SIEMENS

SINUMERIK

SINUMERIK 808D
PLC Subroutines Manual

Operating Instructions

Valid for:
SINUMERIK 808D Turning (software version: V4.4.2)
SINUMERIK 808D Milling (software version: V4.4.2)

Target group:
OEM engineers

12/2012
6FC5397-2EP10-0BA0

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: 6FC5397-2EP10-0BA0
Ⓟ 12/2012 Technical data subject to change

Preface

SINUMERIK 808D documentation

The SINUMERIK 808D documentation consists of the following components:

- Operating Instructions
  - Mechanical Installation Manual
  - Electrical Installation Manual
  - PLC Subroutines Manual
  - Function Manual
  - Parameter Manual
- Diagnostics Manual
- Commissioning Manual
- Programming and Operating Manual (Turning)
- Programming and Operating Manual (Milling)
- Manual Machine Plus (Turning)
- Online Help for Programming and Operating (Turning)
- Online Help for Programming and Operating (Milling)
- Online Help for Manual Machine Plus (Turning)

My Documentation Manager (MDM)

Under the following link you will find information to individually compile your documentation based on the Siemens content:

www.siemens.com/mdm

Target group

This manual is intended for use by the OEM engineers who have basic CNC and PLC knowledge.

Benefits

This manual enables the intended target group to get detailed information about the PLC subroutines and how to use them.

Technical support

Hotline:	+86 400-810-4288
Service and Support	• China: www.siemens.com.cn/808D • Worldwide: http://support.automation.siemens.com

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.

Licensing provisions

The SINUMERIK 808D software 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 penalties or claims for compensation.

In the SINUMERIK 808D, open source software is used. The licensing provisions for this software are included on the Toolbox DVD and are to be observed accordingly.

Table of contents

Overview 1

The PLC subroutines consist of three project files:

- default_turning.ptp (sample application for turning machines)
- default_milling.ptp (sample application for milling machines)
- default_ManMachPlus_T.ptp (sample application for Manual Machine Plus)

With the sample applications, you can get a good understanding about how to create or call a PLC subroutine. You can realize most machine functions by re-organizing the PLC subroutines or modifying some must networks.

⚠ WARNING

Make sure that you perform a complete machine test to all subroutines used in your main program, in order to verify that all the subroutines called by the main program function as you desire. Failure to observe may cause personal injury or property damage.

System resource distribution

The system resources can be distributed into three parts:

- PLC system
  - Inputs
    I0.0 to I2.7 (24 inputs of the SINUMERIK 808D)
    I3.0 to I8.7 (distributed 48 inputs)
  - Outputs
    Q0.0 to Q1.7 (16 inputs of the SINUMERIK 808D)
    Q2.0 to Q5.7 (distributed 32 outputs)
  - Memory
    M0.0 to M255.7 (256 bytes)
  - Non-volatile memory
    DB1400.DBX0.0 to DB1400.DBX127.7 (128 bytes)
  - PLC user alarms:
    DB1600.DBX0.0 to DB1600.DBX15.7 (128 user alarms)
  - Timer
    T0 to T15 (100ms timer)
    T16 to T63 (10ms timer)
  - Counter
    C0 to C63 (64 counters)
- NCK
  - PLC machine data: MD14510, MD14512, MD 14514
  - MD14510 machine data INT: DB4500.DBW0 to DB4500.DBW62 (32 words)
  - MD14512 machine data hex: DB4500.DBB1000 to EDB4500.DBB1031 (32 bytes)
  - MD14514 machine data real: DB4500.DBD2000 to DB4500.DBD2028 (8 Dword)

- PLC Programming Tool
  - Symbol table: SYM1 to SYM32 (32 symbol tables)
  - Subroutine: SBR0 to SBR63 (64 subroutines)

Structure of the symbol tables

The PLC subroutine library has been designed with symbol addressing method, which helps you easily understand the PLC programs. All the addresses in the subroutine library use symbols for programming. All the interface signals are named with symbols and assigned to different symbol tables.

Symbol table	Table name	Descriptions
1	IO_1	Module I/O are defined by the manufacturer
2	IO_2	Distributed I/O are defined by the manufacturer
3, 5, 7, 13		Reserved for the manufacturer
6	MANMACH	JOG function
14	ASUP	ASUP function
15	PLC_sel_PP	PLC selects part programs
16	IS_MCP	Signals from/to the MCP
17	IS_HMI	Signals from/to the HMI
18	IS_AUX	Auxiliary functions from the NCK
19	IS_NCK	Signals from/to the NCK
20	IS_CHA	Signals from/to the channel
21	IS_AX1	Signals to/from axis 1
22	IS_AX2	Signals to/from axis 2
23	IS_AX3	Signals to/from axis 3
24	IS_AX4	Signals to/from axis 4
27	MD_PLC	PLC machine data
28	ALARM	User alarms
29	NV_MEM	Non-volatile memory
30	SPC_MEM	Special memory bit
31	SBR_MEM	Global memory used in the sample applications and subroutines
32	RESVD1	Reserved for the sample applications and subroutines

Structure of the subroutines

Subroutine No.	Name	Description
0 to 19	-	Reserved for the manufacturer
20	AUX_MCP	Auxiliary function
21	AUX_LAMP	Lamp control, called in the subroutine "AUX_MCP".
22	AUX_SAFE_DOOR	Safe door control, called in the subroutine "AUX_MCP" of a milling application.
23	AUX_CHIP	Chip remover control, called in the subroutine "AUX_MCP" of a milling application.

Subroutine No.	Name	Description
31	PLC_ini_USR_INI	Reserved for the initialization by the manufacturer (this subroutine is automatically called by subroutine 32)
32	PLC_INI	PLC initialization
33	EMG_STOP	Emergency Stop
37	MCP_NCK	Signals from the MCP and the HMI are sent to NCK interfaces
38	MCP_Tool_Nr	Display tool numbers via the LED of the MCP
39	HANDWHL	Handwheel selection via HMI
40	AXIS_CTL	Control of feed axis enable and spindle enable
41	MINI_HHU	Handwheel hand held unit
42	SPINDLE	Spindle function
43	MEAS_JOG	Tool measurement in the JOG mode
44	COOLING	Coolant control (Manual Machine key and M code: M07, M08, M09)
45	LUBRICATE	Lubrication control (interval and time)
46	PI_SERVICE	ASUP (Asynchronous Subroutine Program)
47	PLC_Select_PP	PLC selects a subroutine.
48	ServPlan	Service plan
49	Gear_Chg1_Auto	Automatic gear change of the spindle
50	Gear_Chg2_Virtual	Dummy gear change of the spindle
51	Turret1_HED_T	Turret control of the turning machine (turret type: Hall element transistor, 4/6 position)
52	Turret2_BIN_T	Turret control of the turning machine (turret type: position detection with encodings)
53	Turret3_CODE_T	Hydraulic turret control of the turning machine (turret type: position detection with encodings)
54	Turret2_3_ToolDir	Evaluate tool direction and calculate tool position (called by Turret2_BIN_T, Turret3_CODE_T)
55	Tail_stock_T	Tail stock control for the turning machine
56	Lock_unlock_T	Clamp or release control for the turning machine
58	MM_MAIN	Manual machine
59	MM_MCP_808D	Spindle signal processing for the manual machine
60	Disk_MGZ_M	Disk tool magazine for a milling machine
61, 62		Reserved for the subroutine
63	TOGGLE	Six key-operated switches: K1 to K6 Two delay switches: K7, K8

MCP interface description

Input/output	DB number	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Input (MCP -> PPU), DB1000	DB1000.DBB0	M01	Program test	MDA	Single block	AUTO	REF. POINT	JOG	Hand-wheel
	DB1000.DBB1	Key 16	Key 15	Key 14	Key 13	Key 12	Key 11	Key 10	ROV

Input/output	DB number	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	DB1000.DBB2	100 (INC)	10 (INC)	1 (INC)	Key 21	Key 20	Key 19	Key 18	Key 17
	DB1000.DBB3	Key 32	Key 31	Cycle start	Cycle Stop	RESET	Spindle right	Spindle Stop	Spindle left
	DB1000.DBB4		Key 39	Key 38	Key 37	Key 36	RAPID	Key 34	Key 33
	DB1000.DBB5								
	DB1000.DBB6								
	DB1000.DBB7								
	DB1000.DBB8	Feed override value (in Gray code)							
	DB1000.DBB9	Spindle override value (in Gray code)							
	DB1000.DBB10								
Output (PPU -> MCP), DB1100	DB1100.DBB0	LED 8	LED 7	LED 6	LED 5	LED 4	LED 3	LED 2	LED 1
	DB1100.DBB1	LED 16	LED 15	LED 14	LED 13	LED 12	LED 11	LED 10	LED 9
	DB1100.DBB2	LED 24	LED 23	LED 22	LED 21	LED 20	LED 19	LED 18	LED 17
	DB1100.DBB3			LED 30	LED 29	LED 28	LED 27	LED 26	LED 25
	DB1100.DBB4								
	DB1100.DBB5								
	DB1100.DBB6								
	DB1100.DBB7								
	DB1100.DBB8	7 SEG LED 1							
	DB1100.DBB9	7SEG LED 2							
	DB1100.DBB10								
	DB1100.DBB11								
	DB1100.DBB12							DP 2 [2]	DP 1 [1]

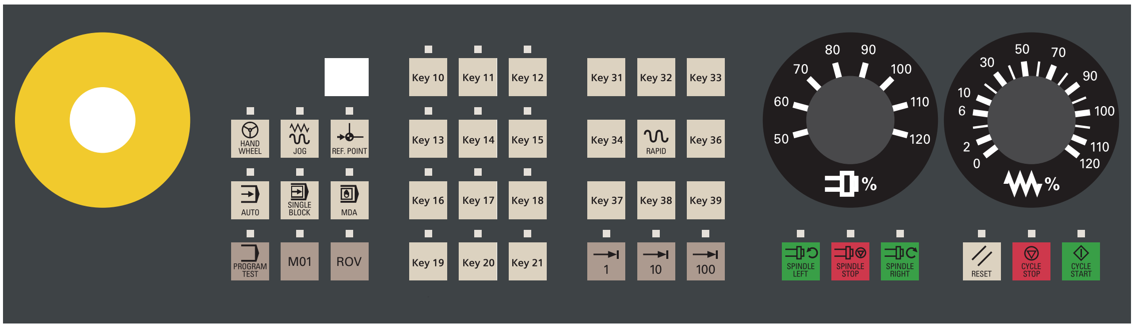
[1] The decimal point of the 7 SEG LED 1.

[2] The decimal point of the 7 SEG LED 2.

Note

Interfaces where nothing has been entered are reserved for the next version.

You can understand the relationship between the PLC interface addresses and the SINUMERIK 808D MCP from the illustration below:



PLC Programming Tool 2

2.1 Installing the 808D Toolbox

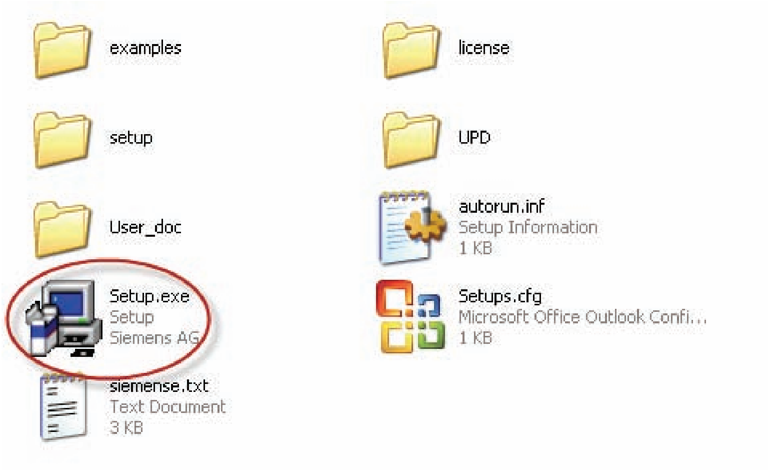
You must install the 808D Toolbox on your PC/PG. The 808D Toolbox contains the following software tools or information:

- Config Data 808D
  - System software update
  - Examples (EasyXLanguage, symbols for MCP customized keys, template for MCP strips, PLC subroutine library)
- PLC Programming Tool

  The tool for creating PLC user programs
- Windows setup software for toolbox
- User documentation
- License information and conditions about Open Source Software (Readme_OSS)
- SinuComPCIN

Installing the 808D toolbox

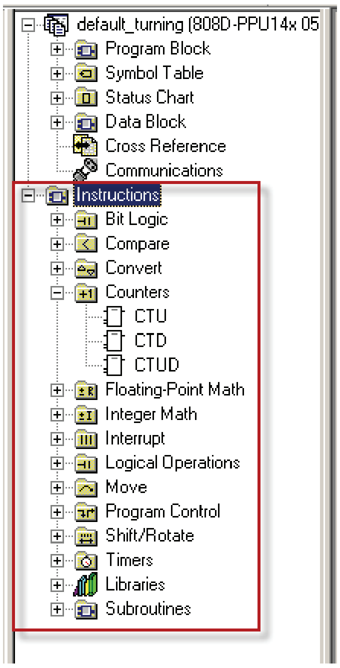
The toolbox installation folder is shown below:



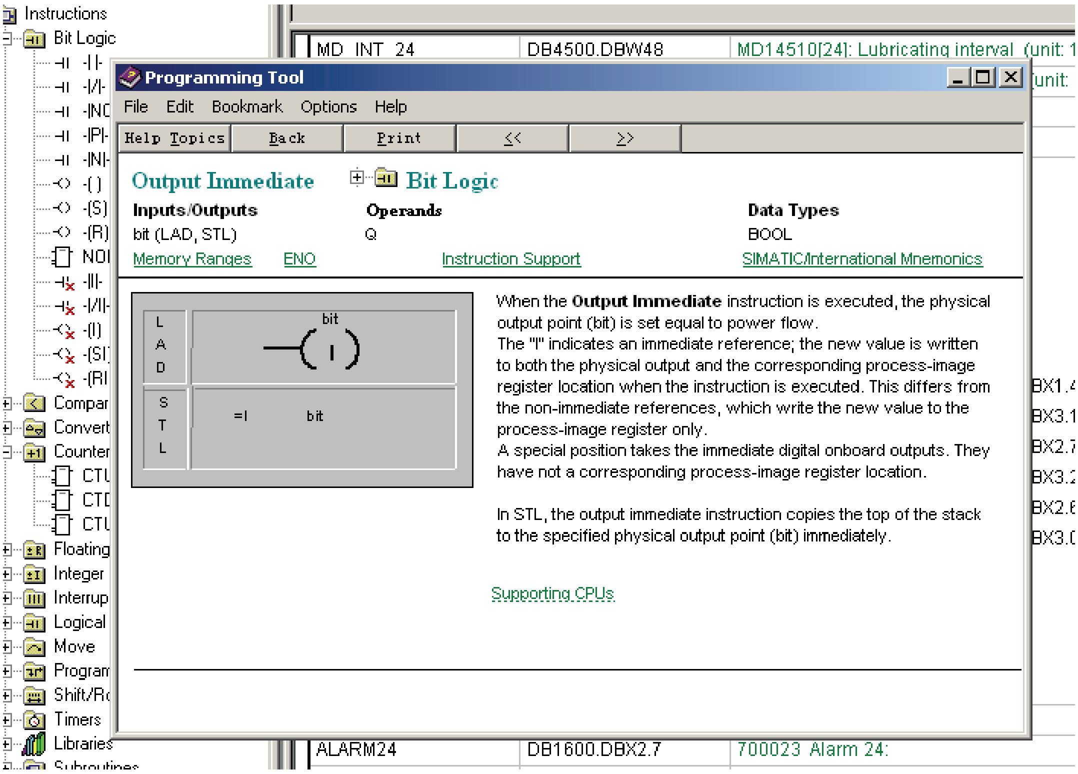
To start the installation, double-click the **Setup.exe** file. In the setup dialog, select one or multiple options that you desire to install.

2.2 PLC instructions

Various instructions are available in the PLC Programming Tool. You can view them in the instruction branch on the instruction tree:



You can right-click over an instruction or directly press the F1 key to view its help information. For example:



Output Immediate

Inputs/Outputs	**Operands**	**Data Types**
bit (LAD, STL)	Q	BOOL

When the **Output Immediate** instruction is executed, the physical output point (bit) is set equal to power flow.
The "I" indicates an immediate reference; the new value is written to both the physical output and the corresponding process-image register location when the instruction is executed. This differs from the non-immediate references, which write the new value to the process-image register only.
A special position takes the immediate digital onboard outputs. They have not a corresponding process-image register location.

In STL, the output immediate instruction copies the top of the stack to the specified physical output point (bit) immediately.

2.3 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 active at POWER ON)

Most data, such as the process image, timers, and counters are non-retentive and are cleared 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 after POWER OFF/ON.

With the aid of the PLC-MD (see user interface), you can pre-assign your program with data or parameterize various parts of the program.

2.4 Program organization

When programming the PLC, you must structure your program into finished program parts (subroutines). The programming language for S7-200 offers you the capability to set up your user program in a structured manner.

There are two types of programs:

- The main program
- The subroutine.

Eight levels of programming are possible.

2.5 Testing and monitoring your program

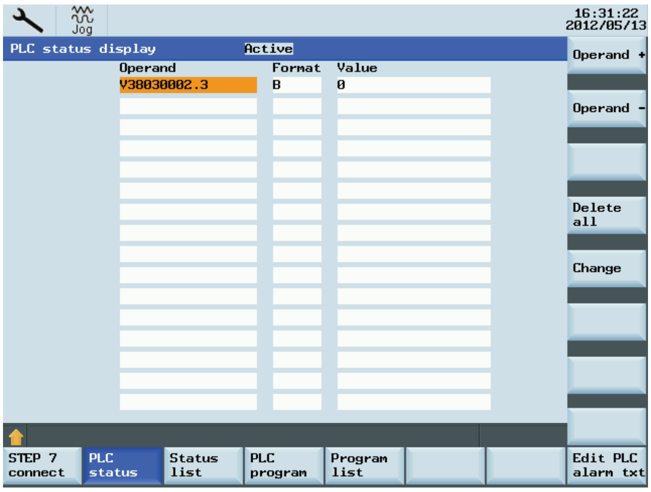
You can check or perform an error analysis of the user program in two methods:

- The HMI of the SINUMERIK 808D
- The PLC Programming Tool

Testing and monitoring a PLC program with the HMI

There are three ways for program testing and monitoring in the "**SYSTEM**" operating area:

- **"PLC" > "PLC status"**



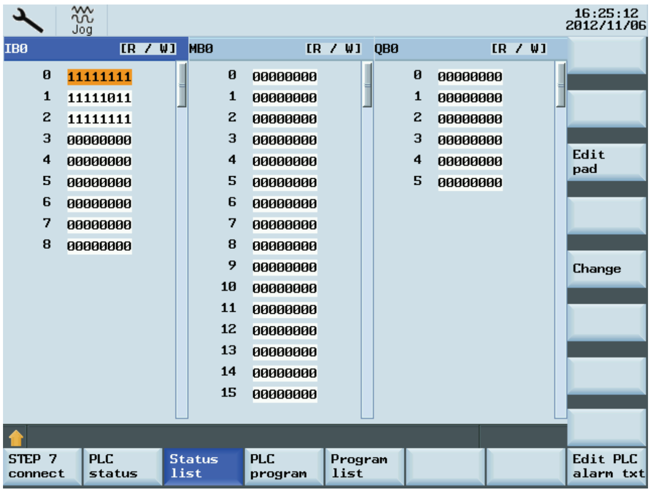
PLC status display Active

Operand	Format	Value
V38030002.3	B	0

You can input an operand to view its status.

Use the softkey "**Operand +**" or "**Operand -**" to increase/decrease the bit of the operand. Use the "**Change**" softkey to change the value of the operand and the "**Delete all**" softkey to delete all the input operands.

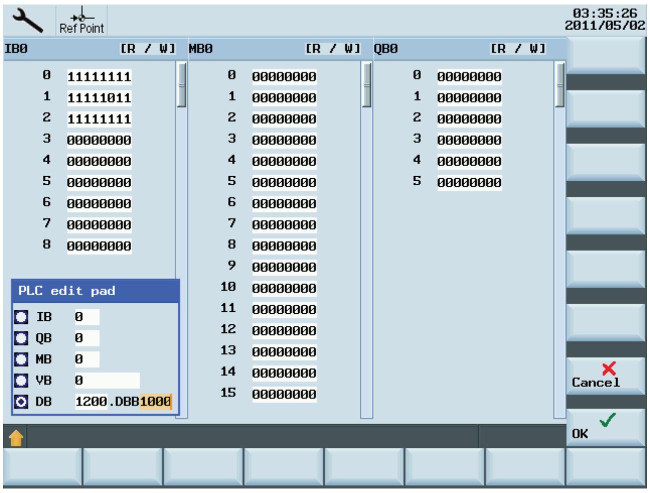
- **"PLC" > "Status list"**



IB0	[R / W]	MB0	[R / W]	QB0	[R / W]
0	11111111	0	00000000	0	00000000
1	11111011	1	00000000	1	00000000
2	11111111	2	00000000	2	00000000
3	00000000	3	00000000	3	00000000
4	00000000	4	00000000	4	00000000
5	00000000	5	00000000	5	00000000
6	00000000	6	00000000		
7	00000000	7	00000000		
8	00000000	8	00000000		
		9	00000000		
		10	00000000		
		11	00000000		
		12	00000000		
		13	00000000		
		14	00000000		
		15	00000000		

You can view the status of a PLC signal.

