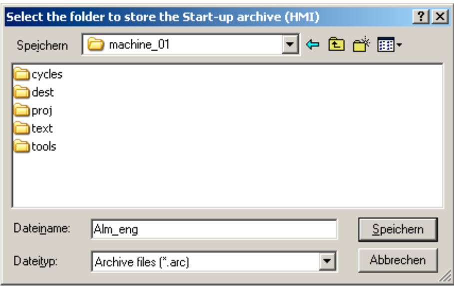


Figure 8-19 Dialog box "Save as database"

8.12.5 Load user files into the control.

Operating sequence

Proceed as follows to load user files directly into the control:

1. Ensure that the RCS802 is connected to the SINUMERIK 802D sl.
2. Start the dialog "Write files to connected SINUMERIK 802D sl" from the menu item "Extras">"Write files to 802…" or by clicking the following icon in the dialog "Select OEM".

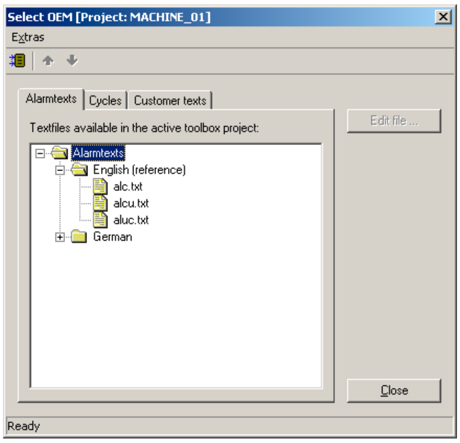


Figure 8-20 Dialog box "Select OEM - Alarmtexts"

3. All user files from the active toolbox project are listed in the lefthand directory tree. Insert the required user files from the lefthand directory tree into the righthand selection list by selecting and clicking on "Add".

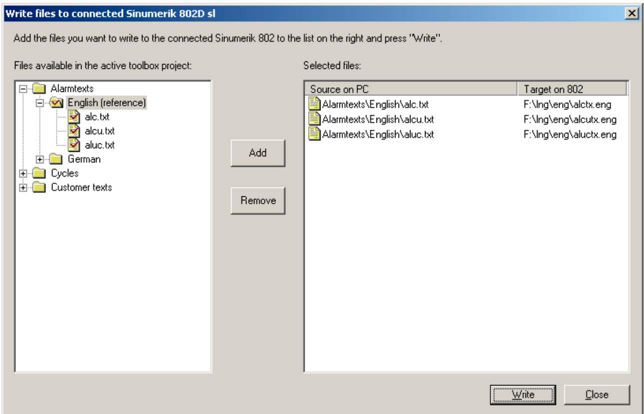


Figure 8-21 Dialog box "Write files to connected SINUMERIK 802D sl"

4. In order to load the selected files into the control, click on "Write".

Note

As a consequence of writing files to the control, an NCK restart or HMI restart can occur!

8.12.6 Loading user files from the control

Operating sequence

Proceed as follows to load user files into the active toolbox project from the control:

1. Ensure that the RCS802 is connected to the SINUMERIK 802D sl.
2. Start the dialog "Read files from connected SINUMERIK 802D sl" from the menu item "Extras">"Read files to 802…" or by clicking the icon in the dialog "Select OEM".

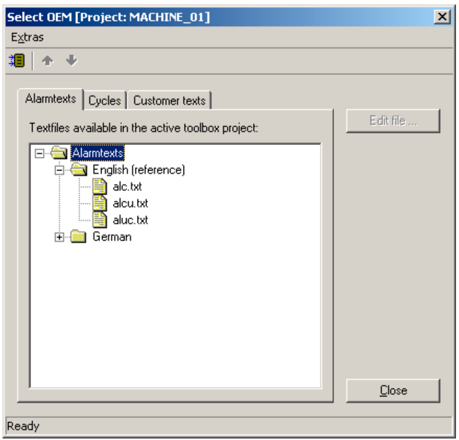


Figure 8-22 Dialog box "Select OEM - Alarmtexts"

3. All of the user files available in the control are listed in the lefthand directory tree. Insert the required user files from the lefthand directory tree into the righthand selection list by selecting and clicking on "Add".

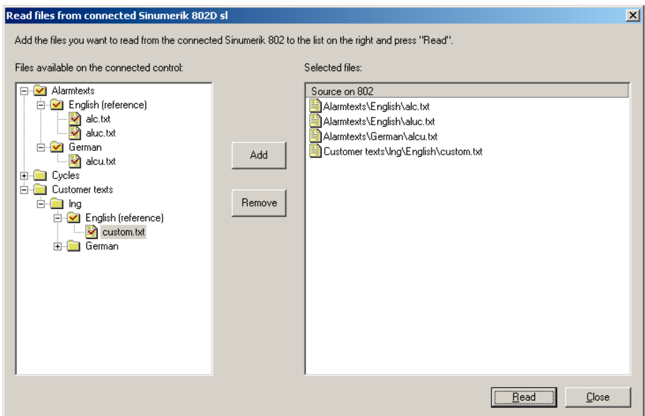


Figure 8-23 Dialog box "Read files from connected SINUMERIK 802D sl"

4. Click on "Read" to load the selected files into the active toolbox project.

Note

The existing files are overwritten when you upload files from the control to the active toolbox project!

8.13 Editing/creating user cycle masks

Overview

You can use the following options to edit/create user cycles for the SINUMERIK 802D sl using the RCS802 tool:

- Creating cycles
- Editing user cycle text
- Editing the new softkeys
- Editing the user cycle masks
- Creating the screens for the user cycle masks

See also

Creating the cycle screen texts and softkey labels (Page 149)

Editing the new softkeys (Page 166)

Editing the user cycle screens (Page 171)

Creating the images for the user cycle screens (Page 175)

8.13.1 Processing user cycle masks

The RCS802 program allows you to manage all the HMI control files necessary to create user cycle masks in your toolbox project.

The "CST", "CUS", "ICO" and "MPF" directories from the active toolbox project are displayed in the dialog box "Select OEM" (Page 147) under the "Cycles" tab.

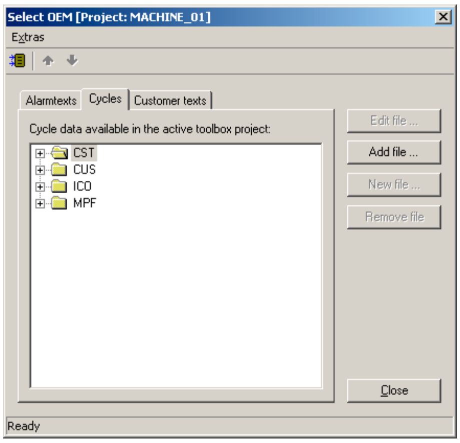


Figure 8-24 Edit user files

To add the active toolbox project HMI control files, proceed as follows:

- Download the files from the connected control (refer to "Loading user files from the control")
- Insert them using the "Add file…" button.
- You can create new files using "New file...".

Save the HMI control files in the following directories:

"CST" directory

The "CST" directory is used to save HMI control files "cov.com" and "sc.com".

The HMI softkeys are parameterized using the "cov.com" file.

The "sc.com" file describes the input masks of the cycles.

Note

The HMI control files "cov.com" and "sc.com" can differ in the individual technologies! Therefore ensure that they are correctly used.

"CUS" directory

Save your own cycles in the "CUS" directory.

"ICO" directory

The graphics (bitmap files *.bmp) associated with the cycles are saved in the "ICO" directory.

"MPF" directory

The "MPF" directory is used to save your NC main programs.

See also

Loading user files from the control (Page 155)

8.13.2 Requirements

Introduction

In the following, all files and programs will be listed and their source given.

The files and programs are necessary during the creation of user cycle masks.

Creating cycles

Reference

SINUMERIK 802D sl Programming and Operating Manual; "Cycles"

Note

The program RCS802 supports editing of user cycles from Version 01.05.03.00 upwards.

Editing user cycle text

- Program: RCS802 V.01.05.03.00 or higher

  The program can be found in the Toolbox802D_sl.
- Files:
  - aluc.txt: Cycle mask texts and cycle messages
  - alc.txt: Cycle alarms
  - alcu.txt PLC user alarm texts

Editing the operating sequence of the new softkeys and the corresponding user cycle masks

- Program: RCS802 V.01.05.03.00 or higher

  The program can be found in the Toolbox802D_sl.
- File: cov.com

  The file cov.com is available in the control after installation of the technology-specific HMI-IBN archive in the NC directory "CST".

  The file is loaded from this directory into the Toolbox project using the RCS802 tool.

Editing the input parameters for the user cycle masks

- Program: RCS802 V.01.05.03.00 or higher

  The program can be found in the Toolbox802D_sl.
- File: sc.com

  The file sc.com is available in the control after installation of the technology-specific HMI-IBN archive in the NC directory "CST".

  The file is loaded from this directory into the Toolbox project using the RCS802 tool.

Creating the screens for the user cycle masks

- Program: RCS802 V.01.05.03.00 or higher

  The program can be found in the Toolbox802D_sl.
- Program: Screen editing program

  This program should allow 16 color bitmaps, size 222*222 pixels, to be edited.

8.13.3 Loading files "cov.com" and "sc.com" in the toolbox project.

Introduction

The HMI control files "cov.com" and "sc.com" are required to set the parameters for softkeys and input screens for the cycles.

Note

The HMI control files "cov.com" and "sc.com" can differ in the individual technologies! Therefore ensure that they are correctly used.

There are several ways to load these files for processing in the toolbox project:

- Load from the installed toolbox
- Load from the control

Load from the installed toolbox

The HMI control files are located in the installed toolbox under the corresponding technology in the "Cycles" directory.

Operating sequence

1. Select the main menu "Extras" > "Toolbox Manager" > "Select OEM".
2. Select directory "CST" in the "Select OEM" dialog box.
3. Click on "Add file...".
4. Select the HMI control files from the corresponding technology directory of the installed toolbox.

   e.g. C:\Program Files\Siemens\Toolbox 802D_sl\V01040300\Techno\Turning\Cycles\cyc_mask
5. Click "Open".

   The files are added to the toolbox project.

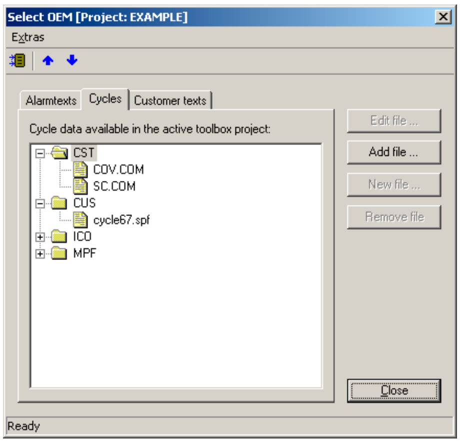


Figure 8-25 Dialog box "Select OEM"

Load from the control

Prerequisite

Initial state is a control specifically configured for the technology (using the start-up archive file from the toolbox). This ensures that all the standard cycless are contained in the NC directory "CST".

Operating sequence

The program RCS802 on your PC is linked to the control.

Load the HMI control files in the toolbox project:

1. Select the main menu "Extras" > "Toolbox Manager" > "Select OEM".
2. In the "Select OEM" dialog box, use either the menu "Extras" > "Read files from 802…" or the icon to open the "Read files from connected Sinumerik 802D sl" dialog box.

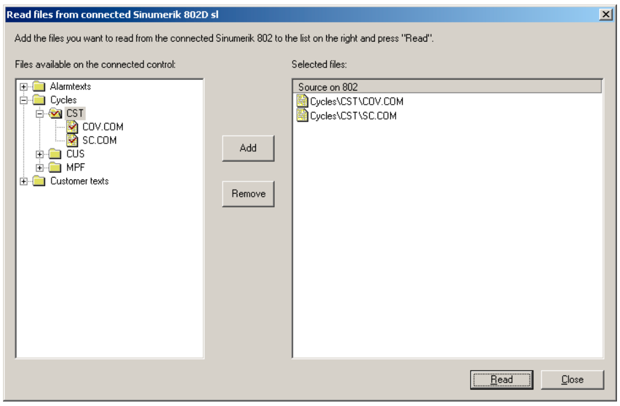


Figure 8-26 Dialog box "Read files from connected SINUMERIK 802D sl"

All of the user files available in the control are listed in the directory tree in the left pane.

3. In the "Cycles/CST" directory, select the two HMI control files.
4. Click "Add" to add the files to the selection list on the right.
5. Click the "Read" button.

   The upload is started.

   After successful upload, the files are located in the toolbox project under "CST".

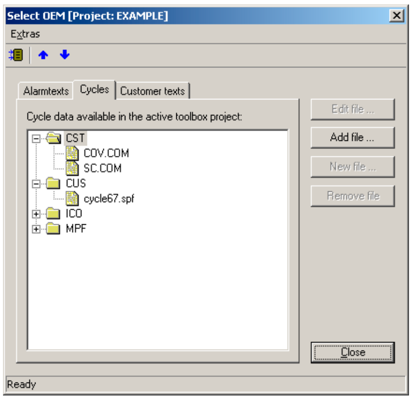


Figure 8-27 Dialog box "Select OEM"

8.13.4 Preparation in RCS802 for editing user files

Reference

See Chapter "Creating and editing a project" (Page 115)

8.13.5 Settings in RCS802

Operating sequence

You need to make the following settings in RCS802 before you can edit the user data:

1. Start the dialog box "RCS802 Settings" via the main menu "Settings" > "OEM".

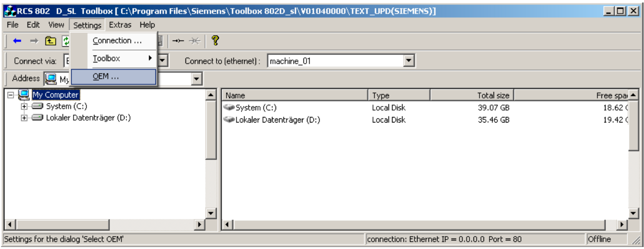


Figure 8-28 Main menu "Settings" > "OEM"

2. In the "RCS802 Settings - Select OEM" dialog box select the program for editing the text and image files.

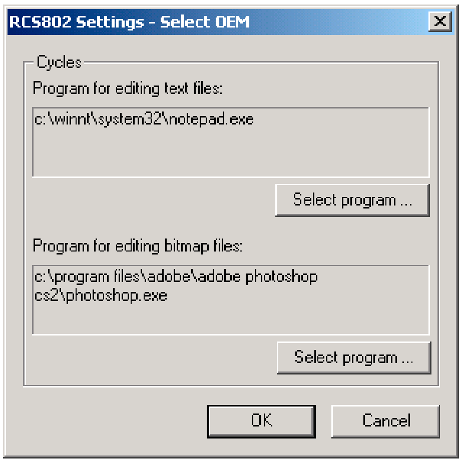


Figure 8-29 "RCS802 Settings - Select OEM"

8.13.6 Create and edit user cycles

Reference

Programming and Operating Manual SINUMERIK 802D sl; "Cycles"

Procedure

The following procedure describes how to create and edit a new cycle.

The file name must adhere to the following rule:

cycleXXXX.spf XXXX corresponds to the user cycle number

In this manual we have used cycle number 67 as the basis for the examples. You will be using the actual user cycle number of your cycle. It must not exceed 4 digits.

As a cycle screen always also transfers values as call parameters to the user cycle, the transfer interface is defined as follows.

PROC CYCLE67(REAL HEIGHT,REAL WIDTH,INT RAD)

PROC is a keyword followed by the cycle name with the cycle number. All the transfer parameters for the screen are contained within brackets with data type and name separated by commas.

Example: cycle67.spf

```
PROC CYCLE67(REAL HEIGHT,REAL WIDTH,INT    ; Declaration of transfer parameters
RAD)
                                           ; V02.01.06 Jun 20, 2008 creation date
                                           ; Example cycle
N20 MSG ("cycle start")                    ; Text
N30 G1 G18 G91 G94 F500 S30 T1
N40 M17
```

Operating sequence "Create and edit new cycle in the toolbox project"

1. Activate the main menu "Extras" > "Toolbox Manager" > "Select OEM…" to open the dialog box for editing user files.

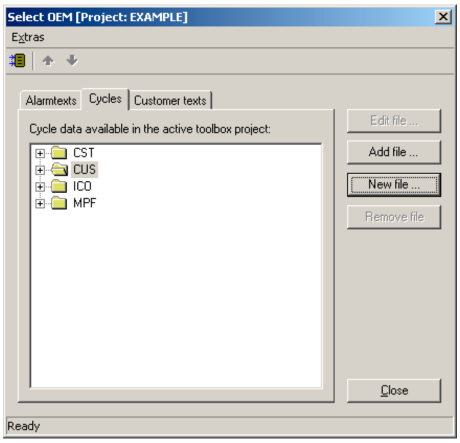


Figure 8-30 Edit user files - Cycles

2. Select directory "CUS" in the "Cycles" tab.

   Click "Add file…" to add user cycles to the toolbox project.
3. Click "New file…".

   The dialog box for creating new user files appears.

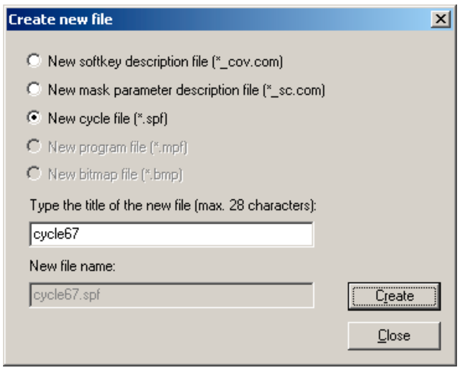


Figure 8-31 Dialog box "Create new file"

4. Select the option "New cycle file (*.spf)".
5. Assign a name according to the instructions.

6. Click "Create".

   The new cycle is created in the toolbox project.

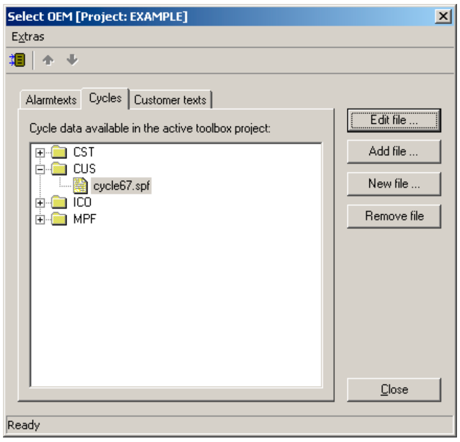


Figure 8-32 "Edit file"

7. Select the new cycle in the "CUS" directory.
8. Click "Edit file" to open the cycle.

   Edit the cycle.

Note

The user cycles are loaded on the control in the NC directory "CUS".

8.13.7 User cycle masks

Introduction

Edit the following in the configuration files for the user cycle screens:

- The display for the respective cycle screen texts, cycle messages and softkey texts.

  ->File "aluc.txt"
- The properties of the new softkeys.

  ->File "cov.com"
- The properties of the user cycle screens.

  ->File "sc.com"

Prerequisite

Initial state is a control specifically configured for the technology (using the start-up archive file from the toolbox). This ensures that all the standard cycles are contained in the NC directory "CST".

Procedure

1. Finish creating your user cycle before creating the cycle screen.
2. Create (Page 149) the cycle screen texts, cycle messages and softkey texts for the user cycle in the file "aluc.txt".
3. Load (Page 160) the files "cov.com" and "sc.com" in the toolbox project.
4. Edit (Page 166) the new softkeys in the file "cov.com".
5. Edit (Page 171) the user cycle screens in the file "sc.com".
6. Create (Page 175) the icons for the user cycle screens.
7. Transfer (Page 153) the user files to the control.

8.13.8 Editing the new softkeys

Introduction

The softkeys are described in the file "cov.com". The file also lists for each softkey the user cycles that are called via this softkey.

Operating sequence

1. Open the "Select OEM" dialog box via the menu "Extras" > "Toolbox Manager" > "Select OEM".
2. After successful upload the "cov.com" file is located in the "Cycles" tab in the "CST" directory for the toolbox project.

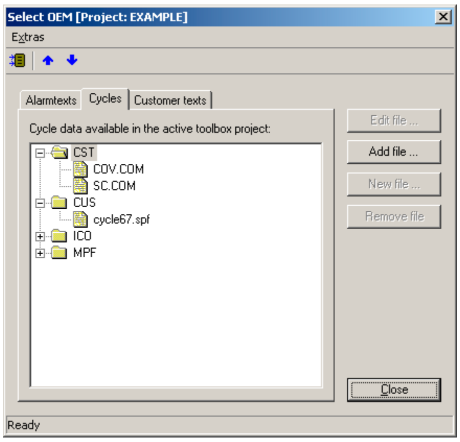


Figure 8-33 Dialog box "Select OEM"

3. Select the file and click "Edit file..." to open the file for editing.

```
S3.0.0\$80271\
S3.1.0\$80282\CN2(CYCLE81)[MCALL]
S3.2.0\$80283\CN2(CYCLE82)[MCALL]
S3.3.0\$80284\CN3(CYCLE83)[MCALL]
S3.4.0\$80274\
S3.4.1\$80991\CN6(CYCLE85)[MCALL]
S3.4.2\$80537\CN7(CYCLE86)[MCALL]
S3.4.3\$80992\CN9(CYCLE87)[MCALL]
S3.4.4\$80993\CN9(CYCLE88)[MCALL]
S3.4.5\$80994\CN9(CYCLE89)[MCALL]
S3.4.6\$80415\[P "pr.dll" "programm" 10000]
S3.5.0\$80538\
S3.5.3\$80298\CN4(CYCLE84)[MCALL]
S3.5.4\$80297\CN4(CYCLE840)[MCALL]
S3.5.6\$80415\[P "pr.dll" "programm" 10000]
S3.6.0\$80415\[P "pr.dll" "programm" 10000]
S3.7.0\$80275\
S3.7.2\$80290\CN11(HOLES1)
S3.7.3\$80291\CN12(HOLES2)
S3.7.6\$80415\[P "pr.dll" "programm" 10000]
S4.0.0\$80272\
S4.2.0\$80340\CN27(CYCLE71)
S4.3.0\$80339\CN28(CYCLE72)
S4.5.0\$80634\
S4.5.4\$80292\CN25(POCKET3)
S4.5.5\$80293\CN26(POCKET4)
S4.6.0\$80541\
S4.6.2\$80296\CN13(LONGHOLE)
S4.6.3\$80294\CN14(SLOT1)
S4.6.4\$80295\CN15(SLOT2)
S4.7.0\$80299\CN18(CYCLE90)
M17
```

Figure 8-34 Extract from a possible file "cov.com" for Milling technology

Syntax of a line in the file "cov.com"

One line describes one softkey.

SK3 to SK6 are available in the horizontal softkey bar.

The vertical softkey bar provides two levels with 8 softkeys in each.

A line consists of three parameters separated by a backslash.

```
Sx.y.z\Alarm number softkey text\Control parameters Comment
```

Example:

```
S3.0.0\$80271\ Select: "Drilling"
S3.1.0\$80228\CN2(CYCLE81) Cycle: "Drilling centering"
```

Table 8- 27 Parameter descriptions in the file "cov.com"

Parameter	Explanation
Sx.y.z	x Horizontal softkey level y Vertical softkey level 1 z Vertical softkey level 2
\Alarm number softkey text	$ +Alarm number from file "aluc.txt"
\Control parameter + Comment	If a lower level is called by pressing the softkey, the control parameter remains blank. If a cycle and associated screen is displayed by pressing the softkey, the "\" is followed by the name of the screen and the calling cycle in brackets.

The following screen shows a SINUMERIK 802D sl HMI where no changes have been made in the "cov.com" file.

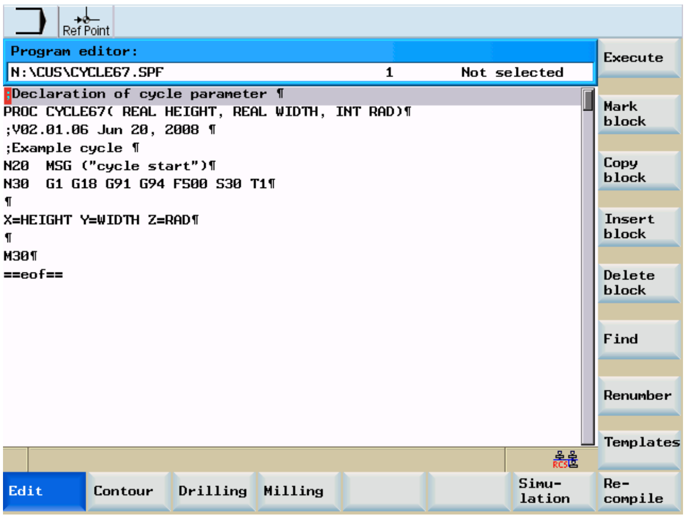


Figure 8-35 SINUMERIK 802D sl with unchanged version of "cov.com"

For our example the "cov.com" file is adapted as follows:

```
S3.0.0\$80271\
S3.1.0\$80282\CN2(CYCLE81)[MCALL]
S3.2.0\$80283\CN2(CYCLE82)[MCALL]
S3.3.0\$80284\CN3(CYCLE83)[MCALL]
S3.4.0\$80274\
S3.4.1\$80991\CN6(CYCLE85)[MCALL]
S3.4.2\$80537\CN7(CYCLE86)[MCALL]
S3.4.3\$80992\CN9(CYCLE87)[MCALL]
S3.4.4\$80993\CN9(CYCLE88)[MCALL]
S3.4.5\$80994\CN9(CYCLE89)[MCALL]
S3.4.6\$80415\[P "pr.dll" "programm" 10000]
S3.5.0\$80538\
S3.5.3\$80298\CN4(CYCLE84)[MCALL]
S3.5.4\$80297\CN4(CYCLE840)[MCALL]
S3.5.6\$80415\[P "pr.dll" "programm" 10000]
S3.6.0\$80415\[P "pr.dll" "programm" 10000]
S3.7.0\$80275\
S3.7.2\$80290\CN11(HOLES1)
S3.7.3\$80291\CN12(HOLES2)
S3.7.6\$80415\[P "pr.dll" "programm" 10000]
S4.0.0\$80272\
S4.2.0\$80340\CN27(CYCLE71)
S4.3.0\$80339\CN28(CYCLE72)
S4.5.0\$80634\
S4.5.4\$80292\CN25(POCKET3)
S4.5.5\$80293\CN26(POCKET4)
S4.6.0\$80541\
S4.6.2\$80296\CN13(LONGHOLE)
S4.6.3\$80294\CN14(SLOT1)
S4.6.4\$80295\CN15(SLOT2)
S4.7.0\$80299\CN18(CYCLE90)


S6.0.0\$85000\                      Selection: "Example SK_H_6"
S6.1.0\$85001\CN100(CYCLE67)        Cycle: "Cycle 67"
S6.2.0\$85010\                      Selection: "further SK-layer"
S6.2.1\$85001\CN100(CYCLE67)        Cycle: "Cycle 67"

M17
```

Figure 8-36 Extract from an adapted "cov.com" file

Table 8- 28 Example:

Line	Explanation
S6.0.0\$85000\	Horizontal softkey 6 is assigned the text behind alarm number 85000.
S6.1.0\$85001\CN100(CYCLE67)	If softkey 6 is activated, vertical softkey 1 is assigned the text behind alarm number 85001. When the softkey is activated, image CN100 and the cycle screen for CYCLE67 are called.

Note

A calling cycle screen must be specified in the description of the second vertical softkey level!

Example:

S6.1.1\$85000\ → incorrect!

S6.1.1\$85000\CN100(CYCLE67) → correct

When the edited file "cov.com" has been uploaded, the SINUMERIK 802D sl HMI is represented as follows:

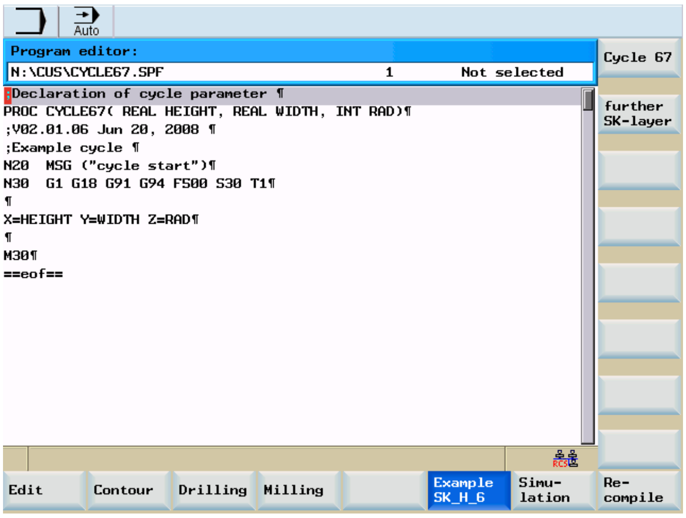


Figure 8-37 SINUMERIK 802D sl with adapted file "cov.com"

Horizontal softkey 6 and the vertical softkey bar correspond to the changes made in the file "cov.com".

8.13.9 Editing the user cycle screens

Introduction

The respective user cycle is described in the file "sc.com". The file describes the sequence and properties of the input parameters in the user cycle screens.

Operating sequence

1. Open the dialog box "Select OEM" via the menu "Extras" > "Toolbox Manager" > "Select OEM".
2. After successful upload the "sc.com" file is located in the "Cycles" tab in the "CST" directory for the toolbox project.

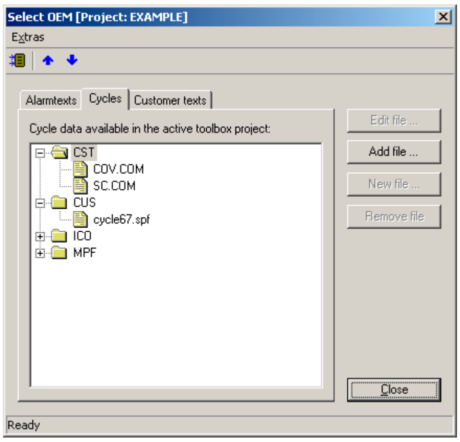


Figure 8-38 Dialog box "Select OEM"

3. Select the file and click "Edit file..." to open the file for editing.

```
//CN34(CYCLE77)
(R///$80020)[$80021/RTP]/B cn280
(R///$80022)[$80023/RFP]/B cn280
(R/0 99999.999//$80024)[$80025/SDIS]/B cn280
(R///$80956)[$80603/DP]/B cn280
(R/0 99999.999//$82161)[$80212/DPR]/B cn280
(R/0 99999.999//$80749)[$80158/PRAD]
(R///$80845)[$80082/PA]
(R///$80846)[$80082/PO]
(R/0 99999.999//$80923)[$80206/MID]/B cn280
(R/0 99999.999//$80947)[$80190/FAL]
(R/0 99999.999//$80744)[$80190/FALD]/B cn280
(R/0.001 99999.999//$80896)[$80117/FFP1]
(R/0.001 99999.999//$80895)[$80115/FFD]
(I/*0 1 2 3//$80404)[$80136/CDIR]
(I/*1 2//$82162)[$80039/VARI]
(R/0.001 99999.999//$80750)[$80336/AP1]


M17
```

Figure 8-39 Extract from a possible file "scv.com" for Milling technology

Layout of the cycle screen

The cycle screen has a uniform layout that cannot be modified (see screen below).

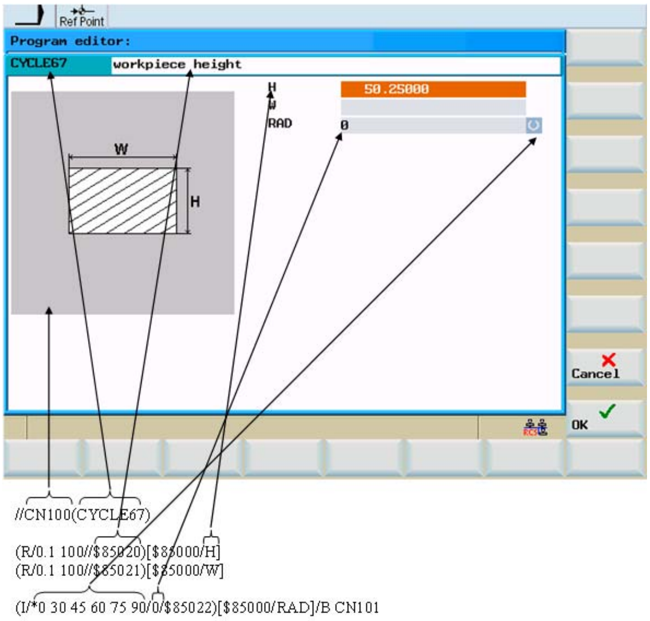


Figure 8-40 Screen for user cycle CYCLE67 and code for CYCLES67 in "sc.com"

Explanation of cycle screen

An image can be inserted on the left-hand side.

The input parameters are on the right.

The selected cycle (CYCLE67) and a cursor text ("workpiece height") are displayed in the header to provide additional information about the selected input parameter.

Syntax of the lines in the file "sc.com"

A line consists of parameters separated by backslashes.

"//" is written at the start of the first line to indicate the beginning of a cycle description. If an image is to appear on the left of the screen at cycle start, the required image is called here (CN100). It is in brackets behind the cycle name ((CYCLE67)).

Now the parameters are defined for the individual variables.

The following table describes the individual parameters that can be entered in the fields for the line:

Field in the line	Description of parameter	Entry in the line
1	Start of variable declaration	(
2	Variable type	R - REAL I - INTEGER C - CHAR S - STRING
3	Separator	/
4	EITHER Minimum and maximum	Two values must be entered: 1. Minimum 2. Maximum
4	OR Toggle field	* plus values A toggle field appears with fixed values which are entered behind the asterisk.
5	Separator	/
6	Default value	Value passed in the cycle if no entry is made.
7	Separator	/
8	Text	Appears in the dialog line
9	End of variable declaration	)
10	Start of description	[
11	Short text	Not used
12	Separator	/
13	Text in the screen	Text preceding the input screen. Max. 5 characters in length.
14	End of description	]
15	Line-specific image	/B name.bmp

Note

Separators, start and end identifiers must always be entered.

Fields 4, 6 and 15 can be left blank.

If no texts are stored via $Alarmnumber, three question marks appear in the associated fields on the screen.

For our example the "sc.com" file is adapted as follows:

```
//CN34(CYCLE77)
(R///$80020)[$80021/RTP]/B cn280
(R///$80022)[$80023/RFP]/B cn280
(R/0 99999.999//$80024)[$80025/SDIS]/B cn280
(R///$80956)[$80603/DP]/B cn280
(R/0 99999.999//$82161)[$80212/DPR]/B cn280
(R/0 99999.999//$80749)[$80158/PRAD]
(R///$80845)[$80082/PA]
(R///$80846)[$80082/PO]
(R/0 99999.999//$80923)[$80206/MID]/B cn280
(R/0 99999.999//$80947)[$80190/FAL]
(R/0 99999.999//$80744)[$80190/FALD]/B cn280
(R/0.001 99999.999//$80896)[$80117/FFP1]
(R/0.001 99999.999//$80895)[$80115/FFD]
(I/*0 1 2 3//$80404)[$80136/CDIR]
(I/*1 2//$82162)[$80039/VARI]
(R/0.001 99999.999//$80750)[$80336/AP1]

//CN100(CYCLE67)
(R/0.1 100//$85020)[$85000/H]
(R/0.1 100//$85021)[$85000/W]
(I/*0 30 45 60 75 90/0/$85022)[$85000/RAD]/B CN101

M17
```

Figure 8-41 Extract from the adapted "sc.com" file

Table 8- 29 Example

Line	Explanation
//CN100(CYCLE67)	Start identifier (//), Image selection (CN100), Cycle call ((CYCLE67))

Table 8- 30 Declaration of the first input value

Line	Explanation
(R/0.1 100//$85020)[$85000/H]	Data type: Real, Validity range: 0.1 to 100, No default value, Text in cursor line: Alarm text number 85020, $85000 - parameter currently not in use, Text preceding the input field: H

If a new image is to appear when a parameter is selected, this image is called at the end of the variable declaration. "/B" is here the keyword and cannot be omitted.

Table 8- 31 Declaration of the third input value

Line	Explanation
(I/*0 30 45 60 75 90/0/$85022)[$85000/RAD]/B CN101	Data type: Integer, Toggle field with values 0, 30, 45, 60, 75, 90, Default setting for toggle field: 0, Text in dialog line: Alarm text number 85022, $85000 - parameter currently not in use, Text preceding the input field: RAD, Image selection: Image CN101 is displayed

Note

The effect of the declared values on the HMI can be seen in the image "Screen for user cycle CYCLE67 and code for CYCLES67 in "sc.com"" in the section "Layout of the cycle screen".

8.13.10 Creating the images for the user cycle screens

Introduction

The cycle icons must be stored as bitmap files (*.bmp) sized to 222*222 pixels as 16-color bitmaps.

The icon name must not exceed 32 characters including file extension (e.g. CN100.bmp).

Prerequisite

You have set the editing program to edit the image files in RCS802 (see "Required settings in RCS802").

Operating sequence

1. The new bitmap files are created with RCS802 via the "Select OEM" dialog box (menu "Extras" > "Toolbox Manager" > "Select OEM").

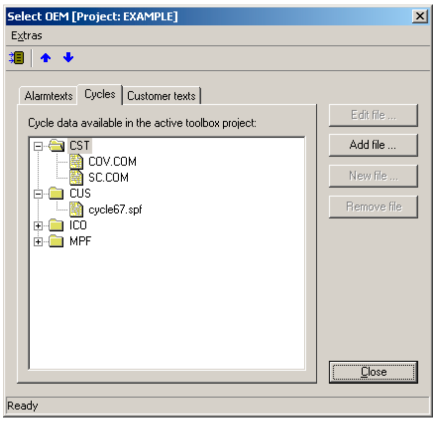


Figure 8-42 Dialog box "Select OEM"

2. Select directory "ICO" in the "Cycles" tab.
3. Click the "New file…" button.

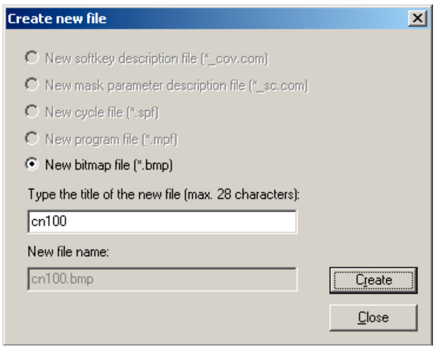


Figure 8-43 Dialog box "Create new file"

4. Assign a name for the image file.
5. Click the "Create" button.

   The image file is created in the toolbox project.

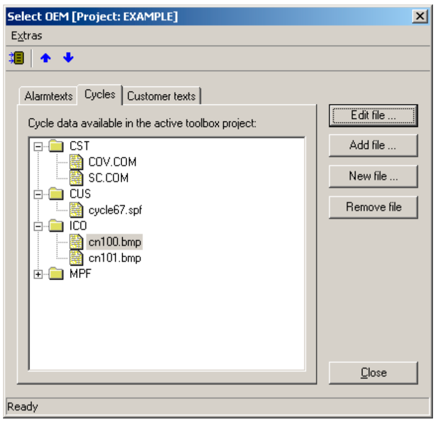


Figure 8-44 Dialog box "Select OEM" with image file

6. Select the new file in the "ICO" directory.
7. Click the "Edit file" button.

   The file opens for editing.

Note

If 16 colors are not sufficient for the display, you can also use 24-bit color depth bitmaps.

8.13.11 Transmission of the user files into the control

Operating sequence

There is a connection from the RCS802 tool to the SINUMERIK 802D sl control.

1. Select the menu "Extras" > "Write files to 802…" in the dialog "Select OEM" or press the following icon:

2. Select the user files to be transferred in the left directory tree of the dialog "Write files to connected Sinumerik 802D sl".
3. Press "Add".

   The files are added to the selection list on the right.
4. Press the button "Write".

   Transmission into the control begins.

   At least the NCK protection level "Manufacturer" must be set on the 802D sl.

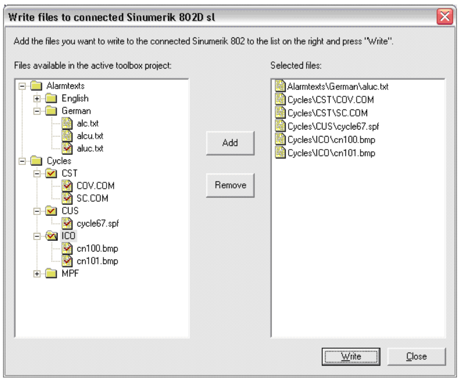


Figure 8-45 Dialog box "Write files to connected SINUMERIK 802D sl"

Storage of the files in the control

- The user text language databases are stored in the manufacturer's drive in the "lng\xxx" directory (where xxx stands for the corresponding language abbreviation, e.g. "deu" for German (Deutsch)).
- The HMI control files "cov.com" and "sc.com" are stored in the NCK directory "CST".
- The cycle files (*.spf) are stored in the NCK directory "CUS".
- The bitmap archive file generated from the graphics "cus_bmp.arj" is stored in the manufacturer drive, in the directory "ico". (New bitmap files are automatically added to the archive "cus_bmp.arj". Existing files with the same name will be overwritten.)
- The NC program files (*.mpf) are stored in the NCK directory "MPF.

Note

There may be an automatic HMI restart after the user data has been transferred into the control!

8.14 Generating user dialogs

8.14.1 Generating user menus

8.14.1.1 Generating softkey menus and dialog forms

User menus can only be inserted if there is a main-menu tag with the name "main" in the XML description. This tag is called by the system after the <CUSTOM> operating area has been activated. Further menu branches and dialog-box activation can be defined within the tag.

```
<menu name= "MAIN">
<OPEN_FORM name= "main dialogue">
<softkey POSITION="1">
<caption>sub menu 1</caption>
<navigation>sub menu 1</navigation>
</softkey>
<softkey POSITION="8">
<caption>sub menu 8</caption>
<navigation>sub menu 8</navigation>
</softkey>
</menu>

<menu name= "sub menu 1">
<OPEN_FORM name= "dialogue 1">
</menu>

<menu name= "sub menu 8">
<OPEN_FORM name= "dialogue 8“>
</menu>
```

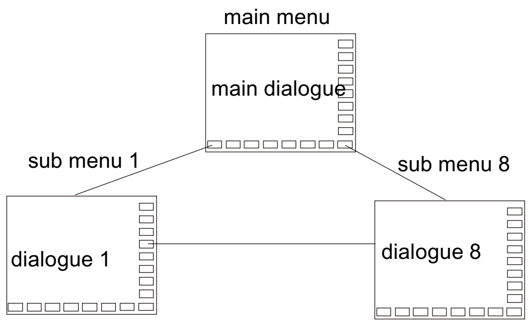
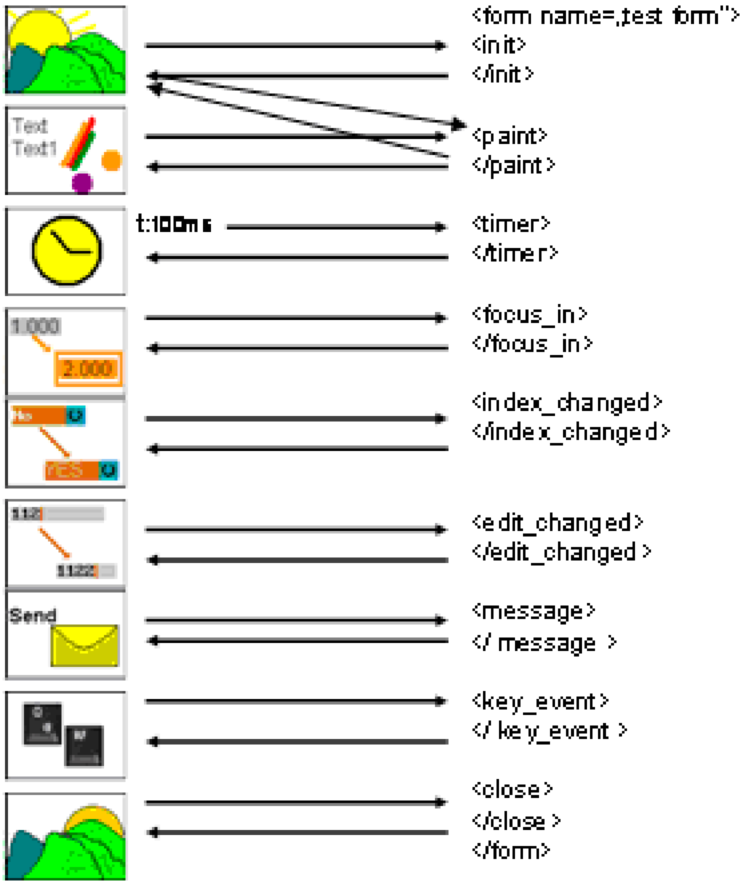


Figure 8-46 Menu structure

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>FORM</td><td>This tag contains the description of a user dialog.

Attributes:
color
Background color of the dialog box (color coding, see Chapter, color coding)
name
Identifier of the form
type
cycle -attribute specifies a cycle form
xpos
X-position of the top left corner of the dialog box (optional)
ypos
Y position of the top left corner (optional)
width
Extension in the X direction (in pixels) (optional)
height
Extension in the Y direction (in pixels) (optional)</td></tr>
<tr><td>FORM continued</td><td>Dialog messages:
INIT
PAINT
TIMER
CLOSE
FOCUS_IN
INDEX_CHANGED (from SW 1.4 SP7)
EDIT_CHANGED (from SW 1.4 SP7)
KEY_EVENT (from SW 1.4 SP7)
MESSAGE (from SW 1.4 SP7)</td></tr>
</table>

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>FORM continued</td><td>
</td></tr>
</table>

Tag identifier	Meaning
FORM continued	**Syntax:** <FORM name = "<dialog name>" color = "#ff0000"> **Example:** <FORM name = "R-Parameter"> <INIT> <DATA_ACCESS type = "true" /> <CAPTION>R - Parameter</CAPTION> <CONTROL name = "edit1" xpos = "322" ypos = "34" refvar = "nck/Channel/Parameter/R[1]" /> <CONTROL name = "edit2" xpos = "322" ypos = "54" refvar = "nck/Channel/Parameter/R[2]" /> <CONTROL name = "edit3" xpos = "322" ypos = "74" </INIT> <PAINT> <TEXT xpos = "23" ypos = "34">R - Parameter 1</TEXT> <TEXT xpos = "23" ypos = "54">R - Parameter 2</TEXT> <TEXT xpos = "23" ypos = "74">R - Parameter 3</TEXT> </PAINT> </FORM>
INIT	**Dialog box message** The tag is executed immediately after the dialog box is generated. All the input elements and hotlinks for the dialog form should be created here.

Tag identifier	Meaning
KEY_EVENT From SW 1.4 SP7	Dialog message The tag KEY_EVENT can be integrated in the form to evaluate keyboard events. The system sends the MF2 keyboard code to the active form if the tag is available in a form. If the variable **$actionresult** is not set to zero, the system then subsequently processes the keyboard event. The keyboard code is provided in the variable **$keycode** as an integer value. **Example:** The character entered into the variable **exclude_key** should be filtered-out of the input stream.

```
<LET name="stream" type="string"/>
<LET name="exclude_key" type="string"/>

<FORM name = "keytest_form">
  <INIT>
    <CONTROL name = "p1" xpos = "120" ypos = "84" width ="200"
refvar="stream" hotlink="true" />
    <CONTROL name = "p2" xpos = "160" ypos = "104" width ="8"
refvar="exclude_key" hotlink="true" />
  </INIT>

  <PAINT>
    <text xpos = "8" ypos = "84">data stream</text>
    <text xpos = "8" ypos = "104">exclude key</text>
  </PAINT>

  <KEY_EVENT>
    <LET name="excl_keycode" type="string"/>
    <OP>excl_keycode = exclude_key</OP>
    <type_cast name="excl_keycode" type="int" />
    <PRINT text="%d %d">$keycode, excl_keycode</PRINT>

    <IF>
      <CONDITION>$keycode == excl_keycode</CONDITION>
      <THEN>
        <op> $actionresult = 0</op>
      </THEN>
    </IF>

  </KEY_IVENT>

</FORM>
```

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>MESSAGE
From SW 1.4 SP7</td><td>Dialog message
If the Send_message operation is executed in the script, then the parser processes the tag message. Values P1 and P2 are provided in the variables $message_par1 and $message_par2 (see the "SEND_MESSAGE" tag).

Syntax:
<pre><MESSAGE>
</MESSAGE></pre>
Example:
<pre><LET name="user_selection" />

<SOFTKEY POSITION="3">
   <CAPTION>Set%nParameter</CAPTION>
   <SEND_MESSAGE>1, 10</SEND_MESSAGE>
</SOFTKEY>
…
…

<FORM>
…
…
<MESSAGE>
<SWITCH>
<CONDITION>$message_par1</CONDITION>
<CASE value="1">
<OP> user_selection = $message_par2 </OP>
…
…
</CASE>
</SWITCH>
</MESSAGE>
…
…
</FORM></pre></td></tr>
</table>

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>SEND_MESSAGEFrom SW 1.4 SP7</td><td>The tag sends a message with two parameters to the active form, which is processed in the tag message (see also MESSAGE).

Syntax:
<code><SEND_MESSAGE>p1, p2</SEND_MESSAGE></code>

Example:
<code><LET name="user_selection" /></code>

<code><SOFTKEY POSITION="3"></code>
<code>   <CAPTION>Set%nParameter</CAPTION></code>
<code>   <SEND_MESSAGE>1, 10</SEND_MESSAGE></code>
<code></SOFTKEY></code>
…
…
<code><FORM></code>
…
…
<code><MESSAGE></code>
<code><SWITCH></code>
<code><CONDITION>$message_par1</CONDITION></code>
<code><CASE value="1"></code>
<code><OP> user_selection = $message_par2 </OP></code>
…
<code></CASE></code>
<code></SWITCH></code>
<code></MESSAGE></code>
…
…
<code></FORM></code></td></tr>
<tr><td>FOCUS_IN</td><td>Dialog box message
The tag is called if the system places the focus on a control. In order to identify the control, the system copies the name of the control to variable $focus_name and the value of the attribute item_data to variable $focus_item_data. The system creates the variables automatically.
This message can be used, for example, to output images depending on the focus position.
Example:
<code><focus_in></code>
<code><PRINT text="focus on filed:%s, %d">$focus_name,</code>
<code>$focus_item_data </PRINT></code>
<code></focus_in></code></td></tr>
<tr><td>PAINT</td><td>Dialog box message
The tag is executed when the dialog box is displayed. All the texts and images which are to be displayed in the dialog box should be specified here.
Further, the tag is executed if the system identifies that parts of the dialog box are to be redisplayed. For example, this can be initiated by closing high-level windows.</td></tr>
</table>

Tag identifier	Meaning
TIMER	Dialog box message The tag is executed cyclically. Each form is assigned a timer that initiates that the timer - tag is executed approx. every 100 ms.

CAPTION	The tag contains the title of the dialog box.
	This tag should be used within the INIT tag.
	
	Syntax:
	`<CAPTION>Titel</CAPTION>`
	
	Example:
	`<CAPTION>my first dialogue</CAPTION>`

CLOSE	Dialog box message This tag is executed before the dialog box is closed.

CLOSE_FORM	The tag closes the active dialog.
	This instruction is only necessary if it involves a cycle dialog that is used in the program editor area. Generally, dialogs are automatically managed and do not have to be explicitly closed.
	
	Syntax:
	`<CLOSE_FORM/>`
	
	Example:
	`<softkey_ok>`
	` <caption>OK</caption>`
	` <CLOSE_FORM />`
	` <navigation>main_menu</navigation>`
	`</softkey_ok>`

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>CONTROL</td><td>The tag is used to generate control elements.
Syntax:
<code><CONTROL name = "<control name>" xpos = "<X position>" ypos = "<Y position>" refvar = "<NC variable>" hotlink = "true" format = "<format>" /></code>

Attributes:
• name
Identifier of the field.
The identifier simultaneously represents a local variable, and must not be used a multiple number of times in the form.
• xpos
X position of the top left corner
• ypos
Y position of the top left corner
• fieldtype
Field type
If no type is specified, the field is set as an edit field.
– edit
Data can be changed
– readonly
Data cannot be changed
combobox
The field displays the corresponding identifiers instead of numerical values.
If the field type "combobox" is selected, then the expressions to be displayed must also be assigned to the field.
The <ITEM> TAG should be used for this purpose.
The combo box saves the index of the currently selected text in the variable belonging to the control (see the attribute refvar).
– progressbar
A progress bar with a value range of 0 to 100 appears.
The valley value and peak value properties can be used to adapt the value range to the data to be displayed.</td></tr>
</table>

Tag identifier	Meaning
CONTROL continued	• **fieldtype** – **listbox** The field type generates an empty list box control. Using the tag **<ITEM>** a list box element can be inserted in the list box. The ITEM attribute **value** allows this element to be assigned a unique value. For example, this can be used to identify the element. Parameters **width** and **height** specify the width and height of the list box. After the control has been created, additional list box elements can be inserted using the functions **AddItem**, **InsertItem** or **LoadItem**. – **graphicbox** The field type generates a 2d broken line graphic control. Using the tag **<ITEM>** a graphic element can be inserted into the control. Parameters **width** and **height** specify the width and height of the box. **Note:** This control is not linked into the clipping. This means that other elements can cover this control. After the control has been created, additional elements can be inserted using the functions **AddItem** or **InsertItem**. The parameter **itemdata** is not evaluated for this control. – **itemlist** The field type generates a static control, which displays the corresponding identifier instead of numerical values. The **<ITEM>** tag can be used to assign an identifier to the field. • **item_data** A user-specific integer value can be assigned to the attribute. This value is given as part of the FOCUS_IN message for identifying the focus field. • **refvar** Identifier of the reference variable that can be linked to the field (optional). • **hotlink** = "TRUE" " If the value of the reference variable changes, then the field is automatically updated (optional). • **format** The attribute defines the display format of the specified variable. Formatting data, see print-Tag (optional).

Tag identifier	Meaning
CONTROL continued	Attributes: • time specifies the data refresh rate (optional). The following specifications are possible: – super fast Refresh time < 100 ms – fast Refresh time approx. 100 ms – normal Refresh time approx. 200 ms – slow Refresh time approx. 500 ms • font The attribute defines the font size used. – 0: 8*8 – 1: 16*8 – 2: 24*16 (only numbers) – 3: 8*8 double the character height – 4: 16*8 double the character height – 5: 24*16 double the character height (only numbers) • color_bk The attribute sets the background color of the control. • color_fg The attribute sets the foreground color of the control. ("color coding" see Chapter "Color coding (Page 241)") • display_format The attribute defines the processing format of the specified variable. This attribute must be used when accessing a PLC float variable, as the access is realized by reading a double word. The following data formats are permitted: – FLOAT – INT – DOUBLE – STRING Assigning expressions (e.g. text or graphic element to be displayed) to a list box, graphics box or combo box: Syntax: <ITEM>Expression</ITEM> <ITEM value ="<Value>">Expression</ITEM>

Tag identifier	Meaning

CONTROL continued	**Example:**
	`<CONTROL name = "button1" xpos = "10" ypos = "10" fieldtype = " combobox ">`
	`<ITEM>text1</ITEM>`
	`<ITEM>text2</ITEM>`
	`<ITEM>text3</ITEM>`
	`<ITEM>text4</ITEM>`
	`</CONTROL>`
	If any integer value is to be assigned to an expression, the attribute **value = "value"** should be added to the tag.
	Rather than consecutive numbers, the control variable now contains the item's assigned value.
	
	Example:
	`<CONTROL name = "button1" xpos = "10" ypos = "10" fieldtype = " combobox ">`
	`<ITEM value = "10">text1</ITEM>`
	`<ITEM value = "20">text2</ITEM>`
	`<ITEM value = "12">text3</ITEM>`
	`<ITEM value = "1">text4</ITEM>`
	</CONTROL>
	Example of a progress bar:
	`<CONTROL name = "progress1" xpos = "10" ypos = "10" width = "100" fieldtype = "progressbar" hotlink = "true" refvar = "nck/Channel/GeometricAxis/actProgPos[1]">`
	`   <PROPERTY min = "0" />`
	`   <PROPERTY max = "1000" />`
	`</CONTROL>`
	Example, list box:
	`<let name="item_string" type="string"></let>`
	`<let name="item_data" ></let>`
	
	`<CONTROL name="listbox1" xpos = "360" ypos="150" width="200" height="200" fieldtype="listbox" />`
	• Adding elements:
	Elements are added using the function **additem** or **loaditem**.
	• Deleting the content:
	The content is deleted using the function **empty**.
	
	`<op> item_string = _T"text1\\n" </op>`
	`<function name="control.additem">_T"listbox1", item_string, item_data </function>`
	`<op> item_string = _T"text2\\n" </op>`
	`<function name="control.additem">_T"listbox1", item_string, item_data </function>`

Tag identifier	Meaning
CONTROL continued	**Example, graphic box:** `<CONTROL name= "graphic" xpos = "8" ypos="23" width="300" height="352" fieldtype="graphicbox" />` • Adding elements: Elements are added using the function **additem** or **loaditem**. The following 2d elements can be used: – Line - l(inc) – Circle sector - c(ircle) – Point - p(oint) **Structure of an element:** <Element type>; coordinates • Line: l; xs; ys; xe, ye I - line marking Xs - X start position Ys - Y start position Xe - X end position Ye - Y end position • Circle: C, xs, ys, xe, ye, cc_x, cc_y, r C - circular sector marking Xs - X start position Ys - Y start position Xe - X end position Ye - Y end position Cc_x – X coordinate, circle center point CC_y – Y coordinate circle center point • Radius: R • Point: P, x, y P - point marking X - X position Y - Y position • Deleting the graphic: The content is deleted using the function **empty**.

Tag identifier	Meaning

CONTROL continued	**Example:**
	`<let name="item_string" type="string"></let>`
	`<let name="s_z" type="double">100</let>`
	`<let name="s_x" type="double">50</let>`
	`<let name="itemdata"></let>`
	…
	…
	`<control name= "gbox" xpos = "6" ypos="24"  width="328" height="356" fieldtype="graphicbox" />`
	…
	…
	`<print name ="item_string" text="p; %f; %f">s_z, s_x</print>`
	`<function name="control.additem">_T"gbox", item_string, itemdata</function>`
	…
	
	Example itemlist:
	`<CONTROL name = "itemlist1" xpos = "10" ypos = "10" fieldtype = " itemlist">`
	`<ITEM value = "10">text1</ITEM>`
	`<ITEM value = "20">text2</ITEM>`
	`<ITEM value = "12">text3</ITEM>`
	`<ITEM value = "1">text4</ITEM>`
	`</CONTROL>`

HELP_CONTEXT	This tag defines the help topic to be called. It should be programmed in the INIT block.
	The name specified in the attribute is supplemented by the prefix XmlUserDlg_ and is transferred to the help system. The associated structure of the help file should be taken from the topic - generating an online help.
	Sequence when activating the help system:
	1. Press the "Info" key.
	2. The dialog supplies the expression "my_dlg_help".
	3. Parser converts the expression into "XmlUserDlg_my_dlg_help" .
	4. Activating the help system.
	5. Submitting the search term "XmlUserDlg_my_dlg_help".
	
	Syntax:
	`<HELP_CONTEXT name="<context name>" />`
	
	Example:
	...
	`<INIT>`
	...
	`<CAPTION>my dialogue</CAPTION>`
	`<HELP_CONTEXT name="my_dlg_help" />`
	...
	`</INIT>`

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>DATA_ACCESS</td><td>The tag controls the behavior of the dialog forms when user inputs are being saved.
The behavior should be defined within the INIT tag.
If the tag is not used, inputs are buffered in each case.
Exception: The attribute hotlink is set to true .

Attribute::
• type = "TRUE" – the input values are not buffered. The dialog form copies the input values to the reference variables directly.
• type = "FALSE" – the values are only copied to the reference variable with the UPDATA_DATA type = "FALSE" tag.

Example:
<code><DATA_ACCESS type = "true" /></code></td></tr>
<tr><td>EDIT_CHANGED
From SW 1.4 SP7</td><td>Dialog box message
This tag is called if the contents of an edit control have changed.
To identify the control, the system copies the name of the control into variable $focus_name and the value of the attribute item_data into variable $focus_item_data. The system creates the variables automatically.

Example:
<code><EDIT_CHANGED></code>
<code><print text="index changed filed:%s, %d"> $focus_name,</code>
<code>$focus_item_data </print></code>
…
…
<code></EDIT_CHANGED></code></td></tr>
<tr><td>INDEX_CHANGED
From SW 1.4 SP7</td><td>Dialog box message
The tag is called, if the operator changes the selection of a combo box.
To identify the control, the system copies the name of the control into the variable
$focus_name and the value of the attribute item_data into the variable $focus_item_data. The system creates the variables automatically.
Note:
A reference variable assigned to the control, has not been aligned to the control variable at this point in time and contains the index of the previous selection of the combo box.

Example:
<code><INDEX_CHANGED></code>
<code><print text="index changed filed:%s, %d"> $focus_name,</code>
<code>$focus_item_data </print></code>
…
…
<code></INDEX_CHANGED></code></td></tr>
</table>

Tag identifier	Meaning

MENU	The tag defines a menu containing the softkey description and the dialog to be opened.
	
	Attribute::
	• **name**
	Menu name
	
	Syntax:
	`<MENU name = "<menu name>">`
	…
	`<open_form …>`
	…
	`<SOFTKEY …>`
	`</SOFTKEY>`
	`</MENU>`

NAVIGATION	This tag defines the menu to be called. This tag can only be set within a softkey block.
	
	Syntax:
	`<NAVIGATION>menu name</NAVIGATION>`
	
	Example:
	`<menu name = "main">`
	`     <softkey POSITION="1">`
	`         <caption>sec. form</caption>`
	`         <navigation>sec_menu</navigation>`
	`    </softkey>`
	`</menu>`
	
	`<menu name = "sec_menu">`
	`     <open_form name = "sec_form" />`
	`    <softkey_back>`
	`        <navigation>main</navigation>`
	`    </softkey_back>`
	</menu>

Tag identifier	Meaning

OPEN_FORM	The tag opens the dialog form given under the name.
	
	Attribute::
	• name
	Name of the dialog form
	
	Syntax:
	`<OPEN_FORM name = "<form name>" />`
	
	Example:
	`<menu name = "main">`
	`     <open_form name = "main_form" />`
	`    <softkey POSITION="1">`
	`        <caption>main form</caption>`
	`        <navigation>main</navigation>`
	`    </softkey>`
	`</menu>`
	
	`<form name="main_form">`
	`    <init>`
	`    </init>`
	
	`    <paint>`
	`    </paint>`
	
	`</form>`

Tag identifier	Meaning

PROPERTY	This tag can be used to define additional properties for an operator control.
	
