Industrial Hydraulics | **Electric Drives and Controls** | Linear Motion and Assembly Technologies | Pneumatics | Service Automation | Mobile Hydraulics

Rexroth
Bosch Group

Rexroth MTC 200
PLC Interface Description

R911296974
Edition 01

Application Manual

Title

Rexroth MTC 200
PLC Interface Description

Type of Documentation **Application Manual**

Document Typecode DOK-MTC200-SPS*GWY*V23-AW01-EN-P

Internal File Reference Document Number, 120-1700-B399-01/EN

Purpose of Documentation This documentation describes the interface between PLC and NC.

Record of Revisions

Description	Release Date	Notes
120-1700-B399-01/EN	10.2003	Valid from version 23

Published by Rexroth Indramat GmbH
Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main

Telephone +49 (0)93 52/40-0 • Tx 68 94 21 • Fax +49 (0)93 52/40-48 85

http://www.boschrexroth.de/

Dept. BRC/ESM3 (MiAl)

Dept. BRC/ESM6 (DiHa)

Note This document has been printed on chlorine-free bleached paper.

Contents

1 Data Types Rexroth MTC 200 1-1

1.1 Overview of the I/O Interface between the PLC and the Rexroth MTC 200 ... 1-1
1.2 Interface Signals Between NC and PLC ... 1-1
Axis Signals ... 1-1
Process Signals ... 1-7
Mechanism Signals ... 1-13
1.3 Process Signals ... 1-16
Monitoring ... 1-16
Power Activation ... 1-23
Operating modes ... 1-30
NC program selection ... 1-39
NC Program Control ... 1-40
NC Program Status ... 1-52
Feed and Spindle Override ... 1-62
NC program synchronization ... 1-63
Tool magazine motion control ... 1-79
Spindle stop Signals ... 1-98
Synchronous Axes ... 1-100
Synchronized Spindles ... 1-101
1.4 Mechanism Signals ... 1-103
External Mechanisms ... 1-103
Mechanism Control Signals ... 1-104
Mechanism Status Signals ... 1-104
1.5 Axis Signals ... 1-106
Axis Monitoring Signals ... 1-107
Axis Control Signals ... 1-109
Axis Status Signals ... 1-128

2 Functions Rexroth MTC 200 2-1

2.1 Auxiliary Functions ... 2-1
M functions ... 2-1
S Functions ... 2-4
T Functions ... 2-7
Q functions ... 2-9
E Functions ... 2-10
2.2 Events ... 2-13
2.3 Diagnosis functions ... 2-17
2.4 Magazine functions ... 2-19

Tool Storage Unit Motion Commands of the PLC 2-20
Magazine homing 2-22
Tool Magazine Positioning 2-24
2.5 Tool Change Functions 2-28
Complete tool change 2-29
Magazine-to-spindle tool change 2-31
2.6 Tool transfer Functions 2-35
Method of Operation of a Tool Transfer 2-35
Tool Transfer Procedure and Function Modules 2-37
Further Tool Transfer Functions 2-41
2.7 Combined spindle/turret axis 2-42
Interrogating and Acknowledging Turret Mode Preselection 2-44
Interrogating and Acknowledging Spindle Mode Preselection 2-46
2.8 Selectable Main Spindle / Rotary Axis Mode 2-47
2.9 Hand Wheel Functions 2-53
2.10 Functions for Asynchronous Turret Mode 2-56
2.11 Functions for Single Dimension Collision Monitoring 2-57
2.12 Enabling the NC 2-59

3 Function Block Diagrams Rexroth MTC 200 3-1

3.1 Standard Function Block Diagrams 3-1
General Comments on NC Data Access Procedures 3-1
Function blocks for NC variables 3-1
Function Blocks for NC Program Memory 3-6
Function blocks for the SERCOS required data channel 3-9
Function Blocks for Accessing Machine Data 3-17
Function Blocks for Accessing Zero Point Data 3-20
Function blocks for serial interfaces 3-23
Function blocks for the graphical user interface 3-24
Function blocks for accessing tool data 3-28
Tool Functions in the PLC 3-44
Function blocks for accessing D corrections "DCD_RD" and "DCD_WR"" 3-60

4 Error Handling MTC200 4-1

4.1 Errors in Functions and Function Blocks 4-1
4.2 Error Numbers 4-10

5 Index 5-1

6 Service & Support 6-1

6.1 Helpdesk 6-1
6.2 Service-Hotline 6-1
6.3 Internet 6-1
6.4 Vor der Kontaktaufnahme... - Before contacting us... 6-1
6.5 Kundenbetreuungsstellen - Sales & Service Facilities 6-2

1 Data Types Rexroth MTC 200

1.1 Overview of the I/O Interface between the PLC and the Rexroth MTC 200

A PLC has been integrated into the control structure in order to adapt the NC to the different input and output functions of the individual machine configurations. Information between the NC and the PLC is exchanged via a common memory area that the NC and PLC can both access.

Each bit of this memory area corresponds to an input or an output signal. Additional memory areas for data transfer have been defined. This I/O section can be addressed like real physical inputs and outputs, but it does not really exist. These inputs and outputs are therefore referred to as "soft I/Os".

Address and significance of each signal in this memory area have been invariably defined and must be used accordingly by the PLC programmer.

By using symbolic operands the PLC programmer is not obliged to know the absolute addresses of the individual signals. The "symbolic operand" and "absolute operand" are allocated subsequently.

This Manual lists and describes each signal that is exchanged between the NC and the PLC via the interface and is available to the PLC programmer.

The signals are subdivided into the following data structures:

Data structure	Brief Description
Process Signals	These signals exist once for each of the 7 NC processes.
Mechanism signals	These signals exist once for each of 25 external mechanisms of the NC.
Axis Signals	These signals exist once for each of the maximum of 32 axes of the NC.
Local inputs and outputs	These signals are additionally available inputs and outputs for the machine operation.

Fig. 1-1: Grouping of signals

1.2 Interface Signals Between NC and PLC

Axis Signals

The following addresses must be used for declaration when the axis signals are employed as a structured standard type:

Axis number	Inputs NC -> PLC	Outputs PLC -> NC
1	%IW*.0	%QW*.0
2	%IW*.8	%QW*.8
3	%IW*.16	%QW*.16
4	%IW*.24	%QW*.24

5	%IW*.32	%QW*.32
6	%IW*.40	%QW*.40
7	%IW*.48	%QW*.48
8	%IW*.56	%QW*.56
9	%IW*.64	%QW*.64
10	%IW*.72	%QW*.72
11	%IW*.80	%QW*.80
12	%IW*.88	%QW*.88
13	%IW*.96	%QW*.96
14	%IW*.104	%QW*.104
15	%IW*.112	%QW*.112
16	%IW*.120	%QW*.120
17	%IW*.128	%QW*.128
18	%IW*.136	%QW*.136
19	%IW*.144	%QW*.144
20	%IW*.152	%QW*.152
21	%IW*.160	%QW*.160
22	%IW*.168	%QW*.168
23	%IW*.176	%QW*.176
24	%IW*.184	%QW*.184
25	%IW*.192	%QW*.192
26	%IW*.0	%QW*.200
27	%IW*.208	%QW*.208
28	%IW*.216	%QW*.216
29	%IW*.224	%QW*.224
30	%IW*.232	%QW*.232
31	%IW*.240	%QW*.240
32	%IW*.248	%QW*.248

Fig. 1-2: List of axis signals

The following table gives you an overview of the axis signals. In addition to the name and the data type of the individual signals, the table lists the signal addresses.

The following points must be noted:

<xx> in the designation stands for the axis number.

<yy> in the address refers to the byte number of the start address of the related axis number.

Example:

You wish to determine the address of the AxxC.RAPID signal (rapid traverse for spindle) for axis number 3.

The following entry exists in the table:

AxxC.RAPID %Q*.(yy+4).6 BOOL

The address results as follows:

Start address for axis no. 3: %QW*.16 <yy> corresponds to 16

Address: %Q*.(16+4).6 = %Q*.20.6

Axis control signals (PLC => NC)

Identifier	Address	Type	Comment	Page
AxxC.HOMLS	%Q*.(yy).0	BOOL	;Homing switch	1-111
AxxC.READY	%Q*.(yy)0.1	BOOL	;Ready for operation	1-109
AxxC.JGPOS	%Q*.(yy)0.2	BOOL	;Positive jogging	1-113
AxxC.JGNEG	%Q*.(yy)0.3	BOOL	;Negative jogging	1-115
AxxC.HOME	%Q*.(yy)0.4	BOOL	;Homing axis	1-113
AxxC.ENABL	%Q*.(yy)0.5	BOOL	;Enabling axis	1-110
AxxC.STRBP	%Q*.(yy)0.6	BOOL	;Position strobe	1-111
AxxC.QDDS	%Q*.(yy)0.7	BOOL	;Real time control bit SERCOS-drive	1-116
AxxC.OTRVL	%Q*.(yy+1).0	BOOL	;Safety limit switch	1-107
AxxC.MTAS	%Q*.(yy+1)0.1	BOOL	;Motor temperature circuit breaker	1-107
AxxC.F12	%Q*.(yy+1)0.2	BOOL	;not assigned, 2#0	
AxxC.MHOLD	%Q*.(yy+1)0.3	BOOL	;Motion hold	1-117
AxxC.M3	%Q*.(yy+1)0.4	BOOL	;Start clockwise spindle rotation	1-124
AxxC.M4	%Q*.(yy+1)0.5	BOOL	;Start anti-clockwise spindle rotation	1-124
AxxC.M5	%Q*.(yy+1)0.6	BOOL	;Spindle stop	1-125
AxxC.M19	%Q*.(yy+1)0.7	BOOL	;Position spindle	1-125
AxxC.GEAR1	%Q*.(yy+2)0.0	BOOL	;Actual gear step, bit 1	1-118
AxxC.GEAR2	%Q*.(yy+2)0.1	BOOL	;Actual gear step, bit 2	1-118
AxxC.GEAR3	%Q*.(yy+2).2	BOOL	;Actual gear step, bit 3	1-118
AxxC.F23	%Q*.(yy+2)0.3	BOOL	;not assigned, 2#0	
AxxC.F24	%Q*.(yy+2)0.4	BOOL	;not assigned, 2#0	
AxxC.F25	%Q*.(yy+2)0.5	BOOL	;not assigned, 2#0	
AxxC.F26	%Q*.(yy+2)0.6	BOOL	;not assigned, 2#0	
AxxC.F27	%Q*.(yy+2)0.7	BOOL	;not assigned, 2#0	
AxxC.SPEED	%Q*.(yy+3).0	BOOL	;Feed and speed limiting	1-127
AxxC.LIMIT	%Q*.(yy+3)0.1	BOOL	;Software limits	1-126
AxxC.LOCK	%Q*.(yy+3)0.2	BOOL	Axis / spindle lock	1-127
AxxC.F33	%Q*.(yy+3)0.3	BOOL	;not assigned, 2#0	
AxxC.F34	%Q*.(yy+3)0.4	BOOL	;not assigned, 2#0	
AxxC.F35	%Q*.(yy+3)0.5	BOOL	;not assigned, 2#0	
AxxC.F36	%Q*.(yy+3)0.6	BOOL	;not assigned, 2#0	
AxxC.F37	%Q*.(yy+3)0.7	BOOL	;not assigned, 2#0	
AxxC.SPHLT	%Q*.(yy+4)0.0	BOOL	;Spindle stop upon program halt	1-122
AxxC.SPSTP	%Q*.(yy+4)0.1	BOOL	;Spindle stop upon program stop	1-121
AxxC.SPSTE	%Q*.(yy+4)0.2	BOOL	;Spindelstop bei Programm-ende	1-122
AxxC.SPRST	%Q*.(yy+4)0.3	BOOL	;No spindle stop upon Control Reset	1-123
AxxC.N_CMD	%Q*.(yy+4)0.4	BOOL	;Spindle command speed reached (inverted)	1-122
AxxC.F45	%Q*.(yy+4)0.5	BOOL	;not assigned, 2#0	
AxxC.RAPID	%Q*.(yy+4).6	BOOL	;Rapid traverse signal for spindles	1-126

AxxC.F47	%Q*.(yy+4)0.7	BOOL	;not assigned, 2#0	
AxxC.OVRD	%QB*.(yy+5)	USINT	;Axis override	1-123
AxxC.SAFSS	%Q*.(yy+6).0	BOOL	;Activating safe standstill	1-137
AxxC.SAFOS	%Q*.(yy+6)0.1	BOOL	;Activating safe operational stop	1-137
AxxC.SAFA1	%Q*.(yy+6)0.2	BOOL	;Activating safely reduced velocity with safely limited abs. position 1	1-137
AxxC.SAFA2	%Q*.(yy+6)0.3	BOOL	;Activating safely reduced velocity with safely limited abs. position 2	1-137
AxxC.SAFAG	%Q*.(yy+6)0.4	BOOL	;Safety agree sensor	1-137
AxxC.SAFRS	%Q*.(yy+6)0.5	BOOL	Safe reference switch	1-137
AxxC.F66	%Q*.(yy+6)0.6	BOOL	;not assigned, 2#0	
AxxC.F67	%Q*.(yy+6)0.7	BOOL	;not assigned, 2#0	
AxxC.F70	%Q*.(yy+7).0	BOOL	;not assigned, 2#0	
AxxC.F71	%Q*.(yy+7)0.1	BOOL	;not assigned, 2#0	
AxxC.F72	%Q*.(yy+7)0.2	BOOL	;not assigned, 2#0	
AxxC.F73	%Q*.(yy+7)0.3	BOOL	;not assigned, 2#0	
AxxC.F74	%Q*.(yy+7)0.4	BOOL	;not assigned, 2#0	
AxxC.F75	%Q*.(yy+7)0.5	BOOL	;not assigned, 2#0	
AxxC.F76	%Q*.(yy+7)0.6	BOOL	;not assigned, 2#0	
AxxC.F77	%Q*.(yy+7)0.7	BOOL	;not assigned, 2#0	

Fig. 1-3: Axis control signals (PLC => NC)

Axis status signals (NC => PLC)

Identifier	Address	Type	Comment	Page
AxxS.RF	%I*.(yy).0	BOOL	;Controller enabling signal	1-128
AxxS.HOMED	%I*.(yy)0.1	BOOL	;Homed	1-130
AxxS.MVPOS	%I*.(yy)0.2	BOOL	;Movement in positive direction	1-132
AxxS.MVNEG	%I*.(yy)0.3	BOOL	;Movement in negative direction	1-130
AxxS.BBDIG	%I*.(yy)0.4	BOOL	;Digital drives ready for operation	1-129
AxxS.IDDS	%I*.(yy)0.5	BOOL	Real time status bit SERCOS drive	1-131
AxxS.SPEED	%I*.(yy)0.6	BOOL	;Feed and speed limiting	1-138
AxxS.LIMIT	%I*.(yy)0.7	BOOL	;Software limits activated	1-139
AxxS.WP0	%I*.(yy+1).0	BOOL	;Waypoint 0	1-131
AxxS.WP1	%I*.(yy+1)0.1	BOOL	;Waypoint 1	1-131
AxxS.WP2	%I*.(yy+1)0.2	BOOL	;Waypoint 2	1-131
AxxS.WP3	%I*.(yy+1)0.3	BOOL	;Waypoint 3	1-131
AxxS.WP4	%I*.(yy+1)0.4	BOOL	;Waypoint 4	1-131
AxxS.WP5	%I*.(yy+1)0.5	BOOL	;Waypoint 5	1-131
AxxS.WP6	%I*.(yy+1)0.6	BOOL	;Waypoint 6	1-131
AxxS.WP7	%I*.(yy+1)0.7	BOOL	;Waypoint 7	1-131
AxxS.F20	%I*.(yy+2).0	BOOL	;not assigned, 2#0	
AxxS.F21	%I*.(yy+2)0.1	BOOL	;not assigned, 2#0	
AxxS.LD90	%I*.(yy+2)0.2	BOOL	;90% load	1-133
AxxS.SYNC	%I*.(yy+2)0.3	BOOL	;Synchronous operation	1-134
AxxS.F24	%I*.(yy+2)0.4	BOOL	;not assigned, 2#0	
AxxS.F25	%I*.(yy+2)0.5	BOOL	;not assigned, 2#0	
AxxS.F26	%I*.(yy+2)0.6	BOOL	;not assigned, 2#0	
AxxS.F27	%I*.(yy+2)0.7	BOOL	;not assigned, 2#0	
AxxS.N_CMD	%I*.(yy+3).0	BOOL	;N actual = N command	1-134
AxxS.N_MIN	%I*.(yy+3)0.1	BOOL	;N act = 0	1-135
AxxS.F32	%I*.(yy+3)0.2	BOOL	;not assigned, 2#0	
AxxS.MD_DX	%I*.(yy+3)0.3	BOOL	;M d >= M dx	1-136
AxxS.F34	%I*.(yy+3)0.4	BOOL	;not assigned, 2#0	
AxxS.N_MAX	%I*.(yy+3)0.5	BOOL	;N command >= N limit	1-135
AxxS.INPOS	%I*.(yy+3)0.6	BOOL	;In position	1-136
AxxS.P_PX	%I*.(yy+3)0.7	BOOL	;P >= Px	1-136
AxxS.MCPOS	%I*.(yy+4).0	BOOL	;Positive movement announcement	1-132
AxxS.MCNEG	%I*.(yy+4)0.1	BOOL	;Negative movement announcement	1-133
AxxS.POSWN	%I*.(yy+4)0.2	BOOL	;In position window	1-133
AxxS.MHOLD	%I*.(yy+4)0.3	BOOL	;Motion hold active	1-139
AxxS.LOCK	%I*.(yy+4)0.4	BOOL	Axis / spindle lock	1-139
AxxS.F45	%I*.(yy+4)0.5	BOOL	;not assigned, 2#0	
AxxS.F46	%I*.(yy+4)0.6	BOOL	;not assigned, 2#0	
AxxS.F47	%I*.(yy+4)0.7	BOOL	;not assigned, 2#0	

AxxS.F50	%I*.(yy+5).0	BOOL	;not assigned, 2#0	
AxxS.F51	%I*.(yy+5)0.1	BOOL	;not assigned, 2#0	
AxxS.F52	%I*.(yy+5)0.2	BOOL	;not assigned, 2#0	
AxxS.F53	%I*.(yy+5)0.3	BOOL	;not assigned, 2#0	
AxxS.F54	%I*.(yy+5)0.4	BOOL	;not assigned, 2#0	
AxxS.F55	%I*.(yy+5)0.5	BOOL	;not assigned, 2#0	
AxxS.F56	%I*.(yy+5)0.6	BOOL	;not assigned, 2#0	
AxxS.F57	%I*.(yy+5)0.7	BOOL	;not assigned, 2#0	
AxxS.SAFAC	%I*.(yy+6).0	BOOL	;Safety function active	1-137
AxxS.SAFRY	%I*.(yy+6)0.1	BOOL	;Safety changeover ready	1-137
AxxS.SAFP1	%I*.(yy+6)0.2	BOOL	;Safe position switch 1	1-137
AxxS.SAFP2	%I*.(yy+6)0.3	BOOL	;Safe position switch 2	1-137
AxxS.SAFP3	%I*.(yy+6)0.4	BOOL	;Safe position switch 3	1-137
AxxS.SAFP4	%I*.(yy+6)0.5	BOOL	;Safe position switch 4	1-137
AxxS.SAFSL	%I*.(yy+6)0.6	BOOL	;Safe starting lockout	1-137
AxxS.SAFEN	%I*.(yy+6)0.7	BOOL	;Safety enable	1-137

Fig. 1-4: Axis status signals (NC => PLC)

Process Signals

The following addresses must be used for declaration when the process signals are employed as a structured standard type:

Process number	Inputs NC => PLC	Outputs PLC => NC
0	%IW*.256	%QW*.256
1	%IW*.276	%QW*.276
2	%IW*.296	%QW*.296
3	%IW*.316	%QW*.316
4	%IW*.336	%QW*.336
5	%IW*.356	%QW*.356
6	%IW*.376	%QW*.376

Fig. 1-5: List of process signals

The following table gives you an overview of the process signals. In addition to the name and the data type of the individual signals, the table lists the signal addresses.

The following points must be noted:

- <xx> in the designation stands for the process number.
- <yy> in the address refers to the byte number of the start address of the related process number.

Example:

You wish to determine the address of the PxxC.CLEAR signal (clear error / reset) for process number 2.

The following entry exists in the table:

PxxC.CLEAR   %Q*.(yy+4).0   BOOL

The address results as follows:

Start address for process no. 2: %QW*.296 => <yy> corresponds to 296

Adresse: %Q*.(296+4).0 = %Q*.300.0

Process control signals (PLC => NC)

Identifier	Address	Type	Comment	Page
PxxC.F00	%Q*.(yy).0	BOOL	;not assigned, 2#0	
PxxC.PARAM	%Q*.(yy)0.1	BOOL	;Transfer strobe parameter test	1-34
PxxC.BLSKP	%Q*.(yy)0.2	BOOL	;Skip NC block	1-41
PxxC.M001	%Q*.(yy)0.3	BOOL	;Optional stop M001	1-44
PxxC.ENABL	%Q*.(yy)0.4	BOOL	;Process enable signal	1-40
PxxC.ADV	%Q*.(yy)0.5	BOOL	;Start of advance program	1-41
PxxC.REV	%Q*.(yy)0.6	BOOL	;Start of reverse program	1-45
PxxC.STOP	%Q*.(yy)0.7	BOOL	;Program stop	1-48
PxxC.MODE0	%Q*.(yy+1).0	BOOL	;Mode 2^0	1-30
PxxC.MODE1	%Q*.(yy+1)0.1	BOOL	;Mode 2^1	1-30
PxxC.JOGM0	%Q*.(yy+1)0.2	BOOL	;Jog mode 2^0	1-35
PxxC.JOGM1	%Q*.(yy+1)0.3	BOOL	;Jog mode 2^1	1-35
PxxC.JOGM2	%Q*.(yy+1)0.4	BOOL	;Jog mode 2^2	1-35
PxxC.SINGL	%Q*.(yy+1)0.5	BOOL	; Single step program execution	1-37
PxxC.F16	%Q*.(yy+1)0.6	BOOL	;not assigned, 2#0	
PxxC.RAPID	%Q*.(yy+1)0.7	BOOL	;Rapid traverse rate	1-38
PxxC.BBSUP	%Q*.(yy+2)0.0	BOOL	;Power supply unit ready for operation	1-20
PxxC.TPSUP	%Q*.(yy+2)0.1	BOOL	;Temperature control of power supply unit	1-21
PxxC.F22	%Q*.(yy+2).2	BOOL	;not assigned, 2#0	
PxxC.F23	%Q*.(yy+2)0.3	BOOL	;not assigned, 2#0	
PxxC.BBMSP	%Q*.(yy+2)0.4	BOOL	;Main drive ready for operation	1-22
PxxC.UDRDY	%Q*.(yy+2)0.5	BOOL	;DC bus voltage available	1-22
PxxC.PWRDY	%Q*.(yy+2)0.6	BOOL	;Main load contactor closed	1-21
PxxC.POWON	%Q*.(yy+2)0.7	BOOL	;PLC power request	1-28
PxxC.EXT24	%Q*.(yy+3).0	BOOL	;External 24 VDC supply OK	1-17
PxxC.LINE	%Q*.(yy+3)0.1	BOOL	;Aux. mains voltage present	1-18
PxxC.ESTAT	%Q*.(yy+3)0.2	BOOL	;Station EMERGENCY STOP	1-19
PxxC.EMACH	%Q*.(yy+3)0.3	BOOL	;Machine EMERGENCY STOP	1-19
PxxC.ESTP1	%Q*.(yy+3)0.4	BOOL	;Guard 1	1-20
PxxC.ESTP2	%Q*.(yy+3)0.5	BOOL	;Guard 2	1-20
PxxC.ESTP3	%Q*.(yy+3)0.6	BOOL	;Guard 3	1-20
PxxC.F37	%Q*.(yy+3)0.7	BOOL	;not assigned, 2#0	
PxxC.CLEAR	%Q*.(yy+4)0.0	BOOL	;Clear error	1-49
PxxC.DP	%Q*.(yy+4)0.1	BOOL	;Define process	1-63
PxxC.RP	%Q*.(yy+4)0.2	BOOL	;Start reverse program	1-70
PxxC.AP	%Q*.(yy+4)0.3	BOOL	;Start advance program	1-66
PxxC.QP	%Q*.(yy+4)0.4	BOOL	;Acknowledge process	1-74
PxxC.POK	%Q*.(yy+4)0.5	BOOL	;Part machined	1-74
PxxC.SP	%Q*.(yy+4).6	BOOL	;NC program number from PLC active	1-40
PxxC.F47	%Q*.(yy+4)0.7	BOOL	;not assigned, 2#0	
PxxC.MGBP	%Q*.(yy+5).0	BOOL	;Move magazine to home position	1-85
PxxC.MGPOS	%Q*.(yy+5)0.1	BOOL	;Rotate magazine in positive direction	1-86
PxxC.MGNEG	%Q*.(yy+5)0.2	BOOL	;Rotate magazine in negative direction	1-87

PxxC.MGHOM	%Q*.(yy+5)0.3	BOOL	;Magazine Homing	1-85
PxxC.F54	%Q*.(yy+5)0.4	BOOL	;not assigned, 2#0	
PxxC.MGNSL	%Q*.(yy+5)0.5	BOOL	;Operation without setup list	1-82
PxxC.MGMAN	%Q*.(yy+5)0.6	BOOL	;Magazine manual/program mode	1-83
PxxC.MGENA	%Q*.(yy+5)0.7	BOOL	;Magazine enabling	1-81
PxxC.MGNTL	%Q*.(yy+6).0	BOOL	;Switching off tool life count	1-83
PxxC.BPSTR	%Q*.(yy+6)0.1	BOOL	;Start NC program restart	1-52
PxxC.MGWTC	%Q*.(yy+6)0.2	BOOL	;Switch off automatic equipment check	1-82
PxxC.MGITW	%Q*.(yy+6)0.3	BOOL	;Ignore worn tools	1-83
PxxC.REPOS	%Q*.(yy+6)0.4	BOOL	;Repositioning	1-51
PxxC.RESTA	%Q*.(yy+6)0.5	BOOL	;Return to contour	1-51
PxxC.F66	%Q*.(yy+6)0.6	BOOL	;not assigned, 2#0	
PxxC.F67	%Q*.(yy+6)0.7	BOOL	;not assigned, 2#0	
PxxC.SCON1	%Q*.(yy+7).0	BOOL	;Synchronous compound 1 ON	1-100
PxxC.SCON2	%Q*.(yy+7)0.1	BOOL	;Synchronous compound 2 ON	1-100
PxxC.SCON3	%Q*.(yy+7)0.2	BOOL	;Synchronous compound 3 ON	1-100
PxxC.SCON4	%Q*.(yy+7)0.3	BOOL	;Synchronous compound 4 ON	1-100
PxxC.SS1ON	%Q*.(yy+7)0.4	BOOL	;Spindle synchronization 1 ON	1-101
PxxC.SS2ON	%Q*.(yy+7)0.5	BOOL	;Spindle synchronization 2 ON	1-101
PxxC.SS1MT	%Q*.(yy+7)0.6	BOOL	;Minimize spindle sync 1 torsion moment	1-100
PxxC.SS2MT	%Q*.(yy+7)0.7	BOOL	;Minimize spindle sync 2 torsion moment	1-100
PxxC.DRYRN	%Q*.(yy+8).0	BOOL	;Switch on test mode	1-52
PxxC.F81	%Q*.(yy+8)0.1	BOOL	;not assigned, 2#0	
PxxC.F82	%Q*.(yy+8)0.2	BOOL	;not assigned, 2#0	
PxxC.F83	%Q*.(yy+8)0.3	BOOL	;not assigned, 2#0	
PxxC.F84	%Q*.(yy+8)0.4	BOOL	;not assigned, 2#0	
PxxC.F85	%Q*.(yy+8)0.5	BOOL	;not assigned, 2#0	
PxxC.F86	%Q*.(yy+8)0.6	BOOL	;not assigned, 2#0	
PxxC.F87	%Q*.(yy+8)0.7	BOOL	;not assigned, 2#0	
PxxC.FOVRD	%QB*.(yy+9)	USINT	;Feed override	1-62
PxxC.ROVRD	%QB*.(yy+10)	USINT	;Rapid traverse override	1-63
PxxC.PRGNR	%QB*.(yy+11)	USINT	;Program number PLC-NC	1-39
PxxC.MGAP	%QW*.(yy+12)	INT	;Current magazine location	1-89
PxxC.MGJG0	%QW*.(yy+13).0	BOOL	;Magazine jog operating mode (Bit 2^0) during variable tool location position	1-87
PxxC.MGJG1	%QW*.(yy+13)0.1	BOOL	;Magazine jog operating mode (Bit 2^1) during variable tool location position	1-87

Fig. 1-6: Process control signals (PLC => NC)

Identifier	Address	Type	Comment	Page
PxxS.POWER	%I*.(yy).0	BOOL	;Power enabled	1-29
PxxS.POWIN	%I*.(yy)0.1	BOOL	;Control voltage interruption	1-57
PxxS.ERROR	%I*.(yy)0.2	BOOL	;Error	1-52
PxxS.ACTIV	%I*.(yy)0.3	BOOL	;NC program active	1-55
PxxS.READY	%I*.(yy)0.4	BOOL	;Ready to start	1-53
PxxS.RUN	%I*.(yy)0.5	BOOL	;NC block active	1-54
PxxS.REV	%I*.(yy)0.6	BOOL	;Reverse program active	1-55
PxxS.STOP	%I*.(yy)0.7	BOOL	;NC program stopped	1-56
PxxS.POWEN	%I*.(yy+1).0	BOOL	;NC power disabled	1-27
PxxS.POWON	%I*.(yy+1)0.1	BOOL	;NC power enabled	1-29
PxxS.RVACT	%I*.(yy+1)0.2	BOOL	Reverse vector active	1-56
PxxS.BPACT	%I*.(yy+1)0.3	BOOL	;NC program restart is active	1-59
PxxS.MDIAC	%I*.(yy+1)0.4	BOOL	;MDI is active	1-57
PxxS.TRANS	%I*.(yy+1)0.5	BOOL	;Polar/cart. coord. translation active	1-53
PxxS.CREST	%I*.(yy+1)0.6	BOOL	;Conditions restored	1-59
PxxS.REPOS	%I*.(yy+1)0.7	BOOL	;Repositioning/restart active	1-59
PxxS.POK	%I*.(yy+2).0	BOOL	;Part machined	1-74
PxxS.DP	%I*.(yy+2)0.1	BOOL	;Define process	1-63
PxxS.RP	%I*.(yy+2)0.2	BOOL	;Start reverse program	1-70
PxxS.AP	%I*.(yy+2)0.3	BOOL	;Start advance program	1-66
PxxS.QP	%I*.(yy+2)0.4	BOOL	;Acknowledge process	1-74
PxxS.LP	%I*.(yy+2)0.5	BOOL	;Lock process	1-66
PxxS.F26	%I*.(yy+2)0.6	BOOL	;not assigned, 2#0	
PxxS.F27	%I*.(yy+2)0.7	BOOL	;not assigned, 2#0	
PxxS.F30	%I*.(yy+3).0	BOOL	;not assigned, 2#0	
PxxS.F31	%I*.(yy+3)0.1	BOOL	;not assigned, 2#0	
PxxS.F32	%I*.(yy+3)0.2	BOOL	;not assigned, 2#0	
PxxS.F33	%I*.(yy+3)0.3	BOOL	;not assigned, 2#0	
PxxS.MGENA	%I*.(yy+3)0.4	BOOL	;Magazine enabled	1-81
PxxS.F35	%I*.(yy+3)0.5	BOOL	;not assigned, 2#0	
PxxS.F36	%I*.(yy+3)0.6	BOOL	;not assigned, 2#0	
PxxS.F37	%I*.(yy+3)0.7	BOOL	;not assigned, 2#0	
PxxS.MGWRN	%I*.(yy+4).0	BOOL	;Life count monitoring: warning limit	1-96
PxxS.MGTWO	%I*.(yy+4)0.1	BOOL	;Life count monitoring: worn	1-94
PxxS.MGERR	%I*.(yy+4)0.2	BOOL	;Tool error status	1-97
PxxS.F43	%I*.(yy+4)0.3	BOOL	;not assigned, 2#0	
PxxS.MGREQ	%I*.(yy+4)0.4	BOOL	;Magazine required	1-91
PxxS.MGMOV	%I*.(yy+4)0.5	BOOL	;Magazine in motion	1-92
PxxS.MGMAN	%I*.(yy+4)0.5	BOOL	;Magazine moved by hand/program	1-92

PxxS.MGBP	%I*.(yy+4)0.7	BOOL	;Magazine in home position	1-93
PxxS.G00	%I*.(yy+5).0	BOOL	;G00 rapid traverse active	1-61
PxxS.G74	%I*.(yy+5)0.1	BOOL	;G74 homing active	1-61
PxxS.G08	%I*.(yy+5)0.2	BOOL	;G08 contouring mode	1-61
PxxS.TREAT	%I*.(yy+5)0.3	BOOL	;Thread cutting active	1-61
PxxS.G96	%I*.(yy+5)0.4	BOOL	;Constant cutting speed	1-62
PxxS.F55	%I*.(yy+5)0.5	BOOL	;not assigned, 2#0	
PxxS.F56	%I*.(yy+5)0.6	BOOL	;not assigned, 2#0	
PxxS.F57	%I*.(yy+5)0.7	BOOL	;not assigned, 2#0	
PxxS.SCON1	%I*.(yy+6).0	BOOL	;Synchronous compound 1 ON	1-100
PxxS.SCON2	%I*.(yy+6)0.1	BOOL	;Synchronous compound 2 ON	1-100
PxxS.SCON3	%I*.(yy+6)0.2	BOOL	;Synchronous compound 3 ON	1-100
PxxS.SCON4	%I*.(yy+6)0.3	BOOL	;Synchronous compound 4 ON	1-100
PxxS.SS1OK	%I*.(yy+6)0.4	BOOL	;Spindle synchronization 1 OK	1-102
PxxS.SS2OK	%I*.(yy+6)0.5	BOOL	;Spindle synchronization 2 OK	1-102
PxxS.SS1ER	%I*.(yy+6)0.6	BOOL	;Spindelsynchronisation 1 Fehler	1-102
PxxS.SS1ER	%I*.(yy+6)0.7	BOOL	;Spindelsynchronisation 2 Fehler	1-102
PxxS.PRGNR	%IB*.(yy+7)	USINT	;Program number NC PLC	1-39
PxxS.MGCP	%IW*.(yy+8)	INT	;Magazine command location	1-93
PxxS.PROC	%IW*.(yy+10)	INT	;Process number	1-74
PxxS.THMIS	%I*.(yy+12).0	BOOL	;Thrust missing	1-60
PxxS.EXCTH	%I*.(yy+12)0.1	BOOL	;Excessive thrust	1-60
PxxS.DRYRN	%I*.(yy+12)0.2	BOOL	;Test mode is active	1-60
PxxS.MODE0	%I*.(yy+12)0.3	BOOL	;Selected operation mode (Bit 2^0)	1-34
PxxS.MODE1	%I*.(yy+12)0.4	BOOL	;Selected operation mode (Bit 2^1)	1-34
PxxS.F125	%I*.(yy+12)0.5	BOOL		
PxxS.F126	%I*.(yy+12)0.6	BOOL		
PxxS.F127	%I*.(yy+12)0.7	BOOL		
PxxS.F130	%I*.(yy+13).0	BOOL		
PxxS.F131	%I*.(yy+13)0.1	BOOL		
PxxS.F132	%I*.(yy+13)0.2	BOOL		
PxxS.F133	%I*.(yy+13)0.3	BOOL		
PxxS.F134	%I*.(yy+13)0.4	BOOL		
PxxS.F135	%I*.(yy+13)0.5	BOOL		
PxxS.F136	%I*.(yy+13)0.6	BOOL		
PxxS.F137	%I*.(yy+13)0.7	BOOL		

Fig. 1-7: Process status signals (NC => PLC)

Mechanism Signals

The following addresses must be used for declaration when the mechanism signals are employed as a structured standard type:

Mechanism number	Inputs NC →PLC	Outputs PLC →NC
1	%IW*.416	%QW*.416
2	%IW*.420	%QW*.420
3	%IW*.424	%QW*.424
4	%IW*.428	%QW*.428
5	%IW*.432	%QW*.432
6	%IW*.436	%QW*.436
7	%IW*.440	%QW*.440
8	%IW*.444	%QW*.444
9	%IW*.448	%QW*.448
10	%IW*.452	%QW*.452
11	%IW*.456	%QW*.456
12	%IW*.460	%QW*.460
13	%IW*.464	%QW*.464
14	%IW*.468	%QW*.460
15	%IW*.472	%QW*.472
16	%IW*.476	%QW*.476
17	%IW*.480	%QW*.480
18	%IW*.484	%QW*.484
19	%IW*.488	%QW*.488
20	%IW*.492	%QW*.492
21	%IW*.496	%QW*.496
22	%IW*.500	%QW*.500
23	%IW*.504	%QW*.504
24	%IW*.508	%QW*.508
25	%IW*.512	%QW*.512

Fig. 1-8: Mechanism signals

The following table gives you an overview of the mechanism signals. In addition to the name and the data type of the individual signals, the table lists the signal addresses.

The following points must be noted:

- <xx> in the designation stands for the mechanism number.
- <yy> in the address refers to the byte number of the start address of the related mechanism number.

Example:

You wish to determine the address of the MxxC.POK signal (Part OK) for mechanism number 7.

The following entry exists in the table:

MxxC.POK %Q*.(yy).1 BOOL

The address results as follows:

Start address for mechanism no. 7: %QW*.440 => <yy> corresponds to 440

Address: %Q*.(440).1 = %Q*.440.1

Mechanism control signals (PLC => NC)

Identifier	Address	Type	Comment	Page
MxxC.QP	%Q*.(yy).0	BOOL	;Acknowledge mechanism	1-104
MxxC.POK	%Q*.(yy)0.1	BOOL	;Part OK mechanism	1-104
MxxC.F2	%Q*.(yy)0.2	BOOL	;not assigned, 2#0	
MxxC.F3	%Q*.(yy)0.3	BOOL	;not assigned, 2#0	
MxxC.F4	%Q*.(yy)0.4	BOOL	;not assigned, 2#0	
MxxC.F5	%Q*.(yy)0.5	BOOL	;not assigned, 2#0	
MxxC.F6	%Q*.(yy)0.6	BOOL	;not assigned, 2#0	
MxxC.F7	%Q*.(yy)0.7	BOOL	;not assigned, 2#0	
frei1	%QB*.(yy+1)	BYTE	;not assigned	
frei2	%QW*.(yy+2)	WORD	;not assigned	

Fig. 1-9: Mechanism control signals (PLC => NC)

Mechanism status signals (NC => PLC)

Identifier	Address	Type	Comment	Page
MxxS.DP	%I*.(yy).0	BOOL	;Define mechanism	1-104
MxxS.RP	%I*.(yy)0.1	BOOL	;Enable reverse mechanism	1-105
MxxS.AP	%I*.(yy)0.2	BOOL	;Enable advance mechanism	1-105
MxxS.LP	%I*.(yy)0.3	BOOL	;Lock mechanism	1-105
MxxS.F4	%I*.(yy)0.4	BOOL	;not assigned, 2#0	
MxxS.F5	%I*.(yy)0.5	BOOL	;not assigned, 2#0	
MxxS.F6	%I*.(yy)0.6	BOOL	;not assigned, 2#0	
MxxS.F7	%I*.(yy)0.7	BOOL	;not assigned, 2#0	
MxxS.PRGNR	%IB*.(yy+1)	USINT	;Program number NC PLC	1-104
MxxS.PROC	%IW*.(yy+2)	INT	;Mechanism number	1-105

Fig. 1-10: Mechanism status signals (NC => PLC)

1.3 Process Signals

The signals described in this Chapter exist once for each process. This permits the individual processes to be used independently of each other. It is possible, for example, to execute an NC program in one process while another process is in jog mode.

Symbolic operands for process and mechanism signals have been defined to make PLC programs clearer.

Explanation of the symbolic operands for a process or mechanism:

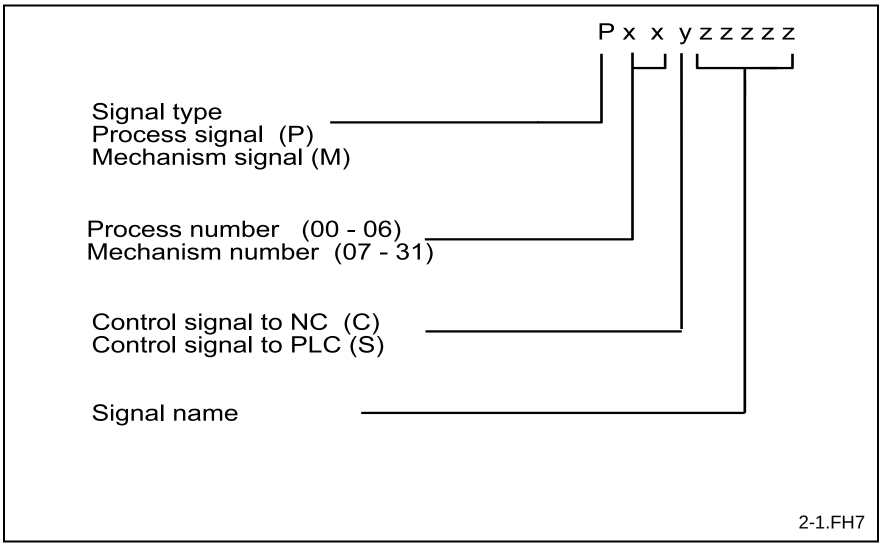


Fig. 1-11: Explanation of symbolic operands

Monitoring

During the power-up phase of a machine and in an emergency shutdown during operation, the corresponding signal inputs must be checked in the correct sequence and with the necessary cross references. This is the only way to unambiguously determine the cause of an emergency shutdown, or the safety device that prevents input power to be activated.

An incomplete or unordered interrogation of error states leads to incorrect diagnoses and to unnecessarily long machine standstill times.

This is why the functionality of monitoring the power and EMERGENCY STOP chain has been integrated in the NC. This ensures that all possible error states are scanned and displayed with the correct priority. It is now within the programmer's responsibility to ensure that the states that are required for the monitoring function are correctly scanned and transferred to the NC.

Note: Correct operation of this monitoring function requires the power and EMERGENCY STOP chain to be wired according to the regulations of the NC Installation Instructions.

The following signals are interpreted in any mode.

Overview of signal assignments for the power and EMERGENCY STOP chain:

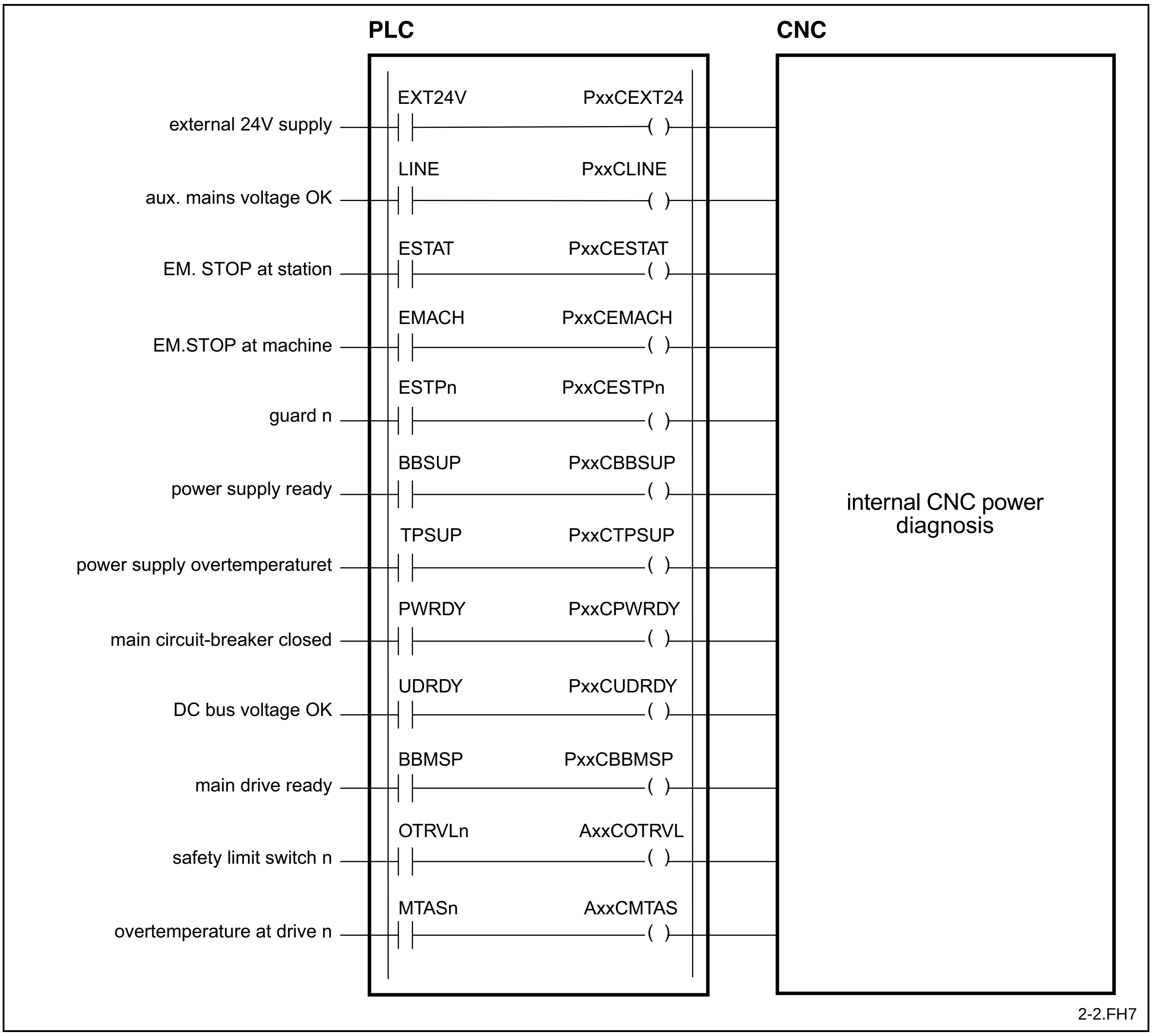


Fig. 1-12: Overview of signal assignments for the power and EMERGENCY STOP chain:

External 24 volt power supply "PxxC.EXT24"

Designation **PxxC.EXT24** = **P***rocess* **xx** **C***ommand* **Ext***ernal* **24** *V Supply*

Control signal PLC →NC

PxxC.EXT24 = 0: no external 24 Volt supply
PxxC.EXT24 = 1: external 24 Volt supply ok

Function The PLC employs this signal to indicate to the NC whether or not the power supply to the inputs and outputs of the station exists.

For this purpose, the station's 24 V power supply is made available as an input signal and passed on to the NC within the PLC program.

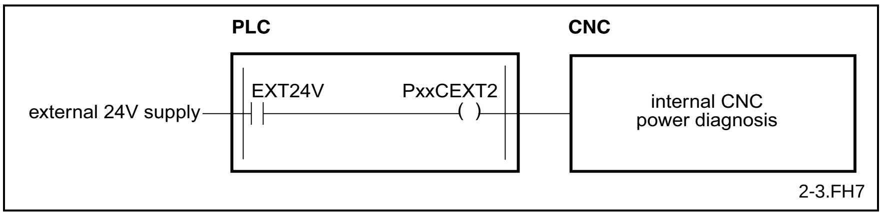


Fig. 1-13: Signal “Supply voltage”

If several 24 V power supplies exist in a station, one input signal must be provided for each of them. The AND operation for all inputs is then reported to the NC.

If no input signal is available for the external 24 V DC power supply, the PLC program must assign a logic "1" to the control signal.

Auxiliary line voltage "PxxC.LINE"

Designation **PxxC.LINE** = **P**rocess **xx** **C**ommand **Line** Control Voltage

Control signal PLC →NC

PxxC.LINE = 0: No auxiliary line voltage

PxxC.LINE = 1: Auxiliary line voltage ok

Function This signal reports to the NC whether or not the "**auxiliary line voltage**" of the drive package is present. Power supply module and drive amplifiers are only able to supply valid status and/or error messages if the auxiliary line voltage is available.

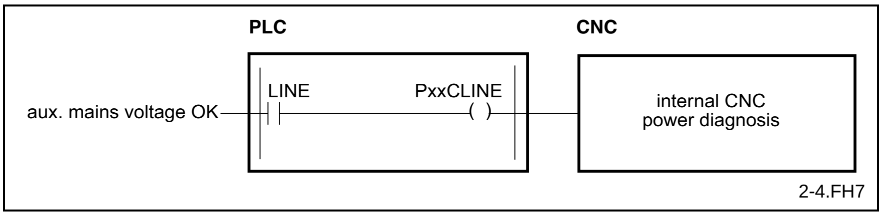


Fig. 1-14: Signal “Auxiliary line voltage voltage”

If there is no input signal for the auxiliary line voltage available, the control signal must either statically be set to "1" or be interconnected with the monitoring signal "external 24 VDC power supply".

Note: The signal is made available if a KDV is employed. It is solely supported by the KDV and exists there as an output.

Station EMERGENCY STOP "PxxC.ESTAT"

Designation **PxxC.ESTAT** = **P***rocess* **xx** **C***ommand* **E***mergency* **Stat***ion*

Control signal PLC →NC

PxxC.ESTAT = 0: EMERGENCY STOP loop opened by pushbutton at the station

PxxC.ESTAT = 1: EMERGENCY STOP loop closed by pushbutton at the station

Function This signal tells the NC whether or not the EMERGENCY STOP loop of the station has been opened by pressing the EMERGENCY STOP pushbutton. If several EMERGENCY STOP pushbuttons exist in a station and can be used for interrupting the EMERGENCY STOP loop, the status of the last pushbutton in the loop must be reported.

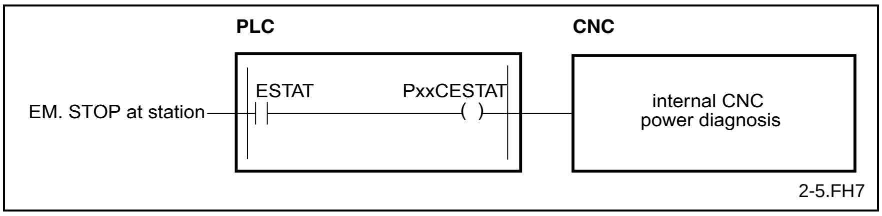


Fig. 1-15: Signal “EMERGENCY STOP station”

If no input signal is available for "EMERGENCY STOP station", the control signal must either statically be set to "1" or be interconnected with the "auxiliary line voltage" monitoring signal.

EMERGENCY STOP machine "PxxC.EMACH"

Designation **PxxC.EMACH** = **P**rocess **xx** **C***ommand* **E***mergency* **Mach***ine*

Control signal PLC →NC

PxxC.EMACH = 0: EMERGENCY STOP loop opened by EMERGENCY STOP machine in the system

PxxC.EMACH = 1: EMERGENCY STOP loop closed by EMERGENCY STOP machine in the system

Function Special safety devices (e.g. pull-wires) permit the power to be shut down simultaneously throughout the machine.

In the EMERGENCY STOP loop of each station there has to be a contact connected downstream from the station's EMERGENCY STOP pushbutton.

This contact's signal line must be fed via the PLC directly to the "PxxC.EMACH" control signal.

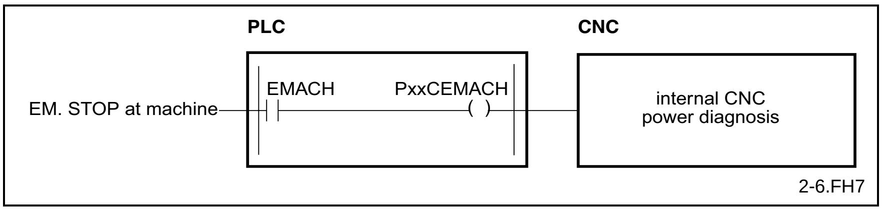


Fig. 1-16: Signal “EMERGENCY STOP machine”

If no input signal is available for "EMERGENCY STOP at system", the control signal must either statically be set to "1" or be interconnected with the "EMERGENCY STOP at station" monitoring signal.

Protective devices "PxxC.ESTPn"

Designation

PxxC.ESTP = **P**rocess **xx C***ommand* **E***mergency* **Stop**

3 control signals PLC →NC

PxxC.ESTP1	= 0:	Protective device 1 opened
PxxC.ESTP1	= 1:	Protective device 1 closed
PxxC.ESTP2	= 0:	Protective device 2 opened
PxxC.ESTP2	= 1:	Protective device 2 closed
PxxC.ESTP3	= 0:	Protective device 3 opened
PxxC.ESTP3	= 1:	Protective device 3 closed

Function

The NC monitoring function permits three additional contacts to be used in addition to pushbuttons for interrupting the station's EMERGENCY STOP loop. The guard function can be triggered by protective screens, light barriers, or proximity switches, for example.

To be able to produce a correct diagnosis, the signal lines must be connected to the NC control signals in the correct sequence.

Note: The general message:
"Protective device not closed" can be defined in more detail within the work space (e.g.: "Left guard rail open" or "Photoelectric barrier tooling interrupted").

WARNING

Opening a guard should <u>only</u> shut down the power of the station in question.

If an open guard interrupts the EMERGENCY STOP loop of the entire system this will always result in a wrong diagnosis.

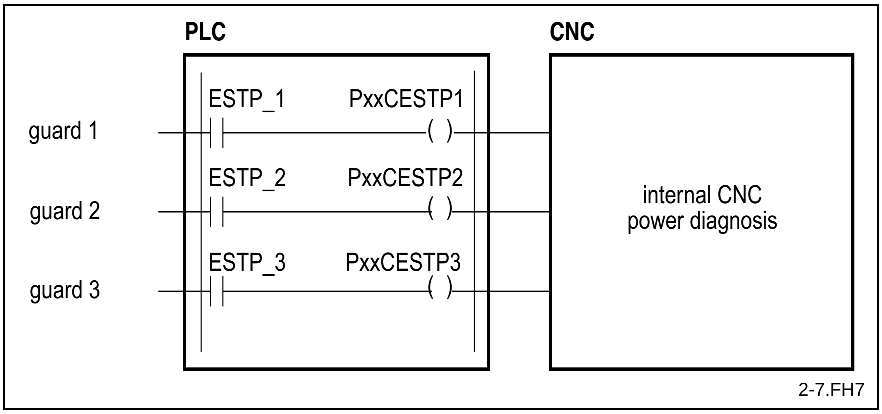


Fig. 1-17: Signal "Protective device"

If none or not all of the guards at the station are used, the related control signals must either statically be set to "1" or be interconnected with the next higher monitoring signals (guard 1 with EMERGENCY STOP at system, guard 2 with guard 1, guard 3 with guard 2).

Power supply unit operational "PxxC.BBSUP"

Designation

PxxC.BBSUP = **P***rocess* **xx C***ommand* **BB Sup***ply*

Control signal PLC →NC

PxxC.BBSUP = 0: Power supply unit is not operational

PxxC.BBSUP = 1: Power supply unit is operational

Function This signal is used to tell the NC that the "power supply unit" of this station is ready for operation. Power can only be activated when the power supply unit is ready. The "BB1 contact" of the power supply unit must be used for this function.

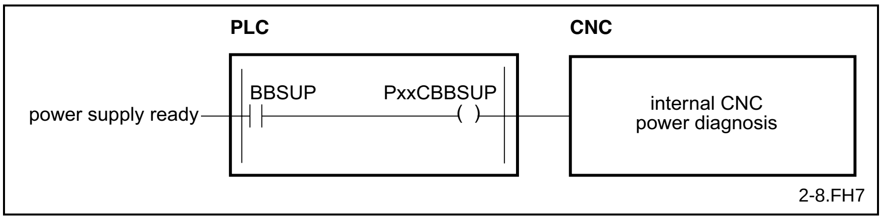


Fig. 1-18: Signal “Power supply unit”

If no input signal is available for an operational power supply unit, this control signal must either statically be set to “1” or be interconnected with the “guard 3” monitoring signal.

Temperature control power supply unit "PxxC.TPSUP"

Designation **PxxC.TPSUP** = **P**rocess **xx** **C**ommand **T**em**p**erature Control **Sup**ply

Control signal PLC →NC

PxxC.TPSUP = 0: Overtemperature at the power supply unit

PxxC.TPSUP = 1: Temperature at power supply unit is normal

Function This control signal reports an excessively high power supply unit temperature to the NC. An excessively high temperature in the power supply unit leads to a power shutdown. Power can be switched back on only after the power supply unit has cooled down.

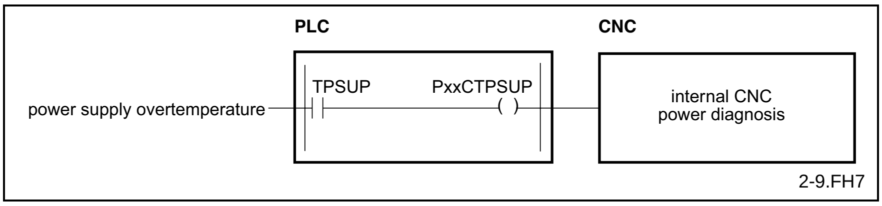


Fig. 1-19: Signal “Overtemperature at the power supply unit”

If no input signal is available for “power supply overtemperature”, the control signal must either statically be set to “1” or be interconnected with the “power supply unit ready” monitoring signal.

Main contactor has closed “PxxC.PWRDY”

Designation **PxxC.PWRDY** = **P***rocess* **xx** **C***ommand* **P***ower* **Re**a**dy**

Control signal PLC →NC

PxxC.PWRDY = 0: Main contactor has not closed

PxxC.PWRDY = 1: Main contactor has closed

Function This signal reports to the NC whether or not the main contactor has closed after the NC has issued the power activation command. This requires an auxiliary contact of the main contactor to be connected via an input signal.

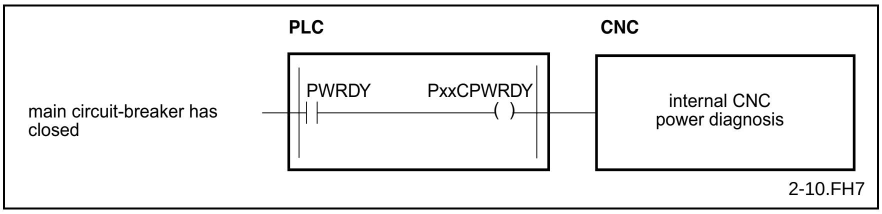


Fig. 1-20: Signal “Main contactor has closed”

If no input signal is available for “Main contactor has closed", the control signal must either statically be set to “1” or be interconnected with the “power supply unit temperature monitoring” monitoring signal.

DC bus voltage OK “PxxC.UDRDY”

Designation

Designation: **PxxC.UDRDY** = **P**rocess **xx C**ommand **UD Re**a**dy**

Control signal PLC →NC

PxxC.UDRDY = 0: DC bus voltage not okay

PxxC.UDRDY = 1: DC bus voltage okay

Function

After power has been switched on, this signal tells the NC whether or not the DC bus voltage has been built up.

This requires the "UD contact" of the power supply unit or the "BB2 contact" of the drive amplifier to be reported to this control signal.