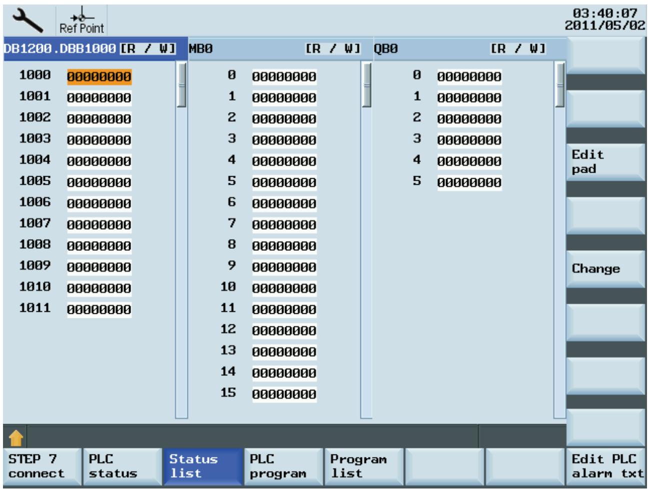
By default, three signal status lists (inputs, flags and outputs) are displayed in three columns. You can change column sequence or assign a new signal status list (variables) with the "**Edit pad**" softkey:



IB0	[R / W]	MB0	[R / W]	QB0	[R / W]
0	11111111	0	00000000	0	00000000
1	11111011	1	00000000	1	00000000
2	11111111	2	00000000	2	00000000
3	00000000	3	00000000	3	00000000
4	00000000	4	00000000	4	00000000
5	00000000	5	00000000	5	00000000
6	00000000	6	00000000		
7	00000000	7	00000000		
8	00000000	8	00000000		
		9	00000000		
		10	00000000		
		11	00000000		
		12	00000000		
		13	00000000		
		14	00000000		
		15	00000000		

PLC edit pad

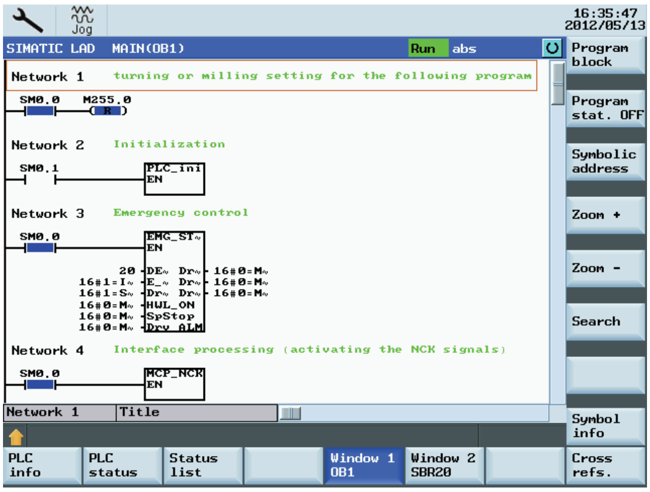
○	IB	0
○	QB	0
○	MB	0
○	VB	0
●	DB	1200.DBB1000



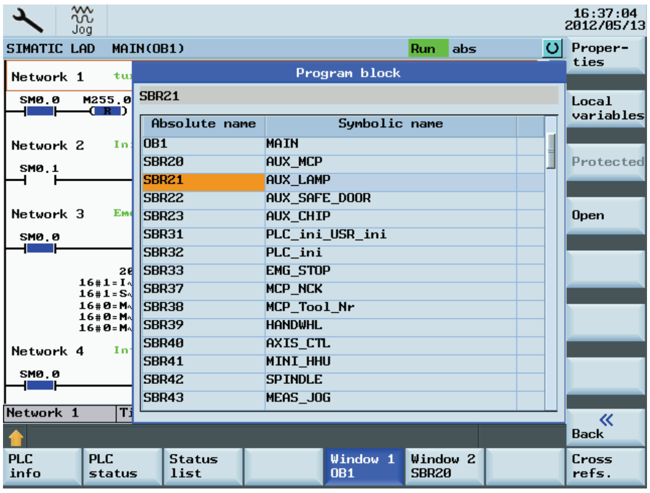
DB1200.DBB1000	[R / W]	MB0	[R / W]	QB0	[R / W]
1000	00000000	0	00000000	0	00000000
1001	00000000	1	00000000	1	00000000
1002	00000000	2	00000000	2	00000000
1003	00000000	3	00000000	3	00000000
1004	00000000	4	00000000	4	00000000
1005	00000000	5	00000000	5	00000000
1006	00000000	6	00000000		
1007	00000000	7	00000000		
1008	00000000	8	00000000		
1009	00000000	9	00000000		
1010	00000000	10	00000000		
1011	00000000	11	00000000		
		12	00000000		
		13	00000000		
		14	00000000		
		15	00000000		

You can also change the status of a PLC signal with the "**Change**" softkey.

- "PLC" > "PLC program"



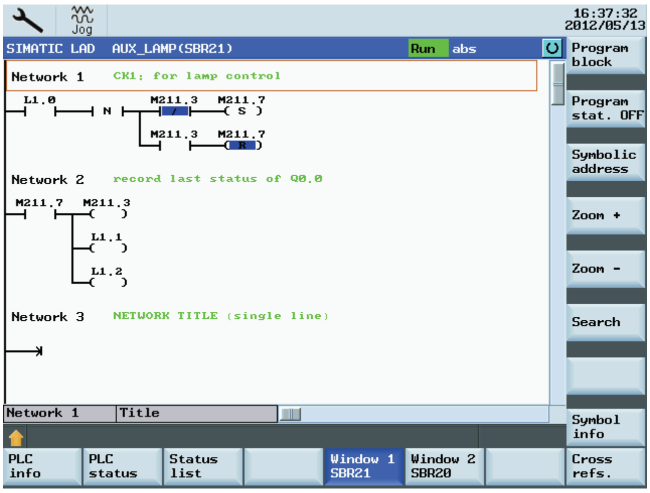
You can view the status of the main program or use the "**Program block**" softkey to view the status of a subroutine:



Program block

SBR21

Absolute name	Symbolic name
OB1	MAIN
SBR20	AUX_MCP
SBR21	AUX_LAMP
SBR22	AUX_SAFE_DOOR
SBR23	AUX_CHIP
SBR31	PLC_ini_USR_ini
SBR32	PLC_ini
SBR33	EMG_STOP
SBR37	MCP_NCK
SBR38	MCP_Tool_Nr
SBR39	HANDWHL
SBR40	AXIS_CTL
SBR41	MINI_HHU
SBR42	SPINDLE
SBR43	MEAS_JOG



Two windows are available for you to view the program.

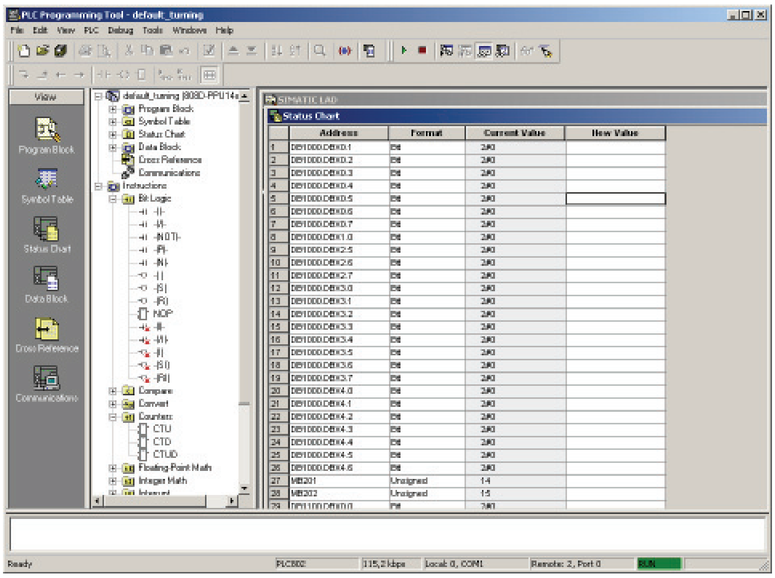
Testing and monitoring a PLC program with the PLC Programming Tool

You can also view the status of your PLC program with the PLC Programming Tool:

1. Establish the communication between the SINUMERIK 808D and the PLC Programming Tool.

   For detailed information about the establishment of the communication between the SINUMERIK 808D and the PLC Programming Tool, refer to section "Establishing a connection with the RS232 interface (Page 17)"

2. Use the menu command "**Debug** > **Chart status**" to view the status of a PLC signal. You can also first click the status chart symbol and then click the symbol to view the status.



	Address	Format	Current Value	New Value
1	DB1000.DBX0.1	Bit	2#0	
2	DB1000.DBX0.2	Bit	2#0	
3	DB1000.DBX0.3	Bit	2#0	
4	DB1000.DBX0.4	Bit	2#0	
5	DB1000.DBX0.5	Bit	2#0	
6	DB1000.DBX0.6	Bit	2#0	
7	DB1000.DBX0.7	Bit	2#0	
8	DB1000.DBX1.0	Bit	2#0	
9	DB1000.DBX2.5	Bit	2#0	
10	DB1000.DBX2.6	Bit	2#0	
11	DB1000.DBX2.7	Bit	2#0	
12	DB1000.DBX3.0	Bit	2#0	
13	DB1000.DBX3.1	Bit	2#0	
14	DB1000.DBX3.2	Bit	2#0	
15	DB1000.DBX3.3	Bit	2#0	
16	DB1000.DBX3.4	Bit	2#0	
17	DB1000.DBX3.5	Bit	2#0	
18	DB1000.DBX3.6	Bit	2#0	
19	DB1000.DBX3.7	Bit	2#0	
20	DB1000.DBX4.0	Bit	2#0	
21	DB1000.DBX4.1	Bit	2#0	
22	DB1000.DBX4.2	Bit	2#0	
23	DB1000.DBX4.3	Bit	2#0	
24	DB1000.DBX4.4	Bit	2#0	
25	DB1000.DBX4.5	Bit	2#0	
26	DB1000.DBX4.6	Bit	2#0	
27	MB201	Unsigned	14	
28	MB202	Unsigned	15	
29	DB1100.DBX0.0	Bit	2#1	

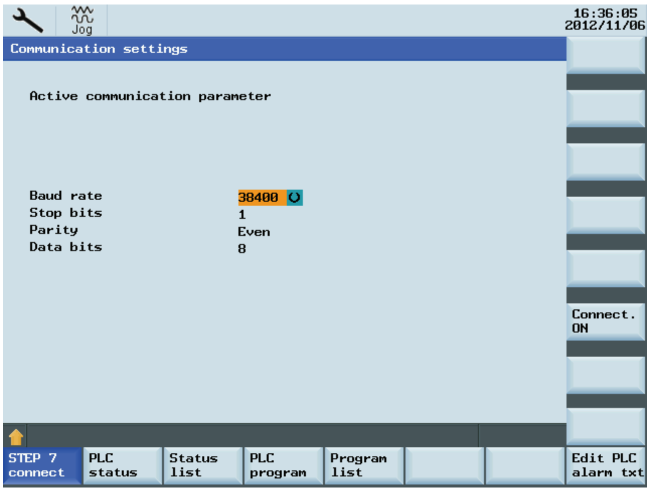
Ready | PLC802 | 115,2 kbps | Local: 0, COM1 | Remote: 2, Port 0 | RUN

2.6 Establishing a connection with the RS232 interface

You can establish a communication between the control system and the PC/PG (PLC Programming Tool) via the RS232 port.

Activating the connection to the control system

You can activate the connection with the HMI: "**SYSTEM**" operating area > "**PLC**" > "**STEP 7 connect**" > "**Connect. ON**":



16:36:05
2012/11/06

Communication settings

Active communication parameter

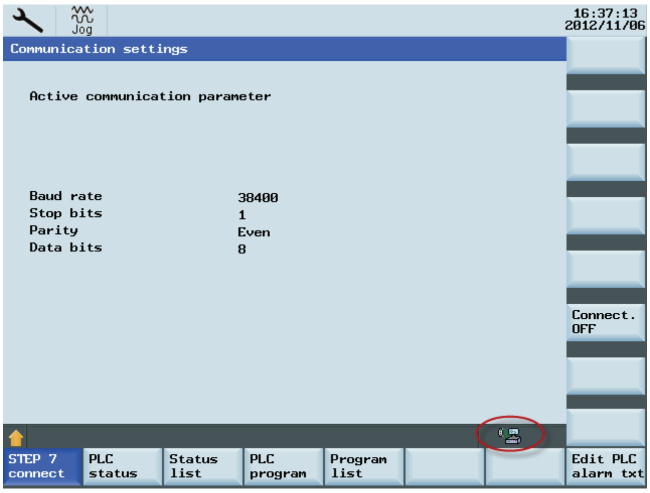
Baud rate	38400
Stop bits	1
Parity	Even
Data bits	8

Use the <**SELECT**> key to select a communication baud rate. The SINUMERIK 808D supports the following baud rates:

- 9.6 kbps
- 19.2 kbps
- 38.4 kbps
- 57.6 kbps

- 115.2 kbps.

The active or inactive state is retained even after a power-on (except when starting with the default data). An active connection is displayed by a symbol in the status bar:



Communication settings

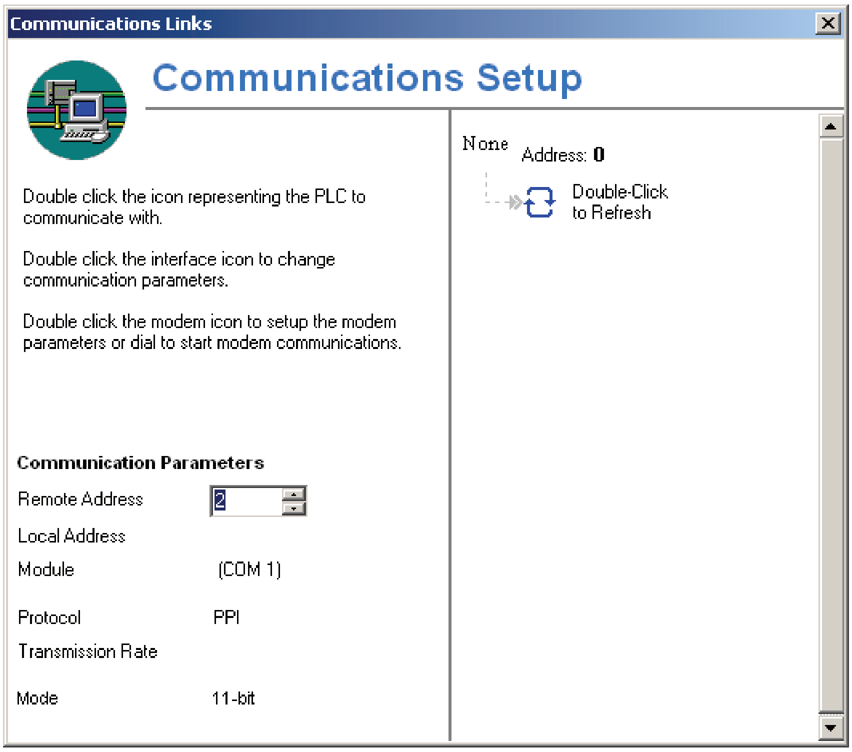
Active communication parameter

Baud rate 38400
Stop bits 1
Parity Even
Data bits 8

Communication settings in the PLC Programming Tool

To setup the PPI parameters in the PLC Programming Tool, proceed as follows:

1. Display the "**Communication Links**" dialog with menu command "**View** > **Communications**", or click communication button in the **View** column, or click the communication symbol Communications in the operations tree:



Communications Links

Communications Setup

Double click the icon representing the PLC to communicate with.

Double click the interface icon to change communication parameters.

Double click the modem icon to setup the modem parameters or dial to start modem communications.

Communication Parameters

Remote Address 2

Local Address

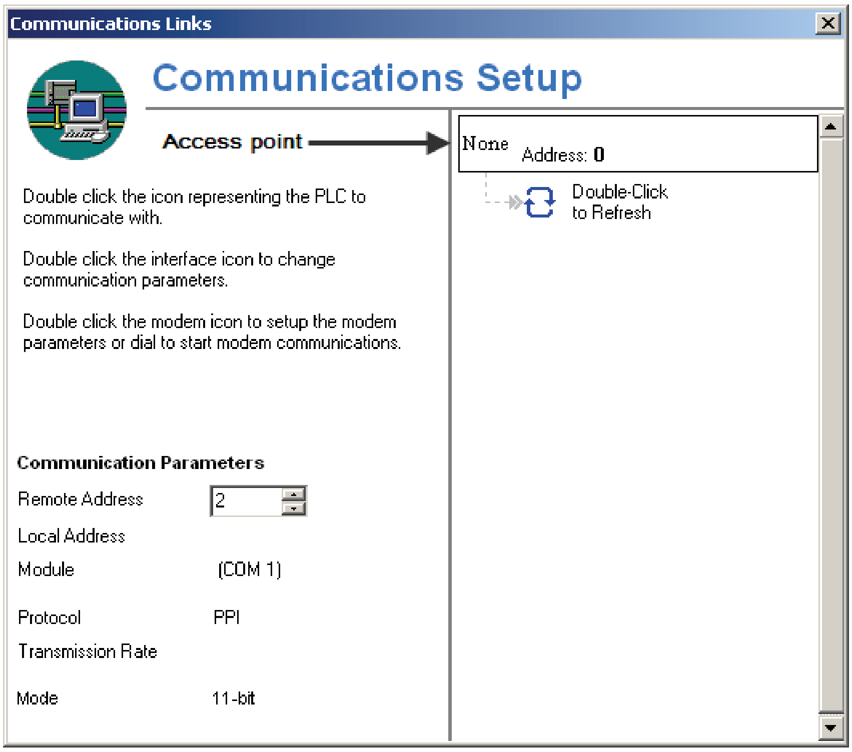
Module (COM 1)

Protocol PPI

Transmission Rate

Mode 11-bit

2. Double click on the "**Access point**" symbol in the "Communication" window on the right side:



Communications Links

Communications Setup

Access point

None Address: 0

Double-Click to Refresh

Double click the icon representing the PLC to communicate with.

Double click the interface icon to change communication parameters.

Double click the modem icon to setup the modem parameters or dial to start modem communications.

Communication Parameters

Remote Address 2

Local Address

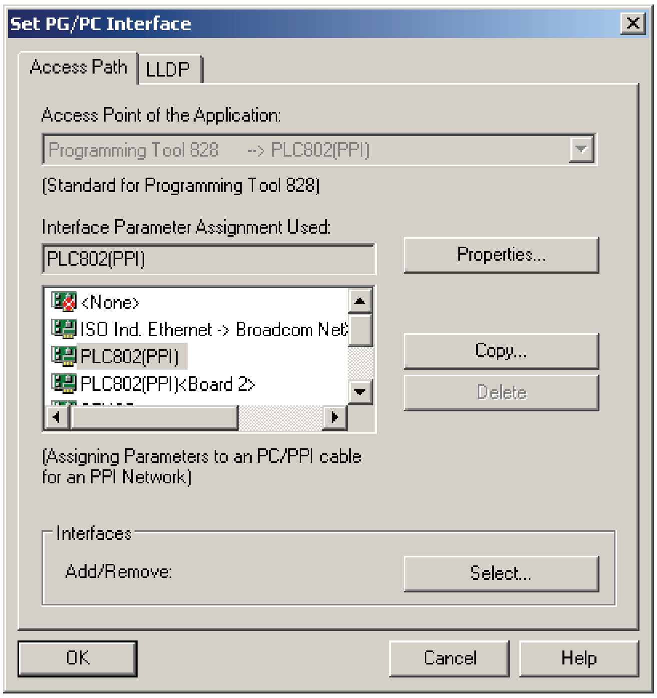
Module (COM 1)

Protocol PPI

Transmission Rate

Mode 11-bit

The "**Set PG/PC Interface**" dialog is displayed as follows:



Set PG/PC Interface

Access Path | LLDP

Access Point of the Application:

Programming Tool 828 --> PLC802(PPI)

(Standard for Programming Tool 828)

Interface Parameter Assignment Used:

PLC802(PPI)

<None>
ISO Ind. Ethernet -> Broadcom Net
PLC802(PPI)
PLC802(PPI)<Board 2>

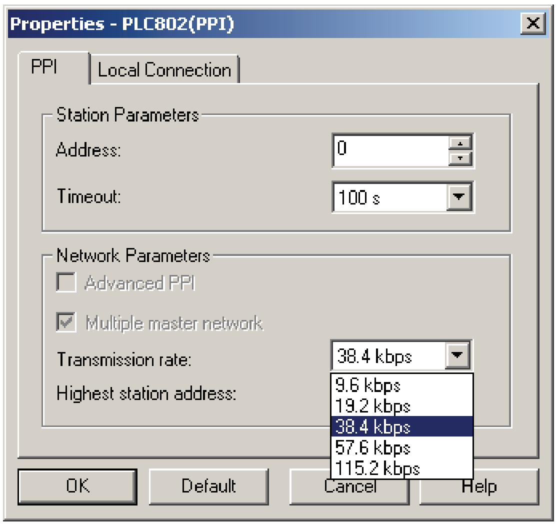
(Assigning Parameters to an PC/PPI cable for an PPI Network)

Interfaces

Add/Remove:

3. Check the PG/PC interface being used. For RS232 communication, you must assign the interface "**PLC802(PPI)**" to the PLC programming tool.

4. Double click the interface "**PLC802(PPI)**" or right-click the "**Properties**" button, and the property dialog is displayed. Set the baud rate for the transmission rate, which the PLC Programming Tool will use to communicate:



Properties - PLC802(PPI)

PPI | Local Connection

Station Parameters

Address: 0

Timeout: 100 s

Network Parameters

□ Advanced PPI

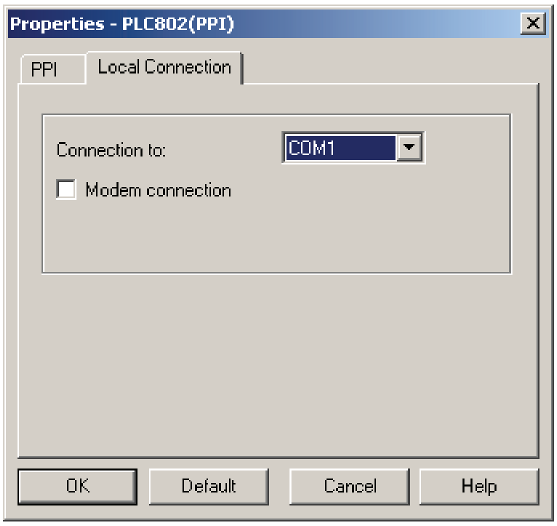
☑ Multiple master network

Transmission rate: 38.4 kbps

Highest station address:

NOTE: The baud rate you select must be in accordance with what you have set on the controller side.