	Attributes:
	• **max** = "<maximum value>"
	• **min** = "<minimum value>"
	• **default** = "<pre-assignment>"
	• **factor** = "conversion factor"
	• **color_bk** = "<background color coding>"
	• **color_fg** = "<font color coding>"
	• **font** = "<Font number>"
	• **password** = "<true>" - entered character is displayed with "*"
	• **multiline** = "<true>" - permits multi-line inputs in an edit control
	• **disable** = "<true/false>" - locks/permits the input in an edit control
	• **abscissa** = "axis name of the first axis of the plane" (only valid for **graficbox**)
	• **ordinate** = "axis name of the second axis of the plane" (only valid for **graficbox**)
	• **transparent**= "Transparent color of a bitmap"
	Color coding (for details on color coding, see chapter, Color coding)
	• **accesslevel** = changing is only permitted for an access level (NC) that is less than or equal to the specified access level.
	The syntax is: <property accesslevel=“access level“ />
	Also see: Access levels (Page 104)
	• **tooltip** = information text is displayed if the cursor is set to the control.
	The syntax is: <property tooltip="information text" />
	Example:
	`<CONTROL name = "progress1" xpos = "10" ypos = "10" width = "100" fieldtype = "progressbar" hotlink = "true" refvar = "nck/Channel/GeometricAxis/actProgPos[1]">`
	`   <PROPERTY min = "0" />`
	`   <PROPERTY max = "1000" />`
	`</CONTROL>`
	`<CONTROL name = "edit1" xpos = "10" ypos = "10">`
	`<PROPERTY min = "20" />`
	`<PROPERTY max = "40" />`
	`<PROPERTY default = "25" />`
	`</CONTROL>`

Tag identifier	Meaning
PROPERTY continued	**Example of "abscissa" and "ordinate":** <LET name="abscissa" type="string"></LET> <LET name="ordinate" type="string"></LET> <FORM> … … <INIT> <OP>abscissa = "nck/Channel/GeometricAxis/name[0]"</OP> <OP>ordinate = "nck/Channel/GeometricAxis/name[1]"</OP> <CONTROL name="_802_c_gbox" xpos="6" ypos="24" width="328" height="356" fieldtype="graphicbox" > <property abscissa="$$$abscissa" /> <property ordinate="$$$ordinate" /> </CONTROL> … </INIT> … </FORM> **Example for "accesslevel":** <control name="c_1" xpos10" ypos="70" width="96" > <property accesslevel="3" /> </control> **Example for "tooltip"**: <control name="c_1" xpos10" ypos="70" width="96" type="string"> <property tooltip="cursor text 1" /> </control>

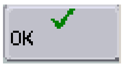
Tag identifier	Meaning
SOFTKEY	The tag defines the properties and responses of a softkey. Attributes: • position Number of the softkey. 1-8 horizontal softkeys, 9-16 vertical softkeys The following attributes become effective from SW 1.4 SP7 and higher: • type Defines the property of the softkey. user_controled - The script defines how the softkey is displayed toggle_softkey - The softkey is displayed alternating between pressed and not pressed • refvar Should only be used in conjunction with toggle_softkey . Reference variable, into which the actual softkey property is copied. A variable, type "String" should be specified, which includes the properties pressed, not pressed or locked (see tag state). • picture Using the attribute, a bitmap can be output left justified on the softkey. The complete path name should be specified. The number of text characters that can be displayed is reduced to the width of the bitmap. The following additional actions can be defined within the softkey block: • caption Softkey text • state Should only be used in conjunction with user_controlled . The tag assigns the required softkey display to the system. Syntax: <state type="<state>" /> The following strings can be specified: – notpressed The softkey is displayed as being not pressed. – pressed The softkey is displayed as being pressed. – disabled The softkey is locked and is displayed in gray. • navigation • update_controls • function

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>SOFTKEY continued</td><td>Syntax:
Standard softkey:
<pre>
<state type="<softkey state>" />
<softkey position = "<1>">
…
…
</softkey>
or
Script-controlled softkey:
<softkey position = "<1>" type="<user_defined>" >
<state type="<softkey state>" />
…
…
</softkey>

or
Toggle softkey:
<softkey position = "<1>" type="<toggle_softkey>" refvar="<variable
name>" >
…
…
</softkey>
</pre>
Example:
<pre>
<let name="define_sk_type" type="string">PRESSED</let>
<let name="sk_type">1</let>

<softkey POSITION="1" type="user_controled" >
    <caption>Toggle%nSK</caption>
    <if>
    <condition>sk_type == 0 </condition>
    <then>
        <op> sk_type = 1 </op>
        <op> define_sk_type = _T"PRESSED" </op>
    </then>
    <else>
        <op> define_sk_type = _T"NOTPRESSED" </op>
        <op> sk_type = 0 </op>
    </else>
    </if>
    <state type="$$$define_sk_type" />
…
…

</softkey>
</pre></td></tr>
</table>

Tag identifier	Meaning
SOFTKEY continued	**Example:** or <let name="curr_softkey_state" type="string">PRESSED</let> </softkey> <softkey POSITION="3" type="toggle_softkey" refvar="curr_softkey_state"> <caption>Toggle%nSK</caption> … </softkey>  
SOFTKEY_OK From SW 1.4 SP7	The tag defines the response of the softkey "OK".  The following additional actions can be defined within the softkey block: • navigation • update_controls • function Syntax: <SOFTKEY_OK> … … </SOFTKEY_OK>
SOFTKEY_CANCEL From SW 1.4 SP7	The tag defines the response of the softkey "Cancel".  The following additional actions can be defined within the softkey block: • navigation • update_controls • function Syntax: <SOFTKEY_CANCEL> … … </SOFTKEY_CANCEL>

Tag identifier	Meaning
SOFTKEY_BACK From SW 1.4 SP7	The tag defines the response of the softkey "Back".  The following additional actions can be defined within the softkey block: • navigation • update_controls • function Syntax: <SOFTKEY_BACK> ... ... </SOFTKEY_BACK>
SOFTKEY_ACCEPT From SW 1.4 SP7	The tag defines the response of the softkey "Accept".  The following additional actions can be defined within the softkey block: • navigation • update_controls • function Syntax: <SOFTKEY_ACCEPT> ... ... </SOFTKEY_ACCEPT>

Tag identifier	Meaning
TEXT	The tag is used to display a text in the specified position. If an alarm number is used, the dialog box displays the text which is saved for the number. Syntax: `<TEXT xpos = "<X position>" ypos = "<Y position>"> Text </TEXT>` Attributes: • xpos X position of the top left corner • ypos Y position of the top left corner • color Text color (color coding, see Chapter Color coding (Page 241)) Value: Text to be displayed

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>IMG</td><td>
The tag is used to display an image in the specified position. The BMP and PNG image formats are supported.

Syntax:
<code><IMG xpos = "<X position>" ypos = "<Y position>" name = "<name>" /></code>
<code><IMG … xrot="angle X axis" yrot=" angle Y axis " zrot=" angle Z axis " /></code>

Attributes:

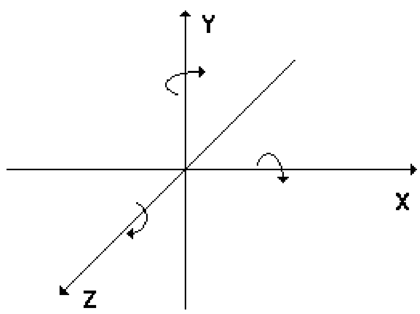
xpos

X position of the top left corner
ypos

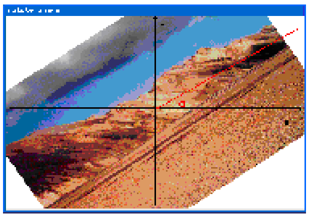
Y position of the top left corner
name

complete path name
transparent

Transparent color of the bitmap (see Chapter "Color coding"):
xrot, yrot, zrot
in the axes X, Y and Z, the image rotates around a point of rotation.
xref, yref, zref
Define point of rotation
If a point of rotation is not specified, then the image rotates around the image center point. In this case, the Z plane is 0.



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

Tag identifier	Meaning
IMG Continued	**Example**: The image is rotated through 34 degrees around the Z axis: <img xpos = "5" ypos = "23" name = "f:/appl/pic3.bmp" height="355" width="550" zrot="34" color_bk="#ffffff" > </img>  **Optional:** If the image display is to differ from the original size, the dimensions can be defined using the attributes **width** and **height**. • **width** Width in pixels • **height** Height in pixels **Examples:** <IMG xpos = "20" ypos = "40" name = "f:/appl/test.bmp" /> <IMG xpos = "5" ypos = "23" name = "f:/appl/test.bmp" height = "355" width = "550"/>

Tag identifier	Meaning
BOX	The tag draws a rectangle at the specified position, colored as indicated. Syntax: `<BOX xpos = "<X position>" ypos = "<Y position>" width = "<X extension>" height = "<Y extension>" color = "<Color code>" />` Attributes: • **xpos** X position of the top left corner • **ypos** Y position of the top left corner • **width** Extension in X direction (in pixels) • **height** Extension in Y direction (in pixels) • **color** Color coding (for details on color coding, see chapter, Color coding)

FUNCTION	Function call
	The tag executes the function body, which is specified under the attribute "**name**.
	
	Attributes:
	• **name** = "Name of the function body"
	• **return** = "Variable name for saving the result of the function"
	
	Values:
	List of variables to be transferred to the function body. The variables must be separated by a comma. A maximum of 10 parameters can be transferred.
	It is also possible to specify constants or text expressions as call parameters. The identifier _T should be placed at the start as a means of identifying text terms.
	
	Syntax:
	`<FUNCTION name = "<function name>" />`
	Calling function expects a return value
	`<FUNCTION name = "<function name>" return = "<Variablenname>" />`
	Parameter transfer
	`<FUNCTION name = "<function name>"> var1, var2, var3 </FUNCTION>`
	`<FUNCTION name = "<function name>"> _T"Text", 1.0, 1 </FUNCTION>`
	
	Examples:
	See "FUNCTION_BODY".

Tag identifier	Meaning
FUNCTION_BODY	Function body The tag contains the function body of a subfunction. The function body needs to be programmed within the DialogGui tag. **Attributes:** • **name** = "Name of the function body" • **parameter** = "Parameter list" (optional) The attribute lists the transfer parameters that are required. The parameters must be separated by a comma. When the function body is called, the values of the parameters specified in the function call are copied to the transfer parameters listed. • **return** = "true" If the attribute is set to true then the local variable $return is created. The function's return value which is forwarded to the calling function on quitting the function should be copied to this variable. **Syntax:** **Function body without parameter** <FUNCTION_BODY name = "<function name>"> … … … </ FUNCTION_BODY> **Function body with parameter** <FUNCTION_BODY name = "<function_name>" parameter = "<p1, p2, p3>"> … <LET name = "tmp></LET> <OP> tmp = p1 </OP> … </FUNCTION_BODY> **Function body with return value** <FUNCTION_BODY name = "<function_name>" parameter = "<p1, p2, p3>" return = "true"> … <LET name = "tmp></LET> <OP> tmp = p1 </OP> … <OP> $return = tmp </OP> </FUNCTION_BODY>

Tag identifier	Meaning
FUNCTION_BODY continued	**Example:** <function_body name = "test" parameter = "c1,c2,c3" return = "true"> <LET name = "tmp">0</LET> <OP> tmp = c1+c2+c3 </OP> <OP> $return = tmp </OP> </function_body> … … … <LET name = "my_var"> 4 </LET> <function name = "test" return = " my_var "> 2, 3,4</function> <print text = "result = %d"> my_var </print> … ...

Tag identifier	Meaning

REQUEST	The tag is used to add a variable to the cyclic reading service (hotlink). As a consequence, the access time to variables, which are not linked to the control, is reduced.
	If a function is to be called automatically when a value changes, then the name of the function should be specified as an additional attribute
	This tag is only processed within the INIT operation.
	
	Attributes:
	• **name**
	Address identifier
	• **function**
	Function name (from SW 1.4 SP7)
	
	Syntax:
	`<REQUEST name = "<NC-Variable>" />`
	`or`
	
	`<REQUEST name = "<NC-Variable>" function="<function name>"/>`
	
	Example:
	`<request name ="plc/mb10" />`
	
	or
	
	`<function_body name="my_function" >`
	`<print text="value changed" />`
	`</function_body>`
	…
	…
	…
	`<request name ="plc/mb10" function="my_function"/>`

UPDATE_CONTROLS	The tag runs a comparison between the operator controls and the reference variables.
	
	Attribute::
	• **type**
	The attribute defines the direction of the data comparison.
	= TRUE – data is read from the reference variables and copied to the operator controls.
	= FALSE – Data is copied from the operator controls to the reference variables.
	
	Syntax:
	`<UPDATE_CONTROLS type = "<Direction>"/>`
	
	Example:
	`<SOFTKEY_OK>`
	`< UPDATE_CONTROLS type="false"/>`
	`</SOFTKEY_OK>`

8.14.1.2 Integrating machining cycles in the program editor (from SW 1.4 SP7)

The system provides the following tags to integrate your own machining cycles into the operating area <PROGRAM> or ALT>+<V>:

- CycleMap
- CYCLE
- NC_INSTRUCTION
- CREATE_CYCLE

Cycle scripts should be saved in the manufacturer's directory **f:\cycles** . The system expects as main module, the file **cycles.xml**. This file has the same structure as the file **xmldial.xml**. In addition, using the **CycleMap** tag, a cycle directory can be created that defines the call of the cycle dialogs for the "recompile" function.

To mark a cycle form, in the **FORM** tag, the attribute **type** should be specified with the value **cycle**. This marking allows the **NC_INSTRUCTION** to be processed.

Example

```
<FORM name = "cycle100_form" type= "CYCLE">
…
…
</FORM>
```

The **NC_INSTRUCTION** tag contains the cycle call to be generated. All cycle parameters should be reserved using space retainers.

Example

```
<FORM name = "cycle100_form" type= "CYCLE">
<NC_INSTRUCTION>Cycle100 ($p1, $p2, $p3)</ NC_INSTRUCTION>
…
…
…
</FORM
```

The **CREATE_CYCLE** tag prepares the values saved in the space retainer variables and generates the NC instruction. If a reference variable is not specified, the NC instruction is copied to the actual cursor position in the part program.

In another case, the function copies the NC instruction into the specified variable.

Generally, the response of the horizontal softkeys 3 to 6 can be overwritten by the softkeys defined in the main menu.

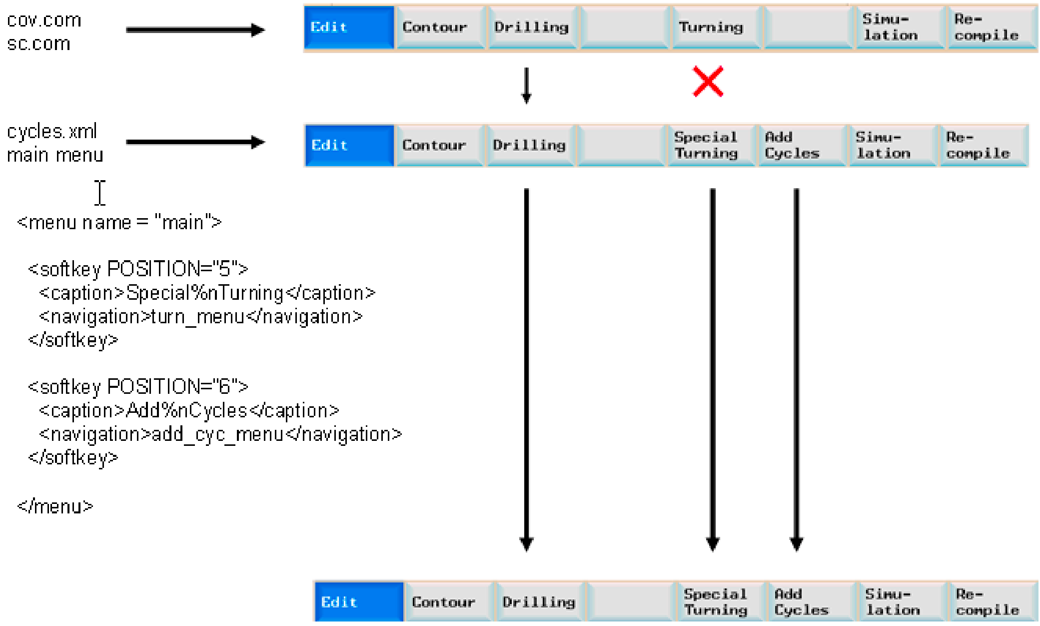


Figure 8-47 Softkey menu in the Program Editor

Tag identifier	Meaning

CycleMap	Creating a cycle directory, which defines the points of entry into the cycle dialogs for the "recompile" function.
	
	Syntax:
	`<CycleMap>`
	`<cycle name="<Cycle name>" >`
	`<navigation>cycle menu</navigation>`
	`</cycle>`
	`</CycleMap>`
	
	Example:
	`<CycleMap>`
	`<cycle name="Cycle100">`
	`<navigation>cycle100_menu</navigation>`
	`</cycle>`
	`<cycle name="Cycle101">`
	`<navigation>cycle101_menu</navigation>`
	`</cycle>`
	`<cycle name="Cycle102">`
	`<navigation>cycle102_menu</navigation>`
	`</cycle>`
	...
	...
	`</CycleMap>`

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>CYCLE</td><td>This tag is used to define the assignment of the cycle to the menu to be called.

Syntax:
<code><cycle name="<cycle name>" >
<navigation> cycle menu </navigation>
</cycle></code>

Attributes:
name
Cycle name
Example:
<code><CycleMap>
<cycle name="Cycle100">
<navigation>cycle100_menu</navigation>
</cycle>
<cycle name="Cycle101">
<navigation>cycle101_menu</navigation>
</cycle>
<cycle name="Cycle102">
<navigation>cycle102_menu</navigation>
</cycle>
…
…
</CycleMap>

<menu name= "cycle100_menu" >
…
…
…</menu></code></td></tr>
</table>

Tag identifier	Meaning

NC_INSTRUCTION	This tag is used to define the NC instruction to be generated.
	All listed cycle parameters are automatically created as string variables of the **FORM** and are available to the **FORM**.
	
	Precondition: The FORMattribute **type** is set to the value **CYCLE**.
	
	Attribute (optional):
	• **refvar**
	If the tag is assigned a reference variable, all parameters are pre-assigned with the values from the NC block saved in the reference variables.
	
	Syntax:
	`<NC_INSTRUCTION> NC instruction with space retainers </ NC_INSTRUCTION>`
	
	Example:
	`<NC_INSTRUCTION>Cycle100($p1, $p2, $p3)</ NC_INSTRUCTION>`
	
	`or`
	
	`<let name="cyc_string" type="string"> Cycle100(0, 1000, 5)</let>`
	…
	…
	…
	`<FORM name = "cycle100_form" type= "CYCLE">`
	`<NC_INSTRUCTION refvar= "cyc_string" >Cycle100($p1, $p2, $p3)</ NC_INSTRUCTION>`
	..
	…
	…
	`</FORM>`

Tag identifier	Meaning
CREATE_CYCLE	The tag generates an NC block, whose syntax is defined by the value of the **NC_INSTRUCTION** tag. If a reference variable is not specified, the instruction inserts the block at the actual cursor position in the part program. **Valid for SW version 1.4 SP6 HF2:** Before generating the NC instruction, the parser calls the CYCLE_CREATE_EVENT tag of the FORM. This tag can be used to calculate the cycle parameters. **Syntax:** `<CREATE_CYCLE/>` **Attribute (optional):** • **refvar** If the tag is assigned a reference variable, the NC instruction is copied to this variable. **Example:**

```
<LET name="cyc_string" type="string"> Cycle100(0, 1000,
5)</LET>

<SOFTKEY_OK>
    <caption>OK</caption>
    <CREATE_CYCLE />
    <close_form />
    <navigation>main_menu</navigation>
</SOFTKEY_OK>

or

<SOFTKEY_OK>
    <caption>OK</caption>
    <CREATE_CYCLE refvar= "cyc_string" />
    <close_form />
    <navigation>main_menu</navigation>
</SOFTKEY_OK>
```

8.14.1.3 Substitution characters

The system offers the option of defining control properties (attribute values) for the runtime. In order to use this function, the desired property must be set in a local variable and the variable name must be transferred to the tag as an attribute value preceded by the character $.

If the tag expects a string as attribute value or value, the $$$ characters must be placed in front of the variable name.

Example:

```
<let name="my_ypos">100</let>
<let name="field_name" type="string"></let>

<control name = "edit1" xpos = "322" ypos = "$my_ypos"
refvar="nck/Channel/Parameter/R[1]" />

<op>my_ypos = my_ypos +20 </op>

<control name = "edit2" xpos = "322" ypos = "$my_ypos"
refvar="nck/Channel/Parameter/R[2]" />

<print name =" field_name" text="edit%d">3</print>
<op>my_ypos = my_ypos +20 </op>

<control name = "$field_name" xpos = "322" ypos = "$my_ypos"
refvar="nck/Channel/Parameter/R3]" />

<caption>$$$field_name</caption>
```

8.14.2 Scope of functions

Overview

The "Generate user dialogs" function offers an open structure and enables the user to develop customer-specific and application-specific HMI interfaces.

The SINUMERIK 802Dsl offers an XML-based script language for generating user dialogs.

This script language makes it possible to display machine-specific menus and dialog forms in the <CUSTOM> operating area on the HMI.

All dialog forms can be designed on a language-neutral basis. In such cases, the system reads out the texts to be displayed from the accompanying language database.

Use

The defined XML instructions offer the following properties:

1. Display dialogs containing the following elements:
   - Softkeys
   - Variables
   - Texts and Help texts
   - Graphics and Help displays
2. Call dialogs by:
   - Pressing the (start) softkeys
3. Restructure dialogs dynamically:
   - Edit and delete softkeys
   - Define and design variable fields
   - Insert, exchange, and delete display texts (language-dependent or language-neutral)
   - Insert, exchange, and delete graphics
4. Initiate operations in response to the following actions:
   - Displaying dialogs
   - Inputting values (variables)
   - Selecting a softkey
   - Exiting dialogs
5. Data exchange between dialogs
6. Variables
   - Read (NC, PLC and user variables)
   - Write (NC, PLC and user variables)
   - Combine with mathematical, comparison or logic operators
7. Execute functions:
   - Subprograms
   - File functions
   - PI services
8. Apply protection levels according to user classes

The valid elements (tags) for the script language are described in the "XML tags" (Page 223) section.

Note

The following section is not intended as a comprehensive description of XML (Extensible Markup Language). Please refer to the relevant specialist literature for additional information.

8.14.3 Fundamentals of Configuration

Configuration files

The defining data for new user interfaces are stored in configuration files. These files are automatically interpreted and the result displayed on the screen. Configuration files are not stored in the software supplied and must first be set up and loaded by the user.

An XML editor or another form of text editor can be used to generate the configuration files.

Note

No distinction is made between upper and lower case letters.

Menu tree principle

Several interlinked dialogs create a menu tree. A link exists if you can switch from one dialog to another. You can use the newly defined horizontal/vertical softkeys in this dialog to call the preceding or any other dialog.

Configured start softkeys can be used to create a further menu tree behind the start menu:

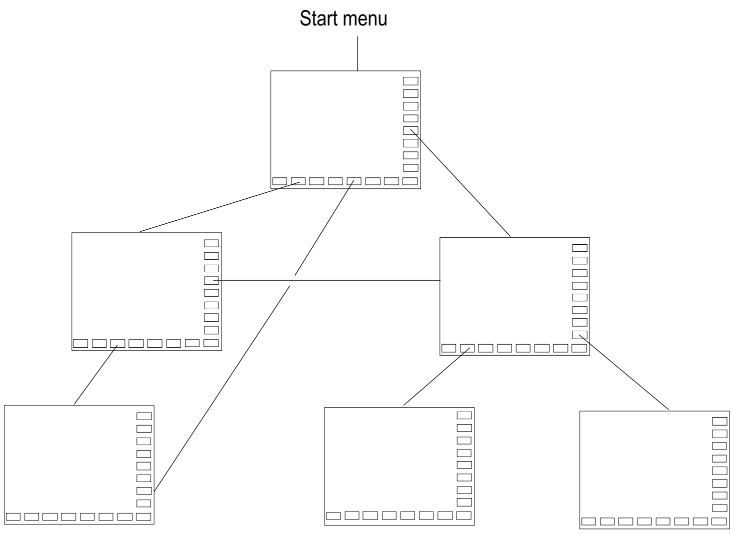


Figure 8-48 Menu tree for user dialogs

Start menu

The start menu is defined by the name "main" in the "xmldial.xml" file. The start menu is used to initiate your own operating sequences.

Loading your own dialogs or additional soft key bars can be linked with the main menu. Additional actions can be performed using these soft key bars.

Returning to the standard application

You can exit the newly created user interfaces and return to the standard application.

8.14.4 Configuration files

Introduction

The screen below shows the manufacturer's drive on the control system.

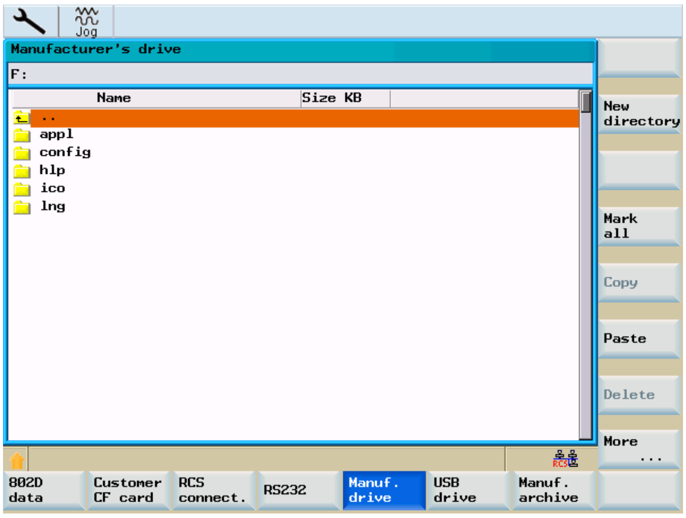


Figure 8-49 Manufacturer's drive

The following files in the control system's "Manufacturer's drive" directory are needed to configure the user dialogs:

Table 8- 32 Files for configuration

File type	Name of the file	Meaning	Storage location in the HMI's <SYSTEM> or <PROGRAM MANAGER> operating area
Script file	"xmldial.xml"	This script file uses XML tags to control the process image of the configured softkey menus and dialog forms in the <COSTUMER> operating area on the HMI.	"Startup files" > "Manufacturer's drive" > in the "appl" subdirectory for the applications
Text file	"aluc.txt"	This text file contains the texts for the menus and dialog forms for the individual languages.	"Startup files" > "Manufacturer's drive" > in the "lng" subdirectory for the languages
Bitmaps	"cus_bmp.arj"	Archive with the bitmaps. The control system supports BMP and PNG formats.	"Startup files" > "Manufacturer's drive" > unpack the "cus_bmp.arj" file in the "ico" subdirectory. Note: If a path to the bitmap file is specified, the files can be stored in this directory directly.
XML files inserted in the "xmldial.xml" control file with the "INCLUDE" XML tag.	E.g. "machine_settings.xml"	These files also contain programmed instructions for displaying the dialog forms and parameters on the HMI.	"Startup files" > "Manufacturer's drive" > in the "appl" subdirectory for the applications

Dependencies of files for configuring user dialogs

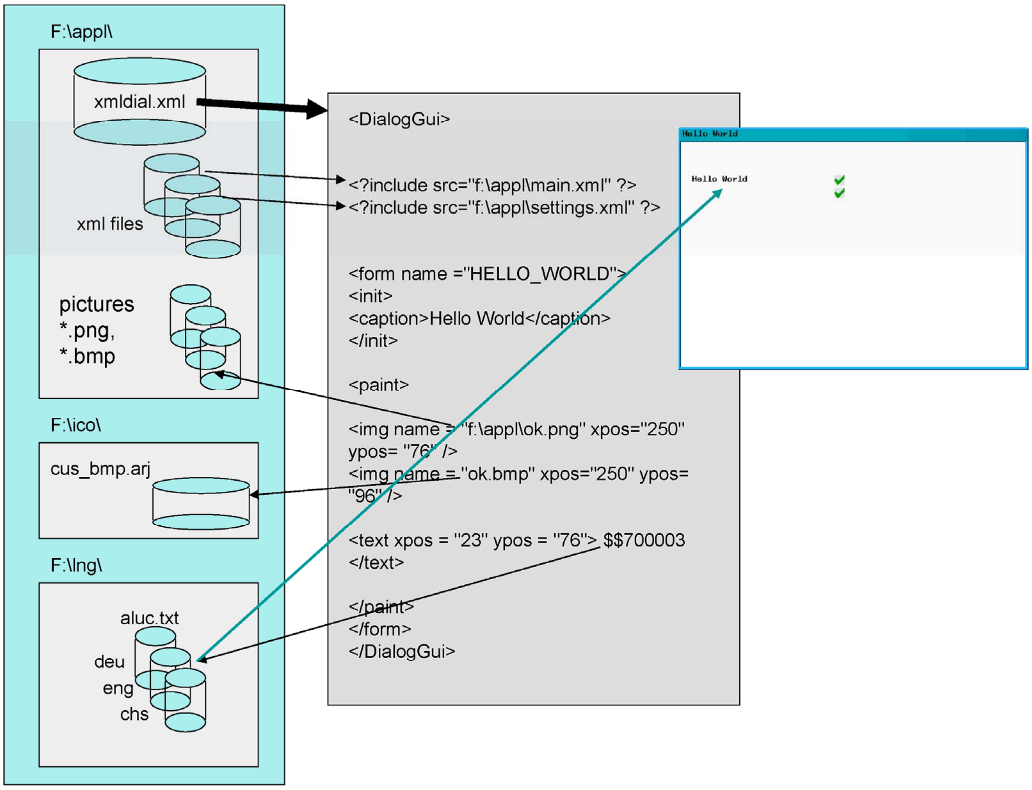


Figure 8-50 Dependencies

Load configuration

As described in the "Storage location in the HMI" column in the previous "Files for configuration" table, the generated files must be copied to the relevant subdirectories in the "Startup files" > "Manufacturer's drive" menu.

Note

As soon as there is an "xmldial.xml" script file in the subdirectory for applications on the manufacturer's drive, the user can start this user dialog in the <CUSTOM> operating area.

After the initial copying process, the control system needs to be reset via "Normal power-up".

Note

The control system will only process the XML scripts if no user application has been entered in the "registry.ini" file.

Example of a user dialog on the HMI

The configured softkey menus are displayed when the <CUSTOM> operating area is called. This enables the user to operate the dialog forms which have been configured.

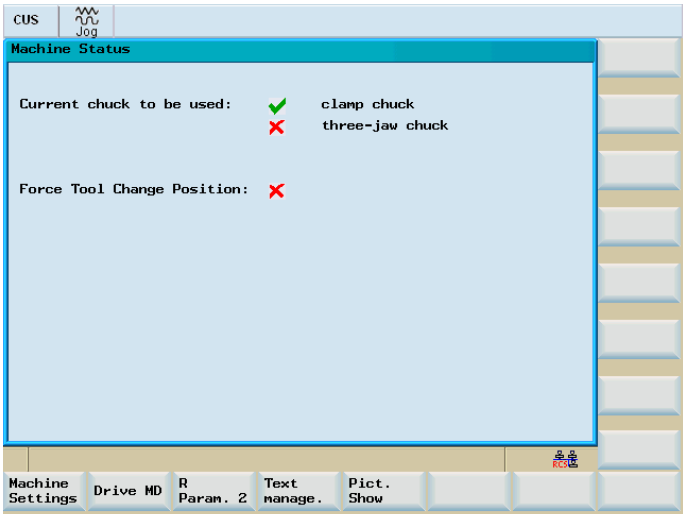


Figure 8-51 Example of a user dialog in the <CUSTOM> operating area

Note

The control system only processes the XML user dialogs if no programmed user dialog boxes are implemented in the control system or entered in the registry.ini file. If necessary, the entry for activating the user dialog (SK7) on the customer hardkey should be deleted.

If configured and programmed dialogs need to be used at the same time, the script language must be used to call the programmed dialogs. The functions required for this purpose are described in Chapter Predefined functions (Page 247).

8.14.5 Structure of configuration file

Overview

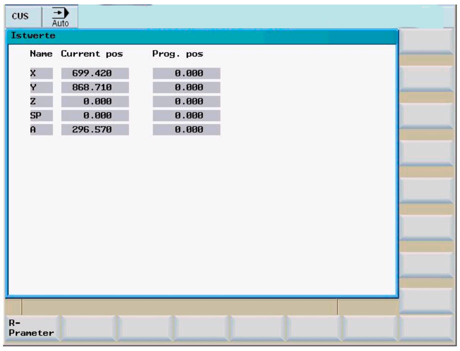
A configuration file consists of the following elements:

- Description of the "main" start menu with start softkeys
- Definition of dialogs
- Definition of variables
- Description of the blocks
- Definition of softkey bars

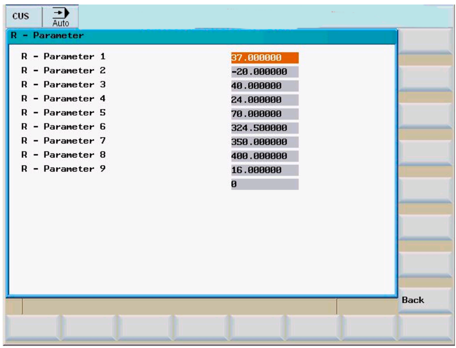
The following screens show an XML script for the "xmldial.xml" file and the corresponding screenshots.

The script contains the dialogs for displaying actual values and residual distances, as well as an R parameter list.

```xml
<DialogGui>

<!--
main menu
It is called by the system software. It starts the application.
The menu tag manages the soft key reactions. One input form can be assigned to a menu tag.
-->

<menu name = "MAIN">
<OPEN_FORM name = "CURRENT_DISPLAY" />

<softkey POSITION="1">
<caption>R-%nPrameter</caption>
<navigation>MENU_R_PARAMETER</navigation> <!-- opens the menu R parameter -->
</softkey>


</menu>


<form name = "CURRENT_DISPLAY">

 <init>

 <caption>Istwerte</caption>

 <control name = "label1" xpos = "36" ypos = "56" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[0]" />
 <control name = "label2" xpos = "36" ypos = "76" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[1]" />
 <control name = "label3" xpos = "36" ypos = "96" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[2]" />
 <control name = "label4" xpos = "36" ypos = "116" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[3]" />
 <control name = "label5" xpos = "36" ypos = "136" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[4]" />


 <control name = "edit1" xpos = "80" ypos = "56" refvar="nck/Channel/GeometricAxis/actProgPos[0]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
 <control name = "edit2" xpos = "80" ypos = "76" refvar="nck/Channel/GeometricAxis/actProgPos[1]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
 <control name = "edit3" xpos = "80" ypos = "96" refvar="nck/Channel/GeometricAxis/actProgPos[2]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
 <control name = "edit4" xpos = "80" ypos = "116" refvar="nck/Channel/GeometricAxis/actProgPos[3]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
 <control name = "edit5" xpos = "80" ypos = "136" refvar="nck/Channel/GeometricAxis/actProgPos[4]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>

 <control name = "edit11" xpos = "210" ypos = "56" refvar="nck/Channel/GeometricAxis/progDistToGo[0]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
 <control name = "edit12" xpos = "210" ypos = "76" refvar="nck/Channel/GeometricAxis/progDistToGo[1]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
 <control name = "edit13" xpos = "210" ypos = "96" refvar="nck/Channel/GeometricAxis/progDistToGo[2]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
 <control name = "edit14" xpos = "210" ypos = "116" refvar="nck/Channel/GeometricAxis/progDistToGo[3]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
 <control name = "edit15" xpos = "210" ypos = "136" refvar="nck/Channel/GeometricAxis/progDistToGo[4]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>

 </init>

 <paint>
 <text xpos= "36" ypos="30">Name</text>

 <text xpos= "80" ypos="30">Current pos</text>
 <text xpos= "210" ypos="30">Prog. pos</text>

 </paint>

</form>

<!-- ****************************************************************
Menu R- Parameter
-->

<menu name ="MENU_R_PARAMETER">
<OPEN_FORM name = "R-Parameter" />

<softkey POSITION="16">
<caption>Back</caption>
<navigation>MAIN</navigation>
</softkey>

</menu>

<form name = "R-Parameter">

 <init>
 <DATA_ACCESS type="true" />
 <caption>R - Parameter</caption>
 <control name = "edit1" xpos = "322" ypos = "34" refvar="nck/Channel/Parameter/R[1]" />
 <control name = "edit2" xpos = "322" ypos = "54" refvar="nck/Channel/Parameter/R[2]" />
 <control name = "edit3" xpos = "322" ypos = "74" refvar="nck/Channel/Parameter/R[3]" />
 <control name = "edit4" xpos = "322" ypos = "94" refvar="nck/Channel/Parameter/R[4]" />
 <control name = "edit5" xpos = "322" ypos = "114" refvar="nck/Channel/Parameter/R[5]" />
 <control name = "edit6" xpos = "322" ypos = "134" refvar="nck/Channel/Parameter/R[6]" />
 <control name = "edit7" xpos = "322" ypos = "154" refvar="nck/Channel/Parameter/R[7]" />
 <control name = "edit8" xpos = "322" ypos = "174" refvar="nck/Channel/Parameter/R[8]" />
 <control name = "edit9" xpos = "322" ypos = "194" refvar="nck/Channel/Parameter/R[9]" />
 <control name = "edit10" xpos = "322" ypos = "214" refvar="nck/Channel/Parameter/R[10]" />
 <control name = "edit10" xpos = "322" ypos = "214" refvar="plc/mb170" />
 </init>

 <paint>
 <text xpos = "23" ypos = "34">R - Parameter 1</text>
 <text xpos = "23" ypos = "54">R - Parameter 2</text>
 <text xpos = "23" ypos = "74">R - Parameter 3</text>
 <text xpos = "23" ypos = "94">R - Parameter 4</text>
 <text xpos = "23" ypos = "114">R - Parameter 5</text>
 <text xpos = "23" ypos = "134">R - Parameter 6</text>
 <text xpos = "23" ypos = "154">R - Parameter 7</text>
 <text xpos = "23" ypos = "174">R - Parameter 8</text>
 <text xpos = "23" ypos = "194">R - Parameter 9</text>
 </paint>
</form>

</DialogGui>
```



Actual values



R parameters

Figure 8-52 XML script and dialog forms with parameters

8.14.6 Language dependency

Language-dependent texts are used for:

- Softkey labels
- Headers
- Help texts
- Any other texts

The language-dependent texts are stored in text files.

Note

You will need to perform the following steps when using these text files:

- Make them available in the required languages.
- Use the RCS802 tool to convert them into the control system's internal format.
- Transfer them into the relevant language directories of the control system.

See also

Language setting and file management (Page 114)

8.14.7 XML identifier

8.14.7.1 General structure

Structure and instructions of the script file for dialog configuration

All dialog configurations should be stored in the **DialogGui** tag.

```
<DialogGui>
…
</DialogGui>
```

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<DialogGui>
…
<FORM name ="Hello_World">
<INIT>
<CAPTION>Hello World</CAPTION>
</INIT>
...
</FORM>

</DialogGui>
```

Instructions

The language offers the following instructions for executing conditional instructions and loop controls:

- For loop
- While loop
- Do with loop
- Conditional processing
- Switch and case instructions
- Operator controls in a dialog form
- Softkey descriptions
- Define variables

For a detailed description of instructions, see Instruction/identifier descriptions (Page 224).

8.14.7.2 Instruction/identifier descriptions

The following XML tags are defined for generating dialogs and menus, and for executing program sequences:

Note

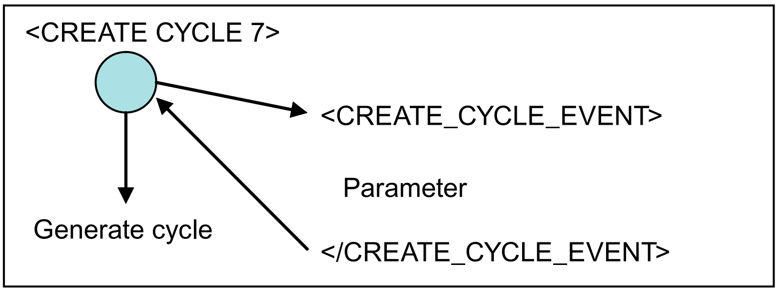
Attribute values that are in quotation marks "<...>" should be replaced by the currently used expressions.

Example:
<DATA_LIST action="read/write/append" id="<list name>">
is programmed as follows:

<DATA_LIST action="read/write/append" id="my datalist">

Tag identifier	Meaning
BREAK	Conditional cancellation of a loop.
CONTROL_RESET	The tag enables one or more control components to be restarted. **Syntax:** `<CONTROL_RESET resetnc="TRUE" />` **Attributes:** • RESETNC = "TRUE" The NC component is restarted • RESETDRIVE = "TRUE" The drive components are restarted.

Tag identifier	Meaning

CREATE_CYCLE_EVENT
From SW 1.4 SP7
	If the parser starts to process tags CREATE_CYCLE, initially, the message <CREATE_CYCLE_EVENT sent> is sent to the active form. This message can be used for preparing the cycle parameters, before the parser generates the NC operation from the parameter list and the generation rule.
	
	Syntax:
	`<CREATE_CYCLE_EVENT>`
	…
	…
	`</CREATE_CYCLE_EVENT>`
	
	Example:
	`<SOFTKEY_OK>`
	…
	`<CREATE_CYCLE />`
	…
	…
	`<SOFTKEY_OK>`
	…
	…
	
	`<FORM>`
	`<NC_INSTRUCTION>MY_CYCLE($P1, $P2)</ NC_INSTRUCTION >`
	…
	…
	`<CREATE_CYCLE_EVENT>`
	`<type_cast name="P1" type="int"/>`
	
	`<OP> P1 = P1 * 150 </OP>`
	
	`</CREATE_CYCLE_EVENT>`
	…
	…
	
	`</FORM>`

Tag identifier	Meaning

DATA	The tag enables the NC, PLC, GUD and drive data to be directly written to.
	The "Component addressing" (Page 242) section contains details on address formation.
	
	Attribute:
	• **name**
	Variable address
	
	Tag value:
	All alphanumeric terms are approved as tag values. If a value is to be written from a local variable directly, the **$** replacement operator preceding the name of the local variable should be used.
	
	Syntax:
	`<DATA name="<variable name>"> value </DATA>`
	
	Example:
	`<DATA name = "plc/mb170"> 1 </DATA>`
	…
	`<LET name = "tempVar"> 7 </LET>`
	`<!— the contents of the local variables "tempVar" are written to bit memory byte 170 →`
	`<DATA name = "plc/mb170">$tempVar</DATA>`

Tag identifier	Meaning
DATA_LIST	The tag enables the listed drive and machine data to be saved or restored. Addresses are listed in lines. The "Component addressing" (Page 242) section contains details on address formation. Up to 20 temporary data lists can be created. Attributes: • action *read*– the values of the listed variables are stored in a temporary memory *append*– the values of the listed variables are added to an existing list *write*– the backed up values are copied to the relevant machine data • id The identifier is used to identify the temporary memory Syntax: <DATA_LIST action="<read/write/append>" id="<list name>"> NC/PLC Address compilation </DATA_LIST> Example: <DATA_LIST action ="read" id="<name>"> nck/channel/parameter/r[2] nck/channel/parameter/r[3] nck/channel/parameter/r[4] $MN_USER_DATA_INT[0] … </ DATA_LIST> <DATA_LIST action ="write" id="<name>" />

Tag identifier	Meaning

DYNAMIC_INCLUDE
From SW 1.4 SP7
	The tag includes an XML script file.
	Contrary to the INCLUDE tag, read-in is first only realized when executing the corresponding operation.
	For large projects, the use of the tag reduces the load time of the customer area and/or the cycle support. Further, the average level of resources is reduced, as not all of the dialogs are always called during a session.
	
	Syntax:
	<DYNAMIC_INCLUDE src="path name"/>
	
	Example:
	`<SOFTKEY POSITION="3">`
	   `<CAPTION>MY_MENU</CAPTION>`
	   `<DYNAMIC_INCLUDE src="f:\cycles\my_submenu.xml"/>`
	   `<NAVIGATION>MY_MENU</NAVIGATION>`
	</SOFTKEY>

ELSE	Instruction for situations where the condition has not been met (IF, THEN, ELSE)
FORM	The tag contains the description of a user dialog. The relevant tags are described in the section on generating menus and dialog forms. **Syntax:** `<FORM name="<dialog name>" color="#ff0000">` **Attributes:** • color Background color of the dialog form (color coding, see Chapter Color coding (Page 241)) – Default white • name Identifier of the form • type Permissible value is *cycle*, which identifies a user cycle screen form • xpos X-position of the top left corner of the dialog box (optional) • ypos Y position of the top left corner (optional) • width Extension in the X direction (in pixels) (optional) • height Extension in the Y direction (in pixels) (optional)
HMI_RESET	The tag initiates an HMI restart. The interpretation is cancelled after this operation.

Tag identifier	Meaning

IF	Conditional statement (**IF, THEN, ELSE**)
	The THEN and ELSE tags are enclosed in the IF tag.
	The condition that is executed in the CONDITION tag follows the IF tag. The further processing of the instructions depends upon the result of the operation. If the function result is true, then the THEN branch is executed and the ELSE branch is skipped. If the result of the function is false, the parser executes the ELSE branch.
	
	Syntax:
	`<IF>`
	`<CONDITION> Condition != 7 </CONDITION>`
	`<THEN>`
	`Instruction for the case: Condition fulfilled`
	`</THEN>`
	`<ELSE>`
	`Instruction for the case: Condition not fulfilled`
	`</ELSE>`
	`</IF>`
	
	Example:
	`<IF>`
	`<CONDITION> "plc/mb170" != 7 </CONDITION>`
	`<THEN>`
	`<OP> "plc/mb170" = 7 </OP>`
	…
	`</THEN>`
	`<ELSE>`
	`<OP> "plc/mb170" = 2 </OP>`
	…
	`</ELSE>`
	`</IF>`

INCLUDE	The instruction includes an XML description. (see also DYNAMIC_INCLUDE in this table) **Attribute:** • **src** Contains the path name. **Syntax:** `<?INCLUDE src="<Path name>" ?>`

<table>
<tr><th>Tag identifier</th><th>Meaning</th></tr>
<tr><td>LET</td><td>
The instruction creates a local variable under the specified name.
Fields:
Using the attribute dim (dimension) single or two-dimensional fields can be created. The field index is used to address the individual field elements.
For a two-dimensional field, initially the line index is specified and then the column index.
Single-dimensional field:
Indices 0 to 4
<table>
<tr><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
</table>
Two-dimensional field:
Index line 0 to 3 and index column 0 to 5
<table>
<tr><td>0,0</td><td>0,1</td><td>0,2</td><td>0,3</td><td>0,4</td><td>0,5</td></tr>
<tr><td>1,0</td><td>1,1</td><td>1,2</td><td>1,3</td><td>1,4</td><td>1,5</td></tr>
<tr><td>2,0</td><td>2,1</td><td>2,2</td><td>2,3</td><td>2,4</td><td>2,5</td></tr>
<tr><td>3,0</td><td>3,1</td><td>3,2</td><td>3,3</td><td>3,4</td><td>3,5</td></tr>
</table>
Attributes:

name
Variable name
type
The variable type can be an integer (INT), double (DOUBLE), float (FLOAT) or string (STRING). If there is no type instruction specified, the system creates an integer variable.

<code><LET name = "VAR1" type = "INT" /></code>

permanent
If the attribute is set to true, then the variable value is saved permanently. This attribute is only effective for a global variable.
dim
The following number of field elements must be specified. For a two-dimensional field, the second dimension is specified after the first dimension separated by a comma.
A field element is accessed via the field index, which is specified in square brackets after the variable name.
name[index] or name[row,column]

Single-dimensional field: dim="<Number of elements>"
Two-dimensional field: dim="<Number of lines>,<number of columns>"
Non-initialized field elements are pre-assigned with "0".

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

Tag identifier	Meaning

LET Continued	**Example:**
	Single-dimensional field:
	`<let name="array" dim="10"></let>`
	Two-dimensional field:
	`<let name="list_string" dim="10,3" type="string"></let>`
	
	Pre-assignment:
	A variable can be initialized with a value.
	`<LET name = "VAR1" type = "INT"> 10 </LET>`
	
	If values comprising NC or PLC variables are saved in a local variable, the assignment operation automatically adapts the format to that of the variables which have been loaded.
	• Pre-assignment for a string variable:
	Texts containing more than one line can be assigned to a string variable if the formatted text is transferred as a value. If a line is to end with a line feed <LF> , the characters "\\n" should be added at the end of the line.
	
	`<LET name = "text" type = "string"> F4000 G94\\n`
	`G1 X20\\n`
	`Z50\\n`
	`M2\\n`
	`</LET>>`
	
	`Fields (Arrays):`
	`<let name="list" dim="10,3">`
	`  {1,2,3},`
	`  {1,20}`
	`</let>`
	
	`<let name="list_string" dim="10,3" type="string">`
	`  {"text 10","text 11"},`
	`  {"text 20","text 21"}`
	`</let>`
	
	Assignment:
	Values made up of the machine data or subroutines can be assigned to a variable using the assignment operation "=".
	A variable remains valid until the end of the higher-level XML block.
	Variables which are to be available globally should be created directly after the **DialogGUi** tag.
	The following must be observed for a dialog box:
	• The message processing opens the corresponding tag.
	• The tag is closed after the message has been executed.
	• All variables within the tag are deleted when closing.

Tag identifier	Meaning
MSG	The operator component shows the message which is indicated in the tag. If an alarm number is used, the dialog box displays the text which is saved for the number. **Example:** <MSG text ="my message" />
MSGBOX	The instruction opens a message box whose return value can be used for branching. **Syntax:** <MSGBOX text="<Message>" caption="<caption>" retvalue="<variable name>" type="<button type>" /> **Attributes:** • **text** Text • **caption** Header • **retvalue** Name of the variables to which the return value is copied: 1 – OK 0 – CANCEL • **type** Acknowledgement options: "BTN_OK" "BTN_CANCEL" "BTN_OKCANCEL" If an alarm number is used for the **"text"** or **"caption"** attribute, the message box displays the text which is saved for the number. **Example:** <MSGBOX text="Test message" caption="Information" retvalue="result" type="BTN_OK" />

Tag identifier	Meaning
OP	The tag executes the specified operations. The operations listed in Chapter "Operators (Page 241) can be executed. For the purpose of accessing the NC, PLC, and drive data, the complete variable name should be placed in **quotation marks**. The "Component addressing" (Page 242) section contains details on address formation. PLC: "PLC/MB170" NC: "NC/Channel/…" **Example:** <LET name = "tmpVar" type="INT"> </LET> <OP> tmpVar = "plc/mb170" </OP> <OP> tmpVar = tmpVar *2 </OP> <OP> "plc/mb170" = tmpVar </OP> **Character string processing:** The operation instruction is able to process character strings and assign the results to the string variable specified in the equation. The identifier _T should be placed at the start as a means of identifying text terms. Formatting of variable values is also possible. The identifier _F should be placed at the start of the formatting regulation, followed by the format instruction. The address is then specified for the variable. **Example:** <LET name="buffer" type="string"></LET> … … … <OP> buffer = _T"unformatted value R0= " + "nck/Channel/Parameter/R[0]" + _T" and " + _T"$$85051" + _T" formatted value R1 " + _F%9.3f"nck/Channel/Parameter/R[1]" </OP>
PASSWORD	The tag opens a dialog for entering the password. Once the entry has been confirmed, the character string is available in the specified reference variable. **Syntax:** <PASSWORD refVar ="<variable name>" /> **Attribute:** • refVar Name of the reference variable **Example:** <PASSWORD refvar="plc/mw107" />
POWER_OFF	A message prompts the operator to switch the machine off. The message text is permanently saved in the system.

Tag identifier	Meaning

PRINT	The tag outputs a text in the dialog line or copies the text to the variable specified.
	If the text contains formatting identifiers, the variable values are inserted at the appropriate places.
	
	Syntax:
	`<PRINT name="Variable name " text="text %Formatting "> Variable, … </PRINT>`
	`<PRINT text="text %Formatting"> Variable, … </PRINT>`
	Attributes:
	• **name**
	Name of the variable where the text is to be stored (optional)
	• **text**
	Text
	
	Formatting:
	The character "%" causes the variable specified as the value to be formatted.
	%[Flags] [Width] [.decimal places] type
	• Flags:
	Optional character for defining output formatting:
	– Right-justified or left-justified ("-") for left-justified
	– Add leading zeros ("0")
	– Fill with blanks
	• Width:
	The argument defines the minimum output width for a non-negative number. If the value to be output has fewer places than the argument defined, the missing spaces are filled with blanks.
	• Decimal places:
	With floating point numbers, the optional parameter defines the number of decimal places.
	• Type:
	The type character defines which data formats are transferred for the print instruction. These characters need to be specified.
	– d: Integer value
	– f: Floating point number
	– s: String

Tag identifier	Meaning
PRINTContinued	Values: Number of variables whose values are to be inserted into the text. The variable types must match the corresponding type identifier for the formatting instruction and must be separated from one another using a comma. Example: Output of a text in the information line <PRINT text="Infotext" /> Output of a text with variable formatting <LET name="trun_dir"></LET> <PRINT text="M%d">trun_dir</PRINT> Output of a text in a string variable with variable formatting <LET name="trun_dir"></LET> <LET name="str" type="string" ></LET> <print name="str" text="M%d ">trun_dir</print>
PROGRESS_BAR From SW 1.4 SP7	The tag opens or closes a progress bar. The bar is displayed below the application window. Syntax: <PROGRESS_BAR type="<true/false>"> value </ PROGRESS_BAR> Attributes: • type = "TRUE" - opens the progress bar • type= "FALSE" - closes the progress bar • min (optional) – minimum value • max (optional) – maximum value Value: • Value Percentage position of the bar Example: <PROGRESS_BAR type="true" min="0" max="101" >20< /PROGRESS_BAR>......<PROGRESS_BAR >50< /PROGRESS_BAR>......<PROGRESS_BAR type="false" >100< /PROGRESS_BAR>

Tag identifier	Meaning
SEND_MESSAGE From SW 1.4 SP7	The tag sends a message with two parameters to the active form, which is processed in the tag message. **Syntax:** <SEND_MESSAGE>p1, p2</SEND_MESSAGE> **Example:** <SOFTKEY POSITION="3"> <caption>Set%nParameter</caption> <send_message>1, 0</send_message> </SOFTKEY> <FORM> … … <MESSAGE> <SWITCH> <CONDITION>$message_par1</CONDITION> <CASE value="1"> … … </CASE> </SWITCH> </MESSAGE> … … </FORM>
SLEEP From SW 1.4 SP7	The tag interrupts script execution for the specified period. The interruption time is obtained from the transferred value multiplied by the time base of 50 ms. **Syntax:** <SLEEP value="Interruption time" /> **Example:** Wait time, 1.5 sec. <SLEEP value="30" />
STOP	Interpretation is cancelled at this point.

Tag identifier	Meaning
SWITCH	The **SWITCH** instruction describes a multiple choice. A term is evaluated once and compared with a number of constants. If the expression matches the constants, the instructions are executed within the **CASE** instruction. The **DEFAULT** instruction is executed when none of the constants match the expression. **Syntax:** <SWITCH> <CONDITION> Value </CONDITION> <CASE value="Constant 1"> Instructions … </CASE> <CASE value="Constant 2"> Instructions … </CASE> <DEFAULT> Instructions … </DEFAULT> </SWITCH>
SHOW_CONTROL From SW 1.4 SP7	The visibility of a control can be controlled using the tag. **Syntax:** <SHOW_CONTROL name="<name>" type="<type>" /> **Attributes:** • **name** Name of the control • **type** = "TRUE" - control becomes visible • **type**= "FALSE" - control becomes invisible (hidden) **Example:** <SHOW_CONTROL name="myEditfield" type="false" /> … … <SHOW_CONTROL name="myEditfield" type="true" />

Tag identifier	Meaning
TYPE_CAST From SW 1.4 SP7	The tag is used to convert the data type of a local variable. **Syntax:** <type_cast name="variable name" type=" new type" /> **Attributes:** • **name** Variable name • **type** new data type
THEN	Operation, if the condition has been fulfilled (**IF, THEN, ELSE**)

FOR	For loop
	`for (initialization; test; continuation) instruction(s)`
	
	Syntax:
	`<FOR>`
	`<INIT>…</INIT>`
	`<CONDITION>…</CONDITION>`
	`<INCREMENT>…</INCREMENT>`
	`Instructions`
	`…`
	`</FOR>`
	The For loop is executed as follows:
	1. Evaluation of the term **initialization** (INIT).
	2. Evaluation of the term **test** (CONDITION) as a Boolean term.
	If the value is false, the For loop is ended.
	3. Execution of the following instructions.
	4. Evaluation of the term **continuation** (INCREMENT)..
	5. Continue with 2.
	All the variables used within the INIT, CONDITION, and INCREMENT branches should be created outside the FOR loop.
	
	Example:
	`<LET name = "count">0</LET>`
	`<FOR>`
	`   <INIT>`
	`   <OP> count = 0</OP>`
	`   </INIT>`
	`   <CONDITION> count <= 7 </CONDITION>`
	`   <INCREMENT>`
	`   <OP> count = count + 1 </OP>`
	`   </INCREMENT>`
	`   <OP> "plc/qb10" = 1+ count </OP>`
	`</FOR>`

Tag identifier	Meaning

WAITING	The tag waits for the component to undergo a hot restart after an NC or drive reset.
	
	Attributes:
	• WAITINGFORNC = "TRUE" - the system waits for the NC to restart
	• WAITINGFORDRIVE = "TRUE" - the system waits for the drives to restart
	
	Syntax:
	`<WAITING WAITINGFORNC = "TRUE"/>`
	
	Example:
	...
	`<CONTROL_RESET resetnc = "true" resetdrive = "true"/>`
	`<WAITING waitingfornc = "true" waitingfordrive = "true" />`
	...

Tag identifier	Meaning

WHILE	WHILE loop
	`WHILE (Test)`
	`   Instruction`
	
	Syntax:
	<WHILE>
	<CONDITION>…</CONDITION>
	Instructions
	…
	</WHILE>
	The While loop is used to execute a sequence of instructions repeatedly while a condition is met. This condition is tested before the sequence of instructions is executed.
	
	Example:
	<WHILE>
	<CONDITION> "plc/ib9" == 0 </CONDITION>
	<DATA name = "PLC/qb11"> 15 </DATA>
	</WHILE>

DO_WHILE	Do while loop
	`DO`
	`   Instructions`
	`WHILE (Test)`
	
	Syntax:
	<DO_WHILE>
	Instructions
	…
	<CONDITION>…</CONDITION>
	</DO_WHILE>
	The Do while loop comprises a block of instructions and a condition. The code within the instruction block is executed first, then the condition is analyzed. If the condition is true, the function executes the code section again. This is continuously repeated until the condition is false.
	
	Example:
	<DO_WHILE>
	<DATA name = "PLC/qb11"> 15 </DATA>
	<CONDITIION> "plc/ib9" == 0 </CONDITION>
	</DO_WHILE>

8.14.7.3 Color coding

The color attribute uses the color coding scheme for the HTML language.

In terms of syntax, color specifications consist of the "#" (hash) character and six digits from the hexadecimal system, with each color represented by two digits.

R – Red

G – Green

B – Blue

#RRGGBB

Example:

color = "#ff0011"

8.14.7.4 Special XML syntax

Characters with special meanings in XML syntax have to be rewritten if they are to be displayed correctly by a general XML editor.

The following characters are affected:

Character	Notation in XML
<	<
>	>
&	&
"	"
'	'

8.14.7.5 Operators

The operation instruction processes the following operators:

Operator	Meaning
=	Assignment
==	Equal to
<, <	Less than
>, >	Greater than
<=, <=	Less than or equal to
>=, >=	Greater than or equal to
|	OR operation in bits
||	Logic OR operation
&, &	AND operation in bits
&&, &&	Logic AND operation
+	Addition
-	Subtraction

Operator	Meaning
*	Multiplication
/	Division
!	Not
!=	Not equal to

Operation instructions are processed from left to right. It may make sense to place terms in parentheses under certain circumstances in order to define the priority for executing subterms.

8.14.8 Addressing components

Address identifiers for the desired data must be created to address NC variables, PLC blocks or drive data. An address consists of the subpaths **component name** and **variable address**. A slash should be used as a separating character.

8.14.8.1 PLC addressing

Addressing the PLC starts with the path section **plc**.

Table 8- 33 The following addresses are permissible:

DBx.DB(f)	Data block
I(f)x	Input
Q(f)x	Output
M(f)x	Bit memory
V(f)x	Variable

DBx.DBXx.b	Data block
Ix.b	Input
Qx.b	Output
Mx.b	Bit memory
Vx.b	Variable

Table 8- 34 Data format **f**:

B	Byte
W	Word
D	Double word

Data format identification is not applicable to bit addressing.

Address **x**:

Valid S7-200 address identifier

Bit addressing:

b – Bit number

Examples:

```
<data name = "plc/mb170">1</data>
<data name = "i0.1"> 1 </data>
<op> "m19.2" = 1 </op>
```

8.14.8.2 Addressing NC variables

Addressing the NC variables starts with the path section **nck**.

This section is followed by the data address; its structure should be taken from the SINUMERIK 802D sl Parameter Manual.

Example:

```
<LET name = "tempStatus"></LET>
<OP> tempStatus ="nck/channel/state/chanstatus" </OP>
```

8.14.8.3 Generating NC/PLC addresses during the runtime

From SW 1.4 SP7, there is the option of generating an address identifier during the runtime.

In this case, the content of a string variable is used as address in an operation statement as well as in the nc.cap.read and nc.cap.write functions.

Observe the following for this type of addressing mode:

- Write the variable names in quotation marks.
- Use three ‚$' characters as prefix for variable names.

Syntax:

```
"$$$variable name"
```

Example:

```
<LET name="var_adr" type="string"></LET>
<PRINT name="var_adr" text="VB9000%d"> 2000</PRINT>
<OP> "$$$var_adr" = 1 </OP>
```

8.14.8.4 Addressing drive components

Addressing the drive components starts with the path section **drive**.

Then the drive device is specified:

CU

DC

The parameter to be set is added to this section.

Example:

```
<LET name="r0002_content"></LET>
<LET name="p107_content"></LET>

<!— Reading of value r0002 on the CU ->

<OP> r0002_content = "drive/cu/r0002" </OP>
<OP> r0002_content = "drive/cu/r0002[CU1]" </OP>

<!— Reading of value r0002 on NX1 ->
<OP> r0002_content = "drive/cu/r0002[CU2]" </OP>

<!— Reading of value p107[0] on the CU ->
<OP> p107_content = "drive/cu/p107[0]" </OP>
<PRINT text="%d"> p107_content </PRINT>

<!— Reading of value p107[0] on the CU ->

<OP> p107_content = "drive/cu/p107[0, CU1]" </OP>
<PRINT text="%d"> p107_content </PRINT>

<!— Reading of value p107[0] on the NX1 ->

<OP> p107_content = "drive/cu/p107[0, CU2]" </OP>
<PRINT text="%d"> p107_content </PRINT>
```

Addressing the drive objects:

To address individual objects, the desired object should be entered in square brackets after the parameter.

Note

The drive object number differs from the numbering used in the drive dialog, since the CU components, ALM, and all connected hubs are integrated with the continuous numbering.

The DO number can be established as follows:

- All connected drive objects are listed in field p978 of the relevant CU corresponding to their slot number.
- It is a question of establishing the field index for the desired slot and adding one to this number. This value is the DO index needed for addressing purposes.

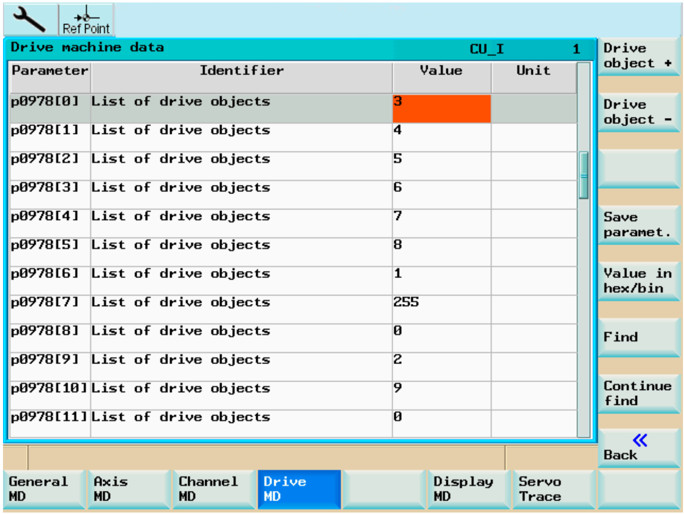


Figure 8-53 Drive parameter p0978 [...] at the control

```
Parameter number[do<DO-index>]
```

Example:

```
p0092[do1]
```

Alternatively, the drive index can be read from a local variable using **$<variable name>** "substitution characters".