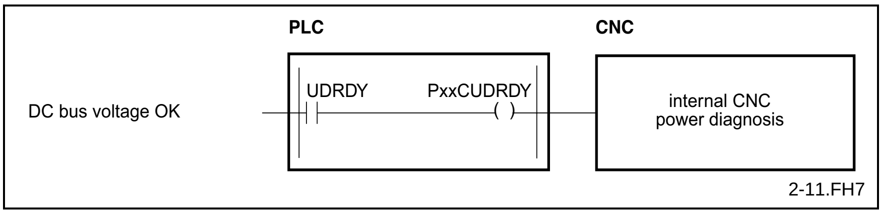


Fig. 1-21: Signal “DC bus voltage”

If no input signal is available for “DC bus voltage”, the control signal must either statically be set to “1” or be interconnected with the “main contactor has closed” monitoring signal.

Main spindle ready “PxxC.BBMSP”

Designation

PxxC.BBMSP = **P***rocess* **xx C***ommand* **BB M***ain* **Sp***indle*

Control signal PLC →NC

PxxC.BBMSP = 0: Main spindle is not ready

PxxC.BBMSP = 1: Main spindle is ready

Function

This control signal is used to tell the NC whether or not the main spindle is ready for operation.

If the station does not possess a main spindle, the signal must statically be set to "1" within the PLC program.

This signal must be simulated if axis movements are to be permitted in certain modes without the main spindle being ready for operation (e.g. test mode without spindle or retracting the axes in jog mode).

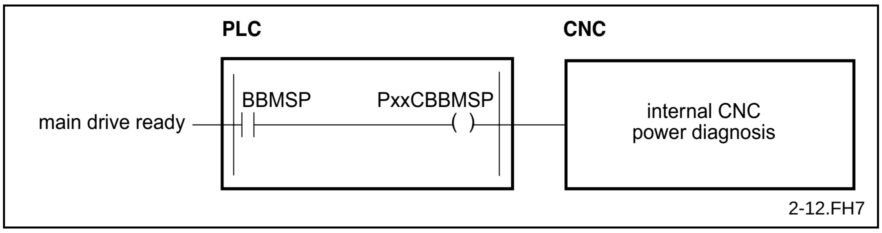


Fig. 1-22: Signal “Main spindle ready”

If no input signal is available for "main drive ready", this control signal must either statically be set to "1" or be interconnected with the "DC bus voltage" monitoring signal.

Power Activation

There is a separate interface to “power activation” for each CNC process. This enables individual stations of a machine to be shut down while work is continued at other stations. Please refer to the *“Technical Documentation”* for an exact description of the power activation wiring.

Using "process xx" as an example , the following figure shows the signal flow required for activating the power to a process:

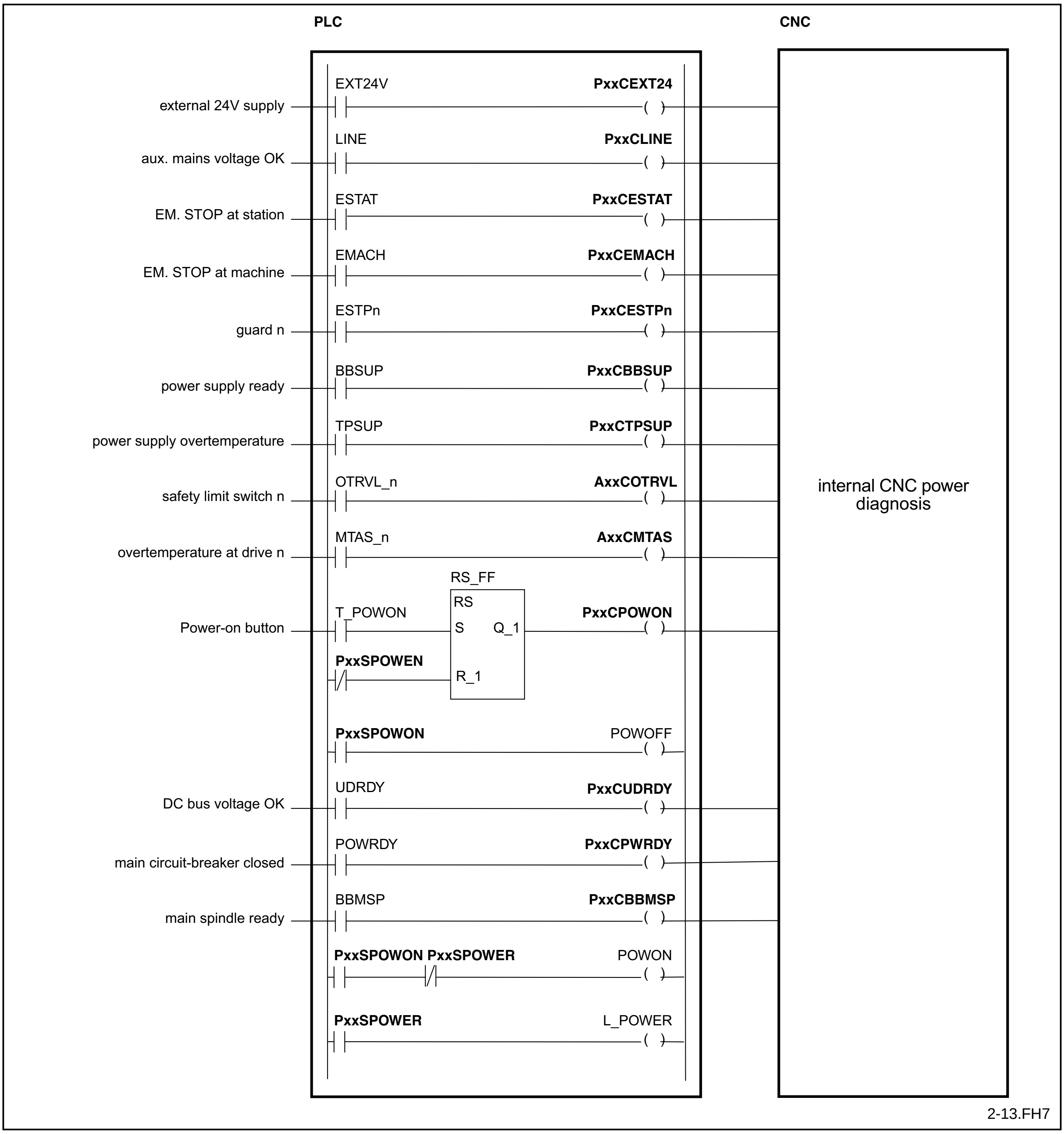


Fig. 1-23: Signal flow for power activation

Note: The interface signals between the NC and the PLC are shown in Fig. 1-23: Signal flow for power activation in bold print.

Legend:
POWOFF: Signal for activation of the main contactor
POWON: corresponds to the charging contactor
L_POWON: Signal lamp

Power signal processing sequence upon power activation:

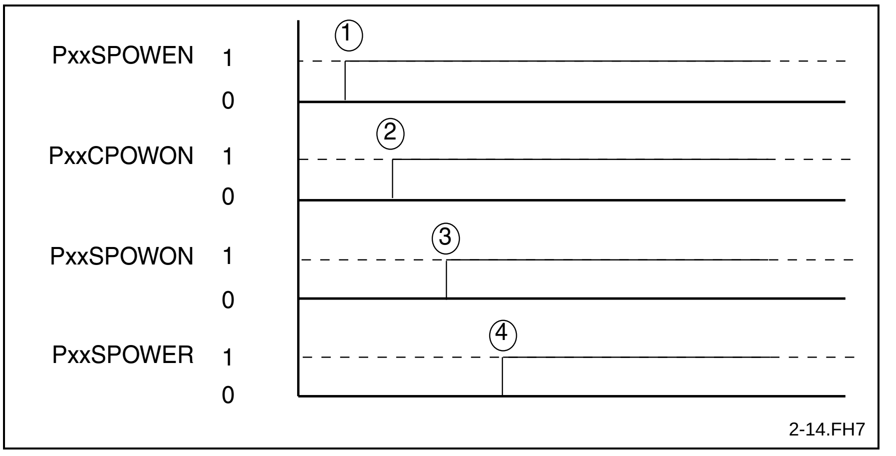


Fig. 1-24: Power signal processing sequence

(1) The NC issues the "power enabled" (PxxS.POWEN) signal once all the essential power activation conditions have been satisfied and reported to the NC.

(2) The PLC issues the "PxxC.POWON" signal to request power activation. This request may be triggered by a pushbutton, for example, that is used for switching on the power.

(3) If power activation is enabled by the NC and the PLC subsequently requests power through the "PxxC.POWON" signal, the NC issues the "power on" signal (PxxS.POWON).

(4) The NC issues the "power present" (PxxS.POWER) signal once all power feedbacks from the PLC have been reported to the NC. Executing NC programs or moving axes manually is only possible after the NC has issued that signal.

Example Printout of an PLC program sequence for power activation

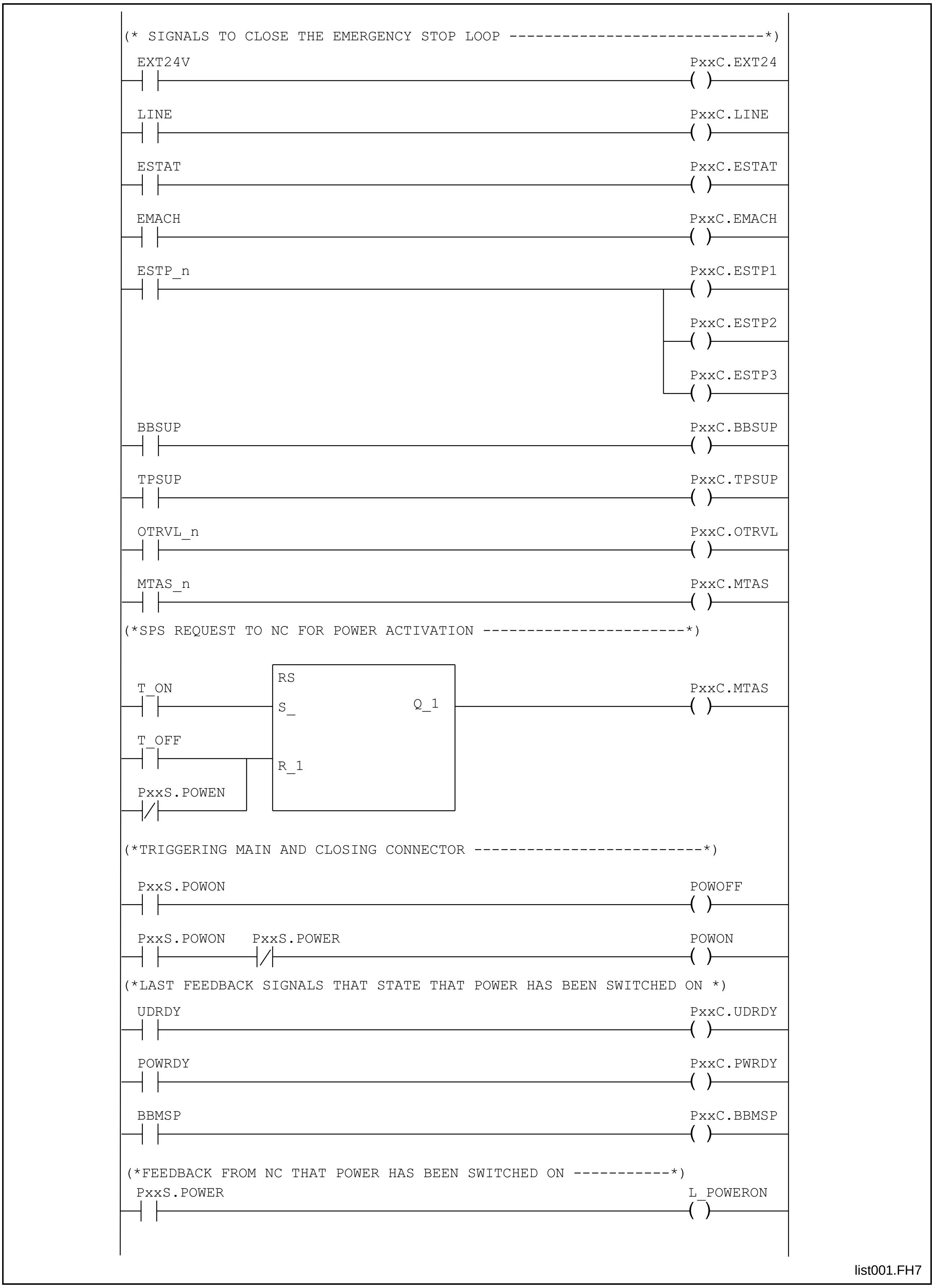


Fig. 1-25:Printout of a PLC program sequence for power activation

Power enable signal "PxxS.POWEN"

Designation

PxxS.POWEN = **P**rocess **xx** **S***tatus* **Pow***er* **En***able*

Status signal NC →PLC

PxxS.POWEN = 0: No power enabling signal from the NC

PxxS.POWEN = 1: Power enabling signal from the NC

The signal is valid in all modes.

Function

The NC employs this signal to tell the PLC that all conditions for power activation have been fulfilled for this station.

The "Power enabled" signal is issued only if the following conditions are fulfilled:

- There is no NC error pending for the process which prevents power from being switched on (PxxS.ERROR=0)
- The 24-V supply to the process is OK (PxxC.EXT24=1),
- The auxiliary mains voltage is present (PxxC.LINE=1)
- The EMERGENCY STOP button of the station has not been pressed (PxxC.ESTAT=1)
- There is no EMERGENCY STOP condition for the system (PxxC.EMACH=1)
- None of the three possible guards is open (PxxC.ESTP1=1, PxxC.ESTP2=1, PxxC.ESTP3=1)
- No limit switch of this process has been actuated (all AxxC.OTRVL=1, xx = axis number)
- The power supply unit is ready for operation (PxxC.BBSUP=1)
- The temperature of the power supply unit is not excessively high (PxxC.TPSUP=1); none of the axes that are involved in the process is excessively hot (all AxxC.MTAS=1, xx = axis number)

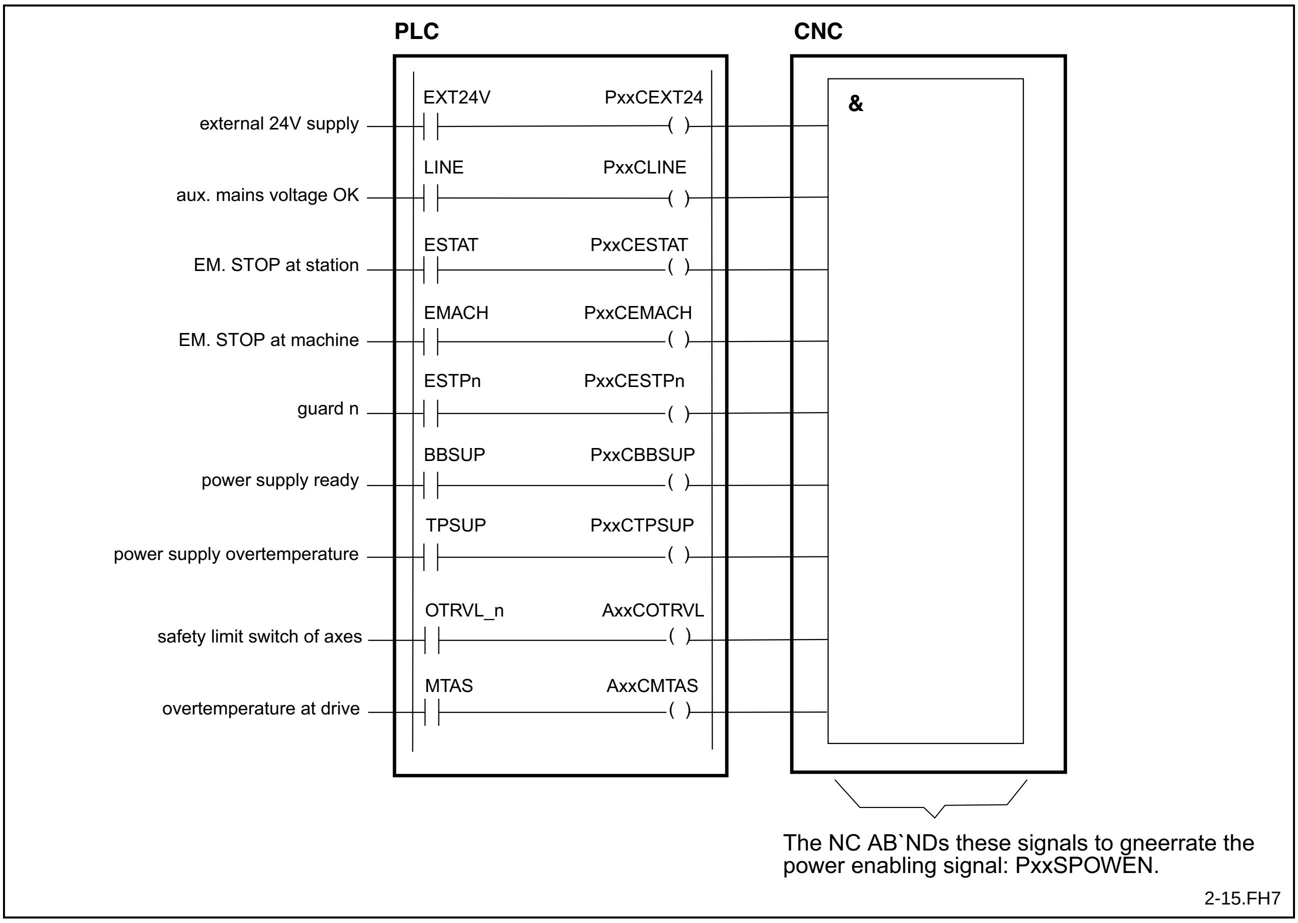


Fig. 1-26: Signal “Power enabled”

The "Power enabled" signal is not output if one of the above-mentioned conditions is not satisfied. The NC station diagnosis will then display the first missing condition.

Power request "PxxC.POWON"

Designation **PxxC.POWON** = **P**rocess **xx C***ommand* **Pow***er* **on**

Control signal PLC →NC

PxxC.POWON = 0: No power request

PxxC.POWON = 1: Power requested by the PLC

Function The PLC employs this control signal to request the NC to activate the power for the station. The power request will be ignored as long as the NC does not issue the "power enabled" (PxxS.POWEN) status signal. Otherwise, the NC issues the “power on” (PxxS.POWON) signal to acknowledge the “power request”.

The power request signal can be activated by pressing the relevant button at the station. In automatic mode, the power request signal may be supplied by central power activation on the plant's main control panel.

If power activation is to depend on additional external conditions (e.g. hydraulics system), those conditions must logically be linked with the power request.

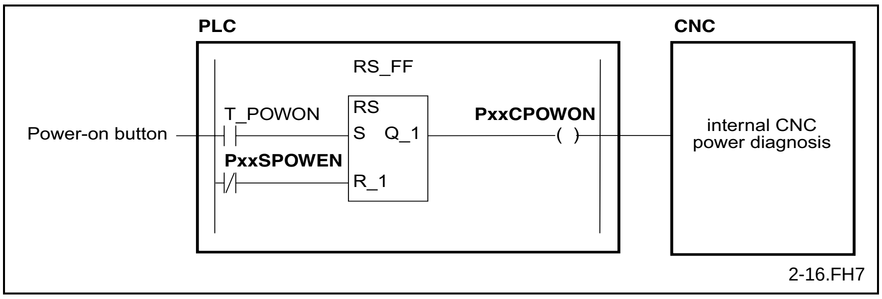


Fig. 1-27: Signal “Power request”

Power on “PxxS.POWON”

Designation **PxxS.POWON** = **P**rocess **xx** **S**tatus **Pow**er **on** is Enable

Status signal NC →PLC

PxxS.POWON = 0: Power on remains not enabled

PxxS.POWON = 1: Power on is enabled

Function The NC employs this signal to request the PLC to close the last contact in the otherwise closed EMERGENCY STOP loop of the station.

This signal must always be connected directly and without any further interconnections to the output that activates the power to the station.

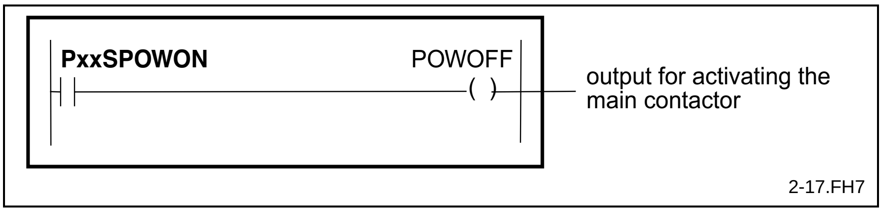


Fig. 1-28: Signal “Power on enable”

Power is on “PxxS.POWER”

Designation **PxxS.POWER** = **P**rocess **xx** **S***tatus* **Power** *is on*

Status signal NC →PLC

PxxS.POWER = 0: Power on has not yet been enabled

PxxS.POWER = 1: Power is on

Function The NC employs this signal to report that all signals are present that must be reported after power activation. If, in addition, all the general enabling signals exist, the NC will be able to execute programs for the station in question or the axes may be moved manually.

The "Power ON" signal is issued only if the following conditions exist:

- The NC reports "power ON" to the PLC (PxxS.POWON=1)
- The main contactor is closed (PxxC.PWRDY=1)
- The DC bus voltage is available (PxxC.UDRDY=1)
- The main spindle is operational (PxxC.BBMSP=1)
- Power enabled is OK (PxxS.POWEN=1)

Rexroth
Bosch Group

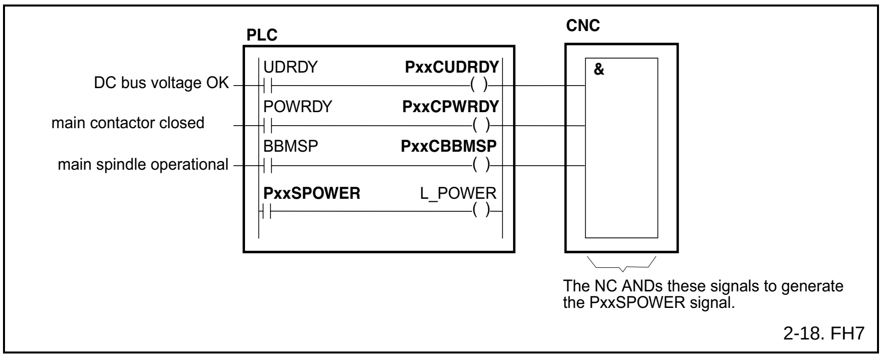


Fig. 1-29: Signal “Power is enabled”

The "Power ON" signal is not issued if one of the above conditions is not satisfied. In this case, the NC station diagnosis shows the first missing condition in the list.

Operating modes

The NC functions are executed under various modes within the control system.

A distinction is drawn between the following modes:

- Automatic mode,
- semi-automatic mode,
- setup mode and
- automatic parameter test

Various process or axis signals are active in certain operating modes only, or change their method of operation as the mode changes. This is true, for example, for the signals "start advance program" (PxxC.ADV) and "start reverse program" (PxxC.REV), or for the signals "positive jogging" (AxxC.JGPOS) and "negative jogging" (AxxC.JGNEG).

If a signal depends on the mode, attention will be drawn to this in the description.

Mode selection "PxxC.MODEn"

Designation

PxxC.MODEn = **P**rocess **xx** **C***ommand* **Mode** *Bit* **n**

2 control signals PLC →NC

PxxC.MODE0

PxxC.MODE1

Function

The two mode selection inputs permit the following modes to be distinguished:

- Automatic mode,
- semi-automatic mode,
- setup mode and
- automatic parameter test

Interpretation of the control signals between the PLC and NC depends on the selected mode.

Changing modes during program execution leads to an immediate stop of the currently executing NC program.

Operating modes	PxxC.MODE0	PxxC.MODE1
Automatic mode	0	0
Setup mode	0	1
Semi-automatic mode	1	0
Automatic parameter test	1	1

Fig. 1-30: Mode coding

"Automatic" mode

"Automatic mode" permits only NC programs to be executed. Moving axes by jog commands is not possible.

Advance program:

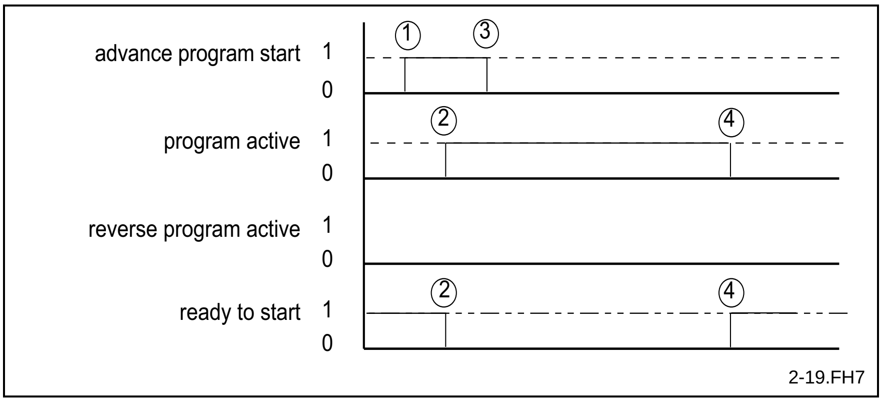


Fig. 1-31: Advance program

(1) The NC advance program is started by a "start advance program" (PxxC.ADV) signal.

(2) The "PROGRAM ACTIVE" (PxxS.ACTIV) signal indicates the execution of the NC advance program. The “reverse program active” (PxxS.REV)remains cleared while an advance program is executed. The "ready to start" (PxxS.READY) signal is canceled when advance program execution starts.

(3) In "Automatic" mode, one impulse is sufficient for starting the NC advance program. The start signal may be cleared once the NC program has started to be executed. The NC program will be completed even if the advance program start signal is no longer applied.

(4) The ”program active” signal (PxxS.ACTIV) is cleared when the NC advance program is completed. At the same time, the subsequent NC advance program will have the "ready to start" (PxxS.READY) signal assigned.

Reverse program:

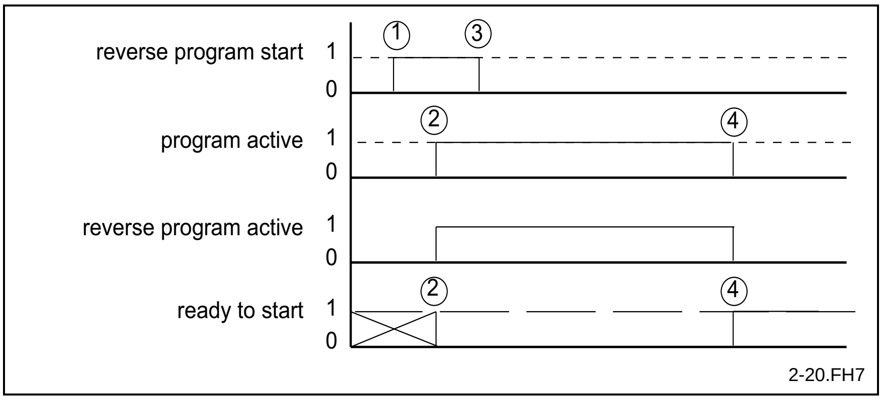


Fig. 1-32: Reverse program

(1) The NC reverse program is started by a "start reverse program" (PxxC.REV) signal.

(2) The "PROGRAM ACTIVE (PxxS.ACTIV) and “reverse program active” (PxxS.REV) signals indicate the execution of the NC reverse program. The "ready to start" (PxxS.READY) signal is canceled at the same time.

(3) In "Automatic" mode, one impulse is sufficient for starting the NC reverse program. The start signal may be cleared once the NC program has started to be executed. The NC program will be completed even if the reverse program start signal is no longer applied.

(4) The “program active” (PxxS.ACTIV) and “reverse program active” (PxxS.REV) signals are cleared when the NC reverse program is terminated. An NC advance program will have the "ready to start" (PxxS.READY) signal assigned if reference exists for all the axes and magazines after the reverse program has been executed.

"Semi-automatic" mode

"Semi-automatic" mode permits only NC programs to be executed. Moving axes by jog commands is not possible.

Advance program:

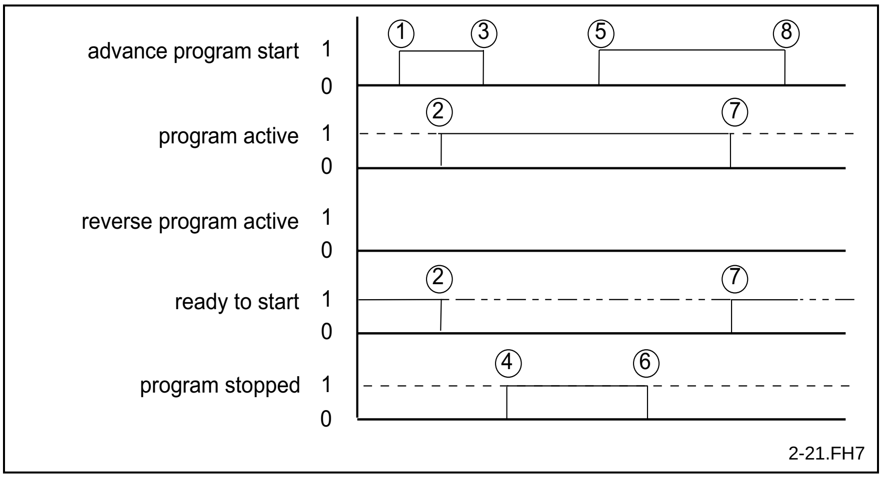


Fig. 1-33: Advance program

(1) The NC advance program is started by a "start advance program" (PxxC.ADV) signal.

(2) The "PROGRAM ACTIVE" (PxxS.ACTIV) signal indicates the execution of the NC advance program. The “reverse program active” (PxxS.REV) remains cleared while an advance program is executed. The "ready to start" (PxxS.READY) signal is canceled when advance program execution starts.

(3) In “semi-automatic” mode, the NC advance program is executed as long as the advance program start signal is applied.

(4) Axes and NC program will be stopped immediately if the start advance signal is removed while the NC advance program is being executed. The interruption is indicated by the “program stopped” (PxxS.STOP) signal.

(5) A new start advance program signal permits the NC advance program to be resumed from the point of interruption.

(6) The restart of the advance program clears the “program stopped” (PxxS.STOP) signal.

(7) The "Program active" signal is cleared when the NC advance program is completed. At the same time, the "Ready to start" signal is assigned for a new advance program.

(8) The start signal may be cleared when the advance program is completed.

Reverse program:

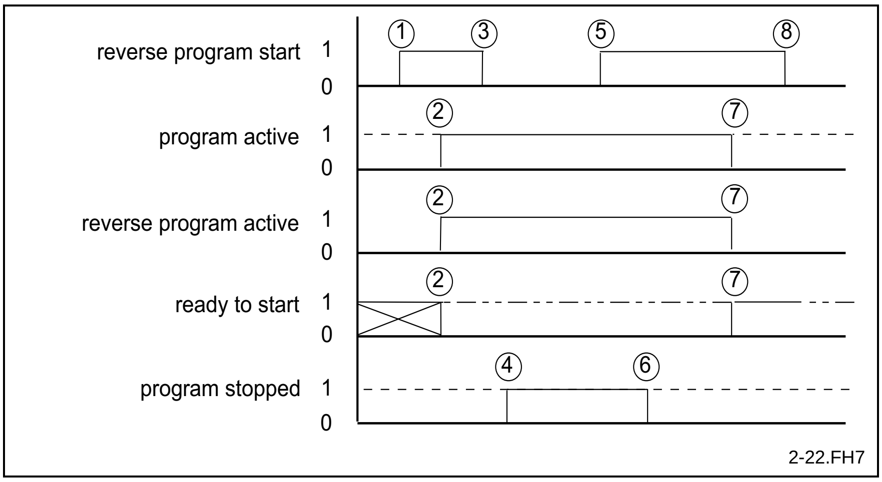


Fig. 1-34: Reverse program

(1) The NC reverse program is started by a "start reverse program" (PxxC.REV) signal.

(2) The “program active” (PxxS.ACTIV) and “reverse program active” (PxxS.REV) signals indicate the execution of the NC reverse program. The “ready to start” signal (PxxS.READY) is canceled at the same time.

(3) In “semi-automatic” mode, the NC reverse program is executed as long as the reverse program start signal is applied.

(4) Axes and NC program will be stopped immediately if the start reverse program signal is removed while the NC reverse program is being executed. The interruption is indicated by the “program stopped” (PxxS.STOP) signal.

(5) A new Reverse program start signal permits the NC reverse program to be resumed from the point of interruption.

(6) The restart of the reverse program clears the “program stopped” (PxxS.STOP) signal.

(7) The "Program active" signal is cleared when the NC reverse program is completed. At the same time, the "Ready to start" signal is assigned for an advance program start.

(8) The start signal may be cleared when the reverse program is completed.

"Setup" mode

"Setup" mode permits NC programs to be executed and axes to be moved by jog commands.

NC program operation is the same as in "Semi-automatic" mode.

The same applies for the execution of NC programs as in "semi-automatic" mode.

“Automatic parameter test mode”

“Automatic parameter test” mode permits optimum values to be determined for:

- the maximum axis velocity,
- the maximum sudden axis velocity change, and
- the maximum axis acceleration

The determined data can be represented in tabular form or in a graphic diagram.

Timing of the 'automatic parameter test':

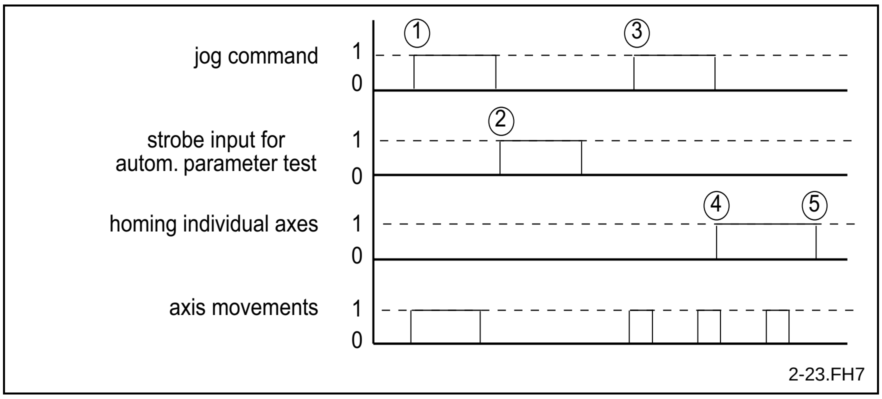


Fig. 1-35: Timing of the "automatic parameter test“

(1) Automatic parameter test is performed within a path the operator approaches by jog movements. Setting the “positive jogging” (AxxC.JGPOS) or “negative jogging” (AxxC.JGNEG) signals moves the axis for which the automatic parameter test is to be performed to the first position.

(2) Setting the “transfer strobe for automatic parameter test” (PxxC.PARAM) signal saves the current positions of all the axes. The axis positions are the starting point of the automatic parameter test.

(3) The operator uses jog commands to move the axes to the next position. The automatic parameter test will be performed between these two positions.

(4) The “single axis homing” (AxxC.HOME) signal starts the test. The specified travel range is traversed repeatedly for this purpose. Removing the “single axis homing” signal aborts the automatic parameter test execution.

(5) Once the automatic parameter test has been completed, the measured results are available to the operator and may be used for setting the parameter values of the axes.

If the test for automatic axis parameter determination is used for several axes at the same time, the axes will be processed sequentially. In other words, the test will only be performed for one axis at a time.

Selected operation mode "PxxS.MODEn"

Designation: PxxS.MODEn = Process xx Status Mode *Bit* n

2 Status signals NC →PLC

PxxS.MODE0

PxxS.MODE1

Function: The two mode selection NC outputs permit the modes to be displayed:

- Automatic mode
- Semi-automatic mode
- Setup mode
- Automatic parameter test

The PxxS.MODEn signals indicate for the PLC the operating mode of the NC control unit. The signals MODE0 and MODE1 correlate in the PxxC and PxxS structures. Depending on the selected operating mode, the status signals PxxS.MODEn are set according to the following table.

Operating modes	PxxS.MODE0	PxxS.MODE1
Automatic mode	0	0
Setup mode	0	1
Semi-automatic mode	1	0
Automatic parameter test	1	1

Fig. 1-36: Coding of operating modes

Enhanced jog mode

“Enhanced jog mode” permits axes to be jogged in an active process (an active NC program that has been stopped) without performing a reset.

To activate this response, the "manual axis jogging causes reset" process parameter must be set to “No”.

If the parameter is set to “No” the following peculiarities must be observed:

- Any pending auxiliary function that has not yet been acknowledged when jogging is started will be aborted. Upon a restart, the incomplete auxiliary function will again be output to the PLC. An auxiliary function that may not be output upon a restart must be programmed in a separate NC block.
- Non-modal G functions are aborted upon the first jog and set again upon the restart (provided the current block number has not been changed).
- Tool change commands must be programmed in a separate block to ensure that they will not be processed again upon restart.
- Tool storage axes and all other feed axes will be stopped upon a changeover to manual mode. Upon restart, the NC will complete the still pending tool storage command.
- Pending process control commands are not reset, for example: DP, SP, RP, AP, WP, LP, POK
- Event monitoring functions are de-activated upon the first jog command. The NC reactivates event monitoring when a restart is performed at the last valid NC block.
- Reverse vectors are de-activated during jogging but are re-activated upon a restart.
- The "PxxS.RUN" and "PxxS.STOP" program status signals are cleared. The "PxxS.ACTIV" and "PxxS.REV" signals are retained.
- Any active "travel to dead stop" function will be reset when the axes are jogged.

Jog mode "PxxC.JOGMn"

Designation **PxxC.JOGMn** = **P**rocess **xx** **C**ommand **Jog**ging **M**ode Bit **n**

3 control signals PLC →NC

PxxC.JOGM0
PxxC.JOGM1
PxxC.JOGM2

Function *“Automatic” mode*

”Semi-automatic” mode

”Jogging” is not possible in these two modes. The NC does not interpret the jog signals.

“Setup”mode

”Jog mode” defines the type of jog movement. A distinction is made between moving over a fixed distance and continuous jogging.

- With continuous jogging, the axis is moved as long as the jog command is applied.
- Jogging a fixed distance means that the axis movement is initiated with the positive flank of the jog command and continued until the full distance has been covered.

The jog command is no longer required during the movement.

Changing the jog mode while an axis is being jogged interrupts the movement of the axis.

The following jog resolutions are available, depending on the number of "programmable decimal places for the distance" entered in the process parameters.

PxxC.JOGM2	PxxC.JOGM1	PxxC.JOGM0	Jog resolution
0	0	0	Continuous jogging
0	0	1	Jogging 1 AE ≙ 0.0001 mm
0	1	0	Jogging 10 AE ≙ 0.001 mm
0	1	1	Jogging 100 AE ≙ 0.01 mm
1	0	0	Jogging 1000 AE ≙ 0.1 mm
1	0	1	Jogging 10000 AE ≙ 1 mm
1	1	0	Jogging 100000 AE ≙ 10 mm
1	1	1	Jog distance from parameter

Fig. 1-37: Programmable decimal places for the distance: 4

PxxC.JOGM2	PxxC.JOGM1	PxxC.JOGM0	Jog resolution
0	0	0	Continuous jogging
0	0	1	Jogging 1 AE ≙ 0.00001 mm
0	1	0	Jogging 10 AE ≙ 0.0001 mm
0	1	1	Jogging 100 AE ≙ 0.001 mm
1	0	0	Jogging 1000 AE ≙ 0.01 mm
1	0	1	Jogging 10000 AE ≙ 0.1 mm
1	1	0	Jogging 100000 AE ≙ 1 mm
1	1	1	Jog distance from parameter

Fig. 1-38: Programmable decimal places for the distance: 5

Note: When jog distance is active, the “jog spindle position” axis parameter is drawn on to jog the spindle. Positive jogging means that the positive position is approached absolutely. Negative jogging means that the negative position is approached.

Single step mode "PxxC.SINGL"

Designation

PxxC.SINGL = **P**rocess **xx** **C**ommand **Singl**e Stop

Control signal PLC →NC

PxxC.SINGL = 0: Cycle mode

PxxC.SINGL = 1: Single step mode

Function

When the “single step mode” signal is set, a stop is performed once a block has been processed. If the signal is set during program execution, an immediate stop is initiated when the next block is active. Otherwise, the execution of the current block is completed and the stop occurs at the end of the block. The “stop” status signal is not set.

The NC clears the "Block active" process status signal between the execution of two blocks. The next block in the advance program is initiated with the "Start advance program" signal. The next block in the reverse program is initiated with the "Start reverse program" signal.

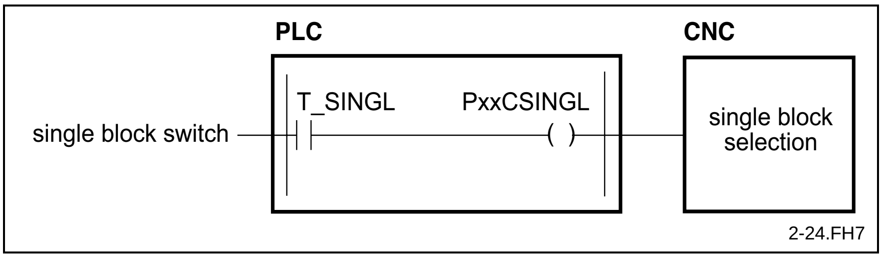


Fig. 1-39: Signal “Single step mode”

Note: An immediate stop is initiated when the next block is active (optimum block transition). Otherwise, the current block is completed, and the stop occurs at the end of the block.

Rapid traverse mode "PxxC.RAPID"

Designation PxxC.RAPID = Process xx C*ommand* Rapid

Control signal PLC →NC

PxxC.RAPID = 0: No rapid traverse rate during jogging

PxxC.RAPID = 1: Rapid traverse rate during jogging

Function *"Automatic" mode*
"Semi-automatic" mode
In these modes, the "Rapid traverse mode" signal is not interpreted.

“Setup” mode

”Automatic parameter test” mode

When the “rapid traverse” signal is set, the rapid traverse velocity that has been programmed in the axis parameters is used for manual axis movements (jogging).

The rapid traverse signal is valid for all axes that are related to that process, except the spindle axes. The axes are moved at the rapid traverse rate as long as the rapid traverse signal is applied.

Moving an axis at rapid traverse rate requires the measurement reference of that axis to be established.

The measurement reference of that axis must first be established by single-axis homing or by program-controlled homing.

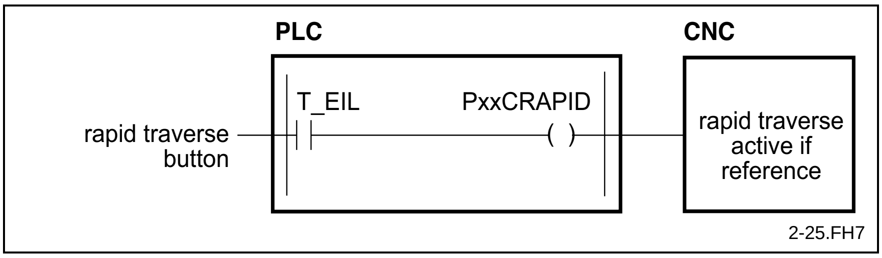


Fig. 1-40: Signal “Rapid traverse mode”

NC program selection

Program number selection “PxxC.PRGNR”

Designation **PxxC.PRGNR** = **P**rocess **xx** **C**ommand **Pro**gram **N**umbe**r**

Data byte PLC →NC

PxxC.PRGNR = 1 ... 99

Function “Program number selection” enables an NC program to be selected via the PLC program for execution. The program is executed upon a program start.

Selecting the program number by the PLC program is only possible for internal processes (P0 - P6).

The program number selection is produced in the PLC program and stored as a byte in the user data memory. For each process, the data byte content is transferred as a program number to the NC. The NC acknowledges the acceptance of the program number selection by preselecting the program number. Preselection is only made if the “PxxC.SP” signal has been set.

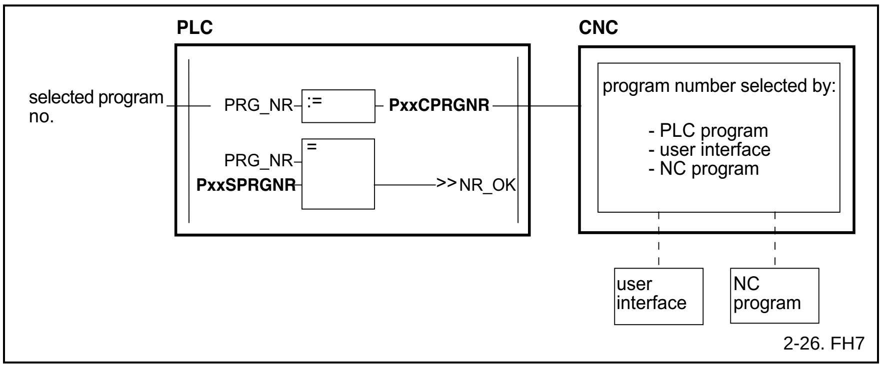


Fig. 1-41: Signal “Selection of program number”

Note: The NC program number can be preselected via the NC program, the user interface, or the PLC. The PLC's preselection only becomes active when the interface signal “PxxC.SP=1”.

NC program number specification "PxxS.PRGNR"

Designation **PxxS.PRGNR** = **P**rocess **xx** **S***tatus* **Pro***gram* **N***umbe***r**

Data byte NC →PLC

PxxS.PRGNR = 1 ... 99

Function *Internal processes (P0 – P6)*
The “NC program number specification”" for internal processes (P0 - P6) is used for confirming the acceptance of a selected program number. The program number selection of the PLC is accepted when the selected and the preselected program numbers are identical.

Note: Acknowledgment of the program number selection by the program number specification only means that the NC has accepted the program number of the PLC.

From version xx.15 onwards, the program number is reported that will be active upon the next NC program start.

Mechanisms (M7 – M31)

The program number specification for mechanisms (M7 - M31) is used in the NC program through the "SPxx:pp" command (xx = mechanism number, pp = program number).

The 'MxxS.PRGNR' signal is used for transferring the program number to the PLC. In the PLC program, the program number must be transferred to the corresponding mechanism.

Accepting NC program number “PxxC.SP”

Designation **PxxC.SP** = **P**rocess **xx C**ommand **S**elect **P**rogram

Control signal PLC →NC

PxxC.SP = 0: No NC program number preselection

PxxC.SP = 1: The PLC preselects the NC program number

Function If this control signal is set, the PLC transmits to the NC the NC program number preselected in “"PxxC.PRGNR”. This signal is effective in all modes.

Note: NC program number preselection from the PC is not possible if the PLC sets this signal to logic "1" in each cycle.

NC Program Control

Process enable "PxxC.ENABL"

Designation **PxxC.ENABL** = **P**rocess **xx C***ommand* **Enabl***e*

Control signal PLC →NC

PxxC.ENABL = 0: No process enable present

PxxC.ENABL = 1: Process enable present

The process enable signal has an effect in all modes.

Function The "Process enable" signal is required for moving an axis or for executing an NC program. Power remains ON when the process enabling signal is removed.

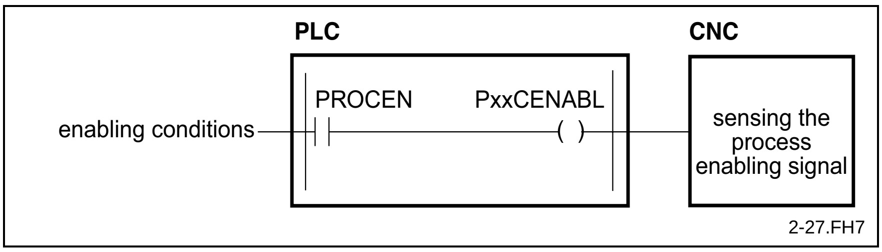


Fig. 1-42: Signal “Process enable”

An immediate stop is initiated for the process if the process enabling signal is removed while an NC program is running. Axis movements are decelerated along a ramp curve, and program execution is interrupted. The NC program of the process can only be started after the process enabling signal has been applied. The “start advance program” signal (PxxC.ADV) can be used for restarting an interrupted advance program.

The “start reverse program” signal (PxxC.REV) can be used for restarting an interrupted reverse program.

If the process enabling signal is removed during a jog movement (single-axis homing, jogging), the jog process is interrupted and axis movement is decelerated along a ramp.

Jog movements can only be continued after the process enabling signal has been reapplied.

NC blocks are skipped "PxxC.BLSKP"

Designation

PxxC.BLSKP = **P**rocess **xx** **C***ommand* **Bl***ock* **Sk***i***p**

Control signal PLC →NC

PxxC.BLSKP	= 0:	NC blocks are not skipped
PxxC.BLSKP	= 1:	NC blocks are skipped

The signal is effective in all modes.

Function

The NC blocks that are marked by an oblique are skipped when the “PxxC.BLSKP” signal is set to “1”. A series of blocks in the NC program will only be skipped in cyclic NC mode (PxxC.SINGL=0) if the signal has transitioned to "1" before the first NC block of the series is processed.

In NC single-block mode (PxxC.SINGL=1), the NC checks whether or not skipping has been activated via the "PxxC.BLSKP=1" signal before it starts processing an NC block. An NC block will not be processed if the signal is "1" before its processing is started.

If the “PxxC.BLSKP” signal is "0", the NC blocks that are marked by an oblique are processed (i.e. not skipped). The NC will only process a series of skipped blocks in cyclic mode (PxxC.SINGL=0) if the signal is "0" before the first NC block of the series is processed. In NC single block mode (PxxC.SINGL = 1), the NC executes all blocks, including the ones with obliques. It only processes a block with an oblique if the related "PxxC.BLSKP" signal is "0" before the execution of the block is started.

Advance program start "PxxC.ADV"

Designation

PxxC.ADV = **P**rocess **xx** **C**ommand **Adv**ance Program Start

Control signal PLC →NC

PxxC.ADV	= 0:	No advance program start
PxxC.ADV	= 1:	Advance program start

This control signal has different meanings in the different modes.

Function

Requirements for the start of the advance program:

- power is available;
- no error,
- the process enabling signal is issued;
- No reverse program active, and
- ready to start.

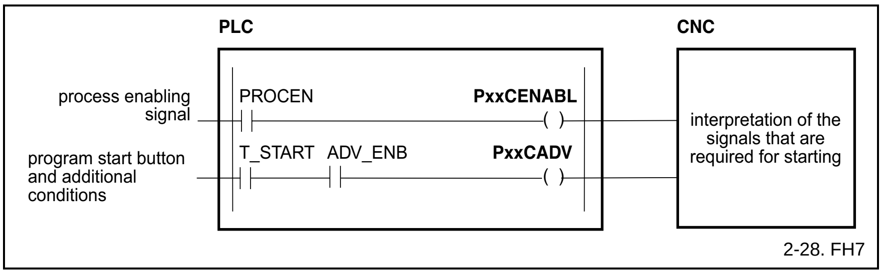


Fig. 1-43: Signal “Advance program start”

“Automatic” mode

A positive edge of the “Advance program start” (PxxC.ADV) signal starts the advance program when the ready to start signal (PxxS.READY=1) has been issued. The "Advance program start" signal can be removed when the advance program is processed.

The restart of an interrupted advance program from the currently active block number is triggered by a positive edge of the start of advance program signal.

"Semi-automatic" mode
"Setup" mode

The positive edge of the "Advance program start" signal (PxxC.ADV) starts the advance program. Unlike in "Automatic" mode, the NC monitors the start advance program signal in the other modes after the program has been started. The advance program is executed as long as the signal is applied. When the signal is removed, active movements are decelerated along a ramp and program execution is interrupted.

A new advance program start restarts program execution.

"Automatic parameter test" mode

Program handling is not possible in this mode.

The following conditions interrupt an advance program:

- Power shutdown,
- error,
- removing the process enabling signal,
- stop,
- changing modes,
- Single step mode, and
- starting a reverse program.

Example Advance program start of a slave process in the PLC program.

Start of advance program:

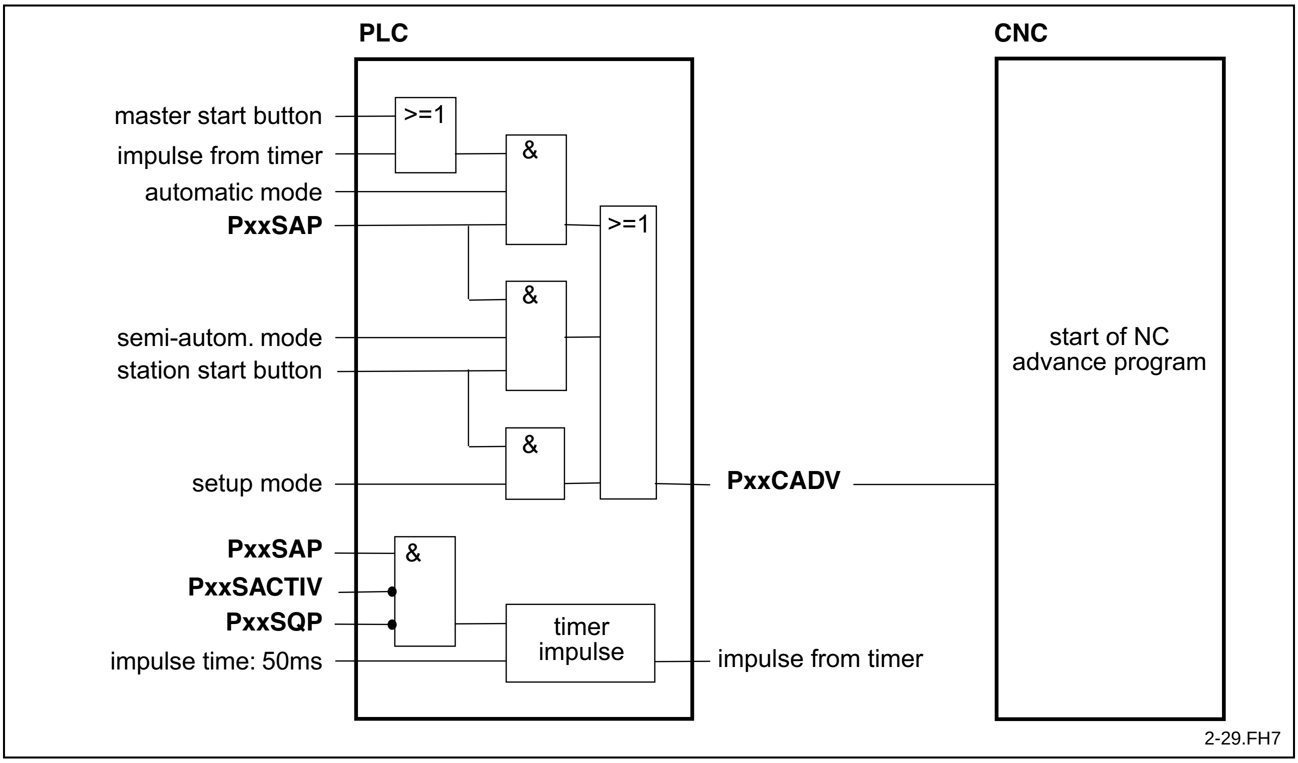
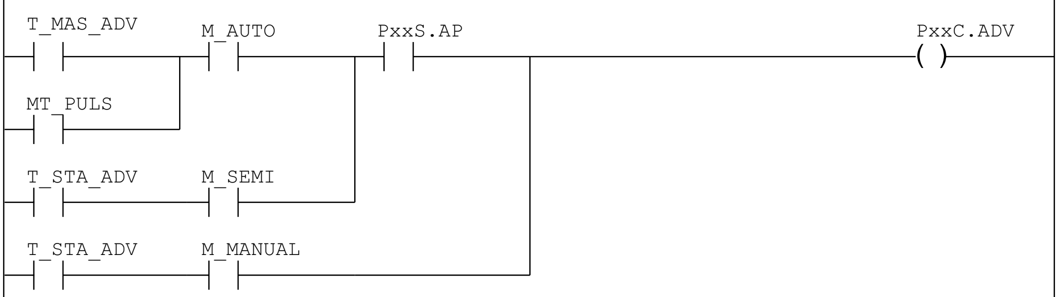


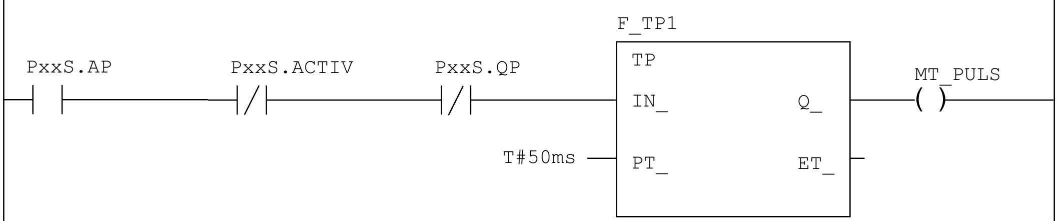
Fig. 1-44: Advance program start of a slave process

(*SPS-START SEQUENCE FOR AN NC ADVANCE PROGRAM -----------------------------------*)



T_MAS_ADV	START KEY FROM MASTER	VAR_INPUT	BOOL
M_AUTO	AUTOMATIC MODE		BOOL
PxxS.AP	START SIGNAL FROM MASTER NC PRG	VAR_INPUT	BOOL
PxxC.ADV	NC PROGRAM START	VAR_OUTPUT	BOOL
MT_PULS	START IMPULSE		BOOL
T_STA_ADV	START KEY FROM THE STATION	VAR_INPUT	BOOL
M_SEMI	SEMI-AUTOMATIC MODE		BOOL
T_STA_ADV	START KEY FROM THE STATION	VAR_INPUT	BOOL
M_MANUAL	MANUAL MODE		BOOL

(*GENERATING THE START IMPULSE ---*)



F_TP1	IMPULSE TIMER		TP
PxxS.AP	START SIGNAL FROM MASTER NC PRG	VAR_INPUT	BOOL
PxxS.ACTIV	NC PROGRAM ACTIVE	VAR_INPUT	BOOL
PxxS.QP	NC PROGRAM TERMINATED	VAR_INPUT	BOOL
MT_PULS	START IMPULSE		BOOL
T#50ms			TIME

list002.FH7

Fig. 1-45: PLC starting sequence for an NC advance program

Rexroth Bosch Group

- Start of the advance program in “automatic” mode: The NC program of the SLAVE process is started when an “APxx” is encountered in the NC program of the MASTER process. The PLC detects the programmed “APxx” NC command through the "PxxS.AP = 1" interface signal. The “PxxS.AP” signal generates a start impulse as long as it has not been acknowledged (PxxS.QP = 0) and no NC program of the SLAVE process is active. This start impulse starts the advance program of the SLAVE process. If program execution is interrupted (error, stop signal, process enabling signal removed, ...), the stopped program can be restarted via the start button on the MASTER control panel.
- Start of the advance program in “semi-automatic” mode: If the SLAVE process is in “semi-automatic” mode, the NC program starts when the start button at the station is pressed after the programmed “APxx” NC command (PxxS.AP = 1) has been recognized. The NC program is executed as long as the start button is pressed.
- In *"Setup"* mode, the NC program only executes as long as the start button at the station is pressed.

Conditional stop "PxxC.M001"

Designation

PxxC.M001 = **P**rocess **xx C**ommand Optional Stop **M001**

Control signal PLC →NC

PxxC.M001	= 0:	M01 does not interrupt NC program execution
PxxC.M001	= 1:	interrupt NC program execution through M01

The signal is effective in all modes.

Function

If the”"PxxC.M001” interface signal is “1”, programming the “M01” function in the NC program causes the program to stop. The auxiliary function is transferred to the PLC and must be acknowledged. The NC interprets the “PxxC.M001“interface signal after the M function has been acknowledged.

The NC stops program execution after the auxiliary function has been acknowledged and the remaining NC block has been processed.

If the interface signal has not been set to “1”, NC processing is not stopped after the M function has been acknowledged.