5. Open the "**Local connection**" tab to specify the COM port to which the RS232 (V24) cable is connected:



Properties - PLC802(PPI)

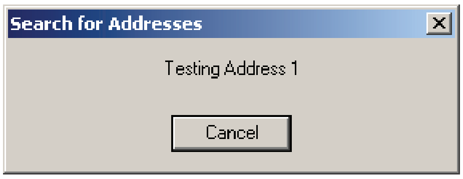
PPI | Local Connection

Connection to: COM1

□ Modem connection

6. Click "**OK**" twice to exit the "**Set PG/PC Interface**" dialog box.

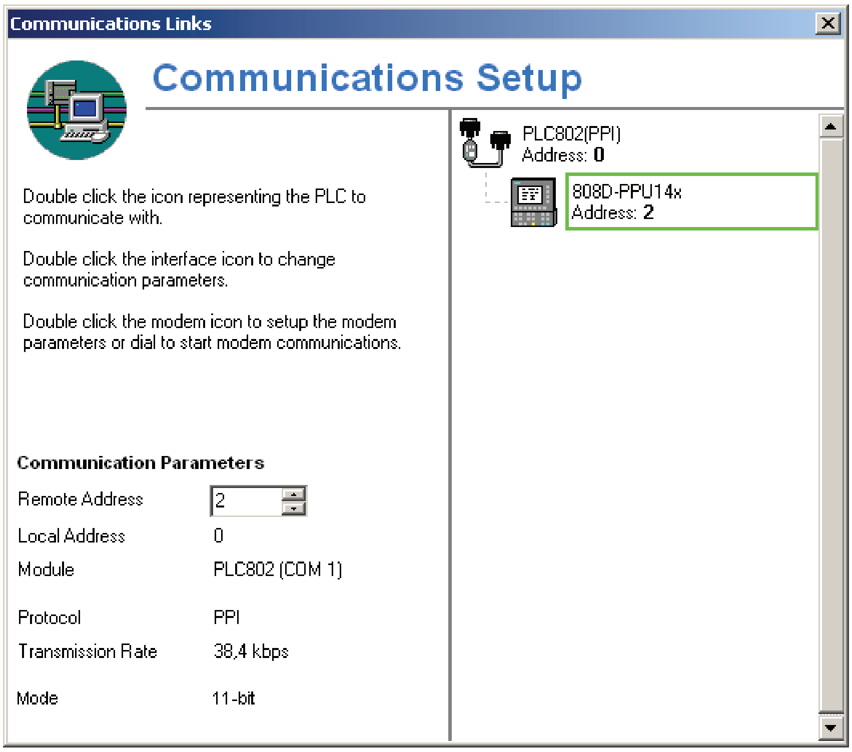
7. On the right of the "**Communications Setup**" dialog box, click the blue text "**Double-Click to Refresh**". It will take several minutes to search for a valid address:



Search for Addresses

Testing Address 1

8. Wait until the icon "**808D-PPU14x, Address 2**" displays as below, and then the connection is ready:



Communications Links

Communications Setup

Double click the icon representing the PLC to communicate with.

Double click the interface icon to change communication parameters.

Double click the modem icon to setup the modem parameters or dial to start modem communications.

Communication Parameters

Remote Address 2

Local Address 0

Module PLC802 (COM 1)

Protocol PPI

Transmission Rate 38,4 kbps

Mode 11-bit

PLC802(PPI)
Address: **0**

808D-PPU14x
Address: **2**

Note

You must activate the connection on the control via the HMI ("**SYSTEM**" operating area > "**PLC**" > "**STEP 7 connect**" > "**Connect. ON**").

2.7 PLC application Download/Upload/Copy/Compare

You can save, copy or over-write the PLC project or the PLC applications in the control system.

This is possible using:

- the PLC Programming Tool
- a USB stick

A **PLC project** contains the PLC user program, including all of the important information (symbols, comments, ...).

You can upload / download the PLC project to the control system with the PLC Programming Tool. With the PLC Programming Tool, you can also import and export it in ".pte" format.

You can also read the PLC project in "**.pte**" format from / to the USB stick directly on the control system.

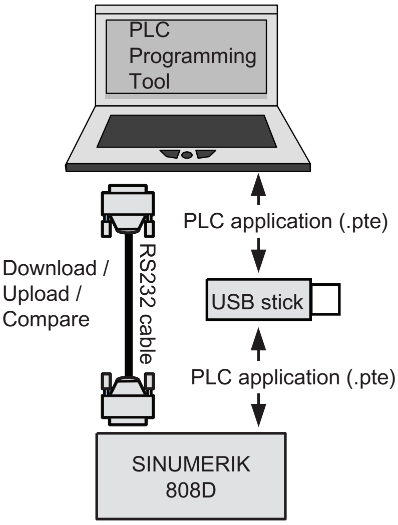


Figure 2-1 PLC applications on the control system

Download

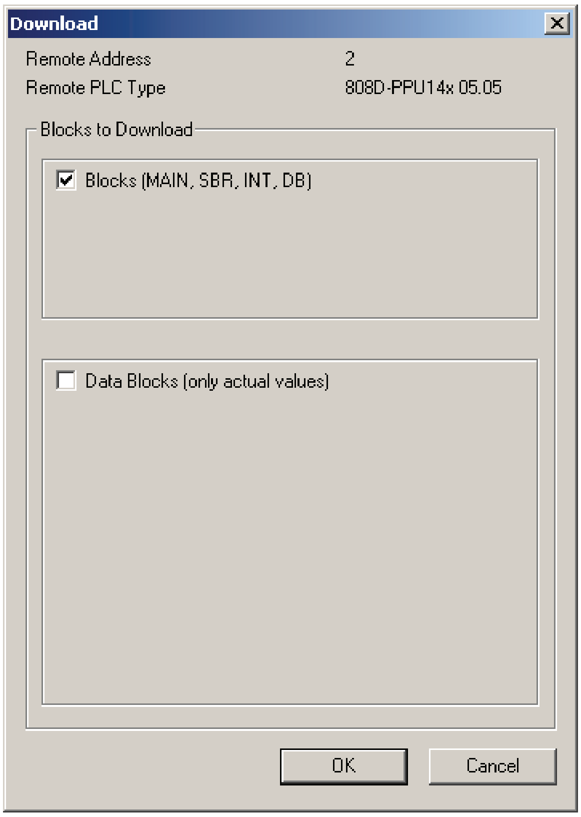
You can write the transferred data into the permanent memory (load memory) of the control system with the PLC Programming Tool or an USB stick.

To download a PLC application with the **PLC Programming Tool**, proceed as follows:

1. Establish the communication between the CNC controller and the PLC Programming Tool with an RS232 cable.

   For information about how to establish a communication between the CNC controller and the PLC Programming Tool, refer to section "Establishing a connection with the RS232 interface (Page 17)".

2. Use the menu command "**File > Download...**" or click the download button ⭳ to start the download, and a download dialog pops up:



Download

Remote Address 2

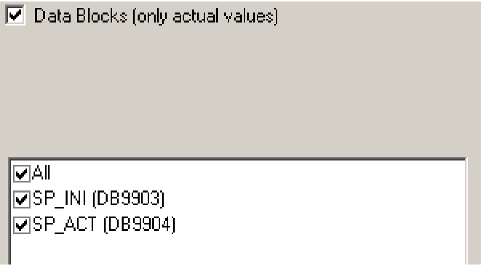
Remote PLC Type 808D-PPU14x 05.05

Blocks to Download

☑ Blocks (MAIN, SBR, INT, DB)

☐ Data Blocks (only actual values)

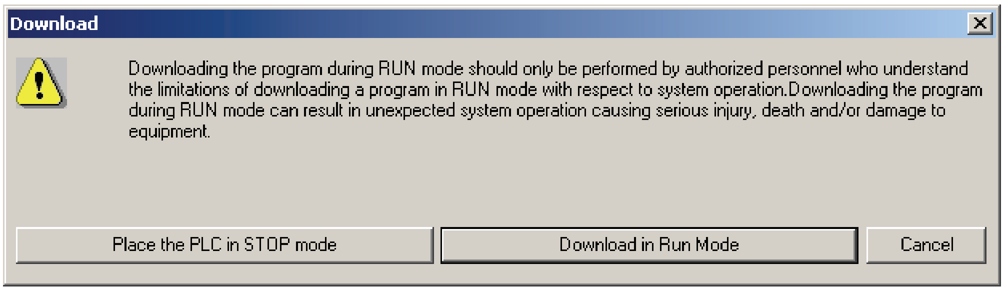
3. Click "**OK**" to proceed directly. You can also select the checkbox "**Data Blocks (only actual values)**" to include the actual values of the data blocks, and then click "**OK**":



☑ Data Blocks (only actual values)

☑All
☑SP_INI (DB9903)
☑SP_ACT (DB9904)

4. Either select to download the PLC application when the PLC is in the **RUN** mode (the "**Download in RUN mode**" button) or in the **STOP** mode (the "**Place the PLC in STOP mode**" button):

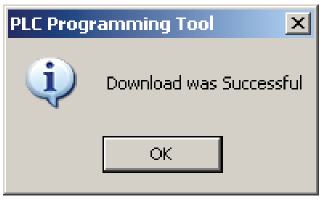


Download

Downloading the program during RUN mode should only be performed by authorized personnel who understand the limitations of downloading a program in RUN mode with respect to system operation.Downloading the program during RUN mode can result in unexpected system operation causing serious injury, death and/or damage to equipment.

It is recommended to download when the PLC is in the STOP mode. Downloading when the PLC is in the RUN mode can cause machine damages or even human injuries.

5. The download starts and it will take several seconds to do it.
6. The download finishes when the message below appears, Click "**OK**" to end your operations:



PLC Programming Tool

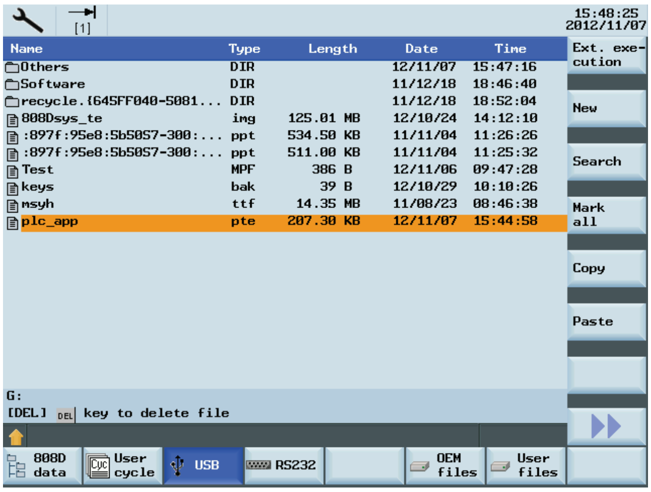
Download was Successful

Note

If you have chosen to download when the PLC is in the **STOP** mode, you can place the PLC at **RUN** mode again with the PLC Programming Tool (click button ▶)

To download an OEM PLC application with an USB stick, perform as follows:

1. Export the PLC application created with the PLC Programming Tool to a USB stick using the menu command "**File > Export...**".
2. Insert the USB stick into the USB interface at the front of the PPU.
3. Open the USB stick via the HMI: "**SYSTEM**" operating area > "**Sys. data**" > "**USB**":



15:48:25
2012/11/07

Name	Type	Length	Date	Time
Others	DIR		12/11/07	15:47:16
Software	DIR		11/12/18	18:46:40
recycle.{645FF040-5081...	DIR		11/12/18	18:52:04
808Dsys_te	img	125.01 MB	12/10/24	14:12:10
:897f:95e8:5b50S7-300:...	ppt	534.50 KB	11/11/04	11:26:26
:897f:95e8:5b50S7-300:...	ppt	511.00 KB	11/11/04	11:25:32
Test	MPF	386 B	12/11/06	09:47:28
keys	bak	39 B	12/10/29	10:10:26
msyh	ttf	14.35 MB	11/08/23	08:46:38
plc_app	pte	207.30 KB	12/11/07	15:44:58

G:
[DEL] key to delete file

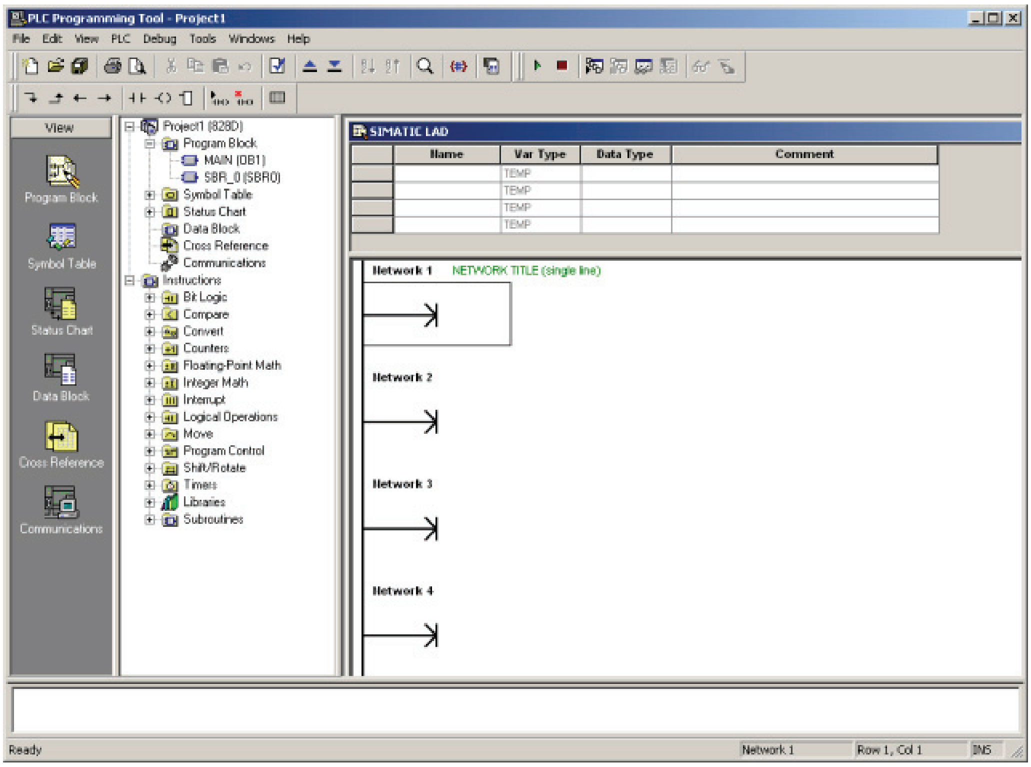
4. Select the .pte file and press the "**Copy**" softkey.
5. Press the "**808D data**" softkey, and enter the "NCK/PLC data" folder by pressing the <**INPUT**> key.
6. Press the "**Paste**" softkey.
7. A warning note appears warning you that the original .pte file will be overwritten. Press "**OK**" to continue.
8. Download has been completed when the progress bar disappears.

Upload

You can backup a PLC application from the permanent memory of the control system using the **PLC Programming Tool** or an USB stick.

To upload a PLC application using the **PLC Programming Tool**, proceed as follows:

1. Create a new and empty PLC application using the PLC Programming Tool:

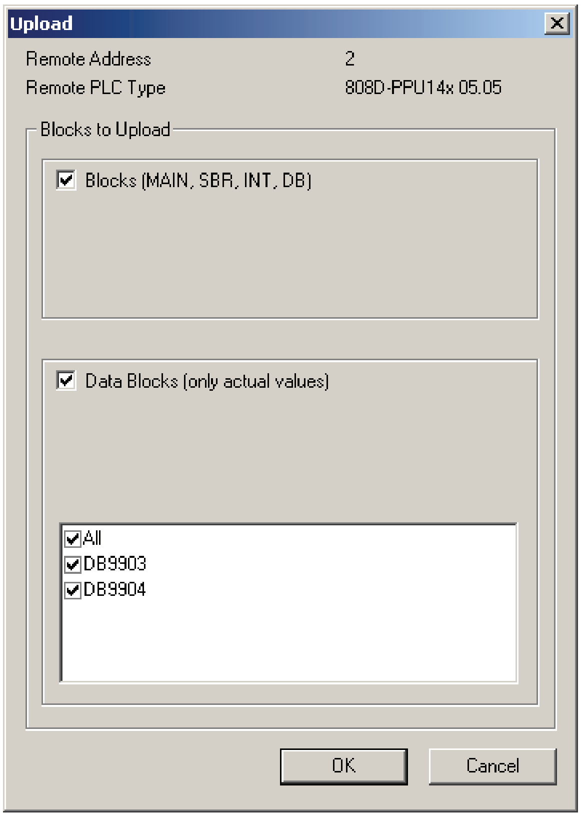


	Name	Var Type	Data Type	Comment
		TEMP		
		TEMP		
		TEMP		
		TEMP		

2. Establish the communication between the CNC controller and the PLC Programming Tool with an RS232 cable.

   For information about how to establish a communication between the CNC controller and the PLC Programming Tool, refer to section "Establishing a connection with the RS232 interface (Page 17)".

3. Use the menu command "**File > Upload...**" or click the upload button ⏏ to start the upload, and the upload dialog pops up:



Upload

Remote Address 2

Remote PLC Type 808D-PPU14x 05.05

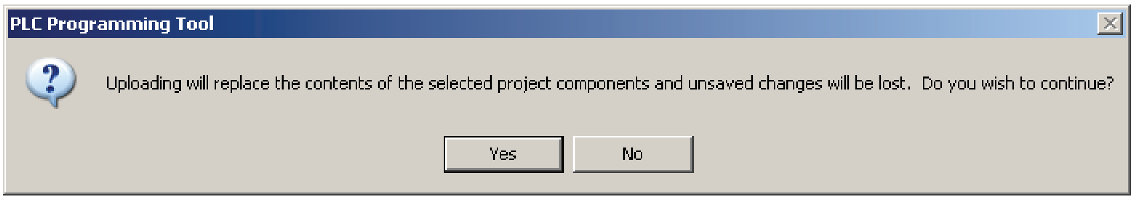
Blocks to Upload

☑ Blocks (MAIN, SBR, INT, DB)

☑ Data Blocks (only actual values)

☑All
☑DB9903
☑DB9904

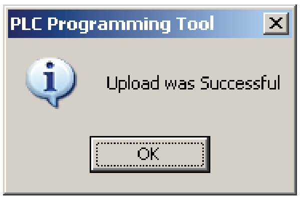
4. Click the "**OK**" button.
5. An information dialog appears saying "Uploading will replace the contents of the selected project components and unsaved changes will be lost. Do you wish to continue?":



PLC Programming Tool

Uploading will replace the contents of the selected project components and unsaved changes will be lost. Do you wish to continue?

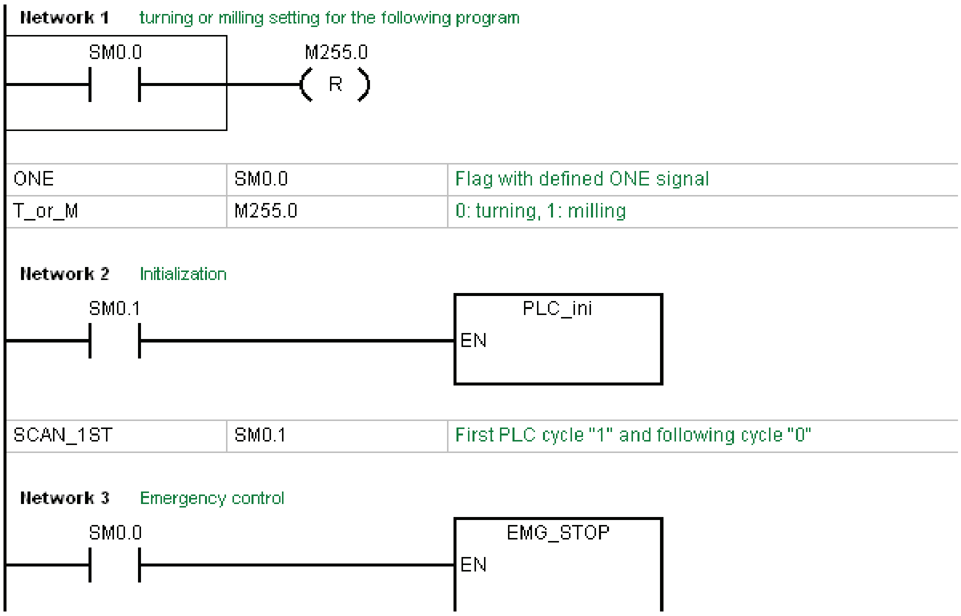
6. Click "**Yes**" and the upload will start.
7. The upload has been completed when the message below appears:



PLC Programming Tool

Upload was Successful

8. Click "**OK**" and you can view the upload result:



Network 1 turning or milling setting for the following program

ONE	SM0.0	Flag with defined ONE signal
T_or_M	M255.0	0: turning, 1: milling

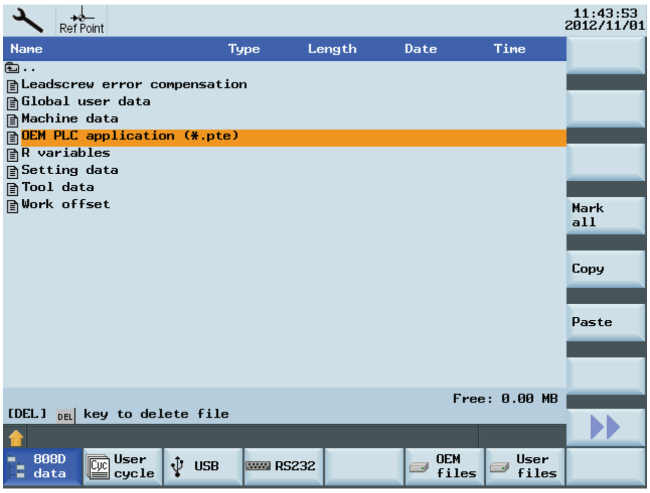
Network 2 Initialization

SCAN_1ST	SM0.1	First PLC cycle "1" and following cycle "0"