```
z.B. DO$local variable
```

Example:

```
<DATA name ="drive/cu/p0092">1</DATA>
<DATA name ="drive/dc/p0092[do1] ">1</DATA>
```

Indirect addressing:

```
<LET name = "driveIndex> 0 </LET>
<OP> driveIndex = $ctrlout_module_nr[0, AX1] </OP>
<DATA name ="drive/dc[do$driveIndex]/p0092">1</DATA>
```

8.14.8.5 Addressing machine and setting data

Drive and setting data is identified by the character $ followed by the name of the data.

Machine data:

$Mx_<name[**index, AX<axis_number>**]>

Setting data:

$Sx_<name[index, AX<axis_number>]>

x:

N – General machine or setting data

C – Channel-specific machine or setting data

A – Axis-specific machine or setting data

Index:

For a field, the parameter indicates the index of the data.

AX<axis_number>:

The required axis (**<axis_number>**) has to be specified for axis-specific data.

Alternatively, the axis index can be read from a local variable using **$<variable name>** "substitution characters".

e.g. AX$localvariable

Example:
```
<DATA name ="$MN_AXCONF_MACHAX_NAME_TAB[0] ">X1</DATA>
```

Direct addressing of the axis:
```
<DATA name ="$MA_CTRLOUT_MODULE_NR[0, AX1] ">1</DATA>
…
…
```

Indirect addressing of the axis:
```
<LET name ="axisIndex"> 1 </LET>
<DATA name ="$MA_CTRLOUT_MODULE_NR[0, AX$axisIndex] ">1</DATA>
```

8.14.8.6 Addressing user data

Addressing user data starts with the path section **gud**, followed by the GUD name.

For a field, after the name, the required field index should be specified in square brackets.

Example:
```
<DATA name ="gud/syg_rm[0]"
<OP>"gud/syg_rm[0]" 0 10 </op>
```

8.14.9 Predefined functions

The script language offers various string processing and standard mathematical functions.

The function names listed below are reserved and cannot be overloaded.

Function name	Meaning

Ncfunc cap read
From SW 1.4 SP7
	The function copies a value from the specified address into a local variable. If the read operation was error-free, then the return variable contains the value zero.
	
	Contrary to the operation instruction, in the event of a fault, this function does not interrupt the processing of the script operations.
	
	Syntax:
	`<function name="ncfunc.cap.read" return="error">`
	`lokale variable, "address"</function>`
	
	Example:
	`<let name="error"></let>`
	`<function name="ncfunc.cap.read" return="error"> 3,`
	`"drive/cu/p0009"</function>`
	`<if>`
	`<condition>error != 0</condition>`
	`<then>`
	`<break />`
	`</then>`
	`</if>`

<table>
<tr><th>Function name</th><th>Meaning</th></tr>
<tr><td>Ncfunc cap write
From SW 1.4 SP7</td><td>The function writes a value into the specified variable. If the write operation was error-free, then the return variable contains the value zero.

Contrary to the operation instruction, in the event of a fault, this function does not interrupt the processing of the script operations.

Syntax:
<code><function name="ncfunc.cap.read" return="error">
local variable or constant, "address"</function></code>

Example:
<code><let name="error"></let>
<function name="ncfunc.cap.write" return="error">
0, "drive/cu/p0009"</function>
<if>
<condition>error != 0</condition>
<then>
<break />
</then>
</if></code></td></tr>
</table>

Function name	Meaning

Ncfunc PI-Service
From SW 1.4 SP7
	Jobs can be transferred to the NCK using the program invocation (PI) service.
	If the service has been executed error-free, the function returns the value 1 in the return variable.
	
	Manipulation of the tool list
	
	_N_CREATO - Create tool
	_N_DELETO - Delete tool
	_N_CREACE - Create tool cutting edge
	_N_DELECE - Delete tool cutting edge
	
	Activation of work offsets
	
	_N_SETUFR - Activates the actual user frame
	_N_SETUDT - Activates the actual user data
	
	Block search
	
	_N_FINDBL - Activate block search
	_N_FINDAB - Cancel block search
	
	Syntax:
	`<function name="ncfunc.pi_service" return="return var"> pi name, var1, var2, var3, var4, var5 </function>`
	
	Attributes:
	• name - Function name
	• return- Name of the variable in which the execution result is saved
	  – Value == 1 – job executed successfully
	  – Value == 0 – faulty job
	
	Tag values:
	• pi name - Name of the PI service (string)
	• var1 tovar5 - PI specific arguments

Function name	Meaning
Ncfunc PI-Service Continued From SW 1.4 SP7	**Arguments:** • _N_CREATO var1 - Tool number • _N_DELETO var1 - Tool number • _N_CREACE var1 - Tool number var2 - Cutting edge number • _N_DELECE var1 - Tool number var2 - Cutting edge number • _N_SETUFR No arguments • _N_SETUDT var1- User data area to be activated – 1 - Tool offset data – 2 - Active basic frame – 3 - Active adjustable frame • _N_FINDBL var1 - Search mode – 2 - Search with contour calculation – 4 - Search for the block end point – 1 - Block search without calculation. • _N_FINDAB No arguments **Example:** • Creating a tool – tool number 3 `<function name="ncfunc.pi_service">_T"_N_CREATO", 3</function>` • Delete cutting edge 1 of tool 5 <function name="ncfunc.pi_service">_T"_N_DELECE", 5, 1</function>

Function name	Meaning
Ncfunc display resolution From SW 1.4 SP7	This function supplies the conversion rule for floating point numbers defined in the control. A string variable must be provided as variable. See also display machine data MD203 DISPLAY_RESOLUTION and MD204 DISPLAY_RESOLUTION_INCH **Syntax:** <function name="ncfunc.displayresolution" return="dislay_res" /> **Example:** <let name="dislay_res" type="string"></let> … <function name="ncfunc.displayresolution" return="dislay_res" /> <control name = "cdistToGo" xpos = "210" ypos = "156" refvar="nck/Channel/GeometricAxis/progDistToGo[2]" hotlink="true" height="34" fieldtype="readonly" format="$$$dislay_res" time="superfast" color_bk="#ffffff"/>
Ncfunc bico to int From SW 1.4 SP7	The function converts a string specified in the BICO format into an integer value. (see SINAMICS). **Syntax:** <function name="ncfunc.bicotoint" return="integer variable">bico-string</function> **Example:** <let name="s_np0480_0" type="string"></let> <let name="i_p0480_0">0</let> <function name="ncfunc.bicotoint" return="i_p0480_0">s_np0480_0</function>

Function name	Meaning
Ncfunc int to bico From SW 1.4 SP7	The function converts an integer value into a BICO format string. (see SINAMICS). **Syntax:** <function name="ncfunc.inttobico" return="string variable">integer variable</function> **Example:** <function name="ncfunc.inttobico" return="s_p0480_0">"drive/dc/p0480[0, DO2]"</function>
Ncfunc is bico str valid From SW 1.4 SP7	This function returns the value zero if it involves a string specified in the BICO format. (see SINAMICS) **Syntax:** <function name="ncfunc.isbicostrvalid" return="integer variable">string varaible</function> **Example:** … <let name="s_np0480_0" type="string"></let> … … <control name = "cp0480_0" xpos = "402" ypos = "76" hotlink="true" refvar="s_np0480_0" > <property item_data="4001" /> </control> … … <function name="ncfunc.isbicostrvalid" return="valid">cp0480_0</function>

Function name	Meaning

Ncfunc password From SW 1.4 SP7	This function sets or deletes a password level.
	• Set password:
	The password should be specified for the required password level as parameter.
	• Delete password:
	A blank string deletes the password level.
	
	Syntax:
	`<function name="ncfunc.password">password`
	`</function>`
	
	Example:
	`<let name="password" type="string"></let>`
	
	`<function name="ncfunc.password" > password`
	`</function>`
	`<function name="ncfunc.password" > _T"CUSTOMER"`
	`</function>`
	
	Delete password:
	<function name="ncfunc.password" > _T"" </function>

Control form color From SW 1.4 SP7	This function provides the text or background color of the dialog box.
	
	Range:
	• BACKGROUND – request color value of the background
	• TEXT – request color value of the text (foreground)
	
	Syntax:
	**<function name="control.formcolor"
	return="variable">_T"range"</function>**
	
	Example:
	`<let name="bk_color"></let>`
	
	`<function name="control.formcolor"`
	`return="bk_color">_T"BACKGROUND"</function>`

<table>
<tr><th>Function name</th><th>Meaning</th></tr>
<tr><td>Control local timeFrom SW 1.4 SP7</td><td>The function copies the local time in a field with 7 array elements.
The name of the variable is expected as call parameter.
The following is stored in an array element:
• Index 0 - year
• Index 1 - month
• Index 2 - weekday
• Index 3 - day
• Index 4 - hour
• Index 5 - minute
• Index 6 - second

Syntax:
<code><function name
="control.localtime">_T"time_array"</function></code>

Example:
<code><!-- index
0 = Year
1 = Month
2 = Day of week
3 = Day
4 = Hour
5 = Minute
6 = Second
-->
<let name="time_array" dim="7" />

  <function name
="control.localtime">_T"time_array"</function></code></td></tr>
</table>

Function name	Meaning
String to compare	Two strings are compared with one another from a lexicographical perspective. The function gives a return value of zero if the strings are the same, a value less than zero if the first string is smaller than the second string or a value greater than zero if the second string is smaller then the first string. **Parameter:** **str1** - string **str2** - comparison string **Syntax:** <function name="string.cmp" return ="<int var>" > str1, str2 </function> **Example:** <let name="rval">0</let> <let name="str1" type="string">A brown bear hunts a brown dog.</let> <let name="str2" type="string">A brown bear hunts a brown dog.</let> <function name="string.cmp" return="rval"> str1, str2 </function> **Result:** rval= 0

Function name	Meaning
String to compare without making a distinction between uppercase/lowercase	Two strings are compared from a lexicographical perspective (the comparison is not case-sensitive). The function gives a return value of zero if the strings are the same, a value less than zero if the first string is smaller than the second string or a value greater than zero if the second string is smaller then the first string. Parameter: **str1** - string **str2** - Comparison string Syntax: <function name="string.icmp" return ="<int var>" > str1, str2 </function> Example: <let name="rval">0</let> <let name="str1" type="string">A brown bear hunts a brown dog.</let> <let name="str2" type="string">A brown Bear hunts a brown Dog.</let> <function name="string. icmp" return="rval"> str1, str2 </function> Result: rval= 0
String left	The function extracts the first nCount character from string 1 and copies this to the return variable. Parameter: **str1** - String **nCount** - Number of characters Syntax: <function name="string.left" return="<result string>"> str1, nCount </function> Example: <let name="str1" type="string">A brown bear hunts a brown dog.</let> <let name="str2" type="string"></let> <function name="string. left" return="str2"> str1, 12 </function> Result: str2="A brown bear"

Function name	Meaning

String right	The function extracts the last nCount character from string 1 and copies this to the return variable.
	
	Parameter:
	str1 - String
	nCount - Number of characters
	
	Syntax:
	`<function name="string.right" return="<result string>"> str1, nCount </function>`
	
	Example:
	`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`
	`<let name="str2" type="string"></let`
	
	`<function name="string. right " return="str2"> str1, 10 </function>`
	Result:
	str2="brown dog."

String middle	The function extracts the specified number of characters from string 1, starting from the iFirst index, and copies these to the return variable.
	
	Parameter:
	str1 - string
	iFirst - start index
	nCount - number of characters
	
	Syntax:
	`<function name="string.middle" return="<result string>"> str1, iFirst, nCount </function>`
	
	Example:
	`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`
	`<let name="str2" type="string"></let`
	
	`<function name="string. middle " return="str2"> str1, 2, 5 </function>`
	Result:
	str2="brown"

Function name	Meaning
String length	The function gives the number of characters in a string. **Parameter:** **str1** - string **Syntax:** <function name="string.length" return="<int var>"> str1 </function> **Example:** <let name="length">0</let> <let name="str1" type="string">A brown bear hunts a brown dog.</let> <function name="string.length" return="length"> str1 </function> **Result:** length = 31
Strings to replace	The function replaces all the substrings found with the new string. **Parameter:** **string** - string variable **find string** - string to be replaced **new string** - new string **Syntax:** <function name="<string.replace>"> string, find string, new string </function> **Example:** <let name="str1" type="string">A brown bear hunts a brown dog. </let> <function name="string.replace" > str1, _T"a brown dog" , _T"a big salmon"</function> **Result:** str1 = "A brown bear hunts a big salmon!"

Function name	Meaning
Strings to remove	The function removes all the substrings found. **Parameter:** **string** - string variable **remove string** - substring to be deleted **Syntax:** <function name="string.remove"> string, remove string </function> **Example:** <let name="index">0</let> <let name="str1" type="string">A brown bear hunts a brown dog. </let> <function name="string.remove" > str1, _T"a brown dog" </function **Result:** str1 = "A brown bear hunts"
Strings to insert	The function inserts a string at the index specified. **Parameter:** **string** - string variable **index** - index (zero based) **insert string** - string to be inserted **Syntax:** <function name="string.insert"> string, index, insert string </function> **Example:** <let name="str1" type="string">A brown bear hunts. </let> <let name="str2" type="string">a brown dog</let> <function name="string.insert" > str1, 19, str2</function> **Result:** str1 = "A brown bear hunts a brown dog"

Function name	Meaning

String delete	The function deletes the defined number of characters starting from the start position specified.
	
	Parameter:
	string - string variable
	start index - start index (zero based)
	nCount - number of characters to be deleted
	
	Syntax:
	`<function name="string.delete"> string, start index`
	`, nCount </function>`
	
	Example:
	`<let name="str1" type="string">A brown bear hunts.`
	`</let>`
	
	`<function name="string.delete" > str1, 2,`
	`5</function>`
	Result:
	str1 = "A bear hunts"

String find	The function searches the transferred string for the first match with the substring.
	If the substring is found, the function provides the index to the first character (starting with zero) or, failing this, -1.
	
	Parameter:
	string - string variable
	find string - string to be found
	
	Syntax:
	`<function name="string.find" return="<int val>">`
	`str1, find string </function>`
	
	Example:
	`<let name="index">0</let>`
	`<let name="str1" type="string">A brown bear hunts a`
	`brown dog. </let>`
	`<function name="string.find" return="index"> str1,`
	`_T"brown" </function>`
	Result:
	Index = 2

Function name	Meaning

String reverse find	The function searches the transferred string for the last match with the substring.
	If the substring is found, the function provides the index to the first character (starting with zero) or, failing this, -1.
	
	Parameter:
	string - string variable
	find string - string to be found
	
	Syntax:
	`<function name="string.reversefind" return="<int val>"> str1, find string </function>`
	
	Example:
	`<let name="index">0</let>`
	`<let name="str1" type="string">A brown bear hunts a brown dog. </let>`
	`<function name="string.reversefind" return="index"> str1, _T"brown" </function>`
	Result:
	Index = 21

String trim left	The function trims the starting characters from a string.
	
	Parameter:
	str1 - string variable
	
	Syntax:
	`<function name="string.trimleft" > str1 </function>`
	
	Example:
	`<let name="str1" type="string">        test trim left</let>`
	`<function name="string.trimleft" > str1 </function>`
	Result:
	str1 = "test trim left"

Function name	Meaning
String trim right	The function trims the closing characters from a string. **Parameter:** **str1** - string variable **Syntax:** <function name="string.trimright" > str1 </function> **Example:** <let name="str1" type="string"> test trim right </let> <function name="string.trimright" > str1 </function> **Result:** str1 = "test trim right"
Sine	The function calculates the sine of the value transferred in degrees. **Parameter:** **double** - angle **Syntax:** <function name="sin" return="<double val>"> double </function> **Example:** <let name= "sin_val" type="double"></let> <function name="sin" return="sin_val"> 20.0 </function>
Cosine	The function calculates the cosine of the value transferred in degrees. **Parameter:** **double** - angle **Syntax:** <function name="cos" return="<double val>"> double </function> **Example:** <let name= "cos_val" type="double"></let> <function name="cos" return="cos_val"> 20.0 </function>

Function name	Meaning
Tangent	The function calculates the tangent of the value transferred in degrees. **Parameter:** **double** - angle **Syntax:** <function name="tan" return="<double val>"> double </function> **Example:** <let name= "tan_val" type="double"></let> <function name="tan" return="tan_val"> 20.0 </function>
arcsin	The function calculates the arcsine of the value transferred in degrees. **Parameter:** **double** - x in the range from -PI/2 to +PI/2 **Syntax:** <function name="arcsin" return="<double val>"> double </function> **Example:** <let name= "arcsin_val" type="double"></let> <function name="arcsin" return=" arcsin_val"> 20.0 </function>
ARCOS	The function calculates the arccosine of the value transferred in degrees. **Parameter:** **double -** x in the range from -PI/2 to +PI/2 **Syntax:** <function name="arcos" return="<double val>"> double </function> **Example:** <let name= "arccos_val" type="double"></let> <function name="arccos" return=" arccos_val"> 20.0 </function>

Function name	Meaning
arctan	The function calculates the arctan of the value transferred in degrees. **Parameter:** **double** - arctan of y/x **Syntax:** <function name="arctan" return="<double val>"> double </function> **Example:** <let name= "arctan_val" type="double"></let> <function name="arctan" return="arctan_val"> 20.0 </function>
DLL load	The function loads an additional user DLL to the memory. **Parameter:** **dll_name** - DLL name **class_name** - name of the function class **Syntax:** <function name="dll.load"> dll_name, class_name </function> **Example:** <function name="dll.load"> _T"customer.dll", _T"customer" </function>
Dll function	The function calls a function from a user DLL. All parameters listed after the parameter ID are transferred to the function called. **Parameter:** **class_name** - name of the function class **id** - of the function **parameter** - maximum seven function parameters (string variables) **Syntax:** <function name="dll.function"> class_name, id, parameter1, parameter2</function> **Example** <function name="dll.function"> _T"customer", 290, _T"par1", _T"par2"</function>

Function name	Meaning
File processing	
Loading a file	The function loads the contents of the file specified to a string variable. **Attribute:** **Return** - name of the local variable **Parameter:** **Progname** - file name **Syntax:** <function name="doc.readfromfile" return="<string var>"> progname </function> **Example:** <let name = "my_var" type="string" ></let> <function name=" doc.readfromfile " return="my_var"> _T"\spf\test.mpf" </function>
Writing to a file	The function writes the contents of a string variable to the file specified. **Parameter:** **progname** - file name **str1** - string **Syntax:** <function name="doc.writetofile" > progname, str1 </function> **Example:** <let name = "my_var" type="string" > file content </let> <function name="doc.writetofile">_T"\spf\test.mpf", my_var </function>
Deleting a file	The function removes the file specified from the directory. **Parameter:** **progname** - file name **Syntax:** <function name="doc.remove" > progname </function> **Example:** <function name="doc.remove">_T"\mpf\test.mpf" </function>

Function name	Meaning
Exist	If the file exists, the function returns the value 1. **Parameter:** **progname** - file name **Syntax:** <function name="doc.exist" return="<int_var>" > progname </function> **Example:** <let name ="exist">0</let> <function name="doc.exist" return="exist">_T"\mpf\test.mpf" </function>
NC program selection	The function selects the program specified for execution. The program must be stored in the NC file system. **Parameter:** **progname** - file name **Syntax:** <function name="ncfunc.select"> progname </function> **Example:** <function name="ncfunc.select"> _T"\mpf\test.mpf" </function>

Function name	Meaning
Setting an individual bit From SW 1.4 SP7	The function is used to manipulate individual bits of the specified variables. The bits can either be set or reset. **Syntax:** <function name="ncfunc.bitset" refvar="address" value="set/reset" > bit0, bit1, … bit9 </function> **Attributes:** **refvar** - specifies the name of the variable, in which the bit combination should be written **value** – bit value, value range 0 and 1 **Values:** The bit numbers starting with zero should be transferred as function values. A maximum of 10 bits per call can be modified. **Example:** <function name="ncfunc.bitset" refvar="nck/Channel/Parameter/R[1]" value="1" > 0, 2, 3, 7 </function> <function name="ncfunc.bitset" refvar="nck/Channel/Parameter/R[1]" value="0" > 1, 4 </function>
Delete control From SW 1.4 SP7	The function deletes the specified picture control. **Syntax:** <function name="control.delete"> control name </function> **Attribute:** **name** – function name **Value:** **control name** – name of the control **Example:** <function name="<control.delete>"> _T"my_editfield" </function>

Function name	Meaning
Add Item From SW 1.4 SP7	The function inserts a new element at the end of the list. Note: The function is only available for the control types "listbox" and "graphicbox". **Syntax:** <function name="control.additem"> control name, item </function> **Attribute:** **name** – function name **Values:** **control name** – Control name **item** - expression to be inserted **itemdata** - integer value; defined by the user **Example:** <let name ="itemdata">1</let> … … … <op> item_string = _T"text1" </op> <function name="control.additem">_T"listbox1", item_string, itemdata </function>

Function name	Meaning
Insert Item From SW 1.4 SP7	The function inserts a new element at the specified position. Note: The function is only available for the control types "listbox" and "graphicbox". **Syntax:** <function name="control.insertitem"> control name, index, item, itemdata </function> **Attribute:** **name** – function name **Values:** **control name** – Control name **index** – position starting with zero **item** - expression to be inserted **itemdata** - integer value; defined by the user **Example:** <let name ="itemdata">1</let> … … … <op> item_string = _T"text2" </op> <function name="control.insertitem">_T"listbox1", 1, item_string, itemdata </function>
Delete item From SW 1.4 SP7	The function deletes an element at the specified position. Note: The function is only available for the control types "listbox" and "graphicbox". **Syntax:** <function name="control.deleteitem"> control name, index </function> **Attribute:** **name** – function name **Values:** **control name** – Control name **index**– index starting at 0 **Example:** <function name="control.deleteitem">_T"listbox1", 1</function>

<table>
<tr><th>Function name</th><th>Meaning</th></tr>
<tr><td>Load Item
From SW 1.4 SP7</td><td>The function inserts a list of expressions into the control.
The function is only available for the control types "listbox" and "graphicbox".

Syntax:
<code><function name="control.loaditem"> control name, list </function></code>

Attribute:
name – function name

Values:
control name – Control name
list- string variable

Structure of the list:
The list contains a number of expressions, which must be separated from one another using a \n.

Example:
<code><let name="item_string" type="string"></let>
<let name="plotlist" type="string"></let>
…
…
<print name ="item_string" text="p; %f; %f; %f; %f\n">s_z, s_x</print>
      <op>plotlist = plotlist + item_string</op>
      <print name ="item_string" text="l; %f; %f; %f; %f\n">s_z, s_x, e_z, e_x </print>
      <op>plotlist = plotlist + item_string</op>
      <op> s_x = e_x </op>
      <op> s_z = e_z</op>
      <op> e_x = s_x + 10 </op>
      <op> e_z = s_z - 100 </op>
      <print name ="item_string" text="l; %f; %f; %f; %f\n">s_z, s_x, e_z, e_x </print>
      <op>plotlist = plotlist + item_string</op>

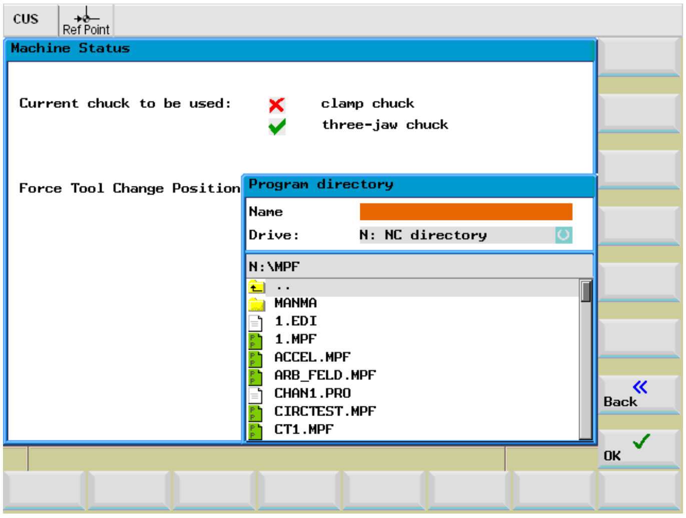
      <function name="control.loaditem">_T"gbox", plotlist</function></code></td></tr>
</table>

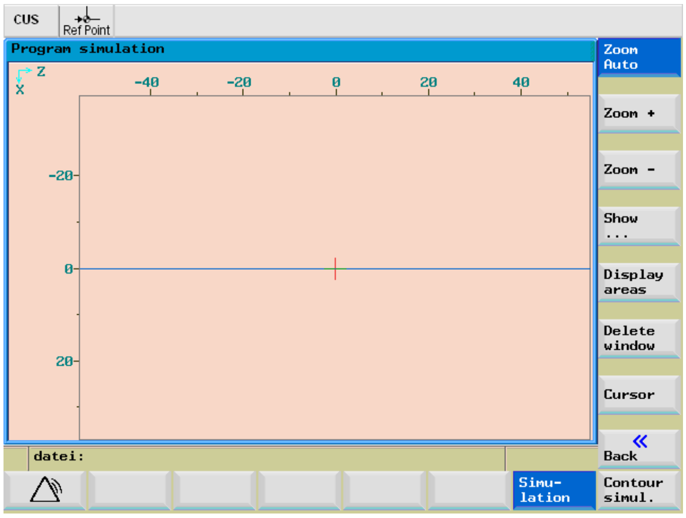
Function name	Meaning
Empty From SW 1.4 SP7	The function deletes the contents of the specified list box or graphic box controls. **Syntax:** <function name="control.empty"> control name, </function> **Attribute:** **name** – function name **Values:** **control name** – Control name **Example:** <function name=" control.empty">_T"listbox1"</function>
Get focus From SW 1.4 SP7	The function supplies the name of the control, which has the input focus. **Syntax:** <function name="control.getfocus" return="focus_name" /> **Attributes:** **name** – function name **return** – a string variable should be specified, into which the control name is copied. **Example:** <let name>="focus_field" type="string"></let> <function name="control.getfocus" return="focus_field"/>

Function name	Meaning
Set focus From SW 1.4 SP7	The function sets the input focus to the specified control. The Controlname should be transferred as text expression of the function. **Syntax:** <function name="control.setfocus"> control name </function> **Attribute:** **name** – function name **Value:** **control name** – name of the control **Example:** <function name="control.setfocus" ">_T"listbox1"</function>
Get cursor selection From SW 1.4 SP7	For a list box, the function supplies the cursor index. The Controlname should be transferred as text expression of the function. **Syntax:** <function name="control.getcurssel" retvar="var"> control name </function> **Example:** <let name>="index"></let> <function name="control.getcurssel" ">_T"listbox1"</function>
Set cursor selection From SW 1.4 SP7	For a list box, the function sets the cursor to the appropriate line. The Controlname should be transferred as text expression of the function. **Syntax:** <function name="control.setcurssel" > control name, index</function> **Example:** <let name>="index">2</let> <function name="control.setcurssel" ">_T"listbox1",index</function>

Function name	Meaning
Get Item From SW 1.4 SP7	For a list box, the function copies the contents of the selected line to the specified variable. A string variable should be specified as reference variable. The Controlname should be transferred as text expression of the function. **Syntax:** <function name="control.getitem" return="var"> control name, index </function> **Example:** <let name>="index">2</let> <let name>="item" type="string"></let> <function name="control.getitem" return="item" ">_T"listbox1",index</function>
Get Item Data From SW 1.4 SP7	For a list box, the function copies the user-specific allocated value of an element to the specified variable. For an edit control, the function copies the user-specific allocated value (item_data) to the specified variable. An integer variable should be specified as reference variable. The Controlname should be transferred as text expression of the function. **Syntax:** <function name="control.getitemdata" return="var"> control name, index </function> **Example:** <let name>="index">2</let> <let name>="itemdata"></let> <function name="control.getitemdata" return="itemdata" ">_T"listbox1",index</function>
Abs From SW 1.4 SP7	This function returns the absolute value of the specified number. **Syntax:** <function name="abs" return="var"> value </function>
SDEG From SW 1.4 SP7	The function converts the specified value into degrees. **Syntax:** <function name="sdeg" return="var"> value </function>
SRAD From SW 1.4 SP7	The function converts the specified value into RADian. **Syntax:** <function name="srad" return="var"> value </function>

Function name	Meaning
SQRT From SW 1.4 SP7	The function calculates the square root of the specified value. Syntax: <function name="sqrt" return="var"> value </function>
ROUND From SW 1.4 SP7	The function rounds of the transferred number to the specified number of decimal places. If the number of decimal places is not specified, then the function rounds off the number, taking into account the first decimal place. Syntax: <function name="round" return="var"> value, nDecimalPlaces </function>
FLOOR From SW 1.4 SP7	The function supplies the largest possible integer value, which is less than or equal to the transferred value. Syntax: <function name="floor" return="var"> value </function>
CEIL From SW 1.4 SP7	The function supplies the smallest possible integer value, which is greater than or equal to the transferred value. Syntax: <function name="ceil" return="var"> value </function>
LOG From SW 1.4 SP7	The function calculates the logarithm of the specified value. Syntax: <function name="log" return="var"> value </function>
LOG10 From SW 1.4 SP7	The function calculates the common (decadic) logarithm of the specified value. Syntax: <function name="log10" return="var"> value </function>
POW From SW 1.4 SP7	The function calculates the value "a^b". Syntax: <function name="pow" return="var"> a, b </function>
MIN From SW 1.4 SP7	The function compares the transferred value and returns the lower of the values. Syntax: <function name="min" return="var"> value1, value2 </function>

Function name	Meaning
MAX From SW 1.4 SP7	The function compares the transferred value and returns the higher of the values. Syntax: <function name="max" return="var"> value1, value2 </function>
RANDOM From SW 1.4 SP7	The function returns a pseudo random number. Syntax: <function name="random" return="var" </function>
Dialogs	
Program selection dialog	The function opens a dialog for selecting a part program. The selected program name is copied to the reference variable specified. The dialog can only be called from within a softkey tag.  Syntax: <function name= "doc.fileselect" return="string var" /> Example: <let name = "program_content" type="string" > </let> <let name = "curr_program" type="string" > </let> <function name= "doc.fileselect" return="curr_program" /> <function name="doc.readfromfile " return=" program_content" > curr_program </function>

<table>
<tr><th>Function name</th><th>Meaning</th></tr>
<tr><td>Program simulation dialog</td><td>The function opens program simulation in the form of a pop-up dialog.
If a program name is specified, then the program is automatically selected. Otherwise, the simulation represents the traversing motion of the currently selected program.
The dialog can only be called from within a softkey tag.
The "Back" softkey function returns the user to the previous dialog.


Syntax:
<function name= "ncfunc.progsimulation"> prog name </function>

Example:
<let name = "curr_program" type="string" > </let>
<function name= "doc.fileselect" return="curr_program" />
<function name= "ncfunc.progsimulation"> curr_program </function></td></tr>
</table>

8.14.10 Debugger

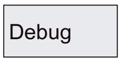
SINUMERIK 802D sl has a debugger for debugging XML instructions.

Note

The debugger is available at the "manufacturer" password level.

The debugger can be activated via the display machine data MD1109 TM_FUNCTION_MASK bit 3.

Operating sequences



- Press the "Debug" softkey in the <CUSTOM> operating area.

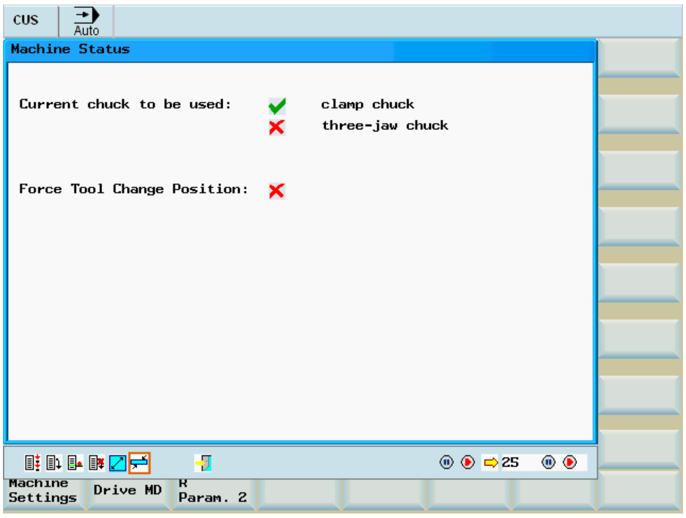


Figure 8-54 Debugger with application window

The command window for the debugger is displayed within the information line.

The debugger stops at the first command.

Figure 8-55 Command window for the debugger

- Use the <TAB> key to switch between the command window and the application window.
- Use the cursor keys to select the functions in the command window for the debugger.

The following functions can be executed:

Step mode 

Program run 

Set interruption point 

End program run 

Show instructions window

Hide instructions window

End debug session

The debugger also shows the program status and the current instruction line.

189

Program stopped		Lit
Program is running		Lit
Execute the instruction on the line		

8.14.10.1 Step mode

Place the cursor above the "Step mode" icon and press <Input>. An instruction is executed.

8.14.10.2 Program run

Place the cursor above the "Program run" icon and press <Input>. Instructions are executed without any interruptions.

You can use the "End program run" or "Step mode" icons to stop the program run.

8.14.10.3 End program run

Place the cursor above the "End program run" icon and press <Input>. The debugger switches to step mode.

8.14.10.4 Show instructions window

Place the cursor above the "Show instructions window" icon and press <Input>. The debugger opens a window showing the instructions to be executed and the variables currently loaded.

Within the Instructions area, this icon indicates the step which is currently being executed (see screenshot below).

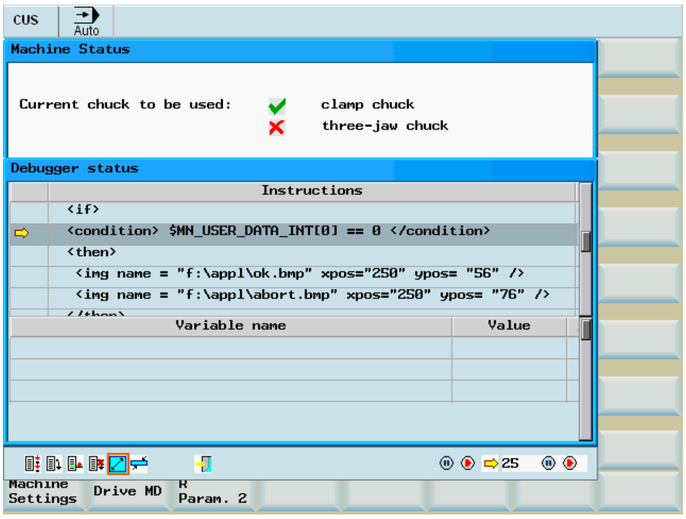


Figure 8-56 Debugger status

You can use the TAB key to switch between the debug window and the instructions window.

Place the cursor in the instructions window to set an interruption point. Next, select the desired instruction and press <Input>.

This icon indicates the interruption point which has been set.

If you press <Input> repeatedly, the debugger will delete the interruption point.

8.14.10.5 Hide instructions window

Place the cursor above the "Hide instructions window" icon and press <Input>. The debugger closes the instructions window.

8.14.10.6 End debug session

Place the cursor above the "End debug session" icon and press <Input>. The debugger closes and the selection menu appears.

8.15 Generating commissioning dialogs

8.15.1 Overview of functions

Purpose

The "Startup Wizard" allows additional devices to be simply commissioned, activated, deactivated and tested. The available equipment and device states are displayed in a list by the control system. The system can manage a maximum of 64 devices.

Softkeys are used to activate or deactivate a device.

The "Startup Wizard" function is available in the <SYSTEM> operating area.

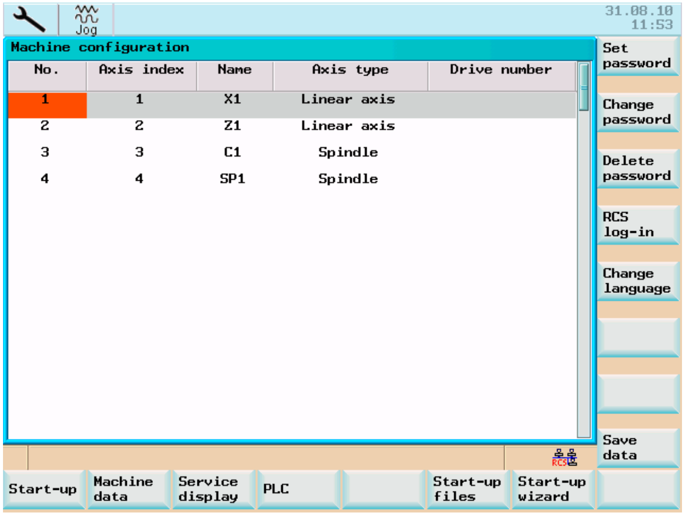


Figure 8-57 The SYSTEM main screen with active "Startup Wizard" softkey

Configuration

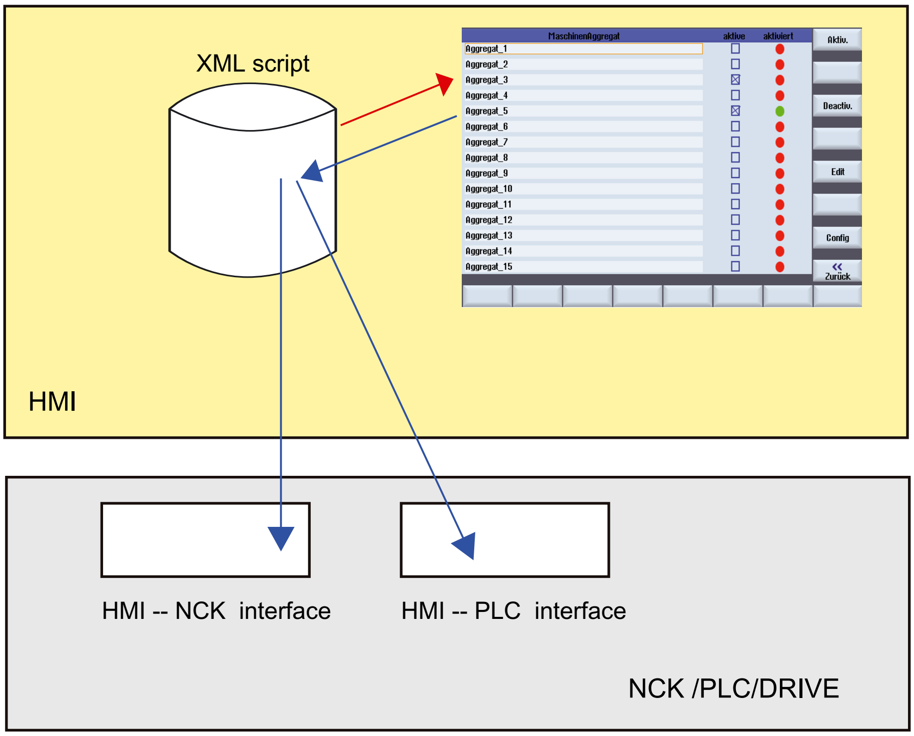


Figure 8-58 Mode of operation of the "Startup Wizard"

To use the "Startup Wizard", the following functions should be configured by the machine manufacturer:

- PLC ↔ HMI interface

  The optional devices are managed via the interface between the user interface and the PLC.

- Script processing

  The machine manufacturer saves the sequences to be executed for commissioning, activating, deactivating or testing a device, in a statement script.

- Parameter dialog (optional)

  The parameter dialog shows device information that is saved in the script file.

Storage of the files

The "Startup Wizard" files are stored on the system CompactFlash card in the "dvm" (machine builder) directory.

File	Name	Target directory
Text file	oem_aggregate_xxx.ts	`F:\lng`
Script file	agm.xml	`F:\dvm`

File	Name	Target directory
Archive file	Any	`F:\dvm\archives`
PLC user program	Any	PLC

8.15.2 Configuration in the PLC user program

Loading configurations

The configurations created are transferred to the manufacturer directory of the control, with the script and text file. Additionally, the corresponding PLC user program should be loaded.

Programming the equipment

Communication between the operator component and the PLC takes place in the PLC user program via data block VB99050000, in which 128 words are reserved for the device management.

PLC words are assigned beginning with Device 1:

Data block	Device designation
VB99050000	Device 1
VB99050004	Device 2
VB99050008	Device 3
VB99050012	Device 4 etc.

Four bytes with the following meanings are used for each device:

Byte	Bit	Description	
0	0	== 1	Device has been started up (HMI acknowledgment)
	1	== 1	Device is to be activated (HMI request)
	2	== 1	Device is to be deactivated (HMI request)
	3-7	Reserved	
1	0-7	Reserved	
2	0	== 1	Device is active (PLC acknowledgment)
	1	== 1	Device has an error
	2-7	Reserved	
3	0-7	Unique identifier for the device	

General sequence

The machine manufacturer must execute the following steps to make the required data available:

1. Creating a PLC user program which activates the device during activation on the PLC.
2. Commissioning of the "standard machine" followed by backup of the data in a series startup archive.
3. Installation of the devices, commissioning, followed by read-out of the data as a differential series startup archive.

Note

Changing the machine configuration

Should there be any need to edit the drive machine data, this should be adapted in the control first. This procedure should be repeated for all devices and constellations.

Adding axes

If the machine is extended with machine axes, it is important to install the drive objects (DO) in a fixed sequence because the series startup archive contains the constellation of the machine manufacturer's reference machine and cannot be applied if the sequence is changed.

It is recommended that the following settings be selected for the "control components":

- NC data
- PLC data
- Drive data
  – ACX format (binary)

8.15.3 Display on the user interface

Dialogs on the user interface

The following dialogs are available for "Startup Wizard":

- The control offers a **configurable dialog**, in which the available devices are shown.
- If first commissioning has not taken place yet, the control opens the **commissioning dialog**.

If a commissioning procedure (XML instruction: "START_UP") has been programmed, and the device has still not been commissioned, then the control starts the commissioning procedure.

This involves a complete data backup before the series startup archives saved in the script file are read in.

In the event of an error, the commissioning engineer can decide whether to roll back the commissioning procedure or to rectify possible errors in machine configurations manually.

- Commissioning can be aborted early with the "Cancel" function. The control then copies the previously saved commissioning files back.

If the machine has to be switched off after successful completion of the commissioning, the XML statement "POWER_OFF" can be used to program that a corresponding message is output on the control.

8.15.4 Creating language-dependent texts

Structure of text file

The XML files with the language-dependent texts must be created in UTF8 format:

Example oem_aggregate_eng.ts

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
       <context>
       <name>EASY_EXTEND</name>
           <message>
           <source>DEVICE_ONE</source>
           <translation>Device one</translation>
           </message>
           <message>
           <source>DEVICE_TWO</source>
           <translation>Device two</translation>
           </message>
       </context>
</TS>
```

Example oem_aggregate_deu.ts

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
       <context>
       <name>EASY_EXTEND</name>
            <message>
           <source>DEVICE_ONE</source>
           <translation>Device one</translation>
           </message>
           <source>DEVICE_TWO</source>
           <translation>Device two</translation>
           </message>
       </context>
</TS>
```

8.15.5 User example for a power unit

Activating the drive object

The drive object to be activated has already been commissioned and deactivated again by the machine manufacturer, to market the axis (axes) as an option.

To activate the axis carry out the following steps:

- Activate the drive object via p0105.
- 2. Enable the axis in the channel machine data.
- Back up the drive machine data via p0971.
- Wait until the data has been written.
- Restart the NCK and the drives.

Programming:

```
<DEVICE>
  <list_id>1</list_id>
  <name> "Activate the drive" </name>

  <SET_ACTIVE>
    <data name = "drive/dc/p105[DO5]">1</data>
    <data name = "$MC_AXCONF_MACHAX_USED[4]">5</data>
    <data name = "drive/dc/p971[DO5]">1</data>
    <while>
      <condition> "drive/dc/p971[DO5]" !=0 </condition>
    </while>
    <control_reset resetnc ="true" resetdrive = "true"/>
  </SET_ACTIVE>

  <SET_INACTIVE>
    <data name = "drive/dc/p105[DO5]">0</data>
    <data name = "$MC_AXCONF_MACHAX_USED[4]">0</data>
    <data name = "drive/dc/p971[DO5]">1</data>
    <while>
      <condition> "drive/dc/p971[DO5]" !=0 </condition>
    </while>
    <control_reset resetnc ="true" resetdrive = "true"/>
  </SET_INACTIVE>

</DEVICE>
```

Activating the PLC-controlled device

The device is addressed via output byte 10 and signals data set ready to the PLC via input byte 9.

The output byte is set to the specified coding for activation. The WHILE loop then waits for the data set ready of the device.

Programming:

```
<SET_ACTIVE>
  <DATA name = "plc/qb10"> 8 </DATA>
  <while>
    <condition> "plc/ib9" !=1 </condition>
  </while>
</SET_ACTIVE>
```

8.15.6 Script language

Note

All of the script elements described in the Generating user dialogs (Page 179) function form the basis for the "Startup Wizard" function. Additional script elements are defined to manage additional devices.

Program parts of the script

The script is divided into the following areas:

- "Startup Wizard" - frame
- Frame to define the actions that can be executed for a device
- Identifier for the device
- Identifier for commissioning the device
- Identifier for activating the device
- Identifier for deactivating the device
- Identifier for testing the device
- Identifier for the parameter dialog

The individual tags are described in the following chapters.

Description

Identifier <tag>	Meaning
AGM	Identifier for the "Startup Wizard"
DEVICE	Identifier for the description of the device. **Attributes:** • **option_bit** The device is assigned a fixed bit number for the option management.
NAME	The identifier specifies the name of the device to be displayed in the dialog. If a text reference is used, the dialog displays the text which is saved for the identifier.
START_UP	The identifier contains a description of the sequences required for commissioning the device.
SET_ACTIVE	The identifier contains a description of the sequences required to activate the device. **Attributes:** • **timeout** The attribute permits a timeout to be specified in seconds. The system interrupts processing if the script has still not been completed after this time.
SET_INACTIVE	The identifier contains a description of the sequences required to shut down the device. **Attributes:** • **timeout** The attribute permits a timeout to be specified in seconds. The system interrupts processing if the script has still not been completed after this time.
TEST	The identifier contains the statements for testing the operating capability of a device. **Attributes:** • **timeout** The attribute permits a timeout to be specified in seconds. The system interrupts processing if the script has still not been completed after this time.
UID	Unique numerical identifier to identify the device in the PLC ↔ HMI interface.
VERSION	Identifier for a version

Negative acknowledgment of the function execution

With the automatically provided variable "$actionresult", the system can inform the XML parser of a negative execution result. If the value is set to zero, the parser aborts the function processing.

Example

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE AGM>
<AGM>                                    Identifier for the "Startup Wizard"
<DEVICE>
 <NAME> Device 1 </NAME>                 Identifier for the device
 <START_UP>                              Identifier for commissioning the device
 …
 </START_UP>
 <SET_ACTIVE>                            Identifier for activating the device
 …
 </SET_ACTIVE>
 <SET_INACTIVE>                          Identifier for deactivating the device
 …
 </SET_INACTIVE>
 <TEST>                                  Identifier for testing the device
 …
 </TEST>
</DEVICE>
…
</AGM>
```

8.15.6.1 CONTROL_RESET

Description

This identifier allows one or more control components to be restarted. Execution of the script is only continued when the control has resumed cyclic operation.

Programming

Identifier:	CONTROL_RESET	
Syntax:	<CONTROL_RESET resetnc="TRUE" />	
Attributes:	resetnc="true"	The NC component is restarted.
	resetdrive="true"	The drive components are restarted.

8.15.6.2 FILE

Description

The identifier enables standard archives to be read in or created.

- Reading in an archive:

  The file name of the archive must be specified for reading in an archive.
- Creating an archive:

  If the attribute create= "true" is specified, the function creates a standard archive (*.arc) under the specified name and stores the file in the …/dvm/archives directory.

Programming

Identifier:	FILE	
Syntax:	<file name ="<archive name>" />	
	<file name ="<archive name>" create="true" group="<range>" />	
Attributes:	name	Identifier for the file name
	create	A commissioning archive is created under the specified name in the `…/dvm/archives/` directory.
	group	Specifies the data groups that are to be contained in the archive. If several data groups are to be saved, the groups should be separated by a blank. The following data groups can be contained in the archive: • NC • PLC • HMI • DRIVES

Example

```
<!-- Create data class archive -->
<file name="user.arc" create="true"
group="nc plc hmi" />

<!—Read archive into the control->
<file name="user.arc" />
```

8.15.6.3 OPTION_MD

Description

The identifier allows option machine data to be redefined. As delivered, the system uses MD14510 $MN_USER_DATA_INT[0] to $MN_USER_DATA_INT[3].

If the PLC user program manages the options, the appropriate data words must be provided in a data block or GUD.

The data is structured in bits. Starting with bit 0, there is a fixed assignment of the bits to the listed devices, i.e. bit 0 is assigned to device 1, bit 1 to device 2, etc. If more than 16 devices are managed, the address identifiers of the device groups 1-3 are assigned via the area index.

Note

Converting the value range

The value range of MD14510 $MN_USER_DATA_INT[i] is from -32768 to +32767. To activate the devices bit-by-bit via the machine data dialog, the bit combination must be converted to decimal representation.

Programming

Identifier:	**OPTION_MD**	
Syntax:	Area 0:	
	<option_md name = "Address identifier of the data" />	
	OR:	
	<option_md name = "Address identifier of the data" index= "0"/>	
	Area 1 to 3:	
	<option_md name = "Address identifier of the data" index= "Area index"/>	
Attributes:	name	Identifier for the address, e.g. $MN_USER_DATA_INT[0]
	index	Identifier for the area index:
		0 (default setting): Device 1 to 16
		1: Device 17 to 32
		2: Device 33 to 48
		3: Device 49 to 64

8.15.6.4 PLC_INTERFACE

Description

This identifier permits the PLC ↔ HMI interface to be redefined. The system expects 128 addressable words.

Default: DB9905 (VB99050000)

Programming

Identifier:	PLC_INTERFACE	
Syntax:	<plc_interface name = "Address identifier of the data" />	
Attributes:	name	Identifier for the address, e.g. "plc/mb170"

Example: `plc/mb170`

8.15.6.5 POWER_OFF

Description

Identifier for a message prompting the operator to switch the machine off. The message text is permanently saved in the system.

Programming

Identifier:	POWER_OFF
Syntax:	<power_off />
Attributes:	--

8.15.6.6 WAITING

Description

After a reset of the NC or the drive, there is a wait for the restart of the respective component.

Programming

Identifier:	WAITING	
Syntax:	<WAITING WAITINGFORNC ="TRUE" />	
Attributes:	waitingfornc="true"	There is a wait for the restart of the NC.
	waitingfordrive="true"	There is a wait for the restart of the drive.

8.15.6.7 XML identifiers for the dialog

Dialog for the parameterization

A dialog can be configured for each device so that additional parameters can be set or output during runtime. This is displayed by pressing the "Additional parameters" softkey.

All of the script elements described in Chapter Generating user dialogs (Page 179) can be used to generate the dialog.

Example

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE AGM>
<AGM>
<DEVICE>
 <NAME> Device 1 </NAME>
 <START_UP>
 …
 </START_UP>
 <SET_ACTIVE>
 …
 </SET_ACTIVE>
…
 <FORM>                                    Identifier for a user dialog
 <INIT>
 <CONTROL name = "edit1" .../>             Identifier for an input field
 </INIT>
 <PAINT>                                   Identifier for text or image display
 <TEXT>hello world !</TEXT>
 </PAINT>
 </FORM>
```