If a running spindle shall also be stopped after the program has stopped (PxxS.RUN=0), "spindle stop at program stop" (AxxC.SPHLT) must be employed for stopping the spindle too. The “program stopped” signal “PxxS.STOP” is not affected when the NC program was stopped.

Note: A change of the “PxxC.M001” interface signal state after a program interruption does not have an effect on the program interruption.

Timing of 'Conditional stop':

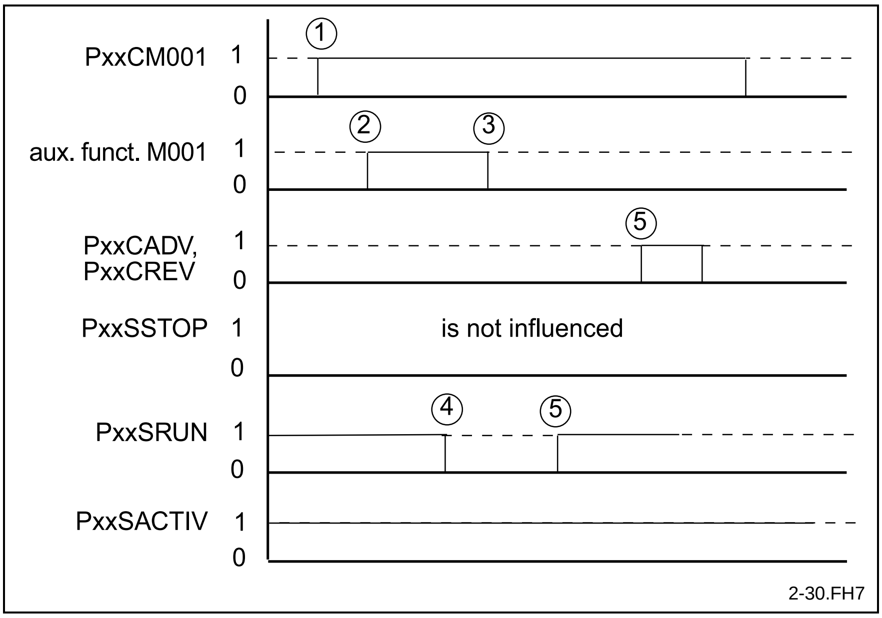


Fig. 1-46: Timing of "Conditional stop"

(1) Setting the "PxxC.M001" interface signal interprets "conditional stop M001" in the NC.

(2) (3) Interrogation and acknowledgment of the "M001" M function.

(4) Execution is stopped (PxxS.RUN = 0) after the M function has been acknowledged and the NC block terminated.

(5) A restart or a reverse program start restarts the stopped NC program and/or initiates a reverse program.

Reverse program start "PxxC.REV"

Designation

PxxC.REV = **P**rocess **xx** **C**ommand **Rev**erse Program Start

Control signal PLC →NC

PxxC.ADV = 0: No reverse program start

PxxC.REV = 1: Reverse program start

Function

Requirements for the start of the reverse program:

- power is available;
- no error condition exists, and
- Process enabling signal issued.

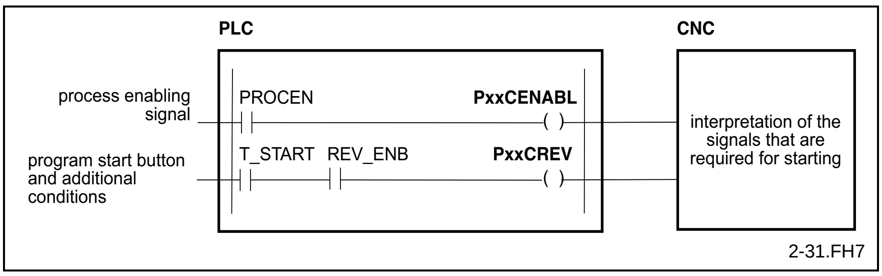


Fig. 1-47: Signal "Reverse program start"

This control signal has different meanings in the different modes.

Rexroth Bosch Group

“Automatic” mode
A positive edge of the "start of reverse program" (PxxC.REV) signal starts the reverse program. A reverse program start may also be initiated while an advance program is being executed or stopped.

The "Reverse program start" signal may be removed when the reverse program is processed.

A positive edge of the "start of reverse program" signal activates the restart of an interrupted reverse program from the currently active block number.

"Semi-automatic" mode
"Setup" mode

The positive edge of the "Advance program start" signal (PxxC.ADV) starts the advance program.

Unlike in "Automatic" mode, the NC monitors the reverse program start signal in the other modes after the program has been started. The reverse program is executed as long as the signal is applied. When the signal is removed, active movements are decelerated along a ramp, and program execution is interrupted. Program execution is resumed (restart) when the signal is reapplied.

"Automatic parameter test" mode

Program handling is not possible in this mode.

The following conditions interrupt a reverse program:

- Power shutdown,
- error,
- removing the process enabling signal,
- stop,
- Single step mode, and
- changing modes.

Example Reverse program start of a slave process in the PLC program.

Start of reverse program:

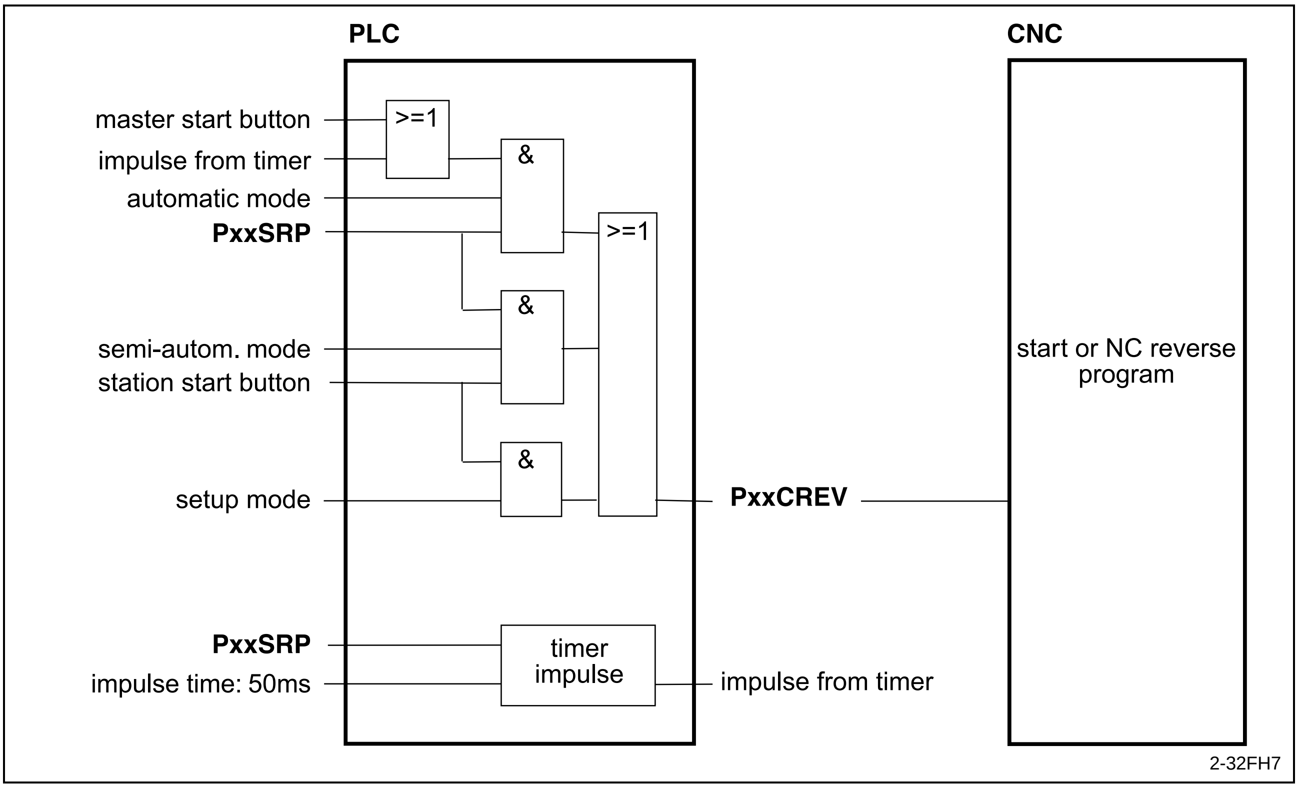
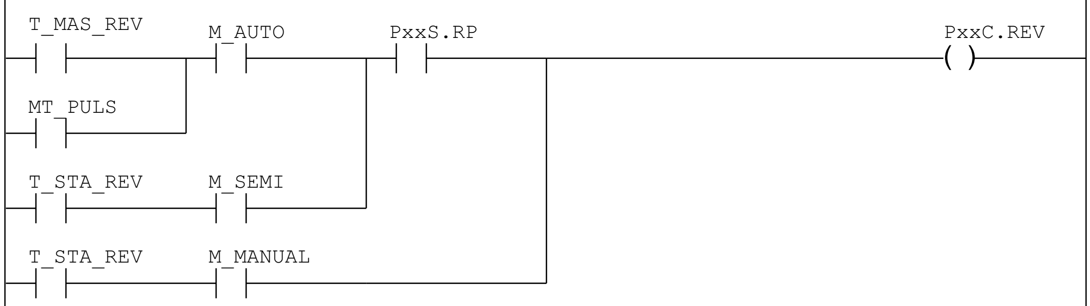
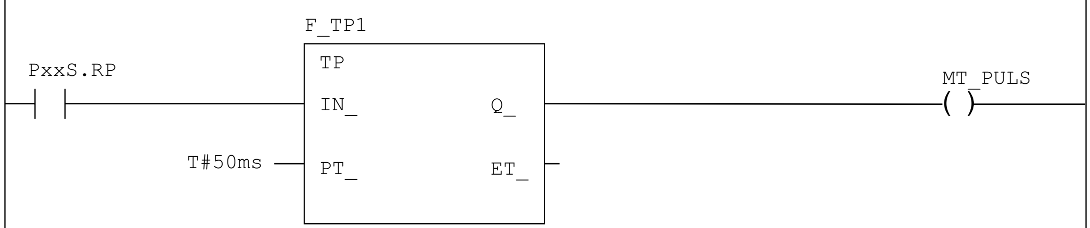


Fig. 1-48: Reverse program start of a slave process

```
(*SPS START SEQUENCE FOR AN NC REVERSE PROGRAM ----------------------------------*)
```




```
T_MAS_REV     START KEY FROM MASTER                VAR_INPUT     BOOL
M_AUTO        AUTOMATIC MODE                                     BOOL
PxxS.AP       START SIGNAL FROM MASTER NC PRG      VAR_INPUT     BOOL
PxxC.REV      NC PROGRAM START                     VAR_OUTPUT    BOOL
MT_PULS       START IMPULSE                                      BOOL
T_STA_REV     START KEY FROM THE STATION           VAR_INPUT     BOOL
M_SEMI        SEMI-AUTOMATIC MODE                                BOOL
T_STA_REV     START KEY FROM THE STATION           VAR_INPUT     BOOL
M_MANUAL      MANUAL MODE                                        BOOL

(*GENERATING THE START IMPULSE --------------------------------------------------*)
```




```
F_TP1         IMPULSE TIMER                                      TP
PxxS.RP       START SIGNAL FROM MASTER NC PRG      VAR_INPUT     BOOL
MT_PULS       START IMPULSE                                      BOOL
T#50ms                                                           TIME
```

list003.FH7

Fig. 1-49: PLC starting sequence for an NC reverse program

- Start of the reverse program in “automatic” mode: The NC program of the SLAVE process is started when an “APxx” is encountered in the NC program of the MASTER process. The PLC detects the programmed "RPxx" NC command through the "PxxS.RP = 1" interface signal. The "PxxS.RP" signal generates a start impulse.
- This start impulse starts the reverse program of the SLAVE process.
- If program execution is interrupted (error, stop signal, process enabling signal removed, ...), the stopped program can be restarted via the start button on the MASTER control panel.
- Start of the reverse program in “semi-automatic” mode: If the SLAVE process is in “semi-automatic” mode, the NC program starts when the start button at the station is pressed after the programmed “RPxx” NC command (PxxS.RP = 1) has been recognized.
  The NC program is executed as long as the start button is pressed.
- In *"Setup"* mode, the NC program only executes as long as the start button at the station is pressed.

Program stop "PxxC.STOP"

Designation

PxxC.STOP = **P**rocess **xx** **C**ommand Program **Stop**

Control signal PLC →NC

PxxC.STOP	= 0:	No program stop
PxxC.STOP	= 1:	Program stop

Program stop has an effect in all modes.

Function

The "Program stop" signal interrupts program execution for this process. Axis movements are decelerated along a ramp, and program preparation is interrupted. The stopped process can be continued by starting an advance or a reverse program. The "Program stop" signal has to have been cleared first, however.

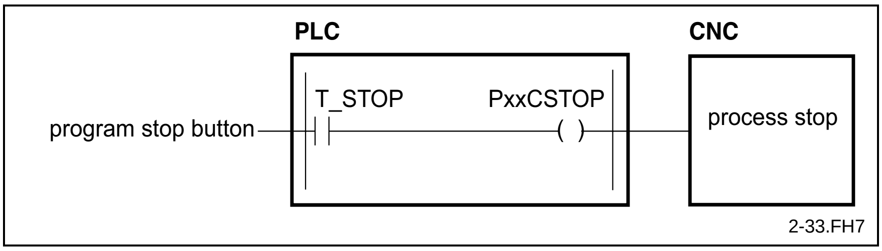


Fig. 1-50: Signal “Program stop”

Note: Until Rexroth MTC 200 Firmware Release 19V02, setting of the “program stop” signal had no effect on jog movements. This behavior has changed significantly as of Rexroth MTC 200 Firmware Release 19V03: Signal "program stop" (PxxC.STOP) is now also effective on jog movements. Continuous and incremental jogging can be cancelled by "program stop". Neither continuous, nor incremental jogging is possible with applied PxxC.STOP.

In conjunction with process synchronization it must be observed that the SLAVE processes which have been initiated by the MASTER process must also be stopped when the MASTER process is stopped (PxxC.STOP = 1).

Error clearing "PxxC.CLEAR"

Designation

PxxC.CLEAR = **P**rocess **xx C**ommand **Clear** *Error*

Control signal PLC →NC

PxxC.CLEAR = 0: No action

PxxC.CLEAR = 1: Clear error / control reset

The signal is effective in all modes.

Function

Depending on whether or not there is an error pending (PxxS.ERROR = 1/0), the signal has two different effects:

- PxxS.ERROR=0, PxxC.CLEAR = 1: →control reset
- PxxS.ERROR=1, PxxC.CLEAR = 1: →the pending error is cleared

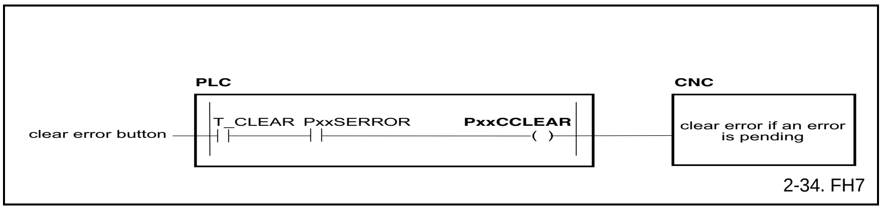


Fig. 1-51: Signal “Clear error”

Deleting errors

A pending error is cleared upon the rising flank of the "Clear error" signal.

The NC clears the "Error" signal to indicate that the error has been cleared.

The "Error" signal is set again and the error is indicated again if the error still exists.

Once an error has been successfully cleared, the advance program which was interrupted by the error cannot be restarted. The only exception is if the "Program execution required" process parameter has also been set to "no" and if the axes have measuring reference. If a start is to be possible without measuring reference, the "Reference required" process parameter must also be set to "no".

When a reverse program is initiated, program segments with reverse vectors will first be processed, followed by the base reverse program (starting from the jump label ".HOME").

Control reset

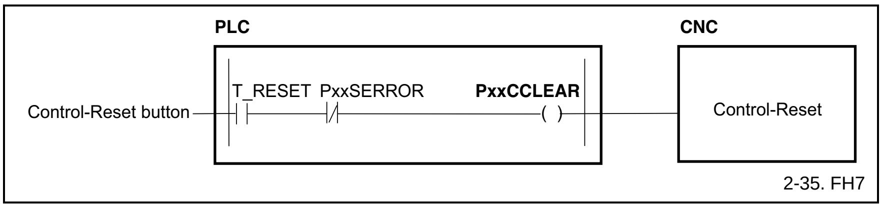


Fig. 1-52: Signal “Control reset”

A stopped NC program can be aborted by a "Control reset" input.

A running spindle is stopped by “Control reset”.

Programmed reverse vectors are cleared. The ".HOME" basic reverse vector is therefore called when a reverse program is initiated. This executes the basic reverse program. Complicated retract movements which permit a program-controlled establishment of the readiness to start by programming reverse vectors are not performed.

The operator must perform those movements in "Setup" mode through jog movements and manual output of auxiliary functions.

Once it has been ensured that the basic reverse program can be executed without interference, the operator may trigger a start of the reverse program and initiate the basic reverse program.

Note:	A "Control Reset" should be performed before the "NC block search" function is used. This ensures that reverse vectors and event monitoring functions are updated.

The following signals and actions are also influenced by a "Control reset":

- a current MDI block is aborted,
- the "part machined" (PxxS.POK) status signal is reset,
- the "tool magazine required" (PxxS.MGREQ) signal is reset,
- the "tool is worn out" (PxxS.MGTWO) status signal is reset,
- the "tool warning limit" (PxxS.MGWRN) status signal is reset,
- the "NC block active" (PxxS.ACTIV) status signal is reset,
- the "reverse program active" (PxxS.REV) status signal is reset,
- the "program stopped" (PxxS.STOP) status signal is reset,
- the "control voltage interruption" (PxxS.POWIN) status signal is reset,
- the "tool error status" (PxxS.MGERR) status signal is reset,
- pending M functions are aborted,
- pending tool transfer commands are aborted,
- an unconditional stop (M00) is terminated,
- a conditional stop (M001) is terminated,
- CW spindle rotation (M3) is aborted,
- clockwise spindle rotation is aborted,
- process control commands are reset: DP, AP, RP, WP, LP, POK,
- the "PxxS.READY" signal is set to "1" if the "homing required" parameter has been set to "No"
- messages that have been generated in the NC program are cleared,
- the magazine is enabled for manual mode,
- magazine commands are aborted,
- the C axis is initialized according to the feedback from the PLC,
- the NC block "N0001" appears in the position display for the currently selected NC program,
- zero offsets are set according to the parameters,
- the upper limit of the override value is set to 100%,
- G codes are set according to their default values,
- internal links are de-activated (threading, tapping, constant cutting speed, feed per revolution, transmit function)
- the zero offset table "O" is selected,
- event monitoring is de-activated,
- process acceleration is set to 100%,
- with active axis transfer, the transferred axes are fetched back to their "mother process",
- the "travel to dead stop" state is reset,
- a pending AxD command from the NC is aborted,
- spindle "1" is used as reference spindle,

- the gear step is initialized according to the PLC specification
- the D corrections are de-activated.

Note: In conjunction with process synchronization it must be ensured that the associated MASTER process receives a process stop signal when a "Control-Reset" or "clear error" command is issued for the SLAVE process.

If this is not ensured, the MASTER process may continue executing its NC program despite a programmed WPx.

Repositioning to the contour "PxxC.REPOS"

Designation **PxxC.REPOS** = **P**rocess **xx C***ommand* **Repos***itioning*

Control signal PLC →NC

PxxC.REPOS = 0: No activation of repositioning

PxxC.REPOS = 1: Repositioning can be executed

Function The PLC sets the "PxxC.REPOS" control signal to activate the repositioning option. The control signal is only interpreted in the NC if

- the operator has interrupted the NC program execution in "Automatic", "Semi-automatic", or "Program execution in manual mode" program mode (PxxS.ACTIV = 1) and
- has left the contour through axis jogging in "Manual" mode.

The NC ignores the interface signal in any other situation.

During repositioning in manual mode, the jog keys respond as in normal jog mode. The NC ensures that the axes do not the overrun start or target position.

In the program modes, a subroutine ".REPOS" is executed to restore the state that existed before the interruption. The machine manufacturer has defined the execution sequence in that program.

The "start advance program" signal (PxxC.ADV) is used to start the subroutine.

The NC always completes the execution of the REPOS subroutine, even if the PLC removes the control signal during execution.

Returning to the contour “PxxC.RESTA”

Designation **PxxC.RESTA** = **P**rocess **xx C***ommand* **Resta***rt*

Control signal PLC →NC

PxxC.RESTA = 0: No activation of returning to position

PxxC.RESTA = 1: Returning to position can be executed

Function The PLC sets the "PxxC.RESTA" control signal to activate the restart option. The NC interprets the control signal in the same way as in the repositioning process.

With respect to restarting in manual mode and in the program modes, the NC also refers to the REPOS subroutine. The same rules are valid for restart and for repositioning.

Start block preprocessing "PxxC.BPSTR"

Designation **PxxC.BPSTR** = **P**rocess **xx** **C***ommand* NC **B**lock **P**reprocessing **Star**t

Control signal PLC →NC

PxxC.BPSTR	= 0:	No start of block preprocessing
PxxC.BPSTR	= 1:	Start of block preprocessing

Function This signal is used for starting block preprocessing. The "NC program restart active" status signal **PxxS.BPACT** must be set as a prerequisite.

Dry run mode "PxxC.DRYRN"

Designation PxxC.DRYRN = **P**rocess **xx** **C**ommand **Dry** **R**u**n**

Control signal PLC →NC

PxxC.DRYRN	= 0:	No dry run
PxxC.DRYRN	= 1:	Dry run selected

Function If the PLC sets the PxxC.MGWTC signal at logic 1, no tool identification test is performed.

Setting the PxxC.DRYRN (DRY RuN) interface signal is a prerequisite for the NC to activate the functions which were selected on the machine data page 62 ´Test mode´

- suppress assisting functions,
- suppress tool transfer/movement, and/or
- Activate test feed

If the interface signal is set to 0, the following test mode functions are not active:

- suppress assisting functions,
- suppress tool transfer/movement, and
- Activate test feed

Activation The NC will consider any changes to the interface signal only in the next program run.

NC Program Status

Error "PxxS.ERROR"

Designation PxxS.ERROR = Process xx *Status* Error

Status signal NC →PLC

PxxS.ERROR	= 0:	No error
PxxS.ERROR	= 1:	Error

This signal has an effect in all modes.

Function The NC employs the "error" signal to notify the PLC of erroneous processes. The uncoded error text is displayed in the diagnosis overview of the user interface or the BTC. In addition, a flashing message “Mechanism error?”? is displayed.

The PLC can use a standard function (MSG_RD) to read the error number.

A pending error must be cleared by setting the "clear error" signal for the related process in the PLC program. The "PxxS.ERROR" error signal is set again if the error condition is still active after the error has been cleared. An NC program cannot be executed as long as an error status is pending.

Transformation active "PxxS.TRANS"

Designation

PxxS.TRANS = **P**rocess **xx S**tatus **Trans**formation is Active

Status signal PLC →NC

PxxS.TRANS	= 0:	Transformation not active
PxxS.TRANS	= 1:	Transformation is active

The NC updates this signal in all modes.

Function

This signal is used to tell the PLC whether or not the translation from cartesian co-ordinates into polar co-ordinates has been activated.

Readiness to start "PxxS.READY"

Designation

PxxS.READY = **P**rocess **xx S**tatus Program is **Ready** to Start

Status signal NC →PLC

PxxS.READY	= 0:	Not ready to start
PxxS.READY	= 1:	Ready to start

The signal is effective in all modes.

Note: Depending on the settings of the two parameters "reverse program execution required" and "reference required", executing a reverse program and/or the axis reference is not necessary for starting the advance program.

Function

The "Ready to start" signal indicates that a process is ready to be started.

Depending on the setting of the “readiness to start required” parameter, a process is for the first time ready to be started after a reverse program has been executed. In this reverse program, a reference dimension must be established for all axes that belong to the process.

If the process has a magazine assigned, that magazine must also be homed. Homing is not necessary if the reference of the axes and/or the magazine exists.

The "ready to start" signal is set when the reverse program is terminated with the "jump with stop" (BST) or "program end with reset" (RET) command.

The "Ready to start" condition is also established when an advance program has been correctly executed. In this case, the forward program must be terminated by a "jump with stop" (BST) or "program end with reset" (RET) command.

The "ready to start" signal is cleared when an advance or reverse program is active (PxxS.ACTIV).

Any type of jogging movement (single axis homing, jogging) cancels the "Ready to start" signal.

Manual magazine movements do not cancel the "ready to start" signal.

Successful completion of the reverse program permits the "Ready to start" signal to be reestablished.

Single-axis homing of all axes that are related to the process does not permit the "ready to start" signal to be established.

Block active "PxxS.RUN"

Designation **PxxS.RUN** = **P**rocess **xx** **S***tatus Program* **Run***ning*

Status signal NC →PLC

PxxS.RUN	= 0:	No NC block being executed
PxxS.RUN	= 1:	An NC block is being executed

The signal is effective in all modes.

Function The "NC block active" signal shows that an NC block is being executed. Distinction between advance and reverse programs is not made here.

The "NC block active" signal is retained when an NC block is stopped by a stop command.

The "NC block active" signal is cleared when a "jump with stop" (BST) or "program end with reset" (RET) block is executed in the program.

In single-step mode, the "NC block active" signal is cleared after each NC block.

Any type of jogging movement (single axis homing, jogging) cancels the "NC block active" signal. Manual magazine movements do not cancel the "NC block active" signal. Setting the "Clear error" output in the PLC program without a pending process error clears the "NC block active" signal (control-reset).

Program active "PxxS.ACTIV"

Designation

PxxS.ACTIV = **P**rocess **xx** **S***tatus Program* **Active**

Status signal NC →PLC

PxxS.ACTIV	= 0:	No NC program active
PxxS.ACTIV	= 1:	NC program active

The signal is effective in all modes.

Function

The "NC program active" signal shows that an NC program is being executed. Distinction between advance and reverse programs is not made here.

The "NC program active" signal is retained when an NC program is stopped by a stop command.

The “program active” signal is cleared when a "jump with stop" (BST) or "program end with reset" (RET) block is executed in the program.

Unlike the "NC block active" signal, the "NC program active" signal is retained after each NC block in single block mode.

Unless the "axis jogging causes reset" parameter has been set to "No", any type of jogging movement (single axis homing, jogging) cancels the "NC program active" signal.

Manual magazine movements do not cancel the "NC program active" signal.

Setting the "Clear error" signal in the PLC program without a pending process error clears the "NC program active" signal (control-reset).

Reverse Program Active "PxxS.REV"

Designation

PxxS.REV = **P**rocess **xx** **S**tatus **Rev**erse Program Active

Status signal NC →PLC

PxxS.REV	= 0:	No reverse program active
PxxS.REV	= 1:	Reverse program active

The signal is effective in all modes.

Function

The "Reverse program active" signal is output in addition to the "NC block active" and "NC program active" signals when a reverse program is executed.

The "reverse program active" signal remains active when an NC reverse program is stopped by a stop command. The "reverse program active" signal is cleared if a "jump with stop" (BST) or "program end with reset" (RET) block has been executed in the program, and a reverse program start signal has not been applied.

In single-block mode, the "Reverse program active" signal is retained after each block.

Any type of jogging movement (single axis homing, jogging) cancels the "Reverse program active" signal. Manual magazine movements do not influence the "reverse program active" signal. Setting the "Clear error" signal in the PLC program without a pending process error clears the "Reverse program active" signal (control-reset).

Reverse vector active “PxxS.RVACT""

Designation

PxxS,RVACT = **P**rocess **xx** **S**tatus **R**everse **V**ector **Act**ive

Status signal NC →PLC

PxxS.RVACT = 0: No reverse vector active

PxxS.RVACT = 1: Reverse vector has been activated in the NC program

This signal becomes active when the NC command for setting a reverse vector (REV.label) was executed in the NC program.

Function

The signal is cleared:

- Upon end of NC program with RET, BST, M2 and M30,
- after a Control Reset, and
- after jogging the axes when the "jogging causes reset" parameter is active.

The signal remains active after the NC is switched off if it was already active prior to switching off.

Program has been stopped "PxxS.STOP"

Designation

PxxS.STOP = **P**rocess **xx** **S**tatus NC Program **Stop**

Status signal NC →PLC

PxxS.STOP = 0: No program stop

PxxS.STOP = 1: Program has been stopped

The signal is effective in all modes.

Function

The "NC program stopped" signal indicates that an NC program has been stopped. The program in question can be an advance or a reverse program.

The "NC program stopped" signal is set as soon as the NC program execution is stopped (program stop in the NC program, PLC program; mode changeover during program execution, etc.) and all axes belonging to the process are stopped. The signal will also be set if the control voltage is interrupted during NC program execution.

The "NC program stopped" signal is not set if a block has been executed in single block mode and the next program start is awaited, or if an "HLT", "M00" or "M01" has been programmed in the NC program.

The "NC program stopped" signal is cleared when the interrupted program is restarted by an advance or reverse program start.

Any type of jogging movement (single axis homing, jogging) cancels the "NC program stopped" signal.

Manual magazine movements do not influence the "NC program stopped" signal. Setting the "clear error" signal in the PLC program without a pending process error clears the “NC program stopped” signal (Control Reset).

Interruption of control voltage "PxxS.POWIN"

Designation **PxxS.POWIN** = **P**rocess **xx** **S**tatus **Pow**er **In**terrupt

Status signal NC →PLC

PxxS.POWIN = 0: No interruption of control voltage

PxxS.POWIN = 1: Program stop through interruption of control voltage

The signal is effective in all modes.

Function The "Control voltage interruption" signal indicates that an executing NC program has been interrupted by switching off the controller power supply. The signals PxxS.POWIN = 1 and "NC program stopped" are set when the power supply is switched back on.

After the control voltage has been interrupted, executing a reverse program re-establishes the ready-to-start condition if this is required for processing an advance program (i.e. the "program execution required" parameter has been set to "`Yes`").

The "Control voltage interruption" signal is cleared when a reverse program is started. Any type of jogging movement (single axis homing, jogging) cancels the "Control voltage interruption" signal.

Manual magazine movements do not cancel the "control voltage interruption" signal.

Setting the "clear error" output in the PLC program without a pending process error clears the "control voltage interruption" signal (Control Reset).

MDI active (MDI = MANUAL DATA INPUT) "PxxS.MDIAC"

Designation **PxxS.MDIAC** = **P**rocess **xx** **S**tatus **MDI** Mode is **Ac**tive

Status signal NC →PLC

PxxS.MDIAC = 0: No MDI block ready for execution

PxxS.MDIAC = 1: MDI block ready for execution or being executed

The signal has an effect in "setup" mode.

Function Individual NC blocks have to be entered and executed during a station startup. The NC program block can, in the simplest case, be an auxiliary function (e.g. clamping the workpiece) or a complete tool changing subroutine. Such an NC block is entered in MDI mode (Manual Data Input) via the PC and BTC input devices.

"MDI" mode is a submode of the "Setup" mode. The controller automatically transitions to this mode after an MDI block has successfully been transferred and is available in the Rexroth MTC 200 for processing. Transferring the MDI block requires "Setup" mode to be preselected. The NC issues a "PxxS.MDIAC" output signal for each process to inform the PLC program of this internal state.

The "**MDI active**" signal is set once an MDI block has been successfully transferred to the Rexroth MTC 200 and is ready for processing. MDI block processing is initiated by an advance program start (PxxC.ADV). Once the "MDI active" signal has been set, it is retained when modes change.

The "MDI active" signal remains active until the MDI block has been completely processed or MDI is aborted. MDI mode is aborted by initiating a reverse program, jogging an axis or making a Control Reset of the process. The "MDI active" signal is not stored when the power supply is switched off.

Examples

Utilization of the "MDI mode active" signal

- The "MDI active" signal is used as a condition that causes the "ADVANCE" key to start blinking. A blinking "ADVANCE" key indicates that an MDI block is ready to be processed.

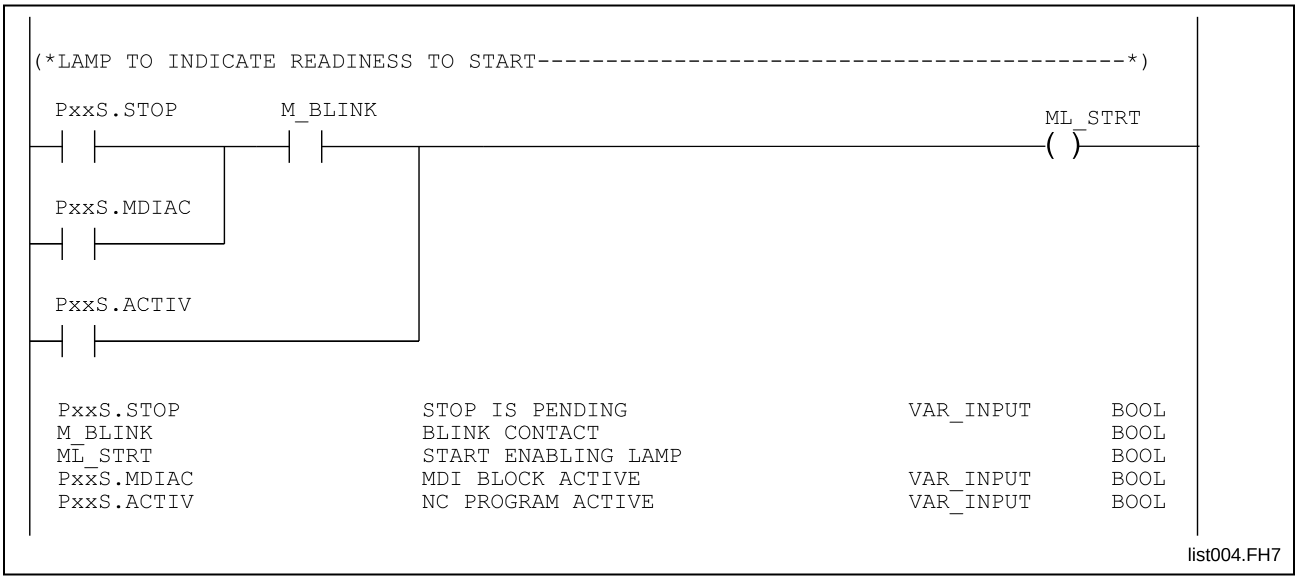


Fig. 1-53: Lamp as indication of readiness to start

- The "MDI active" signal permits a separate "MDI start" key to exist on the station control panel. The key is interconnected with the "MDI active" signal. An advance program start is only supplied if the "MDI active" signal is applied. This prevents incorrect input and confusion between MDI mode and setup.

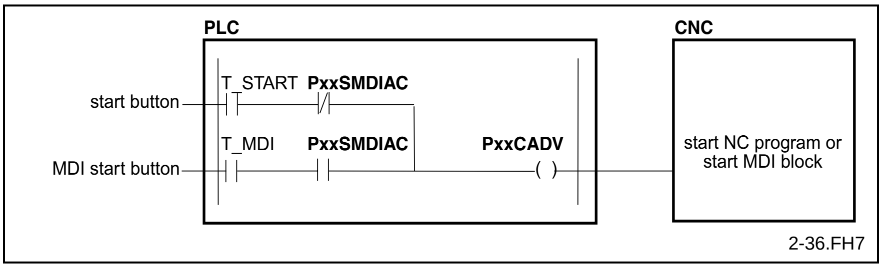


Fig. 1-54: Signal “MDI enable”

Repositioning / restart terminated "PxxS.CREST"

Designation

PxxS.CREST = **P**rocess **xx** **St**_atus_ **C**_onditions_ **Rest**_ored_

Status signal NC →PLC

PxxS.CREST	= 0:	Repositioning / restart is not yet terminated
PxxS.CREST	= 1:	Repositioning / restart is terminated

Function

The NC sets this status signal to inform the PLC that all axes that are relevant to repositioning or restart have adopted the state they had before the interruption.

In this context, relevant axes are:

- all spindles,
- main spindles with rotary axis capability,
- combined turret/spindle axes that are in spindle mode,
- all feed axes whose "Adjustment window for ..." axis parameter is different from "0", and
- the tool magazine axis if the "Reposition tool storage axis" process parameter has been set to "`Yes`".

Provided that the related control signals (PxxC.REPOS, PxxC.RESTA) have been set, the NC only updates the status signal during repositioning or restart.

Repositioning / restart active "PxxS.REPOS"

Designation

PxxS.REPOS = **P**rocess **xx** **S**_tatus_ **Repos**_itioning_

Status signal NC →PLC

PxxS.REPOS	= 0:	Repositioning / restart not active
PxxS.REPOS	= 1:	Repositioning / restart is active

Function

The NC sets this status signal to notify the PLC that repositioning or restart has been activated.

NC program restart is active "PxxS.BPACT"

Designation

PxxS.BPACT = **P**rocess **xx** **S**_tatus NC_ **B**_lock_ **P**_reprocessing_ **Act**_ive_

Status signal NC →PLC

PxxS.BPACT	= 0:	NC program restart not active
PxxS.BPACT	= 1:	NC program restart is active

Function

The NC employs this status signal to tell the PLC that NC program restart is active.

The signal is only activated after

- the NC program restart function has been selected via the GUI,
- a valid target block has been selected,
- the "ready to start" signal (PxxS.READY=1) has been applied for the process,
- the NC program is available in the controller,
- the program has been selected for execution,

- automatic mode has been preselected,
- and no NC program is active (PxxS.ACTIV=0).

Adaptive Feed regulation thrust missing "PxxS.THMIS"

Designation PxxS.THMIS = **P**rocess **xx** **S**tatus **Th**rust **Mis**sing

Status signal NC →PLC

PxxS.THMIS = 0: The machining moment has exceeded the specified minimal machining moment during machining.

PxxS.THMIS = 1: The machining moment has not exceeded the specified minimal machining moment during machining.

Function The NC updates the interface signal when it switches off the adaptive feed regulation (G26). In addition, the NC resets this signal upon program end and Control Reset.

If during machining with active adaptive feed regulation the machining moment does not exceed the minimal machining moment, the NC records this by setting the "thrust missing" (PxxS.THMIS) interface signal when the adaptive feed regulation is switched off .

Adaptive Feed Regulation Excessive Thrust "PxxS.EXCTH"

Designation PxxS.EXCTH = **P**rocess **xx** **S**tatus **Exc**essive **Th**rust

Status signal NC →PLC

PxxS.EXCTH = 0: The current adaptive feed regulation does not exceed maximum feed reduction.

PxxS.EXCTH = 1: The current adaptive feed regulation does exceed maximum feed reduction.

The NC cyclically updates the interface signal when adaptive feed regulation (G26) is active.

Function If during machining with active adaptive feed regulation the current feed reduction exceeds the maximal feed reduction, the NC records this by setting the "excessive thrust" signal. The NC allows the interface signal to remain active only for as long as the current feed reduction exceeds the maximal feed reduction.

The NC continues the machining regardless of whether the current feed reduction exceeds the maximal feed reduction. Only when the current feed reduction reaches 100% (or feed movement = 0) will the NC stop the machining and generate a corresponding error message.

Dry run mode "PxxS.DRYRN"

Designation PxxS.DRYRN = **P**rocess **xx** **S***tatus* **Dry** **R***u***n**

Status signal NC →PLC

PxxS.DRYRN = 0: No dry run

PxxS.DRYRN = 1: Dry run selected

Function The status signal indicates that test mode was selected by the PLC.

Rapid traverse with G0 active "PxxS.G00"

Designation PxxS.G00 = Process xx St*atus* G00

Status signal NC →PLC

PxxS.G00 = 0: No rapid traverse (G00) active

PxxS.G00 = 1: Axis movement is performed at rapid traverse rate (G00)

Function This status signal informs whether there is a movement with rapid traverse rate (G00) of the related process.

Active homing of feed axes using G74 "PxxS.G74"

Designation PxxS.G74 = Process xx St*atus* G74

Status signal NC →PLC

PxxS.G74 = 0: Homing (G74) not active

PxxS.G74 = 1: Homing of a feed axis (G74) active

Function This status signal informs whether homing with "G74" is active for a feed axis.

Velocity-optimized block transition active "PxxS.G08"

Designation PxxS.G08 = Process xx St*atus* G08

Status signal NC →PLC

PxxS.G08 = 0: No velocity-optimized block transition (G08) active

PxxS.G08 = 1: Velocity-optimized block transition (G08) active

Function This status signal informs whether a velocity-optimized block transition (G08) has been programmed for the related process.

Thread cutting or thread tapping active "PxxS.TREAT"

Designation PxxS.TREAT = Process xx Status Threat

Status signal NC →PLC

PxxS.TREAT = 0: No threading / tapping active

PxxS.TREAT = 1: Threading / tapping active

Function This status signal informs whether a threading or tapping G code has been programmed for the related process.

In this context, the following G codes are interpreted:

- G33 thread cutting,
- G63 rigid tapping (spindle is stopped at the end of the motion),
- G64 rigid tapping (spindle continues rotating at the end of the motion), and
- G65 tapping with the spindle as master axis.

G code for constant cutting code active “PxxS.G96”

Designation PxxS.G96 = Process xx St*atus* G96

Status signal NC →PLC

PxxS.G96 = 0: G96 not active

PxxS.G96 = 1: G96 is active

Function This status signal informs whether the "G code" for constant cutting speed (G96) has been programmed for the related process.

This means that the NC determines the spindle speed that is compatible with the current turning diameter.

Feed and Spindle Override

Feed override "PxxC.FOVRD"

Designation **PxxC.FOVRD** = **P**rocess **xx C**ommand **F**eedrate **Ov**er**r**i**d**e

Control signal PLC →NC

PxxC.FOVRD = 0 ... 255

Function "Feed override" is valid for linear and rotary axes. Feed override is interpreted in the NC, irrespective of the mode; it has an effect on any axis movement (except on homing digital axes).

The valid range of override weighting by the PLC program is between 0 and ... 255 %. If too high an override value is selected, the NC delimits the axis and/or process velocity to the maximum values that have been defined in the parameters. The velocities of the individual axes are proportionally reduced if several axes are moved simultaneously. Thus, the programmed path (circle, straight line) is retained.

Note: Override weighting must always be generated in the PLC program. Axis movement is not possible without override weighting. An NC diagnosis is output if a motion command is output to an axis and override weighting 0% has been selected.

Bosch Rexroth provides function blocks that can be used for outputting the override weighting to the NC.

From Version 04.14/xx onwards, this signal will only be interpreted for interpolating axes.

There is an axis-related override value (AxxC.OVRD) for spindle and magazine axes.

Spindle override "PxxC.SOVRD"

Designation **PxxC.SOVRD** = **P**rocess **xx C**ommand **S**pindle **Ov**er**r**i**d**e

Control signal PLC →NC

PxxC.SOVRD = 0 ... 255

Function "Spindle override" is valid for spindle axes.

Spindle override is interpreted in the NC irrespectively of the mode; it has an effect on any spindle movement (except on homing digital spindles).

The valid range of override weighting by the PLC program is between 0 and 255%. If too high an override value is selected, the NC delimits the spindle speed to the maximum values that have been defined in the parameters.

Note: Override weighting must always be generated in the PLC program. Axis movement is not possible without override weighting. An NC diagnosis is output if a speed command is output to the spindle while override weighting 0% has been selected.

Bosch Rexroth provides function blocks that can be used for outputting the override value to the NC.

From Version 04.14/xx onwards, this signal will no longer be interpreted.

Rapid override "PxxC.SOVRD"

Designation **PxxC.ROVRD** = **P**rocess **xx** **C***ommand* **R***apid* **Ove***r***r***i***de**

Control signal PLC →NC

PxxC.ROVRD = 0 .. 255

Function The NC interprets this override value in all axis movements that are performed using "G00".

The valid range of override weighting by the PLC program is between 0 and 255%.

Note: If there is no external G00 override selector switch available, the same signal should be applied to the "PxxC.ROVRD" interface signal and to the "PxxC.FOVRD" interface signal.

NC program synchronization

Define process "PxxS.DP", "PxxC.DP"

Designation **PxxS.DP** = **P**rocess **xx** **S**tatus **D**efine **P**rocess

Status signal NC →PLC

PxxS.DP = 0: SLAVE process not defined

PxxS.DP = 1: SLAVE process defined in the NC MASTER process

Designation **PxxC.DP** = **P**rocess **xx** **C**ommand **D**efine **P**rocess

Control signal PLC →NC

PxxC.DP = 0: SLAVE process not defined

PxxC.DP = 1: SLAVE process defined in the NC MASTER process

Function A process (SLAVE process) must be defined in the calling process (MASTER process) before an advance or reverse program is started. An error message is automatically defined when an advance or reverse program start is programmed without a preceding process definition.

Programming a "DP" command in the NC program of the MASTER process causes the "define process" status signal to be output to the PLC. A process (PxxS.DP) and a mechanism (MxxS.DP) can be defined.

Defining a SLAVE process in the NC program of the MASTER process:

```
.
.
DP1         ;Define process 1
SP1 5       ;Preselect program number 5 for process 1
AP1         ;Start process 1 advance program
WP1         ;Wait until process 1 is completed
POK         ;Part OK if POK for all defined processes

BST .START
.           ;Next NC program
.
```

The "define process" (PxxS.DP, MxxS.DP status signals can be used in the PLC program for interrogating whether the NC program of the MASTER process contains the "DP<process number>" command for the corresponding SLAVE process. Defining a process or mechanism in the NC program requires prerequisites (such as automatic mode or power available) to exist before an advance or reverse program can be started.

If necessary prerequisites are missing, a diagnosis with a specification of the missing prerequisites can be generated in the PLC program.

Example

Signal flow of the "define process" signals for process 1

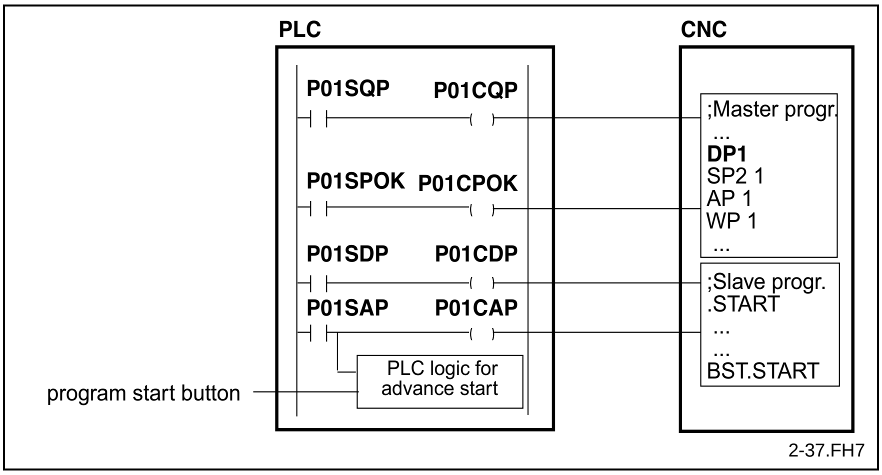


Fig. 1-55: Signal flow of signals “Define process”

In addition to transferring the “define process” (PxxS.DP) status signal from the MASTER process to the "define process" (PxxC.DP) control signal of the SLAVE process, diagnoses for missing prerequisites are derived in the PLC program.

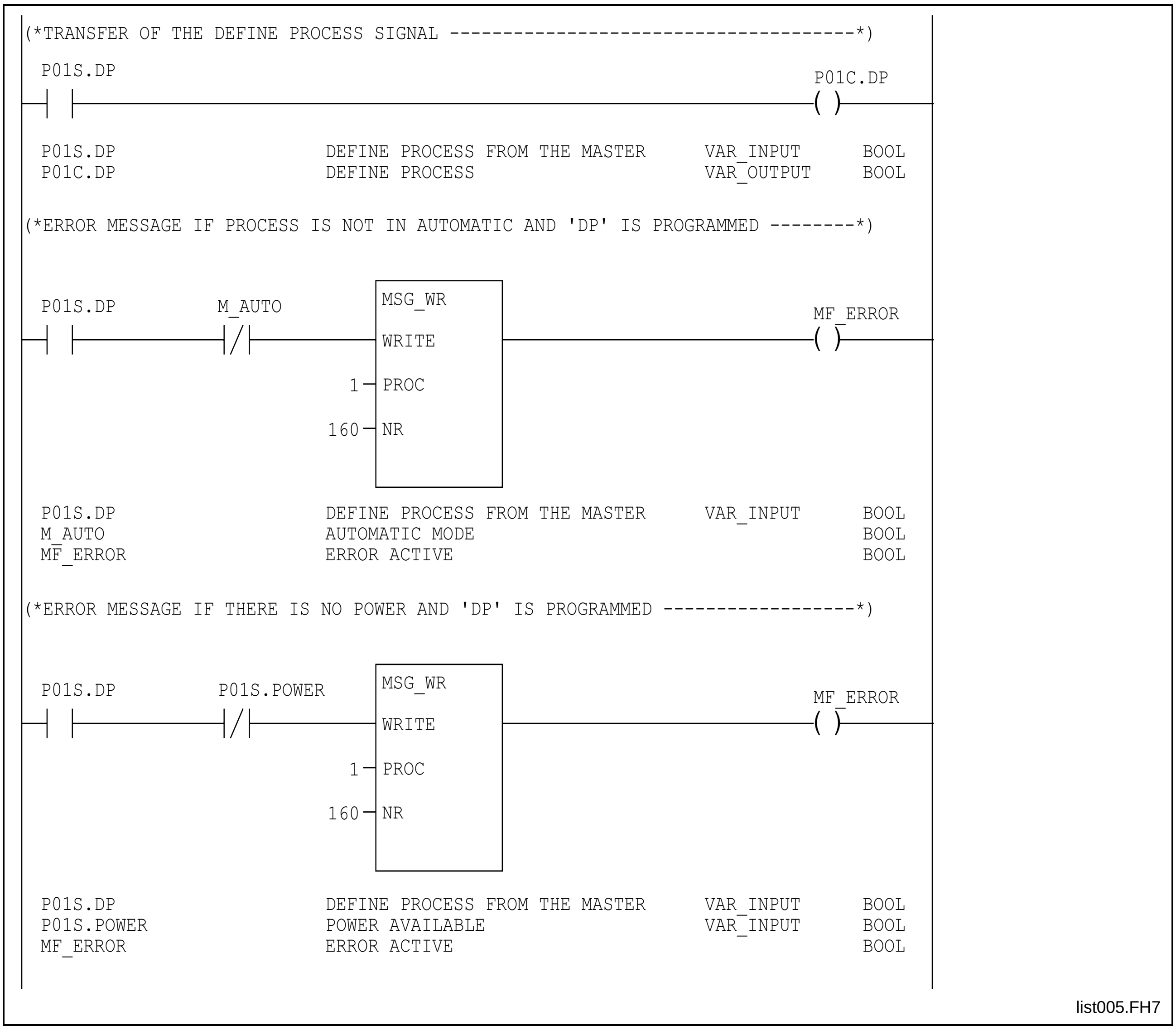


Fig. 1-56: Transferring of the signal “Define process”

Lock process “PxxS.LP"

Designation

PxxS.LP = **P**rocess **xx** **S**tatus **L**ock **P**rocess

Status signal NC →PLC

PxxS.LP = 0: Enable process for motion

PxxS.LP = 1: Lock process for motion

Function

Provided it is not required for machining, a SLAVE process can be kept in a defined state (on cams or in a defined working range for example) and/or disabled for movements during the program execution of a MASTER process.

The status signal remains applied until the calling process has terminated its program (BST or RET) or a Control Reset is performed.

Programming an “LP” command in the NC program of the MASTER process causes the “lock process" signal to be output to the PLC. A process (PxxS.LP) or a mechanism (MxxS.LP) can be locked.

Defining a SLAVE process in the NC program of the MASTER process:

```
.
.
LP1         ;lock process 1
DP2         ;Define process 2
SP2 5       ;Preselect program number 5 for process 2
AP2         ;Start process 2 advance program
WP2         ;Wait until process 2 is completed
POK         ;Part OK if POK for all defined processes

BST .START
.           ;Next NC program
.
```

The "lock process" (Pxx.SLP, MxxS.LP) signals can be used in the PLC program for interrogating whether a SLAVE process has been disabled in the NC program of the MASTER process. The defined state must be defined in the PLC program (e.g. mechanical home position of the feed axis on the reference switch).

Any violation of this requirement should interrupt program execution in the MASTER process and in the SLAVE process. If several processes process a workpiece at the same time as the SLAVE process, program execution must be interrupted for those processes, too.

Program execution interruption can be triggered by a program stop or by removing the process enabling signal. This incorrect input must also be indicated by a diagnosis in the PLC program.

Example

Signal flow for "Lock process" for process 1

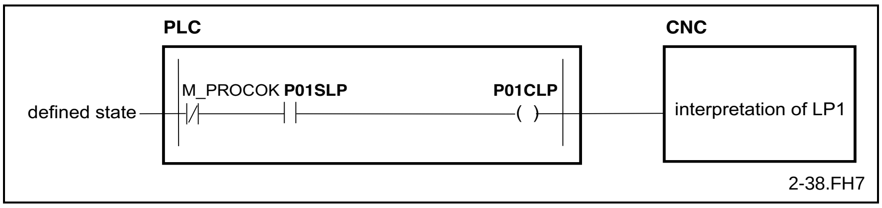


Fig. 1-57: Signal “Lock process”

Advance process enabling “PxxC.AP", “PxxS.AP"

Designation **PxxC.AP** = **P**rocess **xx** **C**ommand **A**dvance **P**rocess Start

Control signal PLC →NC

PxxC.AP = 0: No advance program enable

PxxC.AP = 1: Advance program enable

Designation **PxxS.AP** = **P**rocess **xx** **S**tatus **A**dvance **P**rocess Start

Status signal NC →PLC

PxxS.AP = 0: No advance program enable

PxxS.AP = 1: Advance program enable

Function In the NC program of the MASTER process, the “AP<process number>“ command starts the advance program of a SLAVE process or mechanism.

Starting the advance program of a SLAVE process in the NC program of a MASTER process:

```
.
.
DP1          ;Define process 1
SP1 5        ;Preselect program number 5 for process 1
AP1          ;Start process 1 advance program
WP1          ;Wait until process 1 is completed
POK          ;Part OK if POK for all defined processes

BST .START
.            ;Next NC program
.
```

The “advance program enabling” (PxxS.AP) status signal is used for telling the PLC program that the NC program of the MASTER process contains an advance program start for that SLAVE process.

Advance program enabling has two different effects in the PLC program:

- Advance program enabling is required for generating the "advance program start" signal for the SLAVE process in automatic mode and semi-automatic mode. In these two modes, advance program start for the SLAVE process or mechanisms is only permitted in the PLC program if an advance program start has also been programmed in the NC program of the MASTER process and if the “advance program enabling” status signal from the MASTER process is present.
- Correct acknowledgment of the execution of the SLAVE process NC program requires the “advance program enabling” (PxxS.AP) status signal to be linked directly with the “advance program enabling” (PxxC.AP) control signal.

Signal flow to the process 1 advance program signals

Example

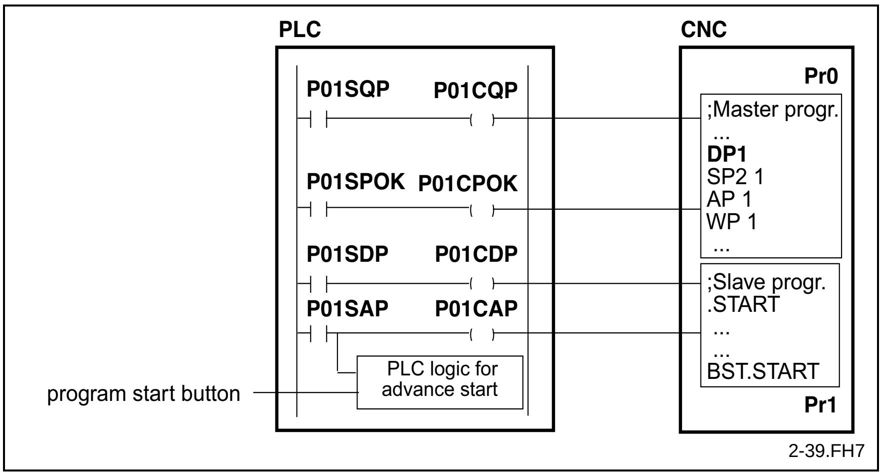


Fig. 1-58: Signal flow to the advance program signals

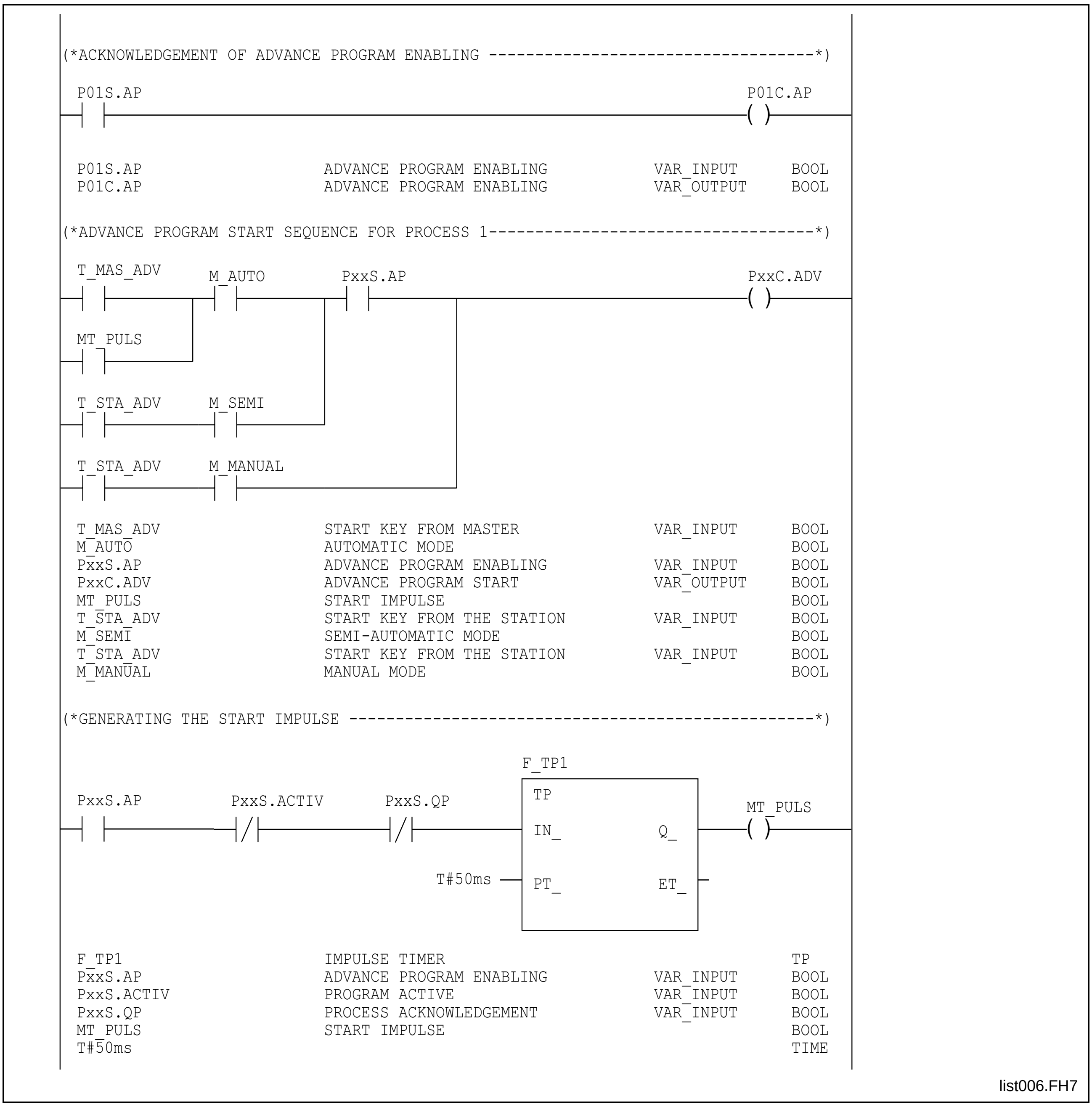


Fig. 1-59: Acknowledging advance program enable

Starting the advance program from the SLAVE process in “Automatic” mode:

- The advance program is started after the "APxx" NC command in the MASTER process has become active, the PLC has recognized that command via the "P01SAP" status signal, and the start impulse has changed to logic "1". If the program is stopped (by process stop, error, or cancellation of the process enabling signal, for example), the advance program may be restarted via the MASTER’s start key.