Network 3 Emergency control

To upload a PLC application using an USB stick, proceed as follows:

1. Insert the USB stick into the USB interface at the front of the PPU.
2. Locate the PLC application (**.pte** file): "**SYSTEM**" operating area > "**Sys. data**" > "**808D data**" > "**NCK/PLC data**" > "**OEM PLC application (*.pte)**":



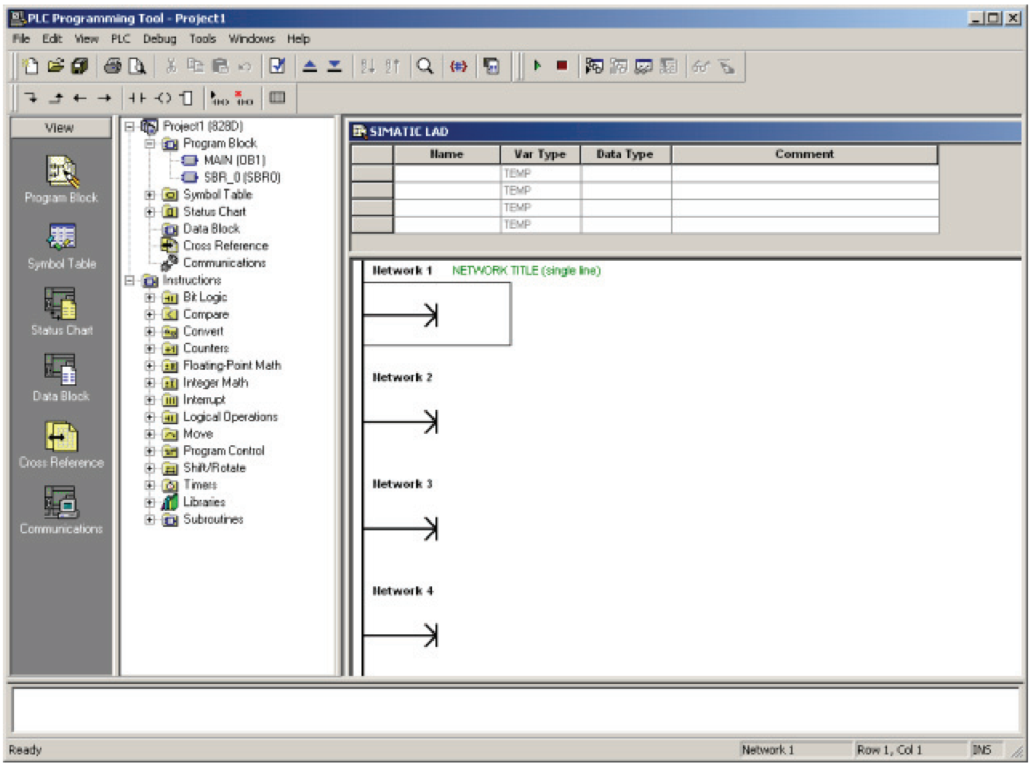
Name	Type	Length	Date	Time
..				
Leadscrew error compensation				
Global user data				
Machine data				
OEM PLC application (*.pte)				
R variables				
Setting data				
Tool data				
Work offset				

Free: 0.00 MB

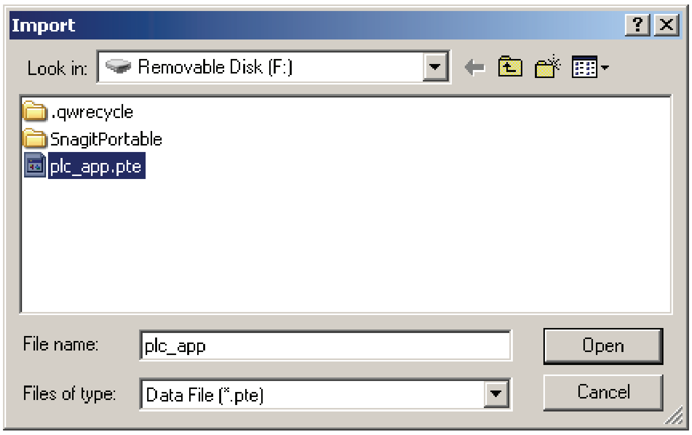
[DEL] DEL key to delete file

3. Press the "**Copy**" softkey.
4. Press the "**USB**" softkey and paste the .pte file into the USB stick with the "**Paste**" softkey.
5. Unplug the USB stick and plug it into the PC/PG.

6. Create an empty PLC application with the PLC Programming Tool:



7. Import the .pte file from the USB stick using the menu command "**File > Import...**":



Look in: Removable Disk (F:)

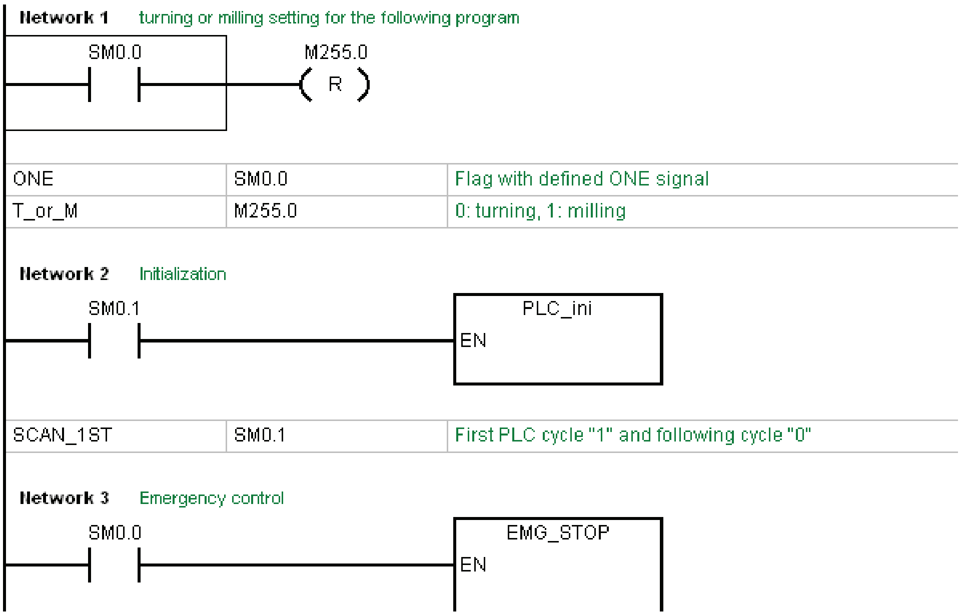
File name: plc_app

Files of type: Data File (*.pte)

8. Click "**Open**" button or double-click on the **.pte** file.

9. It will take several seconds to import the **.pte** file.

10.After successfully importing the PLC application, you can view the imported result:

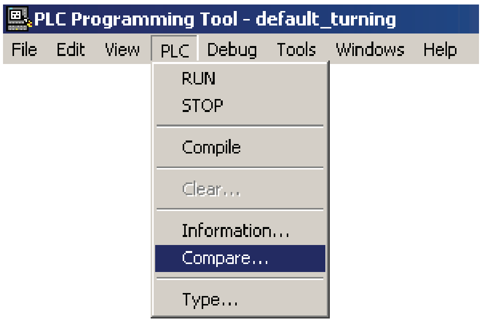


ONE	SM0.0	Flag with defined ONE signal
T_or_M	M255.0	0: turning, 1: milling

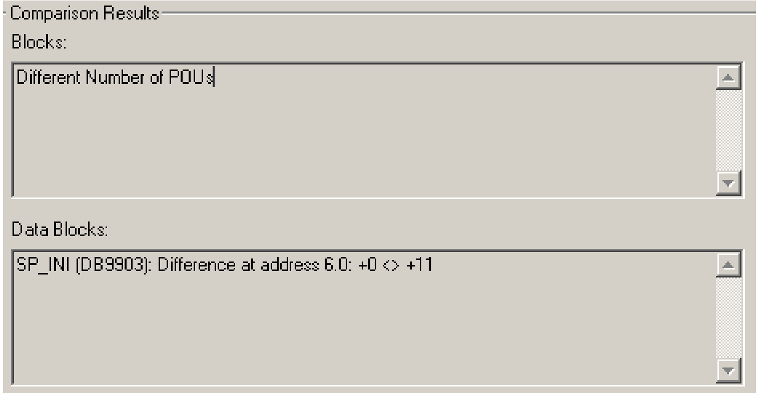
SCAN_1ST	SM0.1	First PLC cycle "1" and following cycle "0"

Compare

You can compare the project in the **PLC Programming Tool** with the project on the control system by using the menu command "PLC > Compare...":



Press the "**Begin**" button and the comparing begins. Wait for a few seconds, then you can view the compare results:


Comparison Results

Blocks:

Different Number of POUs

Data Blocks:

SP_INI (DB9903): Difference at address 6.0: +0 <> +11

Version display

Call up via the "**SYSTEM**" operating area > "**Serv. displ.**" > "**Version**"

- **PLC application**
  The transferred project that is active in the working memory of the PLC after the control system has started up.

  In the **PLC 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.

Descriptions of the PLC subroutines 3

3.1 PLC machine data

Table 3- 1 USER_DATA_INT

No.	Unit	Description
14510 [12]	-	Layout of the traverse keys
14510 [13]	0.1s	Time for spindle braking
14510 [20]	-	The maximum number of tool bits
14510 [21]	0.1s	Turret clamping time
14510 [22]	0.1s	Monitoring time for searching a tool
14510 [24]	1 min	Lubricating intervals
14510 [25]	0.01s	Lubricating duration

Table 3- 2 USER_DATA_HEX

No.	Descriptions
14512 [16]	Bit 1: function of chip remover (milling) Bit 2: function of safe door (milling) Bit 3: when the function of safe door is active, it can be triggered by M01/M02 (milling). Bit 4: control of X axis rotation monitoring Bit 5: control of Y axis rotation monitoring Bit 6: control of Z axis rotation monitoring Bit 7: handwheel assignment with the MCP / HMI
14512 [17]	Bit 0: turret (turning); tool magazine (milling) Bit 1: clamping function (turning) Bit 2: tail stock function (turning) Bit 3: selection between handwheel and hand-held unit (0: handwheel; 1: hand-held unit)
14512 [18]	Special configurations for a machine Bit 2: automatic lubrication after first power-on (factory setting) Bit 4: signal that external spindle stops Bit 5: spindle positioning direction Bit 6: the hardware limit is independent from a PLC program Bit 7: each feed axis has a hardware limit switch (activated when Bit 6 = 0)
14512 [19]	Bit 1: function of spindle braking Bit 2: password clearing by power-on (0: delete the password; 1: do not delete the password) Bit 7: manual machine function (this function become active if you have installed licensed turning machine system and called it with a PLC subroutine)

3.2 Conventions for the symbols used in the subroutines

The symbols used in the subroutines follow the conventions listed below:

- Leading characters designate the destinations of interface signals
  - P_: to PLC interface
  - H_: to HMI interface
  - N_: to NCK interface
  - M_: to MCP interface
- Subsequent characters are for areas
  - N_: NCK
  - C_: Channel
  - 1_: Axes
  - M_: MCP

  Other short forms of the symbols
  - HWL: **H**ard**w**are **L**imit
  - HW: **H**and**w**heel
  - RT: **R**apid **T**raverse
  - TK: **T**raverse **k**ey
  - ACT: **Act**ive
  - SEL: **Sel**ected
- A symbol consists of a maximum of 11 upper case characters and numbers (including the leading character). Except for underlines, you cannot use any other special symbols like =, +, -, [ ], etc.

Symbol tables 1 to 2

The symbol tables 1 to 2 are for the manufacturer-defined SINUMERIK 808D I/O. Here you can define input/output tables for your machines.

Symbol tables 3 to 5, 7 to 13

These ten symbol tables are reserved for the subroutines.

Note

Important!

If the colour of a symbol is displayed in red, this indicates that the naming does not follow the rules of the PLC Programming Tool. In this case, you must check:

- whether the name uses special symbols like =, +, -, [ ], etc..
- whether the leading character is a number.

If there is a red wave under a symbol name or address, it indicates that you have used the same symbol name or address for another symbol (there is also a red wave under its name or address). In this case, you must check the repeated name or address and change it.

Symbol table 6: MANMACH

This symbol table is for the function of a manual machine.

Symbol table 14: ASUP

This symbol table is for the ASUP function. It defines the start, operating results and other properties of the ASUP function.

Symbol table 15: PLC_sle_PP

This symbol table is for the PLC to select part programs. It defines the interface signals for the PLC to select part programs.

Symbol table 16: IS_MCP

This symbol table is for the manufacturer-defined MCP interface signals. With this symbol table you can define the MCP signal tables for your machines.

Within the PLC subroutine library and all the subroutines, all the MCP-related signals are defined in this symbol table.

An MCP signal is transferred to the interface area via a subroutine. For the SINUMERIK 808D MCP, you can use subroutine 37 to transfer the MCP input/output signals to the interface area. For your own MCP, you just need to create a subroutine to realize the transfer, and the other subroutines are still available.

Symbol table 17: IS_HMI

The symbol table 17 defines interface signals to/from the HMI.

Symbol table 18: IS_AUX

The symbol table 18 defines the interface signals from the NCK channel, including the auxiliary functions, D functions, H function, etc.

Symbol table 19: IS_NCK

The symbol table 19 defines the interface signals to/from the NCK.

Symbol table 20: IS_CHA

The symbol table 20 defines the interface signals to/from the channel.

Symbol table 21, 22, 23, 24, 25: IS_AX1, IS_AX2, IS_AX3, IS_AX4, IS_AX5

These five symbol tables define the interface signals to/from the axes.

Symbol table 26: IS_AX_P1

The symbol table 26 defines the interface signals to/from the PLC axis.

Symbol table 27: MD_PLC

The symbol table 27 defines the PLC machine data from the NCK.

Symbol table 28: ALARM

The symbol table 28 defines each bit of a PLC user alarm (V1600 000x).

Symbol table 29: NV_MEM

The symbol table 29 defines all the user data (128 byte, retentive) used in the standard subroutines.

Symbol table 30: SPC_MEM

The symbol table 30 defines the seven special memories of the SINUMERIK 808D (SM0.0 to SM0.6).

Symbol table 31: SBR_MEM

The symbol table 31 defines the memories used in the standard subroutines. These memories can be used as the global variables.

Symbol table 32: reserved for the subroutines

The symbol table 32 is a reserved symbol table.

3.3 Subroutine 20 - AUX_MCP (machine auxiliary functions)

Purpose

Subroutine 20 is used to control machine auxiliary functions like lamps, safe door (milling) and chip remover (milling).

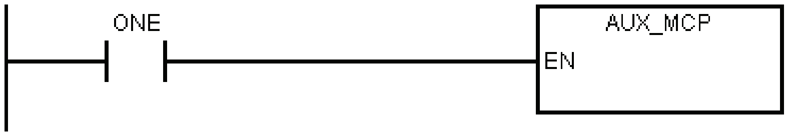
Local variable definition

None

Relevant PLC machine data

None

Example for calling subroutine 20



3.4 Subroutine 21 - AUX_LAMP (working lamp)

Purpose

Subroutine 21 is used to control working lamp and can be called in AUX_MCP. One-time pressing of the "**LAMP**" key activates the working lamp while double pressing de-activates the working lamp.

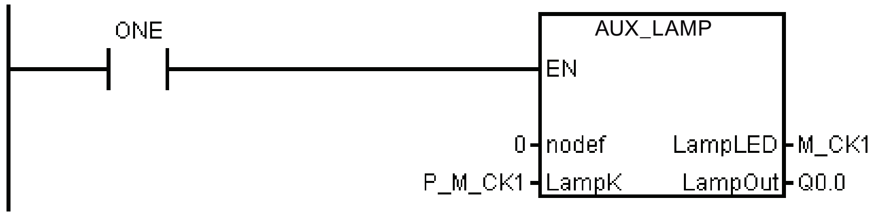
Local variable definition

None

Relevant PLC machine data

None

Example for calling subroutine 20



3.5 Subroutine 22 - AUX_SAFE_DOOR (safe door)

Purpose

Subroutine 22 is used to control safe door and can be called in AUX_MCP.

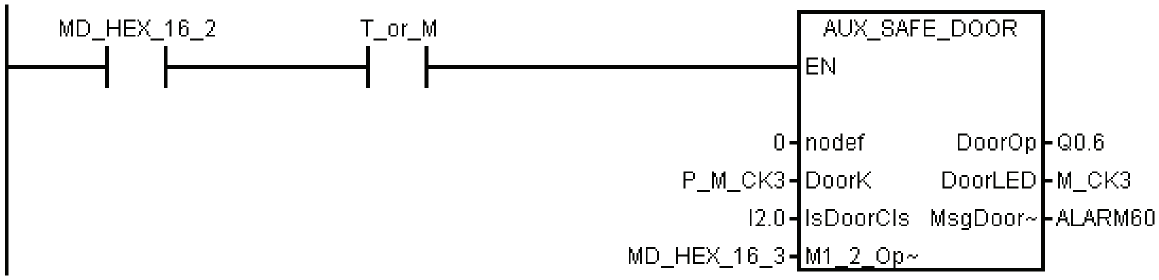
Local variable definition

None

Relevant PLC machine data

No.	Description
14512 [16].2	Selection of safe door function (0: do not use; 1: use)
14512 [16].3	Safe door function is activated by M01/M02 (0: deactivate; 1: activate)

Example for calling subroutine 22



3.6 Subroutine 23 - AUX_CHIP (chip remover)

Purpose

Subroutine 23 is used to control chip remover and can be called in AUX_MCP. At first-time pressing of the "Chip Forward" key the output "ChipFwd" is high; at second-time pressing of the "Chip Forward" key the output "ChipFwd" becomes low. When the output "ChipFwd" is low, the output "ChipRev" becomes high at pressing "Chip Reverse" key and becomes low at releasing the "Chip Reverse" key.

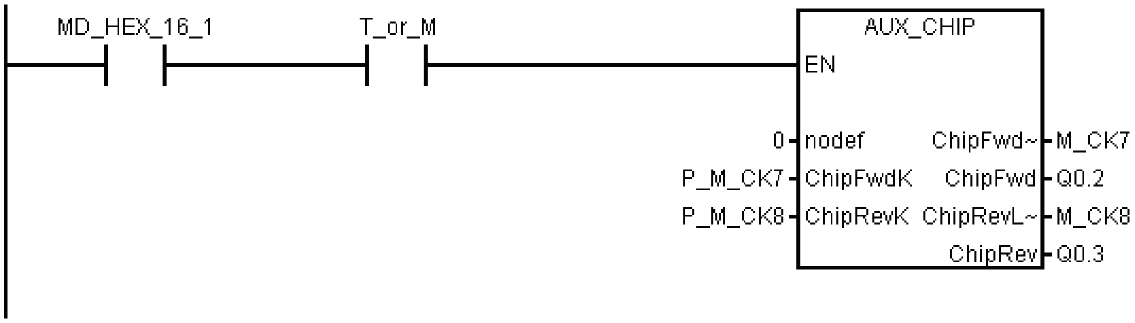
Local variable definition

None

Relevant PLC machine data

No.	Description
14512 [16].1	Selection of chip remover function (0: do not use; 1: use)

Example for calling subroutine 23



3.7 Subroutine 31 - PLC_ini_USR_ini (user initialization)

Purpose

Subroutine 31 is used for user initialization and can be called in subroutine PLC_INI. Since the subroutine PLC_INI can be called only during the first PLC cycle, the subroutine PLC_ini_USER_ini can be called also only during the first PLC cycle.

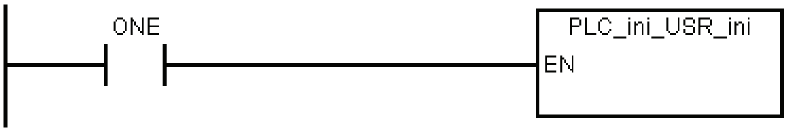
Local variable definition

None

Relevant PLC machine data

None

Example for calling subroutine 31



3.8 Subroutine 32 - PLC_INI (PLC initialization)

Purpose

Subroutine 32 is executed at the first PLC cycle (SM0.1). This subroutine set NCK interface signals according to the machine settings defined by PLC machine data. In this subroutine, the following interface signals are set:

- DB3200.DBX6.7: feed override of the NCK channel becomes active
- DB380x.DBX1.5: measurement system 1 of the axes is active
- DB380x.DBX1.7: feed override of the axes is active
- DB1700.DBX1.3: active ROV

At the end of this subroutine, subroutine 31 (PLC_ini_USR_ini) is automatically called. You can program the initialization of customer PLC project in the subroutine 31.

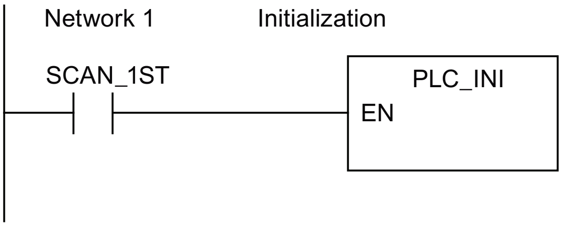
Local variable definition

None

Relevant PLC machine data

None

Example for calling subroutine 32



3.9 Subroutine 33 - EMG_STOP

NOTICE
Do check whether this subroutine complies with relevant safety requirements or not.

Purpose

Subroutine 33 handles the Emergency Stop. Pressing down the Emergency Stop button produces an emergency stop alarm and disables the 65 enable signal for SINAMICS V60. If you want to clear the emergency stop alarm, you must first release the Emergency Stop button and then press the **RESET** key on the MCP.

This subroutine can activate the alarm below:

Alarm 700016: DRIVE NOT READY

Local variable definition

Table 3- 3 Inputs

Variable	Type	Description
DELAY	WORD	Power on/off sequence delay (unit: 10ms)
E_KEY	BOOL	Emergency Stop key (NC)
Drv_RDY	BOOL	Drive Ready: signal for SINAMICS V60 ready
HWL_ON	BOOL	Triggered by the hardware limit switch of any axis (NO) [1]
SpStop	BOOL	External spindle stop (NO) [2]
Drv_ALM	BOOL	Drive alarm: signal for SINAMICS V60 alarm
NO: Normal Open NC: Normal Close		

[1] This input can come from signal OVlmt of subroutine 40, and triggers the emergency stop when the hardware limit appears.