```
</DEVICE>
…
</AGM>
```

8.15.6.8 SOFTKEY_OK, SOFTKEY_CANCEL

Description

The identifier SOFTKEY_OK overwrites the standard behavior when closing a dialog by means of the "OK" softkey. The identifier SOFTKEY_CANCEL overwrites the standard behavior when closing a dialog by means of the "CANCEL" softkey.

The following functions can be performed within this identifier:

- Data manipulation
- Conditional processing
- Loop processing

Programming

Identifier:	**SOFTKEY_OK**
Syntax:	<SOFTKEY_OK> … </SOFTKEY_OK>
Identifier:	**SOFTKEY_CANCEL**
Syntax:	<SOFTKEY_CANCEL> … </SOFTKEY_CANCEL>

9 Starting Up the PLC

9.1 Overview

General information

The PLC is intended to control machine-related functional sequences. It is realized as a software PLC.

The user program - a PLC cycle - is always executed in the same order of sequence.

- Refresh of the process image (inputs, user interface, timers)
- Processing of communication requests (operator panel, PLC 802 programming tool, version 3.0 and higher)
- Editing of the user program
- Evaluation of alarms
- Output of the process image (outputs, user interface)

During the cycle, the PLC executes the user program from the first to the last operation. The user program accesses the hardware inputs/outputs only via the process image and not directly. The PLC refreshes the hardware I/Os at the beginning or end of program execution. Thus, these signals are stable over a whole PLC cycle.

The user program can only be created using the PLC 802 Programming Tool, version 3.1 and higher, with the S7-200 programming language using ladder diagram. Ladder diagram is a graphical programming language for representing electric circuit diagrams.

Note

PLC 802 Library including a description which can be installed from the toolbox CD is provided as the basis for the PLC user program. This contains a subroutine library and example programs.

If the Stop and Reset buttons are not implemented as normally closed contacts, a break in the line cannot be detected.

Monitoring can take place via software solutions, as shown in the example MCP_802D (SBR 34) of the subroutine library.

9.2 Programming Tool PLC802

The Programming Tool PLC 802 programming package provides a user-friendly environment for developing, editing, and observing the logic to control your applications.

9.2.1 Selecting the target system

In the Programming Tool PLC802, the CPU type can be selected as the preset. In the operation tree, the operations that cannot be used for the target system, are marked with a red X ().

By presetting the CPU type, an error check of the program already takes place when the program is written.

Note

If the CPU type is not preset when opening a new project, all of the operations, addresses and functions in the Programming Tool PLC802 are available and can be used in the program. No check occurs during the input. Errors in the presetting for the CPU type will not be displayed until after the download has been completed and the control restarted.

Procedure

- You are now in the Programming Tool PLC802.
- Select "Target system" > "CPU type" from the menu, or right-click "Project name (CPU type)" in the operation tree.

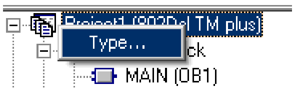


Figure 9-1 Select the CPU type by clicking with the right mouse key

- You select a target system from the list box.

  Example:

  Range and functional limitations of the **latest** firmware version of the 802Dsl TM plus are taken into consideration. In order to ensure that both the CPU type and the product version of the firmware are taken into consideration when the range checks are carried out, you can have the Programming Tool PLC802 read the CPU type information directly from the target system. For more information, refer to the Programming Tool PLC802 online help.

- Reading the removed CPU type using the Programming Tool PLC 802

  To read-out the CPU type and product version of the firmware, click on the "Read target system" button in the "CPU-Type" dialog box.

  The CPU type and the firmware version are displayed in the list box.

9.2.2 Interface to PLC

Independently of the installed hardware, the following options are available for the connection setup between the control system and the PG/PC:

- Via RS232 cable

  The parameters that are preset in the PLC802 programming tool must be accepted. No further adaptation is required.
- Via modem-V24 cable

  Communication settings must be adjusted in the control and in the PLC802 programming tool.
- Via a crossed TP cable for an Ethernet peer-to-peer connection

  Communication settings must be adjusted in the control and in the PLC802 programming tool.
- Optionally via a network (Ethernet)

  Communication settings must be adjusted in the control and in the PLC802 programming tool.

You can set up the communication or you can edit the communication settings at any time.

9.2.2.1 Establishing a connection via the RS232 interface

The RS232 (V24) port can be used for connecting between the control system and the PC/PG (Programming Tool PLC802).

Activating the connection to the control system

The connection is activated on the operator panel of the control in the <SYSTEM> operating area via softkeys "PLC" > "STEP 7 connect." > "Connect. active". The active or inactive state is retained even after Power On (except power-up with the default data). An active connection is displayed by a symbol in the status bar.

Communication settings in the PLC802 programming tool

To setup the PPI parameters in the PLC802 programming tool, proceed as follows:

1. In the navigation bar, click the communication icon or select "View" > "Communication" from the menu.

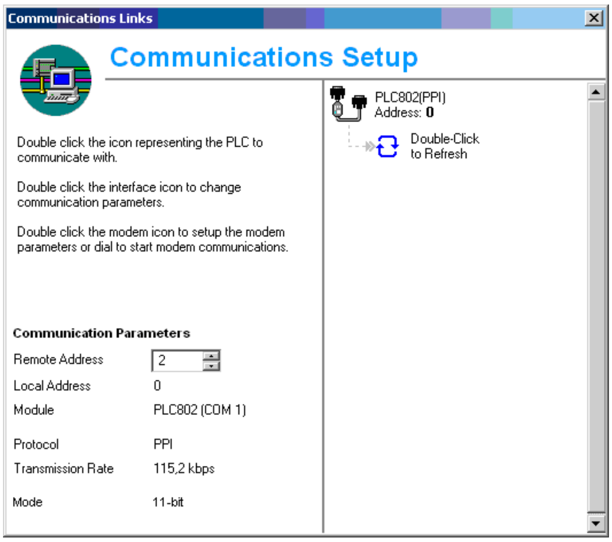


Figure 9-2 Communication settings

2. Double click on the "Access point" symbol in the "Communication" window.

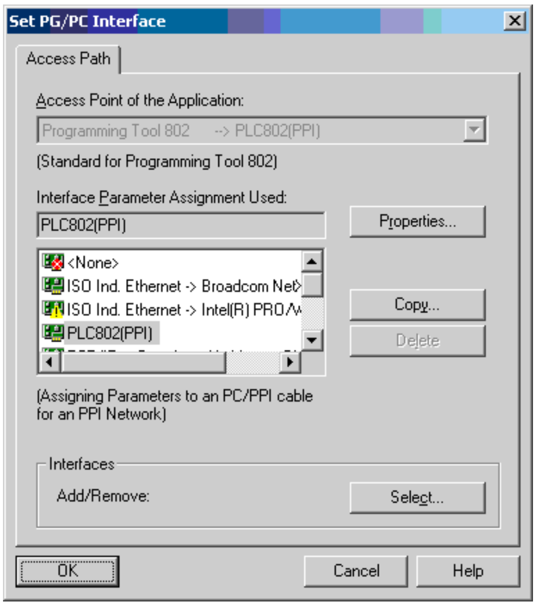


Figure 9-3 Setting the interface

3. Check the PG/PC interface in use. For RS232 communication, the interface 802D(PPI) must be assigned to the PLC802 programming tool.
4. Set the baud rate for the transmission rate, which the Programming Tool PLC802 will use to communicate. The 802D sl supports 9.6 kBaud, 19.2 kBaud, 38.4 kBaud, 57.6 kBaud and 115.2 kBaud.

5. Open the "local connection" tab.

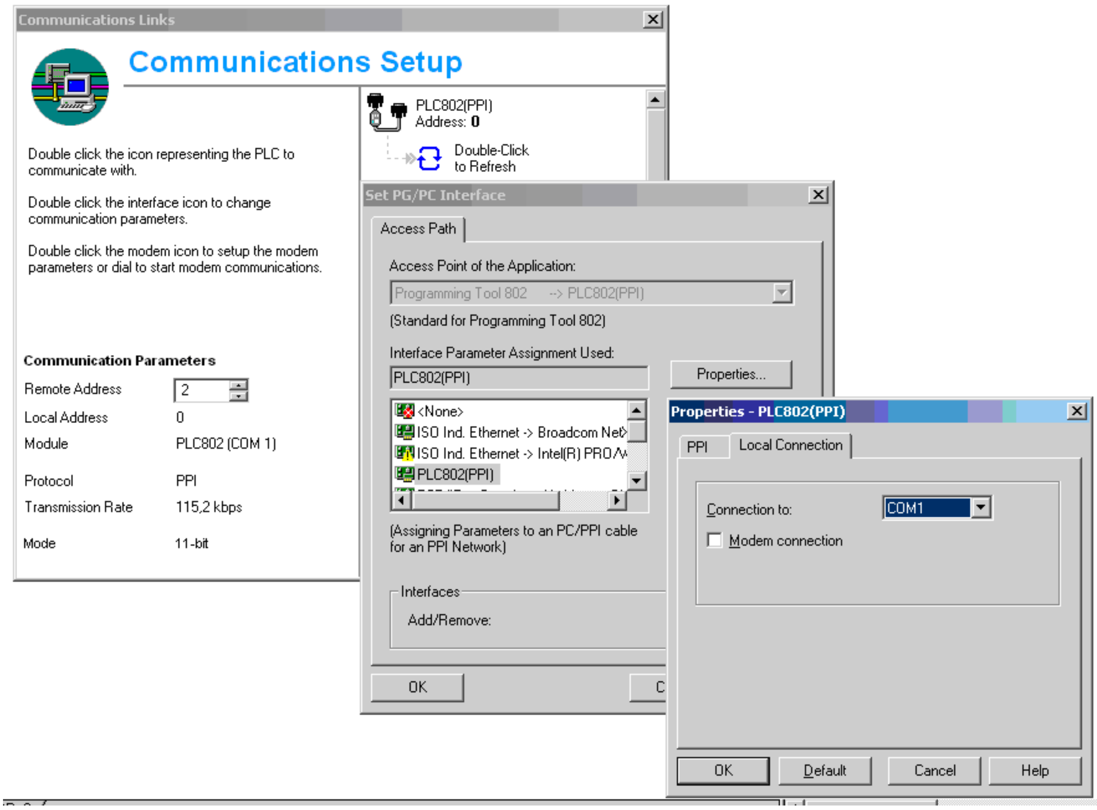


Figure 9-4 Opening the "Local connection" window

6. In the "local connection" tab, specify the COM port to which the RS232 (V24) cable is connected.
7. Click "OK" twice to exit the "Set PG/PC Interface" dialog box.
8. On the right of the "Communications Setup" dialog box, click the blue text "Double-Click to Refresh".

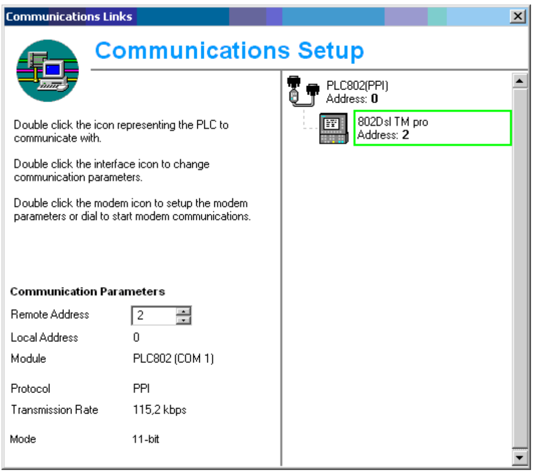


Figure 9-5 Communications link

Note

The connection must be activated on the control (<SYSTEM> operating area > "PLC" > "Step 7 connect." > "Connect. active").

9.2.2.2 Establishing a connection via modem

The RS 232/modem (V24) interface can be used for connecting between the control system and the PC/PG (e.g. programming tool PLC802).

Activating the connection on the control system

The connection is activated on the operator panel of the control system in the <SYSTEM> operating area via softkeys "PLC" > "STEP 7 connect." > "Connect. active".

You must select "ON" to activate the modem.

Note

Prerequisite for modem functions: Bit 5 is set in the general machine data MD19334 SYSTEM_FUNCTION_MASK.

The active or inactive state is retained even after Power On (except power-up with the default data). An active connection is displayed by a symbol in the status bar.

Communication settings in the Programming Tool PLC802

Proceed as follows to set up the modem connection:

1. In the navigation bar, click the communication icon or select "View" > "Communication" from the menu.

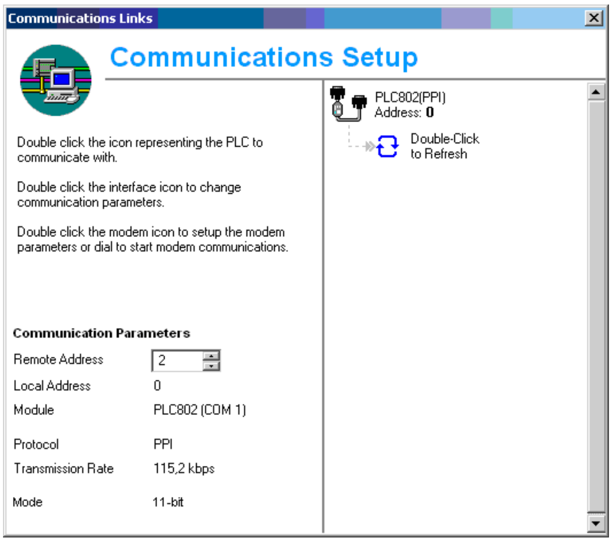


Figure 9-6 Communication settings

2. Double click the "Access point" symbol in the "Communication" window.

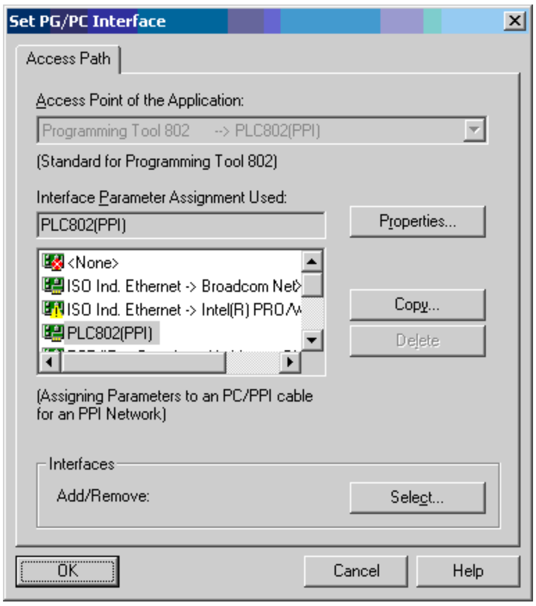


Figure 9-7 Setting the interface

3. Check the PG/PC interface in use. For RS232 communication, the interface "PLC802D(PPI)" must be assigned to the PLC802 programming tool.

4. For the transmission rate, set the baud rate that the PLC802 programming tool will use to communicate via the modem.

   The SINUMERIK 802D sl supports 9.6 kBaud, 19.2 kBaud, 38.4 kBaud, 57.6 kBaud, and 115.2 kBaud.

5. Open the "Local Connection" tab.

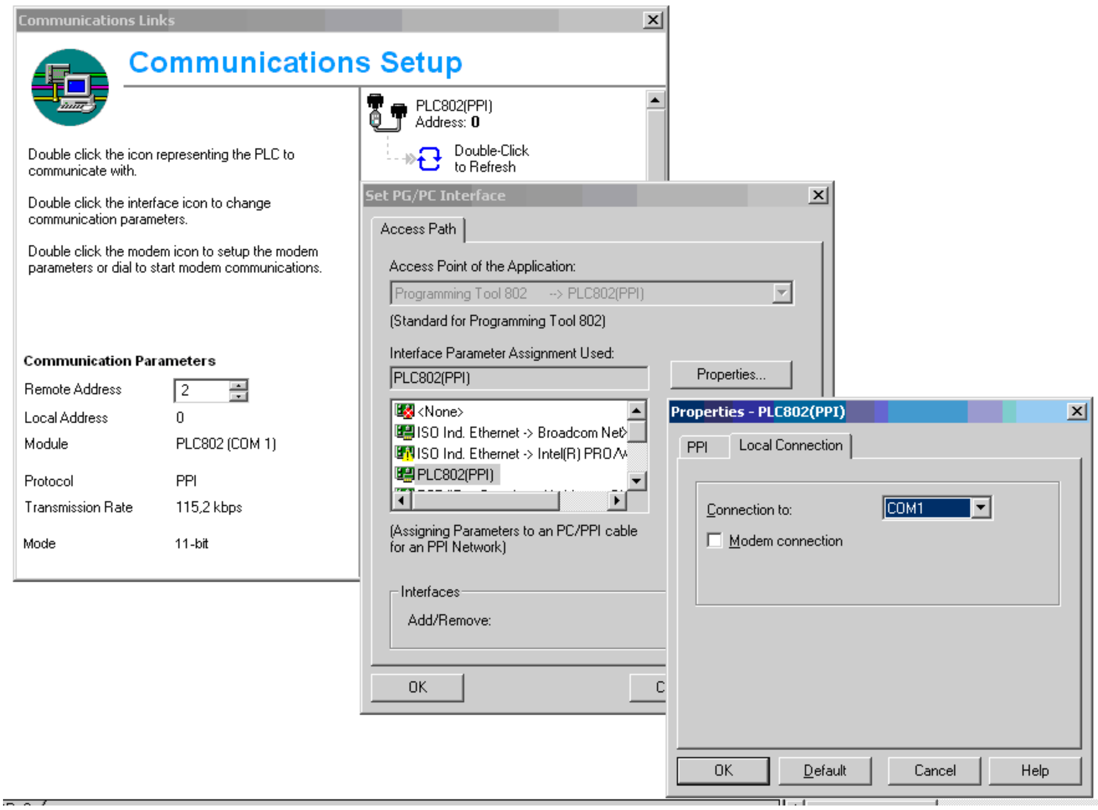


Figure 9-8 Opening the "Local Connection" window

6. On the "Local Connection" tab, specify the COM port to which the modem (V24) cable is connected.

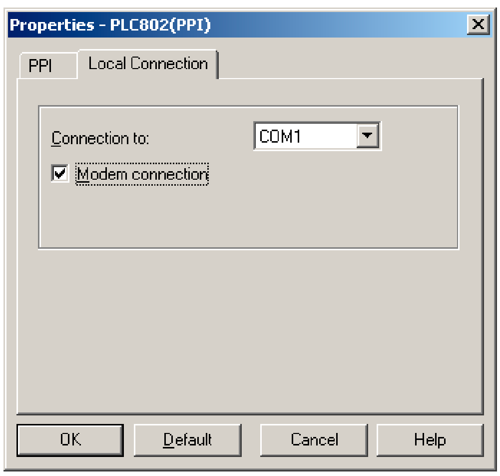


Figure 9-9 Modem link

7. Check the "Modem connection" box to set up the modem communication connection.
8. Click "OK" twice to exit the "Set PG/PC Interface" dialog box.
9. If you have selected a modem for the interface "PLC802D(PPI)" (e.g. Macom 33.6), proceed with the communication settings shown below.

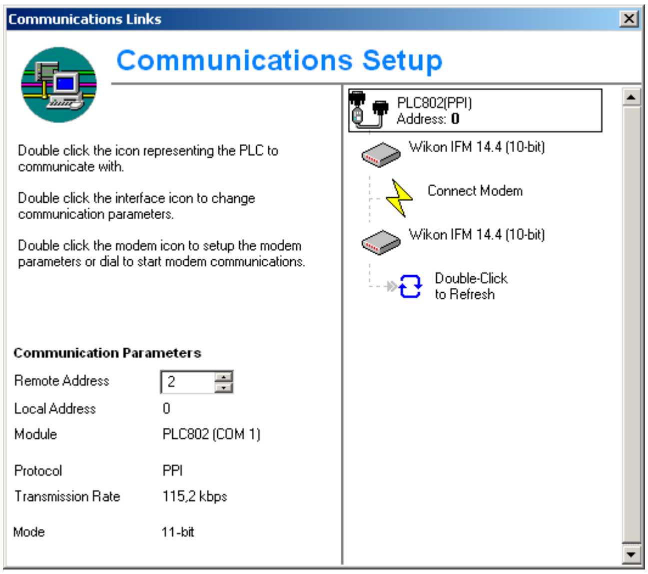


Figure 9-10 Modem selection

Double-click the modem displayed to set it to sender or receiver.

10. To connect, double-click "Connect Modem" and enter the telephone number of the receiver (control system).

9.2.2.3 Connection via Ethernet/peer-to-peer

The PLC802 programming tool needs port 102 for Ethernet communication.

Release communication port 102 on the control

Release is effected on the operator panel of the control in the <SYSTEM> operating area using softkeys "Service display" > "Service control" > "Service network" > "Service firewall".

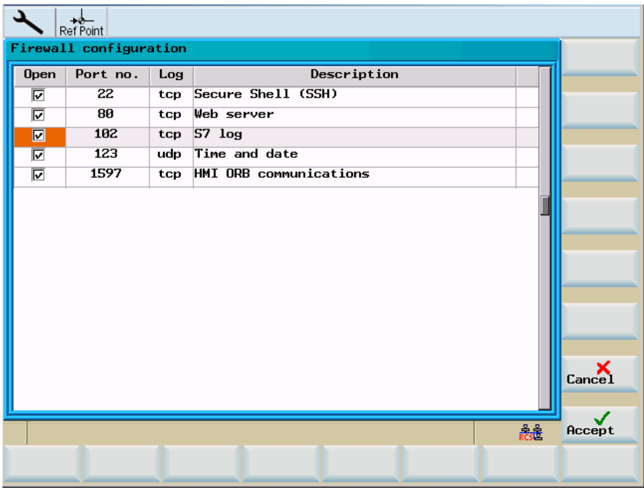


Figure 9-11 Port No. 102

The field for port No. 102 must be selected in the "Firewall Configuration" window.

Communication settings on the PLC802 programming tool

Proceed as follows to setup the network connection:

1. In the navigation bar, click on the communication symbol or select View > Communication from the menu.
2. Double click on the "Access point" symbol in the "Communication" window.

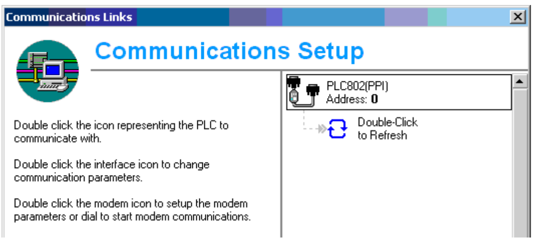


Figure 9-12 Ethernet communication settings

The "Set PG/PC interface" dialog box opens

3. In the box "Interface parameterization used", select "TCP/IP" with the arrow on the Ethernet card for your PC.

Note

You can find the name of your Ethernet card under the start menu "Start" > "Settings" > "Network connections".

This menu shows the appropriate TCP/IP device names of the Ethernet card in the "Device name" box.

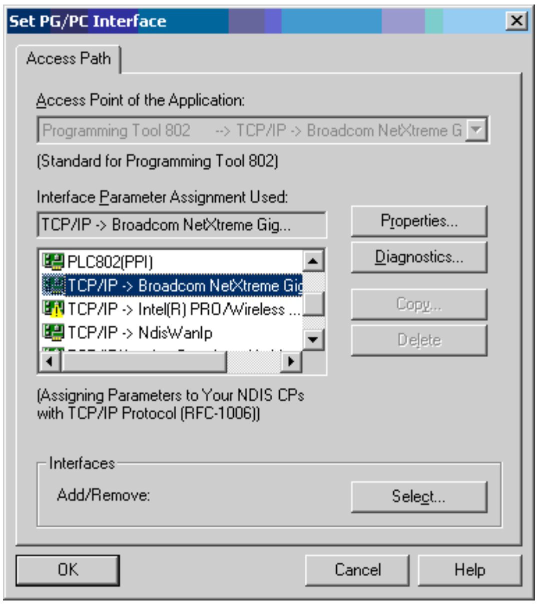


Figure 9-13 Network card settings

4. In the "Set PG/PC interface" dialog box, select the "OK" button.
5. Enter the IP address for the corresponding 802Dsl control in the "Communication Parameters" group box (see following figure).
   – If there is an Ethernet peer-to-peer connection, enter the IP address "169.254.11.22".

     This Ethernet peer-to-peer connection is activated on the control in the <SYSTEM> operating area with softkeys "Service display" > "Service control" > "Direct connect."
6. Double-click on the Refresh symbol to establish a connection to the specified IP address.
   – If a connection exists and the type of target system is successfully determined, the icon for the target system will appear in the "Communication" box.
   – If the connection attempt fails, the IP address is displayed as "not available" in the "Communication" dialog box.
   – If a connection exists, but the PLC802 Programming Tool cannot determine the type of target system, the IP address will appear as "unknown".

Note

The connection must be enabled at the control system (**Port 102**).

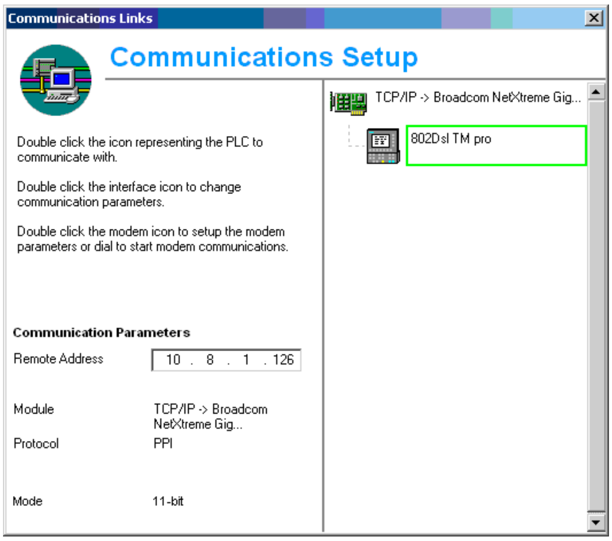


Figure 9-14 Network communication connection

9.3 First commissioning of the PLC

In the delivered condition of the SINUMERIK 802D sl, the PLC user program only consists of a NOP statement (no operation) and is stored in permanent memory. A PLC user program corresponding to the requirements of the machine must be created by the user.

9.4 Commissioning modes

Introduction

The 802D sl supports two different commissioning modes:

- After POWER ON, pressing the <SELECT> button while the control system is powering up.
- In the <SYSTEM> operating area, the "Startup" softkey after control startup.

On the PG/PC, commissioning takes place using the PLC802 programming tool.

Commissioning

The table below describes:

- Selection options for each respective commissioning mode
- Responses within the PLC

Table 9- 1 Commissioning modes

Selection			Response			
PCU Startup menu while the control system is powering up (802D sl)	PCU Startup menu after the control system has powered up (802D sl)	PT PLC802 (PG/PC)	PLC program preselection	Program status	Retentive data (supported)	MD for the PLC in the user interface
	NCK Start Up *					
Normal powerup	Normal powerup		User program ***	Run	Unchanged	Accepting the active PLC MD
Power-up with default values	Power-up with default values		User program ***	Run	deleted	Standard PLC MD
Power-up with saved data	Power-up with saved data		User program ***	Run	saved data	Saved PLC MD
PLC - Stop after POWER ON		PLC stop possible in Run or Stop	Unchanged	Stop	Unchanged	Accepting the active PLC MD
PLC clear all / Default PLC program			NOP user program	Run	deleted	Standard PLC MD
	PLC Start Up **					
	Cold restart	Run (after Stop)	User program ***	Run	Unchanged	Accepting the active PLC MD

Selection			Response			
PCU Startup menu while the control system is powering up (802D sl)	PCU Startup menu after the control system has powered up (802D sl)	PT PLC802 (PG/PC)	PLC program preselection	Program status	Retentive data (supported)	MD for the PLC in the user interface
	Cold restart and debug mode		User program ***	Stop	Unchanged	Accepting the active PLC MD
	CPU memory reset		User program ***	Run	deleted	Accepting the active PLC MD
	CPU memory reset and debug mode		User program ***	Stop	deleted	Accepting the active PLC MD
* <SYSTEM> operating area > "Startup" > "NC" * <SYSTEM> operating area > "Startup" > "PLC" *** loads from the permanent memory into RAM memory						

Via the debug mode, the PLC remains in PLC stop after the control system start up. All startup modes that are set via a softkey only become effective after the next control system start up.

The Run mode activates the cyclic operation.

In the Stop mode, the following actions are activated:

- All hardware outputs blocked
- Profibus–DP is inactive
- No cyclic operation (active user program not executed)
- The process image is no longer refreshed (frozen)
- EMERGENCY STOP active

The user only has the capability of loading a corrected or new project into the control system in Stop mode. The user program only becomes active after the next start up of the control system or of Run mode.

9.5 PLC alarms

9.5.1 Overview

The control system displays a maximum of 8 PLC alarms (system alarms or user alarms).

The PLC manages the alarm information per PLC cycle. It saves or deletes the alarms in the alarm list in chronological order based on the time of their occurrence. The first alarm in the list is always the last alarm that occurred.

When there are more than 8 alarms, the first seven alarms that occurred and the last (chronologically) are displayed with the highest deletion priority.

Alarm response and cancel criteria

The PLC also manages the alarm responses. The alarm reactions are always in effect regardless of the number of active alarms. Depending on the type of alarm response, the PLC activates the necessary action.

A cancel criterion must be defined for each alarm. By default, the PLC uses the cancel criterion SELF-CLEARING (see configuration of user alarms).

The following clearing criteria are possible:

- POWERONCLEAR: The alarm is canceled by turning off / turning on the control system (POWER ON).
- CANCELCLEAR: The alarm is cancelled by pressing the Cancel key or Reset key (analog NCK - alarms).
- SELF-CLEARING: The alarm is cleared by the no longer existent cause of the alarm.

The clearing conditions have the following priority:

- POWERON CLEAR - system alarms (highest priority)
- CANCEL CLEAR - system alarms
- SELF-CLEARING - system alarms
- POWERON CLEAR - user alarms
- CANCEL CLEAR - system alarms
- SELF-CLEARING - user alarm (lowest priority)

The responses that an alarm is supposed to trigger in the PLC are defined for each alarm. By default, the PLC uses the alarm response SHOWALARM.

The following are alarm responses:

- PLC - Stop : No further user programs are executed, PROFIBUS-DP inactive and disabling of the hardware outputs.
- EMERGENCY STOP: The PLC reports the EMERGENCY STOP signal to the NCK in the user interface after executing the user program.

- Feed disable: The PLC reports the FEED DISABLE signal to the NCK after processing the user program in the user interface.
- Read-in disable: The PLC reports the READ-IN DISABLE signal to the NCK after processing the user program in the user interface.
- NC Start disable: The PLC reports the NC START DISABLE signal to the NCK after processing the user program in the user interface.
- SHOWALARM: This alarm has no alarm response.

9.5.2 General PLC alarms

Note

see SINUMERIK 802D sl diagnostics guide

9.5.3 User alarms

The subareas (0, 1) are available to the user in the user interface " 1600xxxx " for defining a user alarm.

- Subarea 0: 8 x 8 bits for setting the user alarms (0 ->1 edge)
  byte 0 : Bit 0 => 1. user alarm " 700000 "
  byte 1: Bit 0 => 9. user alarm " 700008 "
  byte 7: Bit 7 => 64. user alarm " 700063 "
  byte 15: Bit 7 => 128. user alarm " 700127"

  A new user alarm is activated with the respective bit (subarea 0) with a 0/1 edge.
- Subarea 1: Variables of the user alarms

  Subarea 1 is provided for additional user information. It can only be written or read as a double word.
- Subarea 2: Alarm response
  Byte 0 : Bit 0 => NC Start disable
  Bit 1 => reading-in disable
  Bit 2 => feed disable of all axes
  Bit 3 => EMER STOP
  Bit 4 => PLC STOP

  With the aid of subarea 2, the user can evaluate the active alarm responses. It is read-only.

The user must clear self-clearing user alarms by resetting the respective bit in subarea 0 (1 -> 0 edge).

For the other user alarms, the PLC clears the corresponding user alarms after detecting the corresponding clearing conditions. If the bit of the user alarm is still on, the alarm reappears.

Method of operation of a user alarm

A user alarm has a higher priority than the corresponding signal in the user interface (e.g. NC Start disable, read-in disable and EMER stop).

Example:
MD14516[0] $MN_USER_DAT_PLC_ALARM = 8
While alarm 700000 is pending, alarm 3000 EMER Stop is also pending, although the interface signal V26000000.1=0.

Configuring user alarms

A configuration byte exists for each alarm. The user alarms can be configured by the user in the machine data **MD14516 $MN_USER_DATA_PLC_ALARM**.

Default setting MD14516[0...63]: 0 => SHOWALARM/SELF-CLEARING user alarm

Setup of the configuration byte:

- Bit0 - Bit5 : Alarm responses
- Bit6 - Bit7 : Clearing criterion

Alarm responses: Bit0 - Bit 5 = 0: Showalarm (default)
Bit0 = 1: NC Start disable
Bit1 = 1: Read-in disable
Bit2 = 1: Feed disable of all axes
Bit3 = 1: EMER Stop
Bit4 = 1: PLC Stop
Bit5 = reserved

Cancel criteria: Bit6 + Bit7 = 0: SELF-CLEARING alarm (default)
Bit6 = 1 : CANCELCLEAR alarm
Bit7 = 1 : POWERONCLEAR alarm

The user alarm response PLC-Stop always has the clearing condition POWER ON.

Alarm texts

The user has two options for defining his own alarm texts.

- Via <SYSTEM> operating area > "PLC" > "Exec." PLC Alarm txt"
- Via tool box: Editing and loading the alarm text file with the aid of the RCS802 tool

If the user does not assign a user alarm text, only the alarm number is displayed.

The % symbol in the alarm text designates an additional variable. The variable type represents the display form of the variable.

These variable types are possible:

- %D whole decimal numbers
- % I whole decimal numbers
- %U Decimal number without sign
- %O whole octal number

- %X whole hexadecimal number
- %B binary representation of 32 bit value
- %F 4 byte floating point number

Examples - user alarm texts (Note: The text after "//" is a comment and is not displayed.)

- 700000 " " // only user alarm number
- 700001 " HW limit switch axis X +"
- 700002 " %D " // only variable as a whole decimal number
- 700003 " Alarm number with fixed alarm text and variable %X "
- 700004 " %U Alarm number with variable and fixed alarm text "
- 700005 "Monitoring of axis active : %U"

  Display: 700005 "Monitoring of axis active : 1
  or 700005 monitoring of axis active : 3

9.6 PLC Programming

9.6.1 Overview

The PLC user program is created with the aid of the PLC 802 programming tool.

In the "SIMATIC S7-200 Automation System System Manual" documentation, you will find the handling instructions for an S7-200. The PLC 802 programming tool implements a subset of this documentation.

The following must be observed as compared to the basic S7-200 MicroWin system:

- It is only possible to program the user program in a ladder diagram.
- Only a subset of the programming language for the S7-200 is supported.
- The compilation of the user program is done offline on a PG/PC or automatically during the download into the control system.
- The project can be loaded into the control system (download).
- It is possible to load the project from the control system (download).
- No indirect addressing of the data is possible. Therefore, there are no programming errors in this respect while the program is running.
- The user must manage his data and process information by type.
  For all accesses to the data, the agreed data type must be consistently used.

How Do I
Information1 T-value memory size DInt (32 Bit)
Information 2 Override memory size byte (8 Bit)

User data
memory double word MD0 DInt (Information 1)
memory byte MB4 byte (Information 2)

- Furthermore, the alignment of the data to certain memory addresses is dependent upon the type of data (alignment). The alignment is done to byte addresses, which can be divided by the byte length of the data type with no remainder.

  BOOL and BYTE can begin at any byte address (0, 1, 2, 3, ...),
  WORD and INT must begin at an even byte address (0, 2, 4, 6, ...) and
  DWORD, DINT and REAL must begin at a byte address that is divisible by 4 (0, 4, 8, 12, ...).

  How Do I

  Memory bit MB0.1,MB3.5
  memory byte MB0,MB1,MB2
  memory word MW0,MW2,MW4
  MW3, MW5 ... are not permitted
  memory double word MD0,MD4,MD8
  MD1,MD2,MD3, MD5 ... are not permitted

Table 9- 2 PLC data types permitted in the control system

Data type	Size	Address alignment	Range for logical Operations	Range for arithmetical Operations
BOOL	1 bit	1	0, 1	-
BYTE	1 bytes	1	00 ... FF	0 ... +255
WORD	2 bytes	2	0000 ... FFFF	-32 768 ... + 32 767
DWORD (Double Word)	4 bytes	4	0000 0000 ... FFFF FFFF	-2 147 483 648 ... +2 147 483 647
REAL	4 bytes	4	-	$\pm 10^{-37} ... \pm 10^{38}$

PLC project

The PLC 802 programming tool always manages a project (combinational logic, symbols and comments). By downloading, it is possible to save all of the essential information of a project in the control system. By uploading, the information is transferred from the control system to the PC.

The control system can save a maximum of 6,000 instructional commands (4,000 for 802D sl value) and 1,500 symbols. The needed PLC memory is influenced by the following components:

- Number of statements
- Number and length of the symbol names
- Number and length of the comments

S7-200 Ladder diagram

The addresses and operations can be defined in the "International" display mode. In the ladder diagram, the user programs his program in networks. Each network corresponds to a logic that reflects a certain sequence. In a ladder diagram, contacts, coils and boxes are possible as basic elements. For the contacts, there are normally open and normally closed contacts. Each coil corresponds to a relay. A box reflects a certain function. A box can be activated using an enable bit.

9.6.2 Overview of commands

Table 9- 3 Operand identifier

Operand identifier	Description
V	data
T	Times
C	Meters
I	Image of digital inputs
Q	Image of digital outputs
M	Flag
SM	Special bit memory
AC	ACCU
L	Local data

Table 9- 4 Structure of V-range addresses (see PLC user interface)

Type ID (module no.)	Range no. (channel and axis No.)	Subarea	Offset	Addressing
00 (10-79)	00 (00-99)	0 (0-9)	000 (000-999)	Symbolic (8-digit)

Table 9- 5 802D sl address ranges

Access	Storage method	802Dsl TM value	802Dsl TM plus 802Dsl GN plus	802Dsl TM pro 802Dsl GN pro 802Dsl CU pro
Bit (Byte.bit)	V*	14000000.0-79999999.7	14000000.0-79999999.7	14000000.0-79999999.7
	I	0.0 – 26.7	0.0 – 26.7	0.0 – 26.7
	Q	0.0 – 17.7	0.0 – 17.7	0.0 – 17.7
	F	0.0 – 255.7	0.0 – 383.7	0.0 – 383.7
	SM	0.0 - 0.6	0.0 - 0.6	0.0 - 0.6

Access	Storage method	802Dsl TM value	802Dsl TM plus 802Dsl GN plus	802Dsl TM pro 802Dsl GN pro 802Dsl CU pro
	T	0–15 (100ms) 16–39 (10ms)	0–15 (100ms) 16–39 (10ms)	0–15 (100ms) 16–63 (10ms)
	C	0 – 31	0 – 31	0 – 63
	L	0.0 - 59.7	0.0 - 59.7	0.0 - 59.7
Byte	VB	14000000- 79999999	14000000- 79999999	14000000- 79999999
	IB	0 – 26	0 – 26	0 – 26
	QB	0 – 17	0 – 17	0 – 17
	MB	0 – 255	0 – 383	0 – 383
	SMB	0	0	0
	LB	0 – 59	0 – 59	0 – 59
	AC	0 – 3	0 – 3	0 – 3
Word	VW	14000000- 79999998	14000000- 79999998	14000000- 79999998
	IW	0 – 24	0 – 24	0 – 24
	QW	0 – 16	0 – 16	0 – 16
	MW	0 – 254	0 – 382	0 – 382
	T	0–15 (100ms) 16–39 (10ms)	0–15 (100ms) 16–39 (10ms)	0–15 (100ms) 16–63 (10ms)
	C	0 – 31	0 – 31	0 – 63
	LW	0 – 58	0 – 58	0 – 58
	AC	0 – 3	0 – 3	0 – 3
Double word	VD	14000000- 79999994	14000000- 79999994	14000000- 79999994
	ID	0 – 20	0 – 20	0 – 20
	QD	0 – 12	0 – 12	0 – 12
	MD	0 – 252	0 – 380	0 – 380
	LD	0 – 56	0 – 56	0 – 56
	AC	0 – 3	0 – 3	0 – 3

V*) The available address ranges are described in the PLC user interface.

References

SINUMERIK 802D sl lists

Example: Addressing in the variables memory

V3801040001.7: Bit access to the "Pulse enable" signal

Table 9- 6 Structure of V-range addresses

Type ID (module no.)	Range no. (channel and axis no.)	Subsection	Offset	Addressing
38 Axis signals	01 Axis 2	4 Signals to drive	001 Byte 1	7 Pulse enable

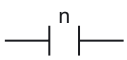
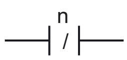
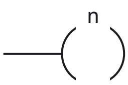
Table 9- 7 Special Marker SM Bit Definition

SM bits	Description
SM 0.0	Bit memory with defined ONE signal
SM 0.1	Initial setting: first PLC cycle '1', subsequent cycles '0'
SM 0.2	buffered data lost - only valid in first PLC cycle ('0' data ok, '1' data lost)
SM 0.3	POWER ON: first PLC cycle '1', subsequent cycles '0'
SM 0.4	60 s clock (alternating '0' for 30 s, then '1' for 30 s)
SM 0.5	1 s clock (alternating '0' for 0.5 s, then '1' for 0.5 s)
SM 0.6	PLC cycle clock (alternating one cycle '0', then one cycle'1')

The user can only view the statement list (STL) in the PT802 in "View STL". In this display method (see table: mnemonic), the sequential processing is displayed.

9.6.3 Explanation of the stack operations

Table 9- 8 BASIC BOOLEAN INSTRUCTIONS

BASIC BOOLEAN INSTRUCTIONS			
Instruction		Ladder Symbol	Valid Operands
Load And Or	normal open n=1 close n=0 open		n: V, I, Q, M, SM, T, C, L
Load Not And Not Or Not	normal close n=0 close n=1 open		n: V, I, Q, M, SM, T, C, L
Output	prior 0, n=0 prior 1, n = 1		n: V, I, Q, M,T, C, L

BASIC BOOLEAN INSTRUCTIONS			
Instruction		Ladder Symbol	Valid Operands
Set (1 Bit)	prior 0, not set prior 1 or ↗	Bit —(S)	S_Bit: V, I, Q, M, T, C, L n = 1
Reset (1 Bit)	prior 0, no reset prior 1 or ↗	Bit —(R)	S_Bit: V, I, Q, M, T, C, L n =1

Table 9- 9 OTHER BOOLEAN INSTRUCTIONS

OTHER BOOLEAN INSTRUCTIONS			
Instruction		Ladder Symbol	Valid Operands
Edge Up	prior ↗ close (1 PLC cycle)	—| P |—	
Edge Down	prior ↗ close (1 PLC cycle)	—| N |—	
Logical Not	prior 0, later 1 prior 1, later 0	—|NOT|—	
No operation		n —[NOP]—	n = 0 ... 255

Table 9- 10 BYTE COMPARES

BYTE COMPARES (Unsigned)			
Instruction		Ladder Symbol	Valid Operands
Load Byte = **And Byte =** **Or Byte =**	a = b close a ≠ b open	a —|==B|— b	a: VB, IB, QB, MB, SMB, AC, Constant, LB b: VB, IB, QB, MB, SMB, AC, Constant, LB
Load Byte≥ **And Byte ≥** **Or Byte≥**	a ≥ b close a < b open	a —|> =B|— b	
Load Byte ≤ **And Byte ≤** **Or Byte ≤**	a ≤ b close a > b open	a —|< =B|— b	
Load Byte ≠ **And Byte ≠** **Or Byte ≠**	a ≠b close a = b open	a —|<> B|— b	

BYTE COMPARES (Unsigned)			
Instruction		Ladder Symbol	Valid Operands
Load Byte > And Byte > Or Byte >	a > b close a ≤b open	a > B b	
Load Byte < And Byte < Or Byte <	a < b close a ≥b open	a < B b	

Table 9- 11 WORD COMPARES

WORD COMPARES (Signed)			
Instruction		Ladder Symbol	Valid Operands
Load Word = And Word = Or Word =	a = b close a ≠ b open	a ==I b	a: VW, T, C, IW, QW, MW, AC, Constant, LW b: VW, T, C, IW, QW, MW, AC, Constant, LW
Load Word≥ And Word≥ Or Word ≥	a ≥ b close a < b open	a > =I b	
Load Word ≤ And Word ≤ Or Word ≤	a ≤ b close a > b open	a < =I b	
Load Word ≠ And Word ≠ Or Word ≠	a ≠ b close a = b open	a < > I b	
Load Word > And Word > Or Word >	a > b close a ≤ b open	a > I b	
Load Word < And Word < Or Word <	a < b close a ≥b open	a < I b	

Table 9- 12 DOUBLE WORD COMPARES

DOUBLE WORD COMPARES (Signed)			
Instruction		Ladder Symbol	Valid Operands
Load DWord = And DWord = Or DWord =	a = b close a ≠ b open	a ==D b	a: VD, ID, QD, MD, AC, Constant, LB b: VD, ID, QD, MD, AC, Constant, LB
Load DWord≥ And DWord ≥ Or DWord ≥	a ≥ b close a < b open	a > =D b	
Load DWord ≤ And DWord ≤ Or DWord ≤	a ≤ b close a > b open	a < =D b	
Load DWord ≠ And DWord ≠ Or DWord ≠	a ≠ b close a = b open	a <> D b	
Load DWord > And DWord > Or DWord >	a > b close a ≤ b open	a > D b	
Load DWord < And DWord < Or DWord <	a < b close a ≥b open	a < D b	

Table 9- 13 REAL WORD COMPARES

REAL WORD COMPARES (Signed)			
Instruction		Ladder Symbol	Valid Operands
Load RWord = And RWord = Or RWord =	a = b close a ≠ b open	a = =R b	a: VD, ID, QD, MD, AC, Constant, LD b: VD, ID, QD, MD, AC, Constant, LD
Load RWord ≥ And RWord ≥ Or RWord ≥	a ≥ b close a < b open	a > =R b	
Load RWord ≤ And RWord ≤ Or RWord ≤	a ≤ b close a > b open	a < =R b	
Load RWord ≠ And RWord ≠ Or RWord ≠	a ≠ b close a = b open	a <> R b	

REAL WORD COMPARES (Signed)			
Instruction		Ladder Symbol	Valid Operands
Load RWord > And RWord > Or RWord >	a > b close a ≤ b open	a > R b	
Load RWord < And RWord < Or RWord <	a < b close a ≥b open	a < R b	

Table 9- 14 TIMER

TIMER			
Instruction		Ladder Symbol	Valid Operands
Timer Retentive On Delay	EN=1, Start EN=0, Stop If T_{Value} ≥ PT, T_{bit}=1	Txxx TONR IN PT	Enable: (IN) S0 Txxx: T0 - T63 (dependent on type of control system) Preset: (PT) VW, T, C, IW, QW, MW, AC, Constant 100 ms T0 - T15 10 ms T16 - T63
Timer On Delay	EN=1, Start EN=0, Stop If T_{Value} ≥ PT, T_{bit}=1	Txxx TON IN PT	Enable: (IN) S0 Txxx: T0 - T63 Preset: (PT) VW, T, C, IW, QW, MW, AC, Constant 100 ms T0 - T15 10 ms T16 - T63
Timer Of Delay	If T_{Value} < PT, T_{bit}=1	Txxx TOF IN PT	Enable: (IN) S0 Txxx: T0 - T63 Preset: (PT) VW, T, C, IW, QW, MW, AC, Constant 100 ms T0 - T15 10 ms T16 - T63

Table 9- 15 COUNTER

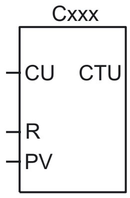
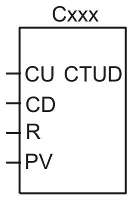
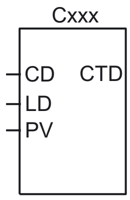
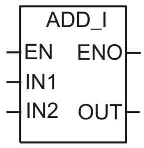
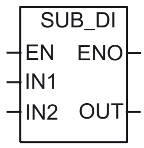
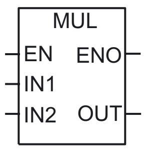
COUNTER			
Instruction		Ladder Symbol	Valid Operands
Count Up	CU ↗, Value+1 R=1, Reset If C_{Value} ≥ PV, C_{bit}=1		Cnt Up: (CU) S1 Reset: (R) S0 Cxxx: C0 - 63 Preset: (PV) VW, T, C, IW, QW, MW, AC, Constant, LW
Count Up/Down	CU ↗, Value+1 CD ↗, Value-1 R=1, Reset If C_{Value} ≥ PV, C_{bit}=1		Cnt Up: (CU) S2 Cnt Dn: (CD) S1 Reset: (R) S0 Cxxx: C0 - 63 Preset: (PV) VW, T, C, IW, QW, MW, AC, Constant, LW
Count Down	If C_{Value} = 0, C_{bit}=1		Cnt Down: (CD) S2 Reset: (R) S0 Cxxx: C0 - 63 Preset: (PV) VW, T, C, IW, QW, MW, AC, Constant, LW

Table 9- 16 MATH OPERATIONS

MATH OPERATIONS			
Instruction		Ladder Symbol	Valid Operands
Word Add **Word Subtract**	If EN = 1, b = a + b b = b - a		Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VW, T, C, IW, QW, MW, AC, LW
DWord Add **DWord Subtract**	If EN = 1, b = a + b b = b - a		Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD
Multiply	If EN = 1, b = a x b		Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VD, ID, QD, MD, AC, LD

MATH OPERATIONS			
Instruction		Ladder Symbol	Valid Operands
Divide	If EN = 1, b = b ÷ a Out: 16 bit remainder Out+2: 16 bit quotient	DIV EN ENO IN1 IN2 OUT	Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VD, ID, QD, MD, LD
Add Subtract Real Numbers	If EN = 1, b = a + b b = b - a	ADD_R EN ENO IN1 IN2 OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD
Multiply Divide Real Numbers	If EN = 1, b = a x b b = b ÷ a	MUL_R EN ENO IN1 IN2 OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD
Square Root	If EN = 1, OUT = √IN	SQRT EN ENO IN OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD

Table 9- 17 INCREMENT, DECREMENT

INCREMENT, DECREMENT			
Instruction		Ladder Symbol	Valid Operands
Increment Decrement Byte	If EN = 1, a = a + 1 a = a -1	INC_B EN ENO IN OUT	Enable: EN In: VB, IB, QB, MB, AC, Constant LB Out: VB, IB, QB, MB, AC, LB
Increment Decrement Word	If EN = 1, a = a + 1 a = a -1 a = /a	INC_W EN ENO IN OUT	Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VW, T, C, IW, QW, MW, AC, LW
Increment Decrement.	If EN = 1, a = a + 1 a = a -1	INV_DW EN ENO IN OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD

Table 9- 18 LOGIC OPERATIONS

LOGIC OPERATIONS			
Instruction		Ladder Symbol	Valid Operands
Byte AND Byte OR Byte XOR	If EN = 1, b = a AND b b = a OR b b = a XOR b	WAND_B EN ENO IN1 IN2 OUT	Enable: EN In: VB, IB, QB, MB, AC, Constant, LB Out: VB, IB, QB, MB, AC, LB
Word AND Word OR Word XOR	If EN = 1, b = a AND b b = a OR b b = a XOR b	WAND_W EN ENO IN1 IN2 OUT	Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VW, T, C, IW, QW, MW, AC, LW