Starting the advance program from the SLAVE process in “Automatic” mode:

- The advance program is started when the MASTER process in the PLC recognizes the "APxx" NC command (P01SAP = 1), and the start key of the MASTER process is pressed.
- The NC program is interrupted as soon as the start key is released.

Starting the advance program from the SLAVE process in “Setup” mode:

- In “Setup” mode, the start is independent of the MASTER process. The SLAVE process executes the program when the start key is pressed.
- The NC program is interrupted as soon as the start key is released.

Advance start impulse

- In "Automatic" mode, a 0-1 impulse suffices for starting an NC program.
- This impulse is generated when an "APxx" is encountered in the MASTER process, no program is being executed, and the process has not yet been completed.
- A start impulse must be employed to be able to perform a restart by pressing the start button after an interruption. Otherwise, restarting the advance program would not be possible with a static "1".

Advance program timing:

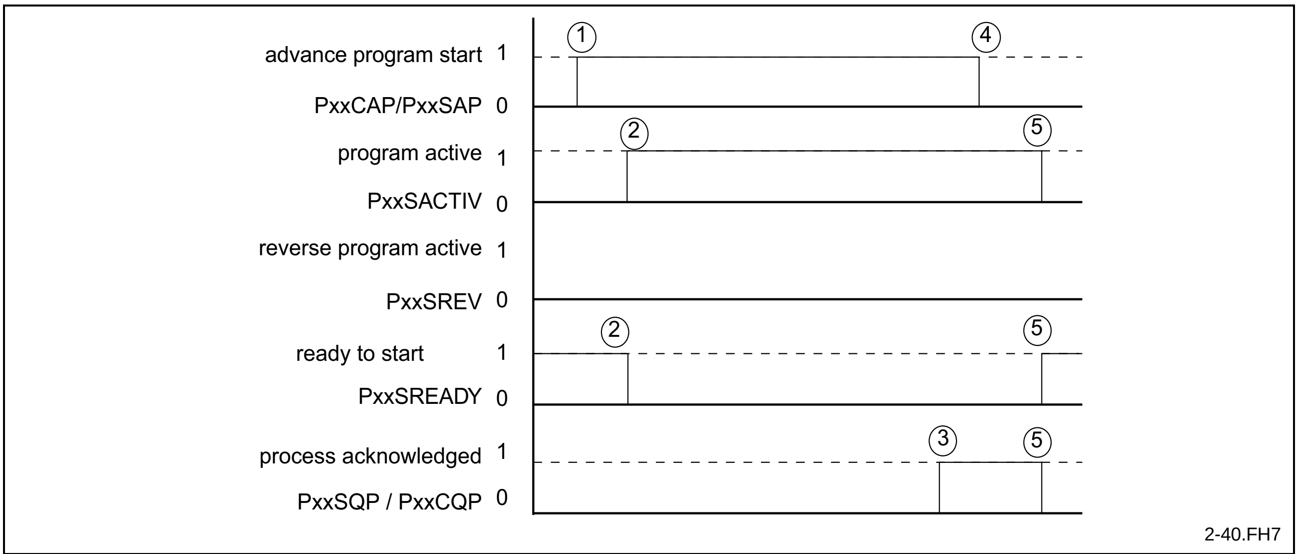


Fig. 1-60: Advance program timing

(1) The advance program of a SLAVE process is started in the NC program of the calling process (MASTER process). The “advance program enabling” (PxxS.AP) status signal is output to the PLC. This signal is used for generating the advance program start (PxxC.ADV) of the SLAVE process in the PLC.

(2) The SLAVE process begins with processing the NC program (PxxS.ACTIV = 1).

(3) The “acknowledge process” (PxxS.QP) status signal is used for informing the MASTER process of the termination of the NC program.

(4) The NC recognizes the acknowledgment of the NC program execution. The MASTER process responds by removing the "advance program enabling" status signal (PxxS.AP).

(5) The SLAVE process recognizes that the MASTER process has cleared the "advance program enabling" control signal. In response, the "acknowledge process" status signal (PxxS.QP) for the MASTER process is reset. The ready-to-start signal is re-issued when the NC program is terminated (PxxS.ACTIV = 0).

Reverse process enabling "PxxC.RP", "PxxS.RP"

Designation **PxxC.RP** = **P**rocess **xx** **C**ommand **R**everse **P**rocess Start

Control signal PLC →NC

PxxC.AP = 0: No reverse program enable

PxxC.AP = 1: Reverse program enable

Designation **PxxS.RP** = **P**rocess **xx** **S**tatus **R**everse **P**rocess Start

Status signal NC →PLC

PxxS.RP = 0: No reverse program enable

PxxS.RP = 1: Reverse program enable

Function In the NC program of the MASTER process, the "RP<process number>" command starts the advance program of a SLAVE process or mechanism.

Starting a reverse program of a SLAVE process in the NC program of the MASTER process:

```
.
.
DP1         ;Define process 1
SP1 2;      Preselect program no. 2
RP1         ;Start process 1 reverse program
WP1         ;Wait until process 1 is completed
.           ;Next NC program
.
```

The "reverse program enabling" (PxxS.RP) status signal is used for informing the PLC program that the MASTER process contains a reverse program start for this SLAVE process.

Reverse program enabling has two different effects in the PLC program:

- Reverse program enabling is required for generating the "reverse program start" (PxxC.REV) signal in "Automatic" mode and in "Semi-automatic" mode. In these two modes, reverse program start for the SLAVE process or mechanism is only permitted in the PLC program if a reverse program start has also been triggered in the NC program of the calling process (MASTER process) and if the "reverse program enabling" status signal (PxxS.RP) is present.
- Correct acknowledgment of the execution of the SLAVE process NC program requires the "reverse program enabling" (PxxS.RP) status signal to be linked directly with the "reverse program enabling" (PxxC.RP) control signal.

Example Signal flow of the reverse program enabling for process 1

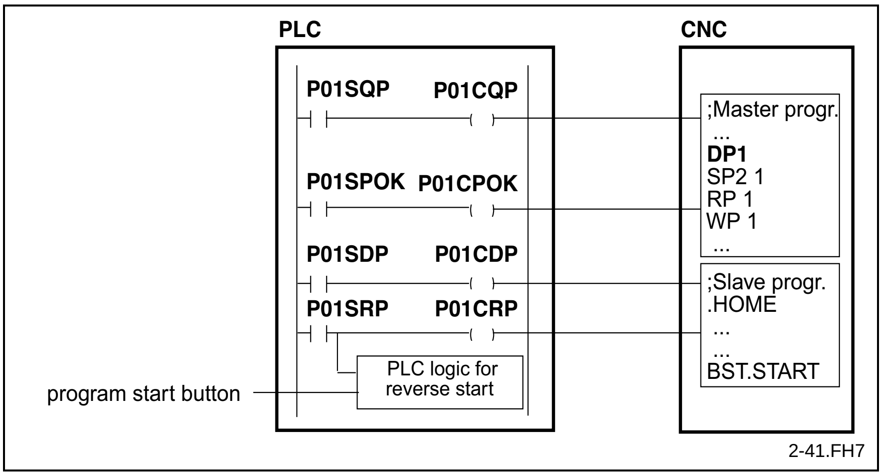


Fig. 1-61: Signal flow of reverse program enable

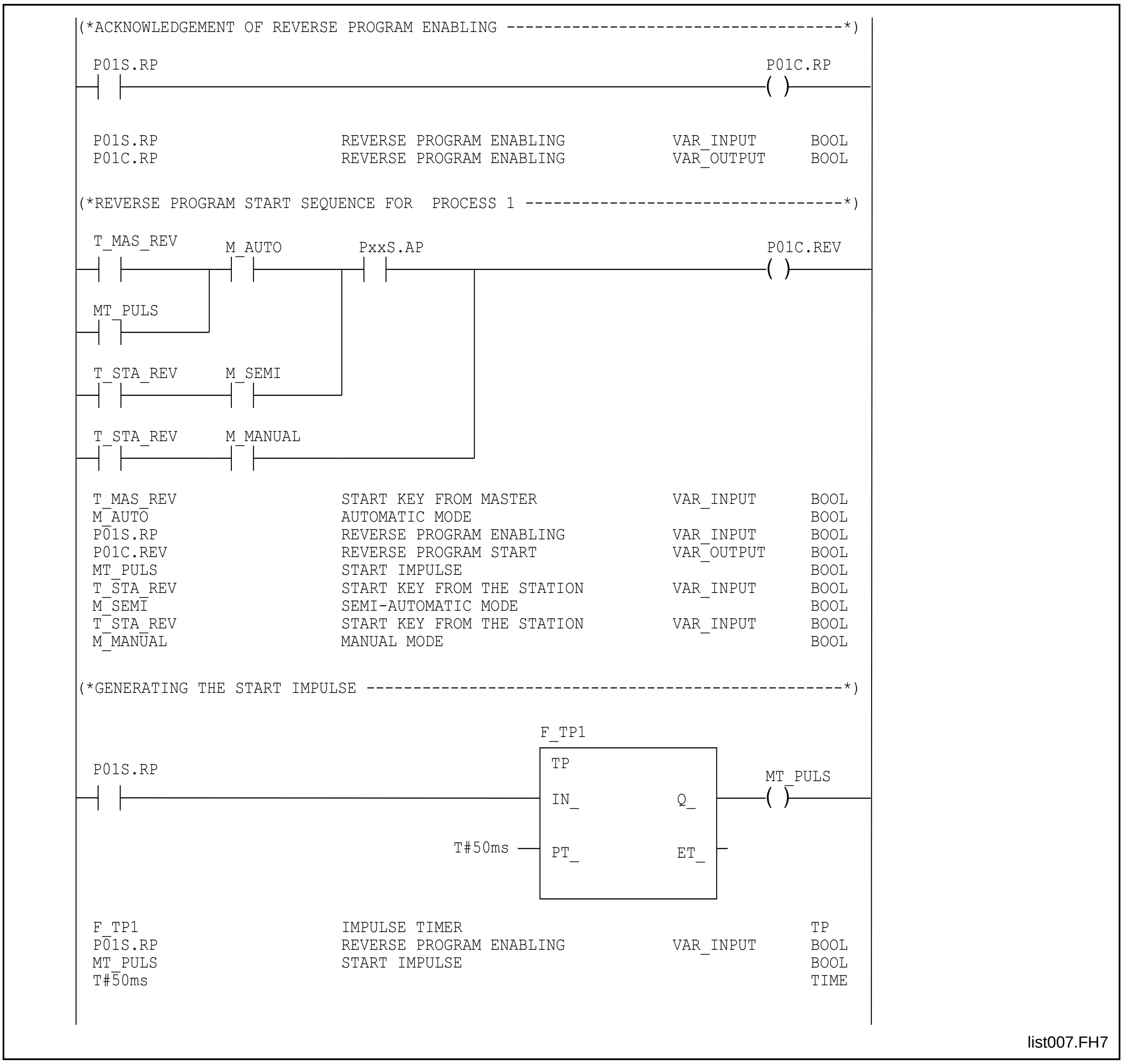


Fig. 1-62: Acknowledging reverse program enable

Starting the reverse program from the SLAVE process in "Automatic" mode:

- The reverse program is started after the "RPxx" NC command in the MASTER process has become active, the PLC has recognized that command via the "PxxS.RP" status signal, and the start impulse has changed to logic "1". If the program is stopped (by process stop, error, or cancellation of the process enabling signal, for example), the reverse program may be restarted via the MASTER’s start key.

Starting the reverse program from the SLAVE process in “Automatic” mode:

- The reverse program is started when the MASTER process in the PLC recognizes the "RPxx" NC command (PxxS.RP = 1) and the start key of the SLAVE process is pressed.
- The NC program is interrupted as soon as the start key is released.

Starting the reverse program from the SLAVE process in “Setup” mode:

- In “Setup” mode, the start is independent of the MASTER process. The SLAVE process executes the program when the start key is pressed.
- The NC program is interrupted as soon as the start key is released.

Reverse start impulse

- In "Automatic" mode, a 0-1 impulse suffices for starting an NC program.
- This impulse is generated when an "RPxx" is encountered in the MASTER process.
- A start impulse must be employed to be able to perform a restart by pressing the start button after an interruption. Otherwise, restarting the reverse program would not be possible with a static "1".

Reverse program timing:

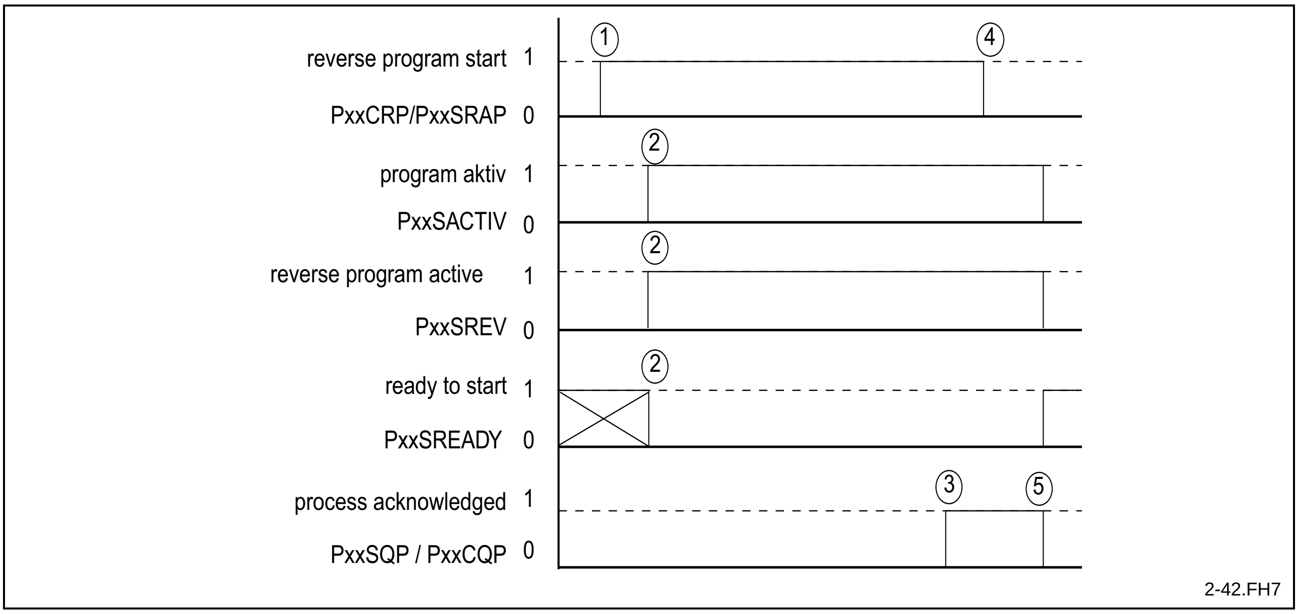


Fig. 1-63: Reverse program timing

(1) The reverse program of a SLAVE process is started in the NC program of the calling process (MASTER process). The "reverse program enabling" (PxxS.RP) status signal is output to the PLC. This signal is used for generating the reverse program start (PxxC.REV) of the SLAVE process in the PLC.

(2) The SLAVE process begins with processing the NC program (PxxS.ACTIV = 1).

(3) The “acknowledge process” (PxxS.QP) status signal is used for informing the MASTER process of the termination of the NC program.

(4) The NC recognizes the acknowledgment of the NC program execution. The MASTER process responds by removing the "reverse program enabling" status signal (PxxS.RP).

(5) The SLAVE process recognizes that the MASTER process has cleared the "reverse program enabling" control signal. In response, the "acknowledge process" status signal (PxxS.QP) for the MASTER process is reset. The ready-to-start signal is re-issued when the NC program is terminated (PxxS.ACTIV = 0).

Process acknowledgment "PxxC.QP", "PxxS.QP"

Designation **PxxC.QP** = **P**rocess **xx** **C**ommand **Q**uit **P**rocess

Control signal PLC →NC

PxxC.QP = 0: Program termination is not acknowledged

PxxC.QP = 1: Program acknowledgement terminated for SLAVE process

Designation **PxxS.QP** = **P**rocess **xx** **S**tatus **Q**uit **P**rocess

Status signal NC →PLC

PxxS.QP = 0: Program termination is not acknowledged

PxxS.QP = 1: Program acknowledgement terminated by SLAVE process

Function The NC outputs the "process acknowledgment" (PxxS.QP) status signal for a process whose advance or reverse program has been terminated in "Automatic" mode or "Semi-automatic" mode (BST or RET).

The advance or reverse programs were started by a higher-level process (MASTER process).

The NC clears the "process acknowledgment" status signal (PxxS.QP) when the "advance program enabling" (PxxC.AP) or "reverse program enabling" (PxxC.RP) control signal is removed in the PLC program.

The "process acknowledgment" control signal (PxxC.QP) is used for informing the MASTER process of the termination of a SLAVE process it has called.

Note: Please refer to the sections *"Advance program enabling"* and *"Reverse program enabling"* for examples of linking the "process acknowledgment" signal flow.

Part machined (part OK) "PxxC.POK", "PxxS.POK"

Designation **PxxC.POK** = **P**rocess **xx** **C**ommand **P**art is **OK**

Control signal PLC →NC

PxxC.POK = 0: Part not machined

PxxC.POK = 1: Part machined

Designation **PxxS.POK** = **P**rocess **xx** **S**tatus **P**art is **OK**

Status signal NC →PLC

PxxS.POK = 0: Part not machined

PxxS.POK = 1: Part machined

Function The "part machined" (PxxS.POK) status signal of a process that does not define any further processes is set if the "POK" command has been programmed in its NC program.

The "part machined" status signal (PxxS.POK) of a process in which further processes are defined is set if all defined processes issue the "part machined" control signal (PxxC.POK) during the execution of the "POK" NC command to signal that the part is being machined.

The NC clears the "part machined" status signal (PxxS.POK) when an NC program is newly started, a Control Reset is performed, or one of the axes of that process is jogged.

MASTER PROCESS and SLAVE Process in Different Controllers Shown as Example

Mechanism 8 of the MASTER NC controls process 2 of the SLAVE NC.

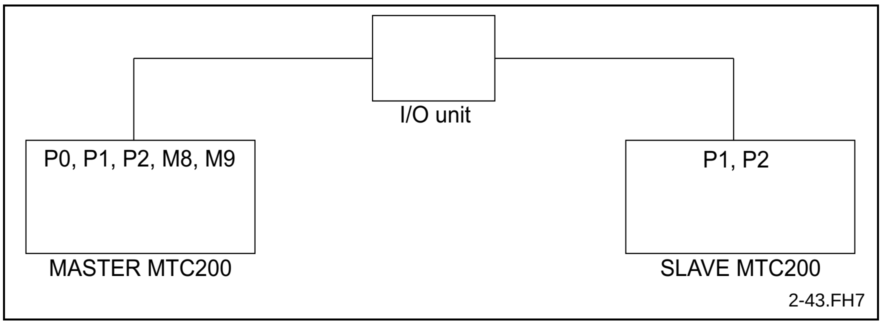


Fig. 1-64: MASTER PROCESS and SLAVE Process in Different Controllers Shown as Example

An I/O unit with corresponding wiring makes the connection between the MASTER NC and the SLAVE NC.

The following signals must be exchanged between the MASTER NC and the SLAVE NC:

MASTER NC	SLAVE NC	Comment
A_M08CDP →	E_M08SDP	;Define process
A_M08CAP →	E_M08SAP	;Advance program enabling
A_M08CRP →	E_M08SRP	;Reverse program enabling
E_M08SQP	←A_M08CQP	;Acknowledge process
E_M08SPOK	←A_M08CPOK	;Part machined
A_REVERSE →	E_REVERSE	;Main control panel "REVERSE" key
A_ADVANCE →	E_ADVANCE	;Main control panel "ADVANCE" key

Fig. 1-65: Exchange of signals between master/slave NC

The signal flow of the "define process", "advance process", "reverse process", "process acknowledgment" and "part machined" signals and of the two "ADVANCE" and "REVERSE" start keys on the main control panel are shown as examples in the following overviews.

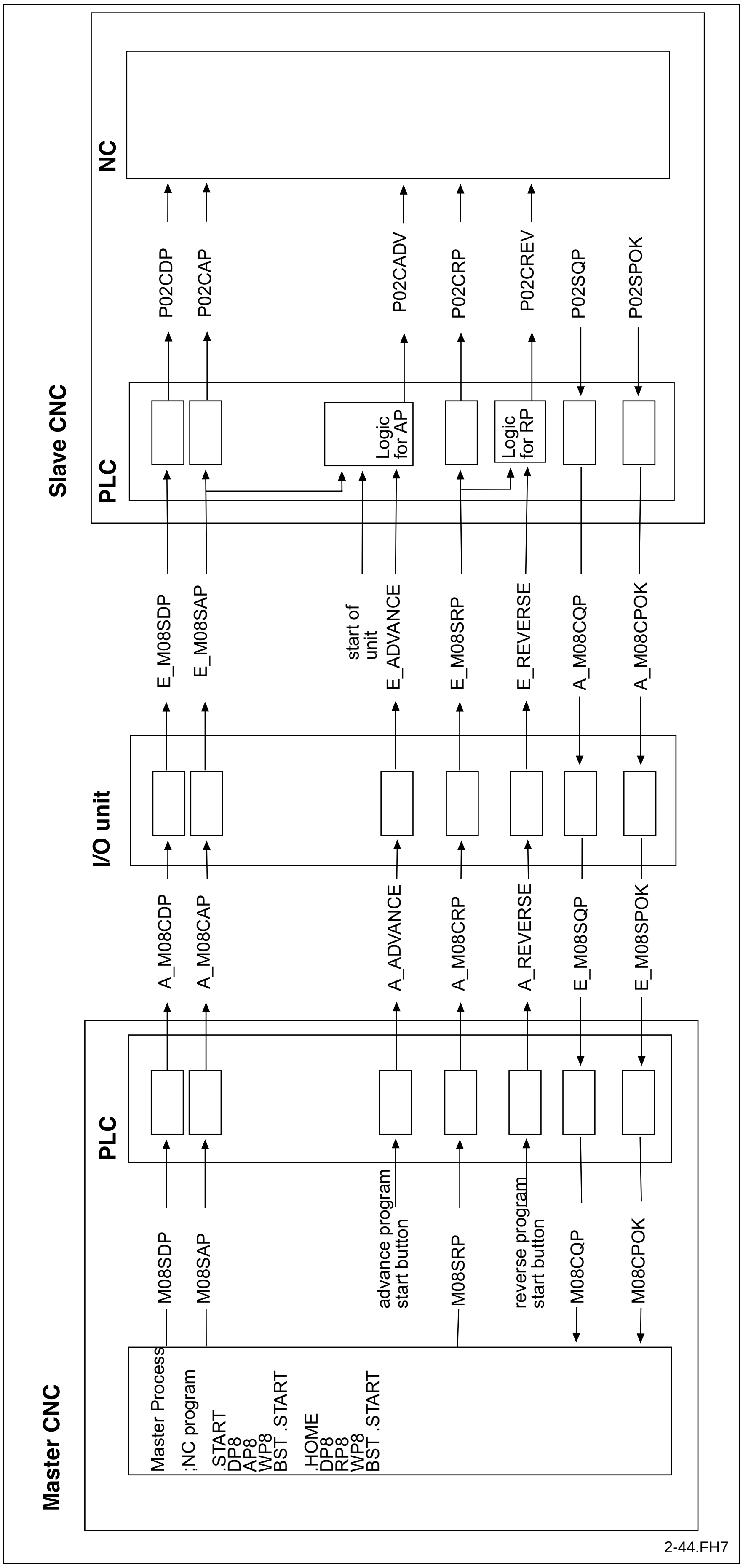
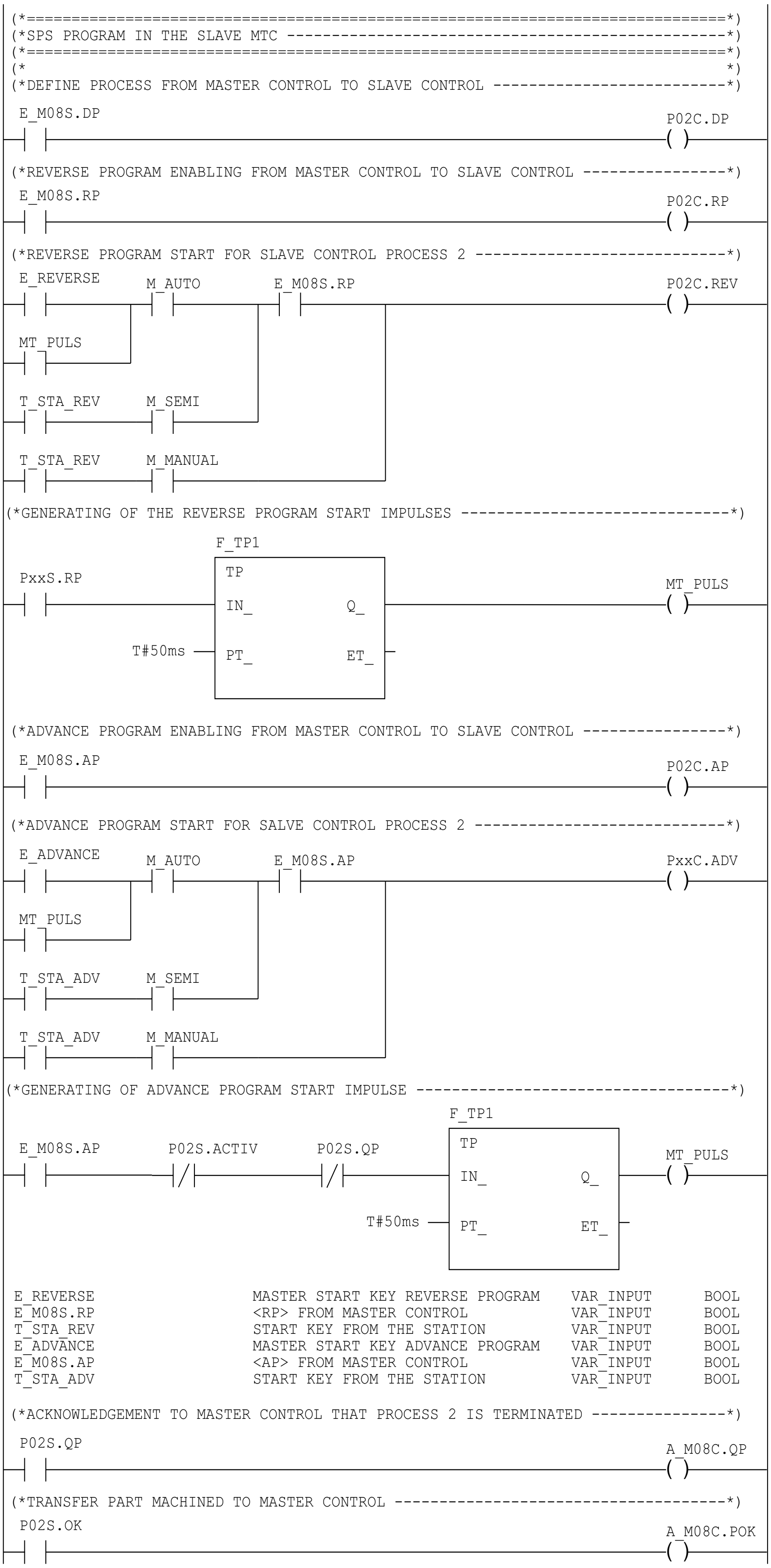


Fig. 1-66: Signal flow in the master/slave process

```
(*===================================================================================*)
(*SPS PROGRAM IN THE MASTER MTC                                                      *)
(*===================================================================================*)
(*                                                                                   *)
(*TRANSFER DEFINE PROCESS TO SLAVE CONTROL ---------------------------------------*)

 M08S.DP                                                                  A_M08C.DP
--| |------------------------------------------------------------------------( )----

M08S.DP                   DEFINE PROCESS 8                  VAR_INPUT      BOOL
A_M08C.DP                 <DP> TO SLAVE CONTROL             VAR_OUTPUT     BOOL

(*TRANSFER REVERSE PROGRAM ENABLING TO SLAVE CONTROL -----------------------------*)

 M08S.RP                                                                  A_M08C.RP
--| |------------------------------------------------------------------------( )----

M08S.RP                   START REVERSE PROG. PROCESS 8     VAR_INPUT      BOOL
A_M08C.RP                 <RP> TO SLAVE CONTROL             VAR_OUTPUT     BOOL

(*TRANSFER ADVANCE PROGRAM ENABLING TO SLAVE CONTROL -----------------------------*)

 M08S.AP                                                                  A_M08C.AP
--| |------------------------------------------------------------------------( )----

M08S.AP                   START ADVANCE PROG. PROCESS 8     VAR_INPUT      BOOL
A_M08C.AP                 <AP> TO SLAVE CONTROL             VAR_OUTPUT     BOOL

(*ACKNOWLEDGEMENT TO MASTER CONTROL ----------------------------------------------*)

 E_M08S.QP                                                                M08C.QP
--| |------------------------------------------------------------------------( )----

E_M08S.QP                 <QP> FROM SLAVE CONTROL           VAR_INPUT      BOOL
M08C.QP                   ACKNOWLEDGE PROCESS 8             VAR_OUTPUT     BOOL

(*PART MACHINED TO MASTER CONTROL ------------------------------------------------*)

 E_M08S.POK                                                               M08C.POK
--| |------------------------------------------------------------------------( )----

E_M08S.POK                <POK> FROM SLAVE CONTROL          VAR_INPUT      BOOL
M08C.POK                  PART MACHINED PROCESS 8           VAR_OUTPUT     BOOL

(*START SIGNALS FROM MAIN CONTROL PANEL TO SLAVE CONTROL -------------------------*)

 E_ADVANCE                                                                A_ADVANCE
--| |------------------------------------------------------------------------( )----

E_ADVANCE             MASTER START KEY ADVANCE PRG.         VAR_INPUT      BOOL
A_ADVANCE             ADVANCE START KEY FROM MASTER TO SLAVE VAR_OUTPUT    BOOL

 E_REVERSE                                                                A_REVERSE
--| |------------------------------------------------------------------------( )----

E_REVERSE             MASTER START KEY REVERSE PROGRAM      VAR_INPUT      BOOL
A_REVERSE             REVERSE START KEY FROM MASTER TO SLAVE VAR_OUTPUT    BOOL
```

list008.FH7

Fig. 1-67: PLC program of the Master MTC



list009.FH7

Fig. 1-68: PLC program of the Slave MTC

Tool magazine motion control

'**Tool magazine motion control**' is performed independently of the tool magazine's drive mode (PLC or NC controlled).

Tool magazine motion commands are used for controlling the movement from the NC side. Tool magazine control and status signals are used for controlling the movement from the PLC side.

The following figure shows the basic operation of tool magazine motion control:

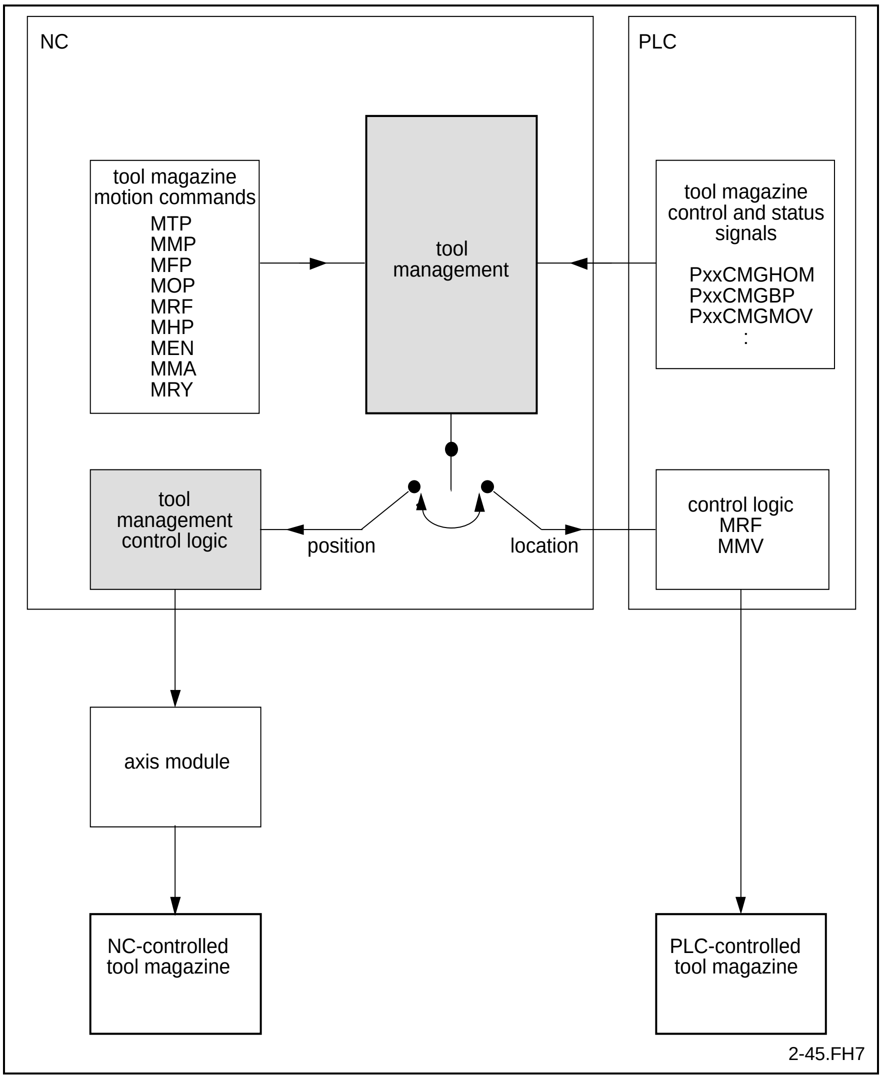


Fig. 1-69: Motion control of a tool magazine

The following notation is used for the tool magazine control signals and status signals of the PLC:

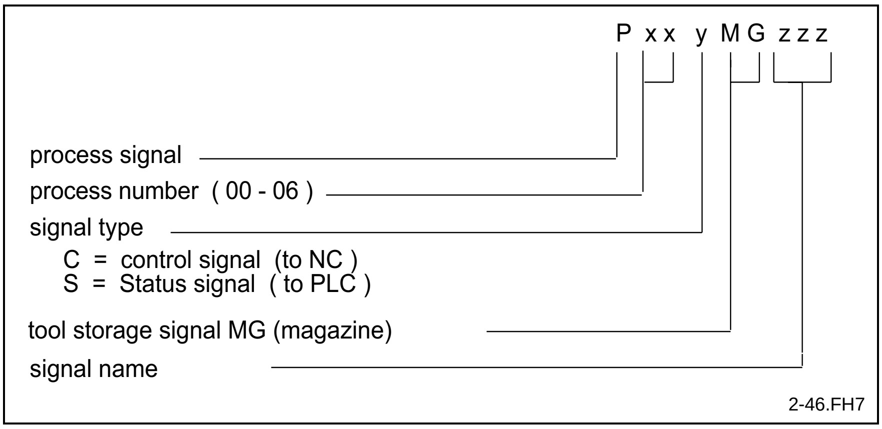


Fig. 1-70: Explanation of symbolic operands

Tool magazine control signals

The following control signals are available:

Tool magazine control signals	V22_20030123
Signal	**Meaning**
PxxC.MGENA	Enabling the tool magazine **P**rocess **xx** **C**ommand **Mag**azine **Ena**ble
PxxC.MGNSL	Operation without setup list **P**rocess **xx** **C**ommand **Mag**azine **n**o **S**etup **L**ist
PxxC.MGNTL	Deactivating the Life Count **P**rocess **xx** **C**ommand **Mag**azine **n**o **T**oo**l** Calculation
PxxC.MGMAN	Mode of tool storage unit **P**rocess **xx** **C**ommand **Mag**azine Mode **Man**ual
PxxC.MGHOM	Move tool storage unit to reference position **P**rocess **xx** **C**ommand **Mag**azine move to **Hom**ing
PxxC.MGBP	Tool storage to home position **P**rocess **xx** **C**ommand **Mag**azine **B**ase **P**osition
PxxC.MGPOS	Moving the tool storage unit by one location in the positive direction **P**rocess **xx** **C**ommand **Mag**azine **Pos**itive Direction
PxxC.MGNEG	Moving the tool storage unit by one location in the negative direction **P**rocess **xx** **C**ommand **Mag**azine **Neg**ative Direction
PxxC.MGAP	Actual position **P**rocess **xx** **C**ommand **Mag**azine **A**ctual **P**osition
PxxC.MGWTC	Switches off the automatic equipment check **P**rocess **xx** **C**ommand **Mag**azine **w**ithout **T**ool **C**heck
PxxC.MGITW	Tool check ignores worn tools **P**rocess **xx** **C**ommand **Mag**azine **I**gnore **T**ool **W**ornout
PxxC.MGJGn n = (0, 1)	Magazine jog mode with variable tool locations **P**rocess **xx** **C**ommand **Mag**azine **J**o**g**ging Mode Bit **n**

Wz_speicher_steuersignale_V22_klein_20030123.xls

Both control signals "PxxC.MGENA" and "PxxC.MGMAN" must remain statically applied to enable the tool magazine to be moved through the PLC.

With any other command, a positive flank of the control signal causes the function to be executed (i.e. the tool magazine to be moved).

Enabling the Tool Magazine "PxxC.MGENA"

Designation **PxxC.MGENA** = **P***rocess* **xx** **C***ommand* **M**a**g***azine* **En***able*

Control signal PLC →NC

PxxC.MGENA = 0: Tool magazine has not been enabled

PxxC.MGENA = 1: Tool magazine has been enabled

Valid in all modes.

Function Moving the tool magazine and performing a tool change requires this signal to be set. It is required for program mode and for manual mode.

Deactivating the signal during a movement of the tool magazine or a tool changing process immediately interrupts the movement. The interrupted movement is continued as soon as the signal has returned.

This permits security devices (such as a guard door) to be monitored. When such a device is opened, the "PxxC.MGENA" control signal causes the tool management to cancel axis enabling for an NC controlled tool magazine axis, and to reset the "PxxS.MGENA" status signal for the PLC.

Tool management shows the same response if, for example, an internal immediate stop (e.g. by changing the mode) or a fault occurs or process enabling is canceled.

Note: With PLC-controlled tool magazines and all tool changing procedures, the "PxxS.MGENA" signal must be taken into account in all interconnections that initiate a movement. Movement is permitted only if the enabling signal is active (=1).

Using the enabling signals:

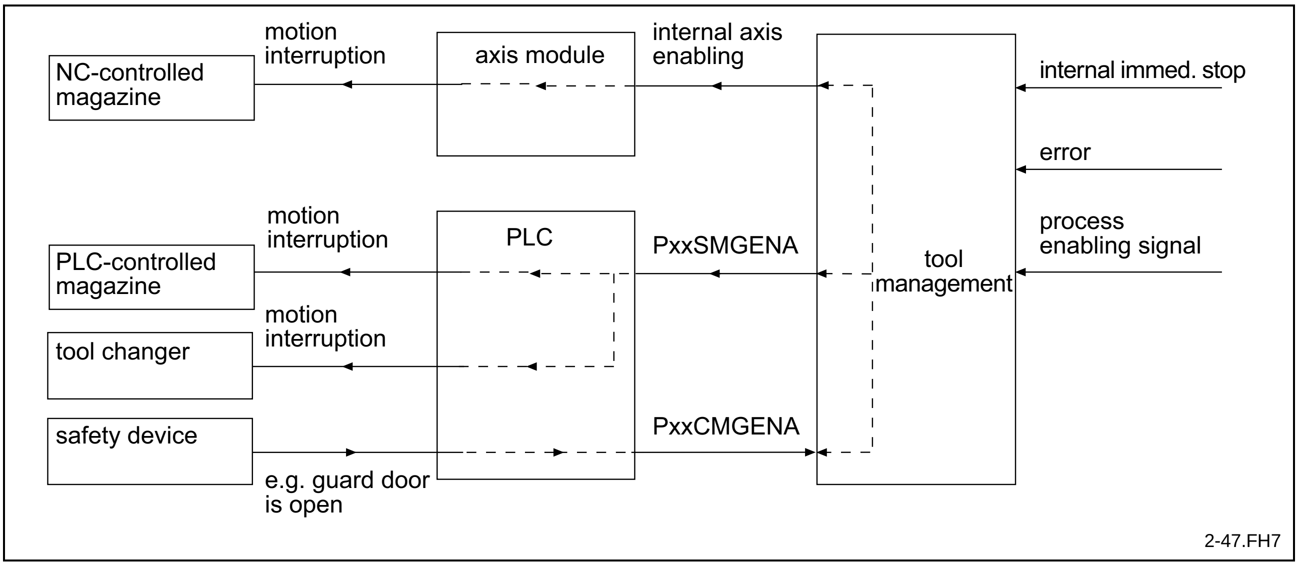


Fig. 1-71: Using the enabling signals

Deactivation of automatic tool check "PxxC.MGWTC"

Designation **PxxC.MGWTC** = **P**rocess **xx** **C**ommand **M**a**g**azine **w**ithout **T**ool **C**heck

Status signal PLC →NC

PxxC.MGWTC = 0: Automat. equipment check is omitted tool check can be executed

PxxC.MGWTC = 1: Automat. equipment check is omitted tool check is not executed

The signal is effective in all modes.

Function Generally, tool management does not perform an automatic tool check upon a restart if the "PxxC.MGWTC" interface signal has been set to logic "1".

Otherwise, an automatic tool check is executed according to the criteria in the section "Tool check" in "Rexroth MTC 200 Tool management projecting" or in "Rexroth MTC 200 NC Programming Instructions" document.

Operation without setup list "PxxC.MGNSL"

Designation **PxxC.MGNSL** = **P**rocess **xx** **C**ommand **M**a**g**azine **n**o **T**ool **S**etup **L**ist

Control signal PLC →NC

PxxC.MGNSL = 0: Tool assignment (T. no.) according to setup list

PxxC.MGNSL = 1: Operation without setup list, tool assignment (T. No.) according to setup list

The signal is effective in all modes.

Function If interface signal "PxxC.MGNSL" is set to logical "1", the NC internally generates an empty setup list that is invisible to the user; therein, it then creates an entry for each part number which occurs at least once in the current tool list. Any already existing setup list is left unconsidered. (See also section "Operation without setup list" in chapter "Setup lists and tool lists" in document "Rexroth MTC 200 Tool management project planning manual".)

Tool Check Ignores Wornout Tools "PxxC.MGITW"

Designation **PxxC.MGITW** = **P**rocess **xx** **C**ommand **M**a**g**azine **I**gnore **T**ool **W**ornout

Control signal PLC →NC

PxxC.MGITW	= 0	Take worn-out tool into account
PxxC.MGITW	= 1	Ignore worn-out tool

The signal is effective in all modes.

Function If PLC signal "PxxC.MGITW" has been deleted (log. "0"), the NC program terminates with an error message during the tool check if there is an alternate tool sequence that does not have any usable tool. Otherwise, no error message is given.

If PLC signal "PxxC.MGITW" has been set (log. "1"), the NC program terminates with an error message during the tool check if there is an alternate tool sequence that does not have any usable tool, including a tool that is worn or broken but otherwise usable. Otherwise, no error message is given.

For further information see section "Equipment check" in the document "Rexroth MTC 200 Tool management project planning manual" or in document "Rexroth MTC 200 NC programming instruction".

Deactivating the Life Count "PxxC.MGNTL"

Designation **PxxC.MGNTL** = **P**rocess **xx** **C**ommand **M**a**g**azine **n**o **T**ool **L**ife Calculation

Control signal PLC →NC

PxxC.MGNTL	= 0:	No influencing of life count
PxxC.MGNTL	= 1:	Life count is not enabled

The signal is effective in all modes.

Function The PLC can employ this signal to influence the NC's integrated tool life counter.

Tool management does not perform a life count for the currently active tool if this signal has been set to logic 1.

Tool magazine mode "PxxC.MGMAN"

Designation **PxxC.MGMAN** = **P**rocess **xx** **C**ommand **M**a**g**azine Mode **Man**ual

Control signal PLC →NC

PxxC.MGMAN	= 0:	Tool magazine is allocated to the NC program
PxxC.MGMAN	= 1:	Tool magazine is to be moved manually

Valid in all modes.

Function

Provided that the NC does not need the tool magazine, the tool magazine mode can be selected independently of the mode of the related NC process.

Block diagram of a tool magazine mode selection:

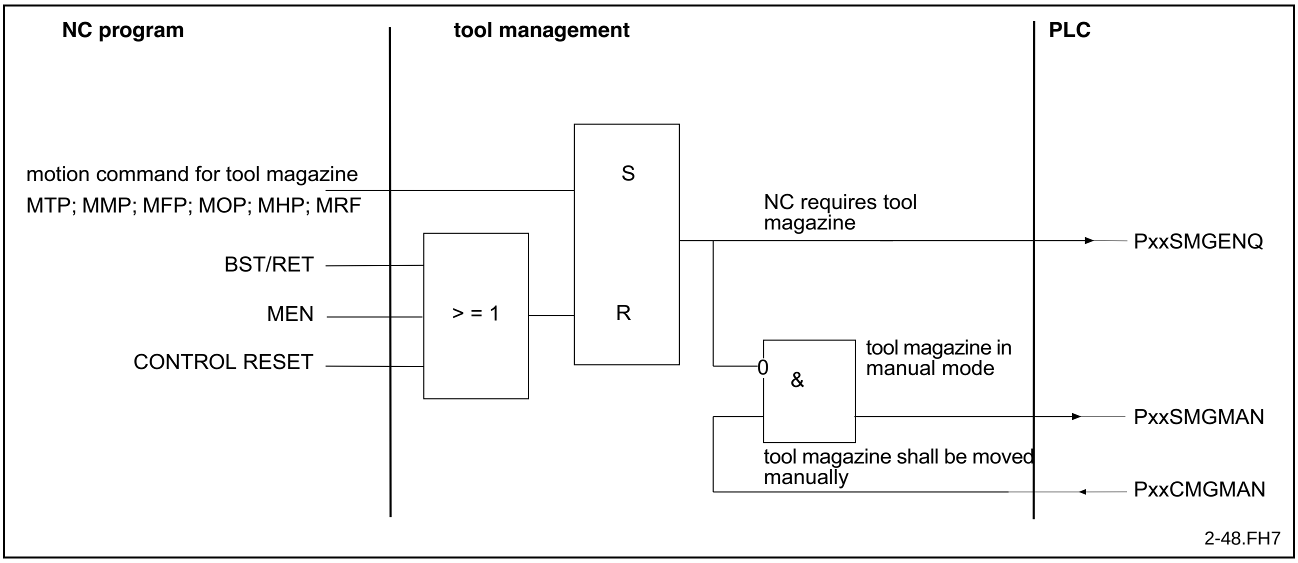


Fig. 1-72: Mode selection of a tool magazine

During program execution, a tool magazine can only be moved manually if the PLC has issued the "PxxC.MGMAN" control signal that tells the tool management that the tool magazine is to be moved manually.

The "PxxS.MGMAN" status signal indicates whether tool magazine control could be transferred to the PLC or whether the tool magazine is in manual mode. When tool magazine control is transferred to the PLC, it remains allocated to the PLC until the "PxxC.MGMAN" control signal is issued to switch over to program mode.

When the request "Move tool magazine manually" (PxxC.MGMAN: 0 →1) is issued if the "NC does not require the tool magazine, the mode is immediately changed to "tool magazine in manual mode". For this process, the NC executes the program up to the next tool magazine command and issues the related status message:
"Station waits for completion of magazine command".

When returning to program operation (PxxC.MGMAN: 1 →0), the status message in the diagnosis display is cleared, and the pending tool magazine command is processed.

Any tool magazine motion command ("MTP", "MMP", "MFP", "MOP", "MHP", MMA or "MRF") that is programmed in the NC program sets the "TOOL MAGAZINE IS NEEDED BY THE NC" request signal. That signal is reset by the NC commands "MEN" (Magazine Enable), "BST" (Branch with Stop) and "RET" (Return to Main Program).

A "control reset" also cancels the "Tool magazine is needed by the NC" request.

Within an NC process, tool storage unit axes are used as asynchronous axes. An NC-controlled tool magazine can therefore be manually controlled during automatic mode.

Homing of tool magazine "PxxC.MGHOM"

Designation

PxxC.MGHOM = **P**rocess **xx C**ommand **M**a**g**azine move to reference position (**Hom**ing)

Control signal PLC →NC

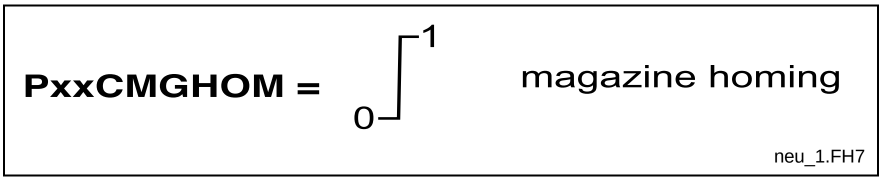


Fig. 1-73: Homing of tool magazine

Valid in all modes.

Function

A positive edge at the "PxxC.MGHOM" control signal initiates tool magazine homing.

With NC-controlled axes, tool management transfers the homing command to the servodrive axis module. With PLC-controlled tool magazines, in contrast, the "MRF" command is employed for transferring the homing command to the PLC.

Thus, the "Tool magazine homing" control signal permits program-independent establishment of a tool magazine reference.

When incremental encoders are employed for the tool magazine axis, the reference position must be established once after the Rexroth MTC 200 has been switched on. It makes sense for this function to be performed automatically in the reverse program when homing of the other NC axes is performed.

With NC-controlled tool magazines, the reference cam, the associated reference switch, and the zero impulse of the drive determine the reference position. The reference position of a PLC-controlled tool magazine is usually defined by a reference cam and the associated reference switch.

The reference position is not necessarily identical to the tool magazine's home position.

Moving the tool magazine to its base position "PxxC.MGBP"

Designation

PxxC.MGBP = **P**rocess **xx C**ommand **M**a**g**azine **B**ase **P**osition

Control signal PLC →NC

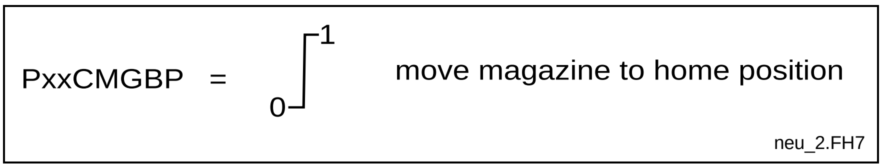


Fig. 1-74: Moving the tool magazine to its base position

Valid in all modes.

Function

A positive edge of the "PxxC.MGBP" control signal tells the tool management that the tool magazine must move to its base position. The tool management then initiates the required movement.

With an NC-controlled tool magazine, the "Cxx.013 Reference position" axis parameter can be used for placing the basic position at any location relative to the tool magazine's actual reference position. When the PLC sets that signal, the NC moves to the basic position by bringing "Location 1" to the reference mark.

Example "Magazine in basic position"

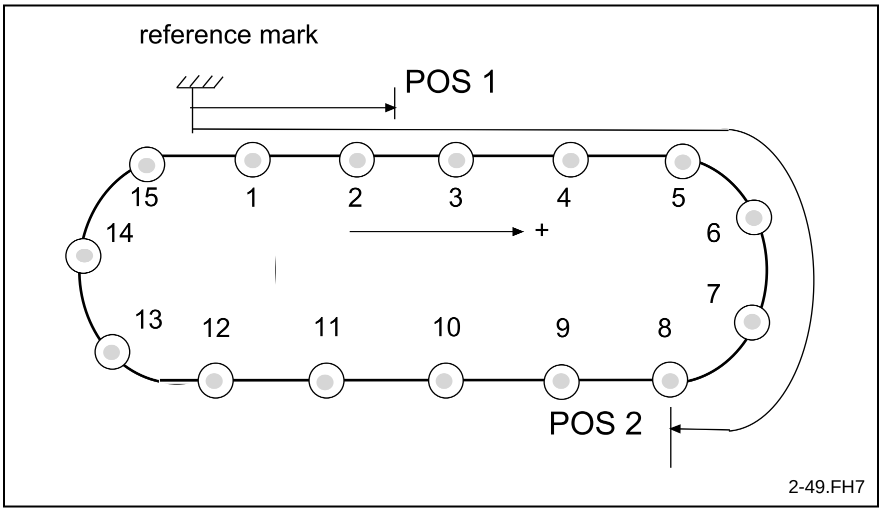


Fig. 1-75: Moving the tool magazine to its basic position

With a PLC-controlled tool magazine, tool management specifies "Location 1" " for the PLC via interface signal "PxxS.MGCP magazine position after the PLC has set the "PxxS.MGBP Move tool magazine to basic position" signal. For the PLC, moving to the basic position is equivalent to normal tool magazine positioning. It positions the tool magazine so that "Location 1" is located at the reference mark (Bxx.021 ... Bxx.024).

Moving the tool storage unit by one location in the positive direction "PxxC.MGPOS"

Designation **PxxC.MGPOS** = **P**rocess **xx** **C**ommand **Mag**azine **Pos**itive Direction

Control signal PLC →NC

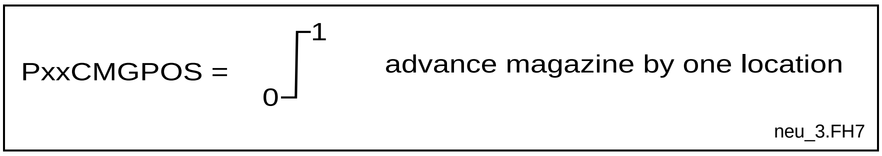


Fig. 1-76: Moving the tool storage unit by one location in the positive direction

Valid in all modes.

Function A positive edge of the "PxxC.MGPOS" control signal causes the tool management to move the tool magazine by one location so that the location with the next higher identification number sits at the reference mark.

In an endlessly rotating tool magazine (e.g. a chain-type magazine), "Location 1" will follow at the reference mark after the last location has been reached.

Similarly, with a non-endlessly rotating tool storage unit (e.g. a linear magazine), "Location 1" will be brought to the reference mark after the last location has been reached.

If the tool axis is not in a fixed magazine or turret location, the location next to the last valid fixed location is approaching upon jogging.

If the nonuniform tool location distribution has been activated with process parameter Bxx.072 "Page No. for variable position of locations", special regulations (refer to section "Magazine Jog Mode with Variable Tool Locations PxxC.MGJGn‘ ") apply to the jogging of tool axes.

Moving the tool magazine by one location in the negative direction "PxxC.MGNEG"

Designation **PxxC.MGNEG** = **P**rocess **xx C**ommand **M**a**g**azine **Neg**ative Direction

Control signal PLC →NC

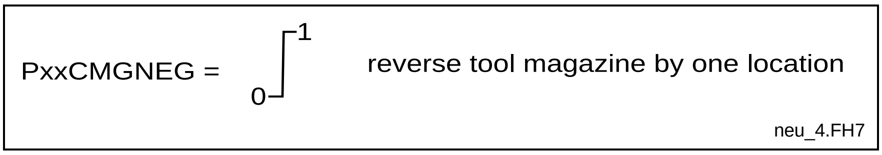


Fig. 1-77: Moving the tool storage unit by one location in the negative direction

Valid in all modes.

Function A positive edge of the "PxxC.MGNEG" control signal causes the tool management to move the tool magazine back by one location so that the location with the next lower number sits at the reference mark.

In an endlessly rotating tool storage unit (e.g. a chain-type magazine), the location with the highest identification number will follow at the reference mark after "Location 1" has been reached.

Similarly, with a non-endlessly rotating tool storage unit (e.g. a linear magazine), the location with the highest identification number will be brought to the reference mark after "Location 1" has been reached.

If the tool axis is not in a fixed magazine or turret location, the location next to the last valid fixed location is approaching upon jogging.

If the nonuniform tool location distribution has been activated with process parameter Bxx.072 "Page No. for variable position of locations", special regulations (refer to section "Magazine Jog Mode with Variable Tool Locations PxxC.MGJGn‘ ") apply to the jogging of tool axes.

Magazine Jog Mode with Variable Tool Locations "PxxC.MGJGn"

Designation **PxxC.MGJGn** = **P**rocess **xx C**ommand **M**a**g**azine **J**o**g**ging Mode Bit **n**

2 control signals PLC →NC

PxxC.MGJG0

PxxC.MGJG1

Valid in all process modes.

Function If the reference to a user machine data page (MDP) in the case of nonuniform tool location distribution has been activated with process parameter Bxx.072 (100 - 299) "Page No. for variable position of locations”, special regulations apply to the jogging of tool axes:

- The jogging of tool axes refers to one of the four change positions P1-P4 (see Fig. 4-11) that have been freely defined in the user machine data page. The values of the machine data page are used in place of the change positions freely defined with parameters Bxx.021 - Bxx.024.

- Using process control signals PxxC.MGJG0 and PxxC.MGJG1, the PLC defines the change position P1-P4 to which the jogging command refers.

The following table shows the effects of the two control signals PxxC.MGJG0 and PxxC.MGJG1:

PxxC.MGJG1	PxxC.MGJG0	Magazine jogging refers to change position …
0	0	Axis position for changing position P1
0	1	Axis position for changing position P2
1	0	Axis position for changing position P3
1	1	Axis position for changing position P4

PxxC_MGJGn_V22_20021126.xls

Fig. 1-78: Effect of PxxC.MGJG0 and PxxC.MGJG1

- With process control signals PxxC.MGPOS (Positive jogging) / PxxC.MGNEG (Negative jogging), the next/previous tool location is moved to the selected position.

Example

Machine data page (MDP)		
Control variable LV1	**Value**	**Meaning**
-4	360	Axis position for changing position P4
-3	280	Axis position for changing position P3
-2	180	Axis position for changing position P2
-1	95	Axis position for changing position P1
0		Axis position for home position (MHP)
1	40	Axis position for magazine location 1
2	72	Axis position for magazine location 2
3	110	Axis position for magazine location 3
4	144	Axis position for magazine location 4
5	182	Axis position for magazine location 5
6	218	Axis position for magazine location 6
7	254	Axis position for magazine location 7
8	290	Axis position for magazine location 8
9	326	Axis position for magazine location 9
10	360	Axis position for magazine location 10

Bxx_072_Werkzeugwechselposition_variable_V22_20030123.xls

Fig. 1-79: Example of a user machine data page for Bxx.072

The words "previous" and "next" tool location refer to the control variable number of the tool location.

- After the change position has been altered (e.g. P1→P2, P2 ? P1), the tool location whose control variable number occurs on the last valid location (based on the old change position, <u>depending on the direction of jogging</u>) is brought to the **new** change position during subsequent jogging. Accordingly, moving the tool storage unit by several tool positions is possible.

Note: Upon activation of the variable locations for NC controlled tool magazines, the parameters Bxx.021..Bxx.024 "Position 1..4“ cease to have an effect. Instead, the logic positions 1 to 4 are read from the machine data page defined in the process parameter Bxx.072 "Page no. for variable locations“ [travel variable –4 (position 4)... -1 (position 1)].