[2] Before the drive system disables the 65 enable signal, the PLC detects the spindle stop signal from NCK to ensure that the spindle has stopped.

Table 3- 4 Outputs

Variable	Type	Description
Drv1_En65	BOOL	65 enable signal of the first SINAMICS V60
Drv2_En65	BOOL	65 enable signal of the second SINAMICS V60
Drv3_En65	BOOL	65 enable signal of the third SINAMICS V60

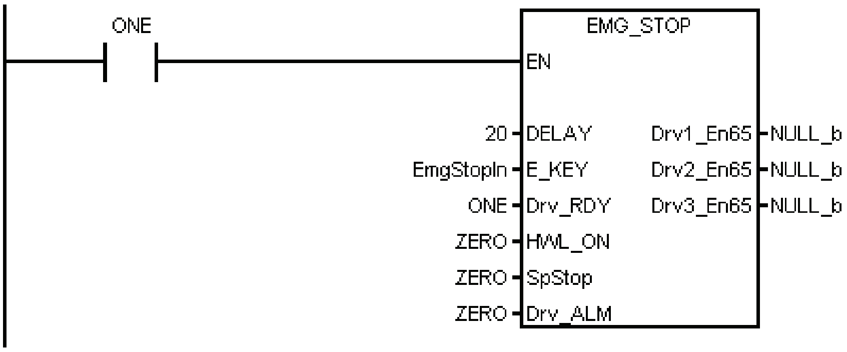
One timer is used in this subroutine:

D_PON	T26	Power-on delay

Relevant PLC machine data

No.	Value	Description
14512 [18].4	1	Spindle has an external stop signal
	0	Spindle has no external stop signal

Example for calling subroutine 33



3.10 Subroutine 37 - MCP_NCK (MCP and HMI signal processing)

Purpose

Subroutine 37 is used to transfer the interface signals from the MCP and HMI to the NCK interfaces, and thus to activate the specific operating mode and control sequences. It has the following main functions:

- Selecting specific operating mode
- Selecting override
- Transferring signals from the HMI to NCK interfaces (for instance, program control, handwheel, etc.)
- Controlling the axis traversing signal according to the PLC machine data

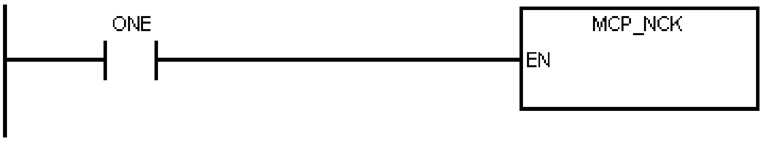
Local variable definition

None

Assigned global variables

None

Example for calling subroutine 37



3.11 Subroutine 38 - MCP_Tool_Nr (display tool number on the MCP)

Purpose

Subroutine 38 is used to display active tool number (< 100) with the 7-segment LED on the MCP. For a tool number ≥ 100, it displays "FF".

Local variable definition

None

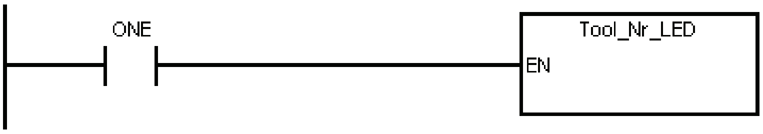
Assigned global variables

None

Relevant PLC machine data

None

Example for calling subroutine 38



3.12 Subroutine 39 - HANDWHL (selecting a handwheel according to HMI interface signals)

Purpose

Subroutine 39 is used to select one of the two handwheels to control an axis (X, Y or Z) in the machine coordinate system or the workpiece coordinate system according to the HMI signals. With the HANDWHEEL key and axis selection key on the MCP, you can assign the handwheel 1 in the workpiece coordinate system to any axis.

Note

You cannot use subroutine 39 together with subroutine 41 - MINI_HHU.

Local variable definition

None

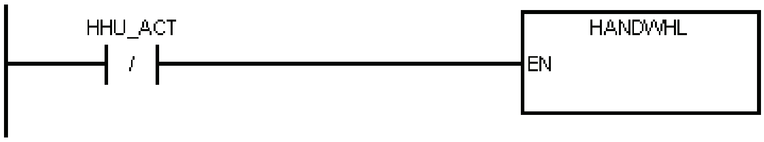
Assigned global variable

None

Relevant PLC machine data

No.	Value	Description
14512 [16].7	1	Handwheel assignment with the HMI
	0	Handwheel assignment with the MCP

Example for calling subroutine 39



3.13 Subroutine 40 - AXIS_CTL (controlling the spindle and axes)

Purpose

Subroutine 40 is used to control the drive pulse enable (DB380xDBX4001.7) and controller enable (DB380xDBX2.1), monitoring the hardware limits and the reference cam signals, and controlling the enable signal for the spindle according to a spindle command (for example, SPINDLE CW, SPINDLE CCW, Mo3, M04, SPOS, etc.). The motor brake is automatically controlled by the SINAMICS V60 drives.

This subroutine provides two ways to realize the hardware limit control:

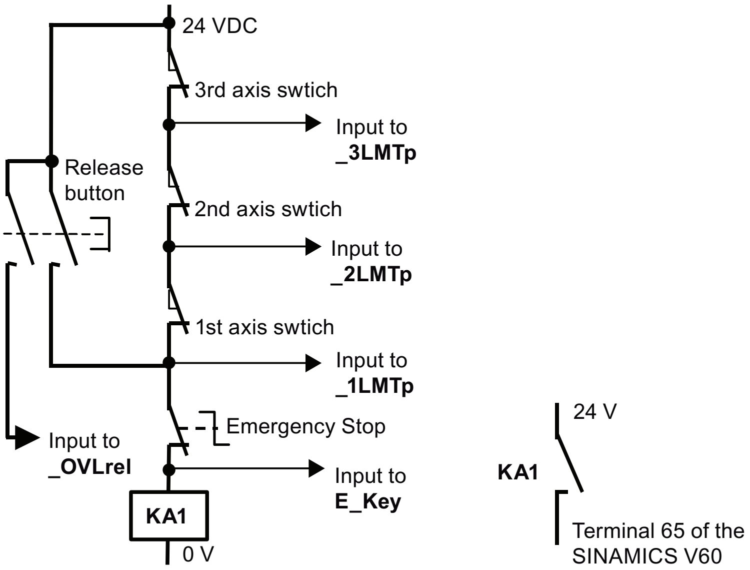
- PLC solution (MD14512 [18] bit 6 = 0)

  Each feed axis has one (MD14512 [18] bit 7 = 1) or two (MD14512 [18] bit 7 = 0) hardware limit switches. This subroutine activates the NCK hardware limit function via the NCK interface DB380xDBX1000.0 or DB380xDBX1000.1 according to the configurations of the hardware limit switches, and thus makes the NCK produce a feed stop signal to an over-distance axis.

  Furthermore, you can also connect the output **OVlmt** of this subroutine with the input **HWL_ON** of subroutine 33 to activate the Emergency Stop automatically once the hardware limit of any axis has been reached.

- Hardware solution (MD14512 [18] bit 6 = 1)

  This solution is independent of the PLC and thus is much safer:



Note

The connection between the terminal 65 of the SINAMICS V60 and the +24 V signal is cut off automatically at any of the hardware limits or an EMERGENCY STOP.

Encoding the hardware limit switches					Result
E_Key	_1LMTp	_2LMTp	_3LMTp	Direction	
0	1	1	1	-	EMERGENCY STOP active
0	0	1	1	DB3900.DBX4.7	1st + over limit
0	0	1	1	DB3900.DBX4.6	1st - over limit
0	0	0	1	DB3901.DBX4.7	2nd + over limit
0	0	0	1	DB3901.DBX4.6	2nd - over limit
0	0	0	0	DB3902.DBX4.7	3rd + over limit
0	0	0	0	DB3902.DBX4.6	3rd - over limit

In the hardware solution above, the feed stop signals for all axes (for example, disconnect the terminal 65 of the SINAMICS V60 via a relay) can be activated via the hardware limit switches when any of the hardware limits is reached or an EMERGENCY STOP happens. You can check the information of the PLC diagnostics from the encoding of the hardware limit switches shown in the table above, and identify the cause (Emergency Stop button or a hardware limit switch of an axis) of the EMERGENCY STOP signal.

Note

When using the hardware solution, you must take below information into consideration:

- You must assign the axes one by one; for example, X axis, Z axis, spindle or X axis, Y axis, Z axis, spindle. You must not assign the axes like X axis, Y axis, spindle, Z axis.
- You must set constant "1" (i.e. SM0.0) to the input signals of the hardware limits for undefined axes; otherwise, the hardware limits of the undefined axes can be activated.

Local variable definition

Table 3- 5 Inputs

Name	Type	Description
NODEF	WORD	Reserved word
DrvEn_65	BOOL	Drive enable signal from AXIS_CTL (SBR40)
Drv1_RDY	BOOL	Ready signal of the 1st drive, reserved [1)]
Drv2_RDY	BOOL	Ready signal of the 2nd drive, reserved [1)]
Drv3_RDY	BOOL	Ready signal of the 3rd drive, reserved [1)]
OPTM	BOOL	Brake release switch (NO), used for drive optimization, reserved
_1LMTp	BOOL	1st axis hardware limit switch + (NC) [2)]
_1LMTn	BOOL	1st axis hardware limit switch - (NC)
_1REF	BOOL	1st axis reference cam (NO)
_2LMTp	BOOL	2nd axis hardware limit switch + (NC) [2)]
_2LMTn	BOOL	2nd axis hardware limit switch - (NC)
_2REF	BOOL	2nd axis reference cam (NO)
_3LMTp	BOOL	3rd axis hardware limit switch + (NC) [2)]
_3LMTn	BOOL	3rd axis hardware limit switch - (NC)

Name	Type	Description
_3REF	BOOL	3rd axis reference cam (NO)
_4REF	BOOL	Reserved

[1] The PLC does not need to process the drive ready signals and alarm signals received from the SINAMICS V60 because the SINUMERIK 808D has already processed these signals.

[2] The hardware limit + is used for the input if there is only one hardware limit switch or when the hardware solution is used.

Table 3- 6 Outputs

Name	Type	Description
_1BRK	BOOL	1st brake release output (high active, reserved) [1]
_2BRK	BOOL	2nd brake release output (high active, reserved) [1]
_3BRK	BOOL	3rd brake release output (high active, reserved) [1]
OVlmt	BOOL	Over-distance output (active at any hardware limit, high active)

[1] Motor brakes are controlled by the SINAMICS V60 drives themselves.

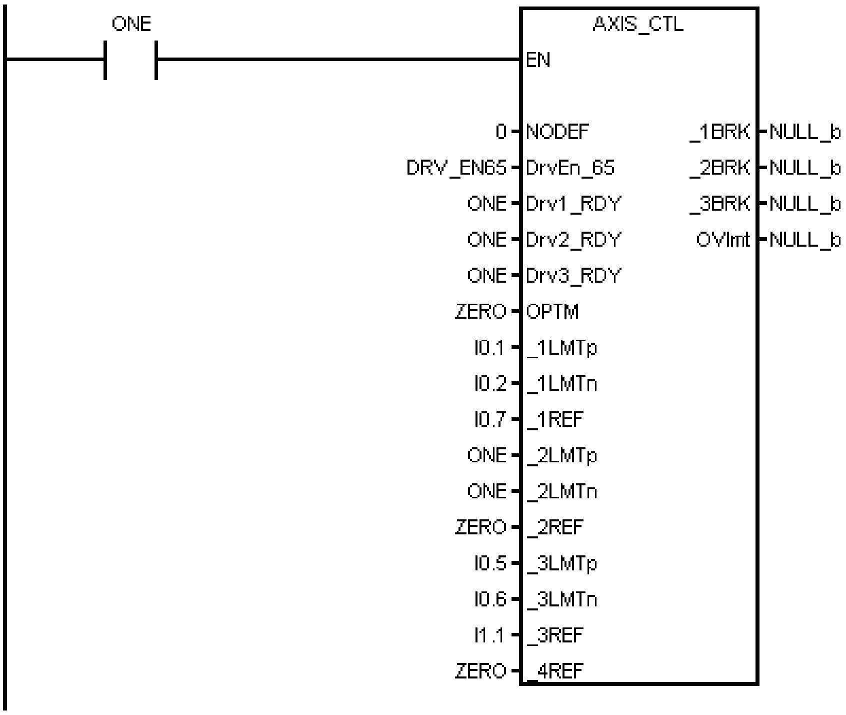
Assigned global variables

SP_CMD	M138.1	Spindle start command (CW or CCW)

Relevant PLC machine data

No.	Value	Description
14512 [18].6	1	Overtravel employs the hardware solution
	0	Overtravel employs the PLC solution
14512 [18].7	1	Each axis has only one hardware limit switch
	0	Each axis direction has an hardware limit switch

Example for calling subroutine 40



3.14 Subroutine 41 - MINI_HHU (handwheel hand-held unit)

Purpose

Subroutine 41 is used to support the customer's handheld units. With a handheld unit, you can assign the handwheels to X axis, Y axis and Z axis, and select incremental override X1, X10, X100 at the same time. You can then use the handwheels to control the movements of your machine.

Local variable definition

None

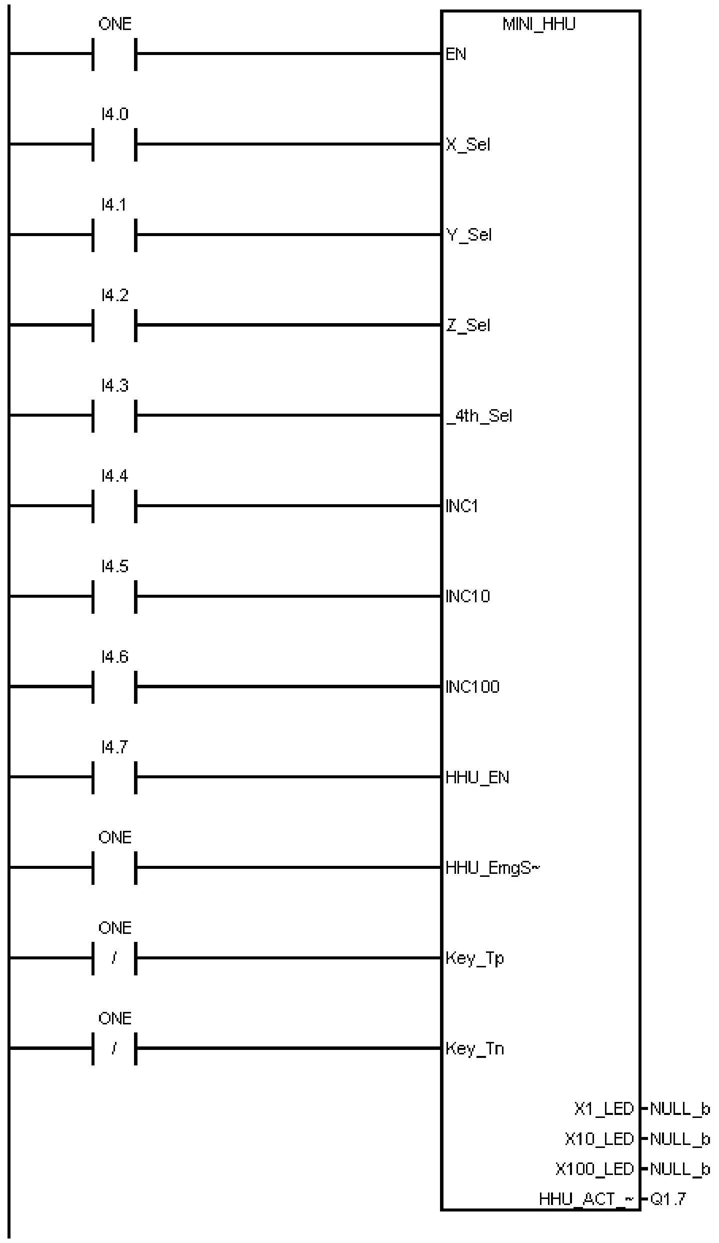
Assigned global variables

None

Relevant PLC machine data

None

Example for calling subroutine 41



3.15 Subroutine 42 - SPINDLE (spindle control)

Purpose

Subroutine 42 is used for spindle control, including the spindle braking function. When the braking function is activated, in the JOG mode press the "SPINDLE LEFT" key or "SPINDLE RIGHT" key and then press the "SPINDLE STOP" key, after that, the spindle brakes. In the AUTO mode, the spindle brakes when it changes rotating direction or coasts down. When the spindle brakes, corresponding output becomes active; meanwhile, the spindle does not accept rotary command until braking completes.

Local variable definition

Table 3- 7 Inputs

Name	Type	Description
DELAY	WORD	Spindle braking duration (unit: 0.1s)
DrvEn	BOOL	Drive enable
SP_EN	BOOL	Spindle action condition (1: allowed; 0: not allowed)
IsBrake	BOOL	Spindle braking function (1: enabled; 0: forbidden)

Table 3- 8 Outputs

Name	Type	Description
SP_brake	BOOL	Spindle braking output
SP_LED	BOOL	Spindle running state

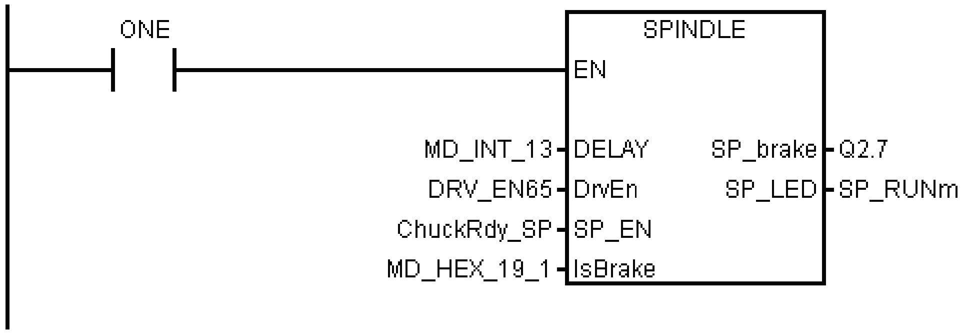
Assigned global variables

SP_B_CMD	BOOL	Spindle braking command
T11	TIMER	Spindle braking timer

Relevant PLC machine data

No.	Type	Description
14510 [13]	BOOL	Spindle braking duration (unit: 0.1s)
14510 [19].1	BOOL	Selection of spindle braking function (1: enabled; 0: forbidden)

Example for calling subroutine 42



3.16 Subroutine 43 - MEAS_JOG (measurement in the JOG mode)

Purpose

Subroutine 43 is used to process the information from the measuring probe and to realize the "measuring in the JOG mode" function. You can use this subroutine to calibrate the probe and measure a tool.

The precondition for calling this subroutine is to call subroutine MCP_NCK (SBR38) in the main program. The "measuring in the JOG mode" function is automatically deactivated if you have changed the operating mode when the function becomes active.

Local variable definition

Table 3- 9 Inputs

Name	Type	Description
Meas_Enable	BOOL	Activating the function of "measuring in JOG mode"
DB1400.DBD64	DWORD	Valid tool number DB1400.DBD64

Assigned global variables

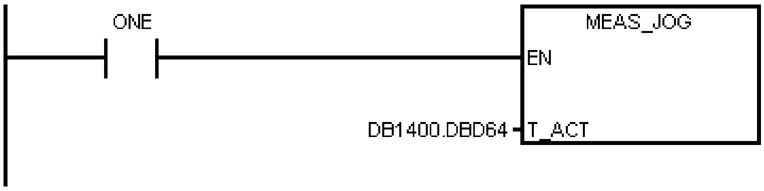
MEAS_OPAUT	M240.0	Measuring in the AUTO mode
CHL_HMI	M240.2	From HMI signals: mode changes during measurement
NO_KEY	M240.3	No JOG key available for the axes
FDI_MEASJOG	M240.5	Meas_JOG forbidden for feed
ON_MEASJOG	M240.6	Meas_JOG activated
PROBE_ON	M240.7	Probe signal released
JOG_MEASJOG	M241.0	Operating mode manually output to Meas_JOG
AUT_MEASJOG	M241.1	Operating mode manually output to Meas_JOG
CHL_MEASJOG	M241.2	Operating mode change forbidden to Meas_JOG
KEY_MEASJOG	M241.3	JOG key Meas_JOG

RES_MEASJOG	M241.4	Reset Meas_JOG
ESC_MEASJOG	M241.5	Interrupt Meas_JOG
DRY_MEASJOG	M241.6	Dry run Meas_JOG
SBL_MEASJOG	M241.7	Single block Meas_JOG

Relevant PLC machine data

None

Example for calling subroutine 43



3.17 Subroutine 44 - COOLING (cooling control)

Purpose

Subroutine 44 is used to start/stop cooling using the buttons on the MCP in the JOG mode, or to start (using the auxiliary function M07/M08 in the part program) or to stop (using the M09 in the part program) cooling in the AUTO/MDA mode. Cooling is forbidden in case of EMERGENCY STOP, cooling motor overload, program test or under the simulation mode.