DWord AND DWord OR DWord XOR	If EN = 1, b = a AND b b = a OR b b = a XOR b	WXOR_DW EN ENO IN1 IN2 OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD
Invert Byte	If EN = 1, a = /a	INC_B EN ENO IN OUT	Enable: EN In: VB, IB, QB, MB, AC, Constant, LB Out: VB, IB, QB, MB, AC, LB
Invert Word	If EN = 1, a = /a	INC_W EN ENO IN OUT	Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VW, T, C, IW, QW, MW, AC, LW
Invert DWord	If EN = 1, a = /a	INV_DW EN ENO IN OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD

Table 9- 19 SHIFT AND ROTATE OPERATIONS

SHIFT AND ROTATE OPERATIONS			
Instruction		Ladder Symbol	Valid Operands
Shift Right Shift Left	If EN = 1, a = a SR c bits a = a SL c bits	SHL_R EN ENO IN N OUT	Enable: EN In: VB, IB, QB, MB, AC, Constant, LB Out: VB, IB, QB, MB, AC Count: VB, IB, QB, MB, AC, Constant, LB
Shift Right Shift Left	If EN = 1, a = a SR c bits a = a SL c bits	SHL_W EN ENO IN N OUT	Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VW, T, C, IW, QW, MW, AC, LW Count: VB, IB, QB, MB,AC, Constant, LB

SHIFT AND ROTATE OPERATIONS			
Instruction		Ladder Symbol	Valid Operands
DWord Shift R DWord Shift L	If EN = 1, a = a SR c bits a = a SL c bits	SHL_DW EN ENO IN N OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD Count: VB, IB, QB, MB, AC, Constant, LB

Table 9- 20 CONVERSION OPERATIONS

CONVERSION OPERATIONS			
Instruction		Ladder Symbol	Valid Operands
Convert Double Word Integer to a Real	If EN = 1, convert the double word integer i to a real number o.	DI_REAL EN ENO IN OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD
Convert a Real to a Double Word Integer	If EN = 1, convert the real number i to a double word integer o.	TRUNC EN ENO IN OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD
Convert BCD to Binary	If EN = 1, convert the BCD value IN to a binary value OUT	BCD_I EN ENO IN OUT	Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VW, T, C, IW, QW, MW, AC, LW
Convert Binary to BCD	If EN = 1, convert the binary value IN to a BCD value OUT	I_BCD EN ENO IN OUT	Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VW, T, C, IW, QW, MW, AC, LW

Table 9- 21 PROGRAM CONTROL FUNCTIONS

PROGRAM CONTROL FUNCTIONS			
Instruction		Ladder Symbol	Valid Operands
Jump to Label	If EN = 1, go to label n.	n JMP	Enable: EN Label: WORD: 0-127
Label	Label marker for the jump.	n LBL	Label: WORD: 0-127
Conditional Return from Subroutine	If EN = 1, exit the subroutine.	RET	Enable: EN

PROGRAM CONTROL FUNCTIONS			
Instruction		**Ladder Symbol**	**Valid Operands**
Return from Subroutine	Exit subroutine.	(RET)	
Conditional End	If EN = 1, END terminates the main scan.	(END)	Enable: EN
Subroutine	If EN ↗, go to subroutine n.	n SBR EN x1 x2 x3 (x... optional parameters)	Label: Constant : 0-63

Table 9- 22 MOVE, FILL AND FIND OPERATIONS

MOVE, FILL AND FIND OPERATIONS			
Instruction		**Ladder Symbol**	**Valid Operands**
Move Byte	If EN = 1, copy i to o.	MOV_B EN ENO IN OUT	Enable: EN In: VB, IB, QB, MB, AC, Constant, LB Out: VB, IB, QB, MB, AC, LB
Move Word	If EN = 1, copy i to o.	MOV_W EN ENO IN OUT	Enable: EN In: VW, T, C, IW, QW, MW, AC, Constant, LW Out: VW, T, C, IW, QW, MW, AC, LW
Move DWord	If EN = 1, copy i to o.	MOV_DW EN ENO IN OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD
Move Real	If EN = 1, copy i to o.	MOV_R EN ENO IN OUT	Enable: EN In: VD, ID, QD, MD, AC, Constant, LD Out: VD, ID, QD, MD, AC, LD
Swap Bytes	If EN = 1, exchange MSB and LSB of w.	SWAP EN ENO IN	Enable: EN In: VW, IW, QW, MW, T, C, AC, LW

9.6.4 Program organization

Each programmer should structure his user program into finished program parts (subroutines). The programming language for S7-200 offers the user the capability to set up his user program in a structured manner. There are two types of programs, the main program and the subroutine. Eight levels of programming are possible.

A PLC cycle can be a simple cycle of the control system-internal interpolation cycle (IPO cycle). The machine manufacturer must set the PLC cycle according to his specific requirements (see machine data "PLC_IPO_TIME_RATIO"). The IPO/ PLC ratio of 1:1 is the fastest possible cyclical processing.

How Do I The programmer writes a sequential control in his main program with the aid of a user-defined cycle counter. This organizes all of the cyclical signals in the subroutine (UP0), UP1/UP2 are called up every two cycles and UP3 controls all of the signals in the grid of three cycles.

9.6.5 Data management

The data can be broken down into three areas:

- non-retentive data
- retentive data
- Machine data for the PLC (this machine data is all POWER ON active)

Most data, such as the process image, timers, and counters are non-retentive and are erased each time the control system is restarted.

For the retentive data, there is a data range of 1400 0000 -1400 0127. This location can be used to save all the data which is to remain valid beyond POWER OFF/ON.

With the aid of the PLC-MD (see user interface), the user can pre-assign his program with data or he can parameterize various parts of the program.

9.6.6 Testing and monitoring your program

Checking or performing an error analysis of the user program can be done using:

- PLC Status: Displaying and changing called up operands
- Status list: Displaying and changing three freely selectable variable boxes
- PLC Program: Displaying and monitoring (status) of the entire user program, including symbols and comments
- Programming Tool PLC802: Connecting a PG/PC and activating the programming tool.

9.7 PLC application Download/Upload/Copy/Compare

The user can save, copy or over-write the PLC project or the PLC applications in the control system.

This is possible using:

- Programming Tool PLC802
- RCS802
- CompactFlash card (CF card)

The **PLC project** contains the PLC user program, including all of the important information (symbols, comments, ...). The **PLC802 programming tool** uploads/downloads it to the control system. The PLC project can also be imported and exported by the programming tool in "*.pte" format. In this format (*.pte) the project can also be read from/to the CompactFlash card, either using the **RCS802 tool** or directly on the control system.

The **PLC user alarm texts** can be created using the RCS802 tool or the alarm text editor on the control system.

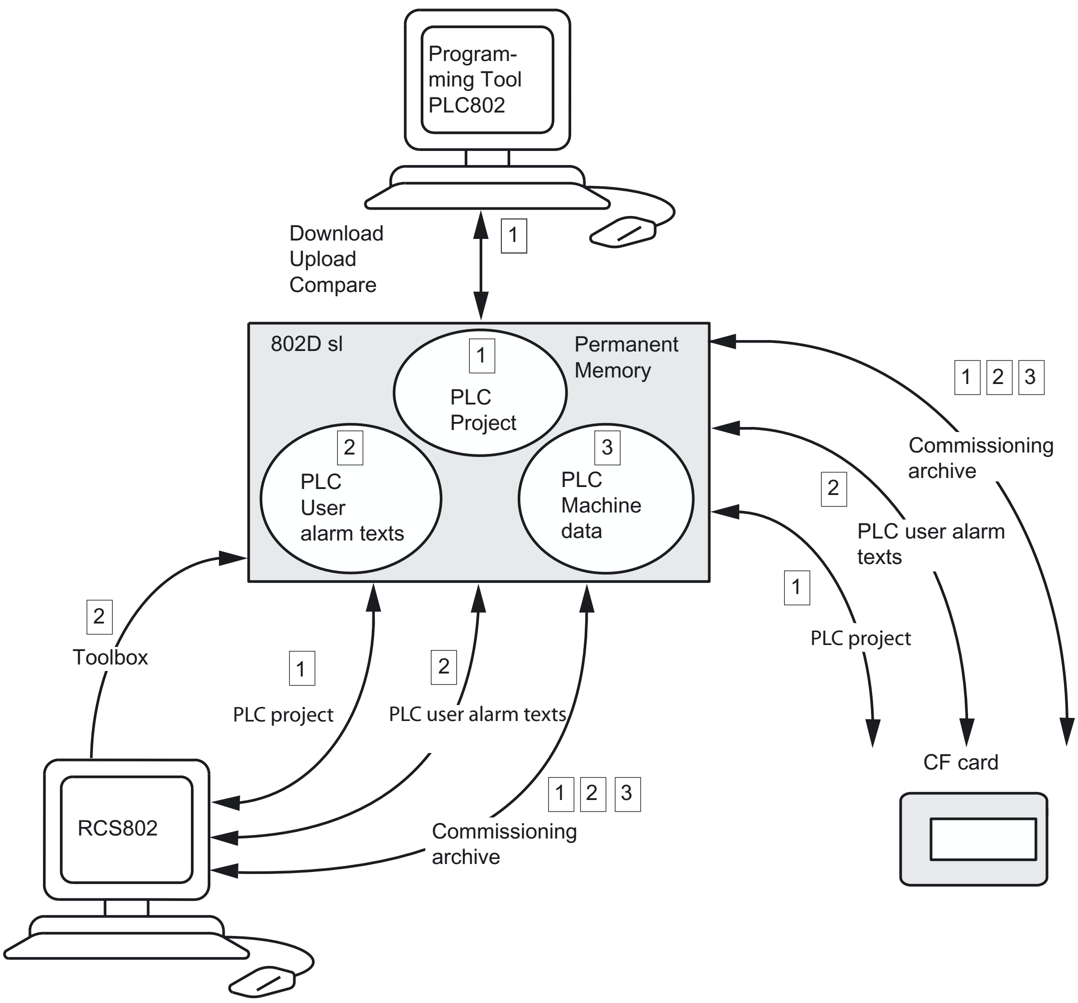


Figure 9-15 PLC applications in the control system

Download

This function writes the transferred data into the permanent memory (load memory) of the control system.

- Download PLC project using the PLC802 programming tool.
- Download (<System> operating area > "Commissioning files" > "802D data" "Commissioning archive (NC/PLC)") using the RCS802 tool or the CompactFlash card.
  For example, for the CompactFlash card:
  At this point, the commissioning archive of the customer's CompactFlash card must be copied and pasted to the Commissioning files directory > Commissioning archive (NC/PLC).
  – NC data
  – NC directories
  – Display machine data
  – Leadscrew error compensation data (LEC)
  – PLC user program
  – PLC user alarm texts
  – Drive machine data
- Read in PLC project with the RCS802 tool or CompactFlash card
- Read in PLC user alarm texts with the RCS802 tool or CompactFlash card
- Transfer the PLC user alarm texts from the Toolbox project using the RCS802 tool.

Upload

The PLC project can be saved from the permanent memory of the control system using the PLC802 programming tool, the RCS802 tool or a CompactFlash card.

- Upload PLC project using the PLC802 programming tool
- Upload (<SYSTEM> operating area > "Startup files" > "802D data" "Startup archive (NC/PLC)") with the RCS802 tool (PLC machine data, PLC project and user alarm texts) or CompactFlash Card
  Note: PLC machine data is part of the general machine data.
- Read out PLC project using the RCS802 tool or copy to a CompactFlash card.
- Read out PLC user alarm texts using the RCS802 tool

Compare

The project in the PLC802 programming tool is compared to the project in the permanent memory (load memory) in the control system.

Version display

Call up via the <System> operating area > "Service display" > "Version"

- PLC Application
  The transferred project that is active in the working memory of the PLC after a control system start up.

  In the PLC802 programming tool in the comments of the OB1 properties, programmers can use the beginning of the first line of comments to add their own additional information in the version display.

9.8 User interface

This interface encompasses all of the signals between the NCK/PLC and the HMI/PLC. In addition, the PLC decodes the auxiliary functions commands for simple further processing in the user program.

References

SINUMERIK 802D sl Function Manual

Commissioning the drives via HMI

10

10.1 Introduction to SINAMICS commissioning

Introduction

A selection menu is available via the HMI for commissioning the SINAMICS drives.

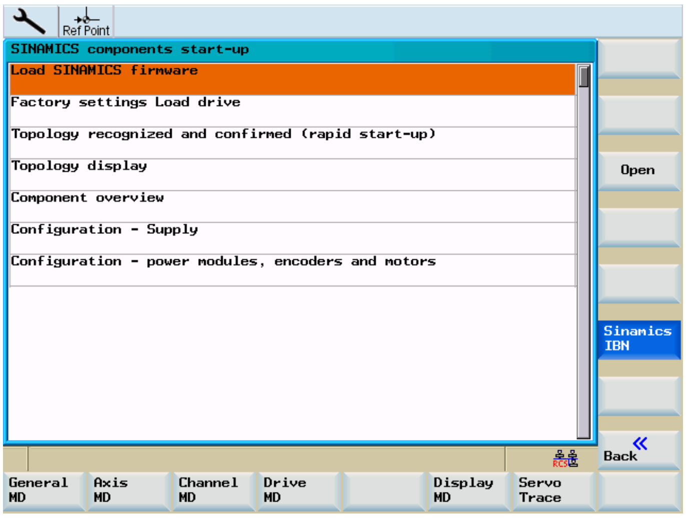


Figure 10-1 "SINAMICS Commissioning" main screen

Commissioning sequence

Carry out the first commissioning of the SINAMICS drives in the following sequence:

1. Load the SINAMICS firmware.
2. If commissioning has already taken place, load the factory settings to the drive.
3. Quick commissioning via topology recognition and confirmation
4. Topology display
5. Component overview
6. Configuration - Supply
7. Configuration - Power units, encoders and motors

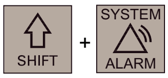
Requirements

All drive components can be contacted from the PCU (connected via DRIVE-CLiQ).

Note

Prior to starting commissioning, switch off all drive enables.

Operating sequence



You are now in the <SYSTEM> operating area.

Machine data

Press the softkeys "Machine data"> "Drive MD".

Drive MD

The main screen for the drive machine data opens.

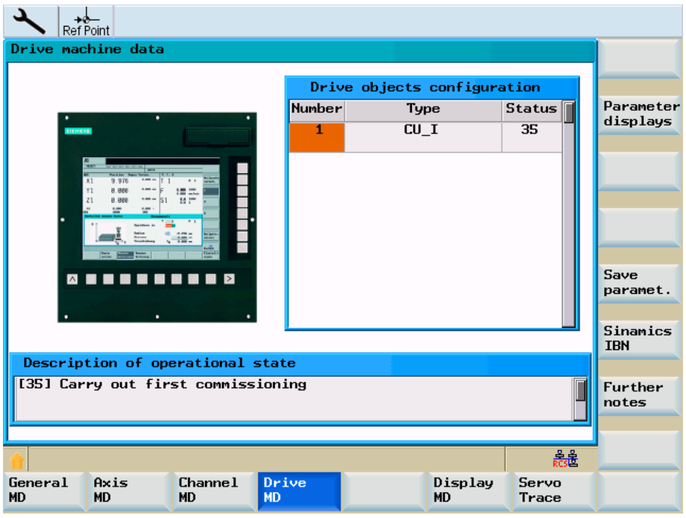


Figure 10-2 Drive machine data (factory settings loaded)

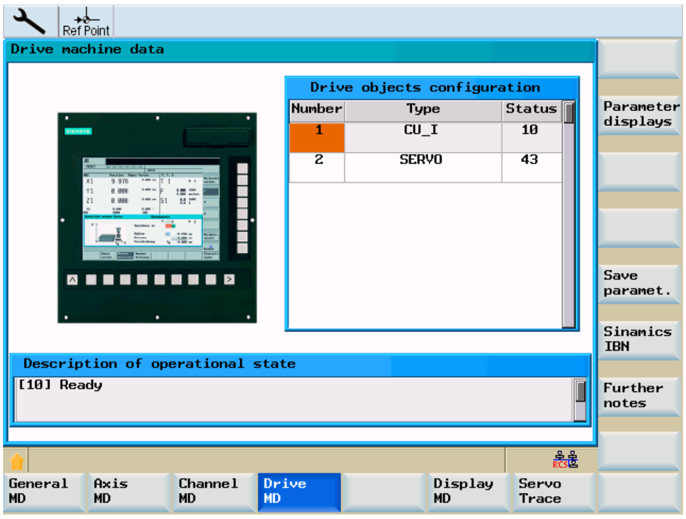


Figure 10-3 Drive machine data, commissioning carried out

The current configuration and the statuses of the control and infeed unit and the drive units are displayed.

Use this main screen for the drive machine data to start commissioning for the SINAMICS drives.

Note

If a password has been set (at least "CUSTOMER"), the "Sinamics IBN" softkey appears, via which you can access the commissioning area.

Sinamics IBN

Press "Sinamics IBN".

The "SINAMICS components commissioning" main screen appears.

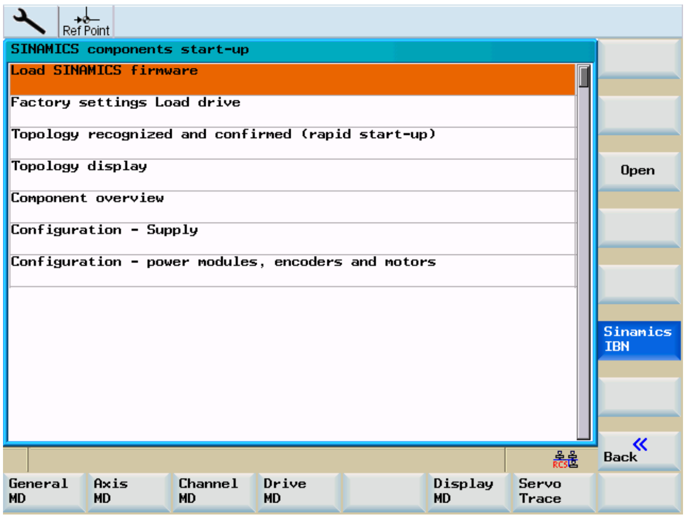


Figure 10-4 "SINAMICS Commissioning" main screen

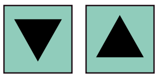
This main screen shows a selection menu with the individual commissioning steps necessary for the SINAMICS drive components.

Note

Carry out the individual steps in the following order.

For example, without a "Topology display" it is not possible to create a component overview.

The basic requirement is the successful completion of "Topology recognition and confirmation".



Use the arrow keys to select the steps for commissioning drive components.

Open the selected step.

The selected area is displayed.

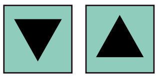
Note

In the following chapters, the individual areas and their properties will be described.

Press "Back" to return to the previous display.

10.2 Load SINAMICS firmware

Operating sequence



Open

You have selected and opened the commissioning step "Load SINAMICS firmware".

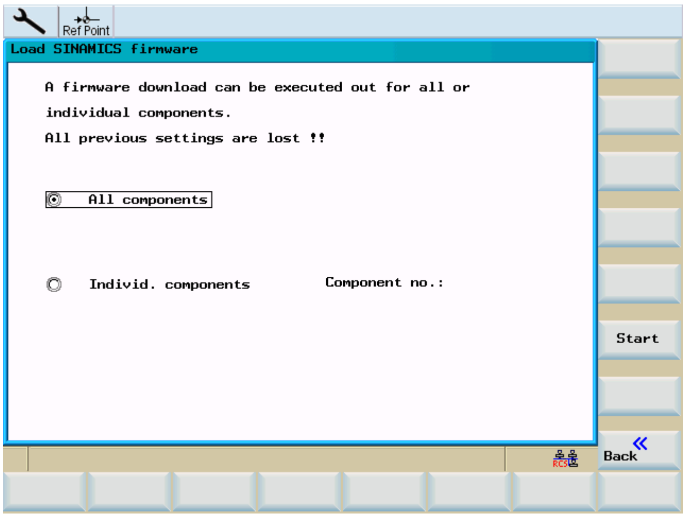


Figure 10-5 Load SINAMICS firmware

Note

The firmware upgrade is an important prerequisite for subsequent commissioning steps.

It is NOT necessary to perform topology recognition for this purpose.

You can load the SINAMICS firmware for all components or just for individual components.

You must enter the component number in "Individ. components".

Note

All previous settings are lost!

Start

Press "Start".

The HMI shows a progress bar in the window so you can track the progress of the download.

The following message is displayed:

"Please wait, download in progress"

“Notice! Do not switch off!"

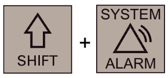
Note

Do not switch off the control during the download.

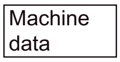
The firmware upgrade is finished. "Back" reappears.

The message "Download completed successfully!" is displayed. After this procedure the SINAMICS needs a Power OFF/ON".

Switch the drive off (cut the power) and then on again.

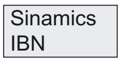


When the control has booted, press <SHIFT> + <SYSTEM>.



Press "Machine data".

Press "Drive MD"



Press "Sinamics IBN".

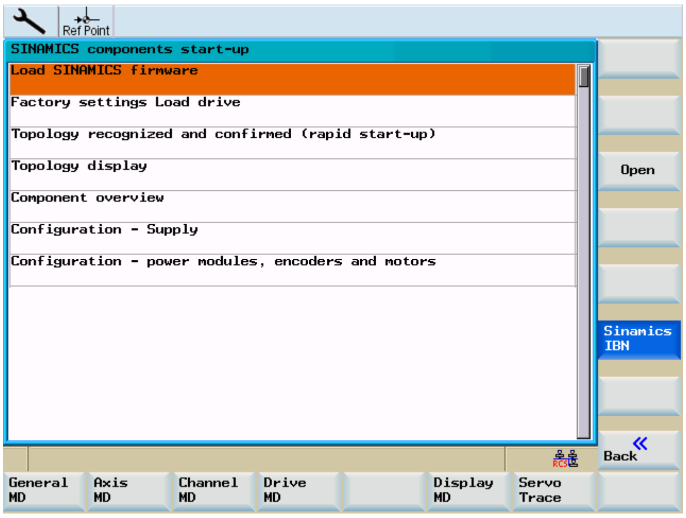
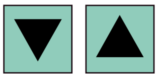


Figure 10-6 "SINAMICS Commissioning" main screen

Continue the first commissioning with the step "Load drive factory settings".

10.3 Load factory settings for the drive (Parameter Reset)

Operating sequence



Open

You have selected and opened the commissioning step "Load factory settings for the drive".

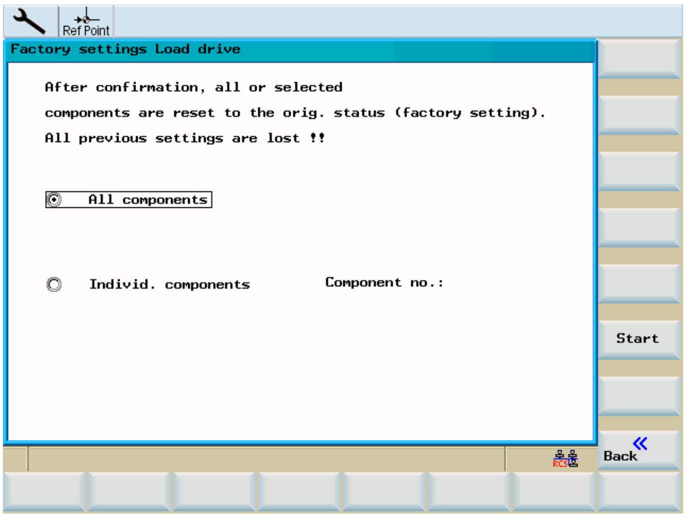


Figure 10-7 Load factory settings for the drive

You can load the factory settings for all components or just for individual components.

You must enter the component number in "Individ. components".

Note

All previous settings are lost!

Start

Press "Start".

The HMI shows a progress bar in the window so you can track the progress of the download.

Successful completion is indicated in the lower part of the window in the form of informational text.

<< Back

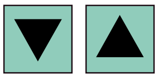
Press "Back".

The selection menu fur commissioning is shown.

Continue the first commissioning with the step "Topology recognition and confirmation".

10.4 Topology recognition and confirmation

Operating sequence



Open

You have selected and opened the commissioning step "Topology recognition and confirmation".

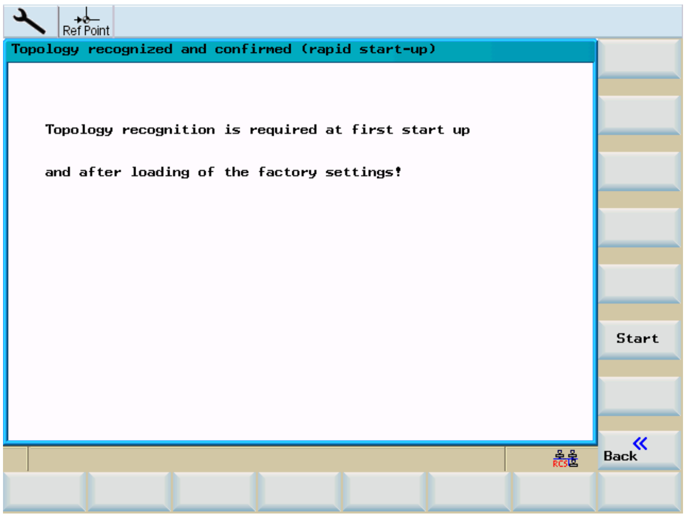


Figure 10-8 Quick commissioning

Note

Topology recognition is necessary during first commissioning and after loading the factory settings!

First commissioning

Start

Press "Start".

The HMI shows a progress bar in the window so that you can track the commissioning progress.

The following steps are carried out for topology recognition:

1. The actual topology of the device (p0098[0]) is read-out and automatically entered in the parameter of the device setpoint topology (p0099[0]).
2. Start the quick commissioning of the SINAMICS drive components.

   The PROFIBUS protocol and the BICO wiring is entered in every drive object.

   When this has ended, the parameter p0978[x] is automatically adapted to the existing configuration.
3. The configuration is saved.

The currently running procedure is documented in the lower part of the window in the form of a short text.

<< Back

The procedure is finished. "Back" reappears.

The prompt "After completion of drive commissioning, a Power OFF/ON is required!" is displayed

Startup protocol

Note

If topology recognition were not to run, use the "Startup protocol" function to save the startup protocol.

Please contact the hotline with this information (see the "Technical Support"section in the preface).

Topology recognition was successfully completed.

Switch the control and drive off (cut the power) and then on again.

SHIFT + SYSTEM ALARM

When the control has booted, press <SHIFT> + <SYSTEM>.

Machine data

Press "Machine data".

Drive MD

Press "Drive MD"

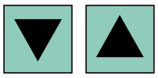
Sinamics IBN

Press "Sinamics IBN".

Continue the first commissioning with the step "Topology display".

10.5 Topology - display

Topology - display



Open

You have selected and opened the commissioning step "Topology display".

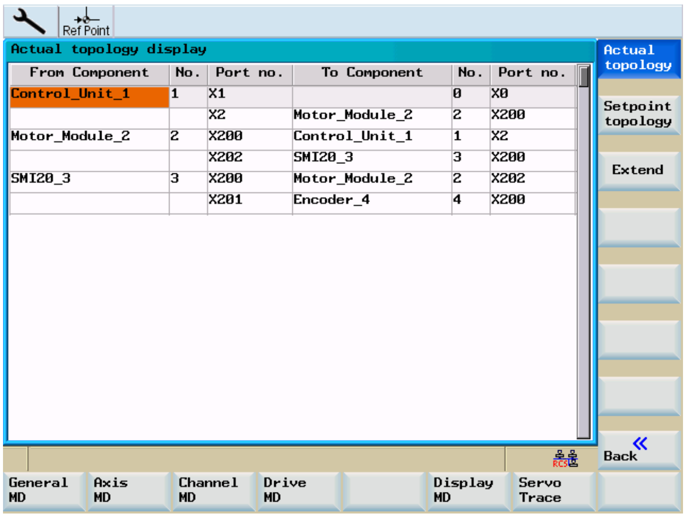


Figure 10-9 Topology Display

The HMI displays a progress bar while the data is being read in and prepared.

For repeated call-up, the screen appears immediately.

The evaluated data is retained until "Power OFF".

The following is displayed:

Actual topology

"Actual topology" of the DRIVE-CLiQ wiring on the drive system.

Note

In the topology view, you can also see the SMC encoder component number, which you commission in step Configuration - Encoders (Page 351).

-> Please make a note of this component number.

Setpoint topology

When you connect a new component (e.g. SMC20) to the drive system via DRIVE-CLiQ, then SINAMICS recognizes the change in the actual topology.

The difference between the setpoint/actual topology is sent to the HMI.

Extended

Further connected components are displayed.

<< Back

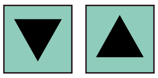
Press "Back".

The selection menu fur commissioning is shown.

Continue commissioning with the step " Component overview".

10.6 Component overview

Operating sequence



Open

You have selected and opened the commissioning step "Component overview".

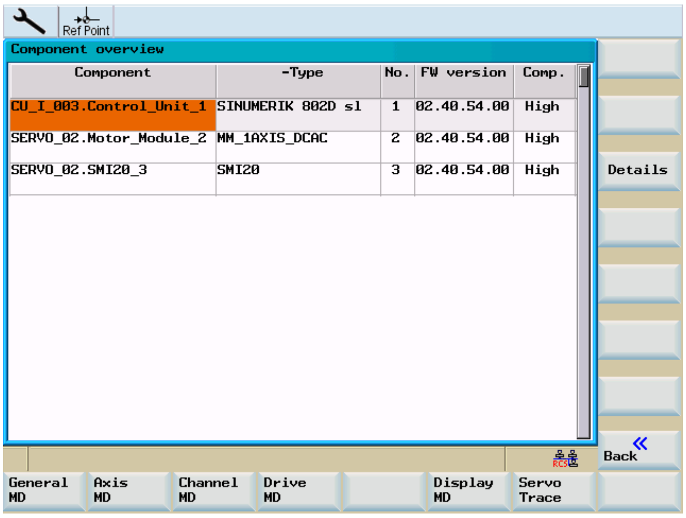


Figure 10-10 Component overview

The component overview is for information only.

The following is shown in the component overview:

- Component name
- Type
- Number
- Firmware versions of all components
- Topology comparison step

Details

Press "Details".

A window containing additional information on the selected components appears.

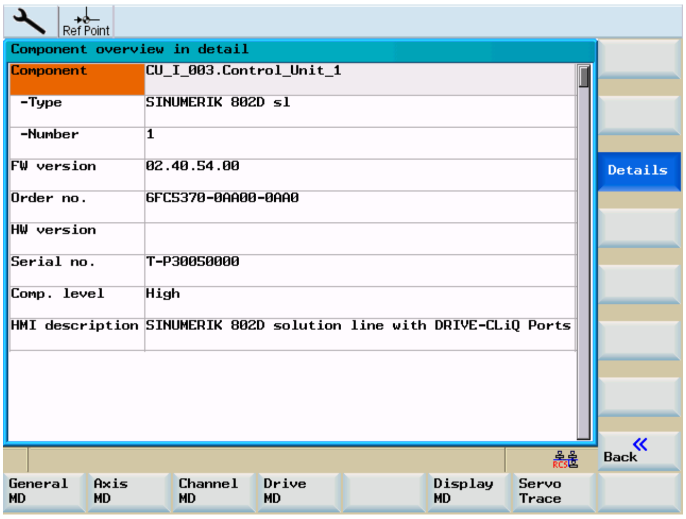


Figure 10-11 further details on the components

Use "<<Back" to return to the previous display.

Press "Back".

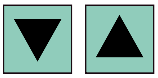
The selection menu fur commissioning is shown.

Continue the first commissioning with the step "Configuration of supply".

10.7 Configuration of supply

10.7.1 Configuration of supply

Operating sequence



Open

You have selected and opened the commissioning step "Configuration - Supply".

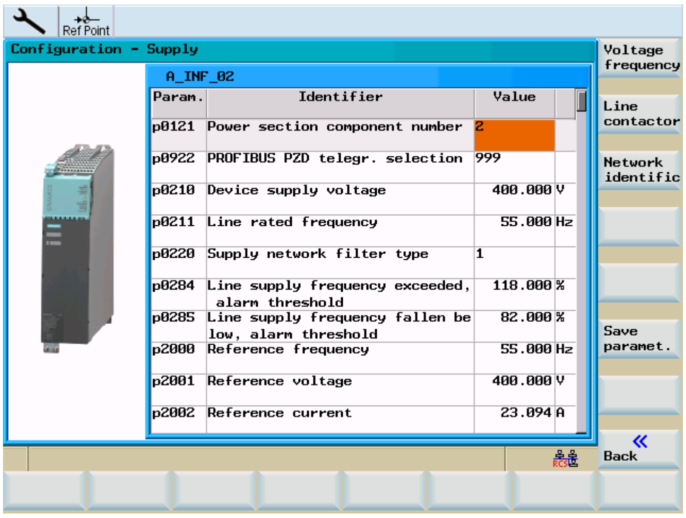


Figure 10-12 Configuration of supply

The current values are displayed in the "Configuration - Supply" window.

Enter new values if necessary.

Save

The configuration is completed with "Save".

<<
Back

Use "Back" to switch back to the selection menu for startup.

Continue the first commissioning with the step "Configuration - Power units, encoders and motors".

Softkeys

Voltage Frequency

The following values can be changed for "Country-specific adaptation of line voltage and frequency":

- Device supply voltage
- Rated line frequency

Line contactor

Activation of parameters for operating a line contactor

Network identific

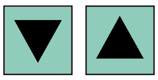
The supply is identified automatically. This involves optimizing control within the supply.

Note

Identification can only be performed after the control system and the drive have been put into operation.

10.7.2 Voltage/Frequency configuration

Operating sequence



Open

You have selected and opened the startup step "Configuration - Supply".

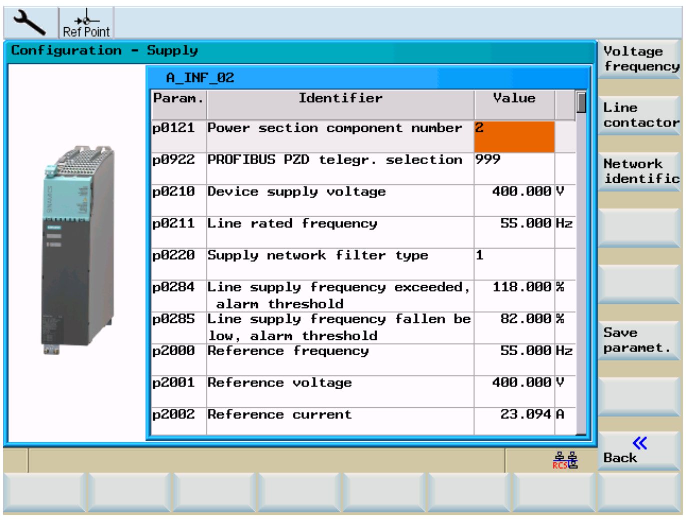


Figure 10-13 Configuration of supply

Voltage Frequency

Use "Voltage Frequency" to open the following window:

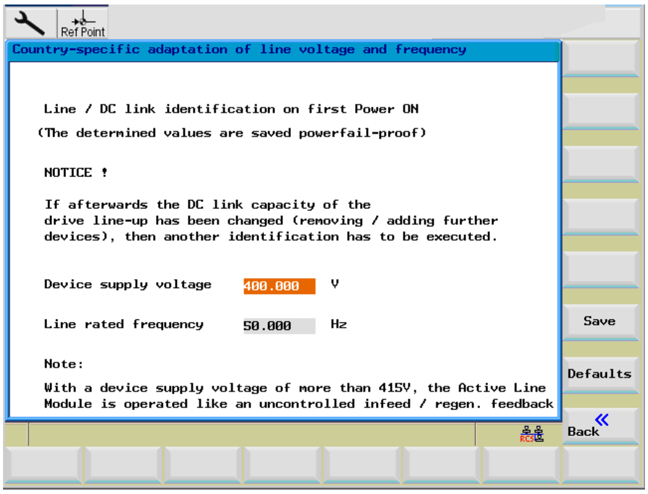


Figure 10-14 Voltage Frequency

The following values can be changed in the window "Country-specific adaptation of line voltage and frequency":

- Device supply voltage
- Rated line frequency

Default values

Enter defaults.

Save

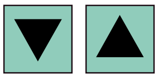
Save changed values.

<< Back

Use "Back" to switch back to the supply configuration window.

10.7.3 Line contactor configuration

Operating sequence



Open

You have selected and opened the startup step "Configuration - Supply".

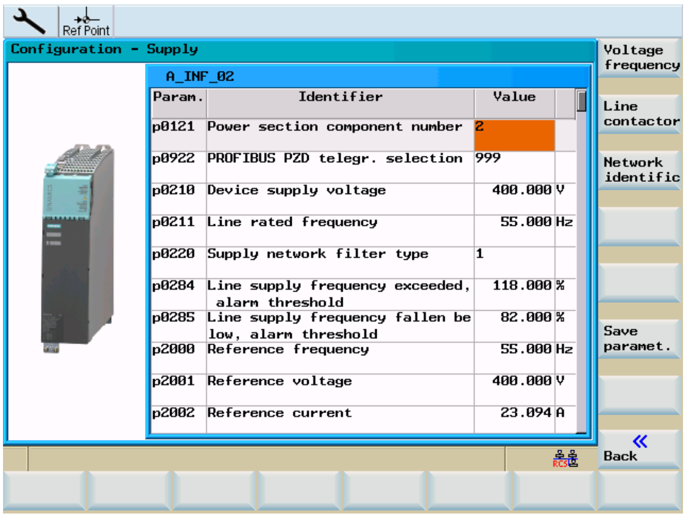


Figure 10-15 Configuration of supply

Line contactor

Use "Line contactor" to activate the parameter settings for operating a line contactor.

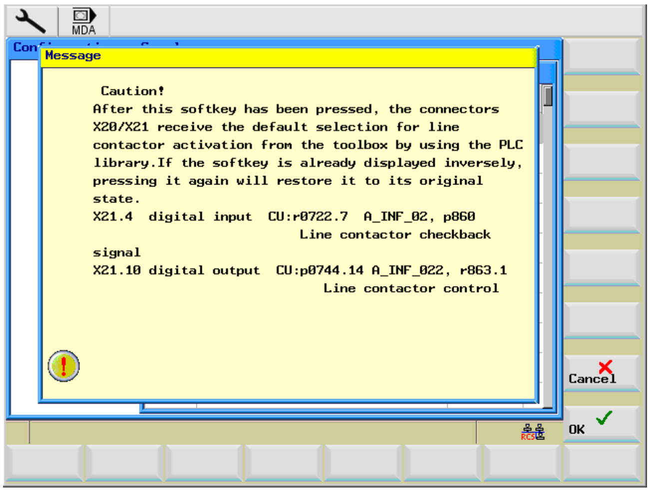
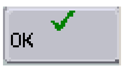


Figure 10-16 Line contactor message text



Confirm the message text for the line contactor settings with "OK".

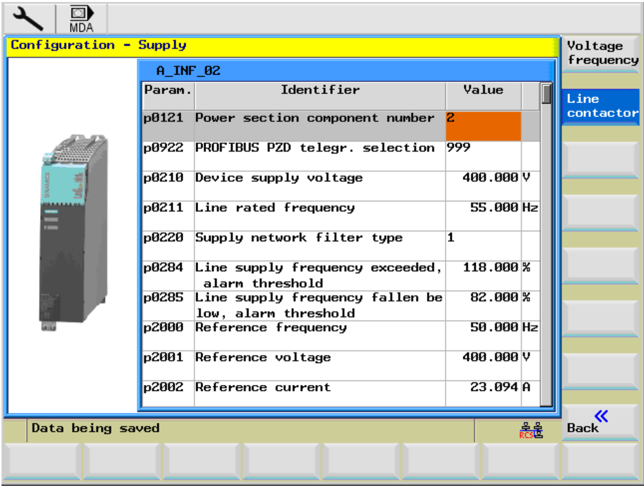


Figure 10-17 The data is saved.

The data is being saved.

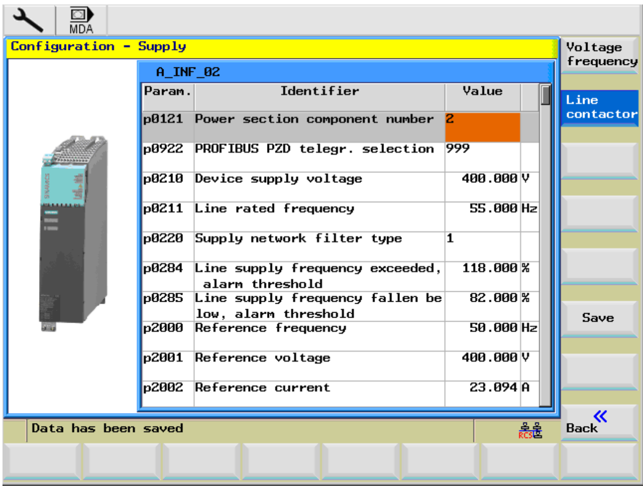


Figure 10-18 Data has been saved.

Data saving has has been completed.

Line contactor

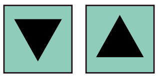
The line contactor has been activated. The "Line contactor" softkey has a colored background.

Note

Pressing the "Line contactor" softkey again deactivates the line contactor once more.

10.7.4 Network identification configuration

Operating sequence



Open

You have selected and opened the startup step "Configuration - Supply".

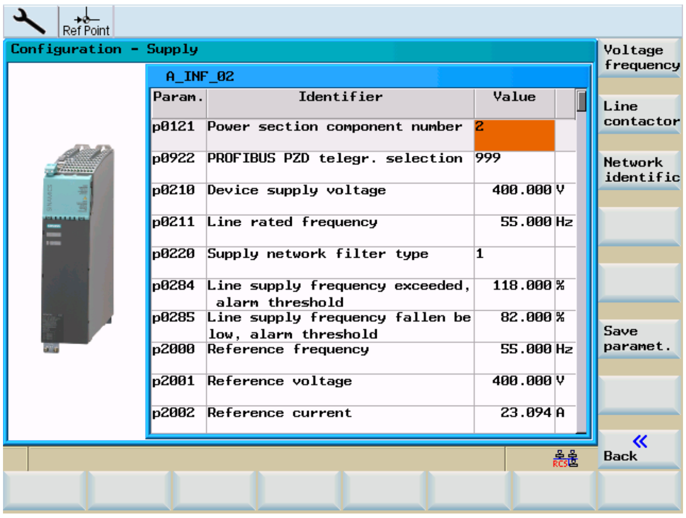


Figure 10-19 Configuration of supply

Network identific

Select "Network identific" to activate automatic identification of the supply.

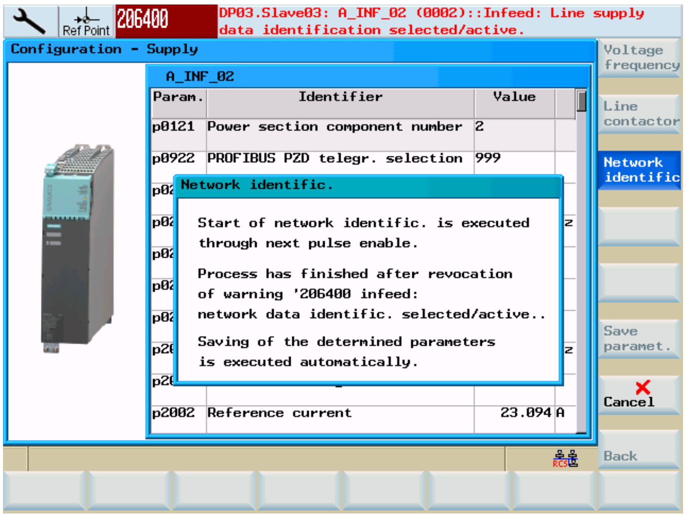


Figure 10-20 Network identification

Please take the following note into account:

"Start of network identific. is executed on next pulse enable ..."

The control within the supply is optimized automatically.

For example, the inductance and capacity of the DC link are determined, as well as the optimum control data for the step-up converter.

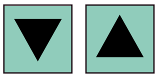
Note

Identification can only be performed after the control system and the drive have been put into operation.

Network identification is executed on a pulse enable.

10.8 Configuration - Power units, encoders and motors

Operating sequence



Open

You have selected and opened the commissioning step "Configuration - Power units, encoders and motors".

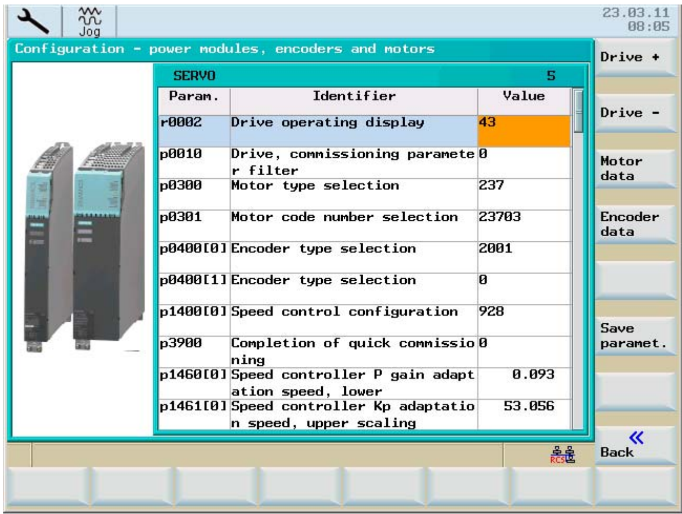


Figure 10-21 Configuration - Power units, encoders and motors

In the "Configuration - Power units, encoders and motors" window, the current values of each of the (drive) components are displayed.

Drive +

Drive -

Press "Drive +" and "Drive -".

The values for the individual power units (SERVOs) are shown.

Configuration options in the list displayed (see previous diagram)

If the connected motor is not an SMI motor, but a standard motor connected via SMC, then there are the following options in this list:

1. Enter the appropriate values in parameters p0300 (motor type selection) and p0301 (motor code number selection).

   These values can be found using the motor MLFB via STARTER /Hotline/List Manual SINAMICS S120.

   Example:

   Motor type = "237" = p0300

   Motor code number = "23706" = p0301
2. The motor data is preassigned by writing parameter p0301.

   Note

   For a short period the HMI is not operable. Wait until the HMI reacts to Cursor Up.

   ---
3. Conclude commissioning with parameter p3900 = "3".

   The motor, closed-loop and open-loop control data are definitively preassigned by writing parameter p3900.

   Note

   For a short period the HMI is not operable. Wait until the HMI reacts to Cursor Up.

Save parameters

4. Press "Save parameter".

   The data will be saved.

Softkeys

Motor data

The following motor types are to be configured:

- Standard motors, listed with corresponding motor data.
- Third-party motors where the motor data can be freely configured.

Encoder data

The following encoder data is to be configured:

- SMI (Sensor Module Integrated)
- SMC (Sensor Module Cabinet)

Save parameters

The entries are saved.

<< Back

Press "Back" to return to the selection menu.

10.8.1 Configuration - Encoders

Operating sequence

Encoder data

The "Encoder data" softkey opens a window for configuring the encoders.

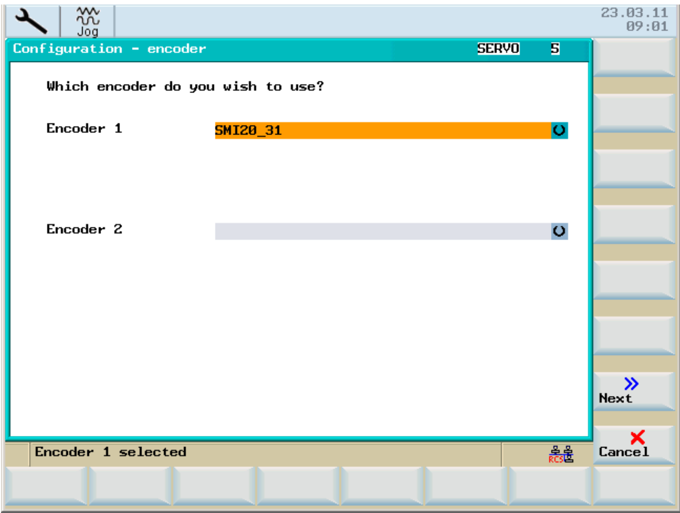
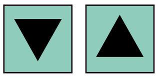


Figure 10-22 Configuration - encoder

The configuration of the first encoder was detected as SMI encoder in the encoder configuration.



Since in this example, the second encoder is an SMC encoder to be configured, use the arrow keys to select "Encoder 2".

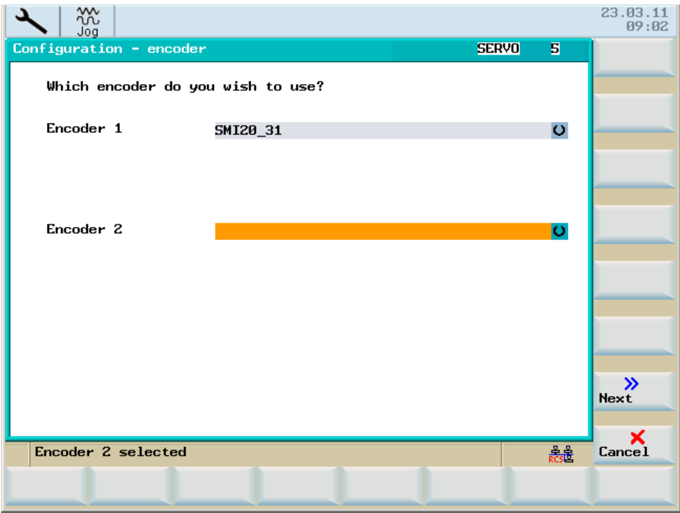


Figure 10-23 Encoder selection

Use the <SELECT> key to select the encoder interface for component selection.

Note

You can find the encoder interface and component number assignment for the current encoder in the topology overview (Page 340).

Using a placeholder (can be selected using the <SELECT> key) you have the possibility of deselecting the encoder assignment.

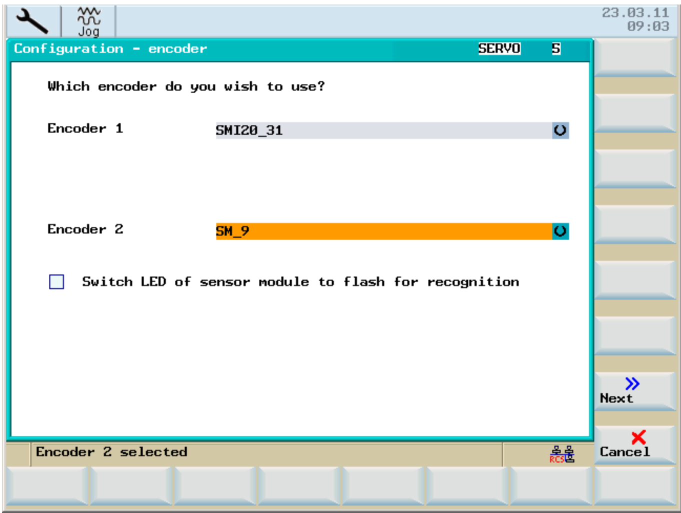


Figure 10-24 Encoder selected

Choose "Switch LED of sensor module to flash for recognition".

With the "Switch LED of sensor module to flash for recognition", at the sensor modules you can see which encoder is to be configured.

The SMC module flashes.

If you have an SMI (a Sensor Module inserted into the flange socket of the motor), then this check box is irrelevant.

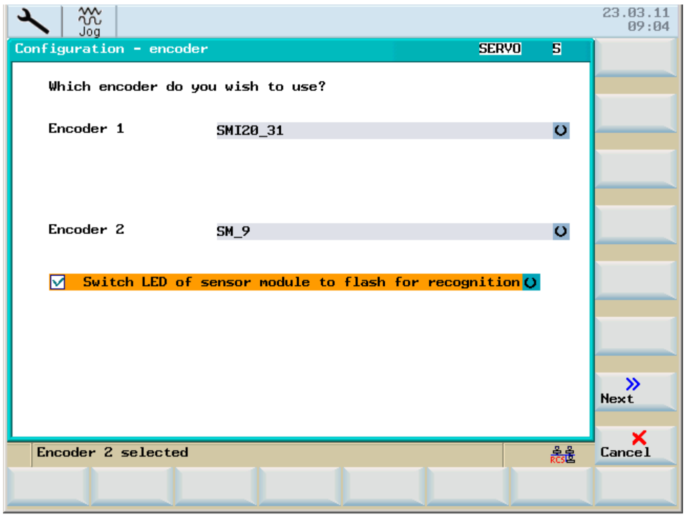


Figure 10-25 Check the encoder interface (LED detection)

If the cursor is on "Encoder 2" or "Switch LED of sensor module to flash for recognition", you can go on to configure the second encoder with ">> Next".

Another window opens for configuring the selected second encoder.

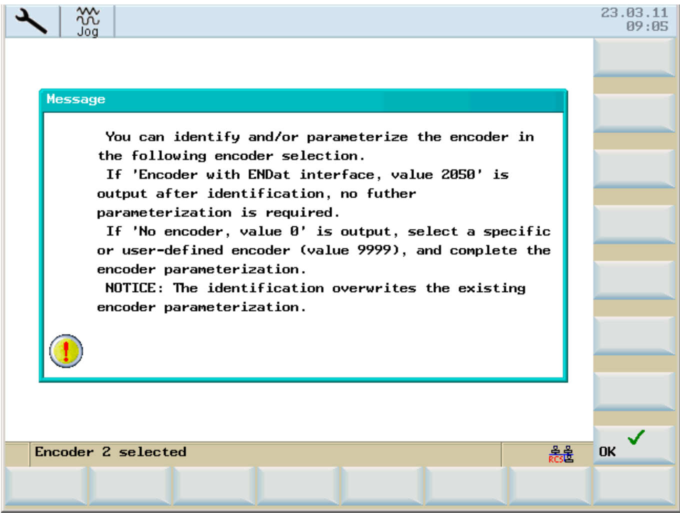
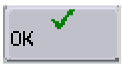


Figure 10-26 Encoder data message



Press "OK" to confirm the message.

Note

If a second encoder is used, the following must be taken into consideration during **first commissioning**:

The encoder data set (EDS) for the second encoder is preassigned with the first encoder's encoder data set.

In this case, you can identify the encoder data of the second encoder using the "Identification" softkey.

Identifica-
tion

Press "Identification" (see screenshot below).

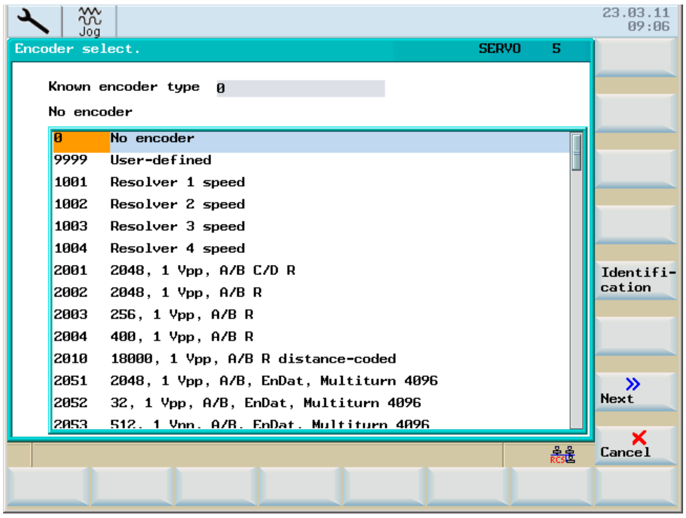
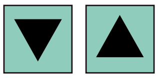


Figure 10-27 Encoder type

With the entry into the encoder selection, "detected encoder type" - >"0" is output in the header line. The cause is the incremental encoder (SMC), which is not detected.



From the list displayed, you can select the required encoder -> in the example 2002 (see screenshot below).

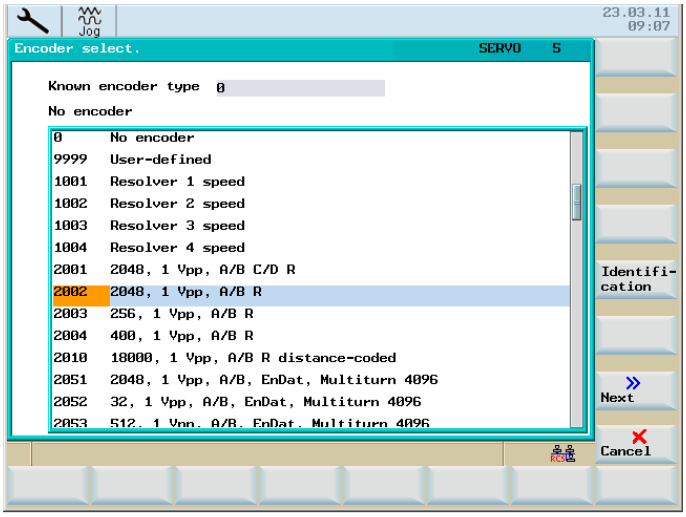


Figure 10-28 Selection of encoder 2002

If the required encoder is not listed, then the encoder can be parameterized user-defined ("9999 user-defined").

Next

If you have selected the encoder, press ">> Next".

You can configure this encoder in the "Parameterization - encoder" window (see screenshot below).

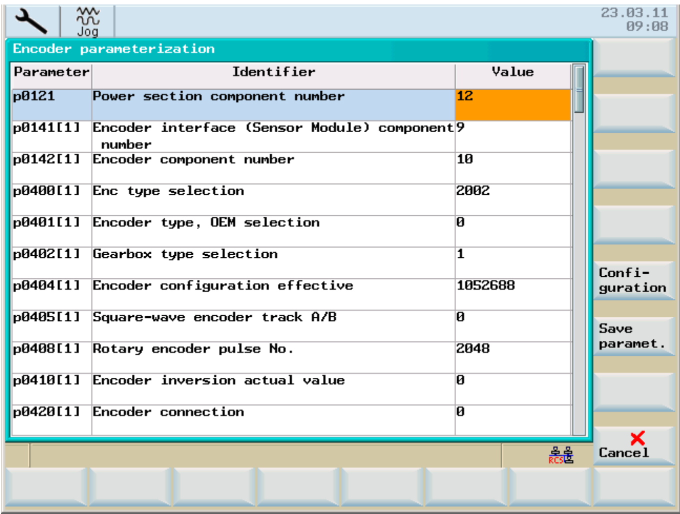


Figure 10-29 Configuration - "Parameterization - encoder"

For encoders listed in the catalog, the speed and position actual values can be inverted in the parameterization of the encoder using p0140, where required.

If the configuration is not yet complete, new values can be entered.

Configura-tion

The encoder configuration can be checked using the "Configuration" softkey.

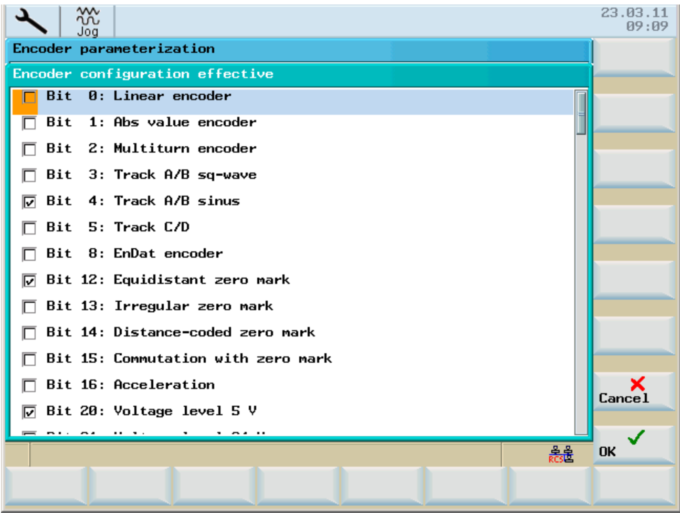


Figure 10-30 Bit-by-bit encoder configuration

Press "Cancel" or "OK".

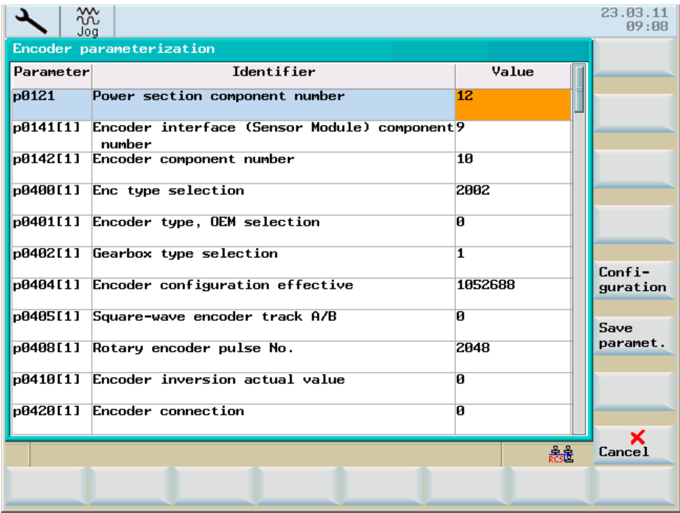


Figure 10-31 Configuration - "Parameterization - encoder"

Save parameters

Press "Save parameter" to save the configuration.

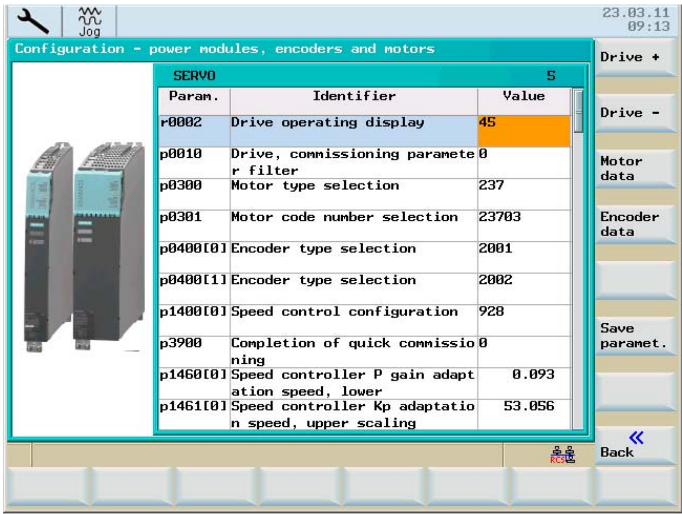


Figure 10-32 Encoder configuration saved

You are back again in the starting menu for configuration - power units, encoders and motors.

You configure the motor data in the next step.

10.8.2 Configuration - Motors

Operating sequence

Motor data

The "Motor data" softkey opens a window containing the motor data.

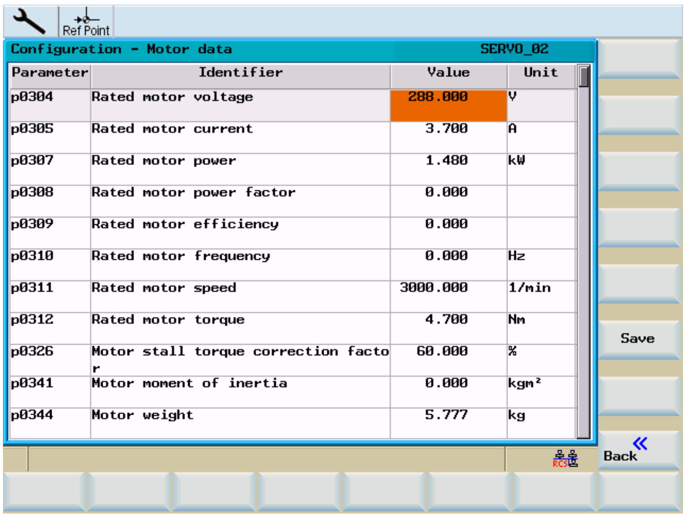


Figure 10-33 Motor data

You can look at the motor data and make changes if necessary.

Save

Use "Save" to save data that has been changed.

The main screen for configuring the power units, encoders and motors is opened.

<< Back

Use "<<Back" to return to the main screen for configuring the power units, encoders and motors.

10.9 First commissioning of the drive completed

First commissioning of the drive completed

After you have carried out to the individual steps for first commissioning, when CU_I is selected the HMI displays the operating state "Ready for operation".

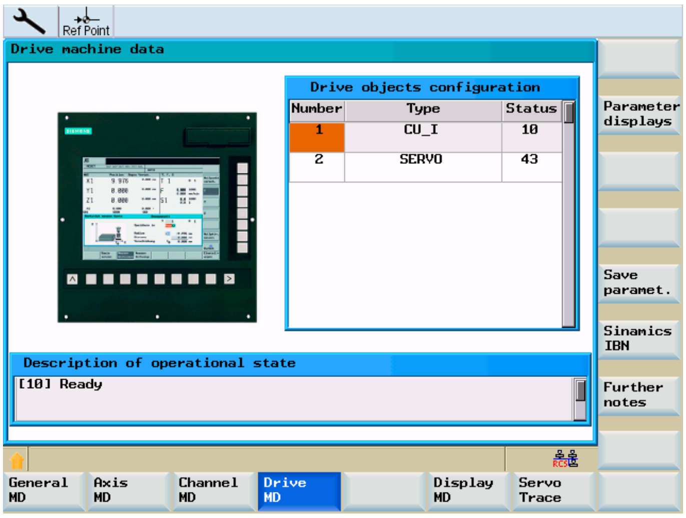


Figure 10-34 Drive machine data

Note

The status display in the window "Configuration of drive objects" is derived from parameter r0002 of the corresponding type.

Reference

SINAMICS S120 List Manual

10.10 Terminal assignment X20 / X21

Introduction

The tables below describe the configuration of digital inputs and outputs on the PCU following SINAMICS S120 commissioning using the HMI.

Table 10- 1 Configuration of terminal X20

Pin no.	Function	Assignment	BiCo source	BiCo sink	Macro no.
1	Input 0/1 edge required	ON/OFF 1 Infeed Line Module **with** DRIVE-CLiQ Connection	CU: r0722.0	Infeed p840	150001
		"Infeed Ready Signal" of Line Module **without** DRIVE-CLiQ connection	SLM X21.1	Drive p864	150005
2	Input	"OFF3 – rapid stop" Function: Braking with configurable OFF3 ramp (p1135)1) then pulse suppression and closing lockout. The drive is stopped by prompts. The braking behavior can be separately set for each servo. Behavior similar to that of terminal 64.	CU: r0722.1	Each drive 2. OFF3, p849	150001 150005
3	Input				No preassignment
4	Input				No preassignment
5		Ground for pin 1 .. 4			
6		24 P			
7	Output				No preassignment
8	Output				No preassignment
9		Ground for pins 7, 8, 10, 11			
10	Input	Bero 1 – zero mark substitute	CU: r0722.10	p495 = 2	--
11	Input	Probe 1 Decentralized Measuring (check that MD13210 = 1!)	CU: p0680[0] = 0	Every drive p488 Index = encoder 1,2,3 = 3	--
12		Ground for pins 7, 8, 10, 11			

1) In conjunction with the drive MD p1135, the following axis MDs should also be dealt with:

- MD36610 $MA_AX_EMERGENCY_STOP_TIME
- MD36620 $MA_SERVO_DISABLE_DELAY_TIME

Table 10- 2 Configuration of terminal X21

Pin no.	Function	Assignment	BiCo source	BiCo sink	Macro no.
1	Input	Digital input $A_IN[1]	CU: r0722.4	CU: p2082[0]	150001 150005
2	Input	Digital input $A_IN[2]	CU: r0722.5	CU: p2082[1]	
3	Input	Digital input $A_IN[3]	CU: r0722.6	CU: p2082[2]	
4	Input	Digital input $A_IN[4]	CU: r0722.7	CU: p2082[3]	
		Line contactor, feedback signal		LM : p0860	--
5		Ground for pin 1 .. 4			
6		24 P			
7	Output	Infeed Operation (Line Module with DRIVE-CLiQ Connection)	LM : r0863.0	CU: p0742	150001
		Digital output $A_OUT[4]	CU: p2091.3		150005
8	Output	Infeed and operational readiness if Line Module with DRIVE-CLiQ connection	LM : r0899.0	CU: p0743	150001
		Digital output $A_OUT[3]	CU: p2091.2		150005
9		Ground for pins 7, 8, 10, 11			
10	Output	Digital output $A_OUT[2]	CU: p2091.1	CU: p0744	150001 / 150005
		Line contactor control	LM : r0863.1		--
	Input	Bero 2 – zero mark substitute	CU: r0722.14	Drive: p0495=5	--
		2. OFF 2	CU: r0722.14	Drive: p0845	--
11	Output	Digital output $A_OUT[1]	CU p2091.0	CU: p0745	150001 / 150005
	Input	Probe 2 Decentralized Measuring (check that MD13210 = 1!)	CU: p0680[1]=0 CU: p0728 Bit 15=0	each drive p489 Index = encoder 1,2,3 = 6	
12		Ground for pins 7, 8, 10, 11			

Drive optimization using the startup tool 11

11.1 Software and hardware prerequisites

Requirements

On the PC/PG, the following prerequisites are necessary for commissioning the SINUMERIK 802D sl using the startup tool:

- Connections to the 802D sl PCU using the following options:
  - Network connection

    using a patch cable via a fixed network.
  - Ethernet peer-to-peer connection of the PG/PC to female connector X5 on the control system,

    using a Crosslink patch cable X.
- Software prerequisites
  - You have installed the startup tool on the PG/PC and have started it.
  - To communicate with the 802D sl PCU, enter the IP address of the server by accessing the following menu items:

    "Commissioning" > "HMI" > "NCU connection".

    The screenshot below shows an example:

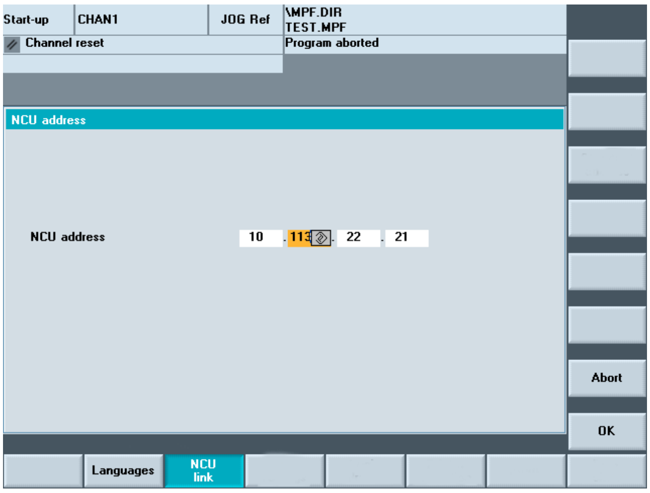


Figure 11-1 Startup tool -> 802D sl PCU IP address

- IP address of the PCU (NCU)

Note

The IP address of the 802D sl PCU depends on the type of connection between the PG/PC and 802D sl PCU.

– Network connection -> IP address, see on the control under operating area <SYSTEM> "Service display" > "Service control" > "Service network" (see the following figure).

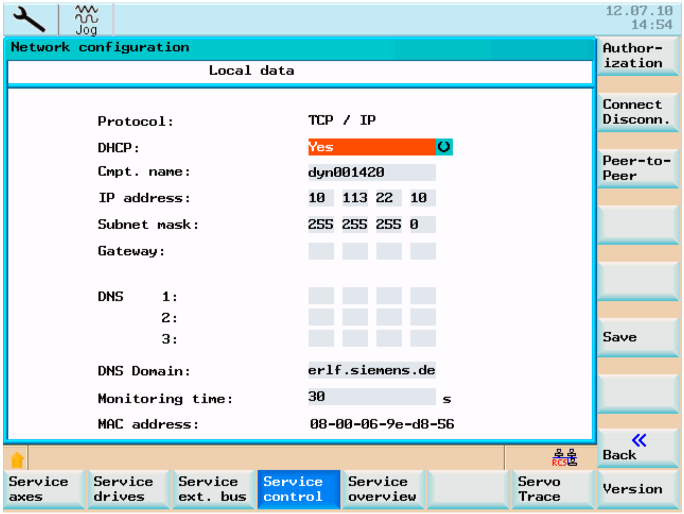


Figure 11-2 802 D sl control system -> PCU IP address

Note

For a network connection, the "Direct connect." vertical softkey must be deactivated on the control in the <SYSTEM> operating area > "Service display" > "Service control."

You can access this area from the screen shown above by pressing the vertical softkey "<< Back".

– Ethernet peer-to-peer connection -> Standard IP address -> 169.254.11.22

Note

You have activated the "Direct connect." vertical softkey on the control in the <SYSTEM> operating area > "Service display" > "Service control" for an Ethernet peer-to-peer connection.

11.2 Drive optimization

11.2.1 Drive optimization (overview)

You can optimize the drives in the "Commissioning" > "Optimization/Test" operating area.

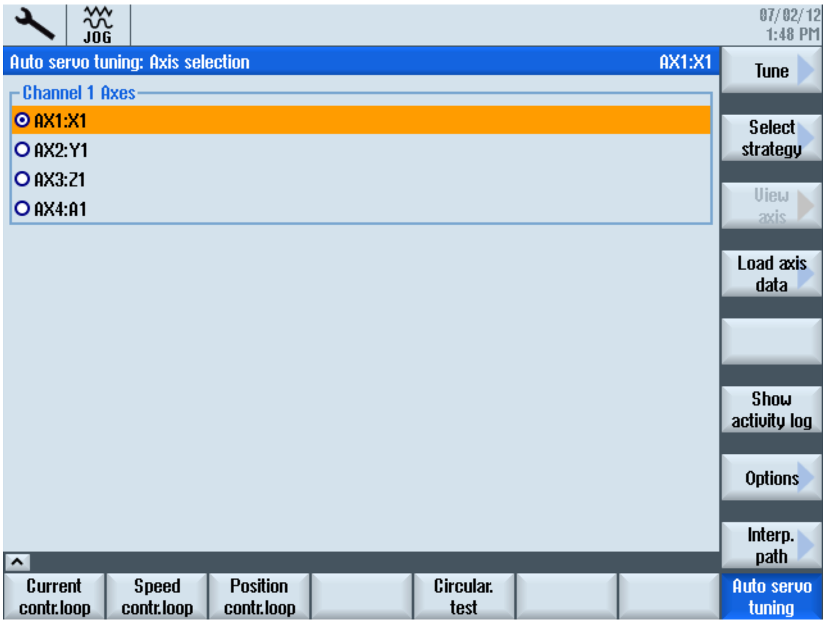


Figure 11-3 Optimization/test

The following functions are available:

- Frequency response measurements for the following control loops:
  - Current controller
  - Speed controller
  - Position controller
- Automatic controller setting
- Function generator
- Circularity test
- Trace
  - Servo trace
  - Drive trace

Measuring functions

The measuring functions make it possible to assess the automatic controller action of the respective control loop (frequency response) by the integrated FFT analysis (Fast Fourier Transformation) without external measuring equipment.

The measurement results are represented graphically as a Bode diagram. HMI file functions can be used to archive the diagrams for documentation purposes and to simplify remote diagnostics.

Circularity test

The circularity test serves to set and assess the dynamic response for interpolating axes and to analyze the contour accuracy on the quadrant transitions (circular contours) achieved by means of friction compensation (conventional quadrant error compensation).

Literature

Function Manual Extended Functions, K3 Compensation, Section "Circularity test"

Servo trace