Actual position "PxxC.MGAP"

Designation

PxxC.MGAP = **P**rocess **xx** **C**ommand **M**a**g**azine **A**ctual **P**osition

Control signal PLC →NC

PxxC.MGAP Actual position

Valid in all modes.

Note: Only required for PLC-controlled tool magazines.

Function

The PLC employs this interface signal to inform tool management of the current position of the tool magazine in relation to the reference mark.

With PLC-controlled tool magazines, position control must be implemented in the PLC. For this purpose, the PLC tool management transfers the command position in the "PxxS.MGCP" interface signal, and employs the "MMV" PLC function to indicate a necessary movement. The PLC must then ensure on its own that the specified command position is reached without delay.

While the tool magazine is moving, the PLC must employ the "PxxC.MGAP" control signal for continually notifying the tool management of the actual position.

Once the tool magazine has reached the command position (command position = actual position), the PLC must use the "MMV_Q" function to acknowledge that state to the tool management.

Note: After the "MMV" motion command has been acknowledged, the tool management does not verify whether the command position has actually been reached.

In a subsequent tool transfer between the magazine and the spindle/gripper, the tool management always uses the location that is returned in "PxxC.MGAP" for the logic tool transfer.

Tool storage unit status signals

Tool management employs status signals to provide the PLC with information about the status of the tool magazine, regardless of the tool magazine's drive mode. The following status signals are available:

Tool magazine status signals	V22_20021104
Signal	**Meaning**
PxxS.MGENA	Enabling the tool magazine **P**rocess **xx** **S**tatus **M**a**g**azine **Ena**ble
PxxS.MGREQ	Requesting the tool storage unit from the NC **P**rocess **xx** **S**tatus **M**a**g**azine **Req**uest
PxxS.MGMOV	Tool storage unit in motion **P**rocess **xx** **S**tatus **M**a**g**azine **Mov**e
PxxS.MGMAN	Tool storage unit in program or manual mode **P**rocess **xx** **S**tatus **M**a**g**azine Mode **Man**ual
PxxS.MGBP	Tool magazine in home position **P**rocess **xx** **S**tatus **M**a**g**azine **B**ase **P**osition
PxxS.MGCP	Command position **P**rocess **xx** **S**tatus **M**a**g**azine **C**ommand **P**osition
PxxS.MGTWO	Life count monitoring: tool worn out **P**rocess **xx** **S**tatus **M**a**g**azine **T**ool **W**orn **O**ut
PxxS.MGWRN	Life count monitoring: warning limit reached **P**rocess **xx** **S**tatus **M**a**g**azine **W**a**rn**ing
PxxS.MGERR	Tool error status **P**rocess **xx** **S**tatus **M**a**g**azine **Err**or

Wz_speicher_statussignale_V22_klein_20021104.xls

Enabling the tool magazine "PxxC.MGENA"

Designation

PxxS.MGENA = **P**rocess **xx** **S**tatus **Ena**ble for **M**a**g**azine

Status signal NC →PLC

PxxS.MGENA = 0: No motion permitted

PxxS.MGENA = 1: Motion for tool magazine enabled

Valid in all modes.

Function

Only required for PLC-controlled tool magazines. The signal is set to logic "1" when a movement is initiated via tool management.

The tool management can employ the "PxxS.MGENA" status signal to stop a movement of the tool magazine and the sequence of the tool changing process at any time. Reasons to do this may come from a safety device (such as a guard door) that causes a "PxxC.MGENA" control signal to be issued to the tool management. An error, removing the process enabling signal, or an internal immediate stop (e.g. by changing the mode) may also cause the "PxxS.MGENA" enabling signal to be removed.

Note: With PLC-controlled tool magazines and in all tool changing processes, the "PxxS.MGENA" signal must be taken into account in all interconnections which initiate a movement (e.g. actuating the valves which release a movement). Movement may occur only if the "PxxS.MGENA" release is active.

Using the enabling signals:

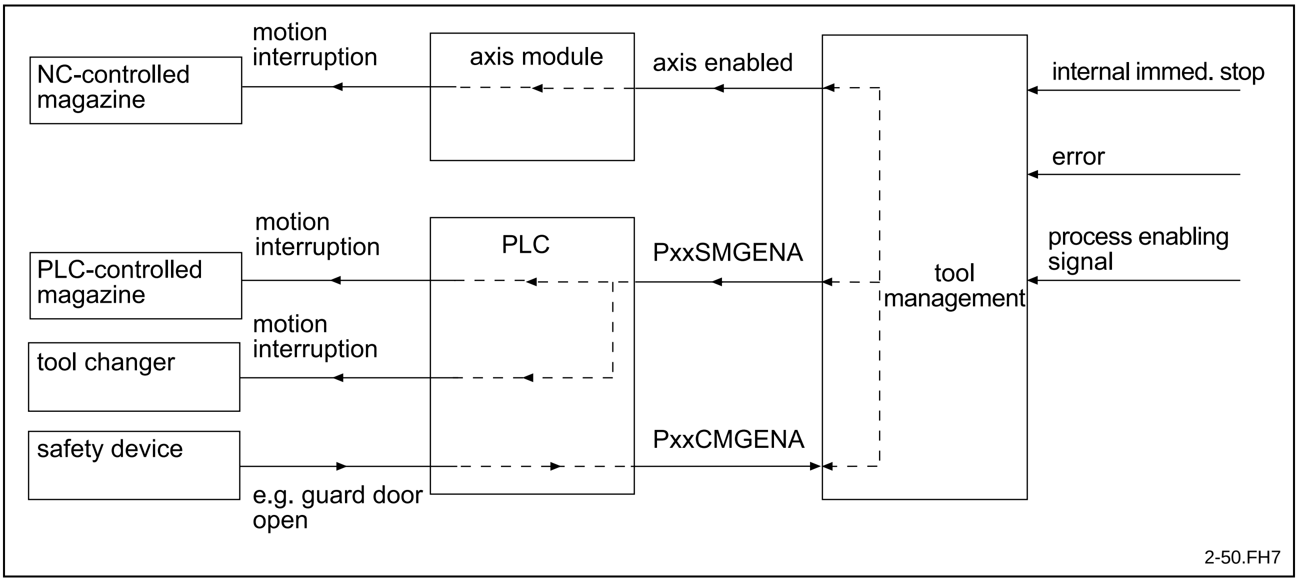


Fig. 1-80: Enabling the tool magazine

Tool magazine request from the NC "PxxS.MGREQ"

Designation **PxxS.MGREQ** = **P**rocess **xx** **S**tatus **M**a**g**azine **Req**uest

Status signal NC →PLC

PxxS.MGREQ = 0: No request from the NC

PxxS.MGREQ = 1: NC needs tool magazine

Valid in all modes.

Function The ""PxxS.MGREQ" status signal is set if a tool magazine movement command addresses the tool management within an executing program.

Tool magazine request from the NC:

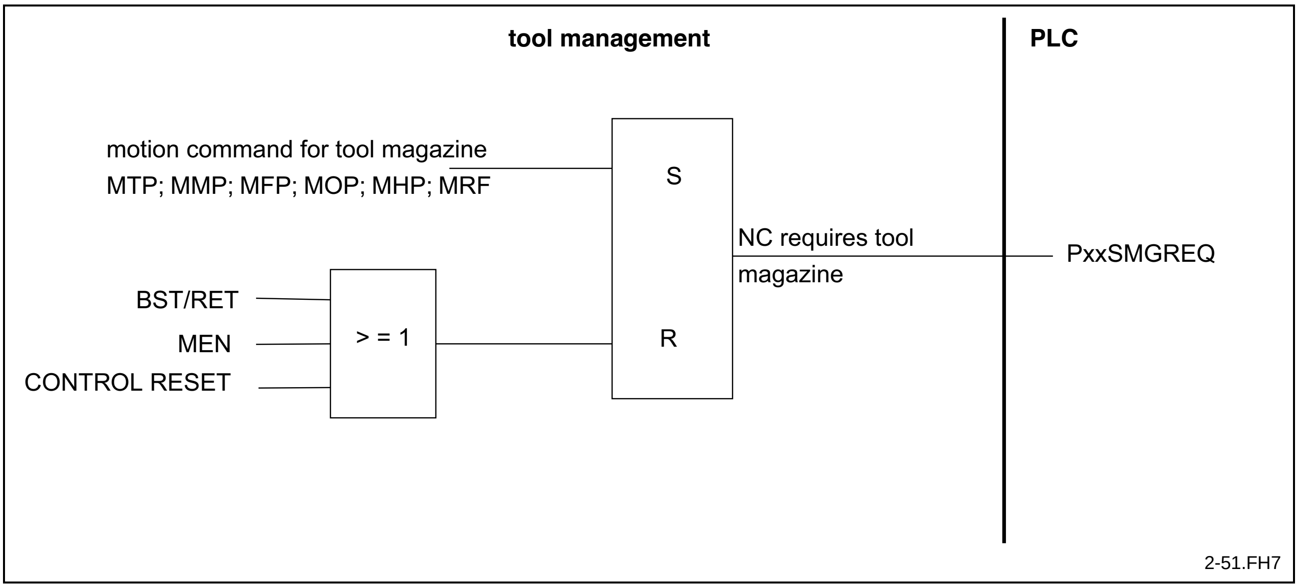


Fig. 1-81: Requesting the tool magazine

During program mode, the "MEN" (Magazine enable), "BST" or "RET" command can be used for resetting the tool magazine request.

A control reset after a program stop also clears the tool magazine request from tool management.

The PLC is only able to switch the tool magazine of a process to manual mode after the "NC needs tool magazine" request signal has been removed.

Tool magazine in motion "PxxS.MGMOV"

Designation **PxxS.MGMOV** = **P**rocess **xx** **S**tatus **M**a**g**azine **Mov**e

Status signal NC →PLC

PxxS.MGMOV = 0: Tool magazine is at standstill

PxxS.MGMOV = 1: Tool magazine is in motion

Valid in all modes.

Function The "PxxS.MGMOV" status signal is output whenever the tool magazine is moving. It is insignificant whether the movement has been initiated by an NC instruction or by a control signal from the PLC.

Tool management ignores any new PLC tool storage unit commands as long as this status signal is active.

Tool magazine in program or manual mode "PxxS.MGMAN"

Designation **PxxS.MGMAN** = **P**rocess **xx** **S**tatus **M**a**g**azine **Man**ual

Status signal NC →PLC

PxxS.MGMAN = 0: Tool magazine is allocated to the NC

PxxS.MGMAN = 1: Tool magazine is allocated to the PLC

Valid in all modes.

Function

The "PxxS.MGMAN" status signal indicates whether or not the tool magazine control has really been transferred to the PLC.

The tool management sets the "PxxS.MGMAN" signal when the PLC sets the "tool magazine is to be moved manually" request signal ("PxxC.MGMAN" control signal), and there is no tool magazine request ("PxxS.MGREQ" status signal) from the NC at the same time.

Tool magazine is in base position "PxxS.MGBP"

Designation

PxxS.MGBP = **P**rocess **xx** **S**tatus **M**a**g**azine is in **B**ase **P**osition

Status signal NC →PLC

PxxS.MGBP	= 0:	The tool magazine is not in its basic position
PxxS.MGBP	= 1:	The tool magazine is in its base position

Valid in all modes.

Function

The "PxxS.MGBP" status signal is output when the tool magazine is in its home position (location 1 at the reference mark).

This signal may be used for enabling machining if machining is possible only when the tool magazine is in its basic position (e.g. a tool magazine that is swiveled into the working space).

Command position "PxxS.MGCP"

Designation

PxxS.MGCP = **P**rocess **xx** **S**tatus **M**a**g**azine **C**ommand **P**osition

Status signal NC →PLC

PxxS.MGCP Command position

Valid in all modes (except with NC command "MMA").

Note: Only required for PLC-controlled tool magazines.

Function

The tool management function employs the "PxxS.MGCP" memory to transfer to the PLC the command position (relative to the reference mark) which the PLC must adopt when it is told to do so via the "MMV" motion function.

The PLC must then ensure on its own that the specified command position is reached without delay.

While the tool magazine is moving, the PLC must employ the "PxxC.MGAP" control signal for continually notifying the tool management of the actual position.

Once the tool magazine has reached the command position (command position = actual position), the PLC must use the "MMV_Q" command to acknowledge the completion of the movement to the tool management function.

Command and actual position of PLC-controlled tool magazines:

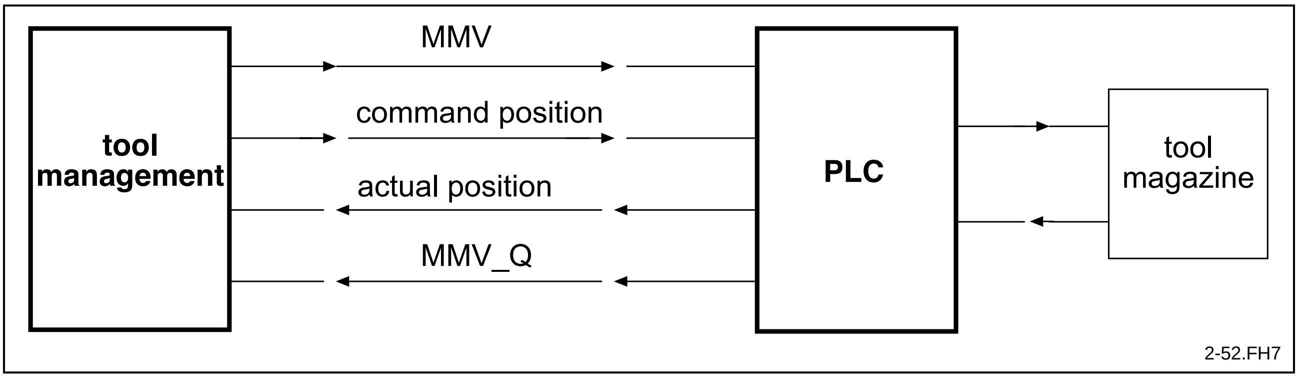


Fig. 1-82: Command and actual position of PLC-controlled tool magazines:

Example A PLC-controlled turret has 8 locations. Each location contains a tool.

Positioning a tool:

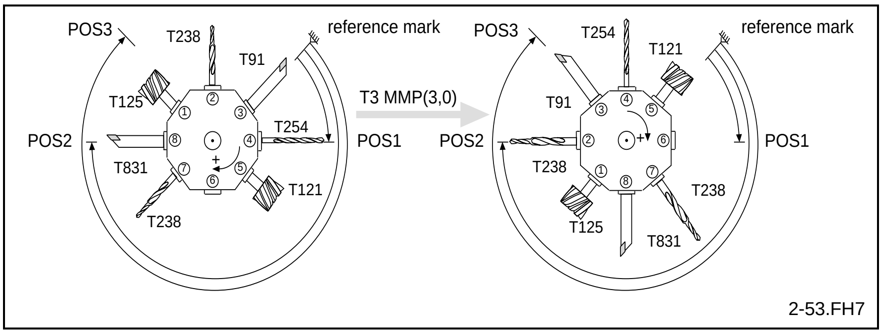


Fig. 1-83: Positioning of a tool

The "T91 MTP (2)" command in the NC program calls up the "T91" ID cutting tool and puts it in working position.

Tool management first locates the tool in the magazine. Then it determines the command position relative to the reference mark that must be transferred to the PLC in order to get the "T91" tool to machining position "1".

To get the "T91" tool to position "2", the tool management transfers location 6 as command position in the "PxxS.MGCP" interface signal, and sets the "MMV" command.

During the swiveling process, the PLC employs the "PxxC.MGAP" signal to continually update the current location in relation to the reference mark.

Once positioning has been completed, the PLC employs the 'MMV_Q' standard interface function for acknowledgment.

Life monitoring: tool worn out "PxxS.MGTWO"

Designation **PxxS.MGTWO** = **P***rocess* **xx** **S***tatus* **M***a***g***azine* **T***ool* **W***orn* **O***ut*

Status signal NC →PLC

PxxS.MGTWO = 0: The tools can still be used

PxxS.MGTWO = 1: At least one tool is worn out

Valid in all modes.

Function Effect of the "tool worn out" signal:

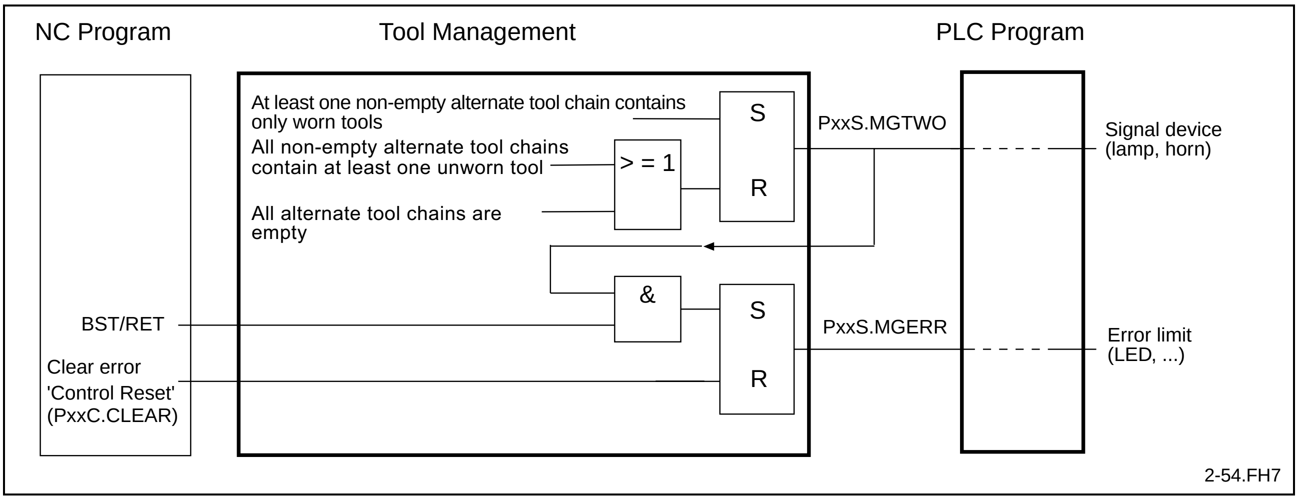


Fig. 1-84: Effect of the signal ”Tool worn out”

Tool management set the PxxS.MGTWO signal set when there is at least one non-empty related tool chain where all the tools are worn.
If all the non-empty alternate tool chains have at least one unworn tool, PxxS.MGTWO is cleared.
If all alternate tool chains are empty, PxxS.MGTWO is also cleared.

The signal is updated:

- during a transition to a different tool edge,
- when an edge is requested again
- when the tool is brought back to the magazine (tool storage unit = magazine),
- when the tool is rotated out of the machining position (tool storage = turret),
- when a tool is canceled using T0 (tool storage unit = turret or no tool storage unit present).
- when modifying the data of a tool with PLC function module TLD_WR or with NC command TLD
- when deleting a tool with PLC function module TL_DELETE or via the interface,
- when resetting a tool to a remaining level time of 100% with PLC function module TL_RESET or via the interface, and
- upon online modification of a tool's data via the interface.

Together with setting the "Tool worn out" signal, the tool management sets the "PxxS.MGERR" tool error status.

Note: Until Firmware version 21VRS, both interface signals PxxS.MGTWO and PxxS.MGWRN were generally reset during equipment check, also if for example all tools of an alternate tool chain were worn out.
As of Firmware version 22VRS, the behavior of both interface signals changed as follows:

- Both signals PxxS.MGTWO and PxxS.MGWRN always represent the current state of the non-empty alternate tool chains.
- Prior to updating PxxXMGWRN and PxxS.MGTWO, the two tool edge status bits "tool edge warning limit reached" and "tool edge worn out" for all tool edges of the process and the tool edge status bits "tool warning limit reached" and "tool worn out" have been addressed.

- In the equipment check, the two tool status bits "Tool warning limit reached" and "Tool worn out" are initialized for all tools in the tool list; based on this, PxxS.MGWRN and PxxS.MGTWO are initialized.

Life monitoring: warning limit reached "PxxS.MGWRN"

Designation

PxxS.MGWRN = **P**rocess **xx S**tatus **M**a**g**azine Tool **W**a**rn**ing

Status signal NC →PLC

PxxS.MGWRN = 0: Tool has not yet reached warning limit

PxxS.MGWRN = 1: Tool has reached warning limit

Valid in all modes.

Function

Tool management set the PxxS.MGWRN signal set when there is at least one non-empty related tool chain where all the tools have reached the warning limit.
If all non-empty related tool chains include at least one tool which has not yet reached the warning limit, PxxS.MGWRN is cleared.
If all alternate tool chains are empty, PxxS.MGWRN is also cleared.

The signal is updated:

- during a transition to a different tool edge,
- when an edge is requested again
- when the tool is brought back to the magazine (tool storage unit = magazine),
- when the tool is rotated out of the machining position (tool storage = turret),
- when a tool is canceled using T0 (tool storage unit = turret or no tool storage unit present).
- when modifying the data of a tool with PLC function module TLD_WR or with NC command TLD
- when deleting a tool with PLC function module TL_DELETE or via the interface,
- when resetting a tool to a remaining level time of 100% with PLC function module TL_RESET or via the interface, and

during Online modification of tool data via interface, the effect of the signal "Warning limit" was reached.

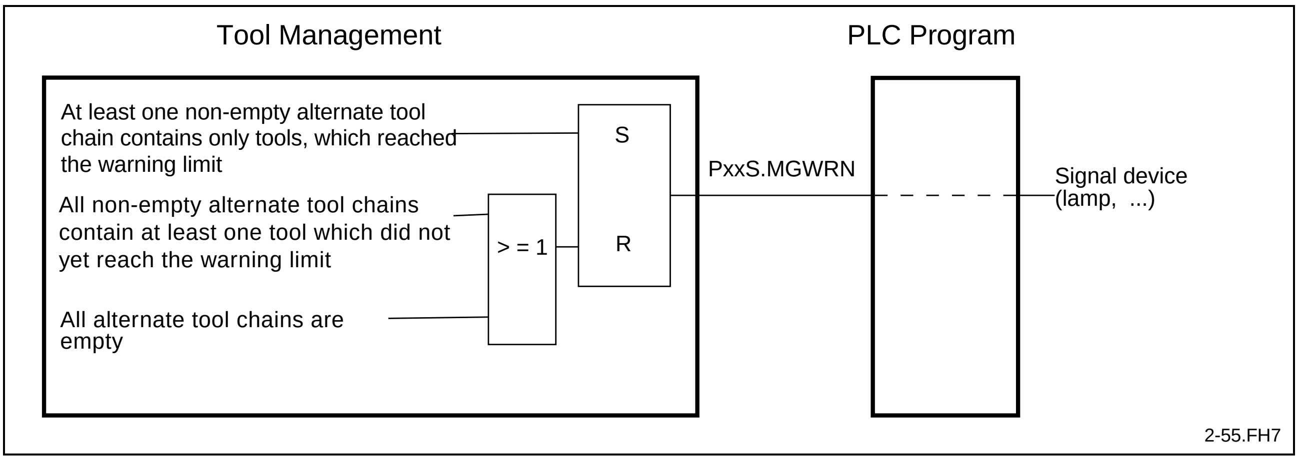


Fig. 1-85: Effect of the signal ”Warning limit reached”

Note: Until Firmware version 21VRS, both interface signals PxxS.MGTWO and PxxS.MGWRN were generally reset during equipment check, also if for example all tools of an alternate tool chain were worn out. As of Firmware version 22VRS, the behavior of both interface signals changed as follows:

- Both signals PxxS.MGTWO and PxxS.MGWRN always represent the current state of the non-empty alternate tool chains.
- Prior to updating PxxXMGWRN and PxxS.MGTWO, the two tool edge status bits "tool edge warning limit reached" and "tool edge worn out" for all tool edges of the process and the tool edge status bits "tool warning limit reached" and "tool worn out" have been addressed.
- In the equipment check, the two tool status bits "Tool warning limit reached" and "Tool worn out" are initialized for all tools in the tool list; based on this, PxxS.MGWRN and PxxS.MGTWO are initialized.

Tool error status "PxxS.MGERR"

Designation

PxxS.MGERR = **P**rocess **xx** **S**tatus **M**a**g**azine **Err**or

Status signal NC →PLC

PxxS.MGERR = 0: No error

PxxS.MGERR = 1: There is an error

Valid in all modes.

Function

Tool management employs this signal to inform the PLC about all detected errors.

The "Tool error status" signal is set when

- the last tool in a group of alternate tools is worn out at program end;
- the tools in the magazine do not match the specifications in the setup list; or
- a tool cannot be found when it is called via the "T word".

After the next "BST" or "RET", at the latest, the "Pxx.SMGERR" status signal will lead to setting of the "PxxS.ERROR" process error.

Effect of the "Tool error status" signal:

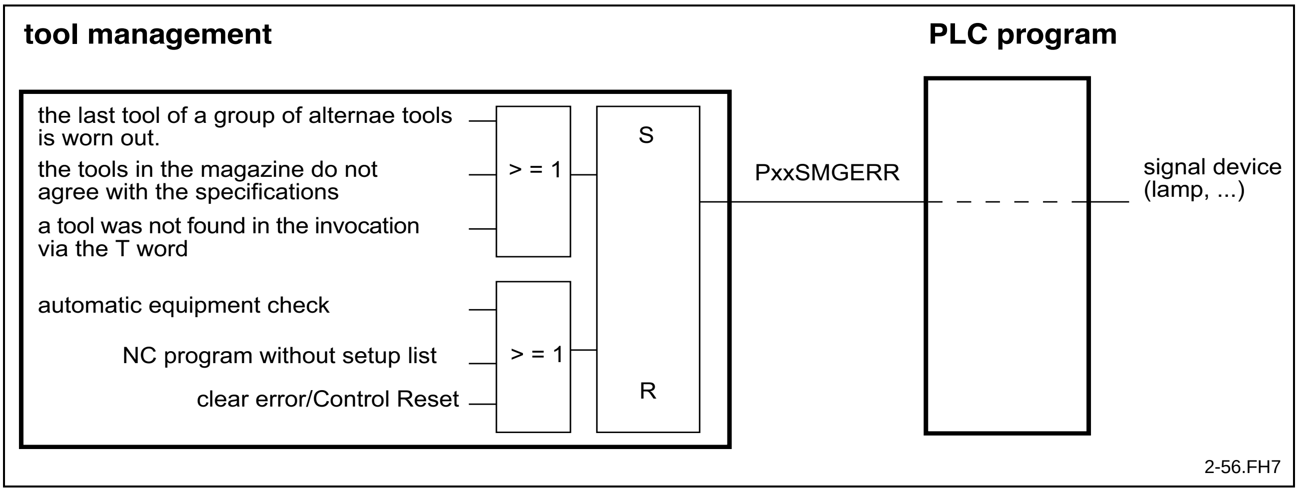


Fig. 1-86: Effect of the signal ”Tool error status”

The signal is only cleared after the next equipment check, when an NC program without a setup list is started, or when a "clear fault" or "control reset" is issued.

An equipment check is performed whenever a different or a modified setup or tool list is loaded into the Rexroth MTC 200 or storage changeover is performed.

Spindle stop Signals

Spindle stop upon program stop "PxxC.SPSTP"

Designation **PxxC.SPSTP** = **P**rocess **xx** **C***ommand* **Sp***indle* **St**o**p**

Control signal PLC →NC

PxxC.SPSTP = 0: Process stop without spindle stop

PxxC.SPSTP = 1: Process stop with spindle stop

Function A running spindle will be stopped upon a process stop if the "Spindle behavior upon process stop" signal is set.

The spindle will start first when the NC program is restarted.

The NC program is continued once the command speed has been reached. Any additional axes will also be moved then.

A spindle will not be affected upon a process stop if the "Spindle behavior upon process stop" signal is cleared.

If the "Spindle stop upon process stop" signal is set and there is no process stop, the spindle will not be stopped. If a process stop exists when the spindle stop signal is set, the spindle will not be stopped. This means that the spindle stop signal must be issued together with the process stop at the latest.

Note: From Version 04.14/xx onwards, this signal will no longer be interpreted. It is replaced by the axis-related "AxxC.SPSTP" signal.

Spindle stop upon stopped NC program "PxxC.SPHLT"

Designation

PxxC.SPHLT = **P**rocess **xx** **C***ommand* **Sp***indle* **Ha**l**t**

Control signal PLC →NC

PxxC.SPHLT = 0: Spindle does not stop with stopped NC program

PxxC.SPHLT = 1: Spindle does stop with stopped NC program

Function

The "spindle stop upon stopped NC program" signal retrospectively stops a running spindle after the NC program has already been interrupted. The signal must be set for this purpose.

The signal does not have an effect if the NC program has not yet been stopped.

The spindle will start first when the NC program is restarted. The NC program is continued once the command speed has been reached. Any additional axes will also be moved then.

Note: From Version 04.14/xx onwards, this signal will no longer be interpreted. It is replaced by the axis-related "AxxC.SPHLT" signal.

Command spindle speed reached "PxxC.N_CMD"

Designation

PxxC.N_CMD = **P***rocess* **xx** **C***ommand Spindle* **N=N$_{\mathbf{CMD}}$**

Control signal PLC →NC

PxxC.N_CMD = 0: Spindle speed reached (negated)

PxxC.N_CMD = 1: Spindle speed not reached yet (negated)

Function

The "spindle command speed reached" signal is required when external spindles are employed. This signal informs the NC that the spindle command speed has been reached.

The main drive amplifier supplies a signal when the command speed is reached.

The signal must be negated when it is transferred to the NC. This means that the signal is cleared when the command speed is reached.

When the NC program is restarted, the spindle starts up first.

Program execution is continued when the "spindle speed reached" signal is cleared.

Programming in the PLC is not required if the signal is not used (signal is not connected in the PLC program).

Note: From Version 04.14/xx onwards, this signal will no longer be interpreted. It is replaced by the axis-related "AxxC.N_CMD" signal.

Synchronous Axes

Activate synchronous axes compound <xx> "PxxC.SCONn"

Designation

PxxC.SCONn = **P**rocess **xx** **C***ommand* **S***ynchronous* **Con***trol* **n** *On*

Control signal PLC →NC (with n = 1 ... 4)

PxxC.SCONn = 0: Deactive synchronous axes compound
PxxC.SCONn = 1: Activate synchronous axes compound

Function

This interface signal can be used for synchronously utilizing up to four feed axes. follower axisThe functionality is required for "following of axes""or "gantry axes" functions.

The PLC sets the "PxxC.SCONn" control signal to activate the related synchronous axes compound.

The NC cancels synchronous operation as soon as the PLC resets the control signal.

A maximum of 4 synchronous axes compounds can be defined (PxxC.SCON1 ... PxxC.SCON4).

In the machine data, the axes are allocated to the synchronous axes compound.

Any axis type may be used for slave or gantry mode. Only axes of the same type may be used within a synchronous axes compound. This means that either only "linear axes", or only "endlessly rotating rotary axes", or only "not endlessly rotating rotary axes" can be combined as a synchronous axes compound.

With synchronous axes compounds for gantry applications, the synchronous axes compound should be activated before power to the drives is switched on. This should be done particularly at the beginning of the homing process in order to enable synchronous homing of all the mechanically coupled axes.

Note: If the PLC activates a synchronous axes compound asynchronously to NC program execution without auxiliary function output, the NC will not activate the synchronous axes compound before all the blocks of the preparation memory have been processed. If an error affecting the synchronous axes compound occurs, that error can only be cleared if the synchronous axes compound is de-activated before the "clear error" signal is issued.

Synchronous axes compound is activated "PxxS.SCONn"

Designation

PxxS.SCONn = **P**rocess **xx** **St**atus **S**ynchronous **Con**trol **n** is On

Control signal PLC →NC (with n = 1 ... 4)

PxxS.SCONn = 0: Synchronous axes compound is deactivated
PxxS.SCONn = 1: Synchronous axes compound is activated

Function

If one of the four possible synchronous axes compounds (PxxS.SCON1... PxxS.SCON4) has been activated, the NC sets the corresponding status signal to "1".

The NC resets the signal

- if the PLC removes the controller enabling signal from one axis of the synchronous compound during the movement;
- as soon as the NC loses control of the drives (if, for example, the residual energy of the DC bus proves insufficient for decelerating the drives); or
- if, in conjunction with axis transfer, the NC returns the transferred axes to the related process after a Control Reset and/or at the program end (BST, RET, M02, M30).

Synchronized Spindles

Minimize torsion moment "PxxC.SSnMT"

Designation **PxxC.SSnMT** = **P**rocess **xx** **C**ommand **S**ynchronous **S**pindle **n** **M**inimize **T**orsion

Control signal PLC →NC (with n = 1,2)

PxxC.SSnMT = 0: Torsion moment not minimized
PxxC.SSnMT = 1: Torsion moment minimized

Function Setting this control signal causes the NC to minimize the torsion moment that exists between the associated synchronous spindle and the leadscrew until the signal is reset.

Such a torsion moment may occur, for example, in shaft machining without tailstock. Despite angle synchronism and lag-free position control, mutual influence may cause an actual position value difference to exist between leadscrew and synchronous spindle when a shaft is gripped by a second spindle (spindles misaligned, for example). The gripping process quasi freezes that actual position value difference. Eventually, this leads to the two spindles working against each other at maximum torque during the synchronous operation. In such a case, minimizing the torsion moment is absolutely necessary.

Note: The signal is only interpreted if main spindle synchronization is active.

Activate spindle synchronization "PxxC.SSnON"

Designation **PxxC.SSnON** = **P**rocess **xx** **C**ommand **S**ynchronous **S**pindle **n** **O**n

Control signal PLC →NC (with n = 1,2)

PxxC.SSnON = 0: Deactivate synchronization
PxxC.SSnON = 1: Activate synchronization

Valid in all modes.

Function The PLC sets this control signal to activate main spindle synchronization. The NC cancels synchronous operation as soon as the PLC resets the control signal.

Specifically in the case of axis transfer, the spindles that participate in synchronization and belong to another master process must be transferred to the process concerned before synchronization is activated, and may not be returned before synchronization is de-activated.

Special spindle movements (such as hunting, drifting, or positioning) that are initiated via an AXD command are not permitted during main spindle synchronization.

When main spindle configuration is switched on, the NC checks the existing configuration. The NC generates an error message and interrupts machining if it detects a mismatch during this check.

Synchronous operation is ok "PxxS.SSnOK"

Designation **PxxS.SSnOK** = **P**rocess **xx** **S**t*atus* **S**y*nchronous* **S**p*indle* **n** *is* **Ok**

Control signal PLC →NC (with n = 1,2)

PxxS.SSnOK = 0: The amount of the actual position difference exceeds the synchronous operation window

PxxS.SSnOK = 1: The amount of the actual position difference is smaller than or equal to the synchronous operation window

Function The NC employs this signal to indicate whether or not the amount of the actual position difference exceeds the synchronous operation window.

The synchronous operation window is used for continually monitoring synchronous operation. The NC checks the difference between the actual position value of the leadscrew and the actual position value of the synchronous spindle, taking the corresponding transmission ratio into consideration.

The synchronous operation window may be modified during program mode and active main spindle synchronization. The modification will not become effective before main spindle synchronization is activated again.

Synchronous operation error limit "PxxS.SSnER"

Designation **PxxS.SSnER** = **P**rocess **xx** **S**t*atus* **S**y*nchronous* **S**p*indle* **n** **E**r*ror*

Control signal PLC →NC (with n = 1,2)

PxxS.SSnER = 0: The amount of the actual position difference is smaller than or equal to the synchronous operation error limit

PxxS.SSnER = 1: The amount of the actual position difference exceeds the synchronous operation error limit

Function The NC employs this signal to indicate whether or not the amount of the actual position difference exceeds the synchronous operation error limit.

The synchronous operation error limit is used as a limit value for continually monitoring the actual position value difference, taking the corresponding transmission ratio into account.

The synchronous operation error limit may be modified during program mode and active main spindle synchronization. The modification will not become effective before main spindle synchronization is activated again.

1.4 Mechanism Signals

External Mechanisms

"External mechanisms" are defined within the system parameters.

External mechanisms may be used in three different ways.

1. The mechanism is entered in the system parameters as an Rexroth MTC 200 process.

In this case, a process in a SLAVE NC can be controlled by a higher-order process in the MASTER NC.

The two processes are synchronized using "mechanism signals".

These signals can only be used for synchronization.

NC axes cannot be allocated to external mechanisms.

By analogy, reference is made to "internal process" synchronization. The messages that are internally produced for the NC process in the SLAVE controller can be displayed via the I/O level in the MASTER NC for the correspondingly associated mechanism.

2. The mechanism is entered as an PLC process in the system parameters and controlled by a higher-level NC process.

In this case, too, synchronization is performed via the mechanism signals. This means that the higher-level process issues a start signal that activates the PLC process (a step sequence, for example). Once the individual machining steps of the PLC process have been completed, the PLC process issues an acknowledgment to the higher-level process to signal the end of its execution.

3. The mechanism is entered as an PLC process in the system parameters. Even so, it is not affected by higher-level processes.

This utilization of external mechanisms enables independent PLC executions to be split according to the process structure of the machine.

Alternatively, the mechanism could be employed as an "additional diagnosis element".

"Additional diagnosis element" means that the mechanism is solely used for displaying messages in the diagnosis menu, thus enhancing the signaling range of the internal processes.

Note: With external mechanisms, all diagnoses (*...600) can be written from the PLC and be edited in the Message Integrator. External mechanisms cannot be allocated to axes.

Mechanism Control Signals

Part machined "MxxC.POK"

Designation **MxxC.POK** = **M**echanism **xx** **C***ommand* **P***art* **OK**

Control signal PLC →NC

MxxC.POK = 0: Part not completely machined

MxxC.POK = 1: Part completely machined

Function The signal must be set to logic "1" if a "POK" has been programmed within a MASTER NC program and the mechanism has been defined as a subordinate unit in that process.

Please refer to the explanation of the internal process signal "PxxC.POK" for details on the "part machined" signal.

If the external mechanism has been allocated to a process in a SLAVE NC, the "PxxS.POK" signal of the SLAVE process must be mapped via the I/O level onto the "MxxC.POK" signal in the MASTER controller.

Acknowledge machining of mechanism "MxxC.QP"

Designation **MxxC.QP** = **M**echanism **xx** **C***ommand* **Q***uit* **P***rocess*

Control signal PLC →NC

MxxC.QP = 0: No acknowledgment that mechanism is terminated

MxxC.QP = 1: Acknowledgment that mechanism is terminated

Function The signal must be set if, within a MASTER NC program, a "WP" has been programmed for the corresponding mechanism and the process that is allocated to the mechanism has completed its program.

If the external mechanism has been allocated a process in a SLAVE NC, the "PxxS.QP" signal must be mapped via the I/O level onto the "MxxC.QP" signal in the MASTER controller.

Mechanism Status Signals

Define mechanism "MxxS.DP"

Designation **MxxS.DP** = **M**echanism **xx** **S***tatus* **D***efine* **P***rocess*

Status signal NC →PLC

MxxS.DP = 0: Mechanism has not been defined

MxxS.DP = 1: Mechanism has been defined

Function The NC sets the signal if the "DPxx" command (<xx> stands for the mechanism number) has been programmed within an NC program.

Please refer to the explanation of the internal process signal "PxxS.DP" for further details of the "define process" signal.

Mechanism program number "MxxS.PRGNR"

Designation **MxxS.PRGNR** = **M**echanism **xx** **S***tatus* **Pro***gram* **N***umbe***r**

Status signal NC →PLC

MxxS.PRGNR (USINT value) : Program number

Function The NC employs this signal to tell the PLC the program number that has been selected for this mechanism.

The signal is used if the external mechanism is an NC process whose machining program is selected via the higher-level process.

Please refer to the explanation of the process-specific interface signals "PxxC.PRGNR" and "PxxS.PRGNR" for more information about NC program selection.

Reverse program start "MxxS.RP"

Designation **MxxS.RP** = **M**echanism **xx** **S***tatus* **R***everse* **P***rocess Start*

Status signal NC →PLC

MxxS.RP = 0: No reverse program start from the MASTER NC program

MxxS.RP = 1: Reverse program start from the MASTER NC program

Function The "reverse program enabling" signal is used for telling the PLC program that an "RP" reverse program start has been programmed for this mechanism in the higher-level process.

Please refer to the explanation of the process-related interface signal "PxxS.RP" for more information about this signal.

Mechanism number "MxxS.PROC"

Designation **MxxS.PROC** = **M**echanism **xx** **S***tatus* **Proc***ess Number*

Status signal NC →PLC

MxxS.PROC (INT value) : Mechanism number

Function This signal is used for telling the PLC the mechanism number.

Advance program start "MxxS.AP"

Designation **MxxS.AP** = **M**echanism **xx** **S***tatus* **A***dvance* **P***rocess Start*

Status signal NC →PLC

MxxS.AP = 0: No advance program start from the MASTER NC program

MxxS.AP = 1: Advance program start from the MASTER NC program

Function The "advance program enabling" signal is used for telling the PLC program that an "AP" advance program start has been programmed for this mechanism in the higher-level process.

Please refer to the explanation of the process-related interface signal "PxxS.AP" for more information about this signal.

Lock mechanism "MxxS.LP"

Designation **MxxS.LP** = **M**echanism **xx** **S***tatus* **L***ock* **P***rocess*

Status signal NC →PLC

MxxS.LP = 0: No lock of mechanism

MxxS.LP = 1: Lock mechanism

Function

The "disable mechanism" status signal is used for telling the PLC program that an "LP" has been programmed for this mechanism in the higher-level process.

Please refer to the explanation of the process-related interface signal "PxxS.LP" for more information about this signal.

1.5 Axis Signals

There is an axis signal interface between the NC and the PLC for each of the up to 32 axes of the Rexroth MTC 200. An NC controlled main spindle or an NC controlled tool magazine is also considered as an axis.

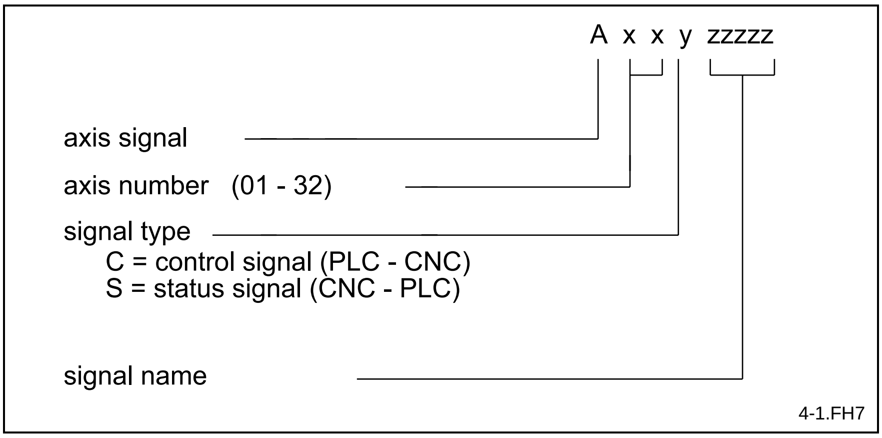


Fig. 1-87: Explanation of symbolic operands

Axis signals are processed in the NC only for the axes that have been entered in the system parameters. Axis signals of axes that are marked as "*non-existing*" in the system parameters will not be processed (even if they have been programmed in the PLC).

The axis signals are divided into the following four groups:

Axis monitoring	**Axis control signals**	**Axis status signals**	**Waypoint signals**
Because of this signals, every axis is monitored and specific diagnosis messages are created in the NCD	These signals are used for controlling the function of each individual axis.	PLC information about states will be told to every axis via this signals	These signals can be used like cams for monitoring the areas in the PLC.

Fig. 1-88: Grouping of axis signals

Axis Monitoring Signals

The monitoring of safety limit switches and thermal protection switches is performed in the NC in order to be able to guarantee a differentiated and always correct diagnosis in conjunction with process control signals. Thus, it is necessary that the alarm lines of the individual axes are directly transferred to the NC, without any further interconnections inside the PLC.

Safety limit switch "AxxC.OTRVL"

Designation **AxxC.OTRVL** = **A**xis **xx** **C**ommand **O**ver**tr**a**v**el **L**imit Switch

Control signal PLC →NC

AxxC.OTRVL = 0: Safety limit switch actuated; EMERGENCY STOP chain open

AxxC.OTRVL = 1: Safety limit switch not actuated

Function This signal is used for telling the NC that a safety limit switch has been actuated and, consequently, the EMERGENCY STOP chain of that station has been opened.

The safety limit switches of all axes in a station are connected in series in the EMERGENCY STOP chain; opening the switch of the first axis therefore causes an error message to be issued for all subsequent axes. The safety limit switches of the individual stations must therefore be connected in series such that the axis with the smallest axis number lies at the beginning of the EMERGENCY STOP chain, while the axis with the highest axis number lies at the end.

If several axes in a station have activated their safety limit switches at the same time, the Rexroth MTC 200 only reports the axis with the smallest axis number in the station diagnosis.

Note: After a safety limit switch has been activated, a hardware jumper (existing service switches) must be connected across the corresponding limit switch. Power can then be switched back on. Next, the axis must be jogged back into the travel range in "Setup" mode, and the jumper must be removed. Subsequent activation of the reverse program permits the unit to be taken back to the home position.

.If the station is used without a safety limit switch, the corresponding signal in the PLC program must statically be set to "1" (rotary axis, main spindle). If this is not done, the NC will not enable the power (PxxS.POWEN).

Note: For spindles and rotary axes, this signal must statically be set to "1".

Thermal protection switch "AxxC.MTAS"

Designation

AxxC.MTAS = **A**xis **xx** **C**ommand **M**otor **T**emper**a**ture **S**witch

Control signal PLC →NC

AxxC.MTAS	= 0:	Motor overtemperature
AxxC.MTAS	= 1:	Motor temperature is OK

Function

This signal is used for telling the NC that the thermal protection switch of the associated motor has responded. Depending on the station's mode, this signal has the following effect:

- Power is switched off. Power can not be switched on as long as one of the axes of a station reports "temperature is too high".
- Power is on, no program is active. A "temperature is too high" message from one of the station axes immediately shuts down the power in that station.
- Program cycle is active. First, the "temperature is too high" message from one of the station axes is merely registered. If the program cycle is terminated within 60 seconds, power will subsequently be shut off. If the program cycle takes more than 60 seconds to complete, the station will immediately be stopped after 60 seconds, and power is switched off afterwards. If the "temperature is too high" message disappears within the 60 second interval or before the end of the cycle, neither an error message is issued nor is the power shut down.

Note: Determine the cause of the temperature rise after the monitoring device has responded (insufficient motor dimensioning, tight mechanical system, etc.).

If the thermal motor protection switches are not connected to inputs at I/O level, this signal must statically be connected to "1" in the PLC program. This is particularly important for digital axes.

Axis Control Signals

The term "Axis control signals" covers all signals that are generated in the PLC and transferred to the NC. Some control signals are only active in a specific mode, others are mode-independent.

Axis operative "AxxC.READY"

Designation AxxC.READY = A*xis* xx C*ommand* READY

Control signal PLC →NC

Valid in all modes.

PxxC.READY = 0: Axis is not ready

PxxC.READY = 1: Axis is ready

Function This signal is used for reporting the operational state of an axis to the NC.

If the process the axis is related to fulfills all requirements for power enabling and if the "power available" signal exists for the process, the NC issues the "controller enabling" (AxxS.RF) signal for the axis concerned when the "ready for operation" (AxxC.READY) signal is applied.

If an axis is moved that is not ready for operation, the NC corrects the command value to the actual value. The controller enabling signal and velocity command values are not output for axes that are not ready for operation. This permits an axis to be clamped.

If the axis receives a motion command from the NC program while the READY signal is inactive, program execution is stopped and the "inactive axis programmed" process error is generated.

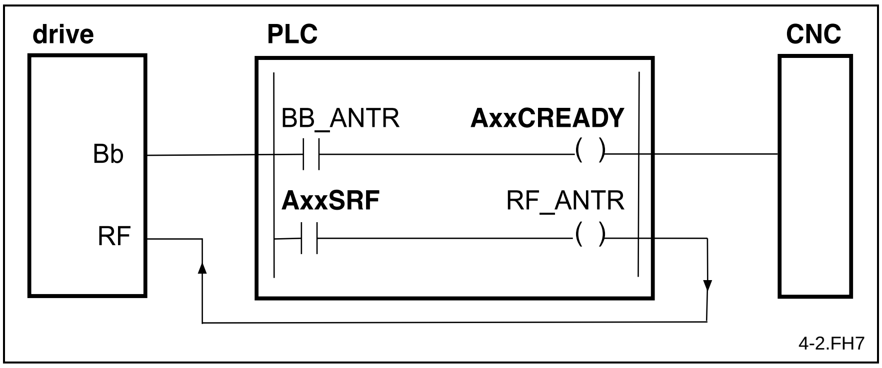


Fig. 1-89: Signal “Axis ready” hard-wired to drive amplifier

If the "BB" signals of the drive amplifiers are not connected with the PLC's I/O system, the "axis operational" signal must be generated as follows.

Rexroth Bosch Group

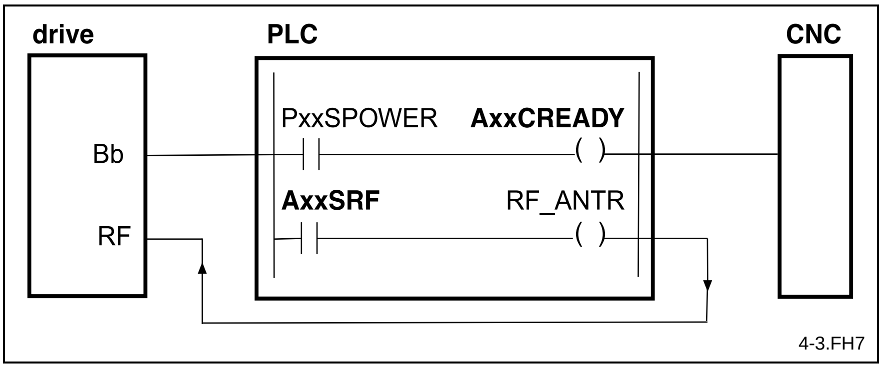


Fig. 1-90: Signal “Axis ready” not hard-wired to drive amplifier

Note: Specific de-activation of the controller enabling signal requires the "Axis ready for operation" signal (AxxC.READY) to be interconnected with specific shutdown conditions (refer to "Controller enabling" interface description).

When digital axes are employed, this signal may be directly connected with the "digital drive ready for operation" status signal (AxxS.BBDIG).

Axis enable "AxxC.ENABL"

Designation

AxxC.ENABL = A*xis* xx C*ommand* Enable

Control signal PLC →NC

AxxC.ENABL = 0: No axis enable

AxxC.ENABL = 1: Axis enable is present

Only valid in "Setup" mode for jogging an axis.

Function

The "Axis enabled" signal has an effect in "Setup" mode only when an axis is jogged.

The axis cannot be moved by a jog signal if the "Axis enabled" signal is missing. Removing the "axis enabled" signal from a jogged axis stops the axis immediately.

The "Axis enabled" signal should be used if jogging an axis is only permitted in certain travel ranges.

Axis enabling is cleared in the PLC program when those ranges are left (e.g. delimiting the travel range by waypoints after the reference dimension has been established, specific cams, or proximity switches that report mechanical end positions).

Subsequently, adequate interlocking in the PLC program will permit jogging to be performed only in the opposite direction.

Programming in LAD:

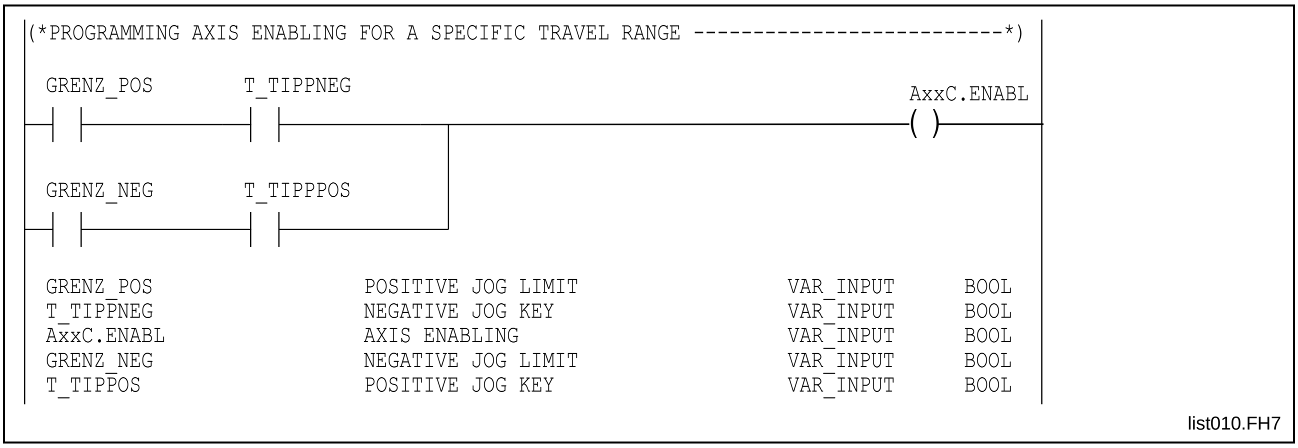


Fig. 1-91: Programming of the axis enable signal of a certain travel range

Note: The "axis enabled" signal must statically be set to "1" if is not used as a safety function in the PLC program.

Homing switch "AxxC.HOMLS"

Designation

AxxC.HOMLS = **A***xis* **xx** **C***ommand* **Hom***e* **L***imit* **S***witch*

Control signal PLC →NC

AxxC.HOMLS = 0: Homing switch not actuated

AxxC.HOMLS = 1: Homing switch actuated

The signal is valid in all modes.

Function

When an incremental measuring system is employed, the "homing switch" axis signal is used for locating the reference position of an axis. Together with the reference impulse of the incremental measuring system, the homing switch determines the reference position of an axis.

With digital drives, the signal is directly interpreted by the drive, and need not be programmed in the PLC.

Programming the signal is not necessary either when absolute measuring systems are employed.

Position strobe "AxxC.STRBP"

Designation

AxxC.STRBP = **A***xis* **xx** **C***ommand* **Str***obe* **P***osition*

Control signal PLC →NC

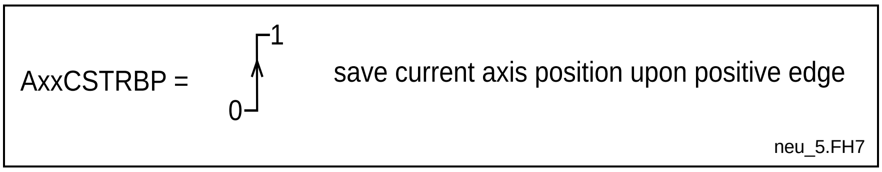


Fig. 1-92: Save current axis position at the positive edge

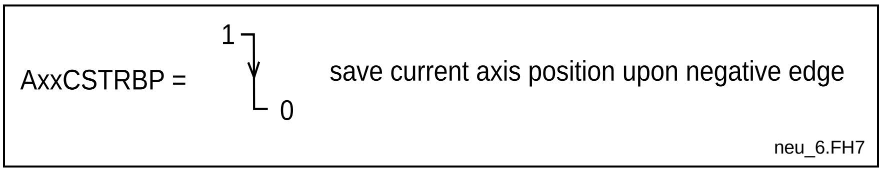


Fig. 1-93: Save current axis position at the negative edge

Valid in all modes.

Function

Upon the positive or negative edge of the "position strobe" signal, the NC accepts the current position of the related axis. For each edge, there is one register available in the NC. Two commands in the NC program are used for reading the axis position. The "PMP" and "NMP" NC commands are used for interrogating the axis position upon the positive or negative edge, respectively.

Example

Determining the center of a workpiece.

An initiator shall be used for determining the center of a workpiece. The axis is moved slowly to move the initiator across the workpiece. The initiator responds when it is above the workpiece. The positions at which the workpiece begins and ends are stored. The workpiece center can be determined from those positions.

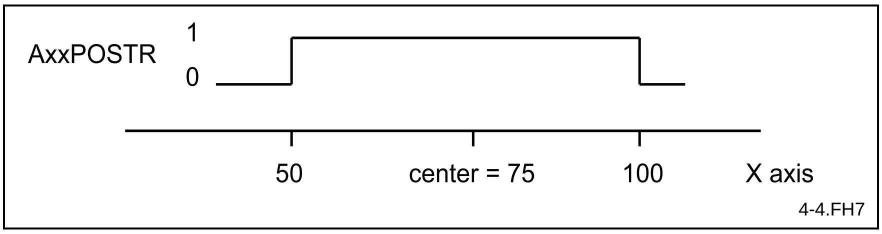


Fig. 1-94: Response behavior of the initiator

The variables (@1=50, @2=100) contain the position values for further processing.

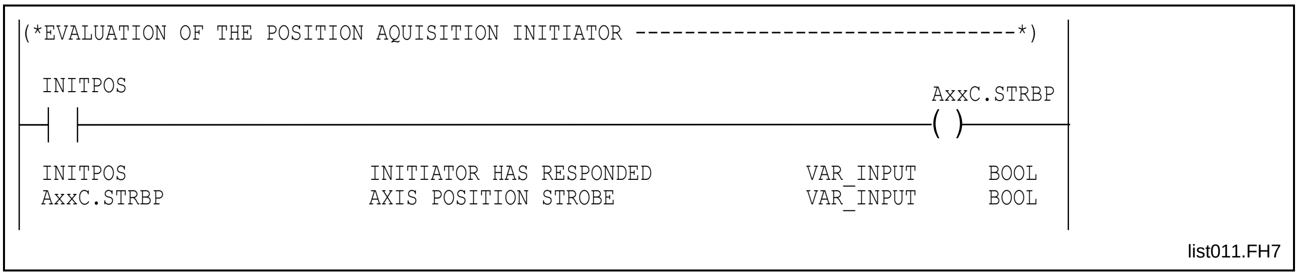


Fig. 1-95: Evaluation of the initiator on position detection

Command sequence in the NC program:

```
G1 X200 F100                ;slow motion over workpiece
@1=PMP (X) @2=NMP (X)       ;transfer of positions in variables
@3=@1+(@2-@1)/2             ;position of the workpiece center
```

Dead times (switching hysteresis of the initiator, PLC cycle time) must be taken into account when the scan time is selected.

Homing of axis "AxxC.HOME"

Designation

AxxC.HOME = **A***xis* **xx** **C***ommand* **Hom***ing*

Control signal PLC →NC

AxxC.HOME	= 0:	No homing
AxxC.HOME	= 1:	Homing command

Only valid in "Setup" mode.

Function

Single-axis homing requires the following requirements to be fulfilled:

- "Setup" mode has been selected;
- the axis enabled signal has been applied;
- the process enabling signal is issued;
- power is available;
- no error condition exists, and
- No NC block is active or the NC program has been stopped.

The "Single axis homing" signal is activated with the positive flank. The "axis homed" signal (AxxS.HOMED) is set after homing has been successfully completed.

See "*Parameter Description*" for a description of the homing procedure.

The following conditions interrupt single-axis homing:

- jog command "positive jogging" or "negative jogging" at the same time;
- axis enabled signal removed;
- removing the process enabling signal,
- jog mode changeover;
- mode changeover;
- process error exists, or
- homing signal removed.

When an axis is jogged, the readiness for starting an advance program is lost. However, starting a reverse program is possible. The effect on the ready-to-start capability is insignificant if the "Reverse program execution required" process parameter has been set to "No".

Note: The signal is not required for absolute measuring systems.