This subroutine can activate the following alarms:

- Alarm 700018: motor overload for the cooling pump
- Alarm 700019: low coolant level

Local variable definition

Table 3- 10 Inputs

Name	Type	Description
C_key	BOOL	Manual operating key (triggering signal)
OVload	BOOL	Cooling motor overload (NC)
C_low	BOOL	Low coolant level

Table 3- 11 Outputs

Name	Type	Description
C_out	BOOL	Coolant output
C_LED	BOOL	Coolant status display

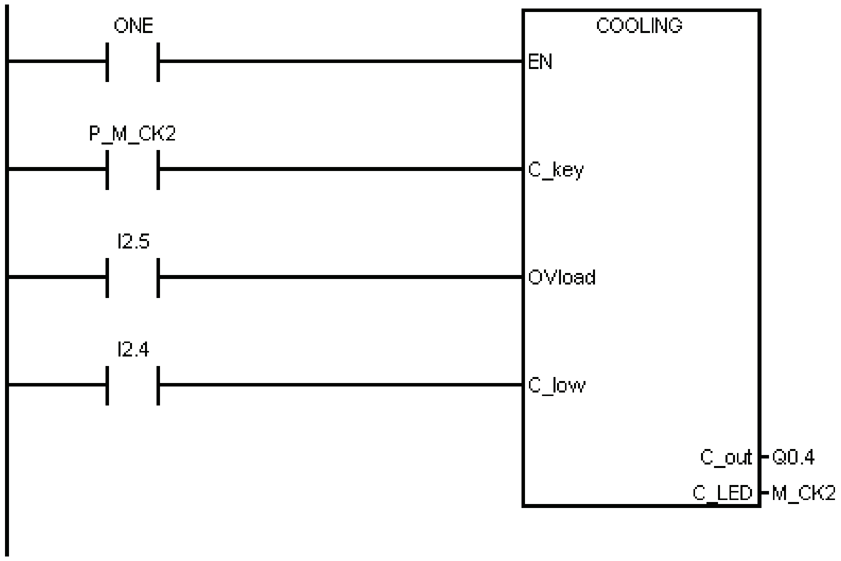
Assigned global variables

COOLon	MB150.0	Coolant on/off status

Relevant PLC machine data

None

Example for calling subroutine 44



3.18 Subroutine 45 - LUBRICAT (control of lubricate)

Purpose

Subroutine 45 is used to control the lubrication according to specific time interval and duration (independent of the distance that the axis has travelled). Meanwhile, a manual button is available to start the lubrication, and you can configure that the lubrication starts automatically each time that the machine is powered up. Normally, lubricating starts automatically and cyclically according to specified time interval **Lintv**, and operates for a specific time **Ltime** at each cycle. Lubrication stops in case of an Emergency Stop, lubrication motor overload, low lubricant level.

This subroutine can activate following alarms:

- Alarm 700020: lubrication motor overload
- Alarm 700021: low lubricant level

Local variable definition

Table 3- 12 Inputs

Name	Type	Description
Lintv	WORD	Lubricating time interval (unit: 1 min)
Ltime	WORD	Lubricating time duration of each cycle (unit: 0.01 s, max. 327.67 s)
L_key	BOOL	Manual lubricating key (triggering signal)
L1st	BOOL	Mode selection: the lubrication starts at the first PLC scan
Ovload	BOOL	Lubricating motor overload
L_low	BOOL	Low lubricate level

Table 3- 13 Outputs

Name	Type	Description
L_out	BOOL	Lubrication output
L_LED	BOOL	Indicant for lubricate output

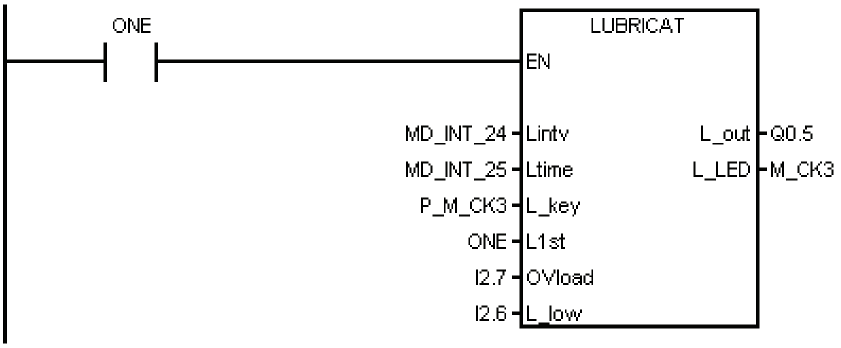
Assigned global variables

L_interval	C24	Timer for the lubricating time intervals (unit: min)
L_time	T27	Timer for very lubricating time duration (unit: 0.01s, max. 327.67 s)

Relevant PLC machine data

No.	Unit	Range	Description
14510 [24]	Min.	-	Lubricating time interval
14510 [25]	0.01 s	100 to 2,000	Lubricating time duration

Example for calling subroutine 45



3.19 Subroutine 46 - PI_SERVICE (Asynchronous Subroutine Program)

Purpose

Subroutine 46 is for realizing functions like ASUP (Asynchronous Subroutine Program), deleting a password and reading current tool number. ASUP function means the execution of PLCASUP1.SPF or PLCASUP2.SPF called by the PLC. SINUMERIK 808D provides two ASUPs for the PLC. The two ASUPs can not be simultaneously executed, and the PLCASUP1.SPF has a higher priority over the PLCASUP2.SPF.

In a program, firstly you can initialize the ASUP1 and ASUP2 by setting "PI index" (DB1200.DBB4001) and "NCK read/write start" (DB1200.DBX4000.0), and then use a rising edge to trigger "ASUP1 start" (DB3400.DBX0.0) and "ASUP2 start" (DB3400.DBX1.0).

Table 3- 14 Relevant machine data

No.	Name
10702	IGNORE_SINGLEBLOCK_MASK
11602	ASUP_START_MASK
11604	ASUP_START_PRIO_LEVEL
20116	IGNORE_INHIBIT_ASUP

Note

Important!

The SINUMERIK 808D provides two user ASUPs. In the sample application, ASUP1 is used for manual tool change and ASUP2 is used for the MANUAL MACHINE of the workpiece on a turning machine with the Manual Machine Plus function.

Local variable definition

Table 3- 15 Inputs

Name	Type	Description
ASUP1_trigger	BOOL	Calling the ASUP1, rising edge active
ASUP2_trigger	BOOL	Calling the ASUP1, rising edge active

Table 3- 16 Outputs

Name	Type	Description
ASUP1Run	BOOL	Running state of the ASUP1
ASUP2Run	BOOL	Running state of the ASUP2
Err1	BOOL	ASUP1 execution error
Err2	BOOL	ASUP2 execution error

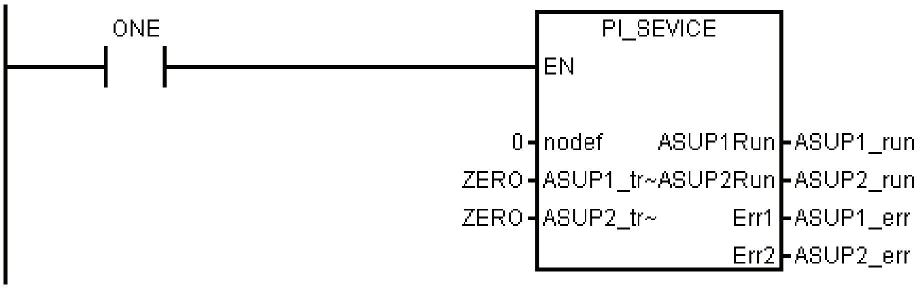
Occupied global variables

IniASUP1	M229.0	Mark of ASUP1 initialization
IniASUP2	M229.1	Mark of ASUP2 initialization

Relevant PLC machine data

None

Example for calling subroutine 46



3.20 Subroutine 47 - PLC_Select_PP (PLC selects a subroutine)

Purpose

Subroutine 47 is used to select a part program.

You firstly need to create a PLC program-calling table, and assign a program index to each part program in this table. In subroutine 47 you can assign DB1700.DBB1000 to the "Program index" to select the corresponding part program.

Local variable definition

Table 3- 17 Inputs

Name	Type	Description
PP_num	BOOL	The index for a part program

Table 3- 18 Outputs

Name	Type	Description
Finish	BOOL	Part program selection finished
Error	BOOL	Part program selection error

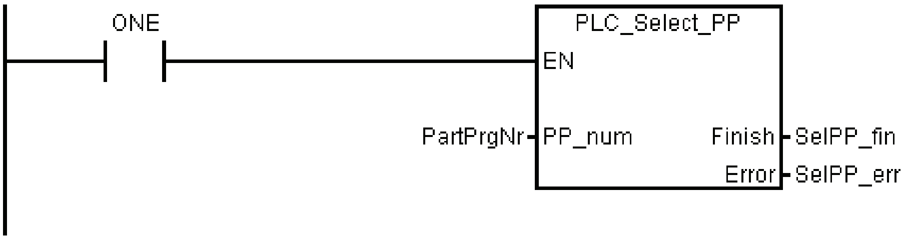
Assigned global variables

SelPP_FinOm	BOOL	M239.6	Indicates that a part program has been selected
SelPP_ErrOm	BOOL	M239.7	Indicates that an error occurs when selecting a part program

Relevant PLC machine data

None

Example for calling subroutine 47



3.21 Subroutine 48 - ServPlan (service plan)

Purpose

To use subroutine 48, you must have created a service plan on the SINUMERIK 808D. When the pre-alarm time arrives, the machine outputs a notification message. When the final alarm time arrives, the machine outputs an alarm message.

Note

To perform a service plan on the PLC, you need to download DB9903(SP_INI) and DB9904(SP_ACT).

Local variable definition

Table 3- 19 Inputs

Name	Type	Description
Deact0	BOOL	Freezing a service plan
AckMsg0	BOOL	Confirming the notification message of a service plan

Table 3- 20 Outputs

Name	Type	Description
HintMsg0	BOOL	Notification message
Alarm0	BOOL	Alarm message

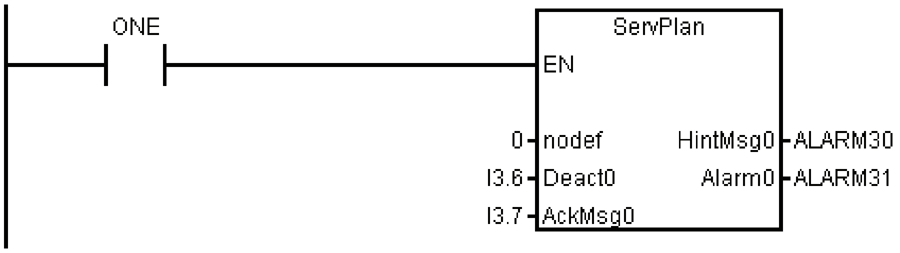
Assigned global variables

ServPlan_msg0	BOOL	Notification message of the first service plan
ServPlan_alm0	BOOL	Alarm message of the first service plan

Relevant PLC machine data

None

Example for calling subroutine 48



3.22 Subroutine 49 - GearChg1_Auto (automatic spindle gear change)

Purpose

Subroutine 49 is used to automatically change the gear for the analog spindle with 2-level gear detection signals.

During a gear change, the spindle oscillates and the PLC outputs the gear change signal. When the PLC detects that the desired gear level has been reached, the gear change has been completed.

You cannot use this subroutine together with subroutine 50.

Local variable definition

Table 3- 21 Inputs

Name	Type	Description
D_CHG	WORD	Gear change delay time (unit: 0.01 s)
D_MON	WORD	Gear change monitoring time (unit: 0.01 s)
D_S0	WORD	Spindle stop delay time (unit: 0.01 s)
T_GC	WORD	Monitor time for whole gear change process, must > D_CHG + D_MON + D_S0 (unit: 0.01 s)
S_hold	BOOL	Signal of zero spindle velocity (NO)
S_alarm	BOOL	Spindle alarm (NO)
LGi	BOOL	Low detection switch (NO)
HGi	BOOL	High detection switch (NO)

Table 3- 22 Outputs

Name	Type	Description
LGo	BOOL	Low gear output
HGo	BOOL	High gear output
LG_LED	BOOL	Status display for the low gear
HG_LED	BOOL	Status display for the high gear

Assigned global variables

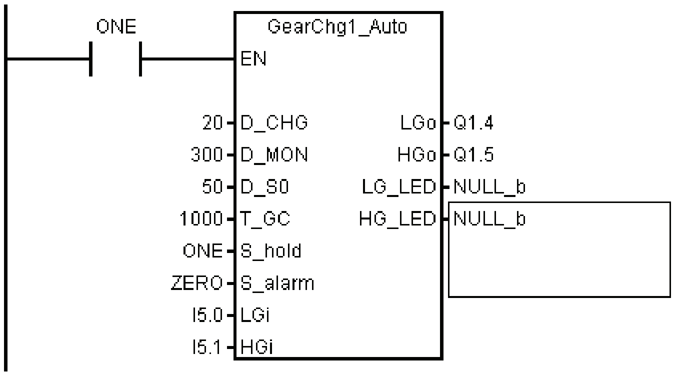
HGom	BOOL	M248.0	Signal indication of high gear stage output
LGom	BOOL	M248.1	Signal indication of low gear stage output
HGcmd	BOOL	M248.2	High gear level command
LGcmd	BOOL	M248.3	Low gear level command
SPhold	BOOL	M248.4	Spindle stops and ready for oscillation
Dstill	BOOL	M248.5	Signal for spindle stop
Dchg	BOOL	M248.6	Spindle gear change delay
Dmon	BOOL	M248.7	Monitoring for the gear change

Req_SP_G_ CHG	BOOL	M244.0	Request for spindle gear change
Req_Low_G	BOOL	M244.1	Request to change to the low gear stage
Req_Hign_G	BOOL	M244.2	Request to change to the high gear stage
D_S0	TIMER	T13	Spindle stop delay
Td_GearChg	TIMER	T24	Gear change delay
Tm_GearChg	TIMER	T25	Delay for monitoring the gear change

Relevant PLC machine data

None

Example for calling subroutine 49



3.23 Subroutine 50 - GearChg2_Virtual (virtual spindle gear change)

Purpose

Using subroutine 50, you can requests the system to switch to the corresponding gear after changing the gear manually. The corresponding gear is set when M41-M45 are executed.

This subroutine must not be used together with GearChg1_Auto (SBR 49).

Local variable definition

Inputs

None

Table 3- 23 Outputs

Name	Type	Description
HL_gear	BOOL	Output of the high/low gear 0: gear-level low gear; 1: gear-level high gear)

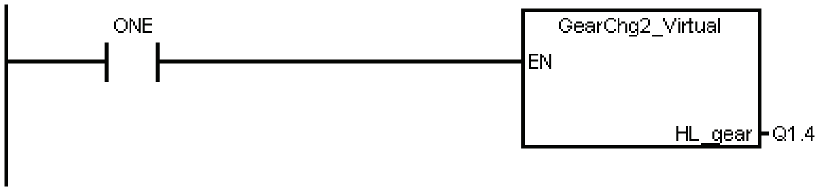
Assigned global variables

None

Relevant PLC machine data

None

Example for calling subroutine 50



3.24 Subroutine 51 - Turret1_HED_T (turret with Hall effect device position sensor)

Purpose

Subroutine 51 is used to control the turret with a Hall effect device positioning sensor, and the turret motor is controlled by the PLC.

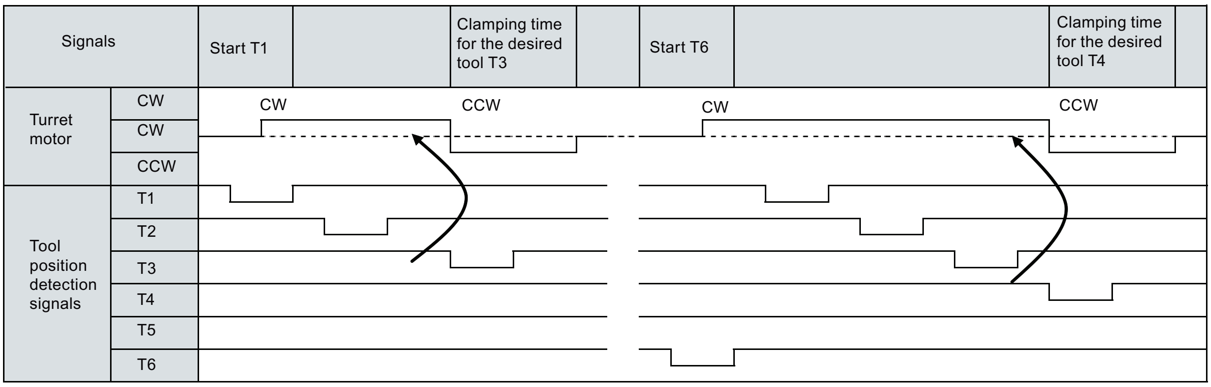
The turret rotates clockwise to search for a tool, and rotates counter-clockwise after positioning the desired tool to clamp it (the turret CCW rotation time can be adjusted). An alarm occurs if the turret fails to position the desired tool after the duration expires. The subroutine verifies the time that the turret rotates CCW, and sets a limit of maximum 3 seconds for this rotation time to prevent the turret motor from being broken.

In the AUTO and MDA modes, the T function starts a tool change operation. In the JOG mode, a short strike on the MCP key changes a turret position.

During a tool change, the NC interface signals "Read-in disable" (DB3200.DBX6.1) and "Feedhold" (DB3200.DBX6.0) are set; this means that the part program can only continue to run after the tool change.

The turret positioning is prohibited in the case of an Emergency Stop, turret motor overload or program test/simulation.

The timing diagram for positioning a tool in the turret using the Hall effect device positioning sensor is shown as follows:


Signals
Start T1
Clamping time for the desired tool T3
Start T6
Clamping time for the desired tool T4
Turret motor
CW
CW
CCW
Tool position detection signals
T1
T2
T3
T4
T5
T6
CW
CCW
CW
CCW

This subroutine can activate the following alarms:

- Alarm 700022: Turret motor overload
- Alarm 700023: Programmed tool number higher than the max. tool number of the turret
- Alarm 700024: Wrong setting of the max. tool number for the turret
- Alarm 700025: No turret positioning signals available
- Alarm 700026: Tool positioning time out

Local variable definition

Table 3- 24 Inputs

Name	Type	Description
Tmax	WORD	Max. tool number of the turret
C_time	WORD	CCW clamping time (unit: 0.1 s)
M_time	WORD	Monitoring time for the tool change
T_polar	BOOL	Polar selection for the tool change 0: tool position low active 1: tool position high active
T_key	BOOL	Manual tool change key (triggering signal)
T_01 to T_06	BOOL	Tool position sensor (low active)
OVload	BOOL	Turret motor overload (NC)

Table 3- 25 Outputs

Name	Type	Description
T_cw	BOOL	Turret positioning
T_ccw	BOOL	Turret clamping
T_LED	BOOL	Status display during the tool change
ERR1	BOOL	No turret positioning signals available
ERR2	BOOL	Programmed tool out of turret range
ERR3	BOOL	Tool positioning time out
ERR4	BOOL	Turret motor overload
ERR5	BOOL	Wrong setting of the max. tool for the turret
ERR6	BOOL	Reserved

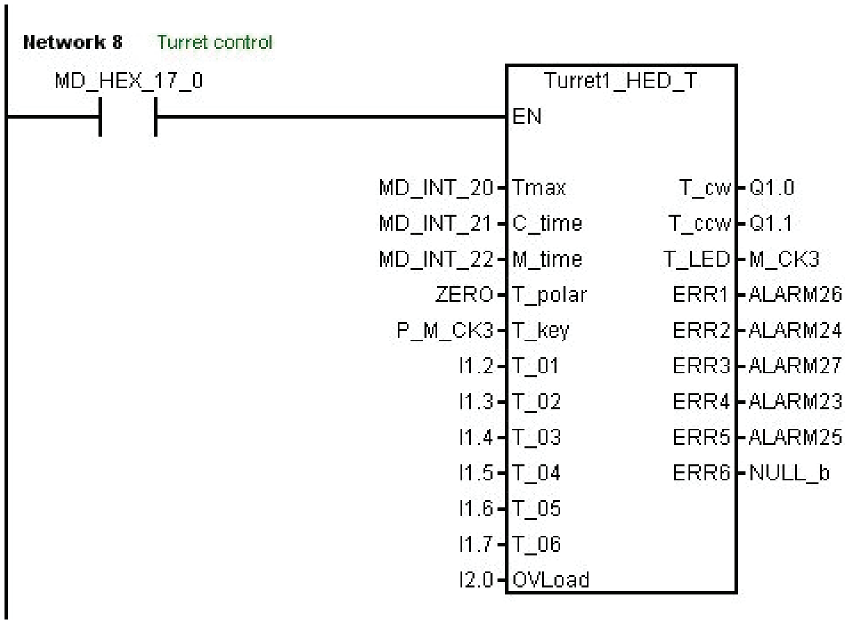
Assigned global variables

T_cw_m	M156.0	Position marking for turret CW rotation
T_ccw_m	M156.1	Position marking for turret CCW rotation
CcwDelay	M156.2	Turret CCW rotation delay
K_active	M156.3	Manual key active
Tpos_C	M156.4	Turret position changed
Tp_eq_Tc	M156.5	Programmed tool number equal to the current tool number
Tp_eq_0	M156.6	Programmed tool number equal to zero
T_P_INDX	MD160	Monitoring the tool change buffer zone in the JOG mode
T_CHL	M168.4	Operating mode locked
Tm1_FindT	T15	Monitoring timer for tool searching
T_CLAMP	T13	Clamping timer for turret 1

Relevant PLC machine data

No.	Unit	Description
14510 [20]	-	Max. tool number (4 or 6)
14510 [21]	0.1 s	Turret clamping time
14510 [22]	0.1 s	Monitoring time for tool searching
14512[17].0	-	Activating the turret function of a turning machine

Example for calling subroutine 51



3.25 Subroutine 52 - TURRET2_BIN_T (turret with binary coding function)

Purpose

Subroutine 52 is used to control the turret with encoder positioning signals and function of dual-direction adjacent tool change. Contact the turret vendor for the working theory and the timing diagram of a tool change.

During a tool change, the NC interface signals "Read-in disable" (DB3200.DBX6.1) and "Feedhold" (DB3200.DBX6.0) are set, so the part program can continue running only after the tool change action.

The turret position action is forbidden in case of an emergency stop, turret motor overload or program test/simulation.