Servo trace provides a graphically assisted analysis of the time response of position controller and drive data.

Drive trace

Drive trace provides a graphically assisted analysis of the time response of drive signals.

11.2.2 Measuring functions

Explanation of the measuring functions

A range of measuring functions allows the time and frequency response of drives and controls to be displayed in graphical form on the screen. For this purpose, test signals with an adjustable interval are connected to the drives.

Measurement/signal parameters

The test setpoints are adapted to the application in question by means of measurement or signal parameters, the units of which are determined by the relevant measuring function or operating mode. The measurement or signal parameter units are subject to the following conditions:

Table 11- 1 Quantity and units for measurement or signal parameters

Size	Unit
Velocity	Metric system: Specification in mm/min or rev/min for translatory or rotary movements Inch system: Specification in inch/min or rev/min for translatory or rotary movements
Distance	Metric system: Specification in mm or degrees for translatory or rotary movements Inch system: Specification in inch or degrees for translatory or rotary movements
Time	Specified in ms
Frequency	Specified in Hz

Note

The default setting for all parameters is 0.

Preconditions for starting measuring functions

To ensure that no erroneous traversing movements due to part programs can be carried out, the measuring functions have to be started in <JOG> mode.

NOTICE

Collision avoidance

When traversing movements are carried out within the framework of measuring functions, no software limit switches and working area limitations are monitored, since these are carried out in follow-up mode.

Prior to starting traversing movements, the user must therefore ensure that the axes are positioned such that the traversing limits specified within the framework of the measuring functions are sufficient to prevent collision with the machine.

Starting measuring functions

Measuring functions initiating a traversing movement are only selected using the specific softkey. The actual start of the measuring function and thus of the traversing movement is always carried out with <NC-START> on the machine control panel.

If the main screen of the measuring function is quitted without the traversing movement being initiated, the selection of the traversing function is canceled.

Once the traversing function has been started, the main screen can be exited without any affect on the traversing movement.

Note

<JOG> mode must be selected when measuring functions are started.

Further safety notices

The user must ensure that when the measuring functions are used:

- The <EMERGENCY STOP> button is always within reach.
- No obstacles are in the traversing range.

Canceling measuring functions

The following events will cancel active measuring functions:

- Hardware limit switch reached
- Traversing range limits exceeded
- Emergency stop
- Reset (mode group, channel)
- NC STOP
- No controller enabling command
- Canceling drive enable
- Canceling traversing enable
- Selection of parking (in position-controlled operation).
- Feed override = 0%
- Spindle override = 50%
- Change in operating mode (JOG) or operating mode JOG not selected
- Actuation of traversing keys
- Actuation of handwheel
- Alarms leading to axis shutdown

11.2.3 Frequency response measurement

11.2.3.1 Measurement of current control loop

Functionality

The current control loop only needs to be measured for diagnostic purposes if there is a fault or if there is no standard data for the motor / power unit combination (third-party motor).

NOTICE
Protection of the machine
The user must take special safety measures when measuring the current control loop (e.g. secure drive clamping) for hanging axes without external counterweight.

Operating path

Operating path for measuring the current control loop: Operating area switchover > "Commissioning" > "Optimization/Test" > "Current control loop"

Measuring functions

The following measuring functions are available for measuring the current control loop:

Measuring type	Measured variable
Reference frequency response (downstream of the current setpoint filter)	Torque-generating actual current value / torque-generating current setpoint
Setpoint step change (downstream of the current setpoint filter)	Measured variable 1: Torque-generating current setpoint Measured variable 2: Torque-generating actual current value

Measurement

The measurement sequence is divided into the following steps:

1. Setting the traverse range monitoring and the enable logic
2. Selecting the measurement type

3. Setting the parameters, softkey: "Measuring parameters"
4. Displaying the measurement results, softkey: "Display"

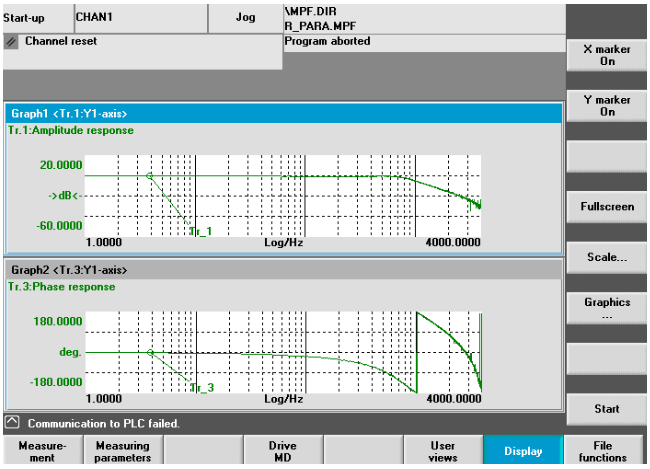


Figure 11-4 Current controller

Measuring parameters

- Amplitude

  Magnitude of the test signal amplitude. Given in percent of the peak torque. Values from 1% to 5% are suitable.
- Bandwidth

  The frequency range analyzed with the measurement.

  The bandwidth depends on the current controller sampling time.

  Example:

  125 µs current controller sampling time, set bandwidth 4000 Hz

11.2.3.2 Speed control loop measurement

Functionality

The response characteristics for the motor measuring system are analyzed when measuring the speed control loop. Various measurement parameter lists are available depending on the basic measurement setting which has been selected.

Operating path

Operating path for measuring the speed control loop: Operating area switchover > "Commissioning" > "Optimization/Test" > "Speed control loop"

Measuring functions

The following measurement functions are available for measuring the speed control loop:

Measuring type	Measured variable
Reference frequency response (downstream of the speed setpoint filter)	Actual speed value motor encoder/speed setpoint after filter
Reference frequency response (upstream of the speed setpoint filter)	Actual speed value motor encoder/speed setpoint after filter
Setpoint step change (downstream of the speed setpoint filter)	Measured variable 1: • Speed setpoint downstream of the filter • Actual torque value Measured variable 2: Actual speed value motor encoder
Interference frequency response (fault downstream of the current setpoint filter)	Actual speed value motor encoder / torque setpoint fct. generator
Disturbance variable step change (fault downstream of the current setpoint filter)	Measured variable 1: • Torque setpoint fct. generator • Actual torque value Measured variable 2: Actual speed value motor encoder
Speed-controlled system (excitation downstream of the current setpoint filter)	Actual speed value motor encoder/actual torque value
Frequency response of the mechanical parts [1)]	Actual speed value measuring system 1/actual speed value measuring system 2
1) The machine axis in question must have both a direct and an indirect measuring system to determine the frequency response of the mechanical parts.	

Measurement

The measurement sequence is divided into the following steps:

1. Setting the traverse range monitoring and the enable logic
2. Selecting the measuring type and measured variable
3. Setting the parameters, softkey "Measuring parameters"
4. Displaying the measurement results, softkey "Display"

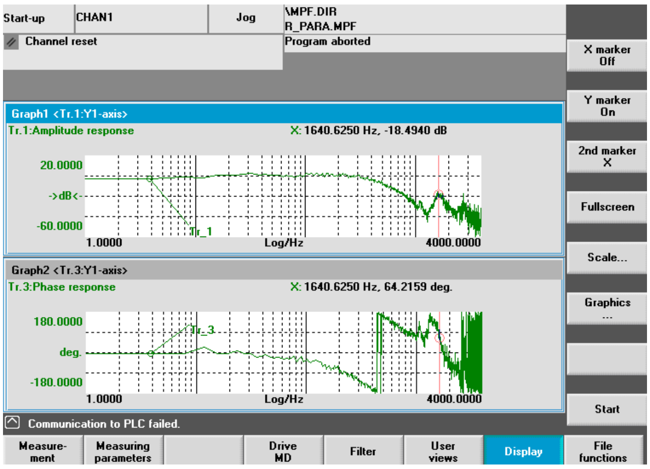


Figure 11-5 Speed controller

In the example shown, the speed control loop has not yet been optimized.

A suitable filter parameterization is used to optimize the dynamic response. This can be called with the "Filter" softkey.

The following figure shows the standard settings for a low-pass filter at 1999 Hz (encoder mounting frequency).

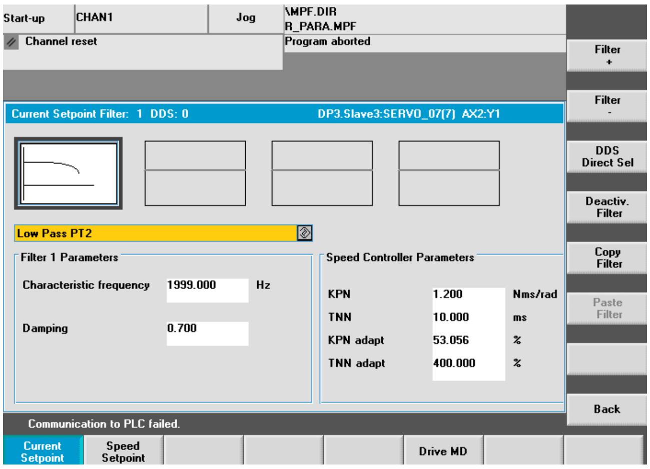


Figure 11-6 Standard settings of the speed control loop filter

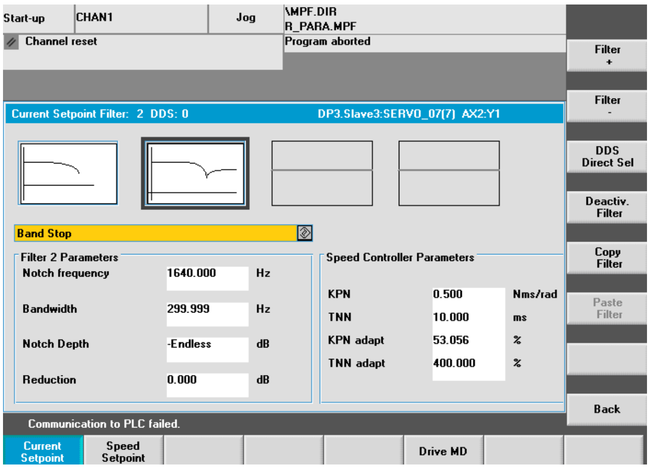


Figure 11-7 Speed control loop filter with rejection band 1190 Hz

The use of a rejection band at 1190 Hz and adaptation of the proportional gain results in the following optimized setting for the speed control loop.

Figure 11-8 Optimized speed control loop

11.2.3.3 Position control loop measurement

Functionality

This measuring function basically analyzes the response to the active position measuring system. If the function is activated for a spindle without a position measuring system, an alarm is displayed. Depending on the measured variable selected, various measurement parameter lists are displayed.

Operating path

Operating path for measuring the speed control loop: Operating area switchover > "Commissioning" > "Optimization/Test" > "Position control loop"

Measuring functions

The following measuring functions are available for measuring the position control loop:

Measuring type	Measured variable
Reference frequency response	Actual position/position setpoint
Setpoint step change	Measured variable 1: Position setpoint Measured variable 2: • Actual position value • System deviation • Following error • Actual speed value
Setpoint ramp	Measured variable 1: Position setpoint Measured variable 2: • Actual position value • System deviation • Following error • Actual speed value

Measurement

The measurement sequence is divided into the following steps:

1. Setting the traverse range monitoring and the enable logic
2. Selecting the measuring type and measured variable
3. Setting the parameters, softkey: "Measuring parameters"
4. Displaying the measurement results, softkey: "Display"

The following figure shows an optimized position control loop in which the K_v factor has been adapted via the machine data MD32200 $MA_POSCTRL_GAIN.

Figure 11-9 Optimized position control loop

Reference frequency response measurement

The reference frequency response measurement determines the transmission ratio of the position controller in the frequency range (active position measuring system).

The setpoint filters, control loop gain (K_v factor) and feedforward control must be parameterized such that resonance is avoided wherever possible over the entire frequency range.

Measuring parameters

- Amplitude

  This parameter determines the magnitude of the test signal amplitude. It should be set to the smallest possible value (e.g. 0.01 mm).

- Bandwidth

  The bandwidth parameter is used to set the analyzed frequency range. The larger this value, the finer the frequency resolution and the longer the measurement time. The maximum value is specified by the position controller cycle ($T_{\text{position controller}}$):

  Bandwidth_{max} [Hz] = 1 / (2 * $T_{\text{position controller}}$ [sec])

  Example:

  Position controller cycle: 2 ms

  $\text{Bandwidth}_{max} = 1 / (2 * 2*10^{-3}) = 250$ Hz

- Averaging

  The accuracy of the measurement and measurement duration increase with this value. A value of 20 is normally suitable.

- Settling time

  This value represents the delay between recording of the measured data and injection of the test setpoint and offset. A value of between 0.2 and 1 s is recommended. Do not set too low a value for the settling times, or the frequency response and phase diagrams will be distorted.

- Offset

  The measurement requires a slight speed offset of a few motor revolutions per minute. The offset must be set such that no speed zero crossings occur at the set amplitude.

Measurement: Setpoint step change and setpoint ramp

The transient or positioning response of the position control in the time range, and in particular the effect of setpoint filters, can be assessed with the step and ramp stimulation functions.

Possible measured variables:

- Actual position value (active position measuring system)
- Control deviation (following error)

Measuring parameters

- Amplitude

  Determines the magnitude of the specified setpoint step change or ramp.

- Measurement time

  This parameter determines the period of time to be recorded (maximum: 2048 position controller cycles).

- Settling time

  This value represents the delay between measured data recording / test setpoint output and the injection of the offset.

- Ramp time

  With default setting: The position reference value is specified with the "Setpoint ramp" according to the set ramp time. In this case, the acceleration limits which currently apply to the axis or spindle are effective.

- Offset

  The step is stimulated from standstill or starting from the constant traverse speed set in this parameter.

  If an offset value other than zero is input, the step change is stimulated during traversal. For the sake of clarity, the displayed actual position value does not include this speed offset.

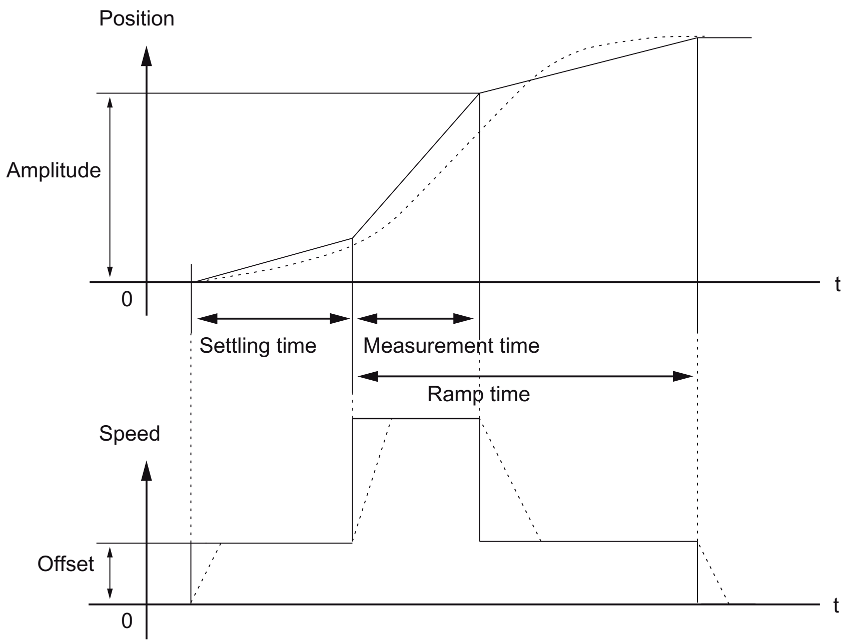


Figure 11-10 Signal chart for position setpoint/ramp measuring function

At maximum axis velocity, there is a (virtual) step change in the velocity (continuous line).

The curves represented by the dashed line correspond to a realistic, finite value. The offset component is excluded from the display graphic in order to emphasize the transient processes.

Measurement: Setpoint step change

To avoid overloading the mechanical system of the machine, the step height is limited to the value specified in the machine data during the "Setpoint step change" measurement:

- MD32000 $MA_MAX_AX_VELO (maximum axis velocity)

This may result in failure to achieve the desired step height.

Measurement: Setpoint ramp

With measurement "Setpoint ramp", the following machine data influences the measurement result:

- MD32000 $MA_MAX_AX_VELO (maximum axis velocity)

  The maximum axis velocity limits the ramp gradient (velocity limitation). The drive does not reach the programmed end position (amplitude).
- MD32300 $MA_MAX_AX_ACCEL (maximum axis acceleration)

  The maximum axis acceleration limits the velocity change (acceleration limitation). This leads to "rounding" on the transitions at the beginning and end of the ramp.

> **NOTICE**
>
> **Protection of the machine**
>
> In normal cases the machine data corresponds exactly with the load capacity of the machine kinematics and should not be changed (increased) as part of the measurements:
> - MD32000 $MA_MAX_AX_VELO (maximum axis velocity)
> - MD32300 $MA_MAX_AX_ACCEL (maximum axis acceleration)

11.2.4 Circularity test measurement

Functionality

The circularity test serves to set and assess the dynamic response for interpolating axes and to analyze the contour accuracy on the quadrant transitions (circular contours) achieved by means of friction compensation (conventional quadrant error compensation).

Operating path

Operating path to circularity test: Operating area switchover > "Commissioning" > "Optimization/Test" > "Circularity test"

Measuring parameters

The parameters are entered in the "Measurement" menu.

- Axis names and axis numbers
- Circle that is to be traversed and the actual position values recorded

  The parameter settings in the input fields "Radius" and "Feed" must correspond to the values from the part program that controls the circular motion of the axes, taking account of the feed override switch setting.

- The "Measuring time" display field shows the measuring time calculated from the "Radius" and "Feed" values for recording the actual position values during the circular movement.

  If only parts of the circle can be represented (i.e. measuring time too short) the measuring time can be increased in the menu by reducing the feed value. This also applies if the circularity test is started from the stationary condition.

Display mode

The following parameter assignments for programming the mode of representation of measurement results can also be made:

- Display based on mean radius
- Display based on programmed radius
- Scaling of the diagram axes

If the calculated measuring time exceeds the time range that can be displayed from the trace buffers (maximum measuring time = position-control cycle frequency * 2048), a coarser sampling rate is used for recording (n * position-control cycle frequency), so that a complete circle can be displayed.

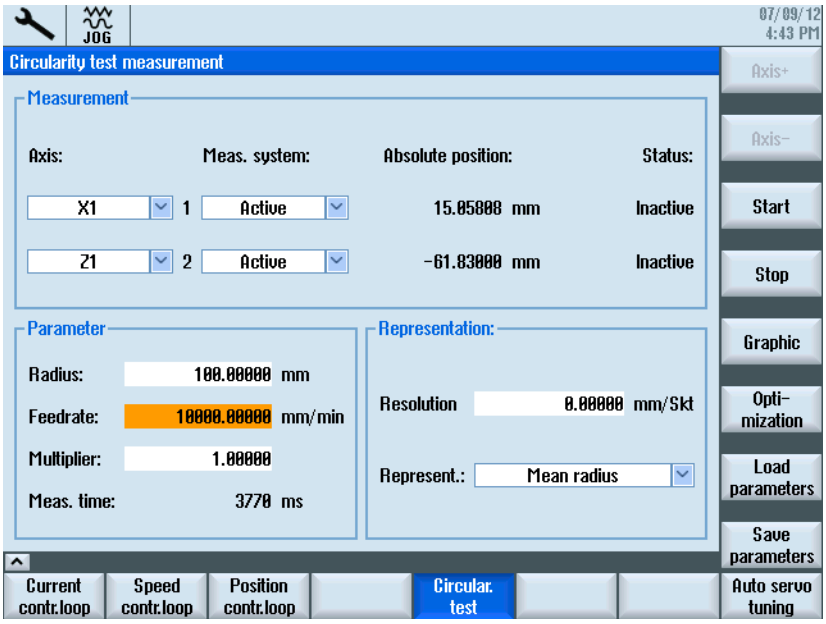


Figure 11-11 Circularity test measurement parameters

The two drives selected for measurement must describe a circular interpolation (G2/G3) with the parameters shown in the example via a part program:

Radius=100 mm, F=10000 mm/min

Measurement

The measurement sequence is divided into the following steps:

1. Setting the parameters, softkey "Measurement" (see above figure).
2. Start measurement with "Start" softkey.

   The selected axes run in the part program.
3. Displaying the measurement results, softkey "Display".

   A switchover is made to the graphic illustration of the recorded circular diagram (see following figure).

Note

If required, a QEC / backlash compensation can be performed via MD32200 $MA POSCTRL_GAIN for the K_V factor optimization.

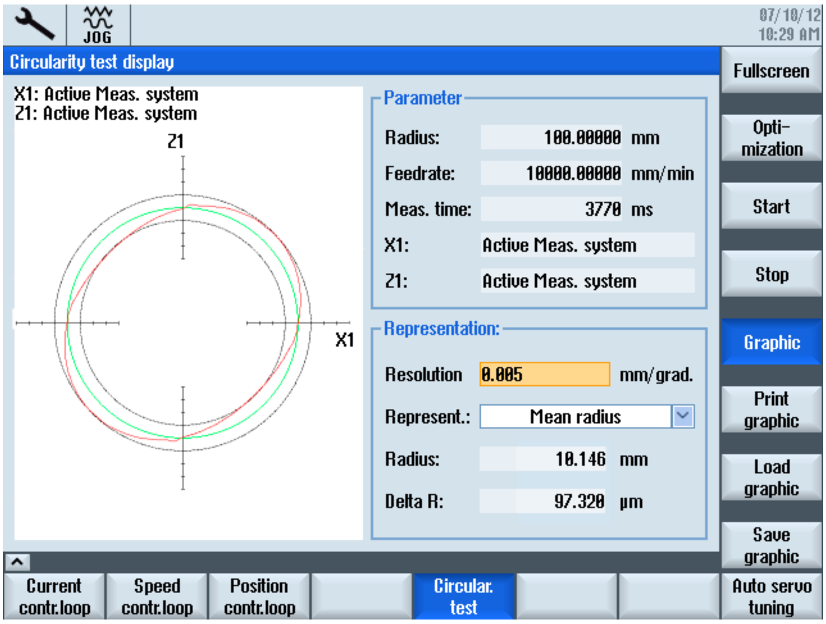


Figure 11-12 Circularity test measurement

11.2.5 Trace

11.2.5.1 Trace overview

Introduction

A trace shows signals over a time interval (signal charts)

The following functions are available:

- Servo trace

  Servo trace offers functions for recording and graphically illustrating the temporal characteristics of values for servo signals, e.g. actual position value, following error etc.

- Drive trace

  Drive trace offers functions for recording and graphically illustrating the temporal characteristics of values for signals from the drive system, e.g. actual speed value, actual current value etc.

  It must be possible for the signals to be recorded to be interconnected via a BICO source.

11.2.5.2 Drive trace

Basic drive trace display

The basic display of the "Drive trace" function is reached from the "Commissioning" > "Optimization/Test" > "Trace" > "Servo trace" operating area.

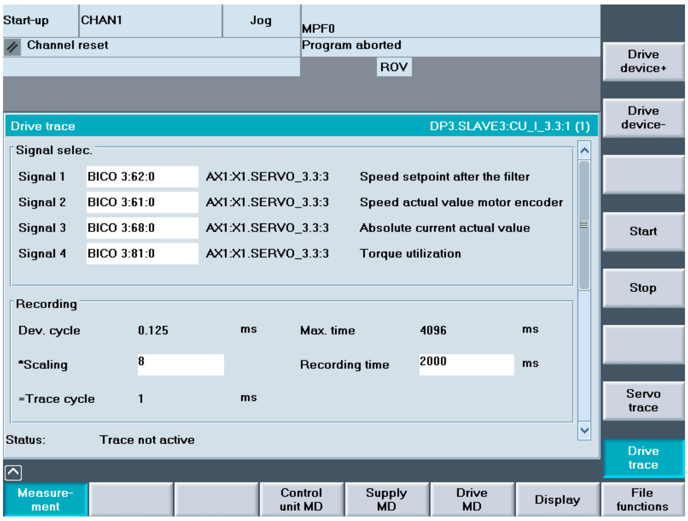


Figure 11-13 Basic "drive trace" display

When you are in the field for signal selection for a signal, you can press the <SELECT> button to move to a window in which you can determine the signal for interconnecting.

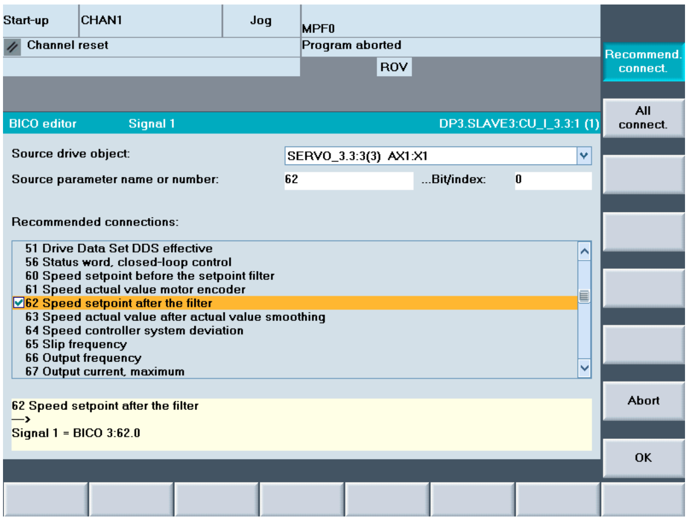


Figure 11-14 Drive trace interconnections

If you scroll through the basic display, the following parameters are displayed:

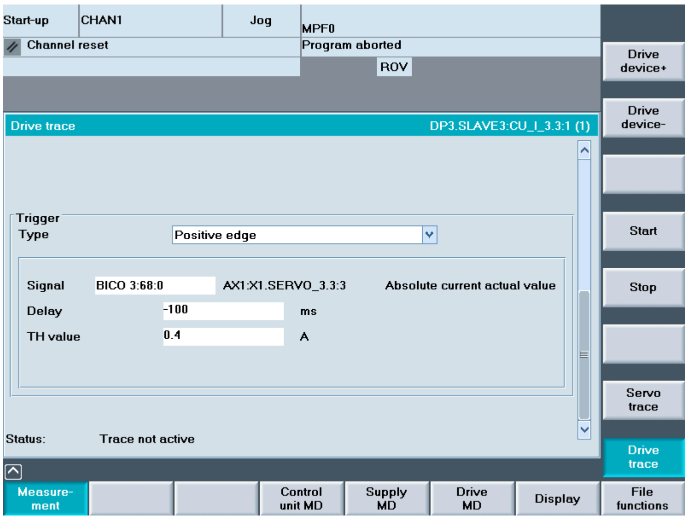


Figure 11-15 Basic drive trace display, "Trigger" parameter

"Trigger type" entry field

The trigger type is displayed in the "Trigger" drop-down list.

Settable trigger conditions:

- "Immediate recording", i.e. no trigger Start the measurement by pressing the "Start" softkey.
- "Positive edge"

- "Negative edge"
- "Entry in the hysteresis band"
- "Exiting the hysteresis band"
- "Trigger on bit mask"

"Trigger on positive edge / negative edge" trigger function

The threshold value for bit-coded values in the form of a decimal value entered in relation to the selected parameter, i.e. it is the weighting value of the individual bits in the sum of all parameters. The trigger function enables the triggering to larger/smaller values depending on the positive/negative edge, i.e. it cannot be explicitly triggered on a bit.

Example:

Signal = CU_I_3_3_1:r722

Threshold value = 3

This means, when the "Positive edge" trigger is used, the trigger is active for the value 3 or higher; the reverse is true when the "Negative edge" trigger is used. To activate the value 3 and the trigger, bits 0,1 (decimal value 3) or bit 2 (decimal value 4) is required.

"Trigger on bit mask" trigger function

The trigger is set to bit 3 in parameter r0722. If the entered values of the bit mask and the trigger condition are same, the trigger starts for the change from low to high, i.e. in this case, bit 3 of the parameter r0722.

Example:

Signal = CU_I_3_3_1:r722

Bit mask = 8

Trigger condition = 8

If the values are different, the trigger is started for a change from high to low of the bits that are defined in the bit mask until the value of the trigger condition is reached. In this example, the trigger starts when bit 1 changes from high to low and bit 0 is still high.

Example:

Signal = CU_I_3_3_1:r722

Bit mask = 3

Trigger condition = 1

Note

When the "Trigger on bit mask" function is used, only bit-coded values can be used (parameters with physical quantity are excluded).

Parameterization in the basic display

The following selection is made in the basic display for the drive trace measurement:

- Drive unit selection
- Signal selection
- Record
- Trigger

"Trigger time"/"Delay" entry field

Direct entry of pre-triggering or post-triggering. With negative input values (leading sign minus -) recording begins at the set time before the trigger event.

With positive input values (without sign) recording starts at the time set after the triggering event.

Secondary condition: Trigger time + measuring time ≥ 0.

Signal selection

Signals to be recorded, e.g. actual speed value, actual current value, etc.

It must be possible to interconnect the signals to be recorded, i.e. a BICO source.

Trigger signal

The trigger (signal) can be used to specify the event with which the recording of values is to start, e.g. the actual speed value is not to be recorded straightaway when the drive trace starts but only when the actual current value is > 10 A (here the actual current value trigger is > 10 A).

Literature

SINAMICS S120 Parameter Manual

"Drive unit+" and "Drive unit-" softkeys

Selection of the drive unit in which recording is to take place.

"Start" and "Stop" softkeys

Trace function recording is started with the "Start" softkey.

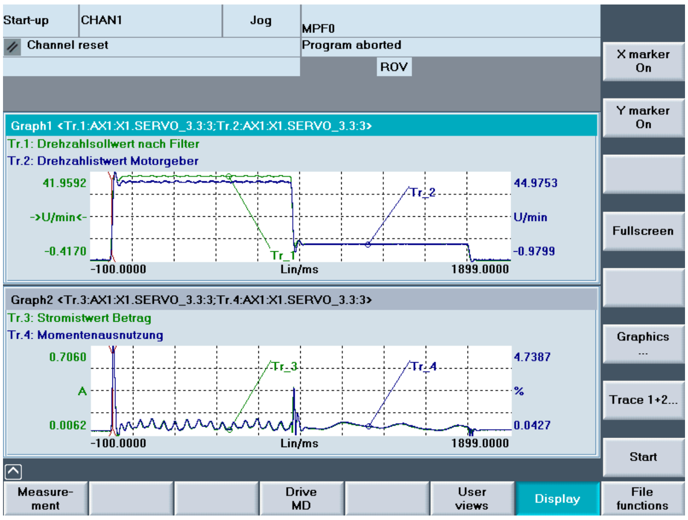


Figure 11-16 Drive trace recording

You can cancel an active recording with the "Stop" softkey or RESET.

11.2.5.3 Servo trace

Basic servo trace display

The basic display of the servo trace function is reached via the operating area "Commissioning" > "Optimization/Test" > "Trace" > "Servo trace".

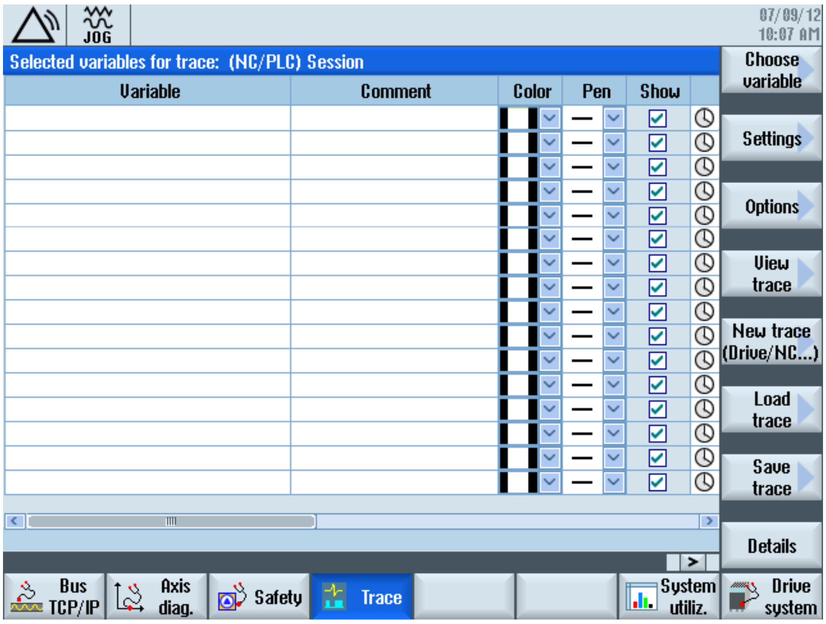


Figure 11-17 "Servo trace measurement" basic display

Parameterization in the basic display

The following selection is made in the basic display for the servo trace measurement:

- Axis/spindle selection
- Measuring signal
- Measurement time
- Trigger time
- Trigger type
- Trigger threshold

Signal selection

"Axis/spindle name" input field

The cursor must be positioned on the "Axis/spindle name" list box of the trace concerned. You can select it with the softkeys "Axis+" and "Axis-" or by accepting a value from the drop-down list box.

"Signal selection" input field

The cursor must be positioned on the "Signal selection" list box of the trace concerned. The selection is made through acceptance from the drop-down list box.

The available selection options depend on the existing configuration and activated functions.

Measuring parameters

"Measuring duration" input field

The measuring time is written directly into the "Measuring duration" input field.

"Trigger time" input field

Direct entry of pre-triggering or post-triggering. With negative input values (leading sign minus -) recording begins at the set time before the trigger event.

With positive input values (without sign) recording starts the time set after the triggering event.

Condition: Trigger time + measuring duration ≥ 0.

"Trigger" input field

The trigger type is displayed in the "Trigger" drop-down list box. The trigger always refers to Trace 1. When the trigger condition is satisfied, Traces 2 to 4 are started simultaneously.

Settable trigger conditions:

- "No trigger", i.e. measurement starts by pressing softkey "Start" (all traces are started time-synchronized)
- "Positive edge"
- "Negative edge"
- "Trigger event from the part program"

  The trace can be started via an NC part program in conjunction with the system variable $AA_SCTRACE [axis identifier].

Reference

SINUMERIK 840D sl / 840Di sl System Variable Manual

"Threshold" input field

Direct input of the trigger threshold.

The threshold is only effective with trigger types "Positive edge" and "Negative edge".

The unit refers to the selected signal.

Softkeys "Axis+" and "Axis-"

Selection of the axis/spindle when the cursor is positioned on the appropriate "Axis/spindle name" list field.

You can also select the axis/spindle directly in the list box from the drop-down list using the cursor.

Softkeys "Start" and "Stop"

Trace function recording is started with the softkey "Start".

With the "Stop" softkey or RESET, you can cancel a running measurement.

11.2.6 Adjusting speed and brake behavior

Introduction

You can adjust the following parameters in the "Commissioning" → "Machine data" → "Drive MD" operating area for the purpose of drive optimization.

Speed adjustment

- Spindle drive:

  p0500 = 102, speed setpoint in p0322 corresponds to setpoint 4000 0000hex
- Feed drive:

  p0500 = 101, speed setpoint in p0311 corresponds to setpoint 4000 0000hex

The speed setpoint can be diagnosed in the relevant drive in r2050[1+2] and r2060[1].

Brake behavior OFF3

Depending on the requirements, the brake behavior for each drive can be adjusted to the signal 2. OFF3.

Default setting p1135 = 0 brake with maximum current.

Using drive-specific parameterization, a flatter braking ramp can be set with the parameters p1135, p1136, p1137 .

Maximum braking ramp setting: 600 seconds.

12 STARTER commissioning tool

Note

Requirements for the SINUMERIK 802D sl

Please note the following requirements before using the STARTER to begin the drive startup for the SINUMERIK 802D sl:

- The SINUMERIK 802D sl must be in the POSITION operating area.
- No operator activities should be performed on the HMI during data communication between the SINUMERIK 802D sl and the STARTER (e.g. "Load project to PG/PC").

Operating sequence for calling the STARTER

To launch the STARTER program (on your PG/PC), click the STARTER icon or select "Start" > "Programs" > "STARTER" > "STARTER" from the Windows Start menu.

Note

The screen forms below have been taken from version V4.1 of the STARTER tool. If your particular version deviates from the version used here, your screen forms may deviate slightly from those shown here.

12.1 The STARTER user interface

You can use STARTER to create the sample project. To perform the individual configurations, use the user interface areas listed below:

- Project navigator: This area displays the elements (e.g., Insert single drive) and objects (e.g., Drive_1) you will insert in the project.
- Working area: Use this area to perform your task for creating the project:
  - When you are configuring the drive, this area contains the Wizards that help you configure the drive objects.
  - When you configure, for example, the parameters for the speed setpoint filter
  - When you call up the expert list, the system displays a list of all the parameters that you can view or change.
- Detailed view: This area provides detailed information on faults and warnings, for example.

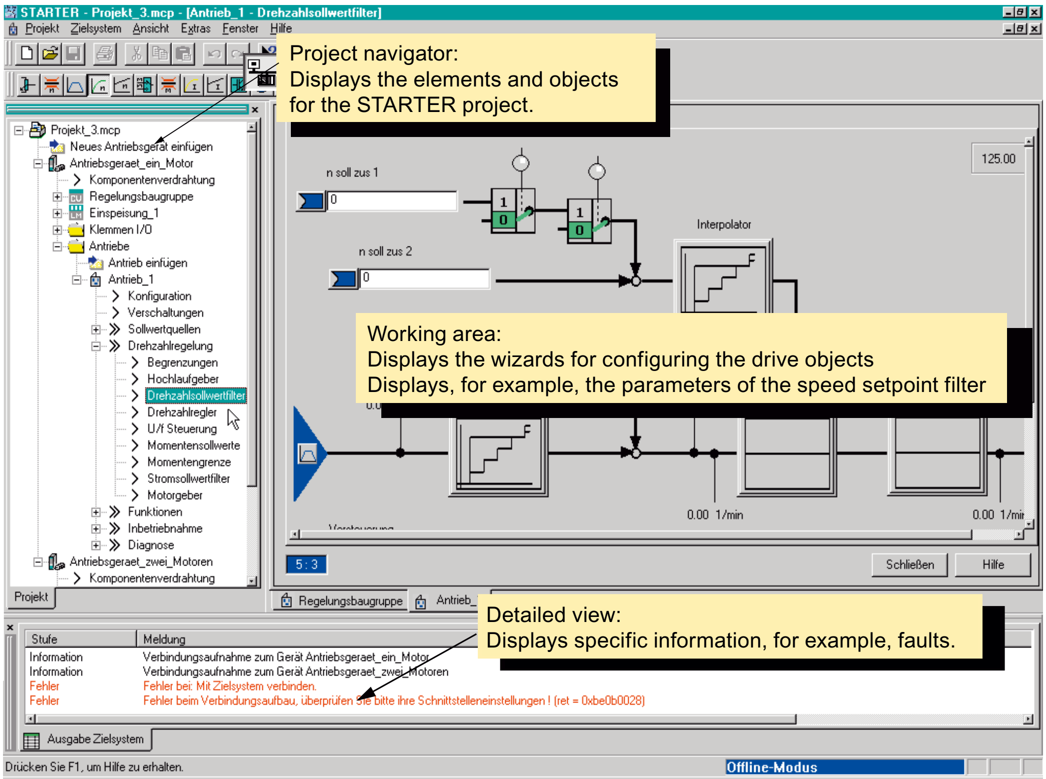


Figure 12-1 The different areas of the STARTER user interface

12.2 Operating philosophy of the STARTER commissioning tool for SINAMICS S120

When creating a drive unit for a SINAMICS S120 system, the following operating philosophy is assumed:

The tool is used to process objects (e.g. **supply**). The object name is user defined.

A drive unit in the terms of the STARTER commissioning tool is always a control unit and the appropriate drives.

The control unit is part of the SINUMERIK 802D sl operator panel control.

With a controlled infeed, the Active Line Module is configured in STARTER. An uncontrolled infeed is not represented in STARTER.

The relevant drive consists, for example, of a Motor Module (power unit) and a motor with an encoder.

The following figure shows the project navigator in STARTER. A project with the name **802D sl** and a drive unit with the name **SINAMICS_IN_802D** have been configured for 6 drives.

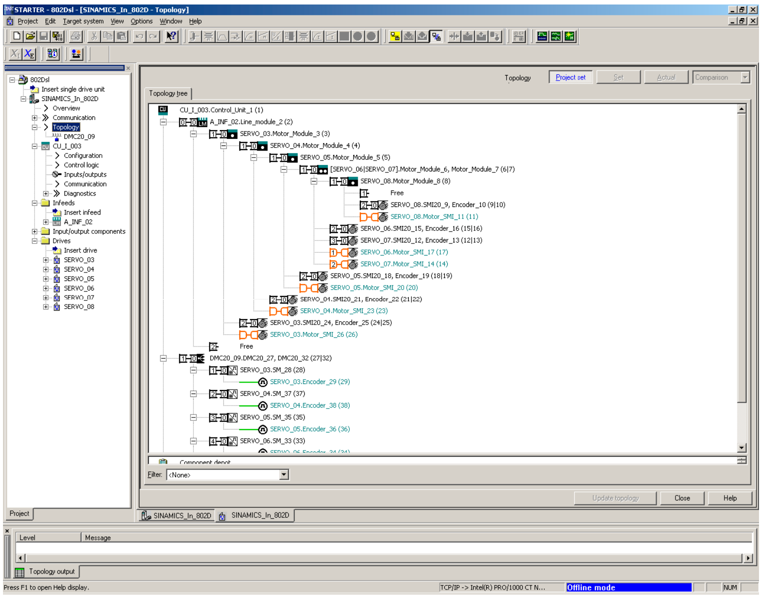


Figure 12-2 SINAMICS_IN_802D

12.3 Change a drive project OFFLINE

Prerequisites

- Components of the drive unit are assembled, completely wired (DRIVE-CLiQ)
- Commissioning has been carried out using the HMI (see Introduction to SINAMICS commissioning (Page 331))

Sequence of operations

To create a new project, proceed as follows:

1. Start the STARTER commissioning tool by clicking the STARTER icon, or via "Start" > "Programs" > "STARTER" > "STARTER" in the Windows Start menu.
2. Select "Project" > "New" from the menu.

   The "Insert single drive" window appears in the project navigator.

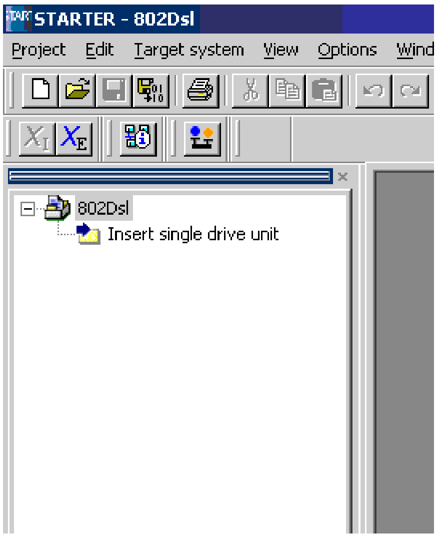


Figure 12-3 Inserting single drive

3. Double-click "Insert single drive".

4. Select the device type, device version and the IP address of the target device (SINUMERIK 802D sl) for online access.

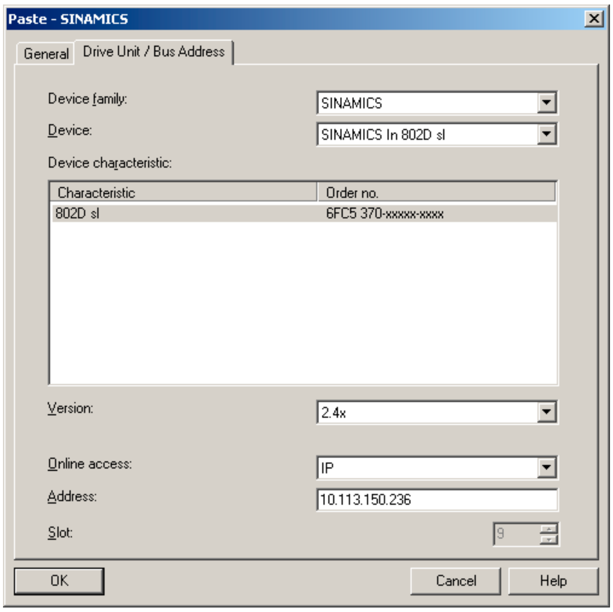


Figure 12-4 Device type, Device version and Online access

5. Confirm the selection with "OK".

   The new device type is inserted and displayed in the project navigator (see the screenshot below).

   To change an existing project in the target device OFFLINE, the project must be downloaded to the PG/PC ONLINE.

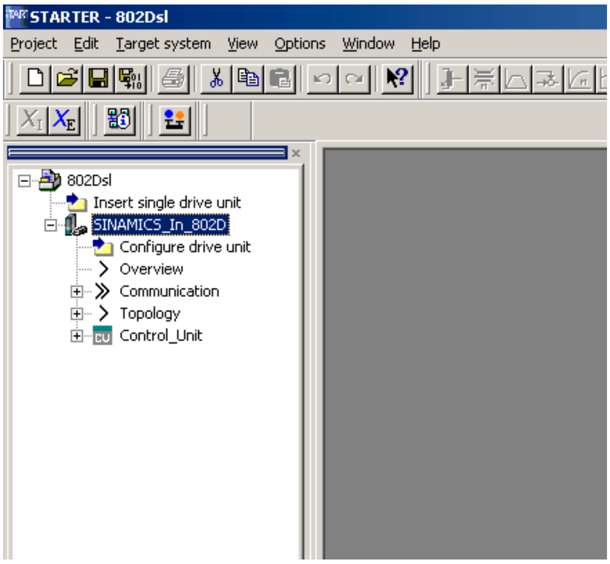


Figure 12-5 Screen after insertion

6. To establish an online connection with the target device, click the "Connect to target system" symbol.

   An ONLINE/OFFLINE comparison is displayed.

Note

If no device is detected, read out the STARTER and SSP versions via "Help" > "About" > "Systeminfo". Please contact the hotline with this information (see the "Technical Support"section in the preface).

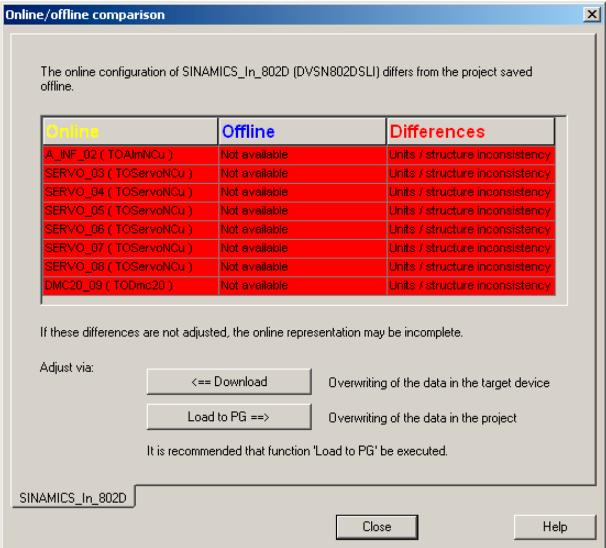


Figure 12-6 Online connection is established

7. Click "Load to PG ==>", "Yes" to confirm the prompt that follows, and "Close" once loading is complete.

   The project is downloaded to the PG/PC.

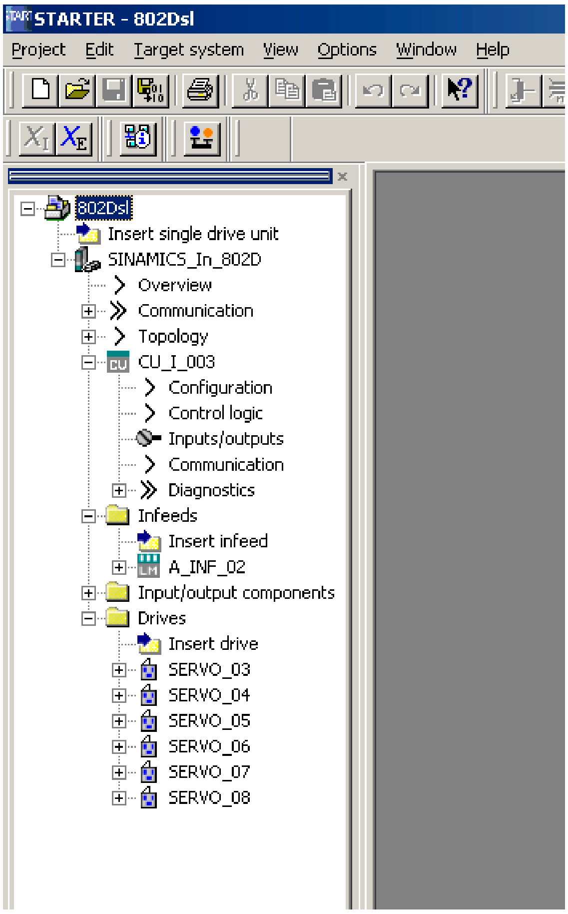


Figure 12-7 Loading the project to the PG

8. Click the "Disconnect from target system" icon to expand or modify a project OFFLINE.

13 Startup grinding cycles

13.1 General information

To be able to work with the cycles supplied, the machine (hardware) and the control (software) must meet certain minimum conditions.

These are described in the following sections.

13.2 Machine type

Cylindrical grinding machines

The cylindrical grinding machines used have two linear axes (X, Z), one workpiece spindle (S1) and a grinding spindle (S2), with or without a measuring system.

The workpiece spindles and grinding spindles can also have external motors and do not have to be known to the NC.

The grinding wheel can be arranged at right angles to the Z axis or at a fixed angle that is not equal to 90 degrees to the Z axis, the X and Z axis are then positioned perpendicular to one another (cartesian machine). It is also possible for the X axis to be oblique with respect to the Z axis (oblique axis).

There are multiple dressers in the working area of the machine for dressing the tools. The dressers are assigned to the grinding wheel. One wheel can be dressed with up to three dressers.

Surface grinding machines

The surface grinding machines used have three linear axes and one grinding spindle (S1).

The grinding spindle can be operated with or without a measuring system.

The grinding tool can only be mounted straight.

There are multiple dressers in the working area of the machine for dressing the tools. The dressers are positioned in front of or above the grinding wheel and can dress grinding tools in the diameter and behind and in front.

13.3 Cycles prerequisites

13.3.1 Hardware requirements

Other hardware requirements must be met by the grinding machine for the use of grinding cycles.

One or two handwheels are required for motion overlay during setup.

Connectors can be provided for the following external devices:

- Structure-borne noise device
- Measurement control
- Sensing probe
- 7 rapid inputs via MCPA for:
  - Measurement control (5 inputs)
  - Structure-borne noise device (2 inputs)

13.3.2 Software requirements

For the function of the cycles, all cycle variables and cycle macros of the tool box must be loaded. Furthermore, the contents of the user data must be assigned meaningful values by the machine manufacturer, such that the cycles can work with these values. Likewise, a minimum scope of PLC functions must be implemented.

Machine data conditions

The following settings must be made in the machine data:

MD number	Identifier	Value
18080	MM_TOOL_MANAGEMENT_MASK=	'H4'
18094	MM_NUM_CC_TDA_PARAM=	10
18096	MM_NUM_CC_TOA_PARAM=	10
18160	MM_NUM_USER_MACROS=	68

In the area of general machine data, all additional parameters of the tool cutting edges, tools and the minimum number of macros must be activated. These settings are necessary for tool management and the dressing of tools.

MD number	Identifier	Value
20150	$MC_GCODE_RESET_VALUES[7]=	8
20150	$MC_GCODE_RESET_VALUES[14]=	2
20150	$MC_GCODE_RESET_VALUES[21]=	2

MD number	Identifier	Value
20150	$MC_GCODE_RESET_VALUES[27]=	2
20310	$MC_TOOL_MANAGEMENT_MASK=	'H4'
21220	$MC_MULTFEED_ASSIGN_FASTIN=	'H2'

In the area of channel-specific machine data, it is necessary to set the reset groups for work offset, plane and diameter. For the cycles with measurement control, it is necessary to assign the rapid input byte.

MD number	Identifier	Value
24100	$MC_TRAFO_TYPE_1=	1024
24110	$MC_TRAFO_AXES_IN_1[0]=	1
24110	$MC_TRAFO_AXES_IN_1[1]=	2
24120	$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=	1
24120	$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=	2
24130	$MC_TRAFO_INCLUDES_TOOL_1=	0

In the channel area, the machine data for inclined axes must be set when using the inclined axis. If necessary, the Reset_Mode_Mask and the Start_Mode_Mask must be adapted in such a way that the transformation is kept after NC start and reset.

A condition for the cycles is diameter programming, which is selected by the cycles.

Required PLC interface signals (PLC to NCK)