Positive jogging "AxxC.JGPOS"

Designation

AxxC.JGPOS = **A**xis **xx** **C***ommand* **J**o**g**g*ing* **Pos***itive*

Control signal PLC →NC

AxxC.JGPOS	= 0:	No positive jogging command
AxxC.JGPOS	= 1:	Positive jogging command

Only effective in "Setup" mode.

Function

Applying the "AxxC.JGPOS" signal in "Setup" mode moves the axis in the positive direction. The type of movement (continuous or incremental axis movement) is defined by the jog mode, and the velocity is defined by the "Rapid traverse rate" or "Jog rate" signal. Rapid traverse rate is only interpreted after reference has been established.

The following are the prerequisites of an axis movement:

- the axis is ready for operation (AxxC.READY = 1) and the controller enabling signal has been activated;
- the controller enabling signal has been applied to the drive (AxxS.RF=1);
- the axis enabling signal has been applied (AxxC.ENABL=1);
- "Setup" mode has been preselected for this station (PxxC.MODE0=0, PxxC.MODE1=1);
- there is no homing command pending for the axis (AxxC.HOME=0);
- there is no jog command in the negative direction for the axis (AxxC.JGNEG=0) and
- No NC block is active or the NC program has been stopped.

The axis can only be moved in the positive direction when all conditions are fulfilled and as long as the "AxC.JGPOS" signal is applied.

If "continuous jogging" has been selected, the axis moves as long as the "AxxC.JGPOS" signal is applied or until the positive travel range limit has been reached.

In any other jog mode (incremental jogging), the positive flank of the signal initiates a movement that is not stopped before the selected distance has been covered.

The following conditions interrupt the jog movement:

- jog command "negative jogging" or single-axis homing at the same time;
- axis enabled signal removed;
- removing the process enabling signal,
- jog mode changeover;
- mode changeover or
- a process error condition exists.

When an axis is jogged, the readiness for starting an advance program is lost. However, starting a reverse program is possible. This response is relevant if the "Reverse program execution required" process parameter has been set to "Yes".

Negative jogging "AxxC.JGNEG"

Designation

AxxC.JGNEG = **A**xis **xx** **C***ommand* **Jo***g*g*ing* **Neg***ative*

Control signal PLC →NC

AxxC.JGNEG = 0: No negative jogging command

AxxC.JGNEG = 1: Negative jogging command

Only effective in "Setup" mode.

Function

Applying the "AxxC.JGNEG" signal in "Setup" mode moves the axis in the negative direction. The type of movement (continuous or incremental axis movement) is defined by the jog mode, and the velocity is defined by the "Rapid traverse rate" or "Jog rate" signal.

The following are prerequisites for an axis movement:

- the axis is ready for operation (AxxC.READY = 1) and the controller enabling signal has been activated;
- the controller enabling signal has been applied to the drive (AxxS.RF=1);
- the axis enabling signal has been applied (AxxC.ENABL=1);
- "Setup" mode has been preselected for this station (PxxC.MODE0=0, PxxC.MODE1=1);
- there is no homing command pending for the axis (AxxC.HOME=0);
- there is no jog command in the positive direction for the axis (AxxC.JGPOS=0), and
- No NC block is active or the NC program has been stopped.

The axis can only be moved in the negative direction when all conditions are fulfilled and as long as the "AxCJGNEG" signal is applied.

If "continuous jogging" has been selected, the axis moves as long as the "AxxC.JGNEG" signal is applied or until the negative travel range limit has been reached.

In any other jog mode (incremental jogging), the positive flank of the signal initiates a movement that is not stopped before the selected distance has been covered.

The following conditions interrupt the jog movement:

- jog command "positive jogging" or single-axis homing at the same time;
- axis enabled signal removed;
- removing the process enabling signal,
- jog mode changeover;
- mode changeover or
- a process error condition exists.

When an axis is jogged, the readiness for starting an advance program is lost.

Starting a reverse program is possible.

This response is relevant if the "Ready-for-start required" process parameter has been set to "Yes".

Writing upon a SERCOS real time control bit "AxxC.QDDS"

Designation

AxxC.QDDS = **A***xis* **xx C***ommand* **Q***utput of* **DDS**

Control signal PLC →NC

AxxC.QDDS	= 0:	Real time control bit is 0
AxxC.QDDS	= 1:	Real time control bit is 1

Note: The signal can only be used for digital drives. Only real-time bit No. 2 can be manipulated.

Function

This signal enables the PLC to write via the NC to a real-time control bit in the digital drive.

The allocation of an ident number to the real-time control bit in the SERCOS parameters determines the function that is triggered when the bit is written to.

Please refer to the "SERCOS Interface" description for a list of the parameters that can be manipulated via a real-time control bit and a description of the procedure for allocating a parameter to the real-time control bit.

With analog drives, this signal is insignificant.

Transition of sensor enabling to the drive.

Example

```
|(*TRANSFERRING SENSOR INPUT ENABLING SIGNAL TO DRIVE -----------------------------*)|
|                                                                                    |
| MESS_ENAB                                                            AxxC.CQDDS    |
|--| |------------------------------------------------------------------( )----------|
|                                                                                    |
| MESS_ENAB            SENSOR ENABLING                     VAR_INPUT      BOOL       |
| AxxC.CQDDS           REAL-TIME CONTROL BIT               VAR_INPUT      BOOL       |
```

list012.FH7

Fig. 1-96: Transition of sensor enabling to the drive.

Allocation of the associated SERCOS parameter:

Ident number for "allocation of real time control bit": S-0-0303

Ident number of "sensor enabling": S-0-0405

The ident number of "sensor enabling" is written to the SERCOS parameter for "real-time control bit allocation".

SERCOS parameter list:

.

.

ID No. S-0-0303: 00405

.

.

.

The drive employs this parameter allocation to interpret the sensor enabling signal.

Motion hold "AxxC.MHOLD"

Designation

AxxC.MHOLD = **A**_xis_ **xx C**_ommand_ **M**_otion_ **Hold**

Control signal PLC →NC

AxxC.MHOLD = 0: Motion hold is enabled
AxxC.MHOLD = 1: Axis motion is blocked

The signal is valid in all modes. It is also interpreted for spindle axes.

Function

When this signal is manipulated, it must be distinguished whether motion hold is set for an interpolating and/or coupled axis or for a non-interpolating axis.

The following applies to non-interpolating axes:

If motion hold for a non-interpolating axis is applied before a movement is announced (AxxC.MHOLD = 1), the axis will not be moved as long as motion hold is active.

If the motion hold has not been applied, the axis will immediately be moved independently of the other axes if a movement announcement is still applied.

The actual movement of the axis is interrupted if the PLC sets the motion hold while an axis is moving. Following a ramp function, the axis is braked at maximum deceleration without affecting the other axes. The movement announcement signal remains applied during the interruption of the movement.

Note: If the PLC sets motion hold when a movement announcement is reported to the PLC, the different propagation times of the individual signals will cause the axis concerned to be moved before the motion hold becomes effective. The axis will then be stopped immediately.

The following applies to interpolating and/or coupled axes:

In this case, interpolating axes / axes that participate in the coupling are also stopped, keeping to the path contour.

The movement is continued automatically once the PLC removes the motion hold signal.

General rule:

If the NC waits for the movement to be enabled, the NC generates a status message when the "Motion hold" signal is set.

In addition, the "Inactive axis programmed" message is not output and an error is not generated by switching off the controller enabling signal while motion hold is active.

Note: A disabled axis may <u>not</u> be programmed with a velocity-optimized block transition (G08).

Gear selection "AxxC.GEARn"

Designation

AxxC.GEARn = **A**xis **xx** **C**ommand **GEAR** Select Bit

3 control signals PLC →NC

AxxC.GEAR1
AxxC.GEAR2
AxxC.GEAR3

These signals have an effect in all modes.

Function

The NC only interprets the gear step selection control signals if a gearbox with at least two gear steps has been entered in the axis parameters.

Distinction is made between

- automatic gear selection (activated by the M function "M40") and
- direct gear step selection.

In *direct gear selection*, the required gear change is requested within the NC program via the M functions "M41" ... "M44".

The PLC interprets these M function, changes gear steps via the corresponding outputs, encodes the control signals for the actual gear step via the feedback messages from the gearbox, and acknowledges the corresponding M function.

The NC considers gear step selection as completed when it receives the M function acknowledgment via the PLC, and interprets the actual gear step bits .

Note: The message with the actual gear step bits must be updated whenever the gear step is changed and the controller is switched on (this means that it must be ensured that the actual gear step is reported in the very first cycle).

Allocation of the M functions to the actual gear step bits and the gear step:

M function *)	**PLC-NC Actual gear Stepp bit 3 AxxC.GEAR3**	**PLC-NC Actual gear Stepp bit 2 AxxC.GEAR2**	**PLC-NC actual gear step bit 1 AxxCGEAR1**	**Gear step**
x41	0	0	0	1
x42	0	0	1	2
x43	0	1	0	3
x44	0	1	1	4
-	1	1	1	n. d.

*) <x> stands for the corresponding spindle index

Fig. 1-97: Allocation of M commands to gear step bits

Note: Gear step selection is only available with digital main spindle drives.

Example Gear range selection

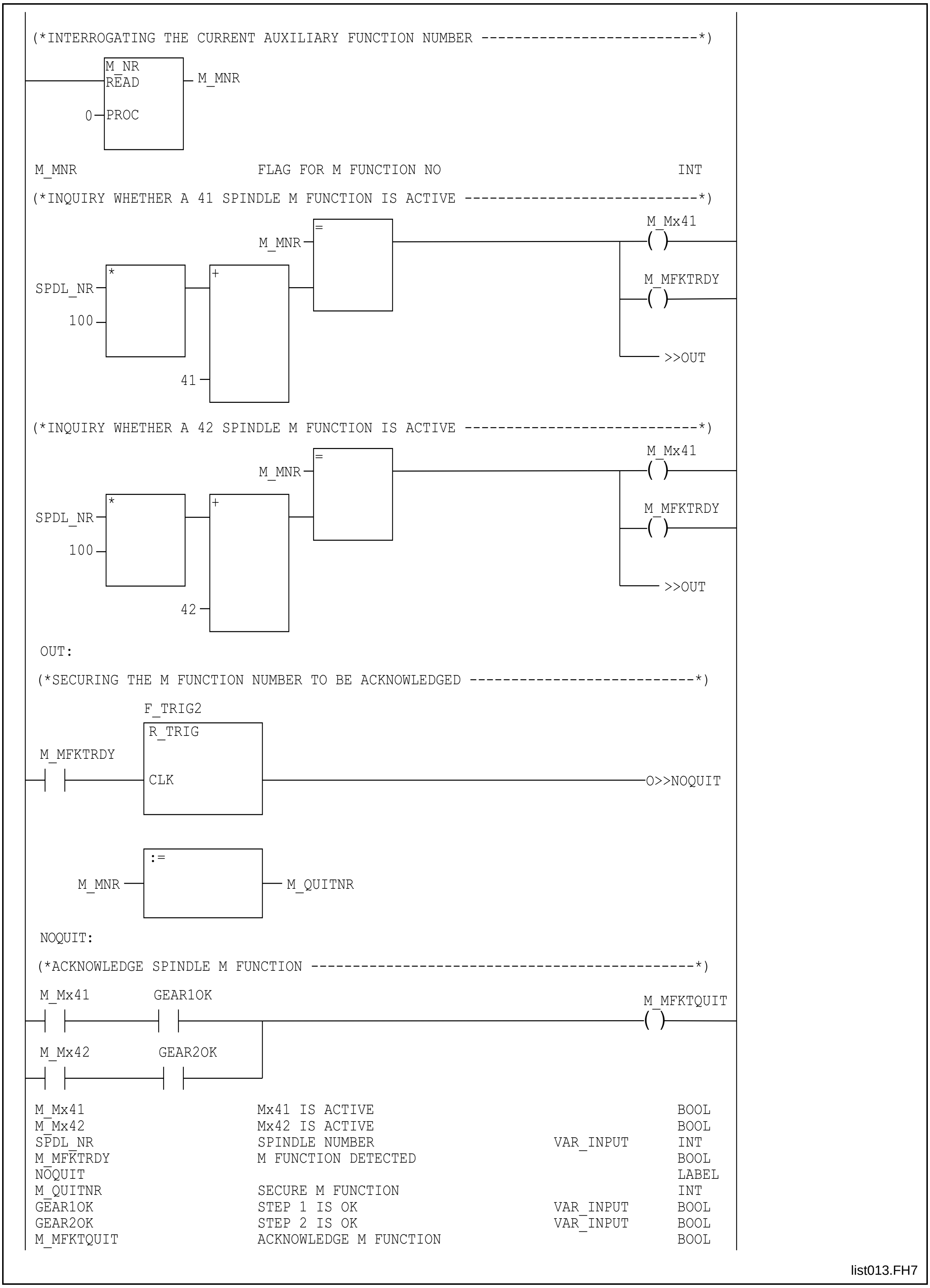
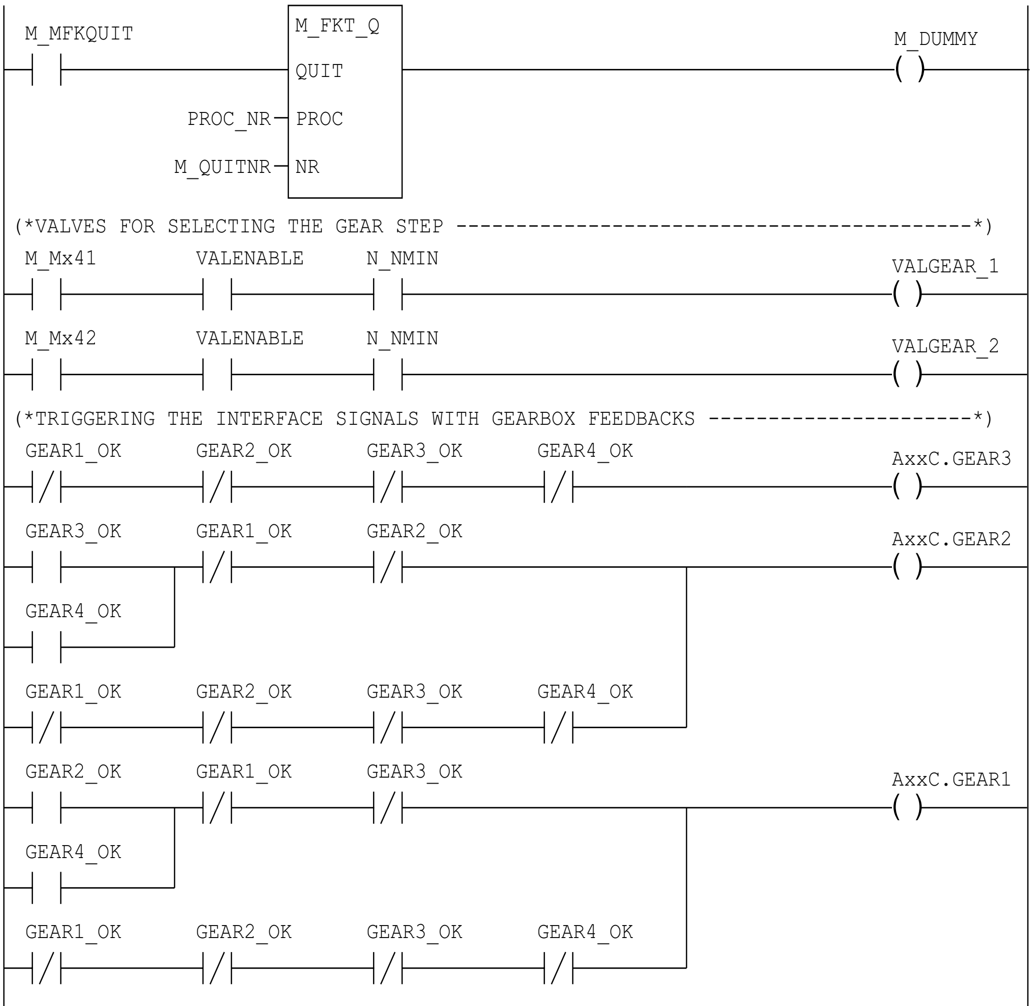


Fig. 1-98: Gear change



M_MFKQUIT	ACKNOWLEDGE M FUNCTION		BOOL
PROC_NR	PROCESS NUMBER	VAR_INPUT	INT
M_QUITNR	SECURE M FUNCTION NUMBER		INT
VALENABLE	ENABLING THE VALVES	VAR_INPUT	BOOL
N_NMIN	N<Nmin MESSAGE	VAR_INPUT	BOOL
AxxC.GEAR3	BIT 3 GEAR STEP SELECTION	VAR_OUTPUT	BOOL
AxxC.GEAR2	BIT 2 GEAR STEP SELECTION	VAR_OUTPUT	BOOL
AxxC.GEAR1	BIT 1 GEAR STEP SELECTION	VAR_OUTPUT	BOOL

list014.FH7

Fig. 1-99: Gear change (continued)

This example shows the gear step selection principle. A specific application usually contains an additional sequence for drifting or hunting the spindle during the changeover process.

Drifting or hunting is activated/de-activated via the PLC by setting the corresponding SERCOS axis parameters. The hunting frequency is entered in the corresponding SERCOS axis parameter via the user interface. The spindle speed specification for drifting is written in a SERCOS axis parameter via the PLC.

Table with the SERCOS parameters that can be used for changing gear:

Parameter	Function	Ident Number
SERCOS parameters that are entered via the user interface		
Hunting speed	When the gear step is changed, the drive hunts at the programmed hunting speed and the hunting cycle time (provided that hunting has been activated).	00213
Hunting cycle time	When the gear step is changed, the drive hunts at the programmed hunting speed and the hunting cycle time (provided that hunting has been activated).	00214
SERCOS parameters that are written via the PLC		
Spindle hunting specification	The drive hunts if bit 0 of this parameter is set to "1". By resetting the bit 0, hunting is again switched off.	65012 - bit 0
Spindle Positioning	The spindle is positioned if bit 1 of this parameter is set to "1".	65012 - bit 1
Command spindle speed specification	The drive drifts if bit 2 of this parameter is set to "1". By resetting the bit 0, drifting is again switched off.	65012 - bit 2
Spindle speed specification	This parameter specifies the speed at which the spindle shall drift. Weighting is 0.0001 rpm. The maximum speed that may be specified is 1000 rpm.	65013
Spindle acceleration specification	The acceleration value for the "spindle speed specification" spindle function is written to this parameter. Weighting is in rad/sec^2.	65014
Spindle angle specification	The spindle angle (in 0.0001°) for the "spindle positioning" spindle function is specified in this parameter.	65015
Parameters that may be interrogated in the PLC		
Spindle hunting acknowledgement	Spindle hunting is activated if bit 8 of this parameter is set.	65012 - bit 8
Spindle positioning acknowledgement	Spindle positioning is activated if bit 9 of this parameter is set.	65012 - bit 9
Spindle speed specification acknowledgement	Spindle drifting is activated if bit 10 of this parameter is set.	65012 - bit 10

Fig. 1-100: SERCOS parameter list for gear change

Spindle stop upon process stop "AxxC.SPSTP"

Designation

AxxC.SPSTP = A*xis* xx C*ommand* S*pindle* Stop

Control signal PLC →NC

AxxC.SPSTP = 0: Process stop without spindle stop

AxxC.SPSTP = 1: Spindle stop with process stop

Function

A running spindle will be stopped upon a process stop if the "Spindle stop upon process stop" signal is set. The spindle will start first when the NC program is restarted. The NC program is continued once the command speed has been reached. Any additional axes will also be moved then.

A spindle will not be affected upon a process stop if the "Spindle behavior upon process stop" signal is cleared.

If the "Spindle stop upon process stop" signal is set and there is no process stop, the spindle will not be stopped. If a process stop exists when the spindle stop signal is set, the spindle will not be stopped. This means that the spindle stop signal must be issued together with the process stop at the latest.

Note: The signal is only interpreted for spindles.

Spindle stop upon stopped NC program "AxxC.S.PHLT"

Designation

AxxC.SPHLT = A*xis* xx C*ommand* Sp*indle* Halt

Control signal PLC →NC

AxxC.SPHLT = 0: Spindle does not stop with stopped NC program

AxxC.SPHLT = 1: Spindle stops with stopped NC program

Function

The "spindle stop upon stopped NC program" signal retrospectively stops a running spindle after the NC program has already been interrupted. The signal must be set for this purpose.

The signal does not have an effect if the NC program has not yet been stopped.

The spindle will start first when the NC program is restarted. The NC program is continued once the command speed has been reached. Any additional axes will also be moved then.

Note: This signal is only interpreted for a spindle.

Command spindle speed reached "AxxC.N_CMD"

Designation

AxxC.*N_CMD = A*xis* xx C*ommand* N = N_{CMD}

Control signal PLC →NC

AxxC.N_CMD = 0: Spindle speed reached (negated)

PxxC.N_CMD = 1: Spindle speed not reached yet (negated)

Function

The "spindle command speed reached" signal is required when external spindles are employed. This signal informs the NC that the spindle command speed has been reached.

The main drive amplifier supplies a signal when the command speed is reached. (KDA: $N=N_{CMD}$ contact). The signal must be negated when it is transferred to the NC. This means that the signal is cleared when the command speed is reached.

When the NC program is restarted, the spindle starts up first.

Program execution is continued when the "spindle speed reached" signal is cleared.

Programming in the PLC is not required if the signal is not used (signal is not connected in the PLC program).

Note: The signal is only valid for spindle axes.
The signal is not required for NC controlled spindles.

Spindle stop upon program end "AxxC.SPSTE"

Designation

AxxC.SPSTE = **A**xis **xx C**ommand **Sp**indle **St**op with Program **E**nd

Control signal PLC →NC

AxxC.SPSTE = 0: No spindle stop upon program end

AxxC.SPSTE = 1: Spindle stop upon program end

Function

If the "spindle stop at program end" signal is not set, a "BST", "RET", "M30" or "M02" in the NC program will not stop a running spindle.

The spindle concerned will be stopped only if the signal is set to logic "1" before the program end is reached.

The stopped spindle will not be started when the NC program is started again. If this response is required, the corresponding auxiliary function must be programmed in the first block of the NC program.

Note: The signal is only valid for spindle axes.

Spindle stop upon control reset "AxxC.SPRST"

Designation

AxxC.SPRST = **A**xis **xx** **C**ommand **Sp**indle stop upon control reset

Control signal PLC →NC

AxxC.SPRST = 0: Control Reset stops the spindle

AxxC.SPRST = 1: Control Reset does not influence the spindle

Function

This control signal can influence the response of the spindle to a Control reset.

If the control signal is set, Control Reset will not stop a running spindle.

If the control signal is not set, Control Reset will stop a running spindle.

Feed override "AxxC.OVRD"

Designation

AxxC.OVRD = **A**xis **xx** **C**ommand **Ov**er**r**ide Value

Control signal PLC →NC

AxxC.OVRD = 0 ... 255

Function

"Feed override" is valid for all non-interpolating axes (i.e. for *spindle axes* and *magazine axes)*. Override is interpreted in the NC, irrespective of the mode; it has an effect on any axis movement (except on homing digital axes).

The valid range of override weighting by the PLC program is between 0 and 255%. The NC limits the axis and/or processor speed to the maximum values set in the parameters if an override value is set that is too large.

Note: Override weighting must always be generated in the PLC program. Axis movement is not possible without override weighting. An NC diagnosis is output if a motion command is output to an axis and override weighting 0% has been selected.

Activate spindle CW rotation (Mx3) "AxxC.M3"

Designation

AxxC.M3 = **A***xis* **xx** **C***ommand* **M3**

Control signal PLC →NC

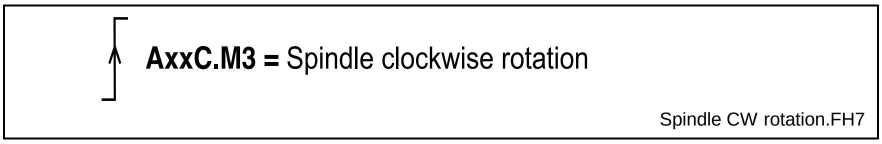


Fig. 1-101: Spindle clockwise direction

The signal is interpreted only in "Setup" mode.

Function

A positive edge of the "AxxC.M3" signal in "setup" mode activates "CW rotation" of the related axis. The NC takes the value stored for the main spindle-related axis parameter ("Jog speed") as the command speed.

Prerequisites for axis movement are:

- the axis is ready for operation (AxxC.READY = 1), and the controller enabling signal has been activated (AxxS.RF=1);
- the axis enabling signal has been applied (AxxC.ENABL=1);
- motion hold is inactive (AxxC.MHOLD=0);
- setup mode is preselected; and
- Spindle CCW rotation, spindle stop, or spindle positioning command have not been applied at the same time.

Once spindle CW rotation has been started, it may be aborted by:

- a Control Reset via the PLC; or
- the spindle stop interface signal (AxxC.M5=1).

Note: The signal is only available for digital main spindles.

Activate spindle CCW rotation (Mx4) "AxxC.M4"

Designation

AxxC.M4 = **A***xis* **xx** **C***ommand* **M4**

Control signal PLC →NC

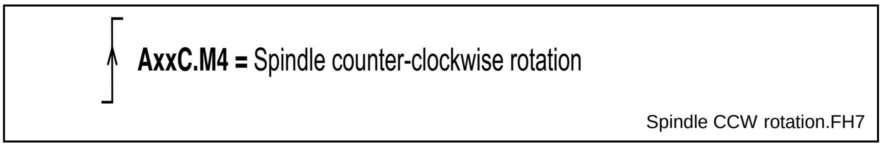


Fig. 1-102: Spindle counter-clockwise direction

The signal is interpreted only in "Setup" mode.

Function

A positive edge of the "AxxC.M4" signal in "setup" mode activates "anti-clockwise rotation" of the related axis. The NC takes the value stored for the main spindle-related "Jog speed" axis parameter as the command speed.

Prerequisites for axis movement are:

- the axis is ready for operation (AxxC.READY = 1), and the controller enabling signal has been activated (AxxS.RF=1);
- the axis enabling signal has been applied (AxxC.ENABL=1);
- motion hold is inactive (AxxC.MHOLD=0);
- "Setup" mode is preselected; and
- spindle CW rotation, spindle stop, or spindle positioning command have not been applied at the same time.

Once spindle anti-clockwise rotation has been started, it may be aborted by:

- a Control Reset via the PLC; or
- the spindle stop interface signal (AxxC.M5=1).

Note: The signal is only available for digital main spindles.

Spindle stop (Mx5) "AxxC.M5"

Designation **AxxC.M5** = **A***xis* **xx C***ommand* **M5**

Control signal PLC →NC

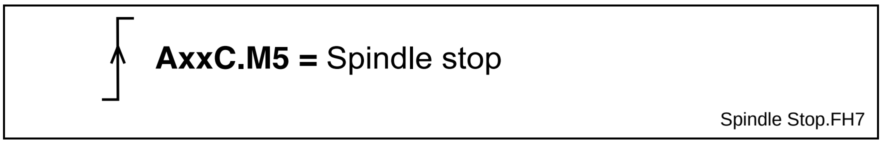


Fig. 1-103: Spindle stop

The signal is interpreted only in "Setup" mode.

Function A positive edge of the "AxxC.M5" signal in "Setup" mode activates "spindle stop" of the related axis.

Prerequisites for stopping the spindle with "AxxC.M5" are:

- Spindle CCW rotation (AxxC.M3) or spindle CW rotation (AxxC.M4) are not applied at the same time

Note: The signal is only available for main spindles.

Spindle positioning (Mx19) "AxxC.M19"

Designation AxxC.M19 = A*xis* xx C*ommand* M19

Control signal PLC →NC

AxxC.M19 : Position spindle

The signal is interpreted only in "Setup" mode.

Function A positive edge of the "AxxC.M19" signal in "Setup" mode issues a spindle positioning command for the related axis.

Prerequisites for positioning the spindle with "AxxC.M19" are:

- Spindle CCW rotation (AxxC.M3) or spindle CW rotation (AxxC.M4) are not applied at the same time

Note: The signal is only available for digital main spindles.

The position that is approached after the signal has been activated is contained in axis parameter Cxx.059 "Parametric spindle jogging distance".

When digital servo drives are used as spindle, the ident number 65008 must be set to logic "1" in order to be able to perform "spindle positioning".

Rapid traverse speed for spindle "AxxC.RAPID"

Designation

AxxC.RAPID = A*xis* xx C*ommand* Rapid

Control signal PLC →NC

AxxC.RAPID	= 0:	No rapid traverse rate during jogging
AxxC.RAPID	= 1:	Rapid traverse rate during jogging

Function

"Automatic" mode
"Semi-automatic" mode
In these modes, the "AxxC.RAPID" signal is not interpreted.

"Manual" mode
"Automatic parameter test" mode
The rapid traverse speed that has been programmed in the setup register is used for manual axis movements (jogging) when the ""rapid traverse for spindle" signal is set.

The spindle rotates at rapid traverse speed as long as the rapid traverse signal is applied.

Note: The signal is only effective for spindle axes.

Changeable software limits "AxxC.LIMIT"

Designation

AxxC.LIMIT = Axis xx C*ommand* Limit

Control signal PLC →NC

AxxC.LIMIT	= 0:	Software limits not active.
AxxC.LIMIT	= 1:	Software limits active.

Function

The AxxC.LIMIT axis control signal activates or deactivates the software limits stored in the machine data.

The variable software limits have no influence in NC blocks which were already calculated by the block preparation when setting the "AxxC.LIMIT" control signal.

The variable software limits must be smaller than the travel range limits set in the axis parameters.

If the axis lies outside the variable software limits upon activation, the limits will take effect in the next axis motion block.

Limiting of feed velocity / spindle speed "AxxC.SPEED"

Designation AxxC.SPEED = Axis xx C*ommand* Speed

Control signal PLC →NC

AxxC.SPEED	= 0:	Feed / speed limiting not activated.
AxxC.SPEED	= 1:	Feed / speed limiting activated.

Function Where a feed axis is concerned, the axis control signal activates the velocity limiting. Where a spindle is concerned, the rotating speed is limited.

Limiting of feed velocity:

The axis control signal activates safety-related velocity limiting.

A set-up engineer can only change the safety-related axis velocity in the machine data if he has the correct password and no NC program is active at this time.

After the PLC interface signal has been set, the safety-related axis velocity will take effect in the next NC block or after an immediate stop. The same applies for a change of the safety-related axis velocity.

Limiting of spindle speed:

The safety-related spindle speed limit can be modified at any time by the PLC using MTD_WR.

The PLC can use the "AxxC.SPEED" axis-specific control signal to activate and deactivate the monitoring of the safety-related spindle speed limit.

The user can see from the "safety-related spindle speed limit" data element in the machine data whether the safety-related spindle speed is effective.

The NC only monitors the spindle speed in relation to the safety-related spindle speed limit in speed mode when the "AxxC.SPEED" control signal is set. The NC takes immediate account of any changes of the safety-related spindle speed limit and the activation or deactivation of the monitoring.

Axis / spindle lock "AxxC.LOCK"

Designation AxxC.LOCK = Axis xx C*ommand* Lock

Control signal PLC →NC

AxxC.LOCK	= 0:	Axis-/Spindle lock off
AxxC.LOCK	= 1:	Axis-/Spindle lock on

A change of edge changes the status for activating or deactivating the axis/spindle lock.

Function The AxxC.LOCK interface signal enables the command value output to the drive to be disabled and the position command value relative to the drive to be held constant. The axis in question is thus locked. The position control circuit remains closed and the remaining following error is compensated.

The position indicator shows the same values during test mode as in normal mode. The controller merely shows the same values for the actual values as for the command values, without the axes moving.

This signal causes the spindle speed command value "0" to be output in test mode for rotating spindles. The spindle speed control circuit remains closed.

The position indicator shows the same values during test mode as in normal mode. The controller merely shows the same values for the spindle speed actual values as for the spindle speed command values, without the spindles moving. If a spindle is under position control, the position command value will be held constant the same as with the feed axes. During test mode the NC does not transmit M19 commands to the drives. The related position changes appear in the position indicator, however, as though the NC had performed the positioning.

Note: When the axis/spindle lock is deactivated, the positions/speed indicator is set to the machine's real actual value (current position value/speed ´0´).

Axis Status Signals

The term **"axis status signals"** covers all signals that are issued from the NC to the PLC in order to report a statement about certain axis modes.

Controller enabling signal "AxxS.RF"

Designation

AxxS.RF = A*xis* xx S*tatus* RF

Status signal NC →PLC

AxxS.RF	= 0:	Controller enabling signal off
AxxS.RF	= 1:	Controller enabling signal on

Function

The "Controller enabling" signal of an axis must never be generated automatically in the PLC. Instead, the PLC must employ the "axis ready for operation" control signal (AxxC.READY) to tell the NC whether the controller enabling signal for the axis shall be activated or de-activated.

The NC commands the axis processor responsible to close the position control loop for the axis. The "controller enabling" signal (AxxS.RF) is output to the PLC when the drive acknowledges controller enabling.

In the PLC program, this signal must be reported to the drive amplifier of the axis without delay and without any further interconnections. Otherwise, the axis processor recognizes that the position control loop is not closed, shuts down the station and issues the error message: "Drive error".

Note: If the digital SERCOS drive interface is used, the controller enabling signal is directly transferred to the drive via the fiber-optic cable. In this case, the "controller enabling" signal that is returned to the PLC is merely a status signal. Transmission to the drive amplifier via the PLC is not necessary here.

Example Using the axis signals for switching the controller enabling signal:

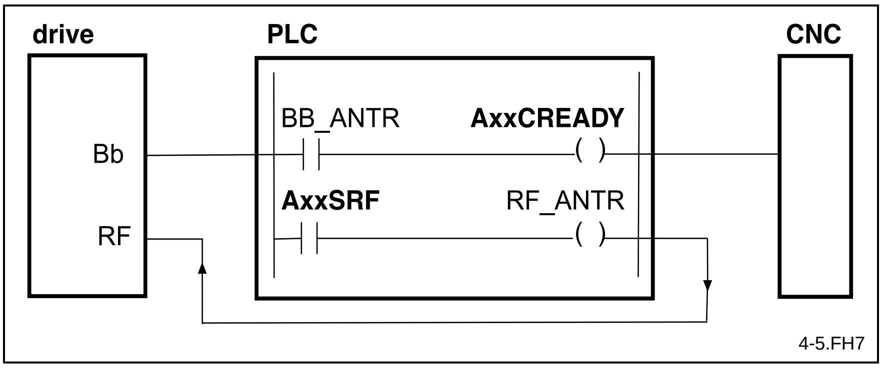


Fig. 1-104: Signal “Controller enabling”

The "AxxS.RF" signal may also be used for triggering a brake (if, for example, the controller enabling signal of a blocked axis is switched off).

Digital drive ready for operation "AxxS.BBDIG"

Designation **AxxS.BBDIG** = **A**xis **xx** **S***tatus* **BB** **Dig***ital Drive*

Status signal NC →PLC

AxxS.BBDIG	= 0:	Digital drive is not ready for operation
AxxS.BBDIG	= 1:	Digital drive is ready for operation

Valid in all modes.

Function The NC employs this signal to report the operational state of a digital drive (in conjunction with the SERCOS fiber-optic ring). The signal is activated as soon as the digital drive's DC bus voltage is established. The drive's diagnostics show "AB" or "AF".

The signal may be directly applied to the "axis ready for operation" control signal "AxxC.READY".

Note: Is only available for digital drives.

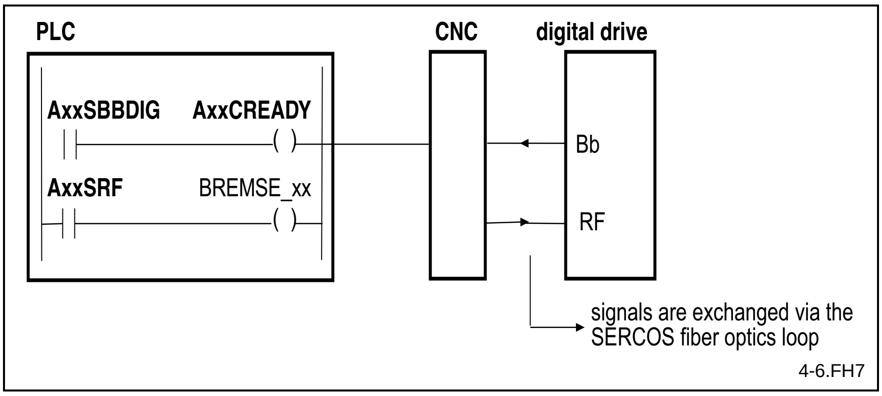


Fig. 1-105: Signal “Digital drive ready”

SERCOS (Serial Real-time Communication System) - digital interface for the communication between controllers and drives in numerically controlled machines.

The NC transfers the signal that is sent via the SERCOS fiber-optic ring to the PLC without modification. Processing the "digital drive ready for operation" signal in the PLC program is identical to processing the "BB" signal of an analog drive amplifier.

Axis homed "AxxS.HOMED"

Designation AxxS.HOMED = Axis xx *Status is* Homed

Status signal NC →PLC

AxxS.HOMED = 0: Axis is not homed

AxxS.HOMED = 1: Axis homed

Function The NC employs this signal to tell the PLC whether or not the axis has been homed.

With incremental position measuring systems, reference must be re-established whenever the controller is switched on. This can be done manually for each individual axis using the "axis homing" signal (AxxC.HOME) or be programmed via the valid NC reverse program.

Single-axis homing of all axes of a process does not make the NC program of that process ready to start. The ready-to-start condition can only be attained by executing an NC reverse program (see process signals: "Ready to start")

The "Axis is homed" signal is used as a validity signal for the waypoint signals of that axis.

Axis movement in positive direction "AxxS.MVPOS"

Designation **AxxS.MVPOS** = **A**xis **xx** **S***tatus* **M***ove* **Pos***itive*

Status signal NC →PLC

AxxS.MVPOS = 0: The axis is stopped or moves in the negative direction

AxxS.MVPOS = 1: The axis moves in the positive direction

The NC employs this signal to tell the PLC that the axis is NC-controlled and is moving in the positive direction.

Note: This signal does not show that the axis is intended to be moved in the positive direction. It shows that a command value has already been issued to the axis. The signal can not be used to activate an axis that has been deactivated.

The signal is not output when digital axes are homed.

Axis movement in negative direction "AxxS.MVNEG"

Designation **AxxS.MVNEG** = **A**xis **xx** **S***tatus* **M***ove* **Neg***ative*

Status signal NC →PLC

AxxS.MVNEG = 0: The axis is stopped or moves in the positive direction

AxxS.MVNEG = 1: The axis moves in the negative direction

The NC employs this signal to tell the PLC that the axis is NC-controlled and is moving in the negative direction.

Note: This signal does not show that the axis is intended to be moved in the negative direction. It shows that a command value has already been issued to the axis. The signal can not be used to activate an axis that has been deactivated.

The signal is not output when digital axes are homed.

Waypoints "AxxS.WPn"

Designation **AxxS.WPn** = **A**xis **xx** **S***tatus* **W***ay* **P***oint*

8 Status signals NC →PLC

AxxS.WPn = 0: The active axis position is smaller than the position of the n-th waypoint

AxxS.WPn = 1: The active axis position is greater than the position of the n-th waypoint

Function The Rexroth MTC 200 permits up to 8 positions to be entered in the axis parameters of each axis. Each position (0 ... 7) is allocated to a status signal "AxxS.WPn" (n = 0 - 7).

Independently of the NC program, the status of the individual signals shows whether the current axis position is less than or greater than the position that is stored in the associated axis parameter.

These waypoints permit working area protection to be programmed.

Note: The NC always outputs the waypoints, even if the axis has not yet been homed. The Rexroth MTC 200 stores the axis positions when the controller is switched off, and restores them when it is switched on. It can therefore be assumed that, even without homing, the positions may be employed for performing rough monitoring activities using the waypoints. Obviously, this is not true after the machine has been manually manipulated or after new parameters have been loaded. The absolute validity of the waypoints can only be assured by including the "axis homed" signal (AxxS.HOMED).

Reading a SERCOS real-time status bit "AxxS.IDDS"

Designation **AxxS.IDDS** = **A**xis **xx** **S***tatus* **I***nput of* **DDS**

Control signal NC →PLC

AxxS.IDDS = 0: Real time status bit is 0

AxxS.IDDS = 1: Real time status bit is 1

Function This signal enables the PLC to read a real-time status bit in the drive via the NC.

The status that is read from the drive depends on the allocation of an ident number to the real-time status bit in the SERCOS parameters.

Please refer to the "SERCOS Interface" description for a list of the parameters that can be allocated to a real-time bit and for an explanation of the procedure for allocating a parameter to the real-time bit.

Example The drive software interprets the actuating contact of a sensor.

According to the state of this contact, the PLC sets an "event" that is used for conditionally executing NC program segments in the associated NC program. The state of the actuating contact is transferred to the PLC via a real-time status bit.

PLC command sequence:

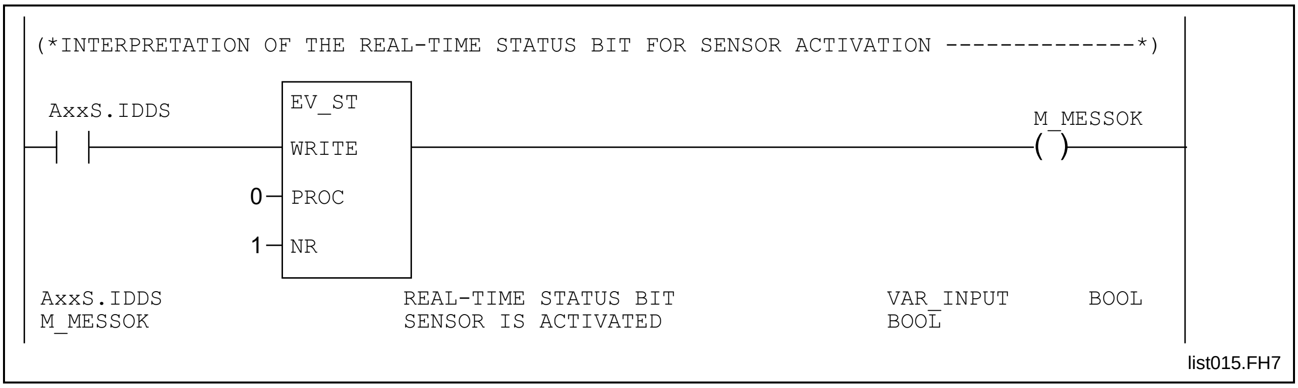


Fig. 1-106: Evaluation of the real-time status bit for sensor deflection

Allocation of the associated SERCOS parameter:

Ident Number for "allocation of real time control bit": S-0-0307
Ident. number of "sensor enabled": S-0-0401

The ident number for the "sensor" function is written to the SERCOS "real-time control bit allocation" parameter.

Parameter list:

.

.

ID No. S-0-0307: 00401

.

.

This allocation can be used in the PLC for interpreting sensor activation via the "AxxS.IDDS" signal.

Announcement of motion in positive direction "AxxS.MCPOS"

Designation

AxxS.MCPOS = **A**xis **xx** **S***tatus* **M***otion* **C***ommand* **Pos***itive*

Status signal NC →PLC

AxxS.MCPOS = 0: There is no motion announcement for the axis

AxxS.MCPOS = 1: There is a motion announcement for the axis

The signal is effective in all modes.

Function

The signal is set if there is a motion announcement in positive direction for the corresponding axis (including spindle axes). The signal remains applied as long as the axis is moving. The motion announcement signals are reset if the motion is interrupted by a stop.

Note: Both motion announcement signals are set during homing, circular movements, and in handwheel mode.

Announcement of motion in negative direction "AxxS.MCNEG"

Designation **AxxS.MCNEG** = **A**xis **xx** **S***tatus* **M***otion* **C***ommand* **Neg***ative*

Status signal NC →PLC

AxxS.MCNEG = 0: There is no motion announcement for the axis

AxxS.MCNEG = 1: There is a motion announcement for the axis

The signal is effective in all modes.

Function The signal is set if there is a motion announcement in the negative direction for the corresponding axis (including spindle axes).

The signal remains applied as long as the axis is moving.

The motion announcement signals are reset if the motion is interrupted by a stop.

Note: Both motion announcement signals are set during homing, circular movements, and in handwheel mode.

Axis in positioning window "AxxS.POSWN"

Designation **AxxS.POSWN** = **A**xis **xx** **S***tatus in* **Pos***ition* **W**i**n***dow*

Status signal NC →PLC

AxxS.POSWN = 0: The axis is outside the positioning window

AxxS.POSWN = 1: The axis is within the positioning window

The signal is effective in all modes.

Function The signal is set once the axis has reached the programmed position and is in the associated positioning window. The signal is reset once the axis leaves the specified positioning window.

Note: The "AxxS.POSWN" signal is not set during continuous jogging. The signal is issued in incremental jogging. The signal is not output after the Mx05 "Spindle stop" spindle command.

90% load is exceeded "AxxS.LD90"

Designation **AxxS.LD90** = **A**xis **xx** **S***tatus* **L**o*a***d** **90** *%*

Status signal NC →PLC

AxxS.LD90 = 0: 90% load not exceeded

AxxS.LD90 = 1: 90% load exceeded

The signal is effective in all modes.

Function The signal is set when the spindle is used in a range in which 90% of its maximum load is exceeded.

Note: The signal is only valid for digital main spindle drives.

Spindle synchronous run "AxxS.SYNC" "AxxS.SYNC"

Designation

AxxS.SYNC = **A**xis **xx S***tatus* **Sync***hron Run*

Status signal NC →PLC

AxxS.SYNC = 0: Synchronous run difference outside of the synchronous run window

AxxS.SYNC = 1: Synchronous run difference within the synchronous run window

The signal is effective in all modes.

Function

The signal turns "1" if the difference between the synchronous position command value or velocity command value of the leadscrew and the synchronous position command value or velocity command value of the synchronous spindle are inside the programmed synchronous run window.

List of the influencing SERCOS parameters:

Parameter	SERCOS ident No.
Synchronous run window position	S-0-0228
Synchronous run window velocity	S-0-0183

Fig. 1-107: SERCOS parameter "Spindle synchronous run"

Note: The signal is only valid for digital main spindle drives.

Spindle command speed reached (N_{act} = N_{com}) "AxxS.N_CMD"

Designation

AxxS.N_CMD = A*xis* xx S*tatus* N = N_{CMD}

Status signal NC →PLC

AxxS.N_CMD = 0: Command speed not yet reached

AxxS.N_CMD = 1: Command speed reached

The signal is effective in all modes.

Function

The signal is set if the difference between the actual velocity value and the velocity command value is within a programmed velocity window.

List of the influencing SERCOS parameters:

Parameter	SERCOS ID No.
Velocity window	S-0-0157
Actual velocity value	S-0-0040
Velocity command value	S-0-0036

Fig. 1-108: SERCOS parameter "Spindle command speed reached"

Note: The signal is only valid for digital main spindle drives.

Spindle speed 0 reached (N N_{min}) "AxxS.N_MIN"

Designation AxxS.N_MIN = A*xis* xx *Status* N ≤N_{min}

Status signal NC →PLC

AxxS.N_MIN	= 0:	Spindle speed N_{min} not yet reached
AxxS.N_MIN	= 1:	Spindle speed < N_{min} reached

The signal is effective in all modes.

Function This signal is set when the actual velocity value is inside the standstill window.

List of the influencing SERCOS parameters:

Parameter	SERCOS ID No.
Standstill window	S-0-0124

Fig. 1-109: SERCOS parameter Spindle speed 0 reached

Note. The "AxxS.N_MIN" signal does <u>not</u> fulfill the personal safety requirements.

The signal is valid only for digital main spindle drives.

Velocity command value exceeded ($N_{com} \geq N_{Max}$) "AxxS.N_MAX"

Designation AxxS.N_MAX = A*xis* xx *Status* $N_{soll} \geq N_{Max}$

Status signal NC →PLC

AxxS.N_MAX	= 0:	Velocity command value not exceeded
AxxS.N_MAX	= 1:	Velocity command value exceeded

The signal is effective in all modes.

Function The signal is set when the velocity command value exceeds a velocity limit value.

List of the velocity limit values:

Parameter	SERCOS ID No.
Positive velocity limit value	S-0-0038
Negative velocity limit value	S-0-0039
Bipolar velocity limit value	S-0-0091

Fig. 1-110: SERCOS parameter "Velocity command value exceeded"

Note: The signal is only valid for digital main spindle drives.

Actual spindle torque value comparison ($M_d \geq M_{dx}$) "AxxS.MD_DX"

Designation

AxxS.MD_DX = A*xis* xx S*tatus* $M_d \geq M_{dx}$

Status signal NC →PLC

AxxS.MD_DX = 0: Actual spindle torque value < torque threshold
AxxS.MD_DX = 1: Actual spindle torque value > torque threshold

The signal is effective in all modes.

Function

The "AxxS.MD_DX" signal is set if the spindle's actual torque value is greater than the selected torque threshold.
List of the influencing SERCOS parameters:

Parameter	SERCOS ID No.
Actual torque value	S-0-0084
Torque threshold	S-0-0126

Fig. 1-111: SERCOS parameter "Spindle torque"

Note: The signal is only valid for digital main spindle drives.

Spindle is in position "AxxS.INPOS"

Designation

AxxS.INPOS = **A**xis **xx S***tatus* **in Pos***ition*

Status signal NC →PLC

AxxS.INPOS = 0: Spindle not in position
AxxS.INPOS = 1: Spindle has reached position

The signal is effective in all modes.

Function

The "AxxS.INPOS" signal is issued if, with respect to the position command value, the actual position value is within the positioning window after an "Mx19" has been processed.
List of the related SERCOS parameters:

Parameter	SERCOS ID No.
In-position window	S-0-0057
Position command value	S-0-0047

Fig. 1-112: SERCOS parameter "Spindle is in position"

Note: The signal is only valid for digital main spindle drives.

Spindle power output ($P = P_x$) "AxxS.P_PX"

Designation

AxxS.P_PX = A*xis* xx S*tatus* $P = P_x$

Status signal NC →PLC

AxxS.P_PX	= 0:	Power output smaller than power threshold
AxxS.P_PX	= 1:	Power output greater than power threshold

The signal is effective in all modes.

Function The signal is set if the output power is greater than the power threshold.

The power threshold value (i.e. the value at and above which the "AxxS.P_PX" signal shall be generated) can be modified in the SERCOS parameters.

List of the related SERCOS parameters:

Parameter	SERCOS ID No.
Power threshold	S-0-0158

Fig. 1-113: SERCOS parameter "Spindle power output"

Note: The signal is only valid for digital main spindle drives.

Axis signals for intelligent safety technology (IST)

The following interface signals, among others, are required to make use of IST functions. Please refer to the description "DOK-CONTRL-IST********-AW0x-DE-P" for details of how the signals work.

Axis control signals for IST

Axis control signal	Comment
AxxC.SAFSS	Activating safe standstill
AxxC.SAFOS	Activating safe operational stop
AxxC.SAFA1	Activating safely reduced velocity with safely limited abs. position 1
AxxC.SAFA2	Activating safely reduced velocity with safely limited abs. position 2
AxxC.SAFAG	Safety agree sensor
AxxC.SAFRS	Reference cam for safe referencing (Safe Refe-rence Switch)

Fig. 1-114: Axis control signals for IST

Axis status signals for IST

Axis status signal	Comment
AxxS.SAFAC	Safety function active
AxxS.SAFRY	Switching closed (Safety Changeover Rea-dy)
AxxS.SAFP1	Safe position switch 1
AxxS.SAFP2	Position switch point 2
AxxS.SAFP3	Position switch point 3
AxxS.SAFP4	Position switch point 4
AxxS.SAFSL	Safe starting lockout
AxxS.SAFEN	Safety enable

Fig. 1-115: Axis status signals for IST

Limiting of feed velocity / spindle speed "AxxS.SPEED"

Designation

AxxS.SPEED = **A**xis **xx S***tatus* **Speed**

Status signal NC →PLC

AxxS.SPEED = 0: Feed / spindle limiting not activated.

AxxS.SPEED = 1: Feed / spindle limiting is activated.

Limiting of feed velocity:

The axis status signal is set as soon as safety-related velocity limiting is active.

If the selected velocity command value exceeds the maximal permissible velocity in program or test mode, the NC will limit the programmed command value and set the "command value is limited" (AxxS.SPEED) interface signal.

If the actual axis velocity lies outside the safety-related velocity limits at the moment the "activate limiting" interface signal is applied, the NC will decelerate that axis to the lowest maximal limit velocity with due regard for the other velocities.

The NC updates the status signal in the following cases:

- with each command value specification,
- and upon each change of axis velocity limits.

When forming this status signal, the NC does not take account of any changes of the command value that are caused by feed override.

Limiting of spindle speed:

If the selected velocity command value exceeds the maximum permissible velocity in program or test mode, the NC will limit the programmed command value and set the "command value is limited" (AxxS.SPEED) interface signal.

If the operator changes a spindle speed limit in the NC or PLC program or if the PLC activates or deactivates the monitoring of the safety-related spindle speed limits, the NC will immediately take account of such a change.

When main spindle synchronization is switched on, the NC calculates the minimal and maximal permissible speed of the leadscrew, taking due account of all transmission ratios and speed limits. The NC recalculates the maximum speed of the leadscrew each time thbat the speed limit for the leadscrew or synchronous spindles changes or each time a safety-related speed limit is activated or deactivated. If limiting is necessary, the NC performs it synchronously for all the spindles participating in the synchronization and indicates this for the spindles by setting the ´command value is limited´ (AxxS.SPEED) status signals.

If the actual spindle speed lies outside the safety-related speed limits at the moment the "activate limiting" interface signal is applied, the NC will decelerate that spindle to the lowest maximal limit speed with due regard for the other speed limits.

If a coupling (G33, G63, G64, G65, G95) or main spindle synchronization is active at the moment of activation, the NC will adapt not only the spindle speed but also the path velocity of the feed axes and/or spindles involved in the coupling or synchronization.

The NC updates the status signal in the following cases:

- with each command value specification (including specification of cutting rate),

- upon constant cutting rate (G96), as soon as the command value reaches one of the preselected speed limits, and upon each change of speed limits.

When forming this status signal, the NC does not take account of any changes of the command value that are caused by spindle override.

Where PLC-controlled spindles are concerned, the limited speed has to be read and transmitted to the external spindle upon each positive edge of this status signal.

Changeable software limits "AxxS.LIMIT"

Designation AxxS.LIMIT = **A**xis **xx** **S***tatus* **Limit**

Status signal NC →PLC

AxxS.LIMIT	= 0:	Software limits were not activated.
AxxS.LIMIT	= 1:	Activation of software limits requested

The status signal is set as soon as the control signal has been transmitted by the PLC and is interpreted by the NC.

Axis / spindle lock "AxxS.LOCK"

Designation AxxS.LOCK = **A**xis **xx** **S***tatus* **Lock**

Status signal NC →PLC

AxxS.LOCK	= 0:	Axis/spindle lock is not active
AxxS.LOCK	= 1:	Axis/spindle lock is active

The status signal indicates that the axis/spindle lock is activated.

Motion hold active "AxxS.MHOLD"

Designation AxxS.MHOLD = **A**xis **xx** **S***tatus* **M***otion* **Hold**

Status signal NC →PLC

AxxS.MHOLD	= 0:	Motion hold is not active
AxxS.MHOLD	= 1:	Motion hold is active

Function The signal is set after the PLC has requested motion hold (AxxC.MHOLD). The signal is only set for the axis for which a motion hold request was transmitted, irrespective of the interpolation compound.

The PLC must first receive an acknowledgment feedback signal before it can assume that motion hold is active.

2 Functions Rexroth MTC 200

2.1 Auxiliary Functions

M functions

Interrogating an M function and specifying the number "M_FKT"

The standard function "M_FKT" permits M functions to be interrogated from the PLC user program.

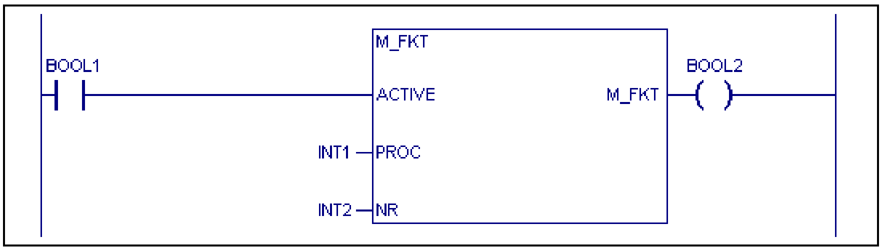


Fig. 2-1: Interrogate M functions

ACTIVE:	BOOL	0	M function interrogation not active
		1	M function interrogation is active
PROC:	INT		Process number [0..6]
NR:	INT		M function number (0..999)

Function result

M_FKT	: BOOL	0	The standard function is not active, or the standard function is active and the NC has not transferred the M function concerned.
	1		The standard function is active and the NC has output the M function concerned.

Note: The function for interrogating the auxiliary functions and specifying the number can be used for all auxiliary functions. The last M function that has been output and not yet acknowledged is recognized.

Acknowledging an M auxiliary function and specifying the number "M_FKT_Q"

All M functions that are transferred from the NC to the PLC must be acknowledged in the PLC user program.

A swift auxiliary function "MQxx" that is executed in an NC program can be interrogated with "M_FKT" or "M_ALL". The auxiliary function is no longer pending after it has been acknowledged with "M_FKT_Q" or "M_ALL_Q". The "M_FKT" and "M_ALL" functions supply the function result "FALSE". If the command "MWxx" is subsequently executed in the NC program, the acknowledgment with "M_FKT_Q" or "M_ALL_Q" is waited for.

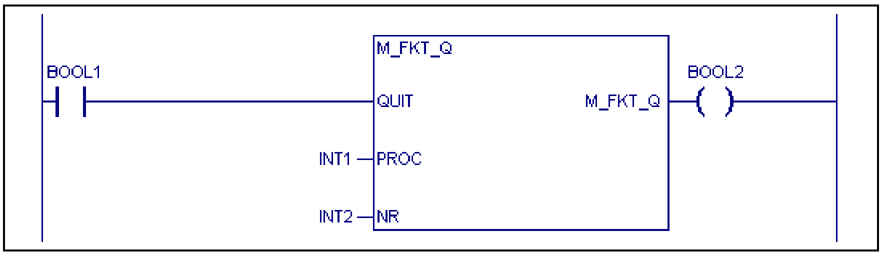


Fig. 2-2: Acknowledge M functions

QUIT:	BOOL	0	No acknowledgment of the M function
		1	The M function is acknowledged
PROC:	INT		Process number [0..6]
NR:	INT		M function number (0..999)

Function result M_FKT_Q : BOOL The QUIT function input is passed on

Note: The function for acknowledging the auxiliary functions and specifying the number can be used for all auxiliary functions.

Acknowledging an M auxiliary function and specifying "M_ALL"

Using the "M_ALL" standard function, the PLC can recognize an M function that is output by the NC. The standard function permits time-optimized program execution to be performed, since it enables the PLC program sections for auxiliary function processing to be skipped if there is no change.

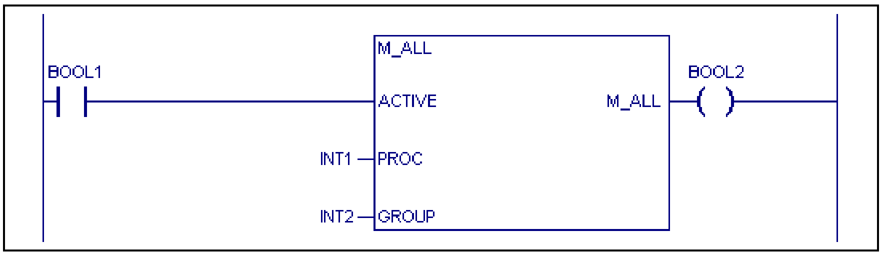


Fig. 2-3: Interrogate all M functions

ACTIVE:	BOOL	0	M function change interrogation is not active.
		1	M function change interrogation is active
PROC:	INT		Process number [0..6]
GROUP:	INT	Auxiliary function group (0..16)	
		0	Inquiry whether the NC has output an M function in one of the 16 M function groups 1...16
		1..16 -	Inquiry whether the NC has output an M function in the specified group (1...16)

Function result

M_ALL:	BOOL	0	The standard function is not active, or the standard function is active and there is no change.
		1	The standard function is active and there is a change.