This subroutine can activate the following alarms:

- Alarm 700022: Turret motor overload
- Alarm 700023: Programmed tool number higher than the max. tool number of the turret
- Alarm 700024: Wrong setting of the max. tool number for the turret
- Alarm 700026: Not able to find expected tool in monitor time
- Alarm 700011: Not able to lock tool in expected time

Local variable definition

Table 3- 26 Inputs

Name	Type	Description
Tmax	WORD	Max. tool number of the turret
Tm_Lck	WORD	Turret clamping time (unit: 0.1 s)
Tm_Chg	WORD	Tool change monitoring
T_1	BOOL	Tool code A x 1
T_2	BOOL	Tool code B x 2
T_3	BOOL	Tool code C x 4
T_4	BOOL	Tool code D x 8
Parity	BOOL	Position parity
Strobe	BOOL	Position strobe
OVload	BOOL	Turret motor overload (NC)
P_Indx	BOOL	Turret pre-indexing sensor
T_key	BOOL	Manual tool change key (triggering signal)

Table 3- 27 Outputs

Name	Type	Description
T_cw	BOOL	Turret CW rotation output
T_ccw	BOOL	Turret CCW rotation output
Magent	BOOL	Turret clamping output
T_LED	BOOL	Status display during the tool change
ERR1	BOOL	Turret motor overload
ERR2	BOOL	Programmed tool out of turret range
ERR3	BOOL	Wrong setting of the max. tool for the turret
ERR4	BOOL	Not able to find pre-index signal in expected time
ERR5	BOOL	Not able to lock in expected time

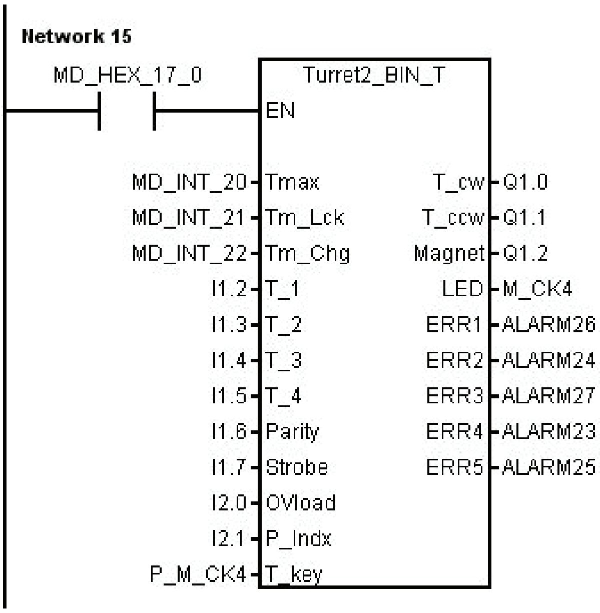
Assigned global variables

T_CURRENT	VD14000064	Current tool (retentive data)
T_cw_m	M156.0	Position marking for turret CW rotation
T_ccw_m	M156.1	Position marking for turret CCW rotation
T_P_INDX	MD160	Monitoring the tool change buffer zone in the JOG mode
T_DES	M164	Desired tool number
T_DIR	M168.0	Direction of adjacent tool change
T_POS	M168.1	Turret tool positioning finished
T_LOCK	M168.2	Turret clamping command
T_MAG	M168.3	Turret magnetic clamping

Relevant PLC machine data

No.	Unit	Description
14510 [20]	-	Max. tool number (4 or 6)
14510 [21]	0.1 s	Turret clamping time
14510 [22]	0.1 s	Monitoring time for tool searching
14512[17].0	-	Activating the turret function of a turning machine

Example for calling subroutine 52



3.26 Subroutine 53 - Turret3_CODE_T (tool change control for turret with coding function)

Purpose

Subroutine 53 is used to control the turret with coded tool positions and function of adjacent tool change. The difference between the subroutine 52 and the subroutine 53 is that the subroutine 52 uses binary tool position codes while the subroutine 53 uses tool position codes made according to a specific common turret.

During a tool change, the NC interface signal "Feedhold" (DB3200.DBX6.0) is set; this means that the part program can only continue to run only after the tool change.

The turret positioning is prohibited in the case of an Emergency Stop, turret motor overload or program test/simulation.

Local variable definition

Table 3- 28 Inputs

Name	Type	Description
M_time	WORD	Monitoring time for the tool change
T_key	BOOL	Manual tool change key (NO)
A to D	BOOL	Tool position code signals
Strobe	BOOL	Position strobe
Lock_i	BOOL	Position clamping
OVload	BOOL	Turret motor overload (NC)

Tool position	Tool position code A	Tool position code B	Tool position code C	Tool position code D
1	0	1	0	0
2	0	0	0	1
3	1	0	0	0
4	0	0	1	0
5	1	1	1	0
6	1	0	1	1
7	1	1	0	1
8	0	1	1	1

Table 3- 29 Outputs

Name	Type	Description
T_UNCLAMP	BOOL	Turret release
T_CLAMP	BOOL	Turret clamping
T_CW	BOOL	Turret CW rotation
T_CCW	BOOL	Turret CCW rotation
T_LED	BOOL	Status display during the tool change

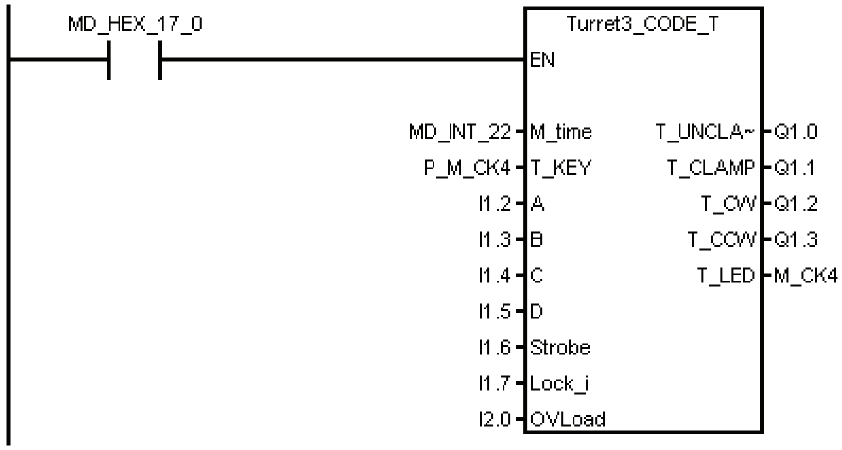
Assigned global variables

Tpos_C	BOOL	M156.4	Tool searching finished
T_cwm	BOOL	M235.6	Mark for turret CW rotation
T_ccwm	BOOL	M235.7	Mark for turret CCW rotation
TK_act	BOOL	M236.4	Mark for manual tool change
Tc_ne_0	BOOL	M237.0	Current tool number is not 0
T_dir	BOOL	M237.1	Direction for searching for an adjacent tool

Relevant PLC machine data

No.	Unit	Description
14510 [22]	0.1 s	Monitoring time for searching for a tool

Example for calling subroutine 53



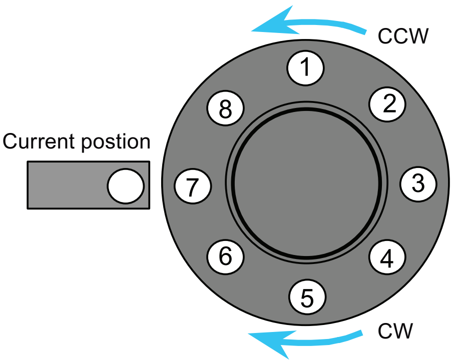
3.27 Subroutine 54 - Turret2_3_ToolDir (tool change direction)

Purpose

Subroutine 54 is used to find out the direction of searching for an adjacent tool and the pre-indexing position (this is, the previous position of the desired tool in the direction of an adjacent tool). To find out the direction, you need to know the max. tool number of the turret and the programmed tool number.

You can use this subroutine to control the turret to search for an adjacent tool on a turning machine or a machine centre. The turret tool position ranges from 2 to 64.

For example:



Tool position number	Current position	Programmed tool number	Pre-indexing position	Direction
1	7	2	1	CCW
2	7	5	6	CW
3	3	8	1	CW
4	1	4	3	CCW
5	6	8	7	CCW

Local variable definition

Table 3- 30 Inputs

Name	Type	Description
Tmax	DWORD	The max. turret tool position number
Pnum	DWORD	Programmed tool number
Tcurr	DWORD	Current position of the turret

Table 3- 31 Outputs

Name	Type	Description
P_INDXo	DWORD	Pre-indexing position: the previous tool position of the desired tool in the direction of an adjacent tool.
DIR	BOOL	Tool change direction: 1: CW; 0: CCW

Assigned global variables

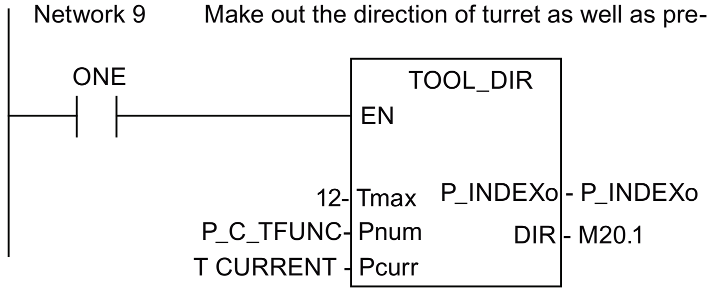
None

Relevant PLC machine data

None

Example for calling subroutine 54

This subroutine is called by subroutine 52 and subroutine 53.



3.28 Subroutine 55 - Tail_stock_T (Tailstcok control program for turning machines)

Purpose

Subroutine 55 is used to control forward or backward movement of the tail stock on a turning machine.

In the JOG mode, press the "Tailstock" key to move the tailstock forward or backward. Pressing "Tailstock" moves the tailstock forward, and one more pressing moves the tailstock backward.

In the AUTO mode, you can use M20 or M21 to control the forward or backward movement of the tailstock.

Local variable definition

Table 3- 32 Inputs

Name	Type	Description
TailCtrl_K	BOOL	Tailstock
SP_status	BOOL	Spindle status

Table 3- 33 Outputs

Name	Type	Description
TailAdv_O	BOOL	Output to move the tailstock forward
TailRet_O	BOOL	Output to move the tailstock backward

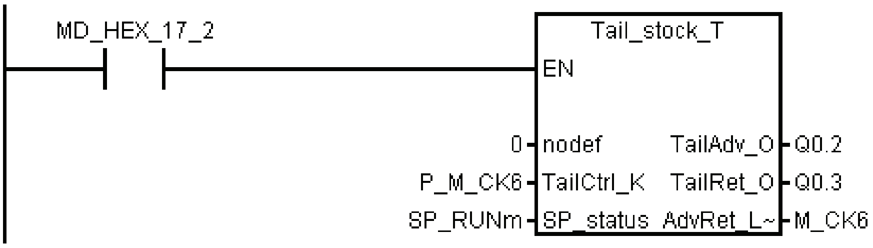
Assigned global variables

SP_RUNm	BOOL	M236.0	Indicate that the spindle is running
TailAdv_m	BOOL	M229.2	Indicates that the tailstock is moving forward
TailRet_m	BOOL	M229.3	Indicates tailstock is moving backward

Relevant PLC machine data

None

Example for calling subroutine 55



3.29 Subroutine 56 - Lock_unlock_T (clamping control for turning machines)

Purpose

Subroutine 56 is used to control the clamping or release for the chuck for a turning machine.

In the JOG mode, press the "External/Inside clamping" key to select either external clamping or inside clamping, and press "Clamp" or "Unclamp" key to clamp or release the chuck. Furthermore, you can also use the "Foot switch" to clamp or release the chuck. Pressing the "Foot switch" for once release the chuck, and one more pressing clamps the chuck.

In the AUTO mode, you can execute M10/M11 to control the clamping or release of the chuck.

Note

The chuck status should be kept when clamping outputs are zero.

Local variable definition

Table 3- 34 Inputs

Name	Type	Description
Delay	WORD	Clamping delay time
LckRel_k	BOOL	Lock / release toggle signal
ExtIn_k	BOOL	External/inside clamping key
S_velo	BOOL	Spindle velocity signals 0: spindle velocity is 0 1: spindle is running
Foot_switch	BOOL	Foot switch signal

Table 3- 35 Outputs

Name	Type	Description
Lck1_O	BOOL	Clamping output 1
Lck2_O	BOOL	Clamping output 2
Lck_LED	BOOL	Clamping state
ExtIn_LED	BOOL	External/inside clamping state: 0: external clamping 1: inside clamping
Err1	BOOL	No chuck operation during the running of the spindle

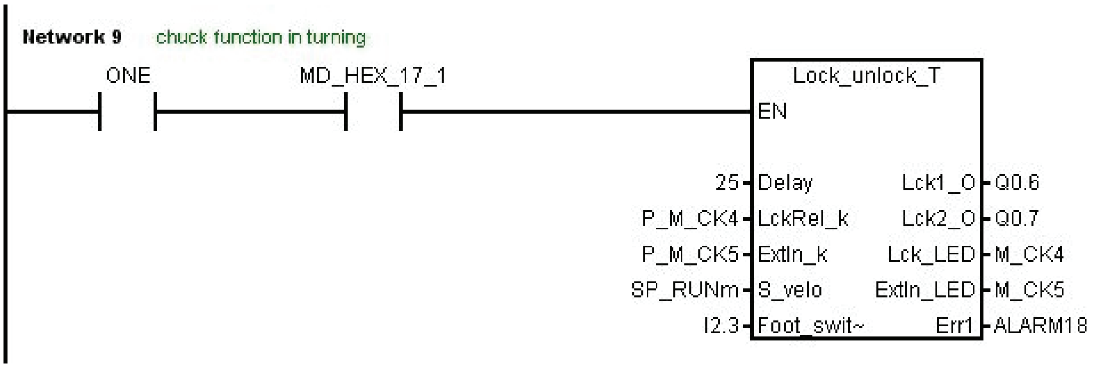
Assigned global variables

ChuckLcked	BOOL	M229.4	Chuck clamped
ChuckLckLED	BOOL	M239.2	Chuck at released state
ExtInLED_Om	BOOL	M239.5	External/inside clamping state
TR_Status	BOOL	M237.6	Chuck release command

Relevant PLC machine data

None

Example for calling subroutine 56



3.30 Subroutine 58 (MM_MAIN)

Purpose

To use subroutine 58, you must have licensed the optional manual machine plus function for the SINUMERIK 808D turning machine. The subroutines 46, 58 and 59 must be used together. This subroutine is used to control the manual machine function after the manual machine interface is activated.

Local variable definition

Table 3- 36 Inputs

Name	Type	Description
TK_X_P	BOOL	Forward on axis X
TK_X_M	BOOL	Backward on axis X
TK_Z_P	BOOL	Forward on axis Z
TK_Z_M	BOOL	Backward on axis Z
RAPID	BOOL	Rapid feed
SP_CW	BOOL	Clockwise rotation of the spindle
SP_CCW	BOOL	Counter-clockwise rotation of the spindle
SP_STOP	BOOL	Spindle stop
NC_START	BOOL	NC start
NC_STOP	BOOL	NC stop
AUTO_ENABLE	BOOL	AUTO mode allowed
MDA_ENABLE	BOOL	MDA mode allowed
ROV	BOOL	Rapid rate

Table 3- 37 Outputs

Name	Type	Description
AL_03	BOOL	Not approaching the reference point on axis X
AL_04	BOOL	Not approaching the reference point on axis Z
AL_09	BOOL	Incorrect start in the spindle direction
AL_11	BOOL	JOG program timeout
AL_12	BOOL	Spindle rate not 100%
AL_13	BOOL	Spindle not being started
AL_14	BOOL	Feed rate 0%
AL_16	BOOL	Spindle direction change in a thread not allowed

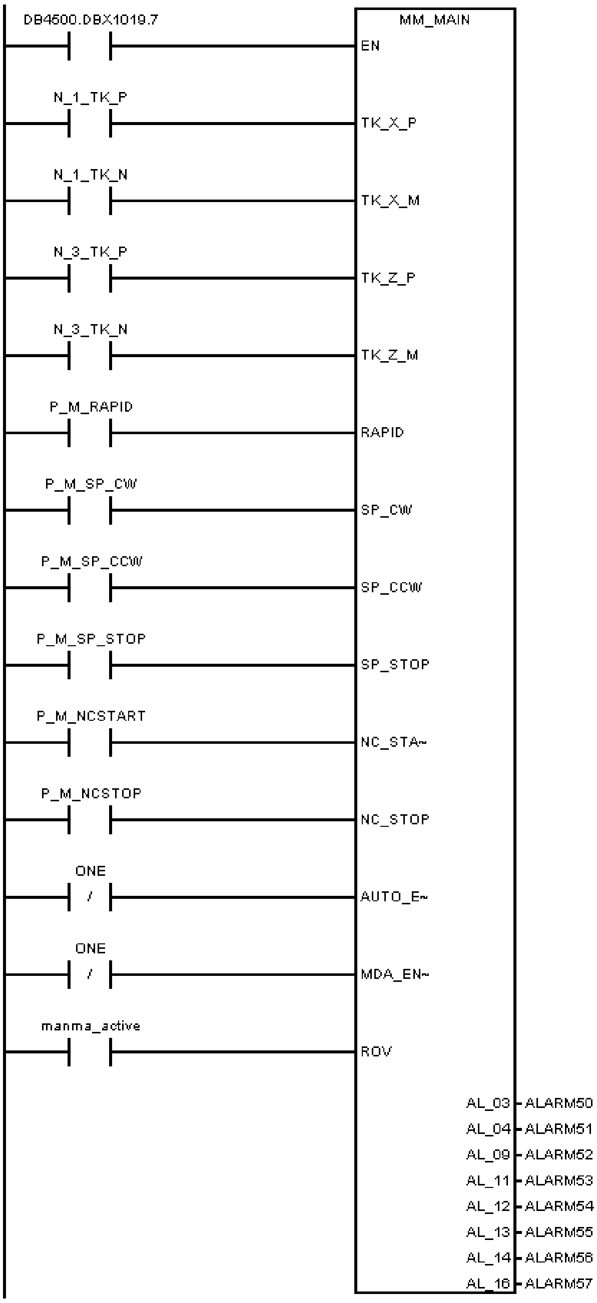
Assigned global variables

Byte	Signal	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
MB170	HMI<->MM						Request for MM HMI startup	MM HMI enabled	MM HMI started
MB171	HMI<->MM								
MB172	HMI<->MM								
MB173	HMI<->MM								
MB174	HMI<->MM	Cone angle 270°-360°	Cone angle 270°	Cone angle 180°-270°	Cone angle 180°	Cone angle 90°-180°	Cone angle 90°	Cone angle 0°-90°	Cone angle 0°
MB175	HMI<->MM						Direction key enabled		Spindle rotated
MB176	HMI<->MM		Working step enabled	Groove enabled	Thread chaining enabled	Drilling enabled	Arc enabled	Cutting enabled	Thread enabled
MB177	HMI<->MM								
MB178									
MB179									
MB180							Recutting canceled	Recutting performed	Recut the thread or not?

Relevant PLC machine data

No.	Unit	Range	Description
MD14512[19].7	-	-	1: to enable the manual machine function 0: to disable the manual machine function

Example for calling subroutine 58



3.31 Subroutine 59 (MM_MCP_808D)

Purpose

To use subroutine 59, you must have licensed the optional manual machine plus function for the SINUMERIK 808D turning machine. The subroutines 46, 58 and 59 must be used together. Normally, the spindle will be stopped after you press the NC reset key. However, when a manual machine is started, you do not want to stop the spindle after pressing the NC reset key. In this case, call subroutine 59 (MM_MCP_808D) after executing subroutine 37 (MCP_NCK). Then you do not need to rewrite subroutine 37 (MCP_NCK).

Local variable definition

Table 3- 38 Inputs

Name	Type	Description
SP_STOP_K	BOOL	Spindle stop

Outputs

None

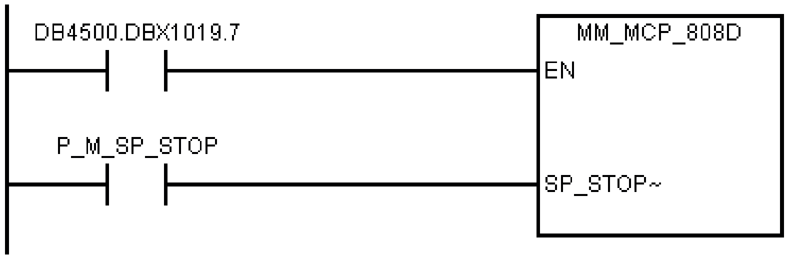
Assigned global variables

None

Relevant PLC machine data

No.	Unit	Range	Description
MD14512[19].7	-	-	1: to enable the manual machine function 0: to disable the manual machine function

Example for calling subroutine 59



3.32 Subroutine 60 - Disk_MGZ_M (disk-style tool magazine for milling)

Purpose

You can use subroutine 60 to control the disk-style tool magazine on a milling machine.

In the reference point mode, initialize the tool magazine by pressing the "Original position of the tool magazine" key.

In the manual mode, you can rotate the tool magazine clockwise or counter-clockwise, and enable the tool magazine to reach the spindle or tool change position respectively through the "Clockwise rotation of the magazine", "Counter-clockwise rotation of the magazine", "Tool magazine reaching the spindle", and "Tool magazine reaching the tool change position" keys.

In the auto mode, you need to execute M06 to call the tool change subroutine when compiling a part program. Subroutine 60 and the tool change subroutine must be used together during the tool change process. Three operations are involved in the tool change control, that is, tool return, tool retrieval, and tool change.

1. The tool return operation is to return the tool on the spindle back to the tool magazine disk when compiling T0 and a tool is located on the spindle.
2. The tool retrieval operation is to get the desired tool from the tool magazine disk and install it on the spindle when compiling Tx (x ≠ 0) and no tool is on the spindle.
3. The tool change operation is to first return the tool on the spindle back to the tool magazine disk and then get the desired tool from the tool magazine disk when compiling Tx (x ≠ 0; x ≠ number of the tool on the spindle).

For details, please refer to the tool change subroutine.