```
N110 DEF CHAN INT _GC_IN_ABR=14 ;* Intermediate dressing upon key
```

V28001001.5

```
N120 DEF CHAN INT _GC_IN_HAND=15 ;* Key for handwheel override
```

V28001001.6

```
N130 DEF CHAN INT _GC_IN_BREAK=13 ;* Key for program interrupt
```

V28001001.4

Key (fast input $A_IN[ ]) for starting intermediate dressing, program interruption and handwheel override (oscillation);
The keys must be either wired directly or connected to the NCK interface via the PLC. For the oscillation, a stroke reversal must be initiated to immediately abort the oscillating motion, i.e. the signals must be decoded in the PLC. After aborting and dressing, the part is again approached using 30 µm relief. For inclined plunge-cutting, the relief is calculated using the angle to the Z axis.

For oscillation, the stroke reversal program control must be initiated by the PLC; for stroke reversal, infeed is always done, irrespective of the type of infeed; furthermore, infeed is automatically done depending on the type programmed for the oscillation reversal points.

Intermediate dressing upon key and program abort key are always operational during infeed; keys for handwheel override and stroke reversal are operational for oscillation operations only.

V32000006.4

A program level cancel is performed in conjunction with the fast inputs.

The machine manufacturer can either always permit this or control this via the M-functions GC_MF[18] and GC_MF[19]. These M functions are always called at the end of a surface. During sparking with structure-borne noise, the functions for handwheel and dressing must be deactivated. The cycle does not evaluate these while structure-borne noise is active.

Program control keys

In the SUBROUTINE-LIB (Toolbox), there is a GRINDING_CTL block, to ensure the correct function of the program control keys. This handles the control of the PLC interface signals for handwheel override, intermediate dressing, program interruption and stroke reversal. The infeed-stop signals for longitudinal grinding can be easily set using a $A_IN[xx]. The input is configured with _GC_IN_FEEDSTOP.

Cycle auxiliary subroutines

The cycle package includes the accessory subroutines that are required for the machining cycles and for setup.

- CYCLE417(INT _N_SITZ,STRING[32] _AKT_ABR_P,INT _N_ABR)

  Cycle for dressing on key. Called up during the grinding cycles. After dressing, further machining is performed with 30 µm of relief.

  – N_SITZ - seat number of the operation comes from higher-level cycle for seat correction

  – AKT_ABR_P - current dressing program name (now ""); the dressing parameters of the tools are used

  – N_ABR - number of dressing strokes

- CYCLE418(INT _N_SITZ)

  Cycle for program abort. A return position is approached, the tool and the coolant are switched off in order to make measurements or checks of the workpiece. After NC start, the further finishing takes place with 30 µm of relief If necessary, the cycle must be adapted to the machine by the machine manufacturer.

  – N_SITZ - seat number of the operation comes from higher-level cycle for seat correction

- CYCLE431(INT _ZU_ART)

  For oscillation and multiple plunge-cutting, the cycle is used to control the infeed in the reversal points and to thereby activate the infeed axis or the oscillating axis. (Use in the cycles CYCLE411 and CYCLE415)

  – ZU_ART - current infeed mode (-1/0/1 means left/both sides/right)

- CYCLE448(INT _MODE, REAL _REVPOS11, REAL _REVPOS12, REAL _DT11, REAL _DT12, REAL _FEED1, REAL _REVPOS21, REAL _REVPOS22, REAL _FEED2, REAL _INFEED, REAL _INFEEDSPEED, REAL _INFEEDEND, INT _INFEEDMODE, INT _FEED2MODE, _N_AUSFEUER, _GAP)SAVE DISPLOF

  Auxiliary cycle for level cancel during program control.

  Cycle required during manual grinding and for surface grinding and used by CYCLE419, CYCLE426, CYCLE427, and CYCLE428.

  Interface see cycle description of auxiliary cycle CYCLE448 (Page 420).

- CYCLE449 (INT _MODE, INT _ZKA, REAL _REVPOS11, REAL _REVPOS12, REAL _DT11, REAL _DT12, REAL _FEED1, REAL _INFEEDEND)

  Auxiliary cycle for level cancel during program control.

  Cycle required during manual grinding and for cylindrical grinding and is used by

  CYCLE452, CYCLE405.

  Interface see cycle description of auxiliary cycle CYCLE449 (Page 422).

Types of grinding wheels

The cycles support two types of grinding wheels: vertical and inclined wheels.

The grinding wheel infeeds only in the minus X direction and/or any (plus/minus) Z direction during cylindrical grinding.

The grinding wheel infeeds only in any (plus/minus) Z direction and only in the minus Y direction during machining by surface grinding. The X axis is the reciprocating axis.

The use of measuring devices and sensors

Measurement control is performed at the same time as the grinding machining on the workpiece diameter. At the allowance coordinates in X for cylindrical grinding, if available, for roughing, grinding and fine finishing, it allows the feed rates to be changed over or the determination of the end position.

13.4 Setup cycles

13.4.1 Scope of functions

For the implementation of the complete grinding process, a distinction is made between setting up and machining. To this end, setup functions and machining cycles are available to the user.

Several operations are performed during setup.

- Grinding wheel data input
  When inputting the wheel data, the geometry of the wheel is defined and a technology for dressing is stored.
- Data input of the dresser
  When inputting the dresser, the dresser geometry is stored
- Entering the dresser(s)
  When entering the dresser, the dresser is "trained" using the wheel.
- Form-truing/dressing
  Grinding tools need to be dressed after a certain time in service to sharpen them and to restore their original profile.

  Dressing of a wheel pursues two objectives:

  – Profiling provides the desired form of the wheel.

  – Sharpening restores the cutting ability and the defined geometry of the grinding wheel.
- Entering the workpiece
- Entering the switching button

The following cycles, which are started or selected via menus are available to the user for set up.

Note

All of the tool offsets are defined as DEFINE in the cycles. In this way, the CPU cycles for cylindrical grinding machines and surface grinding machines can be used, because the plane changes and along with it, the assignment of the geometric axes for the geometry of the grinding wheel.

The parameters of the grinding wheel (shape and technology) are also stored as DEFINE. This means that the user has the capability to work with macros (cycles) as a supply of parameters for the dressing cycles and can therefore use several dressing shapes for a grinding wheel.

The DEFINEs are globally stored in the SMAC.DEF.

Setup cycles

Setup cycles are used directly via the HMI.

CYCLE402 profiling

CYCLE403 Enter workpiece

CYCLE432 dressing

CYCLE433 Measurement control offset

CYCLE441 computation cycle tool offset data (wheel D fields) grinding position

CYCLE444 Dressing rolll with geometry axes

CYCLE445 Enter longitudinal position (with probe)

CYCLE446 GWPS selection

Setup cycles specially for cylindrical grinding

CYCLE401 Enter dresser

CYCLE 404, CYCLE445 Acquiring probe

Setup cycles specially for surface grinding

CYCLE443 Acquiring dresser

Internal cycles

CYCLE404 Enter longitudinal position

CYCLE417 dressing on key

CYCLE418 program interruption

CYCLE435 Compute dressing position

CYCLE436 Compensate dressing amount/dresser wear

CYCLE438 Traverse dresser, wear check of dresser and wheel

CYCLE439 Compute wheel reference points for standard contours

Auxiliary cycles

Auxiliary cycles can be modified by the user.

CYCLE421 roller speed for user

CYCLE422 switch off roller

CYCLE423 coolant on, dresser and wheel

CYCLE424 coolant off, dresser

CYCLE425 Auxiliary cycle for output of the speed (GWPS) limitation

Manual grinding

CYCLE419 Manual grinding

CYCLE448 Auxiliary cycle for level cancel during program control

13.4.2 CYCLE402 profiling

Programming

CYCLE402(DD, DZL, DZR, N_ABR, STARTABRICHTER)

Parameter

Table 13- 1 Parameters of CYCLE402

Parameter	Data type	Meaning
DD	REAL	Form-truing allowance in X
DZL	REAL	Form-truing allowance in Z, left side
DZR	REAL	Form-truing allowance in Z, right side
N_ABR	INT	Number of regular dressing strokes
STARTABRICHTER	INT	Start dresser for profiling

Function

This function is for form-truing a new wheel. The form-truing allowance is added to the acquired reference point, so that form-truing normally starts in the air. At end of this process, one last regular dressing stroke is executed. The form-truing allowance is automatically divided among the dressing amounts. For form-truing a free contour, the form-truing allowance is generated for the starting point and decremented so that only a Z direction is possible. For both sides of the grinding wheel, the user must change the reference points and the dresser and therefore determine the form-truing allowance himself. During the profiling, there is no compensation for dresser wear. The values for the current profiling allowances are at the same time kept in GUDs (_GC_PARR[1]... _GC_PARR[3]).

_GC_PARR[1]	Allowance on the diameter
_GC_PARR[2]	Allowance on the left and at the front
_GC_PARR[3]	Allowance on the right and behind

The value for the number of strokes is kept in the parameter _GC_PARR[4] and counted down by the cycle. For the first profiling, the theoretical profiling dimension is calculated first (if wear = zero).

13.4.3 CYCLE403 Enter workpiece

Programming

CYCLE403(AXIS, SETVALUE, NPKTV)

Parameter

Table 13- 2 Parameters of CYCLE403

Parameter	Data Type	Meaning
AXIS	AXIS	Axis name
SETVALUE	REAL	Setting value
NPKTV	INT	Number of the zero point offset

Function

This function is used to determine the workpiece positions in the machine with respect to the particular axis. The axis name and the setpoint are passed to the cycle via the HMI. The function can be expanded for auxiliary axes. In a 1st step, however, it contains the geometric axes and the workpiece spindle as an axis. For geometric axes, the geometric axis name is transmitted. When setting the axis, the respective wheel-specific offsets (of measurement control, etc.) are set to zero ($TC_DP24, $TC_DP25 for all wheel reference points, cutting edges).

13.4.4 Dressing - CYCLE432

Programming

CYCLE432(D_AB, Z_AB_L, Z_AB_R, FVOR_Z, FVOR_D_L, FVOR_B_L, FVOR_D_R, FVOR_B_R, N_ABR, D_PROF, Z_PROF_L, Z_PROF_R, SUG, SUGV, LEERHUB)

Parameter

Table 13- 3 Parameters of CYCLE432

Parameter	Data type	Meaning
D_AB	REAL	X dressing amount
Z_AB_L	REAL	Z dressing amount, left
Z_AB_R	REAL	Z dressing amount, right
FVOR_Z	REAL	Z feedrate in mm/rev.
FVOR_D_L	REAL	X feedrate, left, in mm/rev.
FVOR_B_L	REAL	Path feedrate, left, in mm/rev.
FVOR_D_R	REAL	X feedrate, right, in mm/rev.
FVOR_B_R	REAL	Path feedrate, right, in mm/rev.
N_ABR	INT	Number of dressing strokes
D_PROF	REAL	X profiling allowance
Z_PROF_L	REAL	Z profiling allowance, left
Z_PROF_R	REAL	Z profiling allowance, right
GWPS	REAL	Grinding wheel peripheral speed
SUGV	REAL	Ratio of the peripheral speeds between dressing roll and wheel
LEERHUB	INT	Idle strokes without infeed at the end of the dressing

Function

The dressing cycle serves to dress the wheel to the required wheel profile.

Sequence of operations

Dressing of a wheel is always started with the shoulders, followed by the diameter. The use of the dressers 1...3 and the appropriate dresser technology (direction) are observed. The wear values are compensated as dependent on the dressers used.

If a profiling allowance is specified, this is processed first. This value can also be used to search for a dresser if no sensor system is installed. When processing the profiling allowance, no dresser wear compensation is currently performed.

The overruns at the relief cuts are kept constant at the Z axis, i.e. when the wheel is getting physically smaller, the overrun increases internally automatically. This does not apply for the overrun when dressing the diameter.

If a crown height is programmed for a wheel with relief cut, this is always observed even if the existing wheel width is reduced. The dressing radius is reduced with the width.

The profiling allowance is taken into account in the base dimension of the dresser when selecting the valid coordinate system. This saves the use of a programmable work offset, which can then be used for the grinding operations. The values for the current profiling allowances are at the same time kept in GUDs (_GC_PARR[1]... _GC_PARR[3]).
Meaning of the parameters:

_GC_PARR[1] – allowance at the diameter,
_GC_PARR[2] – allowance on the left or at the front,
_GC_PARR[3] – allowance on the right or at the rear.

The _GC_PARR[5] parameter can be used to define a fixed speed for the grinding spindle. This speed is used when dressing if no spindle number is assigned (no NC spindle). The _GC_PARR[5] parameter can also be used when working with an externally controlled spindle.

13.4.5 Measurement control offset - CYCLE433

Programming

CYCLE433(T_NR, D_NR, KX, KZ)

Parameter

Table 13- 4 Parameters of CYCLE433

Parameter	Data type	Meaning
T_NR	INT	Current tool number
D_NR	INT	Current tool offset number
KX	REAL	Correction of the X axis
KZ	REAL	Correction of the Z axis

Function

The measurement control offset cycle contains the calculation of an offset value.

The cycle is part of all cycles for measuring head offset and regrinding.

The calculated difference between the finishing dimension for measurement control and actual position in X is loaded as the deviation into the wheel or zero offset in X depending on the GUD variable GC_KORR.

Sequence of operations

This cycle is part of all machining cycles for measuring head offset and regrinding (CYCLE410, CYCLE 411, CYCLE 413, CYCLE 415) for which measurement control is used. The parameter assignment is performed by the cycles.

Explanation of the parameters

T_NR (current tool number)

The current tool number is read in the higher-level machining cycle and assigned there to CYCLE433.

D_NR (current tool offset number)

The current tool offset number is read in the higher-level machining cycle and assigned there to CYCLE433.

KX, KZ (correction of the X or Z axis)

The axis correction is calculated in the higher-level machining cycle and assigned there to CYCLE433.

13.4.6 Computation cycle tool offset data - CYCLE441

Programming

CYCLE441(MX, MZ, ALPHA)

Parameter

Table 13- 5 Parameters of CYCLE441

Parameter	Data Type	Meaning
MX	REAL	Machine constant X
MZ	REAL	Machine constant Z
ALPHA	REAL	Current swivel angle

Function

The offset values for swiveled wheels are computed and stored in the basic dimension. The angle of the tool from ALPHA is included in the calculation. The machine constants can be 0. They are only needed if you are working with a freely swiveling tool (B axis). The prerequisite for this are Cartesian X and Z axes, i.e. these constants can not be used for freely swiveling operations of inclined axes. This requires special conversion of the workpiece coordinate system! It applies to all functions that work with these constants.

13.4.7 Dressing - CYCLE444 profile roller with geometry axes

Programming

CYCLE444(D_AB, FTVOR, FVOR, N_AUSROLL, N_ABR, D_PROF, SUG, SUGV, ABRICHTER)

Parameter

Table 13- 6 Parameter CYCLE444

Parameter	Data Type	Meaning
D_AB	REAL	X dressing amount
FTVOR	REAL	Insertion stroke in mm/rev
FVOR	REAL	Feedrate in mm/rev
N_AUSROLL	REAL	Number of coast down revolutions
N_ABR	INT	Number of dressing strokes
D_PROF	REAL	Form-truing allowance X
GWPS	REAL	Grinding wheel peripheral speed
SUGV	REAL	Ratio of the peripheral speeds between dressing roll and wheel
ABRICHTER	INT	Dresser number (1...3)

Function

This function is used for dressing the grinding wheel with a profile roller. The wear values are then compensated as well, depending on the dressers used. If a form-truing allowance is specified, it is executed as the first step. This value can also be used to search for a dresser if no sensor system is installed. When processing the profiling allowance, no dresser wear compensation is currently performed. The profiling allowance is taken into account in the base dimension of the dresser when selecting the valid coordinate system. This saves the use of a programmable work offset, which is then used for the grinding operations. The coast down revolutions are the number of revolutions taken for the roller to come to a stop against the wheel.

13.4.8 Selection of the grinding wheel peripheral speed - CYCLE446

Programming

CYCLE446(SUG)

Parameter

Table 13- 7 Parameters of CYCLE446

Parameter	Data Type	Meaning
GWPS	REAL	Value of the grinding wheel peripheral speed [m/s or feet/s]

Function

This function is used to switch on the grinding wheel at a desired peripheral wheel speed, including the testing of the max. peripheral wheel speed and RPM. If the speed is exceeded, a message is issued (no alarm). The value is limited to the respective maximum value. This is checked for all wheels that are mounted on the spindle (wheels of a set). A setup menu is also required in order to obtain an overview of the wheels used.

Checking and calculation is performed on the currently largest diameter of the wheels. This is a purely calculated monitoring function. Internally, no limitations are set that implement reliable monitoring. This must be ensured by the user.

For machines without NC spindles, it is possible to use a computation of the necessary speed with a spindle number ≤ 0 if the cycle CYCLE425 is available. In this case, the CYCLE425 receives the computed and limited speed. At this point, the user can give this speed to groups or directly to an external actuator (M functions, etc.). The user must then assign the speed set, which may deviate from the required speed, to parameter _GC_PARR[5]. In this way, the dressing cycle can compute, for example, the necessary dressing feedrate in mm/rev using the correct speed.

13.4.9 Setup cycles specially for cylindrical grinding

13.4.9.1 CYCLE401 Enter dresser

Programming

CYCLE401(AXISVALUE1, AXISVALUE2, AXISVALUE3, ABRICHTER, MX, MZ, PROF)

Parameter

Table 13- 8 Parameters of CYCLE401

Parameter	Data Type	Meaning
AXISVALUE1	REAL	Axis value in the MCS 1st geometric axis
AXISVALUE2	REAL	Axis value in the MCS 2nd geometric axis
AXISVALUE3	REAL	Axis value in the MCS 3rd geometric axis
ABRICHTER	INT	Dresser number (1...3): 1 - left side of the wheel 2 - right side of the wheel 3 - vertical dresser (neither pushing nor pulling)
MX	REAL	Machine constant X
MZ	REAL	Machine constant Z
PROF	INT	Pre-profiled wheel with profile roller

Function

The purpose of this function, for dressers that use the geometry axes, is to determine the dresser positions in the machine. The axis values are determined in machine coordinates by the HMI and transmitted to the cycle if all of the necessary axes have been entered. In doing this, first X and then Z must be measured (for cylindrical grinding machines). In the case of oblique axes, the Z position depends on the X position and the current X-position is therefore relevant to acquisition of Z. Internally, the angle of the tool is also computed from $TC_TPG8. The machine constants can be 0. They are only needed if you are working with a freely swiveling tool (B axis). The prerequisite for this are Cartesian X and Z axes, i.e. these constants can not be used for freely swiveling operations of inclined axes. This requires special conversion of the workpiece coordinate system! It applies to all functions that work with these constants. When a value is set for a pre-profiled wheel the profile depth is also calculated during registration.

13.4.9.2 Acquiring a probe - CYCLE404, CYCLE445

Programming

CYCLE445(Z_POS, F_Z, RICHTUNG)

Parameter

Table 13- 9 Parameters of CYCLE445

Parameter	Data Type	Meaning
Z_POS	REAL	Position Z
F_Z	REAL	Z measuring feed
RICHTUNG	INT	Search direction (-Z / +Z)

Programming

CYCLE404(Z_LPOS, Z_SCH, ZSTW, F_Z_MESS, FFW)

Parameter

Table 13- 10 Parameters of CYCLE404

Parameter	Data Type	Meaning
Z_LPOS	INT	Search direction (-1 = -Z; +1 = +Z)
F_SCH	REAL	Z setpoint position of the shoulder
ZSTW	REAL	Z infeed route
F_Z_MESS	REAL	Key feed
FFW	REAL	Retraction path

Function

This function is used to set the measuring position of the activating probe. The position is set up for each particular workpiece In this way, measurements can be made at a different position than during the setup. Multiple measurements are also possible (e.g. determining the plan dimensions for grinding a defined shoulder thickness).

Because the probe is usually mounted on the grinding headstock, it must also be possible to include probes that are fixed to an oblique axis in the calculation!

At the same time, the probing position is stored as a value in the variable _GC_LERF, so that the probe cycle can make a comparison. Optionally, a variable _GC_LERF for an external centering offset is provided, which can be described by the user (measuring button).

The variable _GC_LNPVZ saves the value of the zero point offset in the Z axis in order to be able to measure at any point on the workpiece and to restore the Z-axis status prior to the new measurement.

The variable _GC_LXPOS saves the position of the X axis during the measurement. This position is reapproached in the longitudinal direction.

The variable _GC_LVER is cleared at this point. Later, it contains the current centering offset.

13.4.10 Setup cycles specially for surface grinding

13.4.10.1 Acquiring a dresser - CYCLE443

Programming

CYCLE443(AXIS, ABRICHTER, MY, MZ, _PROF)

Parameter

Table 13- 11 Parameter CYCLE443

Parameter	Data Type	Meaning
AXIS	AXIS	Geo. axis name
ABRICHTER	INT	Dresser number (1...3)
MY	REAL	Machine constant Y
MZ	REAL	Machine constant Z
PROF	INT	Pre-profiled wheel with profile roller

Function

This function is used to determine the dresser positions in the machine for dressers that are used by means of the geometry axes. The axis values are determined in the NC and the axis name is transmitted to the cycle. It does not matter in which sequence you do this. The required compensation is calculated immediately and entered in the dresser's D field. The machine constants can be 0. They are only needed if the work is being done with a multidirectional swiveling tool (B axis), the conditions for which are Cartesian Y and Z axes, i.e. these constants cannot be used for just any swiveling operation of inclined axes. This requires special conversion of the workpiece coordinate system! It applies to all functions that work with these constants. When a value is set for a pre-profiled wheel the profile depth is also calculated during registration.

13.4.11 Internal cycles

13.4.11.1 Calculating the dressing position - CYCLE435

Programming

CYCLE435(D_AB, Z_AB, D_PROF, Z_PROF, XWP, ZWP, MX, MZ, STARTDS, STARTDA)

Parameter

Table 13- 12 Parameters of CYCLE435

Parameter	Data Type	Meaning
D_AB	REAL	Dressing amount in X radial
Z_AB	REAL	Dressing amount in Z
D_PROF	REAL	Form-truing allowance X radial
Z_PROF	REAL	Form-truing allowance Z
XWP	REAL	Offset of the coordinate system in X
ZWP	REAL	Offset of the coordinate system in Z
MX	REAL	Machine constant X
MZ	REAL	Machine constant Z
STARTDS	INT	Start reference point on the wheel for free contour
STARTDA	INT	Start reference point of the dresser for free contour

Function

The valid coordinate system is calculated for the respective reference point of the wheel and the necessary dresser. This function is also used for changing the reference points during dressing. Internally, the angle of the tool is also computed from $TC_TPG8. The machine constants can be 0. They are only needed if you are working with a freely swiveling tool (B axis). The prerequisite for this are Cartesian X and Z axes, i.e. these constants can not be used for freely swiveling operations of inclined axes. This requires special conversion of the workpiece coordinate system! It applies to all functions that work with these constants. The profiling dimension is added to the respective reference point.

13.4.11.2 Compensating for dressing amount and wear - CYCLE436

Programming

CYCLE436(D_AB, Z_AB_L, Z_AB_R, D_ABR1, Z_ABR1, D_ABR2, Z_ABR2, D_ABR3, Z_ABR3)

Parameter

Table 13- 13 Parameters of CYCLE436

Parameter	Data Type	Meaning
D_AB	REAL	Dressing amount in X radial
Z_AB_L	REAL	Dressing amount in Z, left wheel edge
Z_AB_R	REAL	Dressing amount in Z, right wheel edge
D_ABR1	REAL	Dresser wear in X 1st dresser (left wheel)
Z_ABR1	REAL	Dresser wear in Z 1st dresser (left wheel)
D_ABR2	REAL	Dresser wear in X 2nd dresser (right wheel)
Z_ABR2	REAL	Dresser wear in Z 2nd dresser (right wheel)
D_ABR3	REAL	Dresser wear in X 3rd dresser
Z_ABR3	REAL	Dresser wear in Z 3rd dresser

Function

Depending on the wheel type and dresser type (drawing, plunging, etc.) using the respective dresser, the dressing amount for all references points of the wheel (1-6) and the dresser wear of the dresser in question are compensated.

13.4.11.3 Traversing dresser to check wear of dresser and wheel - CYCLE438

Programming

CYCLE438

There are no parameters.

Function

Depending on the wheel type and dresser type, the wheel diameter, wheel width and the maximum wear values of the dresser tools are checked. If the max. is exceeded, an alarm is issued and the machining stops. For standard wheels, the diameters of reference points 1 and 2 are checked. For free contours, the minimum reference point is determined used for monitoring.

13.4.11.4 Calculating grinding wheel reference points for standard contours - CYCLE439

Programming

CYCLE439

There are no parameters.

Function

Depending on wheel type and dressing angle ($TC_TPG8) the reference points of the wheel (1 - 6) under 0 degrees is computed.

13.4.12 Auxiliary cycles

13.4.12.1 Roll speed for rotating dresser - CYCLE421

Programming

CYCLE421(SUG, SUGV, ABRICHTER)

Parameter

Table 13- 14 Parameters of CYCLE421

Parameter	Data Type	Meaning
GWPS	REAL	GWPS of the wheel
SUGV	REAL	Ratio of the peripheral speeds between dressing roll and wheel
ABRICHTER	INT	Dresser number

Function

Depending on the GWPS ratio between the dresser and wheel, a speed for the dresser is computed and output.

The cycle is the master for the user, because it must be modified by special value outputs, if applicable. Internally, the spindle number of the wheel ($TC_TPG1) and the GWPS ratio are evaluated and stored in the dresser.

The cycle can also be used for retracting or positioning the dressing tool. The call up always takes place after the computed selection of the tool at a lifting or retraction position. The user must ensure that the movement of the dresser is possible at this location. In the example, NC spindles and spindles with a fixed speed are implemented. In the case of a fixed speed, the speed values are in GUDs _GC_PARR[6...8] for the 3 possible dressers. In this case, the GWPS of the grinding wheel is adapted. If both spindles are NC-controlled, the speed of the dresser is calculated and, if applicable, limited. In this case, an operation message appears just as for the grinding wheel. The dressing process takes place despite this.

13.4.12.2 Switching off the rotating dresser - CYCLE422

Programming

CYCLE422(LHUB, ABRICHTER)

Parameter

Table 13- 15 Parameters of CYCLE422

Parameter	Data Type	Meaning
LHUB	INT	Last stroke active
ABRICHTER	INT	Dresser number

Function

The rotating dresser is switched off or reduced to a basic speed.

The cycle is the master for the user, because the procedure is technology-dependent.

Variable LHUB indicates that it is the last dressing stroke. The user can therefore decide whether to execute the switching operation every time or only on the last stroke. If dressing tools are used repeatedly (shoulder + diameter), this must be decoded by the user. In the example, the rotating dresser is switched to a safety speed, if a value is stored in the dresser.

13.4.12.3 Switching on coolant for dresser - CYCLE423

Programming

CYCLE423(ABRICHTER)

Parameter

Table 13- 16 Parameters of CYCLE423

Parameter	Data Type	Meaning
ABRICHTER	INT	Dresser number

Function

Depending on the type of dresser, the coolant for the wheel (spindle) or dresser is switched on. The cycle is the master for the user, because the procedure is machine-dependent.

13.4.12.4 Switching off coolant for dresser - CYCLE423

Programming

CYCLE424(LHUB, ABRICHTER)

Parameter

Table 13- 17 Parameters of CYCLE424

Parameter	Data Type	Meaning
LHUB	INT	Last stroke active
ABRICHTER	INT	Dresser number

Function

Depending on the type of dresser, the coolant for the wheel (spindle) and dresser is switched off. The cycle is the master for the user, because the procedure is machine-dependent.

Variable LHUB indicates that it is the last dressing stroke. The user thus has the capability to decide whether he will carry out the switching procedure each time or only during the last stroke. If dressing tools are used repeatedly (shoulder + diameter), this must be decoded by the user.

13.4.12.5 Output of the GWPS limit - CYCLE425

Programming

CYCLE425(MAXN)

Parameter

Table 13- 18 Parameters of CYCLE425

Parameter	Data Type	Meaning
MAXN	REAL	Maximum permissible speed

Function

The maximum value of the currently permitted speed is transmitted to the cycle in order to forward, if applicable, the value to a monitoring device (monitoring of the set speed). To this end, a spindle number must be assigned in the tool.

For machines without NC spindles, it is possible to use a computation of the necessary speed with a spindle number ≤ 0 if the cycle CYCLE425 is available. In this case, the CYCLE425 receives the computed and limited speed. At this point, the user can now give this speed to groups or directly to an external actuator (M functions, etc.). The user must then assign the speed set, which may deviate from the required speed, to parameter _GC_PARR[5]. In this way, the dressing cycle can compute, for example, the necessary dressing feedrate in mm/rev using the correct speed.

13.4.13 Manual grinding - CYCLE419, CYCLE448

Functionality

The manual grinding function is for grinding (finish-grinding) using a handwheel. This function does not require a workpiece program.

The control performs the oscillating movement of the axis (axes), if this is required.

Prerequisite for the cycles

The calculation resolution in MD10200 $MD_INT_INCR_PER_MM (calculation resolution for linear positions) (MD10210 $MD_INT_INCR_PER_DEG (calculation resolution for angular resolution)) must be at least 10 times higher than the input resolution in the display MD203 DISPLAY_RESOLUTION (display resolution) or MD204 DISPLAY_RESOLUTION_INCH (display resolution for inch dimension system).

The calculation resolution is taken into account when determining the starting position for grinding.

13.4.13.1 Manual grinding - CYCLE419

Programming

CYCLE419(MODE, TOOL, EDGE, ALPHA, SUG, N, REVPOS11, REVPOS12, DT11, DT12, FEED1, REVPOS21, REVPOS22, FEED2)

Table 13- 19 Parameter CYCLE419

Parameter	Data Type	Meaning
MODE	INT	Mode of the oscillating movements: 0 - no function 1 - infeed of 1st geo axis, no oscillation 2 - infeed of 2nd geo axis, no oscillation 3 - infeed of 3rd geo axis, no oscillation 11 - infeed of 1st geo axis, oscillation 2nd/3rd 12 - infeed of 2nd geo axis, oscillation 1st 13 - infeed of 3rd geo axis, oscillation 1st 21 - infeed of 1st geo axis, oscillation 2nd 22 - infeed of 2nd geo axis, oscillation 1st/3rd 23 - infeed of 3rd geo axis, oscillation 2nd 31 - infeed of 1st geo axis, oscillation 3rd 32 - infeed of 2nd geo axis, oscillation 3rd 33 - infeed of 3rd geo axis, oscillation 1st/2nd 3. Geo axis can only be selected, if present.
TOOL	INT	Tool
EDGE	INT	Cutting edge
ALPHA	REAL	Swivel angle of the tool
GWPS	REAL	Grinding wheel peripheral speed
N	REAL	Workpiece speed
REVPOS11	REAL	1. Reversal point of the 1st reciprocating axis
REVPOS12	REAL	2. Reversal point of the 1st reciprocating axis
DT11	REAL	Dwell time at the 1st reversal point of the 1st reciprocating axis (in revolutions for cylindrical grinding and in seconds for surface grinding; value <0 results in intermittent grinding)
DT12	REAL	Dwell time at the 2nd reversal point of the 1st reciprocating axis (in revolutions for cylindrical grinding and in seconds for surface grinding; value <0 results in intermittent grinding)
FEED1	REAL	Feedrate of the 1st reciprocating axis
REVPOS21	REAL	1. Reversal point of the 2nd reciprocating axis
REVPOS22	REAL	2. Reversal point of the 2nd reciprocating axis
FEED2	REAL	Feedrate of the 2nd reciprocating axis in in mm/stroke of the 1st reciprocating axis, if a 2nd reciprocating axis is possible (surface grinding). If the value <0, the end points of the 1st reciprocating axis are not passed

Note

The parameters for the 2nd reciprocating axis are only present optionally if there are 3 geometry axes! For the two reciprocating axes, the reversing points can be taken over using a softkey (e.g. "Position 1 X").

Function

The cycle is only used for manual grinding using a handwheel.

Sequence

While the cycle is running, it is possible to pause it, dress, and terminate it. (During surface grinding, cancellation takes effect at the next reversing point. Cancellation is possible at any time during cylindrical grinding.)

On termination and pausing the infeed axis moves to the return position. This is done by moving over the starting points of the cycle, i.e. during oscillation in Z (cylindrical grinding), movement is initially to the Z starting point and is then canceled in X. The same sequence is executed on resumption of grinding, first to the X return position and then to the starting point Z.

13.4.13.2 Auxiliary cycle for level cancel during program control - CYCLE448 (flat and cylindrical grinding)

Programming

CYCLE448(MODE, REVPOS11, REVPOS12, DT11, DT12, FEED1, REVPOS21, REVPOS22, FEED2, INFEED, INFEEDSPEED, INFEEDEND, INFEEDMODE, FEED2MODE, N_AUSFEUER, GAP)

Parameter

Table 13- 20 Parameter CYCLE448

Parameter	Data Type	Meaning
MODE	INT	Mode of the oscillating movements: 0 - no function 1 - infeed of 1st geo axis, no oscillation 2 - infeed of 2nd geo axis, no oscillation 3 - infeed of 3rd geo axis, no oscillation 11 - infeed of 1st geo axis, oscillation 2nd/3rd 12 - infeed of 2nd geo axis, oscillation 1st 13 - infeed of 3rd geo axis, oscillation 1st 21 - infeed of 1st geo axis, oscillation 2nd 22 - infeed of 2nd geo axis, oscillation 1st/3rd 23 - infeed of 3rd geo axis, oscillation 2nd 31 - infeed of 1st geo axis, oscillation 3rd 32 - infeed of 2nd geo axis, oscillation 3rd 33 - infeed of 3rd geo axis, oscillation 1st/2nd 3. Axis can only be selected, if present
REVPOS11	REAL	1. Reversal point of the 1st oscillating axis
REVPOS12	REAL	2. Reversal point of the 1st oscillating axis
DT11	REAL	Dwell time at the 1st reversal point of the 1st reciprocating axis (in revolutions for cylindrical grinding and in seconds for surface grinding; value <0 results in intermittent grinding)
DT12	REAL	Dwell time at the 2nd reversal point of the 1st reciprocating axis (in revolutions for cylindrical grinding and in seconds for surface grinding; value <0 results in intermittent grinding)
FEED1	REAL	Feedrate of the 1st oscillating axis
REVPOS22	REAL	2. Reversal point of the 2nd reciprocating axis
FEED2	REAL	Feedrate of the 2nd reciprocating axis in in mm/stroke of the 1st reciprocating axis, if a 2nd reciprocating axis is possible (surface grinding). If the value <0, the end points of the 1st reciprocating axis are not passed
INFEEND	REAL	Infeed at the reversal point
INFEEDMODE	INT	Infeed mode at the reversal point: 0 - both sides 1 - Start -1 - End
FEED2MODE	INT	Infeed mode at reversing point X for Z: 0 - both sides 1 - Start -1 - End
N_AUSFEUER	INT	Number of sparking out strokes
GAP	INT	Grinding-in-air infeed active for deactivation and activation of the structure-borne noise sensor during infeed

13.4.13.3 Auxiliary cycle for level cancel during program control - CYCLE449 (cylindrical grinding)

Programming

CYCLE449(MODE, ZKA, REVPOS11, REVPOS12, DT11, DT12, FEED1, INFEEDEND)

Parameter

Table 13- 21 CYCLE449 parameters

Parameter	Data Type	Meaning
MODE	INT	Mode of the oscillating movements: 0 - no function 1 - infeed of 1st geo axis, no oscillation 2 - infeed of 2nd geo axis, no oscillation 3 - infeed of 3rd geo axis, no oscillation 11 - infeed of 1st geo axis, oscillation 2nd/3rd 12 - infeed of 2nd geo axis, oscillation 1st 13 - infeed of 3rd geo axis, oscillation 1st 21 - infeed of 1st geo axis, oscillation 2nd 22 - infeed of 2nd geo axis, oscillation 1st/3rd 23 - infeed of 3rd geo axis, oscillation 2nd 31 - infeed of 1st geo axis, oscillation 3rd 32 - infeed of 2nd geo axis, oscillation 3rd 33 - infeed of 3rd geo axis, oscillation 1st/2nd 3. Axis can only be selected, if present
ZKA	INT	Direction
REVPOS11	REAL	1. Reversal point of the 1st oscillating axis
REVPOS12	REAL	2. Reversal point of the 1st oscillating axis
DT11	REAL	Dwell time at the 1st reversal point of the 1st oscillating axis (in revolutions if an NC spindle)
DT12	REAL	Dwell time at the 2nd reversal point of the 1st oscillating axis (in revolutions, if an NC spindle)
FEED1	REAL	Feedrate of the 1st oscillating axis
INFEEND	REAL	End point of the infeed

14 Data Backup and Series Machine Start-Up

14.1 Data Backup

You have the following options for backing up user data in the control system:

- Internal

  Data are backed up internally on the control system.

- External

  Data can be backed up externally using the following methods:

  – PG/PC

  – CompactFlash Card (customer CF card)

  – USB-FlashDrive (USB drive)

Note

Archiving/data backup

It is recommended to back up the internal SINUMERIK memory on the CompactFlash card regularly. The backed up data can then be retransferred into the SINUMERIK later on. This way the previous status of the unit is restored.

14.1.1 Internal data backup

The data of the limited-buffered memory must be saved via a backup copy to the permanent memory of the control system. This backup is performed internally and is always necessary if the control system is switched off for longer than 60 hours.

Recommendation: After changing important data, it is recommended to carry out a data backup **immediately**.

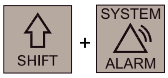
Note

If the backup time has elapsed, you will be able to read in a commissioning archive file approximately 10 minutes after switching on the control system.

Note

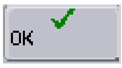
During the data backup, an image of the limited-buffered memory is produced and stored in the permanent memory. A backup of selected data (e.g. only machine data and no workpiece programs) is not possible.

Performing an internal data backup



Save data

In the <SYSTEM> operating area, press the "Save paramet." softkey (at least protection level 3 required).



Click "OK" to confirm the messages that follow.

Note

While the internal data backup is running, the control system must neither be operated, nor be turned off.

Note

Drive machine data is not included within the internal data backup process. This data is permanently backed up to the CompactFlash Card system.

For data management purposes, the drive machine data can be saved to the following location: <SYSTEM> operating area > "Startup files" > "802D data" > "Startup archive (Drive/NC/PLC/HMI)" directory > "Drive machine data".

Loading internally backed-up data

- Boot the control system in the commissioning mode "Reload saved user data".
- If data are lost from the buffer memory, the data saved in the permanent memory will automatically be reloaded to the memory on POWER ON.

  Note

  Message "4062 Data backup copy has been loaded" is displayed on the screen.

Note

The password will have to be entered again after the control system with the backed up data has been powered up.

14.1.2 External data backup

In addition to an internal data backup, the user data of the control system can also be saved externally.

The following possibilities exist for external data back-up:

- External data backup to CompactFlash Card (customer CF card)
- External data backup to USB-FlashDrive (USB drive)
- External data backup using the RCS802 tool (contained in the Toolbox).

  This requires a PG/PC Data backup can be performed via the following interfaces:

  – V24/RS232 interface

  – Ethernet interface

An external data backup should be performed if major data changes have been made or always at the end of the commissioning.

For complete external data backup of a machine, it is enough to create the series startup file.

Variants of external data backup

1. Reading out the data completely: Series startup (Page 428)
2. Files are read out area by area.

   The following user data can be selected in the <SYSTEM> opeating area using the function "Startup files" > "802D data" as **single files**:

   Note

   Sag compensation is ONLY listed if the associated function has been activated.

   Data in the text format

   – Machine data

   – Setting data

   – Tool data

   – R parameters

   – Zero Offset

   – Leadscrew error compensation

   – Sag compensation

   – Global user data

   Start-up archive (Drive/NC/PLC/HMI)

   – Drive machine data

   – NC data

- NC directories
- Display machine data
- Leadscrew error compensation
- Sag compensation
- PLC project
- HMI data and applications

PLC project (.PTE)*

File for license key

3. The following data can also be backed up in the <PROGRAM MANAGER> operating area:
   - CMA cycles machine manufacturer
   - CST Siemens cycles
   - CUS user cycles
   - MPF main programs
   - SDB
   - SPF subprogram files

14.1.2.1 External data backup via CompactFlash Card or USB-FlashDrive

The same data as saved via the serial interface can be backed up to the CompactFlash Card or the USB-FlashDrive.

The data to be backed up is selected via the <SYSTEM> operating area > "Startup files" > "802D data" > "Copy".

Select "Customer CF card" and "Paste" or "USB drive" and "Paste" to save the data to the card.

14.1.2.2 External data backup via the RS232/Ethernet interface

Note

Never connect or disconnect the RS232 cable when the PCU is connected to the mains.

The settings of the RS232 interface of the 802D sl and the COM interface on the PG/PC must be identical.

Storing the startup archive in the PG/PC (transfer from the control into the PC)

See Section Series startup (Page 428)

Data backup in the <SYSTEM> operating area > "Startup files" > "802D data"

In PG/PC directory "802 Data" directory "Data", individual files can be copied from the control and stored in a directory on the PG/PC.

Data backup in the <PROGRAM MANAGER> operating area

In PG/PC directory "802 Data" directory "NC drive", individual files can be copied from the control and stored in a directory on the PG/PC.

References

SINUMERIK 802D sl "Operation and Programming", chapter "Data backup"

14.1.3 Data backup in case of backlight failure

In case of failure of the backlight of the control system, menu-assisted operation is no longer possible.

The following operating sequence describes how you can backup data externally in this situation.

Operating sequence

1. Insert the CompactFlash card into the slot on the front of the control.
2. Switch the control on.
3. Wait until the control has booted.

   The LEDs on the operator panel CNC RDY (green) and NC (yellow) show the status "Ready for operation".
4. Press the <CTRL + S> key combination.

   The external data backup starts.

   The series start-up archive (Drive/NC/PLC/HMI) is exported with the most recent data onto the CompactFlash card with the name "802Dslibn.arc".

   During output, the red and green LEDs flash at a rate of approx. 0.5 Hz on the status and error display (LEDs of the operator panel CNC).
5. Wait until the flashing stops and the LED displays are in the original state. Only then is the writing to the CompactFlash card complete.
6. The write operation has been completed.

   Pull the CompactFlash card out of the slot on the front of the control.

14.2 Series machine startup

Functionality

The aim of standard commissioning is:

- After a first commissioning, putting a further control on the same type of machine into the same state as after first commissioning;

  or

- Putting a new control into the original state as easily as possible in case of service (hardware replacement).

Start-up archive (Drive/NC/PLC/HMI)

The start-up archive (Drive/NC/PLC/HMI) has the following selectable content:

- Drive machine data
- NC data
- NC directories
- Leadscrew error compensation
- Sag compensation

  Note

  Sag compensation is ONLY listed if the associated function has been activated.

- PLC project (*.PTE)
- HMI data and applications

Requirements

A standard commissioning can be performed via the following interfaces for data transfer from/to the control:

- User CompactFlash Card (customer CF card) in the slot at the front of the control.