Note: The "M_ALL" function supplies the result "1" when the NC outputs an auxiliary function of the "GROUP" group.

Acknowledging an M auxiliary function and specifying the group "M_ALL_Q"

All M functions that are transferred from the NC to the PLC must be acknowledged in the PLC user program.

A swift auxiliary function "MQxx" that is executed in an NC program can be interrogated with "M_FKT" or "M_ALL". The auxiliary function is no longer pending after it has been acknowledged with "M_FKT_Q" or "M_ALL_Q". The "M_FKT" and "M_ALL" functions supply the function result "FALSE". If the command "MWxx" is subsequently executed in the NC program, the acknowledgment with "M_FKT_Q" or "M_ALL_Q" is waited for.

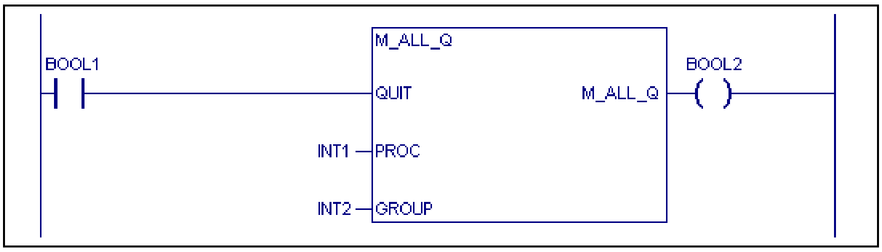


Fig. 2-4: Acknowledging all M functions

QUIT:	BOOL	0 No acknowledgment of the M function 1 The M function is acknowledged
PROC:	INT	Process number [0..6]
GROUP:	INT	Auxiliary function group (0..16) 0 Acknowledgment of all pending M functions in all auxiliary function groups 1..16 - Acknowledgment of all pending M functions in the selected M function group (1...16)

Function result M_ALL_Q : BOOL The QUIT function input is passed on

Note: The "M_ALL_Q" function acknowledges all pending auxiliary functions of the "GROUP" group.

Reading the M auxiliary function number "M_NR"

The "M_NR" standard function permits the M function number (of each function group 1 through 16) from the NC to be read. Reading the M function number permits the state of the M function as it exists in the NC (and appears on the status display) to be retrieved for each M function group at any time.

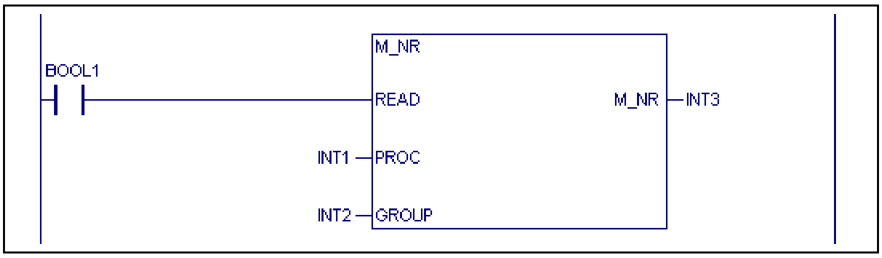


Fig. 2-5: Reading the M function number

READ: BOOL 0 Reading of M function number not active
1 Reading the M function number is active
PROC: INT Process number [0..6]
GROUP: INT M function group number
0 Reading the M function numbers that have not yet been acknowledged in one of the 16 function groups (starting from group 1)
1..16 - Reading the last M function number to have been output within the selected group 1 through 16.

Function result M_NR: INT
GROUP = 0
1:
An M function is not active.
0 bis 999:
The last M function number to have been output and not yet acknowledged. The M function number of the lowest group is displayed when auxiliary functions are output at the same time.
GROUP = 1 through 16
-1:
An M function has not yet been output for the M function group concerned since the time the controller has been switched on.
0.999:
The last M function number to have been output.
It remains pending after acknowledgment until the next M function is output (in the corresponding group).

Note: Within an PLC cycle, one M auxiliary function number can be read for each group (GROUP 1 ... 16). If "0" is applied to the "GROUP" function input, the "M_NR" function only reports the next active M function after the previous M function has been acknowledged.

S Functions

Interrogation an S auxiliary function and specifying the number "S_FKT"

The standard function "S_FKT" permits S functions to be interrogated from the PLC user program.

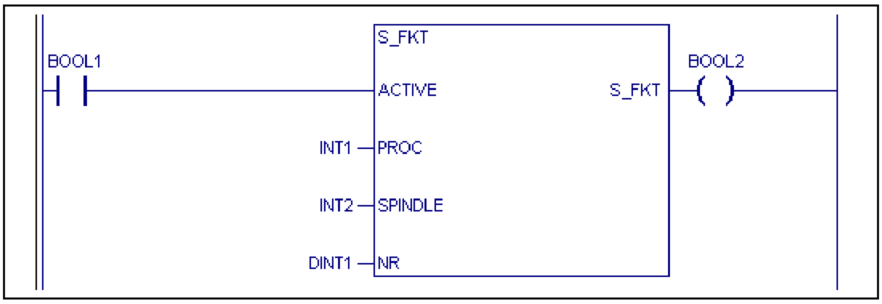


Fig. 2-6: Interrogate S functions

ACTIVE: BOOL 0 S function interrogation not active
1 S function interrogation is active
PROC: INT Process number [0..6]
SPINDLE:INT Spindle number (1..3)
NR: DINT S function number (0.0.99999) without fractional part digits

Function result S_FKT : BOOL 0 The standard function is not active, or the standard function is active and the NC has not transferred the S function concerned.
1 The standard function is active and the NC has output the S function concerned.

Note: The spindle designation "S" corresponds to the spindle number "S1".

The "S_FKT_Q" standard function can be used for acknowledging the S functions the user has initiated using the "Interrogating an auxiliary function" standard function.

With swift S functions, acknowledgment is not necessary since the execution of those functions cannot be interrogated from the NC program.

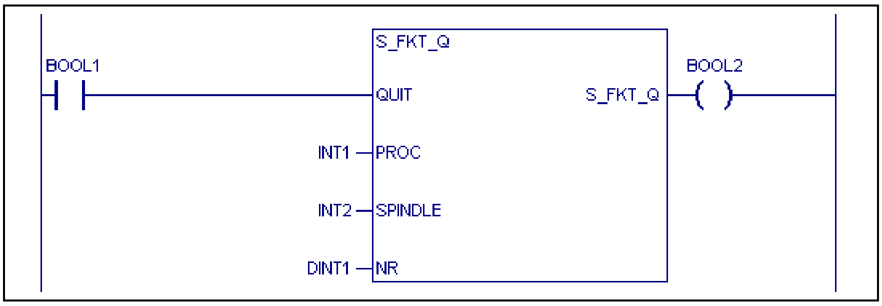


Fig. 2-7: Acknowledge S functions

QUIT:	BOOL	0 No acknowledgment of the S function
		1 The S function is acknowledged
PROC:	INT	Process number [0..6]
SPINDLE:	INT	Spindle number (1..3)
NR:	DINT	S function number (0.0.99999) without fractional part digits

Function result S_FKT_Q : BOOL The QUIT function input is passed on

Interrogation an S auxiliary function without specifying the number "S_ALL"

Using the "S_ALL" standard function, the PLC can recognize an S function that is output by the NC. The standard function permits time-optimized program execution to be performed, since it enables the PLC program sections for auxiliary function processing to be skipped if there is no change.

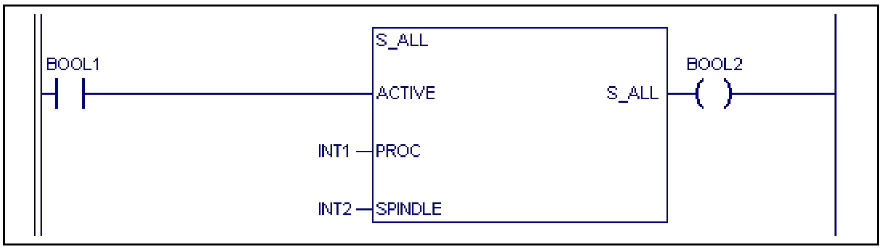


Fig. 2-8: Interrogate all S functions

ACTIVE:	BOOL	0 S function change interrogation is not active.
		1 S function change interrogation is active
PROC:	INT	Process number [0..6]
SPINDLE:	INT	Spindle number (1..3)
		1..3 Inquiry whether the NC has output an M function in the specified group (1...16)

Function result

S_ALL:	BOOL	0 The standard function is not active, or the standard function is active and there is no change.
		1 The standard function is active and there is a change.

The "S_ALL_Q" standard function permits all pending S functions for the selected spindle to be acknowledged.

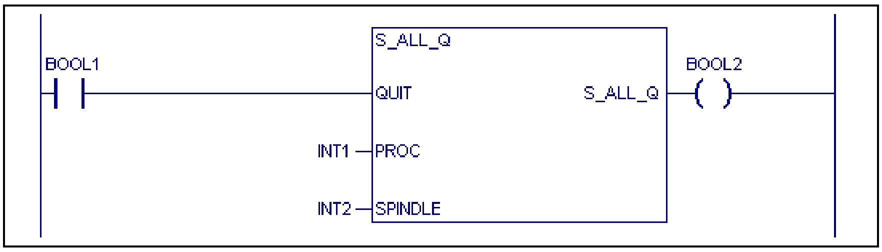


Fig. 2-9: Acknowledging all S functions

QUIT:	BOOL	0 No acknowledgment of the S function 1 The S function is acknowledged
PROC:	INT	Process number [0..6]
SPINDLE:		INT Spindle number (1..3) 1..3 - Acknowledgment of all pending S functions of the selected spindle (1 through 3)

Function result S_ALL_Q : BOOL The QUIT function input is passed on

The "S_NR" standard function permits the S function number (of each spindle 1 ... 3) that is transferred from the NC to be read. Reading the S function number thus permits the state of the S function as it exists in the NC (and appears on the status display) to be retrieved for each spindle. The spindle value is read without fractional part digits.

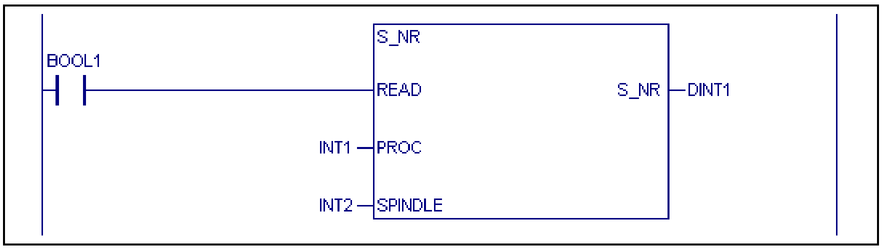


Fig. 2-10: Interrogate of the S function numbers

READ:	BOOL	0 Reading of S function number not active 1 Reading the S function number is active.
PROC:	INT	Process number [0..6]
SPINDLE:INT		Spindle number (1..3)

Function result

S_NR:	DINT	-1 An S function has not yet been output for the S function group concerned since the time the controller was switched on. 0.0.99999 The last S function number to have been output. It remains pending after acknowledgment until the next S function is output.

Note: The spindle value is read without fractional part digits.

T Functions

The standard function "T_FKT" permits T functions to be interrogated from the PLC user program.

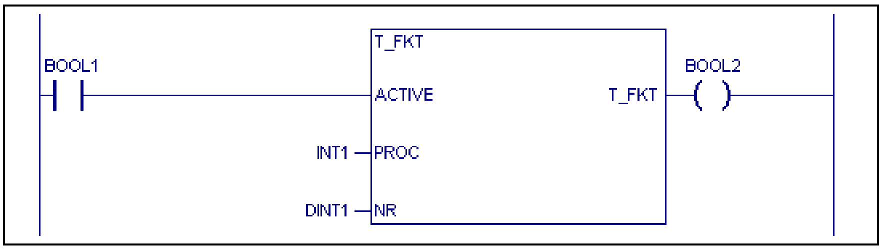


Fig. 2-11: Interrogate T functions

ACTIVE:	BOOL	0	T function interrogation not active
		1	T function interrogation is active
PROC:	INT		Process number [0..6]
NR:	DINT		T function number (0.00.9999999)

Function result

T_FKT	: BOOL	0	The standard function is not active, or the standard function is active and the NC has not transferred the T function concerned.
		1	The standard function is active and the NC has output the T function concerned.

The "S_FKT_Q" standard function can be used for acknowledging the T functions the user has initiated using the "Interrogating an auxiliary function" standard function.

With swift T functions, acknowledgment is not necessary since the execution of those functions cannot be interrogated from the NC program.

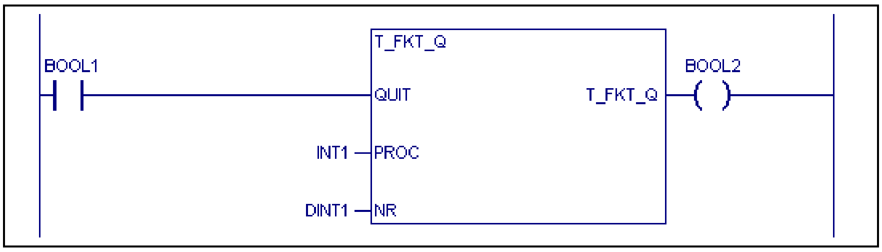


Fig. 2-12: Acknowledge T functions

QUIT:	BOOL	0	No acknowledgment of the T function
			The T function is acknowledged
PROC:	INT		Process number [0..6]
NR:	DINT		T function number (0.00.9999999)

Function result

T_FKT_Q : BOOL The QUIT function input is passed on

Using the "T_ALL" standard function, the PLC can recognize a T function that is output by the NC. The standard function permits time-optimized program execution to be performed, since it enables the PLC program sections for auxiliary function processing to be skipped if there is no change.

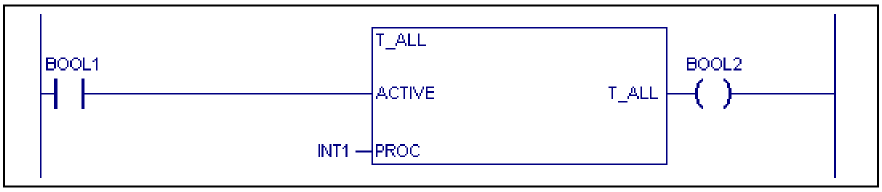


Fig. 2-13: Interrogate all T functions

ACTIVE:	BOOL	0	T function change interrogation is not active.
		1	T function change interrogation is active
PROC:	INT		Process number [0..6]

Function result T_ALL: BOOL

0 The standard function is not active, or the standard function is active and there is no change.

1 The standard function is active and there is a change.

The "T_ALL_Q" standard function permits all pending T functions to be acknowledged.

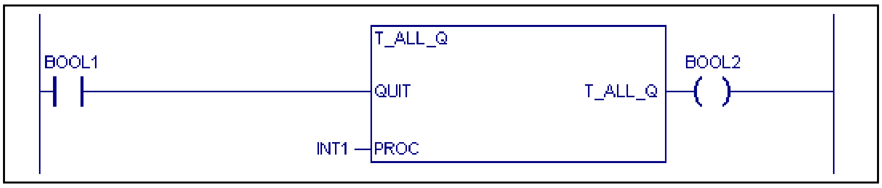


Fig. 2-14: Acknowledging all T functions

QUIT:	BOOL	0	No acknowledgment of the T function
		1	The T function is acknowledged
PROC:	INT		Process number [0..6]

Function result T_ALL_Q : BOOL The QUIT function input is passed on

The "T_NR" standard function permits the T function number that is transferred from the NC to be read. Reading the T function number permits the state of the T function as it exists in the NC (and appears on the status display) to be retrieved for each spindle.

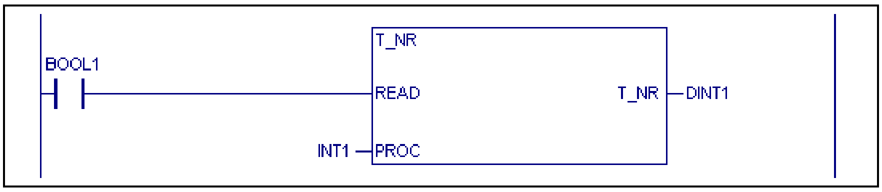


Fig. 2-15: Interrogating the T function number

READ:	BOOL	0	Reading of T function interrogation not active
		1	Reading the T function interrogation is active
PROC:	INT		Process number [0..6]

Function result T_NR: DINT

-1 A T function has not yet been output since the time the controller was switched on.

0..9999999
The last T function number to have been output. It remains pending after acknowledgment until the next T function is output.

Q functions

The standard function "Q_FKT" permits "Q functions" to be interrogated from the PLC user program.

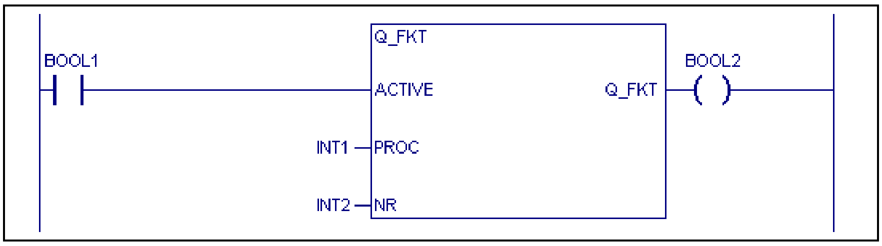


Fig. 2-16: Interrogate Q functions

ACTIVE:	BOOL	0	Q function interrogation not active
		1	Q function interrogation is active
PROC:	INT		Process number [0..6]
NR:	INT		Q function number (0.0.9999)

Function result Q_FKT : BOOL 0 The standard function is not active, or the standard function is active and the NC has not transferred the Q function concerned.

All Q functions that are transferred from the NC to the PLC must be acknowledged in the PLC user program.

A swift auxiliary function "QQxx" that is executed in an NC program can be interrogated with "Q_FKT" or "Q_ALL". The auxiliary function is no longer pending after it has been acknowledged with "Q_FKT_Q" or "Q_ALL_Q". The "Q_FKT" and "Q_ALL" functions supply the function result "FALSE". If the command "QWxx" is subsequently executed in the NC program, the acknowledgment with "Q_FKT_Q" or "Q_ALL_Q" is waited for.

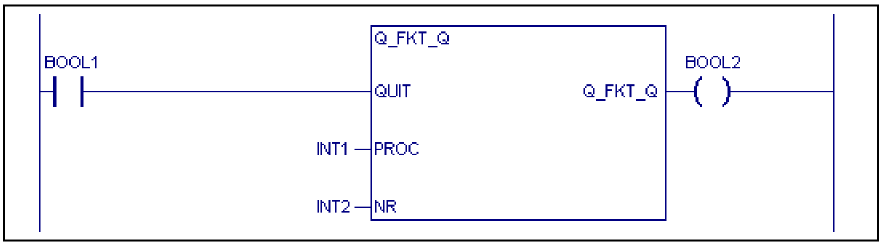


Fig. 2-17: Acknowledge Q functions

QUIT:	BOOL	0	No acknowledgment of the auxiliary function
		1	The auxiliary function is acknowledged
PROC:	INT		Process number [0..6]
NR:	INT		Q function number (0.0.9999)

Function result Q_FKT_Q : BOOL The ACTIVE function input is passed on

Using the "Q_ALL" standard function, the PLC can recognize a Q function that is output by the NC. The standard function permits time-optimized program execution to be performed, since it enables the PLC program sections for auxiliary function processing to be skipped if there is no change.

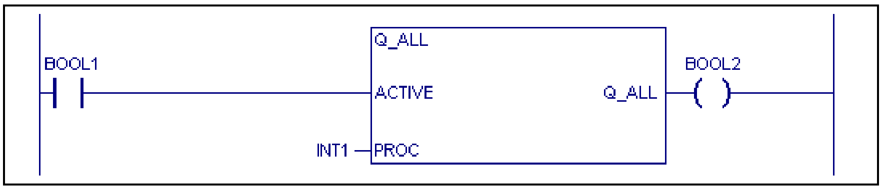


Fig. 2-18: Interrogate all Q functions

	ACTIVE:	BOOL	0 Q function change interrogation is not active. 1 Q function change interrogation is active
	PROC:	INT	Process number [0..6]
Function result	Q_ALL:	BOOL	0 The standard function is not active, or the standard function is active and there is no change. 1 The standard function is active and there is a change.

All Q functions that are transferred from the NC to the PLC must be acknowledged in the PLC user program.

A swift auxiliary function "QQxx" that is executed in an NC program can be interrogated with "Q_FKT" or "Q_ALL". The auxiliary function is no longer pending after it has been acknowledged with "Q_FKT_Q" or "Q_ALL_Q". The "Q_FKT" and "Q_ALL" functions supply the function result "FALSE". If the command "QWxx" is subsequently executed in the NC program, the acknowledgment with "Q_FKT_Q" or "Q_ALL_Q" is waited for.

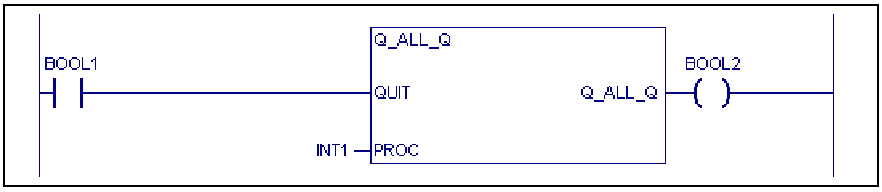


Fig. 2-19: Acknowledging all Q functions

	QUIT:	BOOL	0 No acknowledgment of the Q function 1 The Q function is acknowledged
	PROC:	INT	Process number [0..6]
Function result	Q_ALL_Q :	BOOL	The ACTIVE function input is passed on

The "Q_NR" standard function permits the Q function number that is transferred from the NC to be read.

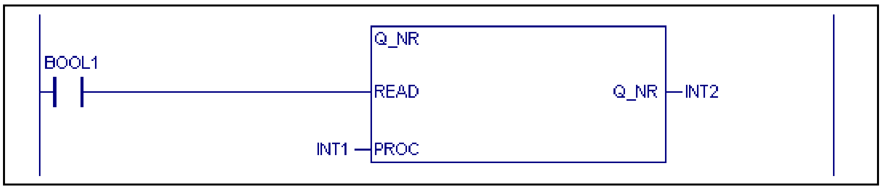


Fig. 2-20: Interrogating the Q function number

	READ:	BOOL	0 Auxiliary function interrogation not active 1 Reading the auxiliary function is active
	PROC:	INT	Process number [0..6]
Function result	Q_NR	: INT	-1 A Q function has not yet been output since the time the controller was switched on. 0 to 999: The last Q function number to have been output. It remains pending after acknowledgment until the next Q function is output.

E Functions

The standard function "E_FKT" permits "E functions" to be interrogated from the PLC user program.

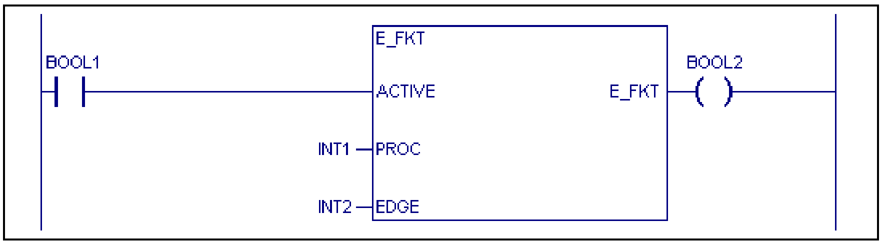


Fig. 2-21: Interrogate E functions

ACTIVE:	BOOL	0	E function interrogation not active
		1	E function interrogation is active
PROC:	INT		Process number [0..6]
EDGE:	INT	Tool edge number (0..9)	

Function result E_FKT : BOOL

0 The standard function is not active, or the standard function is active and the NC has not transferred the E function concerned.

1 The standard function is active and the NC has output the E function concerned.

The "E_FKT_Q" standard function can be used for acknowledging the E functions the user has initiated using the "Interrogating an auxiliary function" standard function.

With swift E functions, acknowledgment is not necessary since the execution of those functions cannot be interrogated from the NC program.

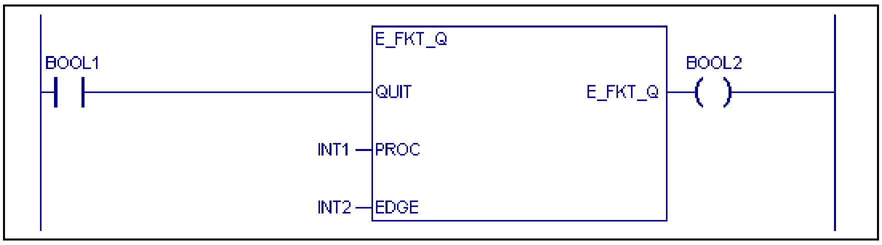


Fig. 2-22: Acknowledge E functions

QUIT:	BOOL	0	No acknowledgment of the E function
		1	The E function is acknowledged
PROC:	INT		Process number [0..6]
EDGE:	INT	Tool edge number (0..9)	

Function result E_FKT_Q : BOOL The QUIT function input is passed on

Using the "E_ALL" standard function, the PLC can recognize an E function that is output by the NC. The standard function permits time-optimized program execution to be performed, since it enables the PLC program sections for auxiliary function processing to be skipped if there is no change.

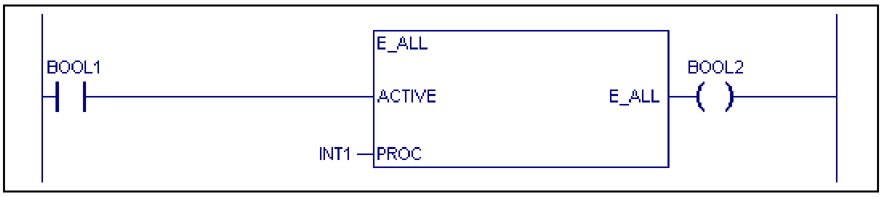


Fig. 2-23: Interrogate all E functions

ACTIVE:	BOOL	0	E function change interrogation is not active.
		1	E function change interrogation is active
PROC:	INT		Process number [0..6]

Function result

E_ALL:	BOOL	0	The standard function is not active, or the standard function is active and there is no change.
		1	The standard function is active and there is a change.

The "E_ALL_Q" standard function permits all pending E functions to be acknowledged.

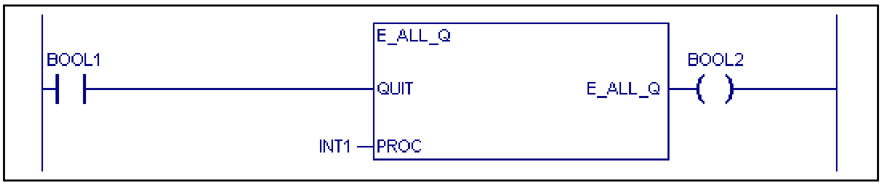


Fig. 2-24: Acknowledge all E functions

QUIT:	BOOL	0	No acknowledgment of the E function
		1	The E function is acknowledged
PROC:	INT		Process number [0..6]

Function result E_ALL_Q : BOOL The QUIT function input is passed on

The "E_NR" standard function permits the E function number that is transferred from the NC to be read. Reading the E function number permits the state of the E function as it exists in the NC (and appears on the status display) to be retrieved for each spindle.

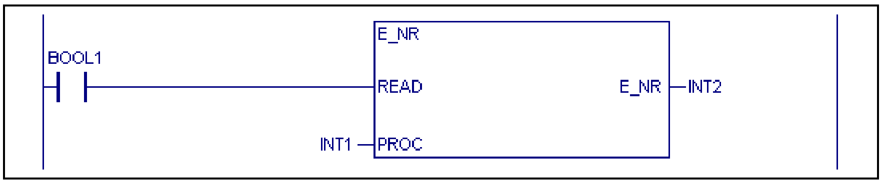


Fig. 2-25: Interrogating the E function number

READ:	BOOL	0	Reading the E function number not active
		1	Reading the E function number is active
PROC:	INT		Process number [0..6]

Function result

E_NR	: INT	-1	An E function has not yet been output since the time the controller was switched on.
		0..9	The last E function number to have been output. It remains pending after acknowledgment until the next E function is output.

Error Handling

Programming errors may prevent the auxiliary functions from being executed correctly. In this case, "error handling" specifies the cause of the error.

Error type of the function blocks					
Interrogation/acknowledgment of the auxiliary functions with specification of the auxiliary function number		Interrogation/acknowledgment of the auxiliary functions without specification of the auxiliary function number		Reading the auxiliary function number	
M_FKT:	-1	M_ALL:	-117	M_NR:	-126
M_FKT_Q:	-2	M_ALL_Q:	-118	S_NR:	-127

S_FKT:	-3	S_ALL:	-119	T_NR:	-180
S_FKT_Q:	-4	S_ALL_Q:	-120	Q_NR:	-128
T_FKT:	-5	T_ALL:	-121	E_NR:	-185
T_FKT_Q:	-6	T_ALL_Q:	-122		
Q_FKT:	-7	Q_ALL:	-123		
Q_FKT_Q:	-8	Q_ALL_Q:	-124		
E_FKT:	-181	E_ALL:	-183		
E_FKT_Q:	-182	E_ALL_Q:	-184		

Fig. 2-26: Error types of the auxiliary functions

Error numbers:

1 Invalid input parameter

- The value of the PROC, NR, GROUP, SPINDLE or EDGE input is negative.
- The PROC input is greater than 6.
- The NR input is greater than 999.
- The GROUP input is greater than 16.
- The SPINDLE input is greater than 3.
- The EDGE input is greater than 9.

2.2 Events

High-speed event processing

In the default structure, the events are exchanged with the PLC user program via the process image. From version 18VRS onwards, exchanging events can bypass the process image.

32 event signals are available for this purpose. To distinguish high-speed events from the process events proper (process no.: 0 - 6), high-speed events are addressed via "process number 7". This means that an event that is allocated to the processes 0 through 6 cannot be used as "high-speed events".

Interrupt processing within the NC, for events 0 ... 7, is not supported in the case of the "high-speed events".

The previous PLC functions that are used for exchanging events with the NC program are retained:

- EVENT: Reading an event
- EV_ST: Assigning an event
- EV_SET: Setting an event
- EV_RES: Resetting an event

Note: The functions that are used for exchanging events can also be programmed within the 2-ms implementation of the PLC.

It must be noted that a cycle time of 200 µs is available for executing the 2-ms implementation. It is consequently recommended to process <u>not more than 5 event functions</u> within this program sequence.

The Bosch Rexroth standard functions:

- EVENT: Reading an event
- EV_ST Writing an event (assignment)
- EV_SET Writing an event (setting)
- EV_RES Writing an event (resetting)

enable NC events to be read or written.

An NC event is a binary piece of information that is exchanged between the processes in the NC or between the NC and the PLC. The event functions enable the PLC to access the events of all processes.

The function interfaces are exactly defined. When a function is invoked, the programmer merely "connects" the individual signals.

Reading an event "EVENT"

The "EVENT" function is used for reading the state of an event in the NC.

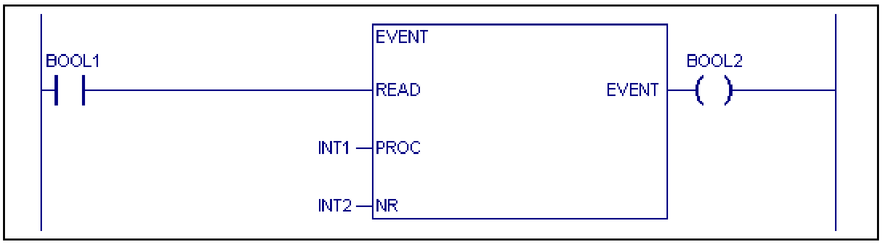


Fig. 2-27: Read event

READ:	BOOL	0 Function not active 1 Read event
PROC:	INT	Process number [0..6]
NR:	INT	Event number (0...31)

Function output

EVENT:	BOOL	0 Function is not active or event is reset 1 Event is set

The "READ" function input activates the "EVENT" function. The "PROC" and "NR" inputs address the NC event that is to be read. The state of the event is provided as function result.

Writing of an event: Saving "EV_ST"

The "EV_ST" function maps the state at the "WRITE" function input onto the event of the NC.

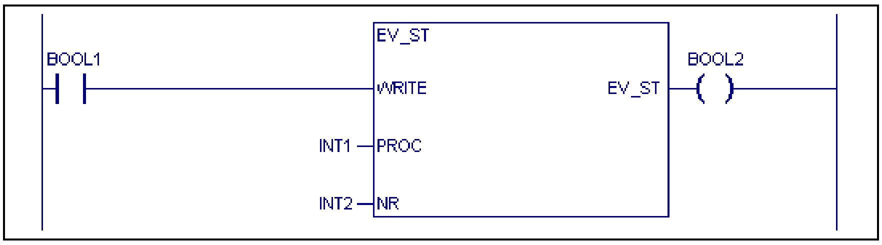


Fig. 2-28: Set event

WRITE:	BOOL	0 Reset event 1 Set event
PROC:	INT	Process number [0..6]
NR:	INT	Event number (0...31)

Function output

EV_ST:	BOOL	Status of the WRITE input

The "WRITE" function input directly influences the event of the NC. The event is reset when the "WRITE" function input is reset. The event of the NC is set when the "WRITE" function input is set. The "PROC" and "NR" inputs address the NC event that is to be influenced. The "WRITE" function input is provided as the function result.

Writing of an event: Conditional setting "EV_SET"

The "EV_SET" function sets the event of the NC according to the status of the "WRITE" function input.

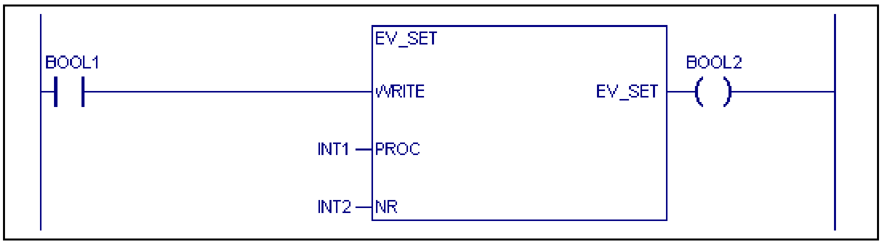


Fig. 2-29: Set event

WRITE:	BOOL	0 Function not active
		1 Set event
PROC:	INT	Process number [0..6]
NR:	INT	Event number (0...31)

Function output

EV_SET: BOOL Status of the WRITE input

The event is not influenced when the "WRITE" function input is reset. The event of the NC is set when the "WRITE" function input is set. The "PROC" and "NR" inputs address the NC event that is to be influenced. The "WRITE" function input is provided as the function result.

Writing of an event: Conditional resetting "EV_RES"

The "EV_RES" function resets the event of the NC according to the status of the "WRITE" function input.

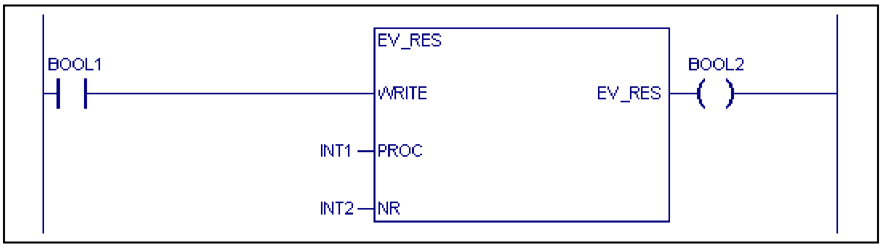


Fig. 2-30: Reset event

WRITE:	BOOL	0 Function not active
		1 Reset event
PROC:	INT	Process number [0..6]
NR:	INT	Event number (0...31)

Function output

EV_RES : BOOL Status of the "WRITE" input

The event is not influenced when the "WRITE" function input is reset. The event of the NC is reset when the "WRITE" function input is set. The "PROC" and "NR" inputs address the NC event that is to be influenced. The "WRITE" function input is provided as the function result.

Error Handling

Programming errors may prevent the "EVENT", "EV_ST", "EV_SET" and "EV_RES" functions from being executed correctly. In such a case, error handling reports the cause of the error.

Error type for the functions	
Reading of events	Writing of events
EVENT: -9	EV_ST: -10 EV_SET: -11 EV_RES: -12

Fig. 2-31: Error types of the event functions

Error numbers:

1 - Invalid input parameter

- The value of the "PROC" or "NR" input is negative.
- The PROC input is greater than 6.
- The PROC input is greater than 31.

Examples of event programming

Switching function output without immediate acknowledgment:

A switching function is output during the execution sequence that shall be processed simultaneously to the NC movements.

The switching function is output by setting an event in the NC program. The event "10" is cyclically interrogated in the PLC program. The output "QCLAMP" is set or the output "QUNCLAMP" is reset in the PLC program when the event is activated by the NC program. Now, the NC program executes in parallel to the switching function output. This means that NC movements can be performed while the parts are being clamped. The "M10" auxiliary function checks the correct execution of the switching function output. The NC program waits at this block until the acknowledgment via the inputs "ICLAMP" or "IUNCLAMP" is available.

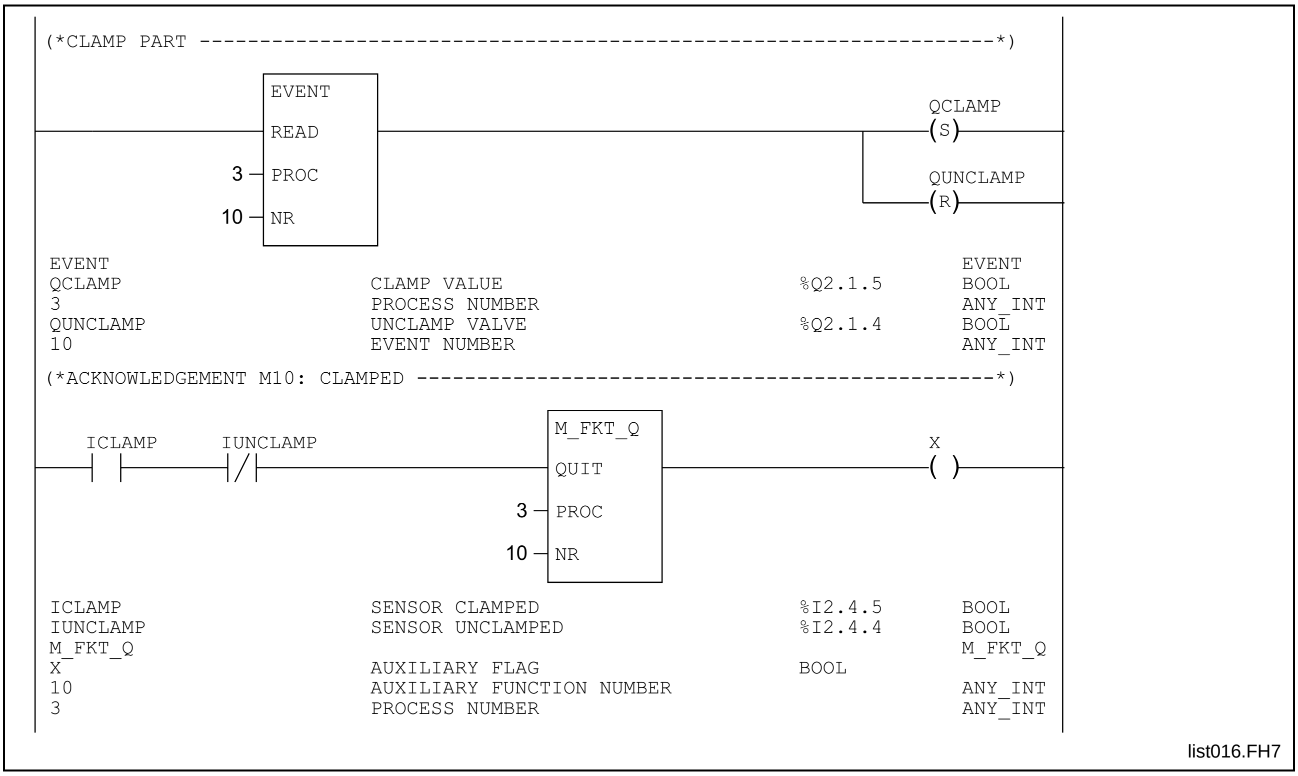


Fig. 2-32: Switching function output without immediate acknowledgment

Controlling NC program execution through events:

Events are generated in the PLC program that lead to different program sections when the NC program is executed.

The current motor type exists as "M_TYP" flag in the PLC program. Comparison with the motor types "M_TYP_A" and "M_TYP_B" influences the events "5" and "6". Event "5" is set if the current motor is of type "M_TYP_A". Event "6" is set if the current motor is of type "M_TYP_B". Depending on these events, the NC program branches to specific program sections (e.g. "BES" - jump if event is set).

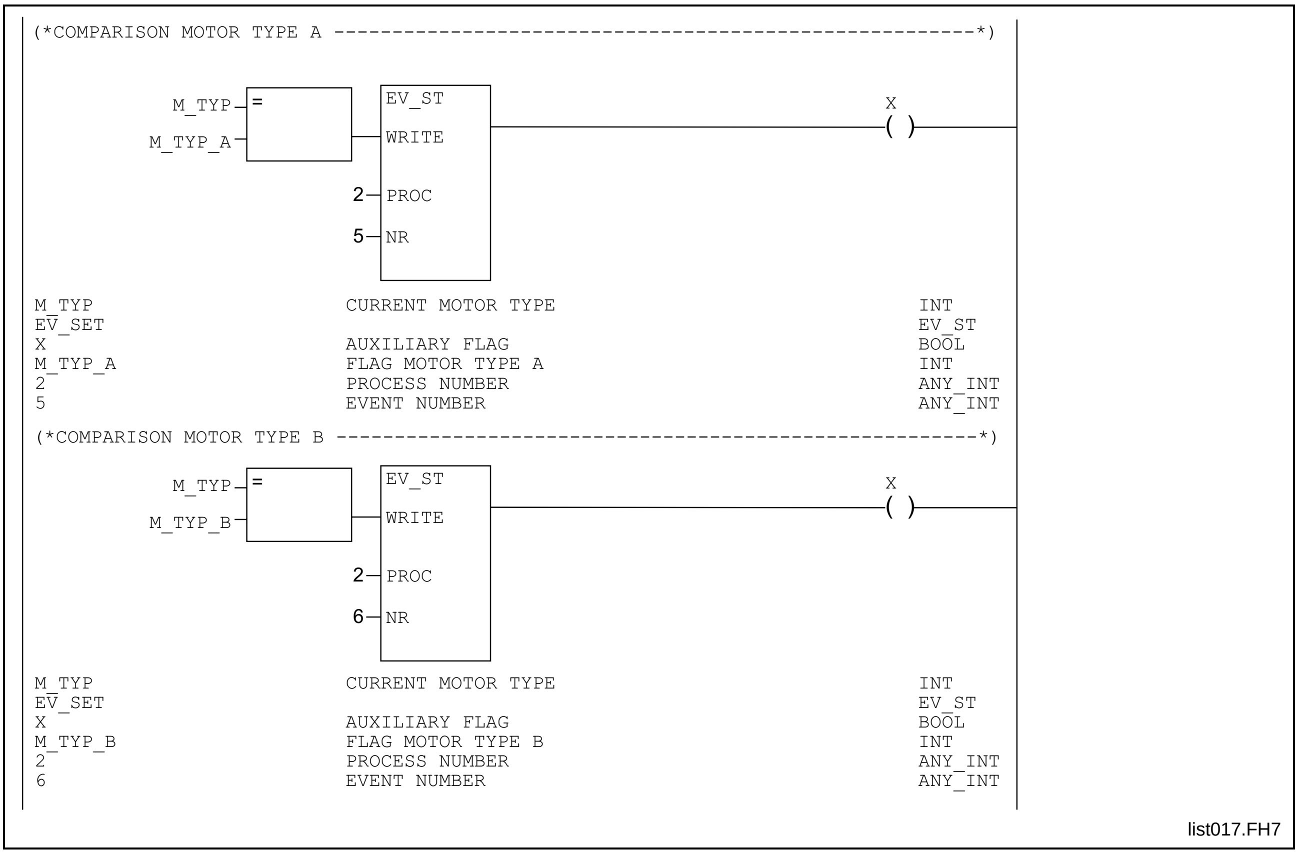


Fig. 2-33: Controlling NC program execution through events

2.3 Diagnosis functions

Diagnosis commands are implemented via Bosch Rexroth standard functions. Corresponding with the message number, the message is generated as a status message or as an error message.

- Status message: Message numbers 151 ... 200
- Error message: Message numbers 201 ... 250
  Message numbers 551 ... 600

Possible are:

- Diagnosis output; directly specified message number MSG_WR
- Reading the NC message number MSG_RD
- Message output with additional information as a number MSG_WR_N
- Message output with additional information as an axis designation MSG_WR_A

Write diagnosis

Message output with direct specification of the message number:

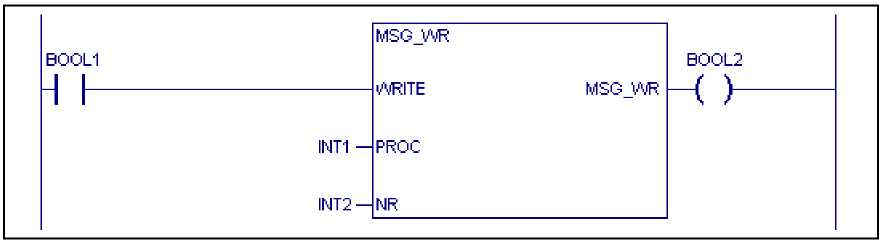


Fig. 2-34: Write message

WRITE: Enable message output
PROC: Process number
NR: Message number (constant)

Function result

The ACTIVE input is passed on.
Function result = ACTIVE

The message number "NR" for process "PROC" is generated by setting the enabling input "WRITE" of "MSG_WR" function. A diagnosis message is not generated if "WRITE" is cleared. The output contains a copy of the input value.

Message output with additional information as a number:

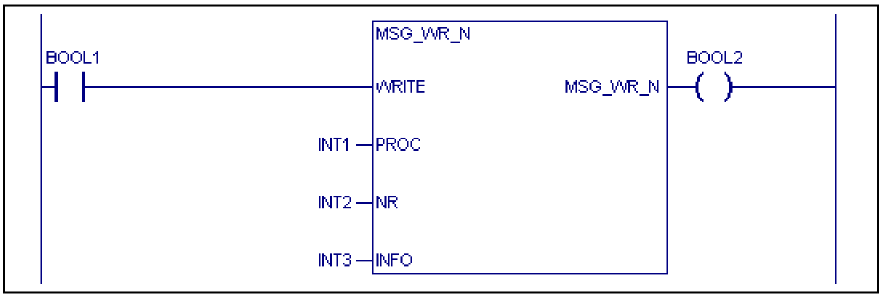


Fig. 2-35: Write message with additional number

WRITE: 0 Function not active
1 Message is written
PROC: Process number [0... 6)
NR: Message number (151 ... 250, 551 ... 600)
INFO: Reference number (0 ... 1023)

Function result

The ACTIVE input is passed on.
Function result = ACTIVE

Example

Editing the message text “Station @ not in automatic mode”

In the diagnosis overview, "Station 2 not in automatic mode" will appear for station 2. This means that @ is replaced with the transferred number.

Message output with additional information as an axis designation:

A message number and a further information can be transferred via standard function "MSG_WR_A" to the NC as state / error message. The number of the reference information is converted into an axis designation (X, Y, Z, ...). The "@" character is replaced with the axis designation when the message is displayed.

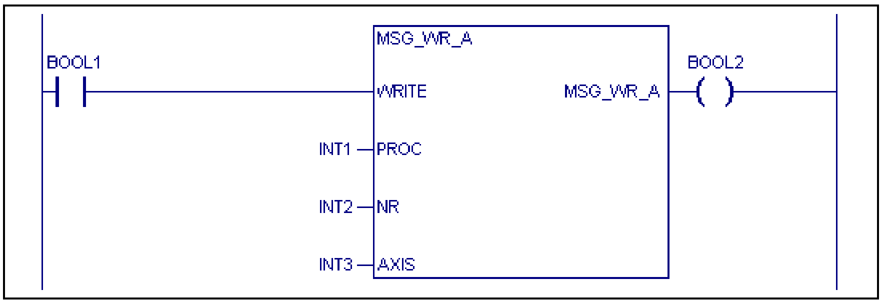


Fig. 2-36: Write message with axis information

WRITE: 0 Function not active
1 Message is written
PROC: Process number [0... 6)
NR: Message number (151 ... 250, 551 ... 600)
AXIS: Axis number (1 ... 32)

Function result

The ACTIVE input is passed on.
Function result = ACTIVE

Example

Editing the message text “Axis @ on positive limit switch.”

In the diagnosis overview, "Axis X on pos. limit switch" will appear for station 2. This means that the transferred number is replaced with the axis designation.

Read diagnosis

Reading a message number:

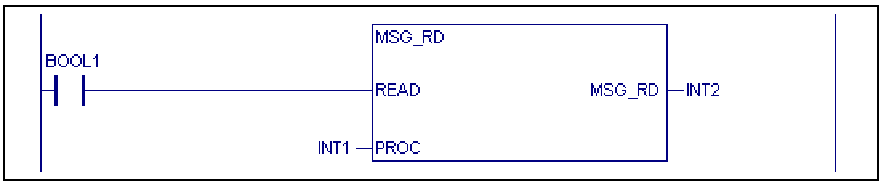


Fig. 2-37: Read message

READ : Enable reading a message number
PROC: Process number

By setting the enable input "READ" of function "MSG_RD", the message number for the process "PROC" is read into the accu. The diagnosis is not read if the "READ" enable input is cleared.

Thus, only the current NC messages are read.

2.4 Magazine functions

With NC-controlled tool magazines, moving the tool magazine does not require any programming in the PLC. Merely the signals within the PLC-NC interface must be connected (e.g. jogging the tool magazine, enabling).

With a PLC-controlled tool magazine, homing and moving to a new location must both be programmed.

The tool magazine interface between PLC and NC must always be connected, irrespective of the axis or tool magazine type.

Tool Storage Unit Motion Commands of the PLC

The PLC program permits different tool magazines and tool changers to be adapted to the Rexroth MTC 200. Handling these modules is supported by various Bosch Rexroth standard functions.

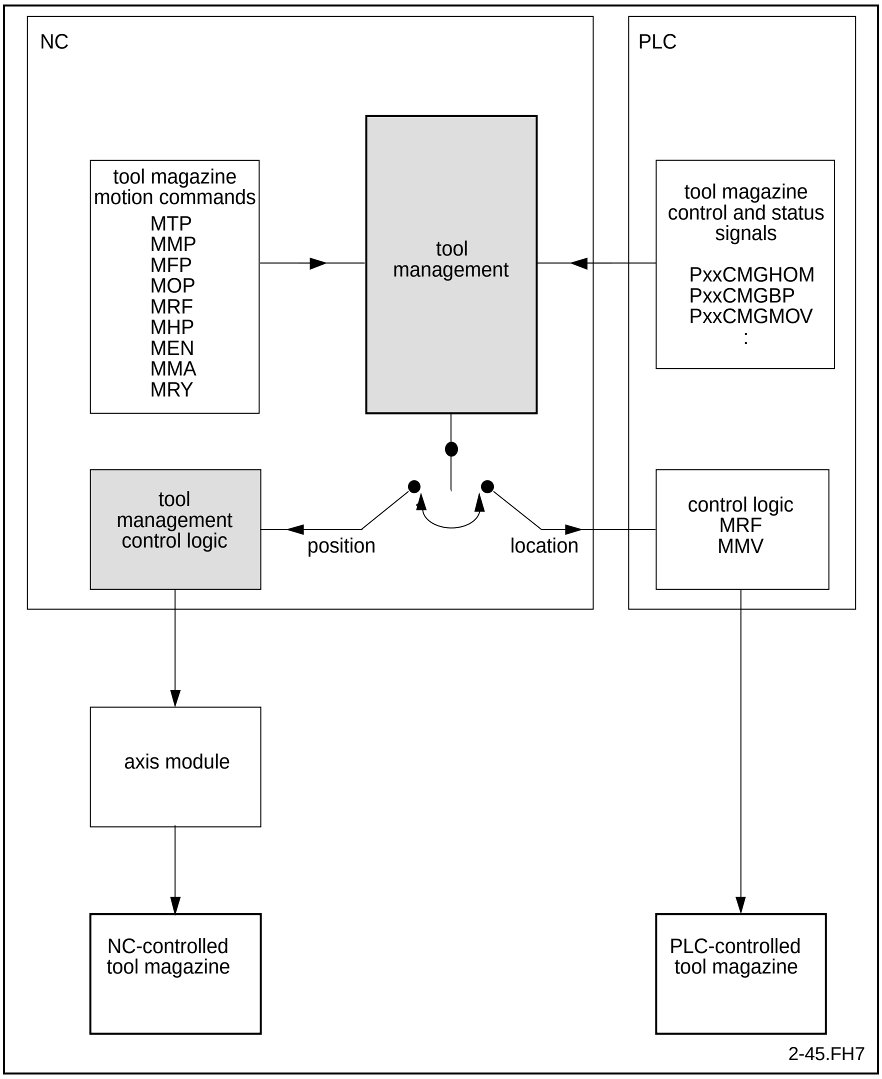


Fig. 2-38: General method of operation of the motion control of a tool magazine

The NC instruction set provides commands for changing tools and moving the tool magazine.

Either the NC itself (NC axis) or the PLC controls the movements of a tool magazine.

If the tool storage unit is moved via the PLC, the "MRF" and "MMV" standard functions in the PLC interpret the type of the movement.

The "MRF" standard function initiates tool magazine homing. The "MRF_Q" standard function notifies the NC of the acknowledgment of successful tool magazine homing.

Homing the tool storage unit by the PLC:

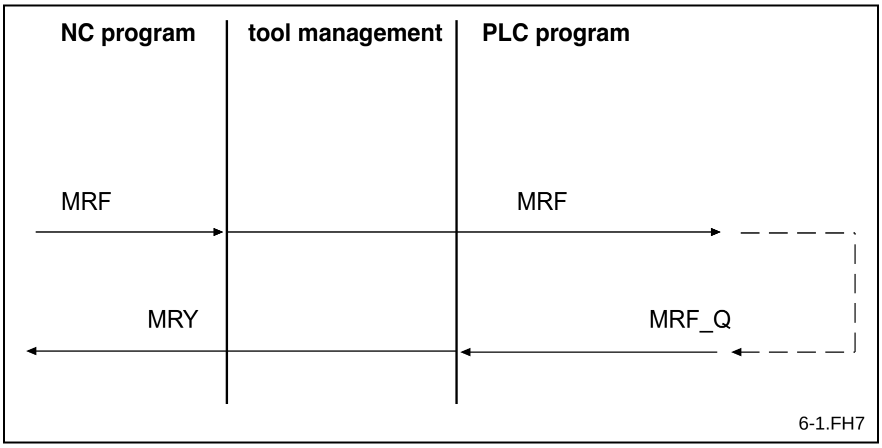


Fig. 2-39: Homing of tool magazine

If a tool magazine is to move to a new location, this is interpreted by the "MMV" standard function. The command position of the corresponding tool is transferred in the NC-PLC interface.

The PLC employs the "MMV_Q" standard function for reporting the acknowledgment of successful positioning.

Positioning the tool magazine by the PLC:

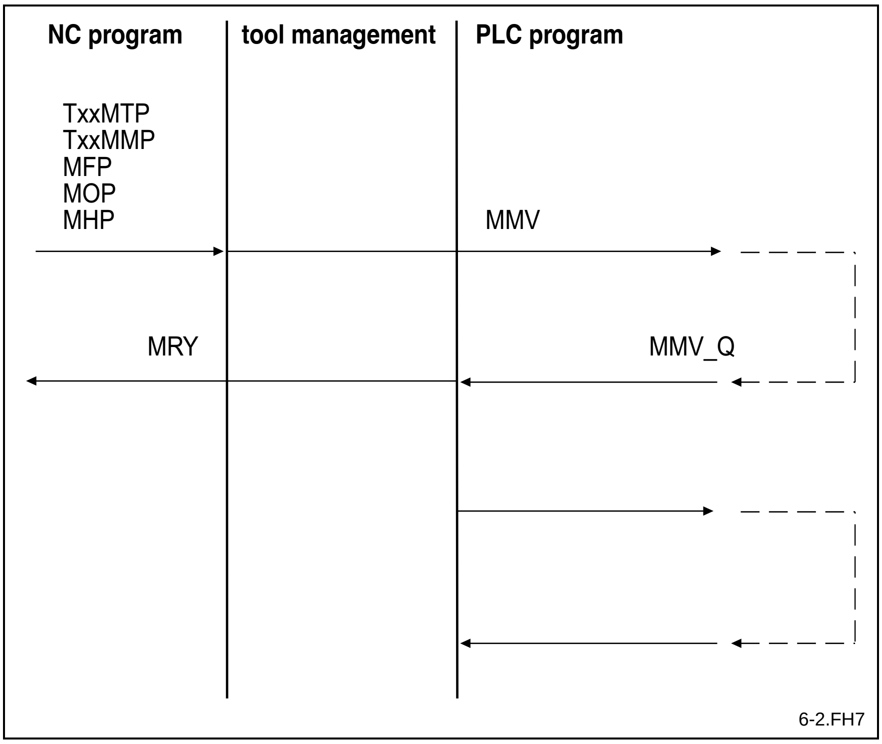


Fig. 2-40: Positioning of tool magazine

Depending on the mechanical structure, changing tools is programmed in different ways. Special tool changing standard functions exist for this purpose.

The "TCH" standard function is used for interrogating the complete tool changing process. Tools are changed between the tool magazine, gripper and spindle without any further action from the NC. The PLC program controls the entire tool changing procedure. The PLC employs the "TCH_Q" standard function for acknowledging the tool changing process.

If NC functions (such as axis movements) are necessary within the tool changing process, the tool changing process is split up into smaller steps.

Changing tools is split up into the sub-steps of changing between tool magazine and spindle and vice versa. The two tool changing directions are sensed by the "TMS" and "TSM" standard functions. The "TMS_Q" and "TSM_Q" functions are used for acknowledging the execution of the tool changing process.

Magazine homing

Interrogation homing "MRF"

If homing a tool magazine is introduced in NC program via "MRF" NC command, this can be evaluated in PLC program via standard function "MRF".

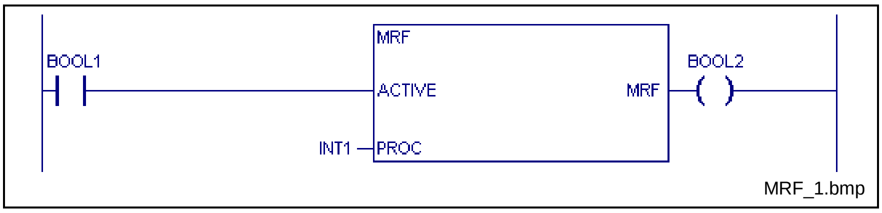


Fig. 2-41: Homing of magazine

ACTIVE: 0 Function not active
1 - Interrogation of NC command "Tool magazine homing"
PROC: Process number (0-6)

Function result
0 No NC command "Tool magazine homing" or function is not active
1 NC command "Tool magazine homing" is active

Acknowledging magazine homing "MRF_Q"

Standard function "MRF_Q" acknowledges homing of a tool magazine.