The following machine data is involved in this subroutine:

MD10715: M_NO_FCT_CYCLE[0]

MD10716: M_NO_FCT_CYCLE_NAME[0]

MD22550: TOOL_CHANGE_MODE

MD22560: TOOL_CHANGE_M_CODE

Local variable definition

Table 3- 39 Inputs

Name	Type	Description
MgzCnt	BOOL	Tool magazine count
MgzRef_k	BOOL	Tool magazine reset, with the current tool number set to 1
MgzCW_k	BOOL	Tool magazine forward
MgzCCW_k	BOOL	Tool magazine backward
MgzSp_k	BOOL	Spindle position key for the tool magazine
MgzOrg_k	BOOL	Original position key for the tool magazine

Name	Type	Description
MgzSp_pos	BOOL	Tool magazine has reached the spindle position
MgzOrg_pos	BOOL	Tool magazine has reached the original position
T_rel_pos	BOOL	Release position fort he tool magazine
T_lck_pos	BOOL	Clamping position for the tool magazine
T_rel_k	BOOL	Tool release key for the spindle
T_rel_EnK	BOOL	Enabling key for the tool release of the spindle

Table 3- 40 Outputs

Name	Type	Description
MgzCW_o	BOOL	Clockwise rotation of the tool magazine
MgzCCW_o	BOOL	Counter-clockwise rotation of the tool magazine
MgzSp_o	BOOL	The spindle position for the tool magazine
MgzOrg_o	BOOL	The original position for the tool magazine
SpRelT_o	BOOL	Releasing a tool
RelT_En_o	BOOL	Enabling releasing a tool
MgzSp_LED	BOOL	Magazine reaches spindle position
MgzOrg_LED	BOOL	Magazine reaches original position

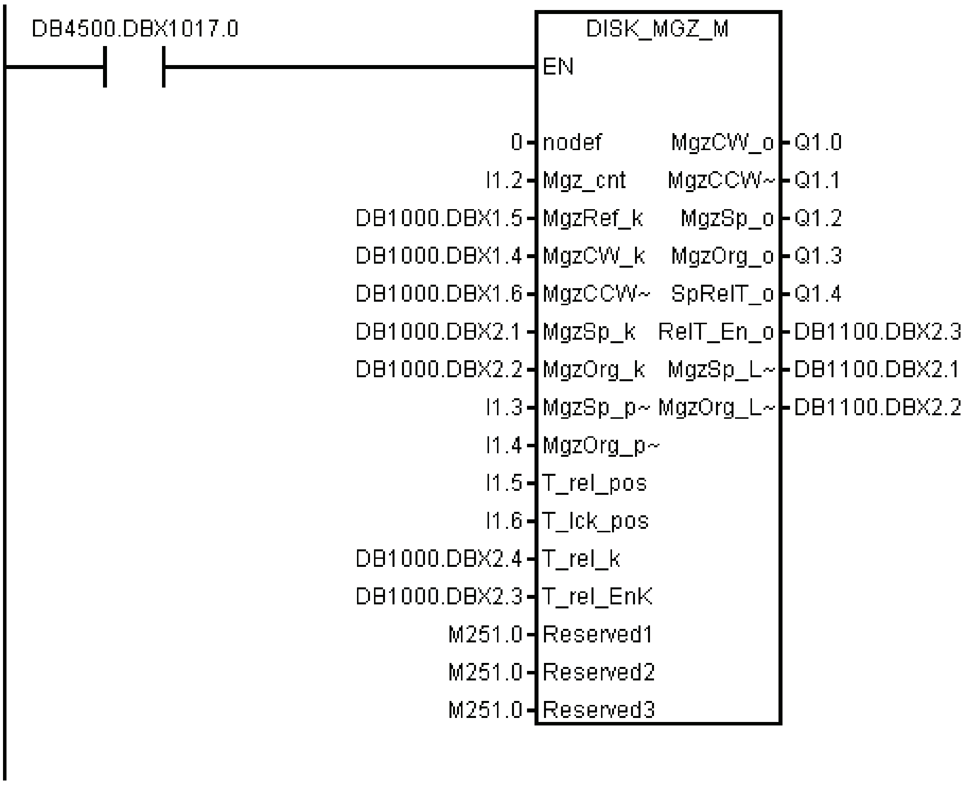
Assigned global variables

MgzCW_cmd	BOOL	M230.0	Command for clockwise rotation of the tool magazine
MgzCCW_cmd	BOOL	M230.1	Command for counter-clockwise rotation of the tool magazine
Mgz_rot_CMD	BOOL	DB4900.DBB24	Tool change command from the tool change subroutine

Relevant PLC machine data

None

Example for calling subroutine 60



3.33 Subroutines 34 to 36, 57, 61 and 62

Explanation

Subroutines 34 to 36, 57, 61 and 62 are reserved for users.

3.34 Subroutine 63 - TOGGLES

Purpose

Two types of switches are provided in subroutine 63, more specifically, a hold switch for switching a circuit on (press) and off (press again), and a delay switch for switching on a circuit and automatically switching it off after a certain time period. A total of six hold switches and two delay switches are available in this subroutine, with the delay duration being configurable. The key inputs or outputs of the subroutine can be connected with any physical inputs or outputs. The inputs and outputs of all idle switches are respectively "ZERO" and "NULL_b" (M255.7).

Local variable definition

Table 3- 41 Inputs

Name	Type	Description
Delay7	WORD	Delay duration of switch 7 (unit: 10 ms)
Delay8	WORD	Delay duration of switch 8 (unit: 10 ms)
Ki_1…Ki_6	BOOL	Input of hold switch 1...input of hold switch 6
Ki_7…Ki_8	BOOL	Inputs of delay switches 7 and 8

Table 3- 42 Outputs

Name	Type	Description
Ko_1…Ko_8	BOOL	Output of switch 1...output of switch 8

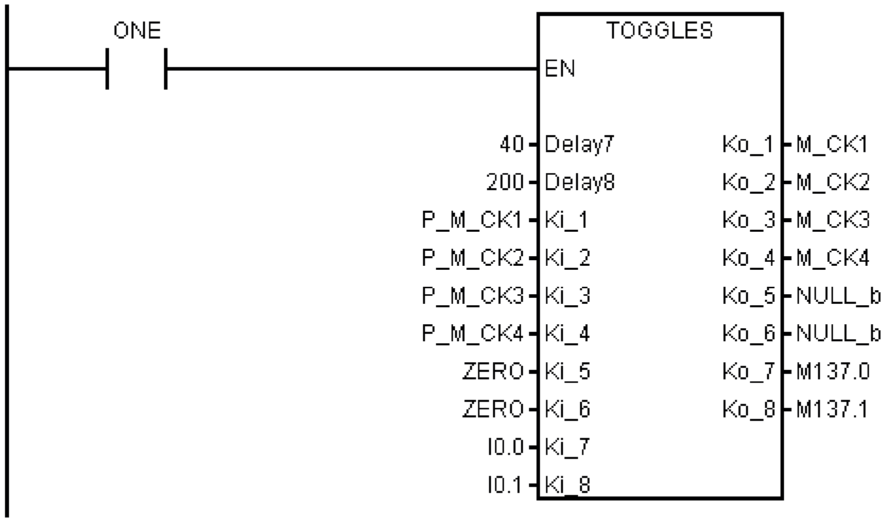
Assigned global variables

K1st1 … K8st1	MB245	State 1 of the hold switch
K1st2 … K8st2	MB246	State 2 of the hold switch
K1on … K8on	MB247	"On" state of the hold switch

Relevant PLC machine data

None

Example for calling subroutine 63



Use of user alarms in the PLC subroutines 4

Some user alarms are activated in a subroutine. In the case that such an alarm is generated, you can search the following list for the subroutine wherein the alarm is activated.

Alarm No.	Interface Address	Alarm Description	From SBR
700010	DB1600.DBX1.2	HHU is active	SBR41: MINI_HHU
700011	DB1600.DBX1.3	Not able to lock tool in expected time	
700012	DB1600.DBX1.4	Spindle in braking progress	SBR42: SPINDLE
700013	DB1600.DBX1.5	Operation while chuck is not locked	SBR56: Lock_unlock_T
700014	DB1600.DBX1.6	Gear-change time out	SBR49: GearChg1_Auto
700015	DB1600.DBX1.7	Gear level position error	
700016	DB1600.DBX2.0	Drives not ready	SBR33: EMG_STOP
700017	DB1600.DBX2.1	Operate chuck when spindle or part program is running	SBR56: Lock_unlock_T
700018	DB1600.DBX2.2	Cooling motor overload	SBR44: COOLING
700019	DB1600.DBX2.3	Coolant liquid position in low level	
700020	DB1600.DBX2.4	Lubrication motor overload	SBR45: LUBRICAT
700021	DB1600.DBX2.5	Lubricant liquid position in low level	
700022	DB1600.DBX2.6	Turret motor overload	SBR51: Turret1_HED_T SBR52: Turret2_BIN_T SBR53: Turret3_CODE_T
700023	DB1600.DBX2.7	Programmed tool number > max. turret on turret number	
700024	DB1600.DBX3.0	Max. tool number setting error	
700025	DB1600.DBX3.1	No position signals from turret	
700026	DB1600.DBX3.2	Not able to find expected tool in monitor time	
700027	DB1600.DBX3.3	Approach reference point again after rotation monitoring	SBR40: AXIS_CTL
700028	DB1600.DBX3.4	Tool is not locked	SBR53: Turret3_CODE_T
700029	DB1600.DBX3.5	Reminding information for 1st service plan	SBR48: ServPlan
700030	DB1600.DBX3.6	Alarm for 1st service plan	
700031	DB1600.DBX3.7	Magazine not in spindle position or original position	SBR60: Disk_MGZ_M
700032	DB1600.DBX4.0	Magazine in spindle position and original position	SBR60: Disk_MGZ_M
700033	DB1600.DBX4.1	Magazine turn key when magazine or spindle not ready	SBR60: Disk_MGZ_M
700034	DB1600.DBX4.2	Block search, tool in spindle <> programmed tool	SBR60: Disk_MGZ_M

Alarm No.	Interface Address	Alarm Description	From SBR
700035	DB1600.DBX4.3	Spindle not reach tool-release pos. in time	SBR60: Disk_MGZ_M
700036	DB1600.DBX4.4	Spindle not reach tool-lock pos. in time	SBR60: Disk_MGZ_M
700049	DB1600.DBX6.1	Reference point X-axis not reached	SBR58: MM_MAIN
700050	DB1600.DBX6.2	Reference point Z-axis not reached	SBR58: MM_MAIN
700051	DB1600.DBX6.3	Wrong spindle direction started	SBR58: MM_MAIN
700052	DB1600.DBX6.4	Watchdog timer JOG-program	SBR58: MM_MAIN
700053	DB1600.DBX6.5	Spindle override not 100%	SBR58: MM_MAIN
700054	DB1600.DBX6.6	Spindle is not started	SBR58: MM_MAIN
700055	DB1600.DBX6.7	Feed override = 0%	SBR58: MM_MAIN
700056	DB1600.DBX7.0	Change of spindle direction not possible in thread	SBR58: MM_MAIN
700059	DB1600.DBX7.3	Safe door not closed, NC start not possible	SBR22: AUX_SAFE_DOOR
700060	DB1600.DBX7.4	Channel not in reset, change PRT not possible	SBR37: MCP_NCK

PLC sample applications 5

5.1 PLC sample application (turning)

This sample application is applicable to machines with the following configurations:

- Two axes: axes X and Z, with a hardware limit switch respectively in the positive and negative directions of each axis
- An analog spindle: SP
- HALL effect device turret with six-working stations
- PLC-controlled timely and quantitatively lubrication system
- PLC-controlled cooling system

Table 5- 1 Assignment of inputs and outputs

Signal	Description	Remark
I0.0	Emergency Stop button	Normally closed
I0.1	Limit switch in the "+" direction of axis X	Normally closed
I0.2	Limit switch in the "-" direction of axis X	Normally closed
I0.3		
I0.4		
I0.5	Limit switch in the "+" direction of axis Z	Normally closed
I0.6	Limit switch in the "-" direction of axis Z	Normally closed
I0.7	Reference point switch of axis X	Normally open
I1.0		
I1.1	Reference switch of axis Z	Normally open
I1.2	Tool path detecting signal T1	Valid at a low level
I1.3	Tool path detecting signal T2	Valid at a low level
I1.4	Tool path detecting signal T3	Valid at a low level
I1.5	Tool path detecting signal T4	Valid at a low level
I1.6	Tool path detecting signal T5	Valid at a low level
I1.7	Tool path detecting signal T6	Valid at a low level
I2.0	Turret motor overload	Normally closed
I2.1	Reserved for other types of turrets	Reserved
I2.2		
I2.3	Chuck foot switch	Normally open
I2.4	Coolant level too low	Normally closed
I2.5	Cooling pump motor overload	Normally closed
I2.6	Lubricant level to low	Normally closed
I2.7	Lubrication pump motor overload	Normally closed
I3.0		Reserved

Signal	Description	Remark
I3.1		Reserved
I3.2		Reserved
I3.3		Reserved
I3.4		Reserved
I3.5		Reserved
I3.6		Reserved
I3.7		Reserved
I4.0	Handheld unit: axis X selected	Valid at a high level
I4.1	Handheld unit: axis Y selected	Valid at a high level
I4.2	Handheld unit: axis Z selected	Valid at a high level
I4.3	Handheld unit: fourth axis selected	Reserved
I4.4	Handheld unit: increment X1	Valid at a high level
I4.5	Handheld unit: increment X10	Valid at a high level
I4.6	Handheld unit: increment X100	Valid at a high level
I4.7	Handheld unit: enabled	Valid at a high level
Q0.0	Working lamp	
Q0.1		
Q0.2	Tailstock forward	
Q0.3	Tailstock backward	
Q0.4	Cooling pump	
Q0.5	Lubrication pump	
Q0.6	Chuck output 1	
Q0.7	Chuck output 2	
Q1.0	Turret motor rotating clockwise	
Q1.1	Turret motor rotating counter-clockwise	
Q1.2	Reserved for other types of turrets	
Q1.3	Reserved for other types of turrets	
Q1.4	Gear shift: low gear level (SBR49: GearChg1_Auto) / Gear level status (SBR50: GearChg2_Virtual)	
Q1.5	Gear shift: high gear level (SBR49: GearChg1_Auto)	
Q1.6		
Q1.7	Handheld unit valid	

Table 5- 2 Definition of user-defined keys on the MCP

User-defined key 1	Working lamp
User-defined key 2	Manual cooling
User-defined key 3	Manual tool change
User-defined key 4	Manual chuck clamping and unclamping
User-defined key 5	Chuck clamping internally/externally
User-defined key 6	Tailstock

Structure of the sample application (OB1)

Call Conditions	Subroutine Name	Description
Each scan (SM0.0)	AUX_MCP (SBR20)	Auxiliary function
First scan (SM0.1)	PLC_INI (SBR32)	PLC initialization
Each scan (SM0.0)	EMG_STOP (SBR33)	Emergency Stop control
Each scan (SM0.0)	MCP_NCK (SBR37)	Transferring MCP and HMI signals to the NCK interface
Each scan (SM0.0)	HANDWHL (SBR39)	Selecting a hand wheel through the interface signal DB1900.DBB1xxx
Each scan (SM0.0)	AXIS_CTL (SBR40)	Coordinate enabling control, hardware limit, etc.
Each scan (SM0.0)	SPINDLE (SBR42)	Spindle control
Each scan (SM0.0)	COOLING (SBR44)	Cooling control
Each scan (SM0.0)	TURRET1 (SBR46)	HALL effect device turret control
Each scan (SM0.0)	ServPlan (SBR48)	Maintenance plan example: first task

Setting relevant PLC machine data

Machine data	Corresponding function	
14510[12]	JOG key layout	
14510[20]	The maximum number of tool positions	
14510[21]	Time for locking a turret (in 0.1s)	
14510[22]	The monitoring time for searching a tool (in 0.1s)	
14510[24]	Lubrication interval (in 1min)	
14510[25]	Lubrication duration (in 0.01s)	
14512[16]	Bit 4	Control of X axis rotation monitoring
	Bit 5	Control of Y axis rotation monitoring
	Bit 6	Control of Z axis rotation monitoring
	Bit 7	Handwheel assignment with the MCP / HMI
14512[17]	Bit 0	Turret function
	Bit 1	Clamping function
	Bit 2	Tailstock function
	Bit 3	Selection between handwheel and hand-held unit (0: handwheel; 1: hand-held unit)
14512[18]	Bit 2	One time automatic lubrication after the power-on
	Bit 4	Stop signal for an external spindle
	Bit 5	Fixing the direction of a spindle
	Bit 6	Hardware limit is independent of the PLC application
	Bit 7	One hardware limit triggered per axis (enabled when bit 6=0)

Machine data	Corresponding function	
14512[19]	Bit 1	Function of spindle braking
	Bit 2	Password clearing by power-on (0: delete the password; 1: do not delete the password)
	Bit 7	MM+ (Manual Machine Plus) function (enabled when the MM+ has been licensed and corresponding PLC subroutine has been called)

5.2 PLC sample application (milling)

This sample application is applicable to machines with the following configurations:

- Three axes: axes X, Y and Z, with a hardware limit switch respectively in the positive and negative directions of each axis
- An analog spindle: SP (the fourth axis)
- PLC-controlled timely and quantitatively lubrication system
- PLC-controlled cooling system

Table 5- 3 Assignment of inputs and outputs

Signal	Description	Remark
I0.0	Emergency Stop button	Normally closed
I0.1	Limit switch in the "+" direction of axis X	Normally closed
I0.2	Limit switch in the "-" direction of axis X	Normally closed
I0.3	Limit switch in the "+" direction of axis Y	
I0.4	Limit switch in the "-" direction of axis Y	
I0.5	Limit switch in the "+" direction of axis Z	Normally closed
I0.6	Limit switch in the "-" direction of axis Z	Normally closed
I0.7	Reference point switch of axis X	Normally open
I1.0	Reference point switch of axis Y	
I1.1	Reference point switch of axis Z	Normally open
I1.2	Disk-style tool magazine: tool magazine count	Valid at a low level
I1.3	Disk-style tool magazine: tool magazine at the spindle position	Valid at a low level
I1.4	Disk-style tool magazine: tool magazine at the original position	Valid at a low level
I1.5	Disk-style tool magazine: tool at the release position	Valid at a low level
I1.6	Disk-style tool magazine: tool at the clamping position	Valid at a low level
I1.7		Valid at a low level
I2.0		Normally closed
I2.1		Reserved
I2.2		
I2.3		Normally open
I2.4	Coolant level too low	Normally closed
I2.5	Cooling pump motor overload	Normally closed
I2.6	Lubricant level too low	Normally closed

Signal	Description	Remark
I2.7	Lubrication pump motor overload	Normally closed
I3.0		Reserved
I3.1		Reserved
I3.2		Reserved
I3.3		Reserved
I3.4		Reserved
I3.5		Reserved
I3.6		Reserved
I3.7		Reserved
I4.0	Handheld unit: axis X selected	Valid at a high level
I4.1	Handheld unit: axis Y selected	Valid at a high level
I4.2	Handheld unit: axis Z selected	Valid at a high level
I4.3	Handheld unit: fourth axis selected	Reserved
I4.4	Handheld unit: increment X1	Valid at a high level
I4.5	Handheld unit: increment X10	Valid at a high level
I4.6	Handheld unit: increment X100	Valid at a high level
I4.7	Handheld unit: enabled	Valid at a high level
Q0.0	Working lamp	
Q0.1		
Q0.2	Chip forward	
Q0.3	Chip backward	
Q0.4	Cooling pump	
Q0.5	Lubrication pump	
Q0.6	Safety door open	
Q0.7		
Q1.0	Magazine rotating clockwise	
Q1.1	Magazine rotating counter-clockwise	
Q1.2	Magazine approaching spindle position	
Q1.3	Magazine approaching original position	
Q1.4	Tool release from the spindle	
Q1.5		
Q1.6		
Q1.7	Handheld unit valid	

Table 5- 4 Definition of user-defined keys on the MCP

User-defined key 1	Working lamp
User-defined key 2	Manual cooling
User-defined key 3	Safe door
User-defined key 4	Manual clockwise rotation of the tool magazine
User-defined key 5	Manual reset of the tool magazine
User-defined key 6	Manual counter-clockwise rotation of the tool magazine
User-defined key 7	Removing chip forward
User-defined key 8	Removing chip backward

Structure of the sample application (OB1)

Call Conditions	Subroutine Name	Description
Each scan (SM0.0)	AUX_MCP (SBR20)	Auxiliary function
First scan (SM0.1)	PLC_INI (SBR32)	PLC initialization
Each scan (SM0.0)	EMG_STOP (SBR33)	Emergency Stop control
Each scan (SM0.0)	MCP_NCK (SBR37)	Transferring MCP and HMI signals to the NCK interface
Each scan (SM0.0)	HANDWHL (SBR39)	Selecting a hand wheel through the interface signal DB1900.DBB1xxx
Each scan (SM0.0)	AXIS_CTL (SBR40)	Coordinate enabling control, hardware limit, etc.
Each scan (SM0.0)	SPINDLE (SBR42)	Spindle control
Each scan (SM0.0)	COOLING (SBR44)	Cooling control
Each scan (SM0.0)	LUBRICAT (SBR45)	Lubrication control

Setting relevant PLC machine data

Machine data	Corresponding function	
14510[12]	JOG key layout	
14510[20]	The maximum number of tool positions	
14510[24]	Lubrication interval (in 1min)	
14510[25]	Lubrication duration (in 0.01s)	
14512[16]	Bit 1	Function of chip remover
	Bit 2	Function of safe door
	Bit 3	When the function of safe door is active, it can be triggered by M01/M02
	Bit 4	Control of X axis rotation monitoring
	Bit 5	Control of Y axis rotation monitoring
	Bit 6	Control of Z axis rotation monitoring
	Bit 7	Handwheel assignment with the MCP / HMI
14512[17]	Bit 0	Tool magazine function
	Bit 3	Selection between handwheel and hand-held unit (0: handwheel; 1: hand-held unit)
14512[18]	Bit 2	One time automatic lubrication after the power-on
	Bit 4	Stop signal for an external spindle
	Bit 5	Fixing the direction of a spindle
	Bit 6	Hardware limit is independent of the PLC application
	Bit 7	One hardware limit triggered per axis (enabled when bit 6=0)
14512[19]	Bit 1	Function of spindle braking
	Bit 2	Password clearing by power-on (0: delete the password; 1: do not delete the password)
	Bit 7	MM+ (Manual Machine Plus) function (enabled when the MM+ has been licensed and corresponding PLC subroutine has been called)

Index