- USB-FlashDrive (USB drive) on the control system
- PG/PC with V24 interface or Ethernet interface.

  In the PG/PC, the RCS802 tool must be used.

Sequence with CompactFlash card

1. Generating a series startup file on the CompactFlash card:
   – CompactFlash card must be inserted in the slot on the front of the control.
   – The control system requires the password for protection level 2.
   – Select the "Start-up archive (Drive/NC/PLC/HMI)" line in the <SYSTEM> operating area > "Start-up files" > "802D data" and copy to the clipboard with "Copy". Select the "Customer CF card" softkey to display the content of the inserted card. If you select the "Paste" softkey and then enter the name for the archive file, the data relating to the series machine startup will be generated on the card.
2. Importing the series startup file from the CompactFlash card into the SINUMERIK 802D sl
   – The CompactFlash card must be inserted!
   – The control system requires the password for protection level 2.
   – In the <SYSTEM> operating area, in "Start-up files" > "Customer CF card", select the line containing the archive required and use "Copy" to paste the data onto the clipboard. Select the "802D data" softkey and choose the line "Startup archive (Drive/NC/PLC/HMI)". The series machine startup (commissioning) is transferred into the control system with the "Paste" softkey.
   – Confirm the start of series machine startup (commissioning) in the display shown on the control system after importing starts.
   – A warm restart of the NC/PLC is performed several times during the series machine startup (commissioning). At the end of the series commissioning, the complete control system is rebooted. After an error-free series commissioning, the control system will be in a fully configured operating state.

USB-FlashDrive sequence

1. Generating a series startup file on the USB-FlashDrive:
   – The USB-FlashDrive must be inserted into the USB plug (USB interface X10) at the rear of the control system's CNC operator panel (PCU).
   – The control system requires the password for protection level 2.
   – Select the "Start-up archive (Drive/NC/PLC/HMI)" line in the <SYSTEM> operating area > "Start-up files" > "802D data" and copy to the clipboard with "Copy". Select the "USB drive" softkey to display the contents of the inserted card. If you select the "Paste" softkey and then enter the name for the archive file, the data relating to the series machine startup will be generated on the card.
2. Importing the series startup file from the USB-FlashDrive into the SINUMERIK 802D sl
   – The USB-FlashDrive must be inserted!
   – The control system requires the password for protection level 2.
   – In the <SYSTEM> operating area, select "Startup files" > "USB drive" and choose the line containing the archive required. Then use "Copy" to paste the data onto the clipboard. Select the "802D data" softkey and choose the line "Startup archive (Drive/NC/PLC/HMI)". The series machine startup (commissioning) is transferred into the control system with the "Paste" softkey.

– Confirm the start of series machine startup (commissioning) in the display shown on the control system after importing starts.
– A warm restart of the NC/PLC is performed several times during the series machine startup (commissioning). At the end of the series commissioning, the complete control system is rebooted. After an error-free series commissioning, the control system will be in a fully configured operating state.

Sequence with PC (RCS802)

1. Creating the start-up archive (Drive/NC/PLC/HMI) in the PG/PC (transfer from the control to the PC):
   – Make the connection between the PG/PC (RCS802) and the control. The control system requires the password for protection level 2.
   – In the directory tree of the RCS802, open Control 802 > 802D data (A:), and select the directory Start-up archive (drive/NC/PLC/HMI). Click "Copy" in the context menu (right mouse button).
   – Select the target directory on the PG/PC in the directory tree and insert the startup archive with "Paste" in the context menu.
2. Importing series startup file from PG/PC into the control
   – Make the connection between the PG/PC (RCS802) and the control. The control system requires the password for protection level 2.
   – In the directory tree of the RCS802, select the commissioning archive to be transferred, right-click to access the context menu, and click "Copy".
   – In the directory tree of the RCS802, open Control 802 > 802D data (A:), and select the directory Start-up archive (drive/NC/PLC/HMI). Click "Paste" in the context menu (right mouse button).
   – The series commissioning then commences. A warm start of the NC/PLC is performed several times. At the end of the series commissioning, the complete control system is rebooted. After an error-free series commissioning, the control system will be in a fully configured operating state.

Update 15

Introduction

The control can be updated using the CompactFlash card in the slot on the front of the control.

Note

Backup/archive the data of the control system (Drive/NC/PLC/HMI) before you start the update!

See section Data Backup and Series Machine Start-Up (Page 423).

Requirements

- The control system is switched off.
- Control system BIOS version 00.00.03.03 or higher.
- You must have installed and started the RCS802 tool from the current toolbox on the PG/PC.

  Use the RCS802 tool to write an "802Dsl.upd" image file to the CompactFlash Card.

Operating sequence for updating with the RCS802 tool

1. In the menu select "File" > "Write CF card".

   The following dialog form opens:

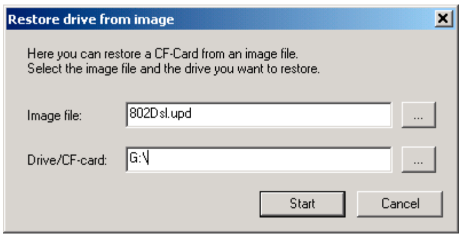
   

   Figure 15-1 Restore drive from image

2. Select the "802Dsl.upd" image file and choose the CompactFlash Card drive as the target.
3. Click "Start".

4. Wait until the write process is complete and click "OK".

Note

Do not insert the CompactFlash Card into the slot on the front of the control system at this stage.

5. Switch the control on.
6. After the control system has been powered up, check whether the general machine data MD11210 $MN_UPLOAD_MD_CHANGES_ONLY = fH.
7. Switch the control off.
8. Insert the CompactFlash Card into the slot on the front of the control system.
9. Switch the control system back on again.
10. Follow the prompt for the update process.

    Press any button to start the update process.

Note

You can choose not to do this by switching off the control system and then removing the CompactFlash Card.

The update process begins.

11. Once the update process is complete, you will be prompted to switch off the control system and remove the CompactFlash Card.

    Follow the instructions.
12. Switch the control on.

    The control system powers up with the new software version.

    The update is complete.
13. Load the backed up/archived data into the control.

Note

You will usually have to perform the following additional steps:

- Reload SIEMENS cycles and SGUD (note: operating errors may result in loss of data!).
- Update the firmware for SINAMICS components.
- You may also need to modify the user PLC program!

In view of this, it is important to take into account the update instructions for the software version in all cases. If you have any questions relating to the update instructions, please contact the hotline (see the "Technical Support"section in the preface for details).

Licensing in SINUMERIK 802D sl — 16

16.1 Licensing in SINUMERIK 802D sl

SINUMERIK 802D sl licensing

The PCU software on the CNC operator panel (PCU) has already been licensed in the factory before delivery.

Depending on requirements, factory licensing is available for the following technologies:

- SINUMERIK 802D sl T/M Value/Plus/Pro
- SINUMERIK 802D sl G/N Plus/Pro
- SINUMERIK 802D sl C/U Pro

With SW1.4 Service Pack 5 and higher, the following "under license" functions can be purchased for the SINUMERIK 802D sl pro:

- Gantry axes
- Sag compensation and angularity error compensation
- Master/Slave
- Drive variables
- Manual Machine Plus Turning, available for the SINUMERIK 802D sl T/M Plus/Pro

These functions must be activated on the control system via the HMI user interface.

Section Activating optional functions (Page 437) describes how to activate these licensed functions on the control system.

Note

Subsequent licenses can be obtained via the Web License Manager (Page 434).

Internet:http://www.siemens.com/automation/license

16.2 Web License Manager

16.2.1 Web License Manager

By using the Web License Manager, you can assign licenses to hardware in a standard Web browser. To conclude the assignment, the License Key must be entered manually at the control system via the HMI user interface.

Internet address

The Internet address of the Web License Managers is:
http://www.siemens.com/automation/license

16.2.2 Assigning licenses

Requirements

The following prerequisites must be met in order to assign a license to a piece of hardware via direct access and HMI user interface:

- The control system is powered up.
- The login data for direct access (e.g. per CoL) is available:
  - License number
  - Delivery note number
- Type of control system
- "CF card serial number" from the CompactFlash Card system

Operating sequences

1. Establish the "CF card serial number" and the software designation via the HMI licensing dialog: <SYSTEM> operating area > "Service display" > "Version" > "License key".

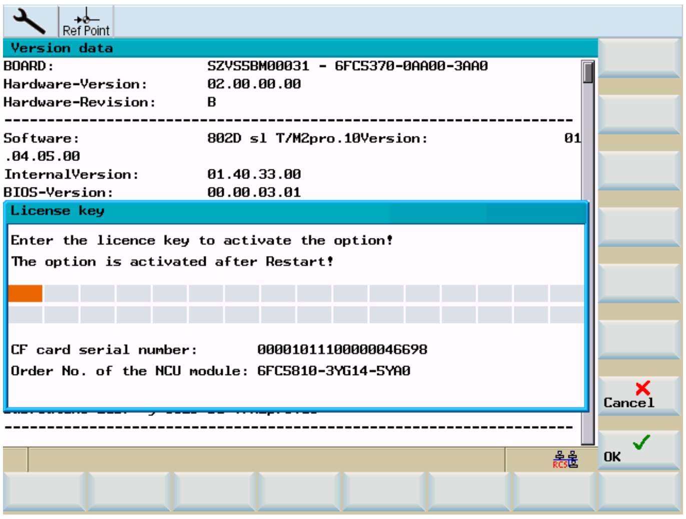


Figure 16-1 "CF card serial number"

Note

Ensure that the "CF card serial number" displayed is also really the one you want to make the assignment for. The assignment of a license to a piece of hardware cannot be reversed via the Web License Manager.

---

2. Go to the Internet page of the Web License Manager:

   http://www.siemens.com/automation/license

3. Login via "Direct access":
   - License number
   - Delivery note number
4. Follow the additional instructions in the Web License Manager.
5. At the end of the process, the Web License Manager shows the license key.

   The following options are now available:
   - Note down the key.
   - Save it in a PDF file.

6. After completing the assignment process, enter the License Key displayed on the Web License Manager into the licensing dialog of the HMI user interface.

   In the <SYSTEM> operating area, select "Service display" > "Version" > "License key".

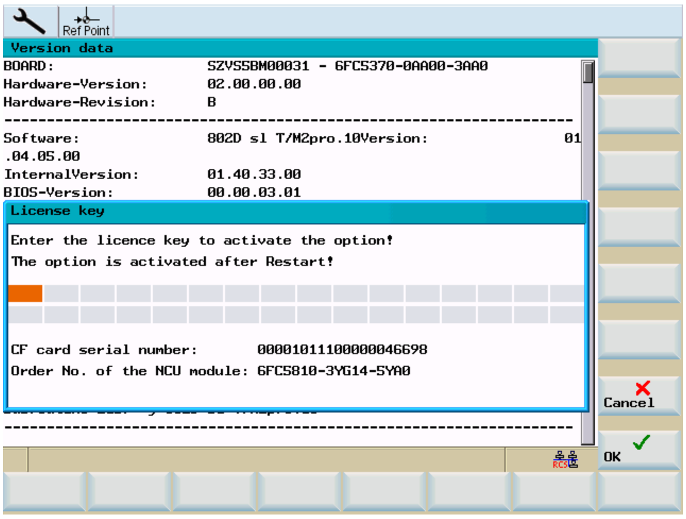


Figure 16-2 License key

7. Press the "OK" softkey to confirm the entry for the new license key.
8. Activate (Page 437) the optional functions.

16.3 Activating optional functions

Operating sequence

SYSTEM ALARM

Service display

Version

In the <SYSTEM> operating area, select "Service display" > "Version".

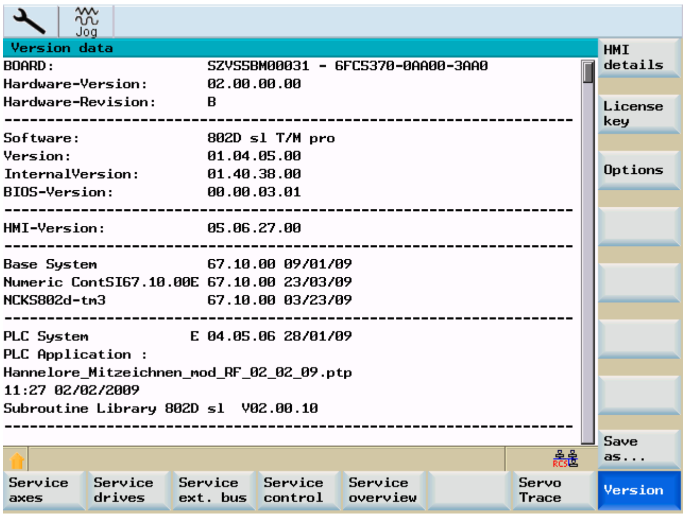


Figure 16-3 Version data

Note

The versions shown on the version screen provide an example of what you may find.

Options

Press "Options".

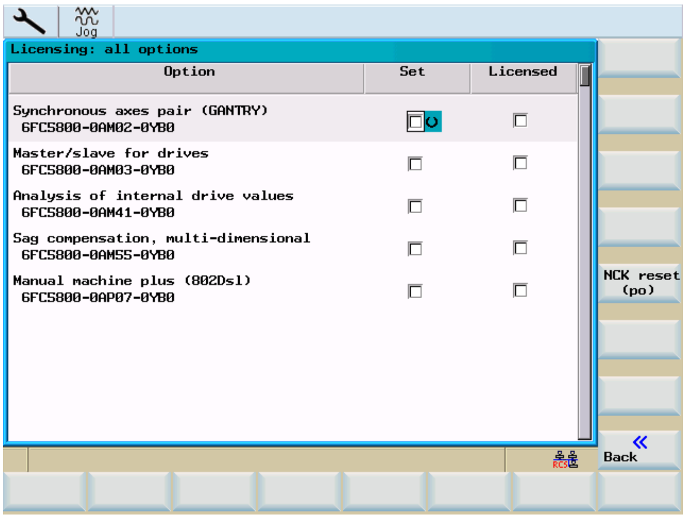


Figure 16-4 Options

Set the licensed options.

NCK reset (po)

Press "NCK reset". A warm restart is triggered on the control system.

The licensed options are set and can be used.

Note

If an option is activated without a valid license, alarm 8081 is output: "1 option(s) that has (have) not been licensed using a license key was (were) set". It will not be possible to operate the machine as normal.

16.4 Internet links

Overview of Internet links used:

No.	Topic	Address
1	Web License Manager	http://www.siemens.com/automation/license
2	Siemens A&D Mall: Customer login	http://mall.automation.siemens.com/
3	Download server	http://software-download.automation.siemens.com

16.5 Import licensing terms

The terms below are important for understanding the license management of SINUMERIK software products.

Term	Description
Software product	"Software product" is generally used to describe a product that is installed on a piece of hardware to process data. Within the license management of SINUMERIK software products, a corresponding license is required to use each software product.
Hardware	In the context of license management of SINUMERIK software products, "hardware" refers to the component of a SINUMERIK control system to which licenses are assigned on the basis of its unique identifier. License information is also saved to remanent memory on this component. • SINUMERIK 802D sl: CompactFlash Card system
License	A license gives the user a legal right to use the software product. Evidence of this right is provided by the following: • CoL (Certificate of License) • License key
CoL (Certificate of License)	The CoL is the proof of the license. The product may only be used by the holder of the license or authorized persons. The CoL includes the following data relevant for the license management: • Product name • License number • Delivery note number • Hardware serial number **Note:** The hardware serial number is only found on a system software CoL or is only available if a bundled license was ordered, in other words the system software included options.
License number	The license number is the feature of a license that is used for its unique identification.
CompactFlash Card system	The CompactFlash Card system represents, as the carrier of all the retentive data of a SINUMERIK solution line control system, the identity of this control system. The CompactFlash Card system includes the following data that is of relevance to license management: • Hardware serial number • License information including the License Key
Hardware serial number	The hardware serial number is a permanent part of the CompactFlash Card system. It is used to identify a control system uniquely. The hardware serial number can be determined by: • CoL (see: Certificate of License > "Note") • HMI user interface (SYSTEM operating area > "Service display" > "Version" > "License key") • Printing on the CompactFlash Card system
License key	The License Key is the "technical representative" of the sum of all the licenses that are assigned to one particular piece of hardware, which is uniquely marked by its hardware serial number.

Term	Description
Option	One option is a SINUMERIK software product that is not contained in the basic version and which requires the purchase of a license for its use.
Product	A product is marked by the data below within the license management of SINUMERIK software products: • Product designation • Order number: • License number

17 Technical data

17.1 technical specifications

User data memory

CompactFlash card, type 1 (CF card)

Connected loads of the PCU

Table 17- 1 Connected loads

Supply voltage	24 V DC (permissible range: 20.4...28.8 V)
Ripple	3.6 Vpp
Current consumption from 24 V	Basic configuration typically 1.5 A (inputs/outputs open)
Power loss • CNC operator panel (PCU) with full CNC keyboard • Machine control panel • PP72/48 I/O module	max. 50 W ≤ 5 W max. 11 W
Starting current, total	5 A

Dimensions and weight

Table 17- 2 Dimensions and weight

CNC operator panel (PCU)	
Dimensions W × H × D (mm)	310 x 330 x 85 310 x 330 x 101 with MCPA module
Weight [g]	approx. 4,900
Full CNC keyboard (horizontal format)	
Dimensions W × H × D (mm)	310 175 32
Weight [g]	approx. 1,700
Full CNC keyboard (vertical format)	
Dimensions W × H × D (mm)	172 x 330 x 32
Weight [g]	approx. 1,700
Machine control panel	
Dimensions W × H × D (mm)	170 x 330 x 128
Weight [g]	approx. 1,500

PP72/48 I/O module	
Dimensions W × H × D (mm)	194 x 325 x 35
Weight [g] • without mounting plate • with mounting plate	 • approx. 300 • approx. 1,200
MCPA module	
Dimensions W × H × D (mm)	89 x 205 x 68
Weight [g]	approx. 300

Torque

Note

Mounting the CNC operator panel (PCU), machine control panel and full CNC keyboard

The maximum permissible tightening torque for the fixing screws is 1.8 Nm and this value must not be exceeded.

Colors

Table 17- 3 Colors

CNC operator panel (PCU), machine control panel (MCP) and full CNC keyboard	SIEMENS COLOR Anthracite 614, STRUCTURE 39, VDI 3400	
Button background	Mid-gray buttons	Light Basic 700 or
	Dark buttons	Medium Basic 701
	Yellow buttons	Pantone Yellow
	Blue buttons	RAL 6034
	Red buttons	Pantone 185 U
	Green buttons	Pantone 345 C
	Icons	RAL 9017
Button foreground	Univers S55/57	

Digital inputs of the PP72/48 I/O module (as per IEC 1131-2/DIN EN 61131-2, type 2 characteristic)

Table 17- 4 Digital inputs of the PP72/48 I/O module

Number of inputs	24 each per terminal strip converter			
parameters	min.	Standard	Maximum	Nominal
Voltage with high level (U_H)	15 V	1)	30 V	24 V
Input current I_{In} at U_H	2 mA	-	15 mA	-
Voltage with low level (U_L)	-30 V	-	5 V	0 V
Signal delay time T_{PHL} 2)	0.5 ms	-	3 ms	-
1) Supply voltage of the digital inputs typical output voltage: V_{CC} - I_{OUT} * R_{ON} V_{CC}: current operating voltage ($P24OUT_{INT}$) to X111, X222, X333: Pin 2 2) In addition, take into account the PROFIBUS-DP communication time and the application cycle time. Incorrect connection causes neither high level nor destruction of the inputs.				

Digital outputs of the PP72/48 (as per IEC 11312/DIN EN 611312)

Table 17- 5 Digital outputs of the PP72/48 I/O module

Number of outputs	16 each per terminal strip converter			
parameters	min.	Standard	Maximum	Nominal
Voltage with high level (U_H)	V_{CC} - 3 V	1)	V_{CC}	24 V
I_{out}	-	-	0.25 A	-
Voltage with low level (U_L)	-	-	-	Output open
Leakage current at low level	-	50 µA	400 µA	-
Signal delay time T_{PHL} 2)	-	-	0.5 ms	-
max. switching frequency 2) • resistive load • inductive load • Lamp	100 Hz 2 Hz 11 Hz	- - -	- - -	- - -
1) Supply voltage of the digital outputs typical output voltage: V_{CC} - I_{OUT} * R_{ON} V_{CC}: Current operating voltage Max. output current I_{OUT}: 0.25 A max. short-circuit current: 4 A (max. 100 µs, V_{CC} = 24 V) Inner flow resistance R_{ON}: 0.4 Ω 2) In addition, take into account the PROFIBUS-DP communication time and the application cycle time. Incorrect connection causes neither high level nor destruction of the outputs.				

General electric features:

- Galvanic isolation using optocouplers
- Current limited to max. 0.25 A
- Protection against:
  - short-circuit
  - Overtemperature
  - loss of grounding
- Automatic disconnection in case of undervoltage

17.2 DMC20 technical data

Table 17- 6 Technical specifications of the DMC20

	Unit	Value
Electronics power supply Voltage Current (without DRIVE-CLiQ or digital outputs)	 V_{DC} A_{DC}	 24 DC (20.4 – 28.8) 0,5
PE/ground connection	At the housing with M4/1.8 Nm stud	
Weight	kg	0,8

17.3 SMC10 technical data

Table 17- 7 Technical data

Sensor Module Cabinet-Mounted SMC10 6SL3055-0AA00-5AAx	Designation	Unit	Value
Electronics power supply Voltage Current (without encoder system) Current (with encoder system) Power loss	 V_{DC} A_{DC} A_{DC} W	 V A A W	 24 DC (20.4 – 28.8) ≤ 0.20 ≤ 0.35 ≤ 10
Specification Transmission ratio (ü) of the resolver Excitation voltage on the SMC10 when ü=0.5 Amplitude monitoring threshold (secondary tracks) of the SMC10	 ü = V_{rms} V_{rms}	 V V	 0.5 4.1 1
Excitation voltage (cannot be parameterized)	V_{rms}	V	4.1
Excitation frequency (synchronized to the current controller clock cycle)		kHz	5 to 10
PE/ground connection		On housing with M4 / 1.8 Nm screw	

Sensor Module Cabinet-Mounted SMC10 6SL3055-0AA00-5AAx	Designation	Unit	Value
Max. encoder cable length		m	130
Weight		kg	0.8
Degree of protection		IP20 or IPXXB	

Table 17- 8 Max. frequency that can be evaluated (speed)

Resolver		Max. speed resolver / motor		
Number of poles	Number of pole pairs	8kHz/125 µsec	4kHz/250 µsec	2kHz/500 µsec
2-pole	1	120,000 rpm	60,000 rpm	30,000 rpm
4-pole	2	60,000 rpm	30,000 rpm	15,000 rpm
6-pole	3	40,000 rpm	20,000 rpm	10,000 rpm
8-pole	4	30,000 rpm	15,000 rpm	7,500 rpm

The ratio between the ohmic resistance R and the inductance L (the primary winding of the resolver) determines whether the resolver can be evaluated with the SMC10. See the following diagram:

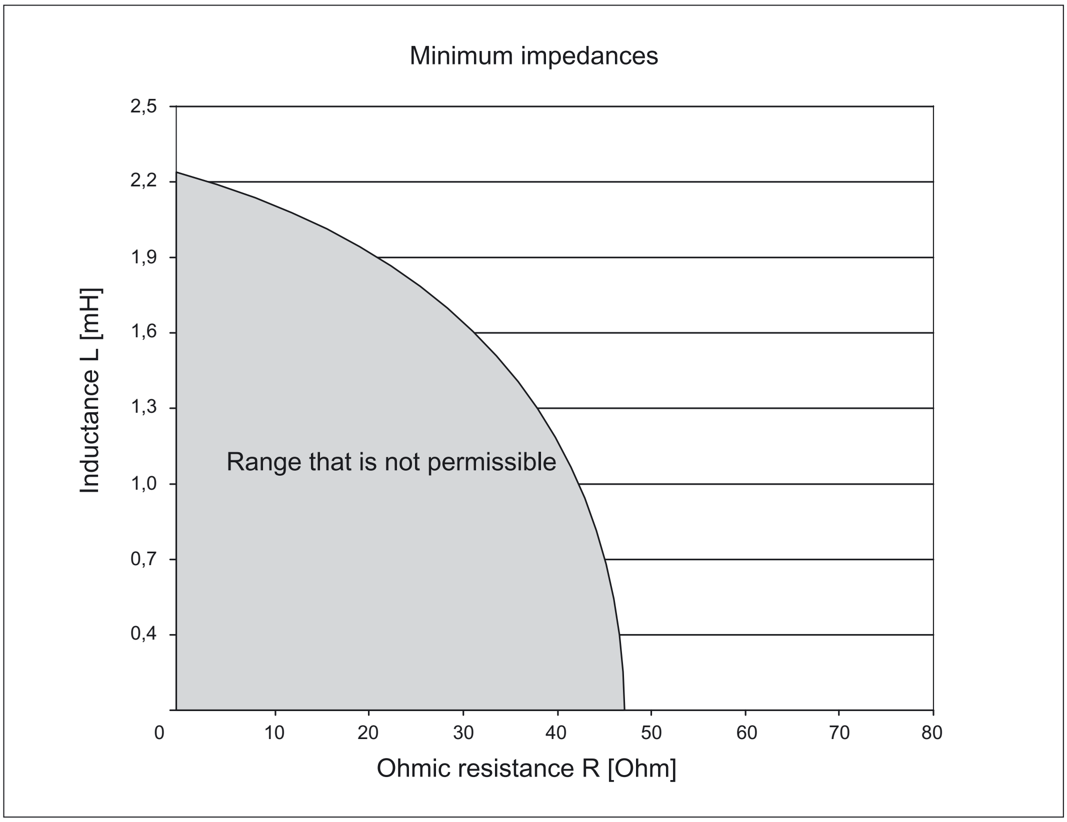


Figure 17-1 Connectable impedances with an excitation frequency f = 5000 Hz

17.4 SMC20 technical data

Table 17- 9 Technical specifications

Sensor Module Cabinet-Mounted SMC20 6SL3055-0AA00-5BAx	Designation	Unit	Value
Electronics power supply Voltage Current (without measuring system) Current (with measuring system) SSI baud rate Power loss	 V_{DC} A_{DC} A_{DC} kHz W	 V A A kHz W	 24 DC (20.4 – 28.8) ≤ 0.20 ≤ 0.35 100 ≤ 10
Measuring system power supply Voltage Current	 $V_{encoder}$ $A_{encoder}$	 V A	 5 V DC (with Remote Sense) 0.35
Encoder frequency that can be evaluated	$f_{encoder}$	kHz	≤ 500
PE/ground connection		On housing with M4 / 1.8 Nm screw	
Weight		kg	0,8
Degree of protection		IP20 or IPXXB	

17.5 SMC30 technical data

Table 17- 10 Technical specifications

Sensor Module Cabinet-Mounted SMC30 6SL3055-0AA00-5CAx	Designation	Unit	Value
Electronics power supply Voltage Current (without measuring system) Current (with measuring system) SSI baud rate Power loss	 V_{DC} A_{DC} A_{DC} kHz W	 V A A kHz W	 24 DC (20.4 – 28.8) ≤ 0.20 ≤ 0.35 100 - 250 ≤ 10
Measuring system power supply Voltage Current	 $V_{encoder}$ $A_{encoder}$	 V A	 5 V DC (with or without Remote Sense)[1] or V_{DC} - 1 V 0.35
Encoder frequency that can be evaluated	$f_{encoder}$	kHz	≤ 500
PE / ground connection		On housing with M4 / 1.8 Nm screw	
Weight		kg	0,45 (Order No. 6SL3055-0AA00-5CA2) 0,8 (Order No. 6SL3055-0AA00-5CA0, 6SL3055-0AA00-5CA1)
Degree of protection		IP20 or IPXXB	

[1] Remote Sense only at X520

Table 17- 11 Specification, measuring systems that can be connected

Parameter	Designation	Threshold	Min.	Max .	Unit
High signal level (TTL bipolar at X520 or X521/X531)[1]	U_{Hdiff}		2	5	V
Low signal level (TTL bipolar at X520 or X521/X531)[1]	U_{Ldiff}		-5	-2	V
Signal level high (HTL unipolar)	U_H[4]	High	17	V_{CC}	V
		Low	10	V_{CC}	V
Signal level low (HTL unipolar)	U_L[4]	High	0	7	V
		Low	0	2	V
High signal level (HTL bipolar)[2]	U_{Hdiff}		3	V_{CC}	V
Low signal level (HTL bipolar)[2]	U_{Ldiff}		$-V_{CC}$	-3	V
High signal level (SSI bipolar at X520 or X521/X531)[1][3]	U_{Hdiff}		2	5	V
Low signal level (SSI bipolar at X520 or X521/X531)[1][3]	U_{Ldiff}		-5	-2	V
Signal frequency	f_S		-	500	kHz
Edge clearance	t_{min}		100	-	ns
Zero pulse inactive time (before and after A=B=high)	t_{Lo}		500	$(t_{ALo\text{-}BHi} - t_{Hi})/2$ [5]	ns
Zero pulse active time (while A=B=high and beyond)	t_{Hi}		500	$t_{ALo\text{-}BHi} - 2{*}t_{Lo}$ [5]	ns

[1] Other signal levels according to the RS422 standard.

[2] The absolute level of the individual signals varies between 0 V and V_{CC} of the measuring system.

[3] Only from Order No. 6SL3055-0AA00-5CA1 and Firmware 2.4.

[4] Only from Order No. 6SL3055-0AA00-5CA2 and Firmware 2.5 SP1 this value can be configured using software. For older firmware releases and Order Nos. less than 6SL3055-0AA00-5CA2 then the "low" threshold applies.

[5] $t_{ALo\text{-}BHi}$ is not a specified value, but is the time between the falling edge of track A and the next but one rising edge of track B.

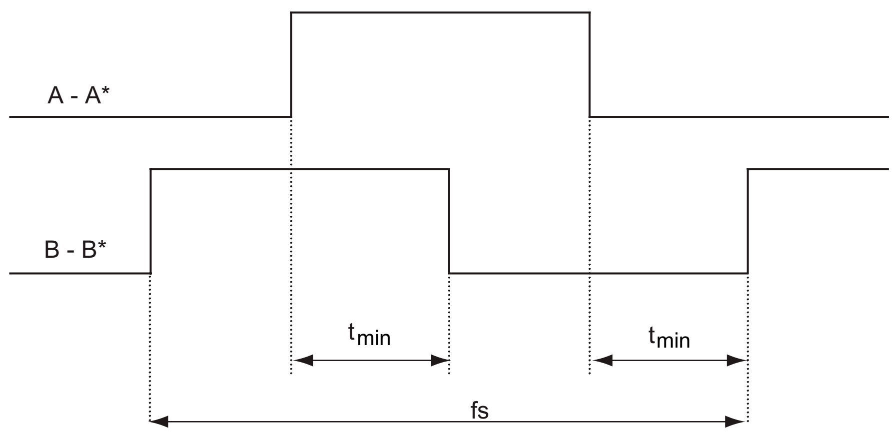


Figure 17-2 Signal characteristic of the A and B track between two edges: Time between two edges with pulse encoders

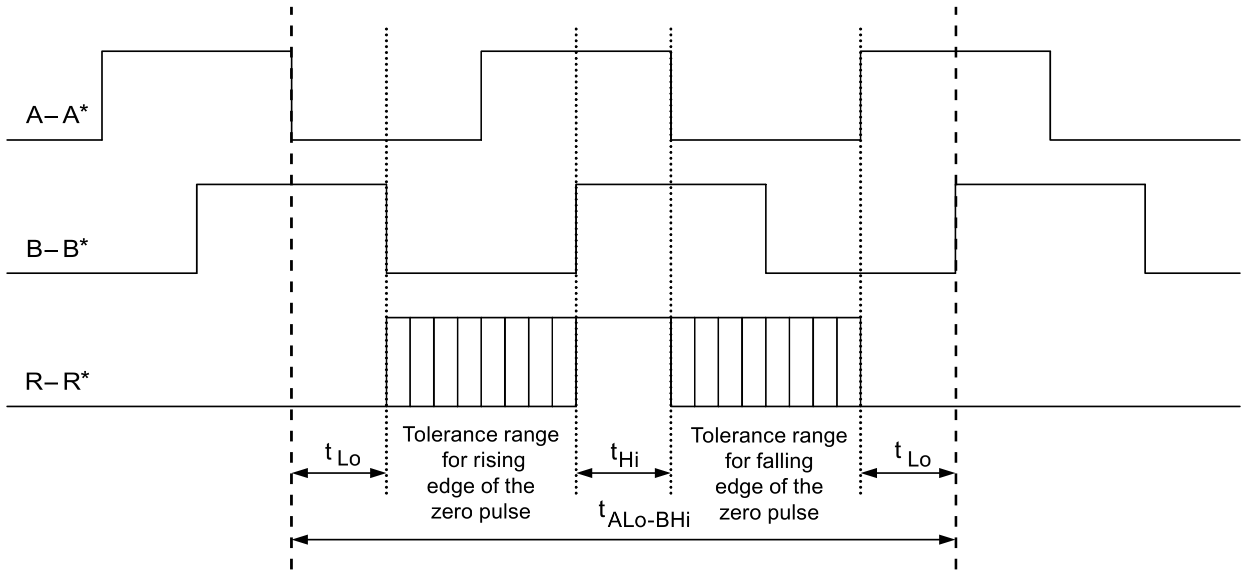
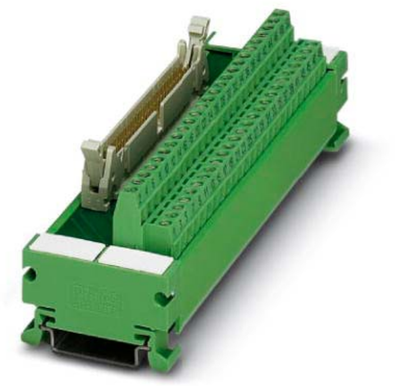
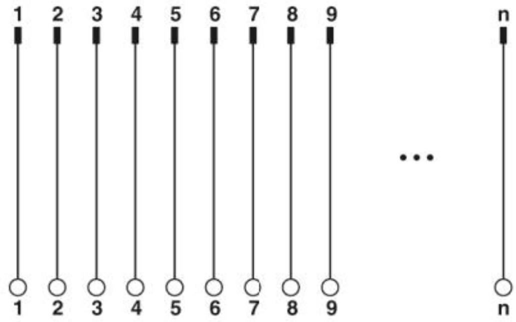


Figure 17-3 Position of the zero pulse to the track signals

17.6 Technical data, terminal strip converter

Terminal strip converter





MLFB: 6EP5406-5AA00

1: 1 interconnection

Technical data

Parameter	Values
Rated voltage U_N	24 V AC 60 V DC
Max. current carrying capacity per branch	1 A
Number of poles	50
Dimensions (WxHxD)	151 x 50 x 40 mm
Weight, approx.	0.15 kg
Ambient temperature (in operation)	-20 ... 55°C
Ambient temperature during storage/transportation	-40 ... 70°C
Mounting position	Any
Degree of pollution	2 according to 61800-5-1
Protection class	II acc. to EN 61800-5-1
Applicable standards	IEC 60664 DIN VDE 0110

17.7 Electro-Magnetic Compatibility

Definition

Electromagnetic compatibility refers to the capability of electrical equipment in reliably performing its dedicated function in an electromagnetic environment, without causing interference in the same environment.

Emission of Radio Interferences

Table 17- 12 Interference emission of electromagnetic fields as per EN 55011: Limit value class A, group 1.

between 20 and 230 MHz	<30 dB (µV/m) Q
between 230 and 1000 MHz	<37 dB (µV/m) Q
measured at a distance of 30 m	

Table 17- 13 Interference emission via network alternating current supply in accordance with EN 55011: Limit value class A, group 1.

between 0.15 and 0.5 MHz	<79 dB (µV) Q <66 dB (µV) M
between 0.5 and 5 MHz	<73 dB (µV) Q <60 dB (µV) M
between 5 and 30 MHz	<73 dB (µV) Q <60 dB (µV) M

Extension of the range of application

If you intend to use the control system in residential areas, you must ensure that the control system meets the requirements of limit value class B to EN 55011 in respect of interference emission.

Recommendation: Install the control system in grounded metal cabinets, such as 8MC cabinets (see NV 21 Catalog). Connect filters to the supply lines.

17.8 Transport and storage conditions

The following data applies to modules that are transported or stored in the original packaging.

Table 17- 14 Shipping and storage conditions

Type of condition	Permissible range
Free fall	≤ 1m
Temperature	From -20 °C to +60 °C
Atmospheric pressure	1,060 to 700 hPa (corresponds to an altitude of 3,000 m)
Relative humidity	5% to 95%, without condensation

17.9 Ambient operating conditions for the operation

Conditions of use

The control system is intended for use as a stationary equipment in a sheltered environment. The conditions of use are compliant with requirements to DIN IEC 68–2–2:

The control system satisfies the operating conditions of the 3C3 class in accordance with DIN EN 607213-3 (operating locations with high traffic densities and in the immediate vicinity of industrial plants with chemical emissions).

The control system must not be operated without additional measures being taken

- in locations with a high proportion of ionizing radiation
- In locations associated with aggressive operating conditions characterized, for example, by:
  - dust
  - caustic vapor or gases.
- In installations requiring special monitoring, such as:
  - Elevator systems
  - electrical equipment in especially hazardous rooms.

An additional requirement for using the control system may be, for example, installation in cabinets.

Climatic environmental conditions

The control system can be used under the following climatic ambient conditions:

Table 17- 15 Climatic environmental conditions

Environmental conditions	Fields of application	Remarks
Temperature	0 to 50 °C	with a simultaneity of 50 %
Relative humidity	from 5% to 95 %	Without condensation, corresponds to relative humidity (RH) severity level 2 in accordance with IEC 1131-2
Atmospheric pressure	from 1,080 to 795 hPa	-
Concentration of pollutants	SO_2: <0.5 ppm; Relative humidity <60 %, no condensation H_2S: <0.1 ppm; Relative humidity <60 %, no condensation	Test: 10 ppm; 4 days 1 ppm; 4 days

Mechanical ambient conditions

The mechanical ambient conditions for the control system are specified in the table below in the form of sinusoidal waves.

Table 17- 16 Mechanical ambient conditions

Mechanical ambient conditions	Operation	Transport (in packaging)
Vibration tested according to DIN EN 60068–2–68	10..0.58 Hz: 0.35 mm 58 to 200 Hz: 50 m/s^2	5...9 Hz: 3.5 mm 9 to 200 Hz: 10 m/s^2
Shock resistance tested according to DIN EN 60068–2–27	10 g peak value, 6 ms duration 100 shocks in each of the 3 axes vertical to one another	10 g peak value, 6 ms duration 100 shocks in each of the 3 axes vertical to one another

Reduction of vibration

If the control system is subjected to major impacts or vibration, appropriate measures must be taken to reduce the acceleration or the amplitude of the vibration.

We recommend installation on shock-absorbing material (e.g., rubber-metal vibration dampers).

17.10 Specifications for Protection Class and Degree of Protection

Protection class

Safety class I according to DIN DIN EN 61140, i.e. protective conductor connection required!

Protection against foreign bodies and water

Degree of protection per DIN EN 60529:

- CNC operator panel (PCU) IP65 (front)
  IP20 (rear)
- Machine control panel (MCP) IP54 (front)
  IP00 (rear)
- PP72/48 I/O module IP00

Dimensional Drawings 18

18.1 CNC operator panel (PCU) dimension drawing and hole drilling template

Note

Dimensions marked with 1) are minimum clearances to adjacent modules.

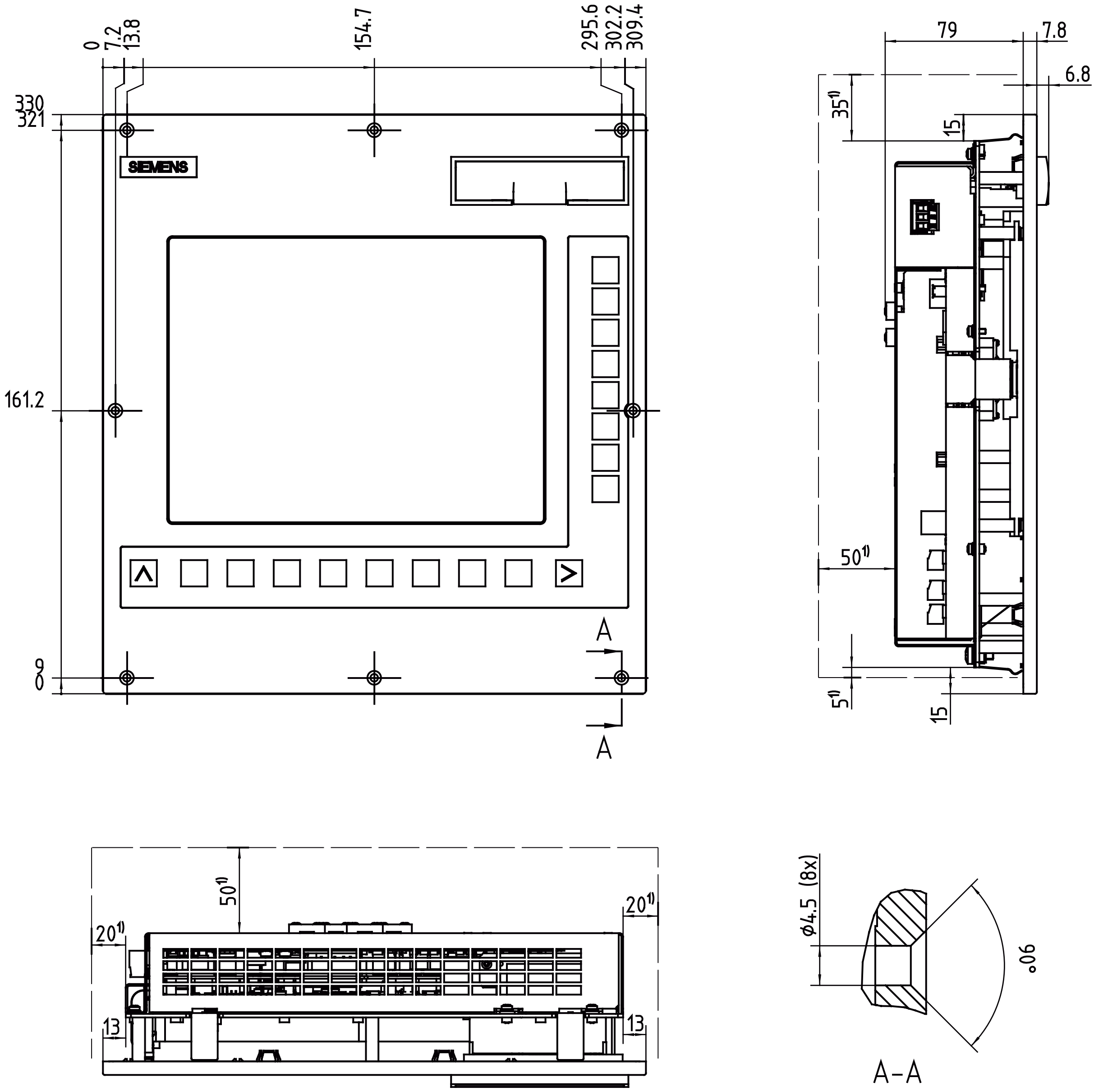


Figure 18-1 CNC operator panel (PCU) dimensional drawing

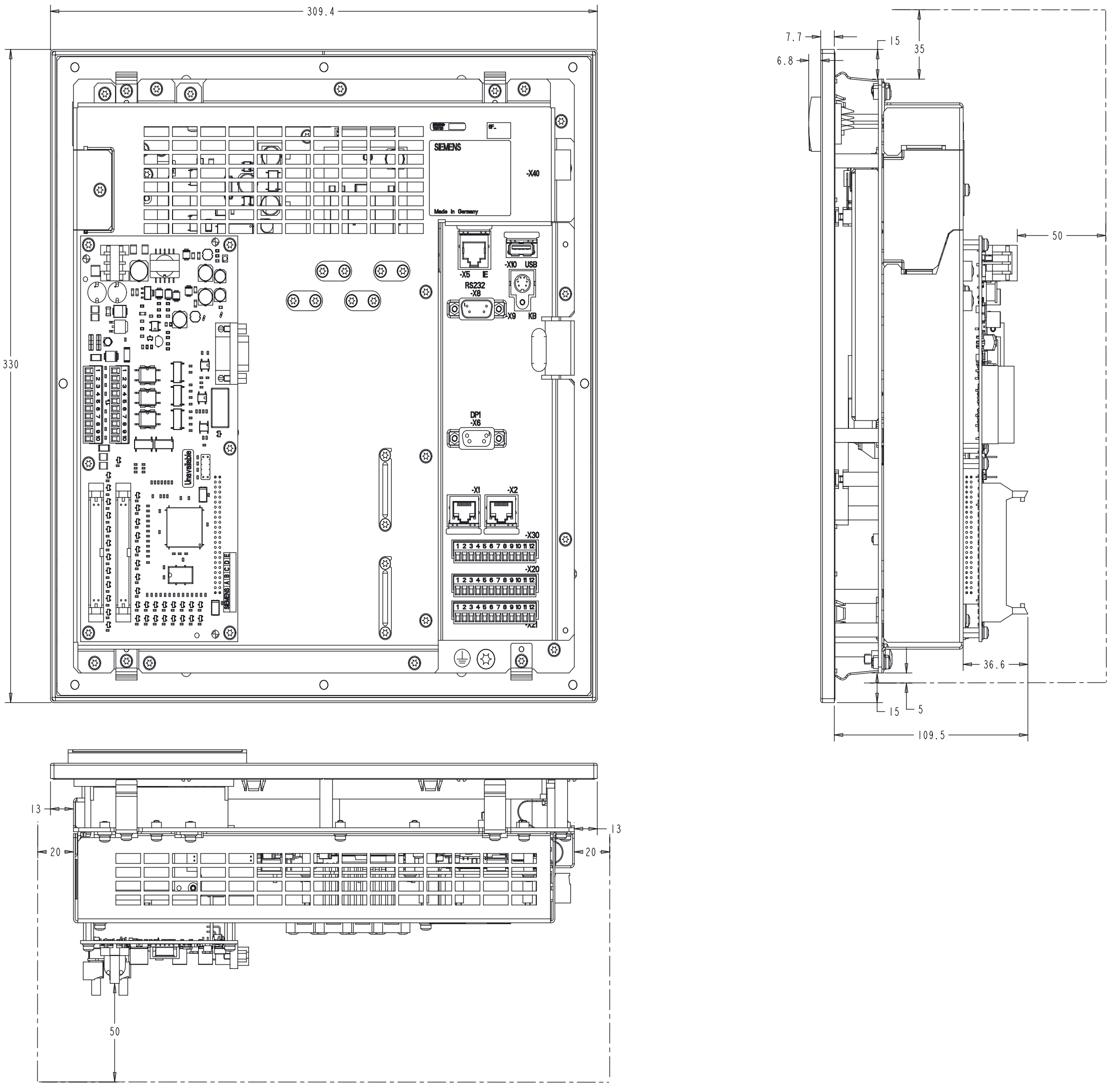


Figure 18-2 Dimensional drawing CNC operator panel with MCPA module

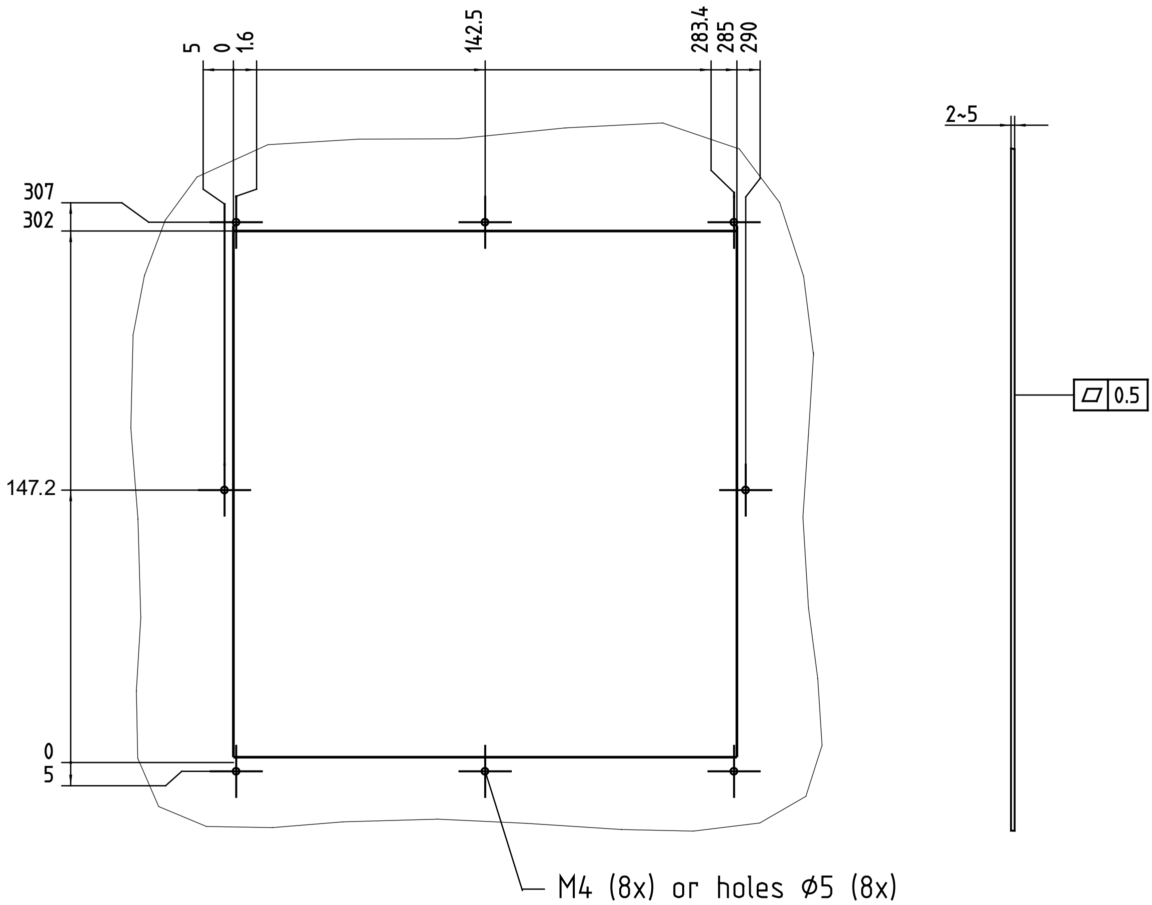


Figure 18-3 CNC operator panel (PCU) hole drilling template

18.2 Machine control panel (MCP) dimension drawing and hole drilling template

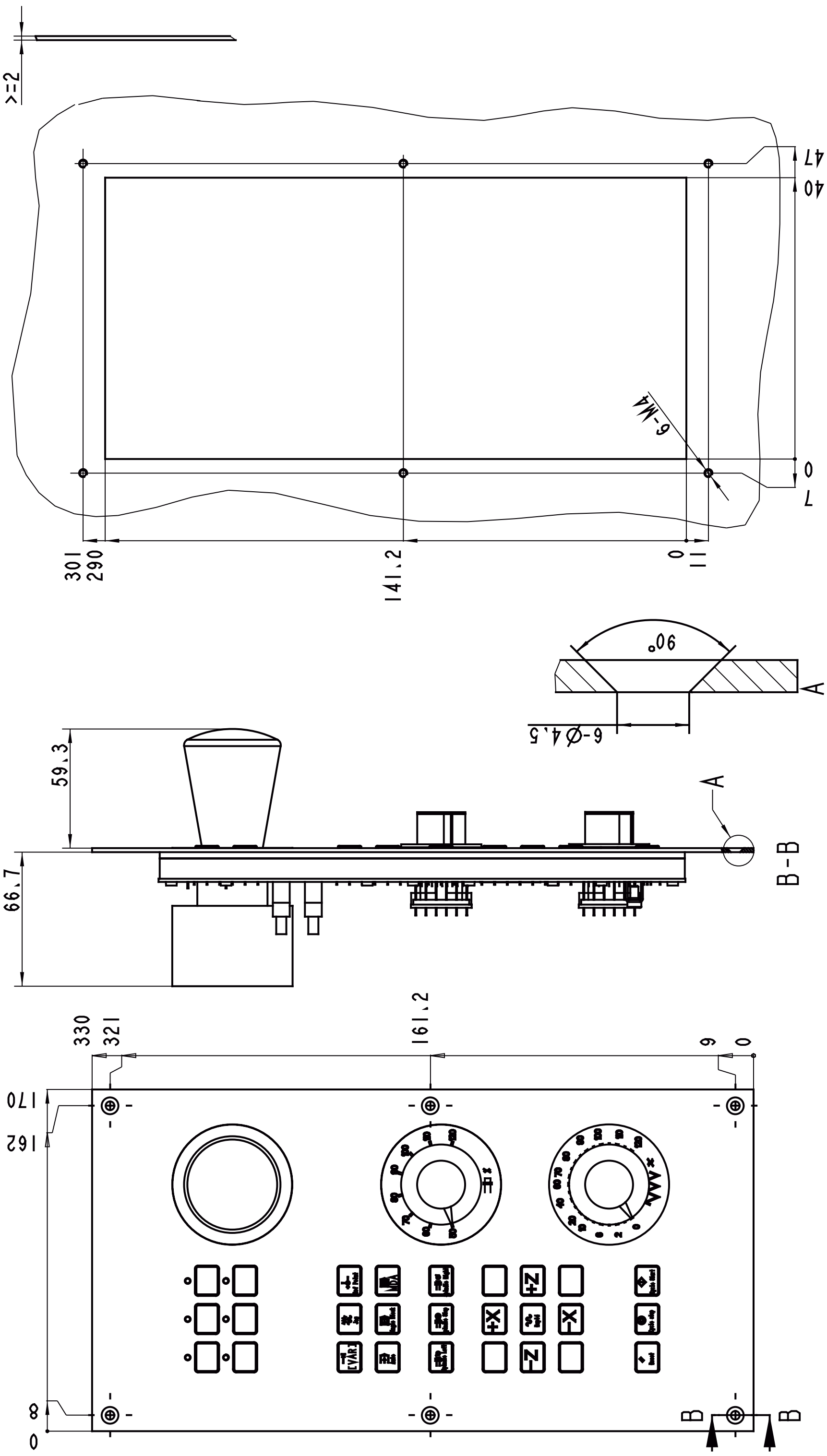


Figure 18-4 Dimension and hole drilling template of the machine control panel MCP

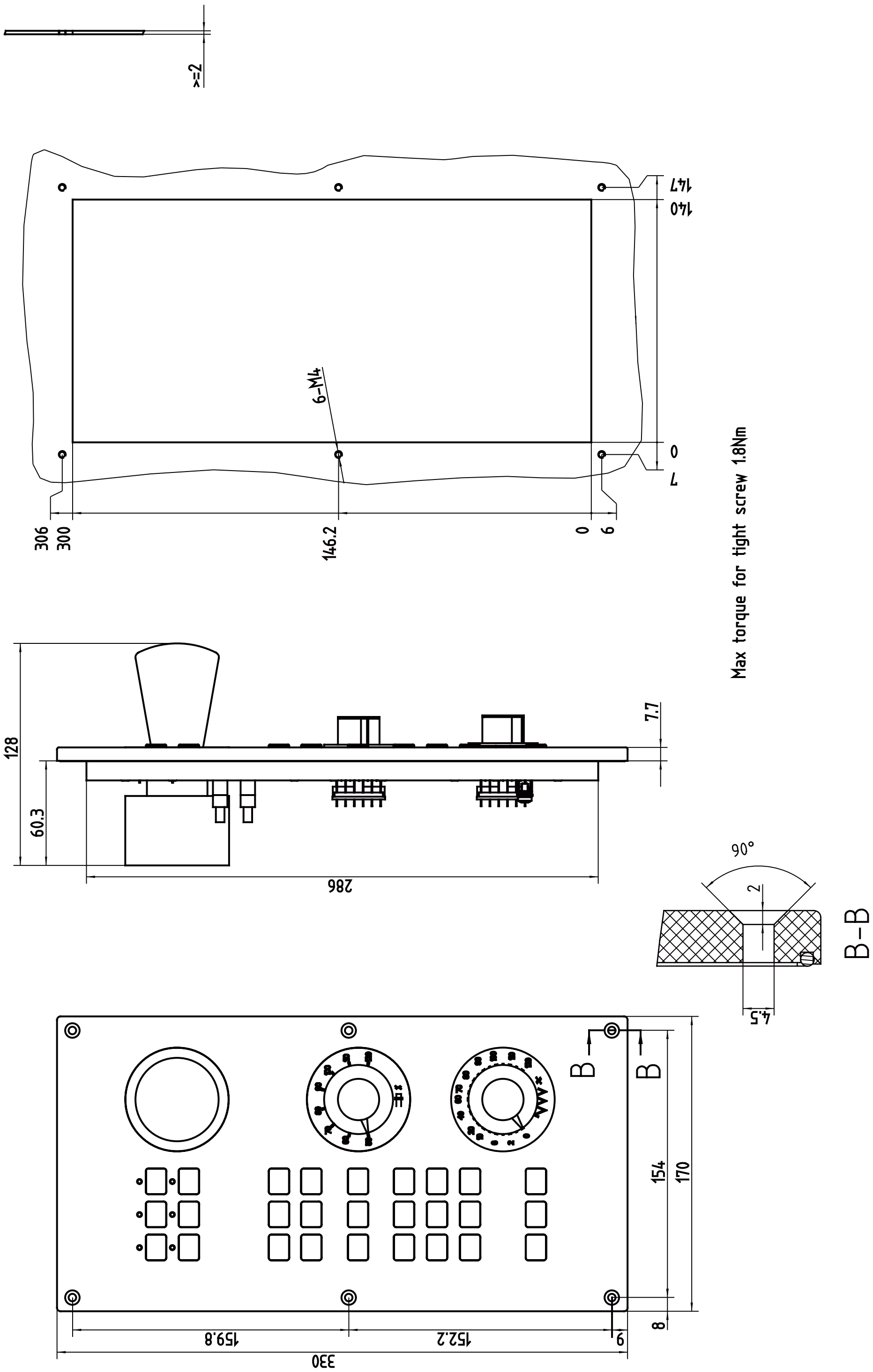


Figure 18-5 Dimension and hole drilling template of the machine control panel MCP 802D sl

18.3 Dimensional drawings and drilling templates of the CNC full keyboard

Dimensional drawings and drilling templates of the CNC full keyboard (mounting next to the PCU)

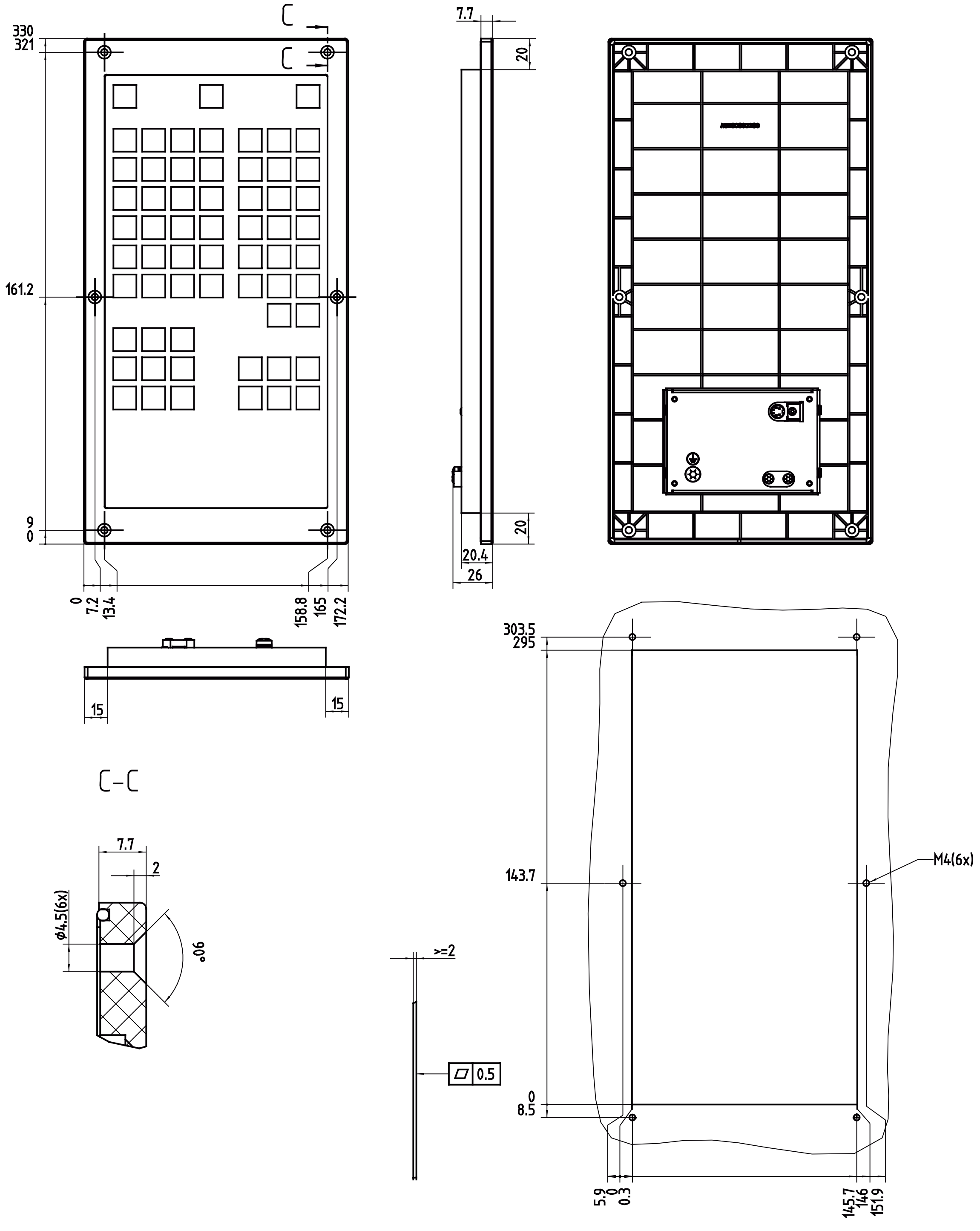


Figure 18-6 Dimensional drawings and drilling templates of the CNC full keyboard (mounting next to the PCU)

Dimensional drawings and drilling templates of the CNC full keyboard (mounting below the PCU)

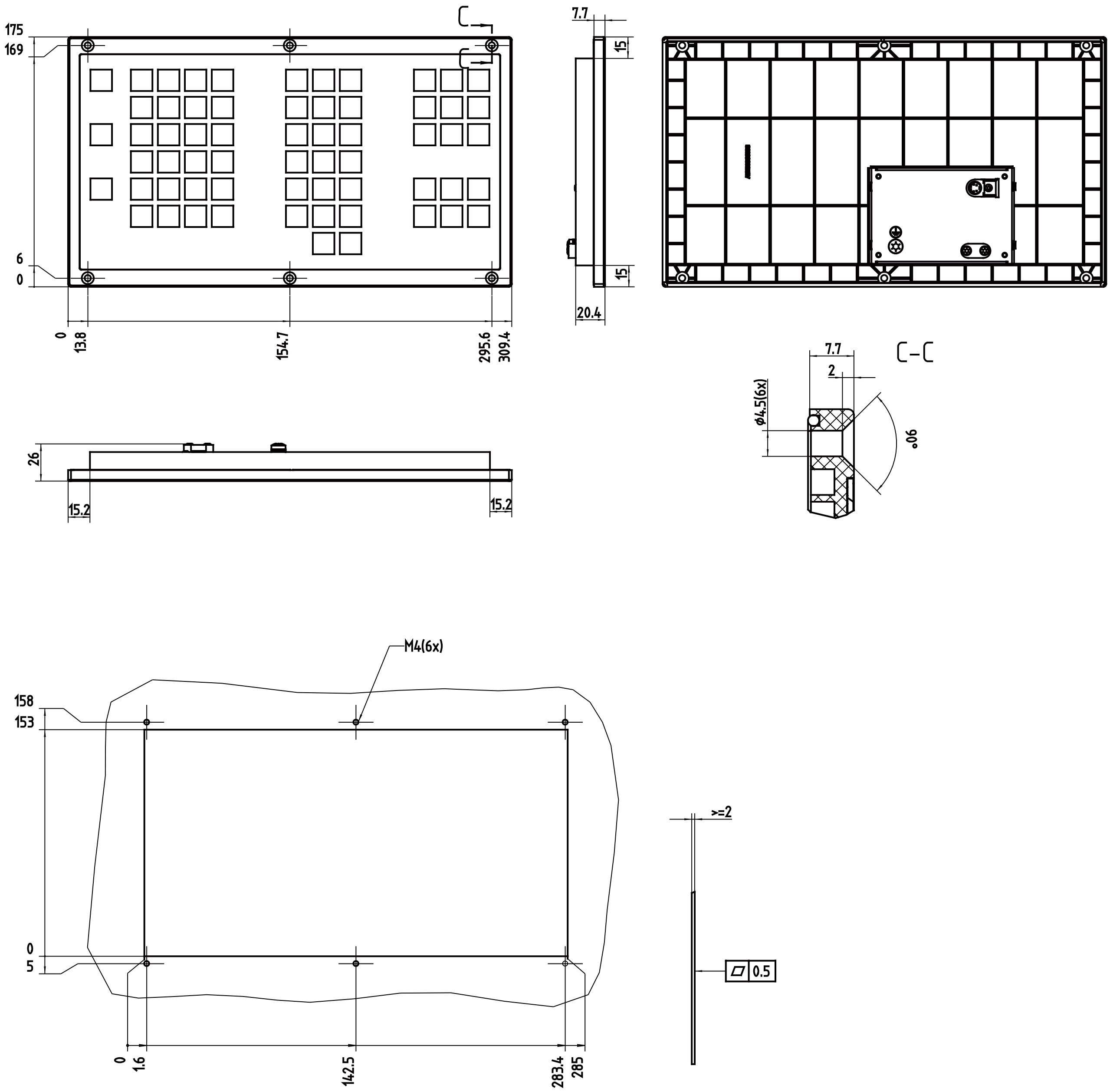


Figure 18-7 Dimensional drawings and drilling templates of the CNC full keyboard (mounting below the PCU)

18.4 PP72/48 I/O module dimension drawing

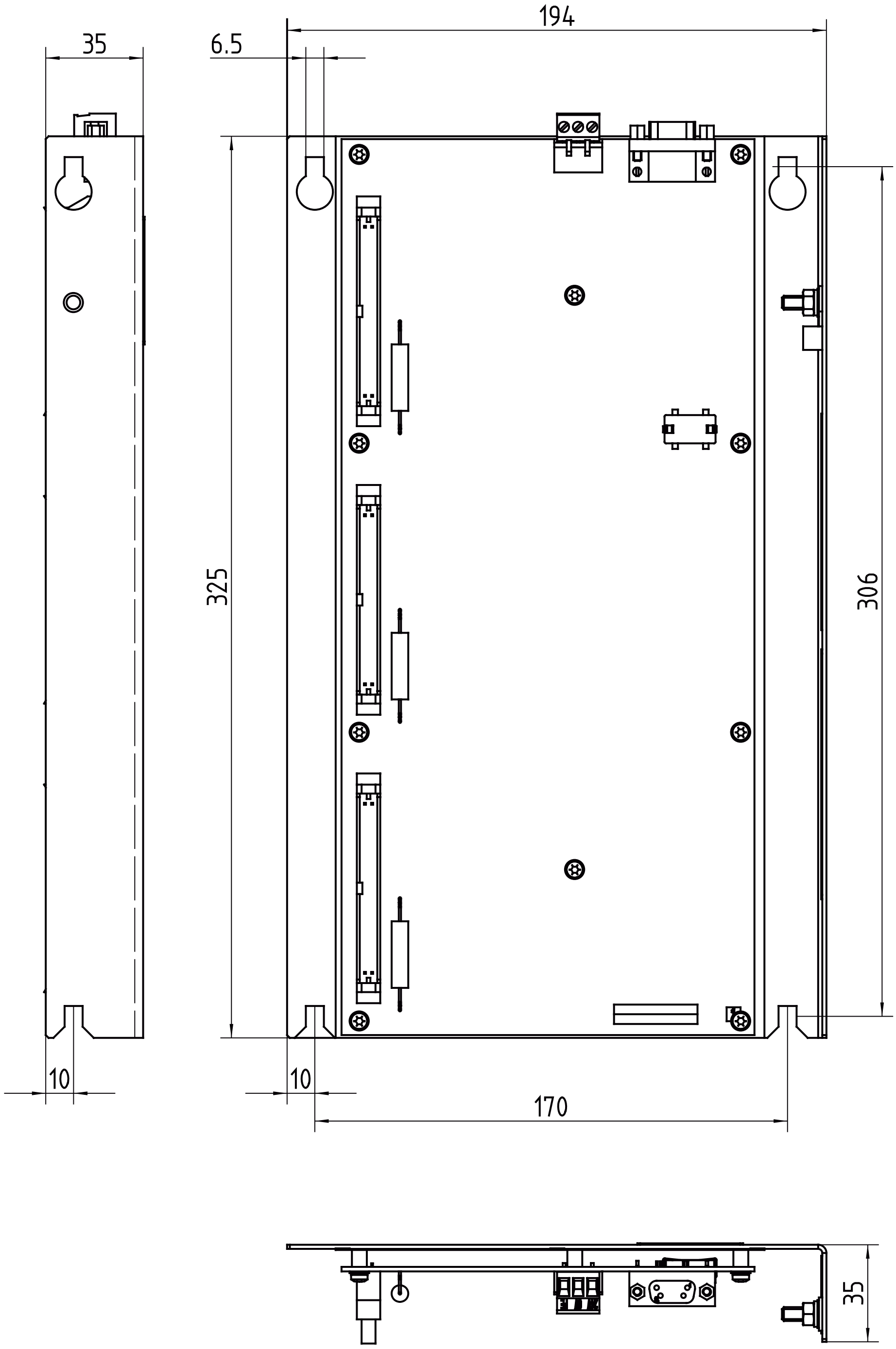


Figure 18-8 PP72/-48 peripheral module dimensional drawing

18.5 Dimensional drawing MCPA module

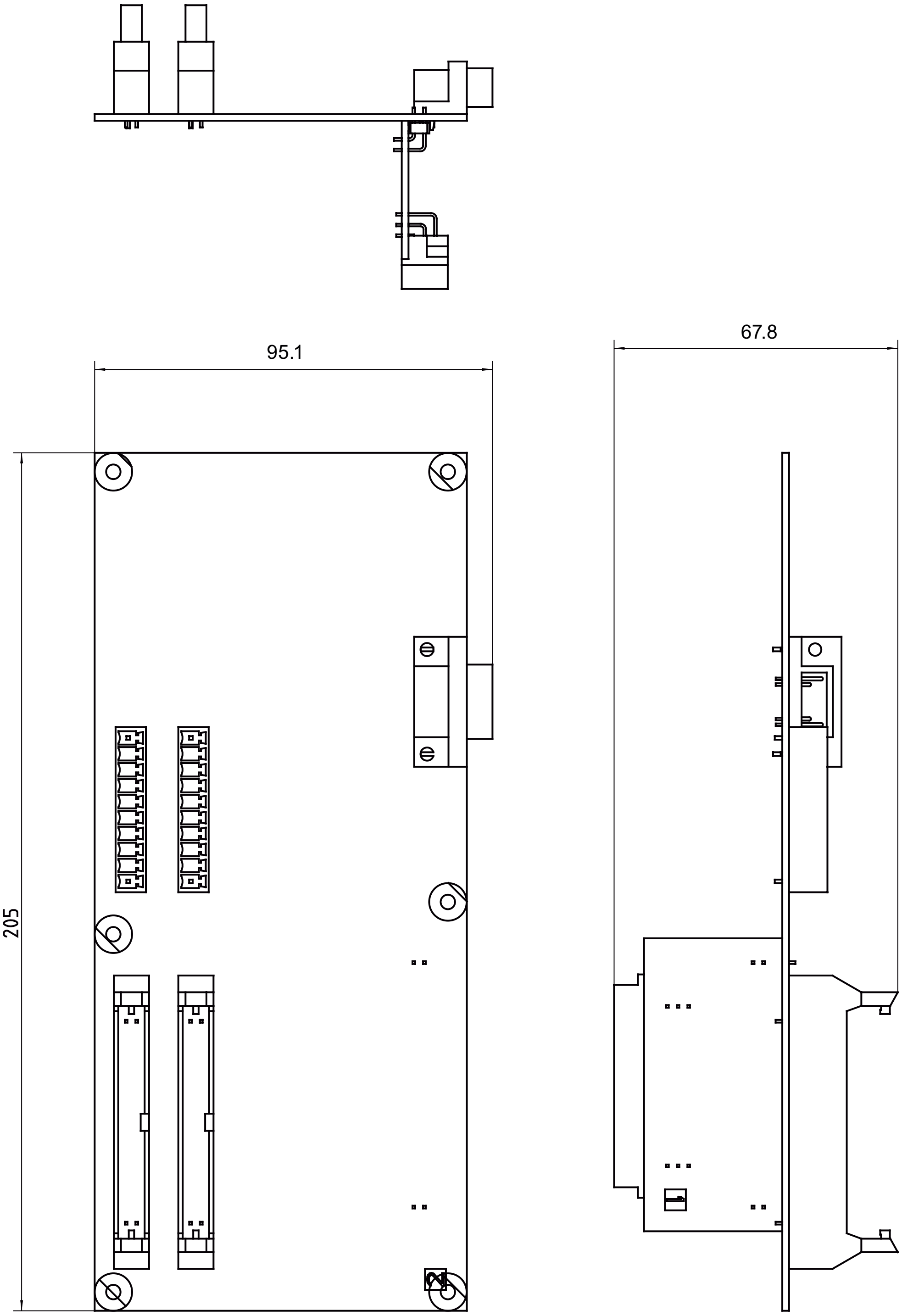


Figure 18-9 Dimensional drawing MCPA module

18.6 ADI4 dimension drawing

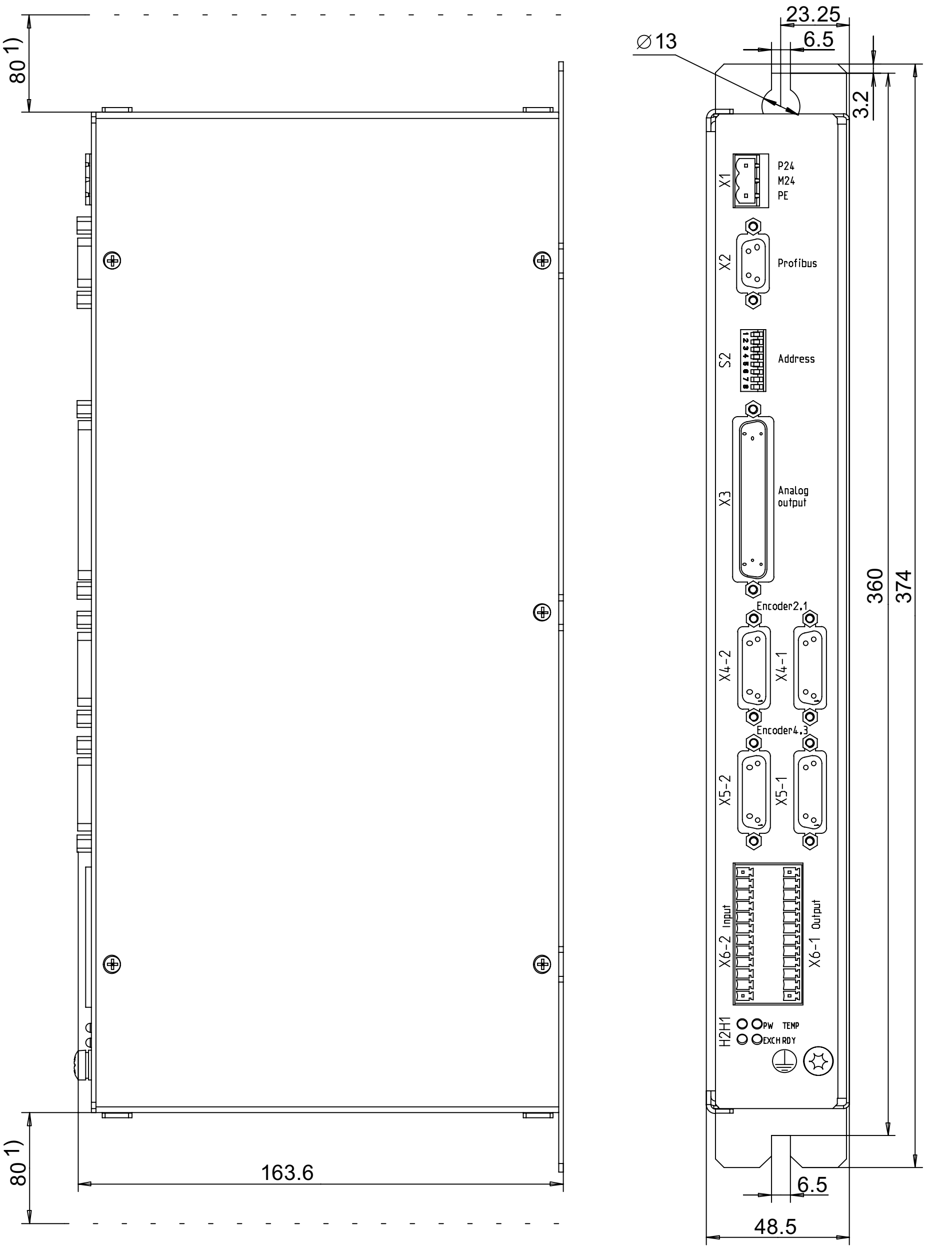


Figure 18-10 Dimension drawing of the ADI4

18.7 Dimension drawing of the DMC20

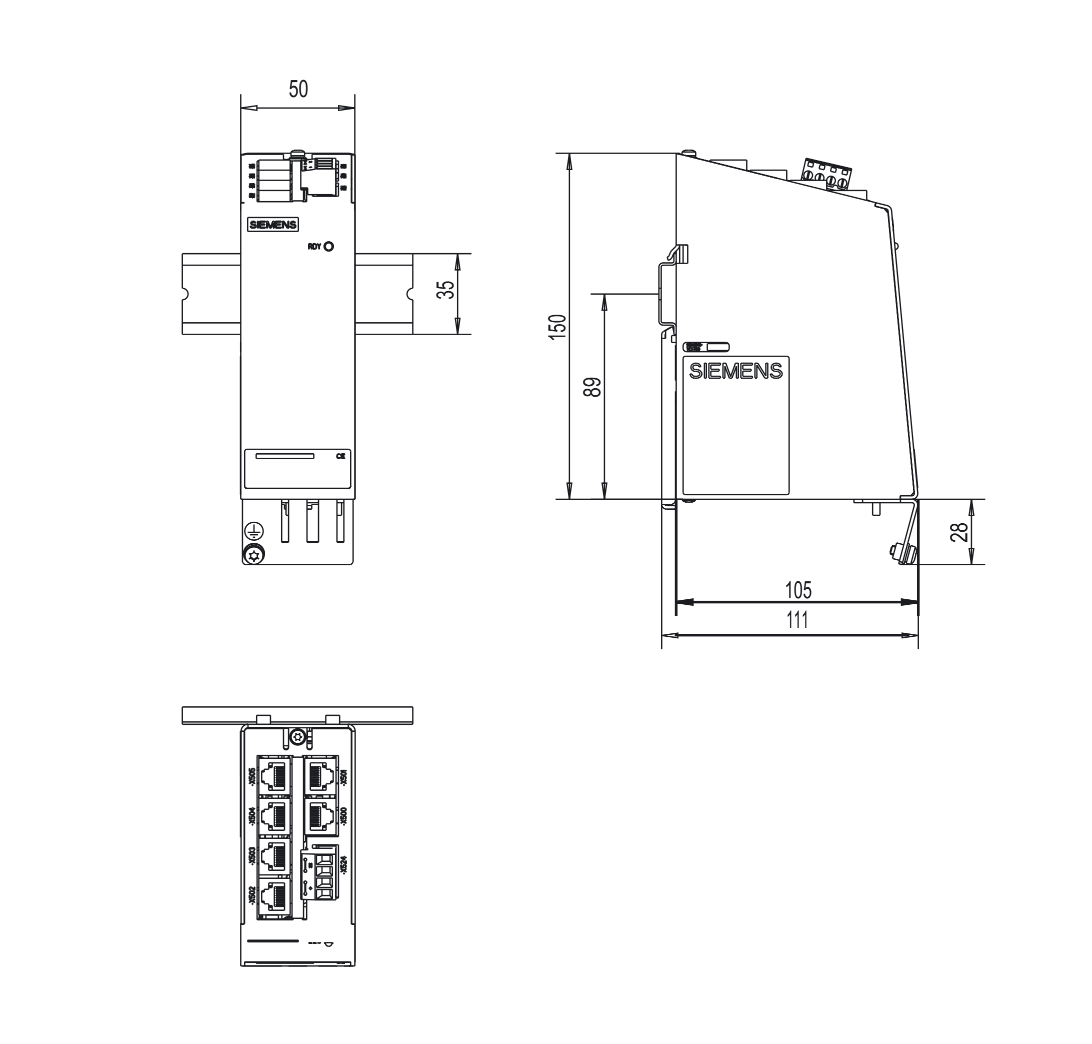


Figure 18-11 Dimension drawing of the DMC20

18.8 Dimension drawing of the SMC10

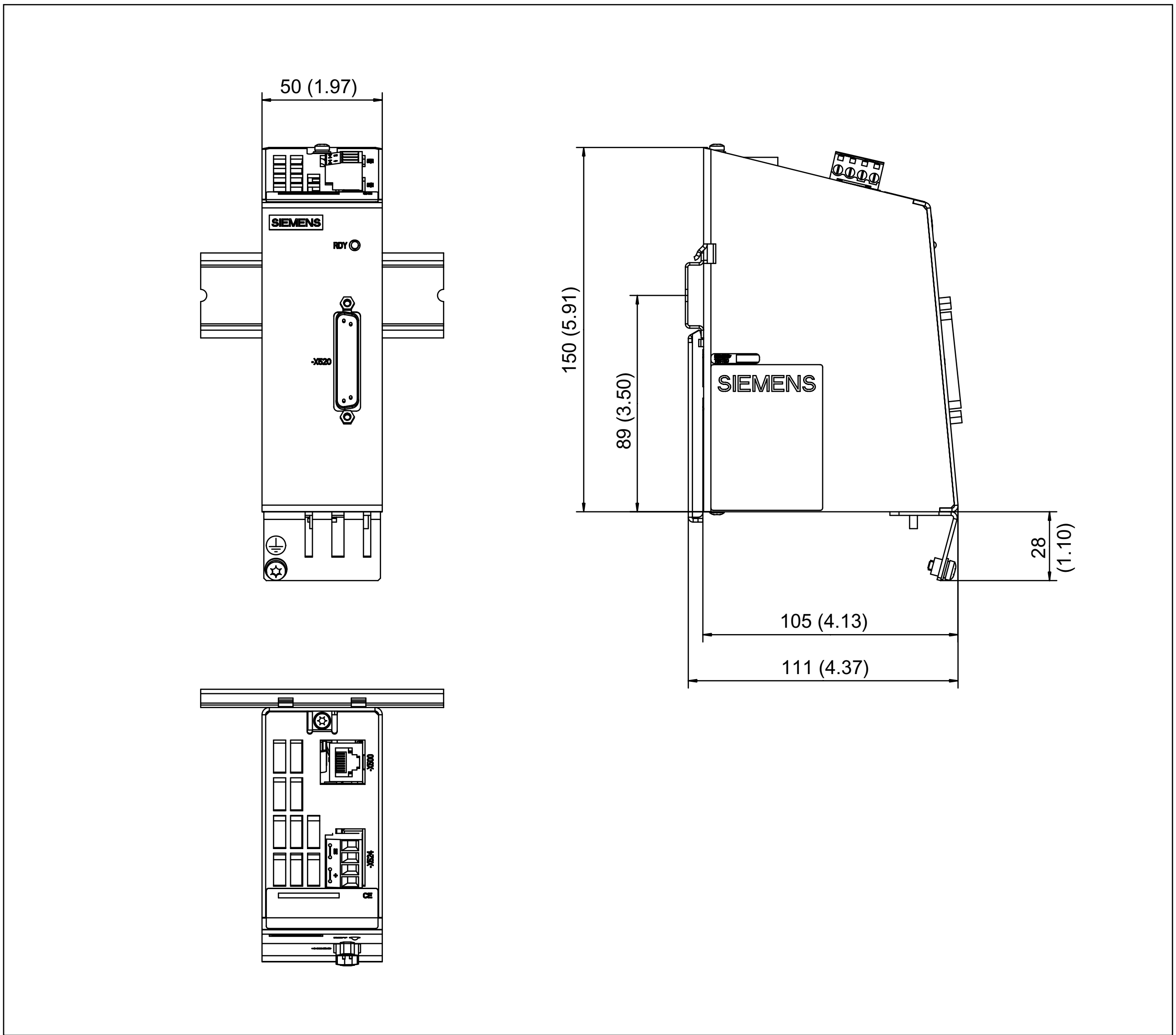


Figure 18-12 Dimension drawing of the Sensor Module Cabinet SMC10, all dimensions in mm and (inches)

18.9 Dimension drawing of the SMC20

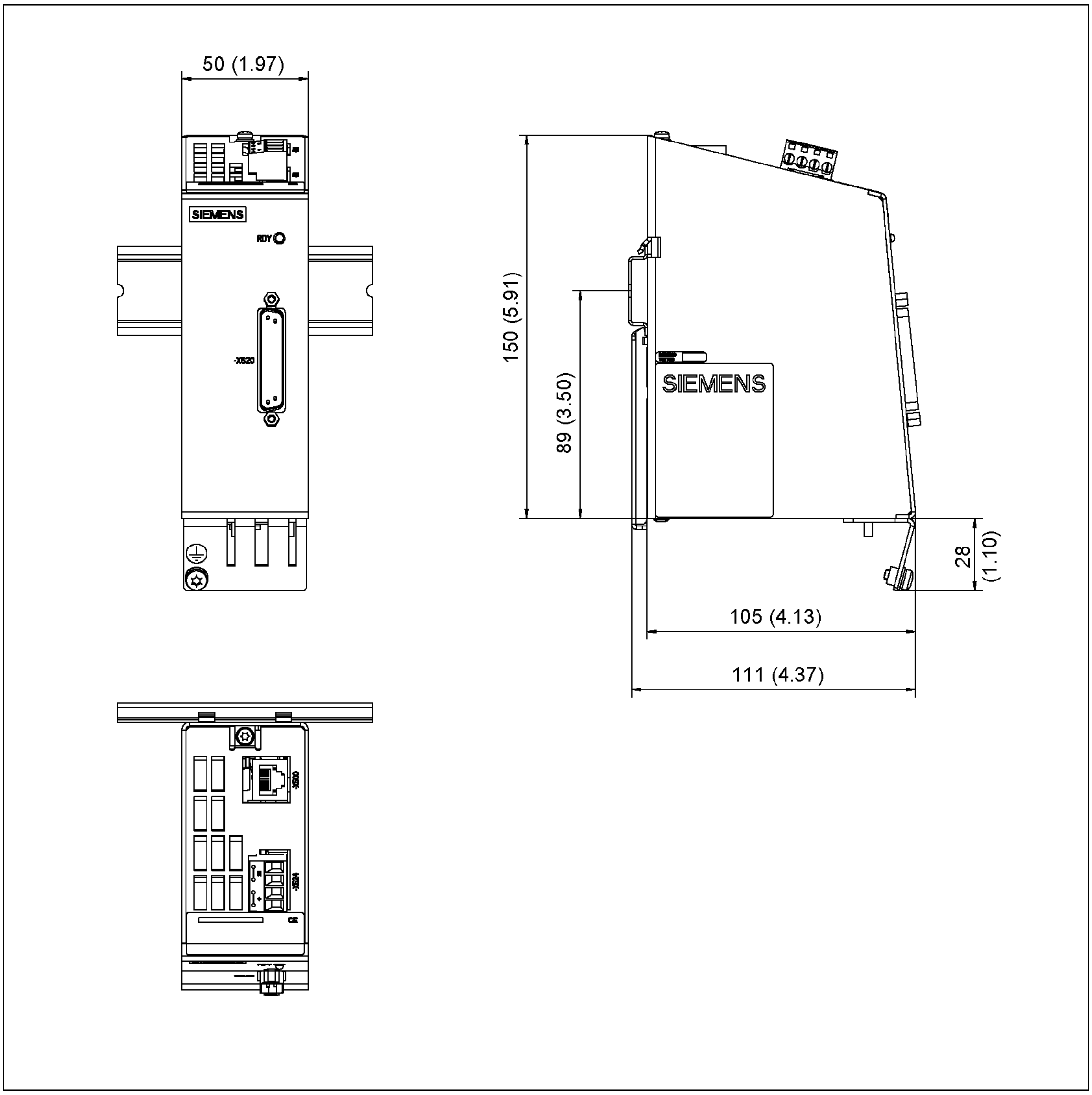


Figure 18-13 Dimension drawing of the SMC20

18.10 Dimension drawing of the SMC30

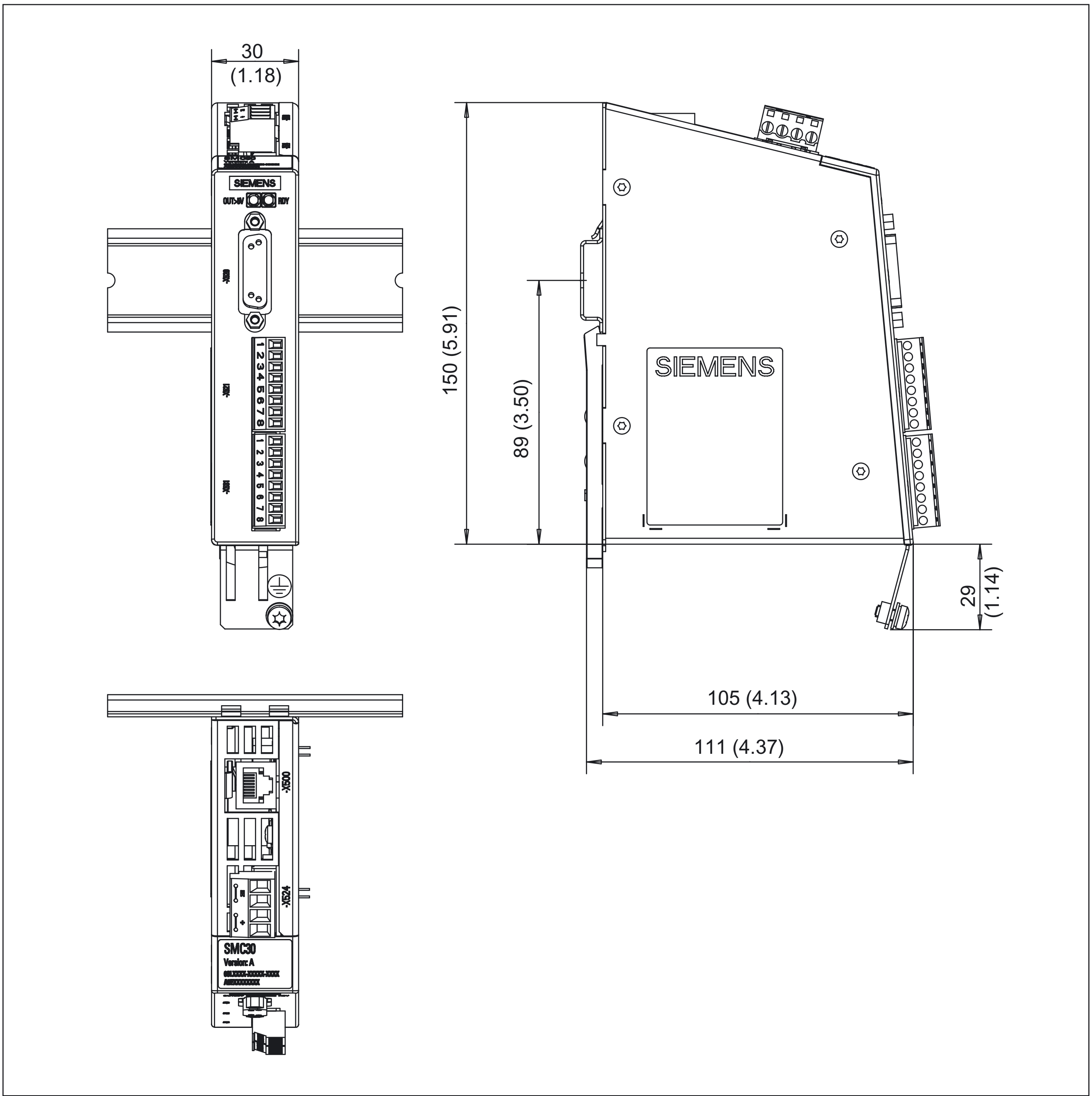


Figure 18-14 Dimensional drawing SMC30: 30 mm wide

As from order number 6SL3055-0AA00-5CA2

ESD guidelines

A

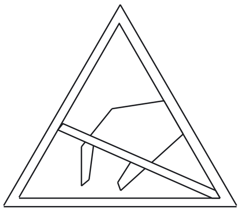
A.1 What does ESD mean?

Definition

All electronic modules are equipped with highly integrated modules or components. Based on their design, these electronic components are highly sensitive to overvoltage and thus to discharge of static electricity.

These **E**lectrostatic **S**ensitive **D**evices/**M**odules are commonly abbreviated **ESD**. The common international designation **ESD** stands for **E**lectrostatic **S**ensitive **D**evice.

Electrostatic sensitive modules are identified by the following symbol:



⚠ CAUTION
Electrostatic sensitive devices may be destroyed by voltages that are undetectable to a human. Voltages of this kind occur as soon as a component or an assembly is touched by a person who is not grounded against static electricity. The damage to a module as a result of overvoltage cannot usually be detected immediately. **I**t may only become apparent after a long period of operation.

A.2 Electrostatic Discharge to Persons

Charge

Any person with a non-conductive connection to the electrical potential of his or her surroundings may be exposed to electrostatic charge.

The figure below shows the maximum electrostatic voltage which can build up on a person coming into contact with the materials indicated. These values correspond with specifications to IEC 801–2.

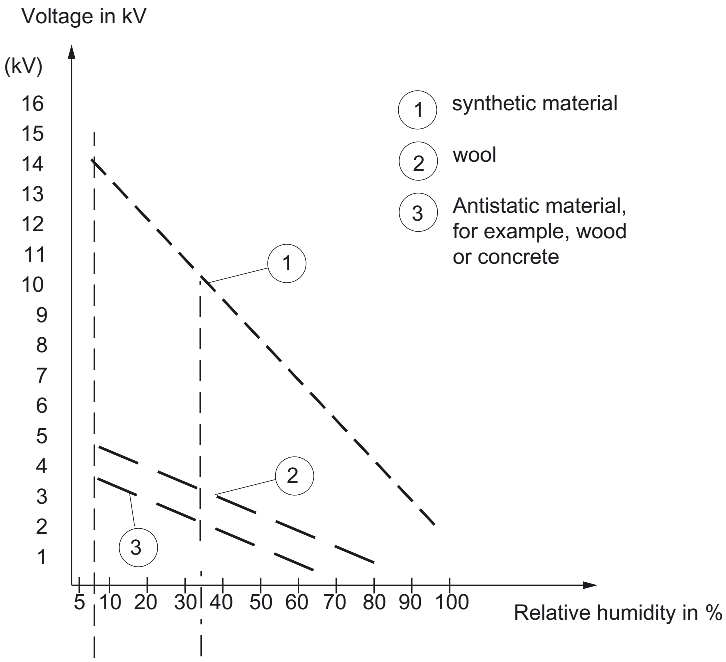


Figure A-1 Electrostatic voltages which can build up on a person

A.3 Basic protective measures against discharge of static electricity

Make sure the grounding is good

When working with electrostatically sensitive devices, make sure that the person, the workstation and the packaging are properly grounded. This helps you avoid static charge.

Avoid direct contact

You should only touch ESD components if this is unavoidable (for example, during maintenance work). When you touch modules, make sure that you do not touch either the pins on the modules or the printed conductors. This prevents any discharge of static electricity to sensitive component and thus avoids damage.

Discharge your body before beginning work on a module. To do so, touch a grounded metallic object. Use only grounded measuring and test equipment.

List of abbreviations

B

B.1 Abbreviations 802D sl

Abbreviation	German	English
AC	Wechselstrom	Alternating Current
ADI	Analog Drive Interface	Analog Drive Interface
ALM	Active Line Module	Active Line Module
AT	AT-Kommadosatz	attention
BERO	Firmenname für einen Näherungsschalter	Tradename for a type of proximity switch
BICO	Binektor-Konnektor-Technologie	Binector Connector Technology
CBC	Communication Board CAN	Communication Board CAN
CBE	Communication Board Ethernet	Communication Board Ethernet
CPU	Zentrale Recheneinheit	Central Processing Unit
CNC	Computerunterstützte numerische Steuerung	Computer Numerical Control
CSM	Control Supply Module	Control Supply Module
CU	Control Unit	Control Unit
DC	Gleichstrom	Direct Current
DMC	DRIVE-CLiQ Hub Module Cabinet	DRIVE-CLiQ Hub Module Cabinet
DO	Antriebsobjekt	Drive Object
DP	Dezentrale Peripherie	Decentralized Peripherals
DRIVE-CLiQ	Drive Component Link with IQ	Drive Component Link with IQ
EDS	Geberdatensatz	Encoder Data Set
EMC	Elektromagnetische Verträglichkeit	Electromagnetic Compatibility (EMC)
EN	Europäische Norm	European Standard
EP	Impulsfreigabe	Enable Pulses
ELCB	Fehlerstrom-Schutzschalter	Earth Leakage Circuit Breaker (ELCB)
HMI	Mensch-Maschine-Schnittstelle	Human Machine Interface
HTL	Logik mit hoher Störschwelle	High-Threshold Logic
IEC	Internationale Norm in der Elektrotechnik	International Electrotechnical Commission
IT	Drehstromversorgungsnetz ungeerdet	Insulated three-phase supply network
LED	Leuchtdiode	Light Emitting Diode
LM	Line Module	Line Module
MCP	Maschinensteuertafel	Machine Control Panel
MCPA	Analoge Maschinensteurtafel	Machine Control Panel Analog
NC	Numerische Steuerung	Numerical Control
NCK	Numerik-Kern mit Satzaufbereitung, Verfahrbereich usw.	Numerical Control Kernel
NCU	Numerical Control Unit	Numerical Control Unit

Abbreviation	German	English
NX	Numerical Extension	Numerical Extension
OP	Bedientafelfront	Operator Panel
PCU	In die Bedientafel integrierte CNC für Bedienoberfläche, Systemsoftware und Soft-PLC	Panel Control Unit
PE	Schutzerde	Protective Earth
PELV	Schutzkleinspannung	Protective Extra Low Voltage
PLC	Speicherprogrammierbare Steuerung (SPS)	Programmable Logic Controller
PP	Peripherie-Modul für PROFIBUS DP	Peripheral module
RCS	Remote Control System	Remote Control System
SBC	Safe Brake Control	Safe Brake Control
SDB	Systemdatenbaustein	system data block
SH	Sicherer Halt	Safe standstill
SIL	Sicherheitsintegritätsgrad	Safety Integrity Level
LEC	Spindelsteigungsfehler-Kompensation	leadscrew error compensation
SSI	Synchron Serielle Schnittstelle	Synchronous Serial Interface
sl	solution line	solution line
SLM	Smart Line Module	Smart Line Module
SMC	Sensor Module Cabinet	Sensor Module Cabinet
SME	Sensor Module External	Sensor Module External
SMI	Sensor Module Integrated	Sensor Module Integrated
SPL	Sichere Programmierbare Logik	Safe Programmable Logic
STW	Steuerwort	Control word
GWPS	Schleifscheiben-Umfangsgeschwindigkeit	
TCU	Thin Client Unit	Thin Client Unit
TM	Terminal Module	Terminal Module
TN	Drehstromversorgungsnetz geerdet	Grounded three-phase supply network
TT	Drehstromversorgungsnetz geerdet	Grounded three-phase supply network
TTL	Transistor-Transistor-Logik	Transistor-Transistor-Logic
TP	Twisted pair	Twisted pair
VPM	Voltage Protection Module	Voltage Protection Module
VS	Spannungsversorgung	Voltage Supply
VSM	Voltage Sensing Module	Voltage Sensing Module
ZSW	Zustandswort	Status word

Appendix C

C.1 User data grinding cycles

The user data is internally processed in the grinding cycles. They are stored in the program manager of the control system (in the directory \DEF) as a definition file and remain stored even when the control is switched off and on.

Description of the user data

The parameters included in the definition files are described as follows:

Name	Type	Default Value	Description
_GC_LERF	REAL		Detected longitudinal position when setting up
_GC_LVER	REAL		Offset during longitudinal position sensing
_GC_LNPVZ	REAL		Initial Z zero shift during calibration
_GC_LXPOS	REAL		X position while longiitudinal position is sensed
_GC_PARR[20]	REAL		REAL type parameters for inter cycle as well as cycle HMI communication
_GC_PAR[0]	INT	0/1	Selection of the type of plunging feedrate in mm/min / specific cutting volumes
_GC_PAR[1]	INT	0/1	Selection of the longitudinal grinding feedrate in mm/min or mm/rev
_GC_PARI[20]	INT		INTEGER type parameters for inter cycle as well as cycle HMI communication
_GC_SYNC	INT	0	HMI synchronisation parameters
_GC_SYNC INIRE	INT	0	Delete synchronisation parameters on reset
_GC_WPC	INT	0	Workpiece counter for dressing interval
_GC_BAXIS	STRING[10]		Name of the swivel axis
_GC_DNUM	INT	7	D number for the 1st data block of dressing data in the tool compensation
_GC_KNVX	INT	0	There it is defined how the detected offset will be taken into account in X: 0 ... Through work offset (NV) 1 ... as wheel diameter offset
_GC_KORR	INT	0	Selection of measurement control compensation computation 0 ... Compensation of the setpoint-actual value difference in the wear of the wheel / dresser 1 ... Compensation of the setpoint-actual value difference in WO in X 2 ... No compensation of the setpoint-actual value difference
_GC_MF[20]	INT		M command number

Name	Type	Default Value	Description
_GC_MF[0] _GC_MF[1] _GC_MF[2] _GC_MF[3] _GC_MF[4] _GC_MF[5] _GC_MF[6] _GC_MF[7] _GC_MF[8] _GC_MF[9] _GC_MF[10] _GC_MF[11] _GC_MF[12] . _GC_MF[13] _GC_MF[14] _GC_MF[15] _GC_MF[16] _GC_MF[17]	INT	3 21 22 33 34 41 42 65 66 80 81 4 7 9	Grinding spindle direction of rotation (M3) Swing in measurement control (M21) Swing out measurement control (M22) Structure-borne noise ON (M33) Structure-borne noise OFF (M34) Advance dresser (M41) Retract dresser (M42) Swing out caliper (M65) Swing in caliper (M66) Enable handwheel (M80) Disable handwheel (M81) Workpiece spindle direction of rotation (M4) Coolant ON (M7) Coolant OFF (M9) Swing in measurement control, program control (M23) Swing out measurement control, program control (M24) Disable stroke reversal if no longitudinal stroke (M27) Enable stroke reversal if longitudinal stroke (M28)
			Number of inputs IN:
_GC_IN_KS	INT	16	Acoustic emission sensor
_GC_IN_MZ0	INT	9	Retract measurement control
_GC_IN_MZ1	INT	10	Time measurement control
_GC_IN_MZ2	INT	11	Switch-over fine finishing measurement control
_GC_IN_MZ3	INT	12	Switch-over finishing measurement control
_GC_IN_MZ4	INT	13	Reserved for inputs/outputs
_GC_IN_ABR	INT	14	Intermediate dressing upon key
_GC_IN_HAND	INT	15	Handwheel key
_GC_IN_BREAK	INT	13	Program interrupt key
_GC_IN_HUB	INT	12	Stroke reversal key
_GC_IN_FEEDSTOP	INT	11	Infeed stop key
_GC_WEARTYP	INT	0	Selection of wear compensation, comparison or nominal dimensions
_GC_SSTAT	INT		Selection ... with/without grinding spindle monitoring
_GC_FEIN[2]	REAL		Global fine compensation
_GC_FEIN[0] _GC_FEIN[1]	REAL		Incremental X fine compensation Incremental Z fine compensation
_GC_SFEIN[10,2]	REAL		Fine compensation seat-specific 1st index … seat number 2nd index ... axis
_GC_RLZTYP	INT	0	Do not approach the return position of the Z-axis in -1-, MCS=0 WCS=1
_GC_RLXTYP	INT	0	Type of return position in
_GC_RLX	REAL		X return position; dresser or workpiece can be collision-free approached using a machine specific return position
_GC_RLZ	REAL		Z return position; dresser or workpiece can be approached without collision using a machine-specific return position.
_GC_BT	REAL		Measurement control tolerance in which a measurement control signal is expected

Name	Type	Default Value	Description
_GC_FWEG	REAL		Free wheel travel path (measurement control)
_GC_SEARCHS			Tag for seat regrinding is evaluated by the cycles so that the individual seat can be identified via a block search.
_GC_SEARCH			Tag for seat regrinding is evaluated by the cycles so that the individual seat can be identified via a block search.
_GC_SEARCHSET			Tag for seat regrinding is evaluated by the cycles so that the axes can be recalibrated.
_GC_SEACRHVALUE[0..2]			Regrinding calibration values
_GC_SUGFEED			Independent of basic system 0 = GWPS in m/s 1 = GWPS in feed/min
_GC_MF[18]			Enable program level abort of CYCLE448
_GC_MF[19]			Blocking and resetting of last program level abort

Note

The values stored as the default must be checked by the machine manufacturer and adapted to the realities of the machine.

C.2 Auxiliary macros for grinding cycles

Definition of ancillary macros

Ancillary macro	Cycle variable from tools management
DEFINE _T_DN	$TC_DP3
DEFINE _T_DV	$TC_DP12
DEFINE _T_DB	$TC_DP21
DEFINE _T_LN	$TC_DP4
DEFINE _T_LV	$TC_DP13
DEFINE _T_LB	$TC_DP22
DEFINE _T_HN	$TC_DP5
DEFINE _T_HV	$TC_DP14
DEFINE _T_HB	$TC_DP23
DEFINE _D_IAB	$TC_DP16[$P_TOOLNO,1]
DEFINE _Z_IAB_L	$TC_DP7[$P_TOOLNO,1]
DEFINE _Z_IAB_R	$TC_DP7[$P_TOOLNO,2]
DEFINE _F_IZ	$TC_DP20[$P_TOOLNO,1]

Ancillary macro	Cycle variable from tools management
DEFINE _F_ID_L	$TC_DP10[$P_TOOLNO,1]
DEFINE _F_IB_L	$TC_DP11[$P_TOOLNO,1]
DEFINE _F_ID_R	$TC_DP10[$P_TOOLNO,2]
DEFINE _F_IB_R	$TC_DP11[$P_TOOLNO,2]
DEFINE _ISUGV	$TC_TPC6[$P_TOOLNO]
DEFINE _SCHEIBENTYP	$TC_TPC1[$P_TOOLNO]
DEFINE _ABRICHTART	$TC_DP19[$P_TOOLNO,1]
DEFINE _BREITE	$TC_TPG5[$P_TOOLNO]
DEFINE _BALLENHOEHE	$TC_TPC2[$P_TOOLNO]
DEFINE _ZYLKORRXE	$TC_TPC4[$P_TOOLNO]
DEFINE _UEBRLZL	ABS($TC_DPC1[$P_TOOLNO,1])
DEFINE _UEBRLZR	ABS($TC_DPC1[$P_TOOLNO,2])
DEFINE _RADIUSL	ABS($TC_DPC2[$P_TOOLNO,1])
DEFINE _RADIUSR	ABS($TC_DPC2[$P_TOOLNO,2])
DEFINE _FASEXL	ABS($TC_DPC3[$P_TOOLNO,1])
DEFINE _FASEXR	ABS($TC_DPC3[$P_TOOLNO,2])
DEFINE _FASEZL	ABS($TC_DPC4[$P_TOOLNO,1])
DEFINE _FASEZR	ABS($TC_DPC4[$P_TOOLNO,2])
DEFINE _SCHULTERL	ABS($TC_DPC5[$P_TOOLNO,1])
DEFINE _SCHULTERR	ABS($TC_DPC5[$P_TOOLNO,2])
DEFINE _HINTERZWL	ABS($TC_DPC6[$P_TOOLNO,1])
DEFINE _HINTERZWR	ABS($TC_DPC6[$P_TOOLNO,2])
DEFINE _HINTERZHL	ABS($TC_DPC7[$P_TOOLNO,1])
DEFINE _HINTERZHR	ABS($TC_DPC7[$P_TOOLNO,2])
DEFINE _UEBRLXL	ABS($TC_DPC8[$P_TOOLNO,1])
DEFINE _UEBRLXR	ABS($TC_DPC8[$P_TOOLNO,2])
DEFINE _SBREITEL	ABS($TC_DPC9[$P_TOOLNO,1])
DEFINE _SBREITER	ABS($TC_DPC9[$P_TOOLNO,2])
DEFINE _ISUG	ABS($TC_TPC5[$P_TOOLNO])
DEFINE _XABHEBBTR	ABS($TC_TPC3[$P_TOOLNO])
DEFINE _IXWP	$TC_TPC9[$P_TOOLNO]
DEFINE _IZWP	$TC_TPC10[$P_TOOLNO]
DEFINE _KONTURPRG	$TC_DPC10[$P_TOOLNO,1]
DEFINE _ABR_3	0
DEFINE _ZIEHEND	1
DEFINE _STOSSEND	2
DEFINE _ZIEHEND_ABR_1	11
DEFINE _STOSSEND_ABR_1	12
DEFINE _ZIEHEND_ABR_2	21
DEFINE _STOSSEND_ABR_2	22

Ancillary macro	Cycle variable from tools management
DEFINE _DABRICHTER	$TC_DP7
DEFINE _PROFTABR	$TC_DPC7
DEFINE _ABRICHTERTYP	$TC_DPC6
DEFINE _ABR1_X_V	ABS($TC_DP8[$P_TOOLNO,1])
DEFINE _ABR1_Z_V	ABS($TC_DP9[$P_TOOLNO,1])
DEFINE _ABR2_X_V	ABS($TC_DP8[$P_TOOLNO,2])
DEFINE _ABR2_Z_V	ABS($TC_DP9[$P_TOOLNO,2])
DEFINE _ABR3_X_V	ABS($TC_DP17[$P_TOOLNO,1])
DEFINE _ABR3_Z_V	ABS($TC_DP18[$P_TOOLNO,1])

C.3 Recycling and disposal

The product is to be disposed of in accordance with national regulations.

The products described in these operating instructions are extensively recyclable on account of the low-toxic composition of the materials used. To recycle and dispose of your old equipment in an environmentally friendly way, please contact an appropriate disposal company.

C.4 Overview

SINUMERIK 802D sl documentation overview

General documentation / catalogs

- SINUMERIK — Advertising brochure
- SINUMERIK 802D sl — Catalog NC 61
- SINAMICS S120 — Catalog D21.1 Drive converter chassis units

User documentation

- SINUMERIK 802D sl — Programming and Operating Manual
  - – Turning
  - – Milling
  - – Cylindrical grinding
  - – Surface grinding
  - – Nibbling
  - – Manual machine plus
- SINUMERIK 802D sl — Diagnostics Manual
- SINUMERIK 802D sl 840D sl — Programming Manual ISO turning / milling

Manufacturer / service documentation

- SINUMERIK 802D sl — Operating instructions
- SINUMERIK 802D sl — Function Manual
- SINUMERIK 802D sl 840D sl — Function Manual ISO dialects
- SINUMERIK 802D sl — Parameter Manual
- SINUMERIK — EMV directives

Electronic documentation

- SINUMERIK SINAMICS Motors — DOCONCD DOCONWEB

Index

A

B

C

D