The "MRY" command can be used in the NC program for interrogating the execution of the tool magazine command.

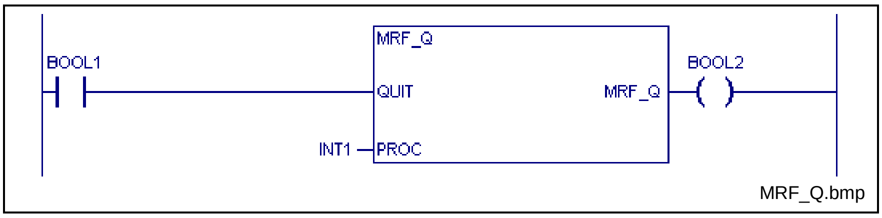


Fig. 2-42: Acknowledging homing of magazine

QUIT: 0 Function not active
1 Acknowledgment of the NC command "Tool magazine homing"
PROC: Process number (0-6)

Function result
0 NC command "Tool magazine homing" acknowledged or function is not active
1 "NC command "Tool magazine homing" is acknowledged

"Tool Magazine Homing" Sequence between NC and PLC

```
NC Program                          PLC program
Process 1                           Process1

                 ( 1 )       |          (2)                    (3)          |
   N010          MRF         |      +---------+                             |
                             |      |MRF      |                             |
                             +------|ACTIVE   |              QMAGPOS        |
                             |   1 -|PROC     +---------------(   ) --------|
                             |      +---------+                             |
                             |          (4)                                 |
                             |      +---------+                             |
                             | IREF |MRF_Q    |                             |
                 ( 5 )       +--| +-|QUIT     |                 X           |
   N011          MRY         |   1 -|PROC     +---------------(   ) --------|
                             |      +---------+                             |
```

Timing of tool magazine homing:

```
                      (1)
NC command            +-----------------------------------------------------┐
MRF               ----+                                                     └--------

                          (2)
PLC function              +-------------------------------------------------┐
MRF               --------+                                                 └--------

                                                                       (5)
PLC function                                                           +-----┐
MRF_Q             -----------------------------------------------------+     └-----

                              (3)
Output                        ┌-----------------------------------------┐
QMAGPOS           ------------+                                         └----------

                                                                       (4)
Input                                                                  ┌-----------
IREF              -----------------------------------------------------+

Actual tool
magazine
location        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx1111111111
```

1. The "tool magazine homing" (MRF) command is processed in the NC program and output to the PLC.
2. The "MRF" standard function recognizes the NC homing command in the PLC program.
3. Tool magazine homing is initiated in the PLC program by setting the output.
4. The "Homing switch" input shows the homing position of the tool magazine. The movement is stopped and the NC homing command is acknowledged.
5. The "MRY" command in the NC program is used for interrogating the completion of tool storage unit homing.

Sample program: PLC-controlle, endless-rotating tool magazine

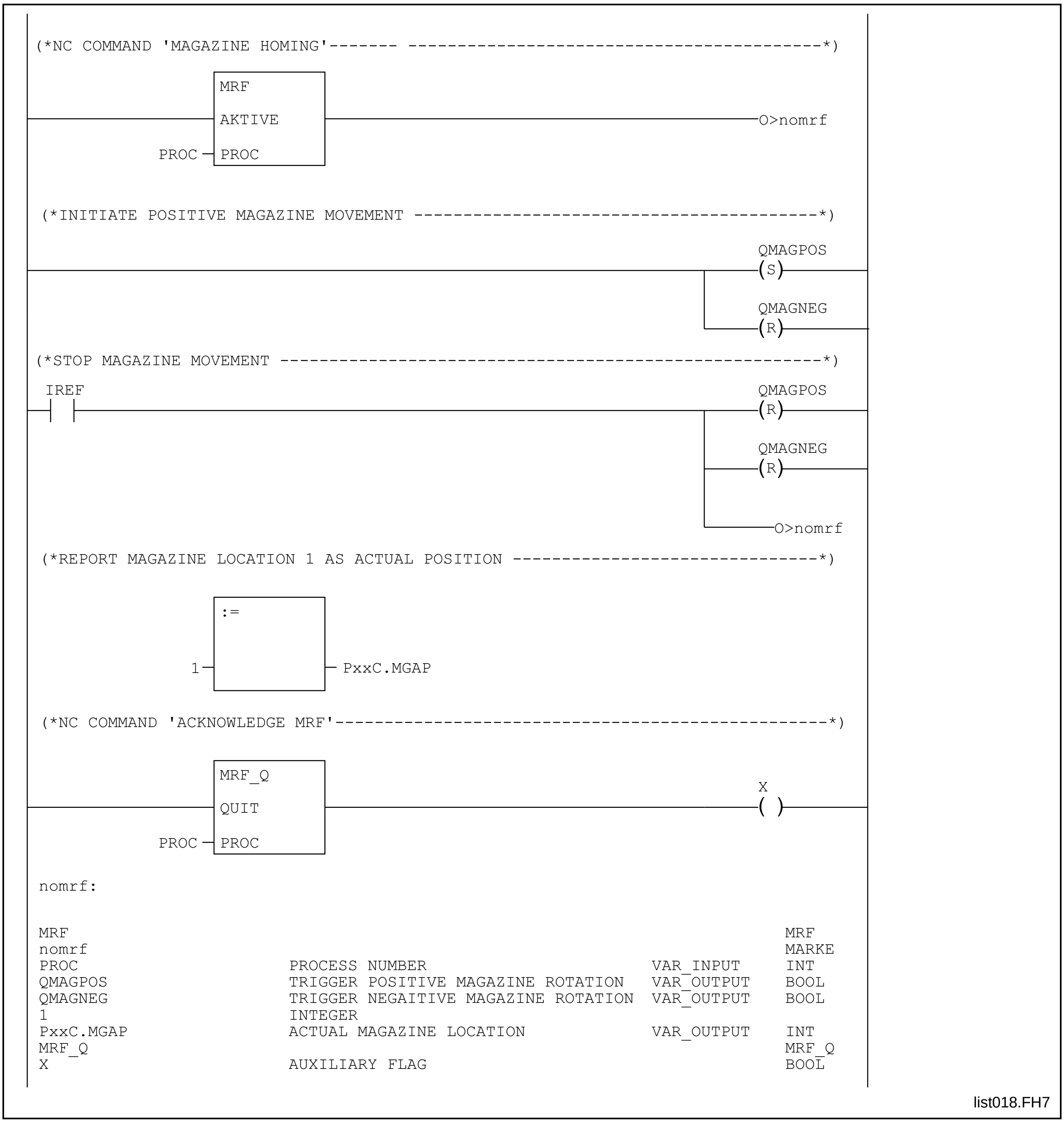


Fig. 2-43: PLC-controlled, endlessly rotating tool magazine

Tool Magazine Positioning

The tool magazine must be moved to be able to utilize different tools in one working cycle and to move the programmed tool to the required location (tool change, loading or machining location).

Interrogating magazine to a new location "MMV"

If the NC program executes an NC command that causes the tool magazine to move (e.g. "MTP", "MFP", etc.), this can be interpreted in the PLC program by the "MMV" standard function.

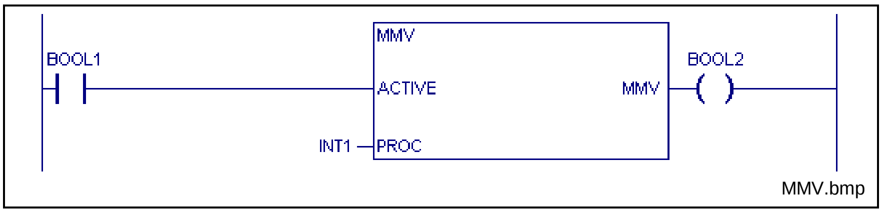


Fig. 2-44: Evaluate magazine movement

ACTIVE: 0 Function not active
1 Interrogation of NC command "Rotate tool magazine"
PROC: Process number (0-6)

Function result

0 No NC command "Rotate tool magazine" or function is not active
1 NC command "Rotate tool magazine" is active

Acknowledging magazine to a new location "MMV_Q"

Turning a toom magazine is acknowledged via standard function "MMV_Q". The NC command that initiates the movement was executed by the PLC program. The "MRY" command in the NC program can be used for interrogating the execution of the command.

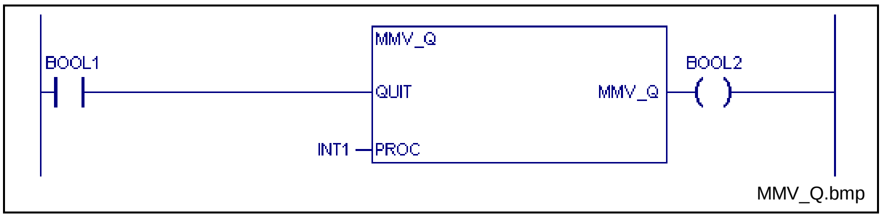


Fig. 2-45: Acknowledging movement of magazine

QUIT: 0 Function not active
1 Acknowledgment of the NC command "Rotate tool magazine"
PROC: Process number (0-6)

Function result

0 Acknowledgment of the "Rotate tool magazine" NC command or function is not active
1 NC command is acknowledged

Sequence of Tool Magazine Positioning between NC and PLC

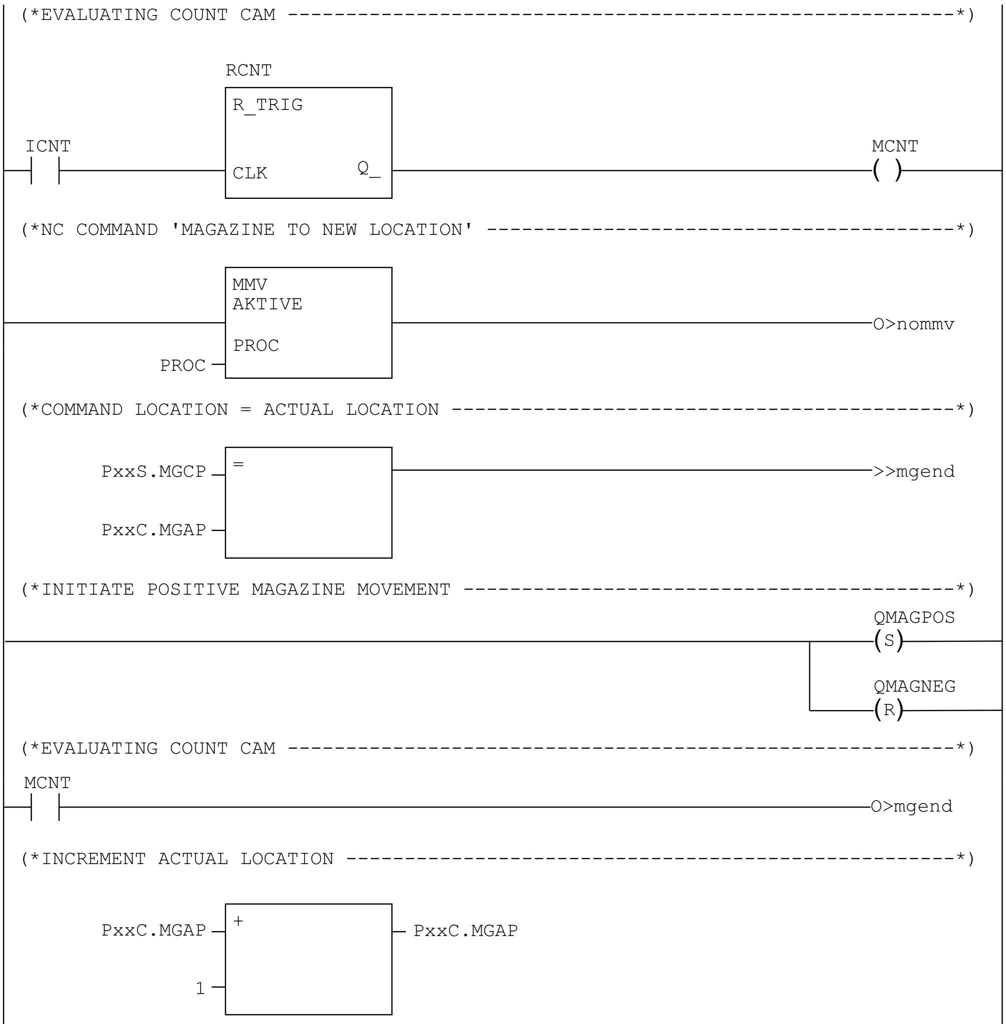
```
NC                        program      PLC                                              program
Process 1                              Process1

               ( 1 )             |            (2)                   (3)          |
  N010      T3  MTP              |         +---------+                           |
                                 |         |MMV      |                           |
                                 +---------|ACTIVE   |            QMAGPOS        |
                                 |     1 --|PROC     +-------------(    ) --------|
                                 |         +---------+                           |
                                 |   Input             Evaluation                |
                                 +---|  +---------------------        (4)        |
                                 |                     Count impulse              |
                                 |  (5)    +---------+           (6)              |
                                 |Comm.loc.-|  =      |      +---------+          |
                                 |          |         |      |MMF_Q    |          |
                                 | Act.loc.-|         +----|QUIT      +-          |
                                 |          +---------+  1-|PROC      |          |
               ( 7 )             |                         +---------+
  N011          MRY              |
```

Timing of tool magazine positioning:

```
                   (1)
NC command         +-----------------------------------------------------┐
T3 MTP         ----+                                                     └--------
                       (2)
PLC function           +-------------------------------------------------┐
MMV            --------+                                                 └--------
                                                                     (7)
PLC function                                                         +-----┐
MMV_Q          ------------------------------------------------------+     └-----
                           (3)
Output                     ┌------------------------------------------┐
QMAGPOS        ------------+                                          └----------
                                                                     (7)
Eingang        ---------------┐                  ┌--┐                ┌-----------
ICNT                          └------------------┘  └----------------+
Actual tool
magazine
location      xxxxx33333333333333333333333333333333333333333333333333333333333333
Actual tool
magazine
location     11111111111111111111111111111222222222222222222222222222333333333333
```

1. The "T3 MTP" command is processed in the NC program and output to the PLC.
2. The "MMV" standard function is used for recognizing the tool magazine rotation command in the PLC program.
3. Tool magazine rotation is initiated in the PLC program by setting the output.
4. A "Count cam" input sensing logic generates the current actual tool magazine location.
5. The programmed tool magazine location is reached when the command location and the actual location of the tool magazine are the same.
6. The tool magazine movement is stopped and the NC tool magazine command is acknowledged.
7. The "MRY" command in the NC program is used for sensing the completion of the movement of the tool magazine to a new command location.

Program by way of example: PLC-controlled, endlessly rotating tool magazine



R_TRIG			R_TRIG
ICNT	COUNT CAM	VAR_INPUT	BOOL
MCNT	COUNT IMPULSE FLAG	VAR	BOOL
MMV			MMV
nommv			LABEL
PROC	PROCESS NUMBER	VAR_INPUT	INT
PxxS.MGCP	COMMAND MAGAZINE POSITION	VAR_INPUT	INT
mgend			LABEL
PxxC.MGAP	ACTUAL MAGAZINE POSITION	VAR_OUTPUT	INT
QMAGPOS	TRIGGER POSITIVE MAGAZINE ROTATION	VAR_OUTPUT	BOOL
QMAGNEG	TRIGGER NEGATIVE MAGAZINE ROTATION	VAR_OUTPUT	BOOL
1	INTEGER		

list019.FH7

Fig. 2-46: PLC-controlled, endlessly rotating tool magazine

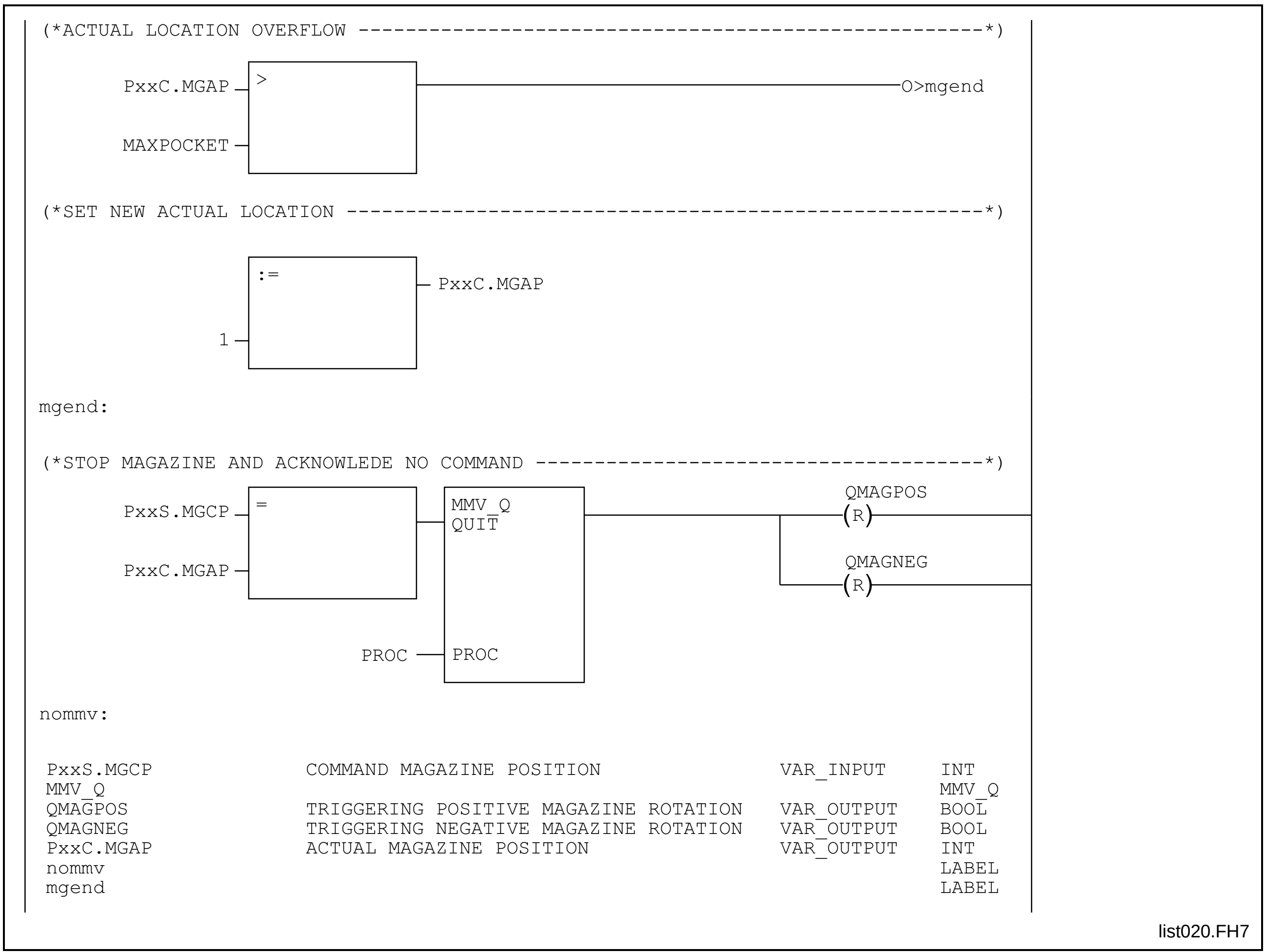


Fig. 2-47: PLC-controlled, endlessly rotating tool magazine (continued)

2.5 Tool Change Functions

Tool change functions are required only if a magazine is used for tool storage and if tools must be exchanged (via grippers) between the spindle and the magazine during machining.

Tool change functions are not permitted in conjunction with turrets or in systems that employ only one spindle.

Due to the different mechanical structures of the tool changing systems, the tool changing process proper must always be implemented in the PLC. To support programming, the Rexroth MTC 200 provides special tool change commands.

A tool change process is always initiated in the NC program, executed in the PLC, and acknowledged before NC block processing is continued.

The PLC employs the "TCH" command to interrogate a complete tool change, and performs it between magazine, gripper(s) and spindle without any further NC action.

The tool change sequence must be split into smaller steps if changing tools requires NC functions (such as axis movements) to be performed.

The "TMS" and "TSM" commands enable the PLC to sense the partial change processes "magazine-to-spindle tool change" and "spindle-to-magazine tool change".

The PLC uses the "TRY" acknowledgment signal to notify the NC of the successful execution of a complete or partial change process. The NC then continues to the next block.

The following PLC commands can be used for any process.

Complete tool change

The "TCH" command is used in the NC program for initiating a general tool change. The entire change process is performed when this command is programmed. The next command in the NC program will be executed only after the tool change has been performed and acknowledged.

In the PLC program, the change process is split into partial processes. The tool transfer commands are used for notifying the Rexroth MTC 200 of the current tool location. The "TCH" command is only acknowledged after the entire change process has been completed.

Complete tool change 'TCH'

The "TCH" standard function in the PLC program permits a complete tool change "TCH" to be interrogated that is active in the NC program.

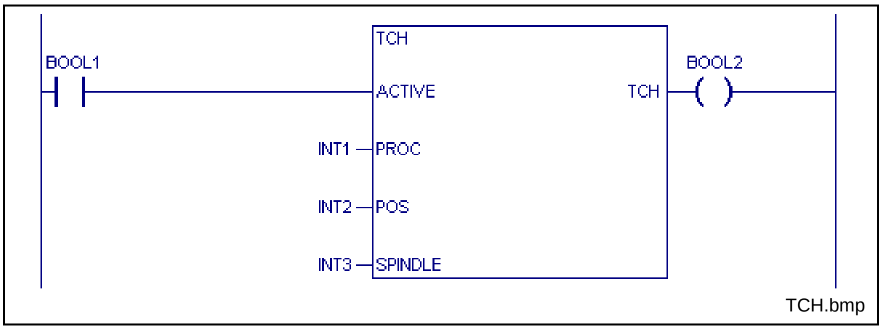


Fig. 2-48: Interrogate tool change

ACTIVE: 0 Function not active
1 Interrogation of "Complete tool change"
PROC: Process number (0-6)
POS: Position (1-4)
SPINDLE: Spindle number (1-3)

Function result

0 No "Complete tool change" or function is not active
1 NC command "Complete tool change" is active

Acknowledging complete tool change "TCH_Q"

The "TCH_Q" standard function in the PLC program is used for acknowledging the execution of the complete tool change. This completes tool change processing in the PLC. The NC program is continued when the tool change is acknowledged.

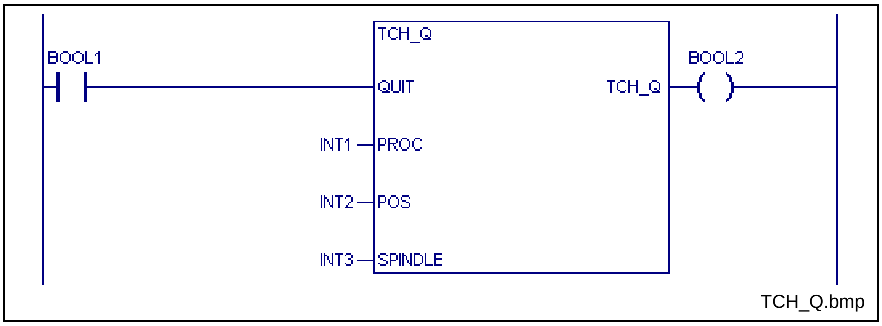


Fig. 2-49: Acknowledging a tool change

QUIT: 0 Function not active
1 Acknowledgment "Complete tool change"
PROC: Process number (0-6)
POS: Position (1-4)
SPINDLE: Spindle number (1-3)

Function result

0 NC command "Tool change from magazine to spindle" acknowledged or function is not active
1 NC command "Tool change from magazine to spindle" is acknowledged

Complete Tool Change Sequence

```
NC                      program     PLC                                              program
Process 1                           Process1

                        ( 1 )       |                                               |
  N010      TCH                     |                   +---------+                 |
                                    |                   |TCH      |     (2)         |
                                    +-------------------|ACTIVE   +-------+         |
                                    |              1 -- |PROC     |       |         |
                                    |                   +---------+       |         |
                                    |              +----------------------+         |
                                    |              |    Initiates complete          |  (3)
                                    |              +--------- tool change           |
                                    |                                               |
                                    |  +------------------------------------+       |
                                    |  |  Physical and logical              |       |  (4)
                                    |  |  tool change                       |       |
                                    |  +------------------------------------+       |
                                    |                                               |
                                    |     Complete tool    ---------------+         |  (5)
                                    |     change executed                 |         |
                                    |           +-------------------------+         |
                                    |           |       +---------+                 |
                                    |           |       |TCH_Q    |                 |  (6)
                                    |           +-------|QUIT     +--               |
                                    |              1 -- |PROC     |                 |
                                    |                   +---------+                 |
                        ( 7 )
  N011 ...
       ...
```

Timing of complete tool change:

```
                    (1)
NC command          +---------------------------------------------+
TCH             ----+                                             +--------

                        (2)
PLC function            +-----------------------------------------+
TCH             --------+                                         +--------

                                                            (6)   (7)
PLC function                                                +-----+
TCH_Q           --------------------------------------------+     +-----

                        (3)                  (4)
Tool change             +-----------------------------------+
proc. running   --------+                                   +----------

                                                            (5
Ready message                                               +-----------
t. ch. compl.   --------------------------------------------+
```

1. In the NC program, the complete tool change is initiated by the "TCH" command, and output to the PLC. The NC program now waits until the PLC acknowledges the tool change.
2. The "TCH" standard function permits the complete NC tool change to be recognized in the PLC program.
3. The tool change sequence is initiated in the PLC program.

4. Usually, the sequence of a complete tool change consists of several actions (e.g. open/close gripper, clamp/unclamp clamping device in spindle) and of tool transfer messages to tool management (see tool transfer functions).
5. The complete tool change has been performed. This can be checked in the PLC program using feedback messages (end positions).
6. The "TCH_Q" standard function is used for acknowledging the NC tool change command.
7. The NC program recognizes the completion of the complete tool change. The NC program is continued.

Magazine-to-spindle tool change

Besides the complete tool change command "TCH", the NC instruction set contains special tool change commands. Changing tools is performed in smaller steps between magazine, gripper and spindle, or directly between magazine and spindle.

On the NC side, the entire tool change process can be split into individual steps (such as "clamp/unclamp tool", magazine movements to approach a new tool, or axis and spindle movements in the magazine).

PLC standard functions for tool transfer messages are used for notifying tool management of the tool location. A further standard function is used in the PLC program for acknowledging the completion of the tool change process.

Request tool change of magazine to spindle 'TMS'

The "TMS" standard function can be used in the PLC program for interrogating a "TMS" magazine-to-spindle tool change that is active in the NC program.

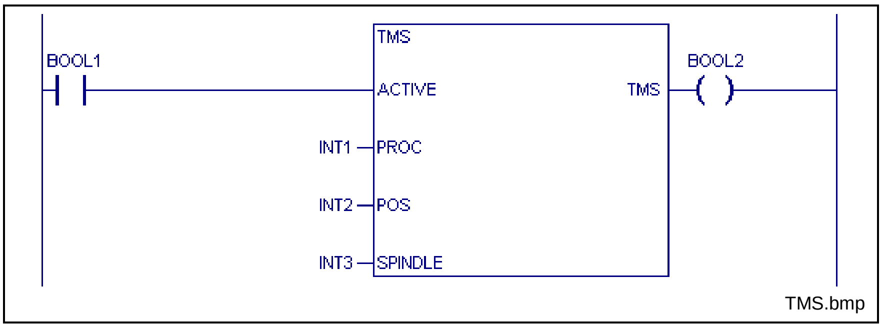


Fig. 2-50: Evaluating a magazine-to-spindle tool change

ACTIVE: 0 Function not active
1 Interrogation "Magazine-to-spindle tool change"
PROC: Process number (0-6)
POS: Position (1-4)
SPINDLE: Spindle number (1-3)

Function result

0 No "Magazine-to-spindle tool change" or function is not active
1 NC command "Magazine-to-spindle tool change" is active.

Acknowledgment of magazine-to-spindle tool change 'TMS_Q'

The "TMS_Q" standard function is used in the PLC program for acknowledging the completion of the magazine-to-spindle tool change. This completes tool change processing in the PLC. The NC program is continued when the tool change is acknowledged.

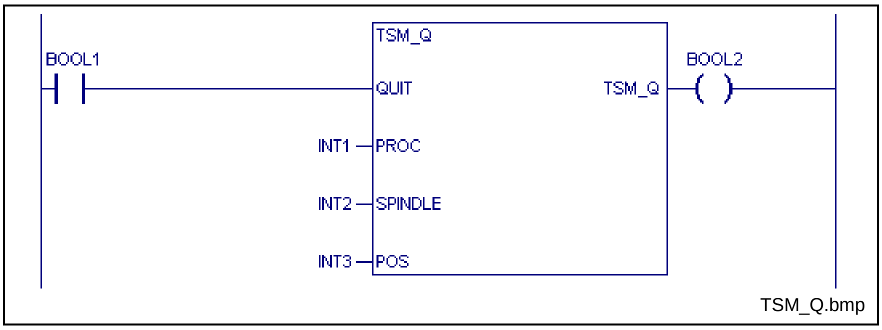


Fig. 2-51: Acknowledging a magazine-to-spindle tool change

QUIT: 0 Function not active
1 Acknowledgement "Magazine-to-spindle tool change"
PROC: Process number (0-6)
POS: Position (1-4)
SPINDLE: Spindle number (1-3)

Function result

0 NC command "Tool change from magazine to spindle" acknowledged or function is not active
1 NC command "Tool change from magazine to spindle" is acknowledged

Magazine-to-spindle tool change sequence

```
NC                  program    PLC                                              program
Process 1                      Process1

                      ( 1 )    |                                            |
    N010      TMS              |                          +---------+       |
                               |                          |TMS      |   (2) |
                               +--------------------------|ACTIVE   +-----+ |
                               |                    1 --  |PROC     |     | |
                               |                          +---------+     | |
                               |                   +----------------------+ |
                               |                   |       Initiates tool   | (3)
                               |                   +--------- ch. from mag  |
                               |                             to spindle     |
                               |  +------------------------------------+    |
                               |  |  Physical and logical              |    | (4)
                               |  |  tool change                       |    |
                               |  +------------------------------------+    |
                               |                                            |
                               |  Tool ch. from mag.   --------------------+ | (5)
                               |  to spindle executed                      | |
                               |              +----------------------------+ |
                               |              |           +---------+        |
                               |              |           |TMS_Q    |        | (6)
                               |              +-----------|QUIT     +--      |
                               |                    1 --  |PROC     |        |
                      ( 7 )    |                          +---------+        |
    N011 ...
         ...
```

Timing of magazine-to-spindle tool change:

```
                           (1)
NC command                 +---------------------------------------------------┐
TMS                    ----+                                                   └--------

                               (2)
PLC function                   +-----------------------------------------------┐
TMS                    --------+                                               └--------

                                                                           (6)   (7)
PLC function                                                                +----┐
TMS_Q                  -----------------------------------------------------+    └-----

                           (3)                 (4)
Tool change                    +--------------------------------------------┐
proc. running          --------+                                            └-----------

                                                                            (5
Ready message                                                               ┌-----------
t. ch. compl.          -----------------------------------------------------+
```

1. In the NC program, the magazine-to-spindle tool change is initiated by the "TMS" command, and output to the PLC. The NC program now waits until the tool change is acknowledged.
2. The "TMS" standard function in the PLC program recognizes the magazine-to-spindle tool change.
3. A tool change sequence is initiated in the PLC program.
4. Usually, a magazine-to-spindle tool change sequence consists of an action (e.g. clamping the clamping device in the spindle) and a tool transfer message to tool management (see tool transfer functions).
5. The magazine-to-spindle tool change has been completed. This can be checked in the PLC program using feedback messages.
6. The "TMS_Q" standard function is used for acknowledging the NC tool change command.
7. The NC program recognizes the completion of the magazine-to-spindle tool change. The NC program is continued.

Interrogation of spindle-to-magazine tool change 'TSM'

The ""TMS" standard function can be used in the PLC program for interrogating a "TMS" spindle-to-magazine tool change that is active in the NC program.

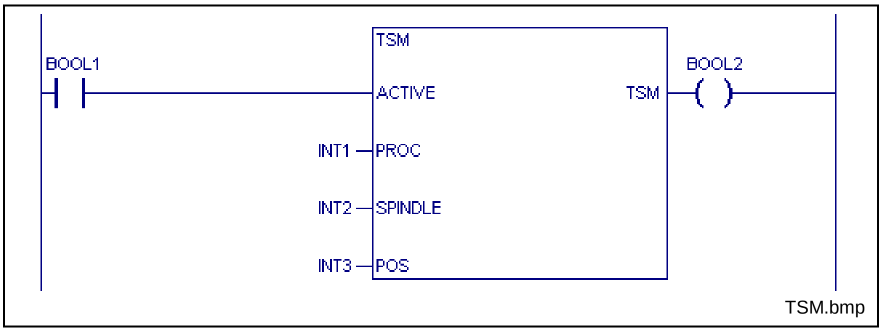


Fig. 2-52: Evaluating a spindle-to-magazine tool change

ACTIVE: 0 Function not active
1 Interrogation "Spindle-to-magazine tool change"
PROC: Process number (0-6)
SPINDLE: Spindle number (1-3)
POS: Position (1-4)

Function result

0 No "Spindle-to-magazine tool change" or function is not active
1 NC command "Spindle-to-magazine tool change" is active

The "TSM_Q" standard function is used in the PLC program for acknowledging the completion of the spindle-to-magazine tool change. This completes tool change processing in the PLC. The NC program is continued when the tool change is acknowledged.

Acknowledgment of spindle-to-magazine tool change 'TSM_Q'

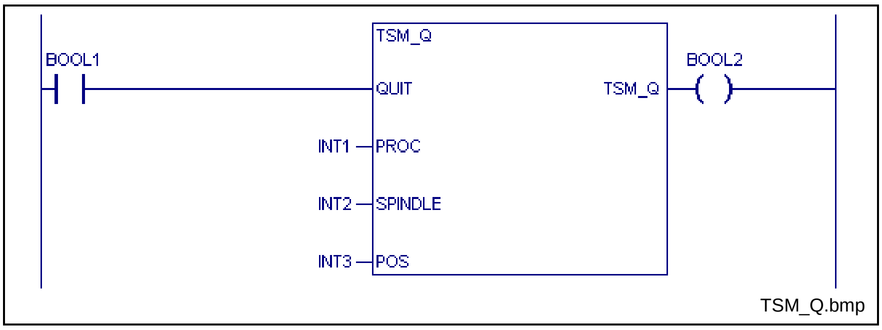


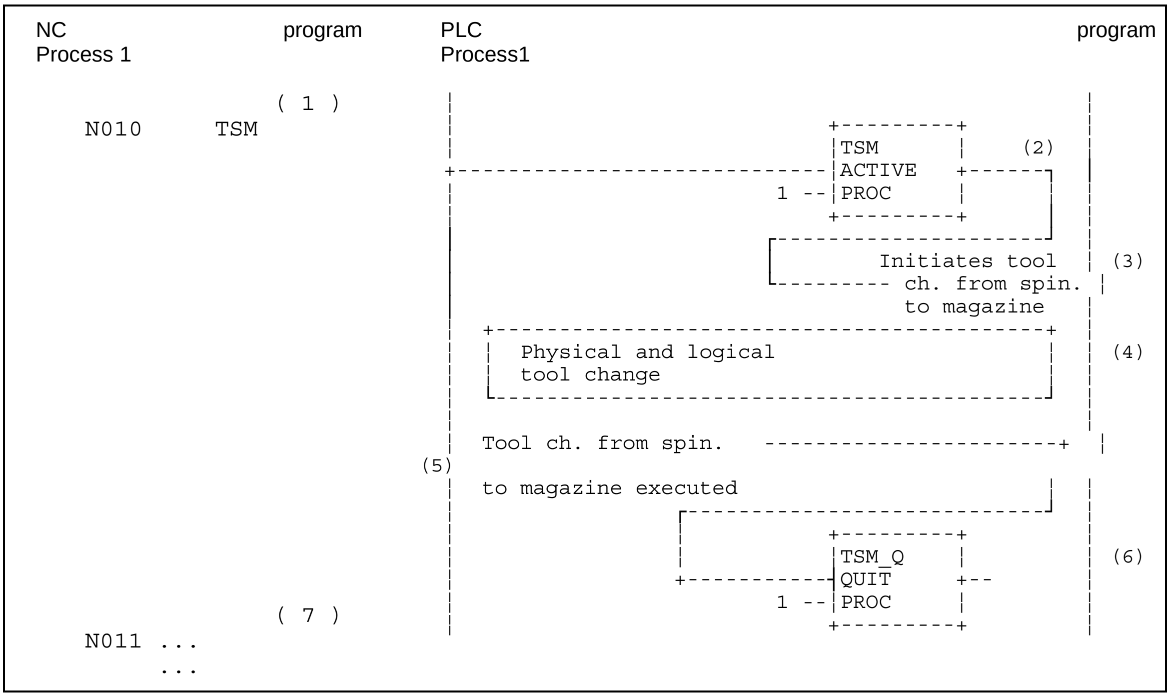
Fig. 2-53: Acknowledging a spindle-to-magazine tool change

QUIT: 0 Function not active
1 Acknowledgement "Spindle-to-magazine tool change"
PROC: Process number (0-6)
SPINDLE: Spindle number (1-3)
POS: Magazine position offset (1-4)

Function result

0 NC command "Spindle-to-magazine tool change" acknowledged or function is not active
1 NC command "Spindle-to-magazine tool change" is acknowledged

Spindle-to-magazine tool change sequence



Timing of spindle-to-magazine tool change:

```
                        (1)
NC command             +---------------------------------------------------┐
TSM                ----+                                                   └--------

                            (2)
PLC function               +-----------------------------------------------┐
TSM               --------+                                                └--------

                                                                       (6)   (7)
PLC function                                                            +-----┐
TSM_Q              -----------------------------------------------------+      └-----

                        (3)                 (4)
Tool change                +-------------------------------------------┐
proc. running      --------+                                           └-----------

                                                                        (5
Ready message                                                           ┌-----------
t. ch. compl.      -----------------------------------------------------+
```

1. In the NC program, the spindle-to-magazine tool change is initiated by the "TSM" command, and output to the PLC. The NC program now waits until the tool change is acknowledged.
2. The "TCH" standard function permits the complete NC tool change to be recognized in the PLC program.
3. A tool change sequence is initiated in the PLC program.
4. Usually, a spindle-to-magazine tool change sequence consists of an action (e.g. unclamping the clamping device in the spindle) and a tool transfer message to tool management (see tool transfer functions).
5. The spindle-to-magazine tool change has been completed. This can be checked in the PLC program using feedback messages.
6. The "TMS_Q" standard function is used for acknowledging the NC tool change command.
7. The NC program recognizes the completion of the spindle-to-magazine tool change. The NC program is continued.

2.6 Tool transfer Functions

Method of Operation of a Tool Transfer

Rexroth MTC 200 tool management monitors the current location of the tools of a process. Besides being in a magazine pocket or in a spindle, a tool can be in a gripper during the tool changing process.

Any transport of a tool between these locations that is initiated by the PLC program must be reported to tool management via the tool transfer commands.

The tool transfers are always initiated by tool transfer standard functions in the PLC (e.g. clamping or unclamping of tool clamping devices in spindles; opening or closing grippers).

The tool transfer messages from the PLC notify NC tool management of the transfer of a tool. A defined handshake procedure between PLC and NC that employs the tool transfer function eventually leads to an exchange of the tools in tool management.

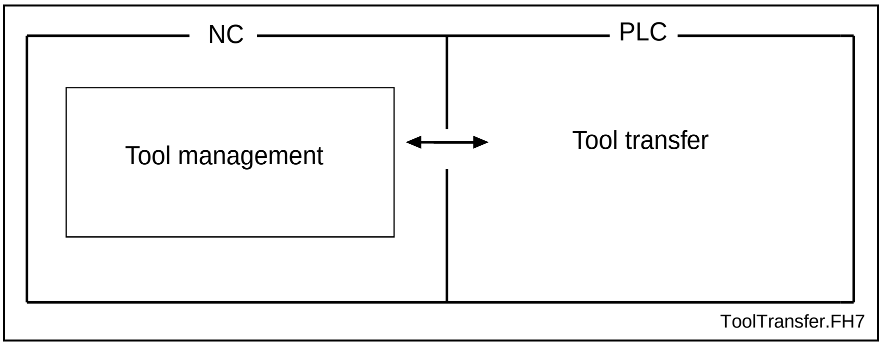


Fig. 2-54: Method of operation of a tool transfer

The following table gives an overview of the tool transfer options:

Tool transfer		ABBREVIATION
from	to	
Magazine	Spindle	XMS
Magazine	Gripper	XMG
Spindle	Magazine	XSM
Spindle	Gripper	XSG
Gripper	Magazine	XGM
Gripper	Spindle	XGS

Werkzeugtransferbefehle_V22_20021126.xls

Fig. 2-55: Overview of tool transfer commands

Example

There are several standard functions available for tool transfer. The "XMS" standard function (magazine-to-spindle tool transfer) in the PLC program initiates the tool transfer from the magazine to the spindle.

The tool management function checks the requested tool transfer for errors. The "XMS_NA" standard function (no permission for magazine-to-spindle tool transfer) permits errors to be interrogated.

A tool transfer is completed by the "XMS_Q" ("magazine-to-spindle tool transfer terminated" acknowledgment) standard function in the PLC program.

The "XMS_CA" (abort magazine-to-spindle tool transfer) standard function permits incorrect tool transfers to be aborted.

The following sections give a more detailed description of the sequence of tool transfer functions using magazine-to-spindle tool transfer as an example.

Tool Transfer Procedure and Function Modules

Magazine-to-spindle tool transfer 'XMS'

The "XMS" standard function is used in the PLC program for initiating the magazine-to-spindle tool transfer. The tool management function checks the requested tool transfer and reports the result to the PLC.

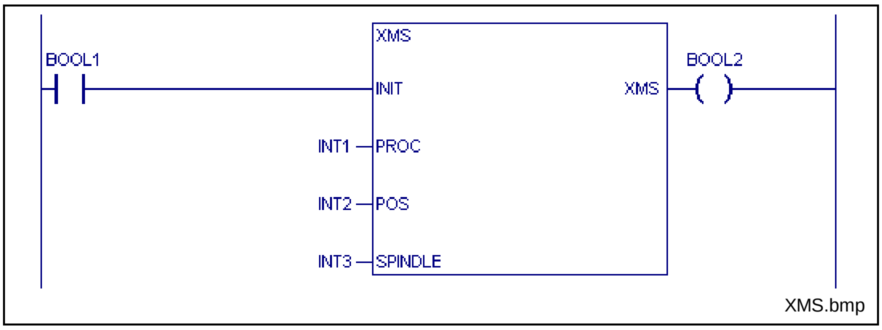


Fig. 2-56: Interrogating a magazine-to-spindle tool transfer

ACTIVE: 0 Function not active
1 Initiate magazine-to-spindle tool transfer
PROC: Process number (0-6)
POS: Position (1-4)
SPINDLE: Spindle number (1-3)

Function result

0 Function is not active ("INIT"=0), or magazine-to-spindle tool transfer has been initiated
1 Magazine-to-spindle tool transfer is initiated

Magazine-to-spindle tool transfer permitted 'XMS_PA'

The tool management function checks a requested tool transfer and reports the result to the PLC. The "XMS_PA" standard function in the PLC program enables the feedback message "tool transfer permitted" to be interrogated.

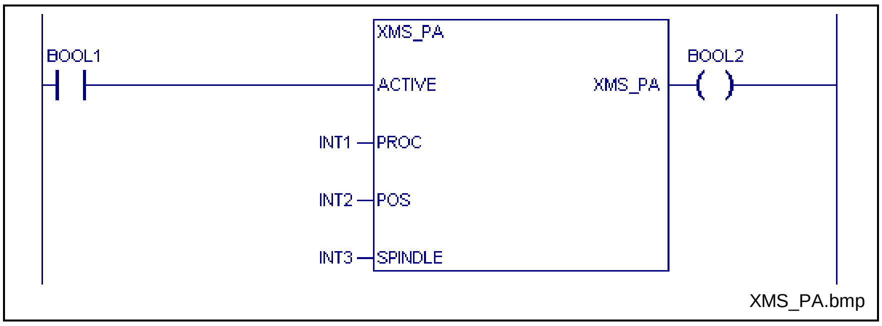


Fig. 2-57: Interrogating a tool transfer permission

ACTIVE: 0 Function not active
1 Interrogation of "Magazine-to-spindle tool transfer permitted"
PROC: Process number (0-6)
POS: Position (1-4)
SPINDLE: Spindle number (1-3)

Function result

0 Function is not active ("ACTIVE"=0) or magazine-to-spindle tool is not permitted
1 Magazine-to-spindle tool transfer is permitted

Magazine-to-spindle tool transfer not permitted 'XMS_NA'

The tool management function checks a requested tool transfer and reports the result to the PLC. Any error caused by the requested tool transfer can be interrogated with the "XMS_NA" standard function.

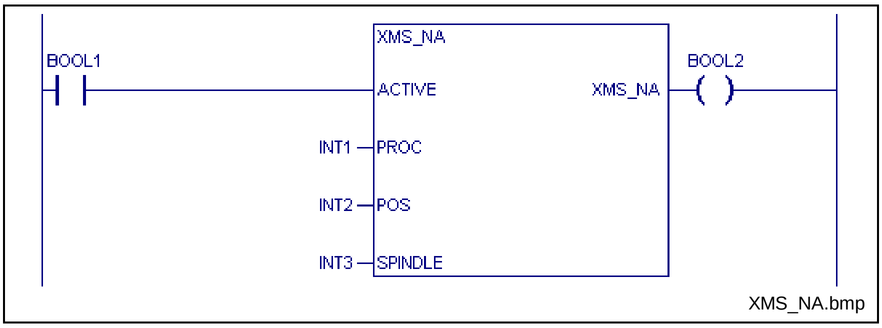


Fig. 2-58: Interrogating a tool transfer permission

ACTIVE: 0 Function not active
1 Interrogation of "Magazine-to-spindle tool transfer is not permitted"
PROC: Process number (0-6)
POS: Position (1-4)
SPINDLE: Spindle number (1-3)

Function result

0 Function is not active ("ACTIVE"=0) or magazine-to-spindle tool transfer is permitted.
1 Magazine-to-spindle tool transfer is not permitted

Acknowledgment of magazine-to-spindle tool transfer 'XMS_Q'

A tool transfer that has been authorized by the tool management function is terminated by the "XMS_Q" standard function in the PLC program.

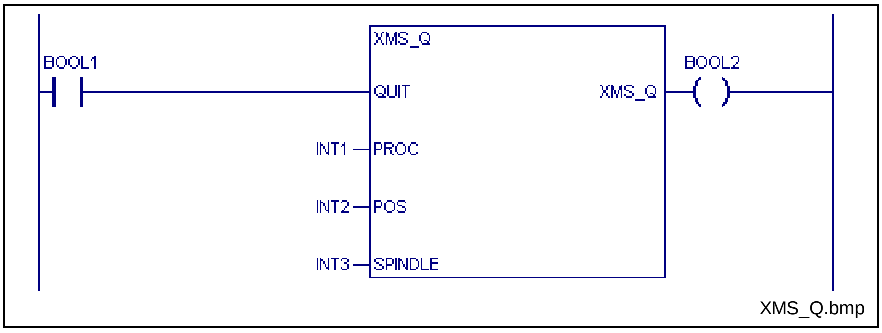


Fig. 2-59: Acknowledging a tool transfer

ACTIVE: 0 Function not active ("QUIT"=0)
1 Terminate magazine-to-spindle tool transfer
PROC: Process number (0-6)
POS: Position (1-4)
SPINDLE: Spindle number (1-3)

Function result

The "QUIT" input is passed on.

Aborting magazine-to-spindle tool transfer 'XMS_CA'

The "XMS_CA" standard function permits the tool transfer to be aborted in the event of a malfunction during the transfer.

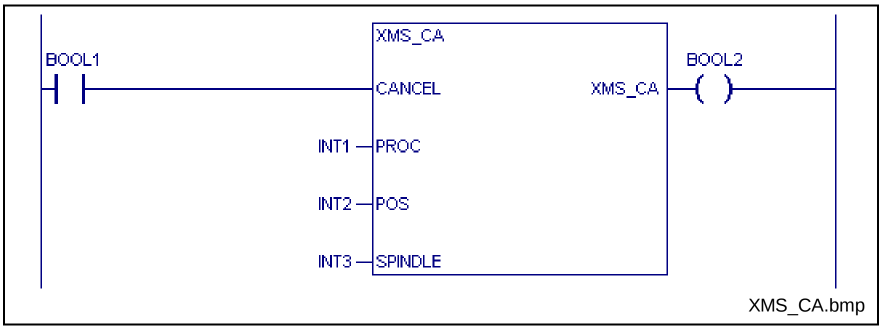
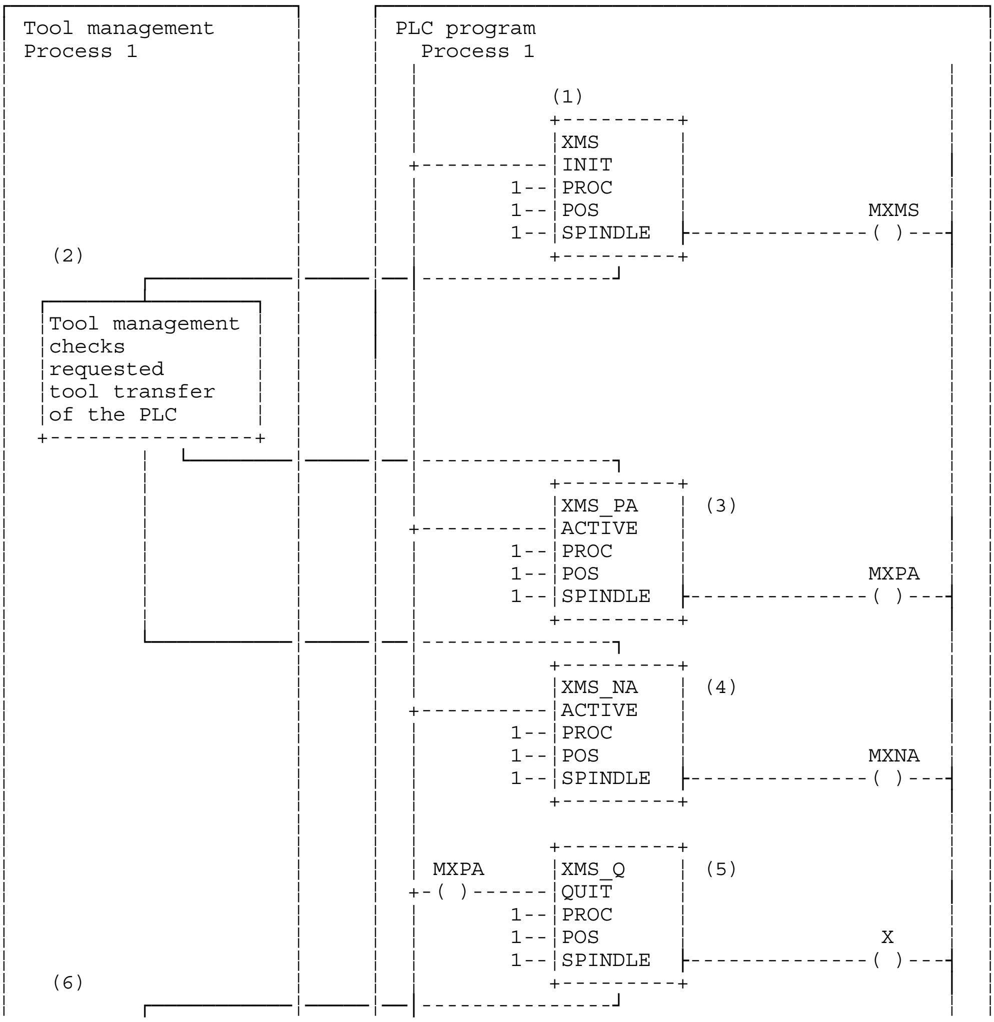


Fig. 2-60: Aborting a tool transfer

ACTIVE: 0 Function not active
1 Abort magazine-to-spindle tool transfer
PROC: Process number (0-6)
POS: Position (1-4)
SPINDLE: Spindle number (1-3)

Function result The "CANCEL" input is passed on.

Tool transfer sequence



```
 |    +-------|--------+  |     | |                                   | |
 |    |Tool management |  |     | |                                   | |
 |    |executes        |  |     | |                                   | |
 |    |requested       |  |     | |                                   | |
 |    |tool transfer   |  |     | |                                   | |
 |    |of the PLC      |  |     | |                                   | |
 |    +----------------+  |     | |                                   | |
 |                        |     | |                                   | |
 |                        |     | |                 +----------+      | |
 |                        |     | | CLRERR   MXNA   |XMS_CA    | (7)  | |
 |                        |     | +-| |------| |----|CANCEL    |      | |
 |                        |     | |            1--|PROC        |      | |
 |                        |     | |            1--|POS         |   X  | |
 |                        |     | |            1--|SPINDLE     |---( )-| |
 |     (8)                |     | |                 +----------+      | |
 |           +------------|-----|-|-------------------------+         | |
 |    +------|---------+  |     | |                                   | |
 |    |Tool management |  |     | |                                   | |
 |    |aborts          |  |     | |                                   | |
 |    |requested       |  |     | |                                   | |
 |    |tool transfer   |  |     | |                                   | |
 |    |of the PLC      |  |     | |                                   | |
 |    +----------------+  |     | |                                   | |
 +------------------------+     +-------------------------------------+
```

Tool transfer timing:

```
                          (1)
FK input                 +----------------------------------------------------+
XMS                  ----+                                                    +--------

                               (2)
FK output                     +-----------+
XMS                  ---------+           +----------------------------------------------

FK input                      +-------+
XMS_PA               ---------+       +------------------------------------------------

                                  (3)
FK output                        +-------------------------------+
XMS-PA               ----------+                                 +--------------------

                                               (5)                              (6)
FK input                                      +--------------------------------+
XMS-Q                -------------------------+                                +-------

FK output                                     +--------------------------------+
XMS-Q                -------------------------+                                +-------
```

1. The magazine-to-spindle tool transfer is initiated by the "XMS" standard function in the PLC program.
2. The tool management function checks the requested tool transfer.
3. The "XMS_PA" standard function checks in the PLC program whether the requested tool transfer is permitted (XMS_PA: eXchange tool from Magazine to Spindle Positive Acknowledge).
4. The "XMS_NA" standard function checks in the PLC program whether the requested tool transfer is not permitted (XMS_NA: eXchange tool from Magazine to Spindle Negative Acknowledge).
5. If tool transfer is permitted, the "XMS_Q" standard function in the PLC program completes the tool transfer.
6. The tool management function performs the logical tool transfer.
7. The requested tool transfer can be aborted with the "XMS_CA" standard function if it is not permitted.
8. The tool management function aborts the tool transfer.

Further Tool Transfer Functions

Upon activation of the "XFER_CHK" function, the tool transfer always refers to the current actual magazine location.

```
PxxS.MGCP ──┤ :=        ├── PxxC.MGAP
```

The tool transfer functions between:

- magazine and gripper,
- spindle and magazine,
- spindle and gripper,
- gripper and magazine, and
- gripper and spindle

execute in the same way as the tool transfer between magazine and spindle.

The following functions are available for tool transfer:

PLC tool transfer commands						V22_20021029
from	**to**	**initiate**	**permitted**	**terminate**	**not permitted**	**abort**
Magazine	Spindle	XMS	XMS_PA	XMS_Q	XMS_NA	XMS_CA
Magazine	Gripper	XMG	XMG_PA	XMG_Q	XMG_NA	XMG_CA
Spindle	Magazine	XSM	XSM_PA	XSM_Q	XSM_NA	XSM_CA
Spindle	Gripper	XSG	XSG_PA	XSG_Q	XSG_NA	XSG_CA
Gripper	Magazine	XGM	XGM_PA	XGM_Q	XGM_NA	XGM_CA
Gripper	Spindle	XGS	XGS_PA	XGS_Q	XGS_NA	XGS_CA

Werkzeugtransferbefehle_SPS_V22_20021029.xls

Fig. 2-61: Overview of tool transfer commands

Checking a tool transfer 'XFER_CHK'

Tool management of the Rexroth MTC 200 is performed in the NC section. The tool change moves the tools between magazine, gripper and spindle. Tool transfer standard functions in the PLC initiate and acknowledge the tool transitions (e.g. clamping or unclamping of tool clamping devices in spindles; opening or closing grippers). Up to Version 04.10, initiating a tool transfer is possible only with an active tool change command of the NC. If tool change is not active, an error message is generated when a tool transfer is initiated.

Activating the "XFER_CHK" standard function suppresses the error message in the event of a tool transfer without active tool change. Thus, a tool change may also be performed outside an active NC tool change command. Furthermore, it is now possible to add tools to the tool list or to shift and cancel tools from the operator interface as well as from the PLC while the NC program is running; however, this does not apply for the current tool or the tool in the reference spindle.

If the function has not been activated or is not used, error handling responds as before. An error message is generated when a tool transfer without active tool change is initiated. While an NC program is running, tools cannot be added, shifted, or cancelled.

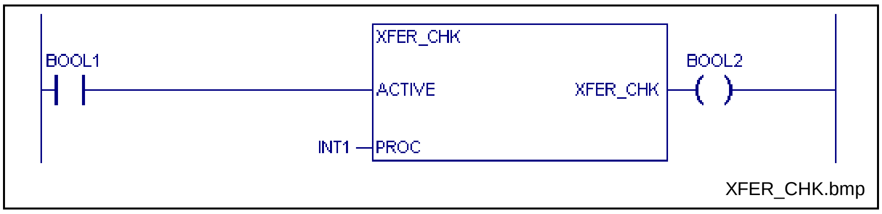


Fig. 2-62: Checking a tool transfer

ACTIVE: 0 Function is not activated. Error message with tool transfer without active tool change for the specified process

1 Suppress error message with tool transfer without active tool change for the specified process

PROC: Process number (0-6)

The ACTIVE input is passed on.

Function result

Function result = ACTIVE

2.7 Combined spindle/turret axis

The combined spindle/turret axis (CST axis) is a special axis type that is used for controlling a tool turret with driven tools.

The special feature of this axis type is the fact that, using a servodrive, both tools (spindle mode) and turret (turret mode) can be driven. The control automatically selects spindle or turret operating mode on the basis of the NC commands.

Turret mode is preselected whenever a tool magazine movement command is executed in the NC program. Once the turret has been positioned, the NC automatically switches back to spindle mode.

The *interrogation* of the mode is controlled by two standard functions:

- interrogating the magazine axis preselection: "MAG_ACT" and
- Interrogating spindle preselection "SPDL_ACT".

There are two more standard functions for *acknowledging* the mode selection:

- acknowledging the magazine axis preselection: "MAG_Q" and
- acknowledging the spindle preselection "SPDL_Q".

Note: Detailed documentation regarding the CST axis can be found in the "Rexroth MTC 200 Combined Spindle/Turret Axis" description of functions.

Application-related comments on the operation of a combined spindle/turret axis

To improve the processing of exceptions when using CST axes, the following items need to be complied with:

Behavior after control reset

Following a control reset (e.g. switching on/off or parameter transfer), spindle mode is the default selection in the control. A subsequent spindle movement command (e.g. "M3 S1000") starts the axis motion immediately. To move the axis, the NC does not require an acknowledgment from the PLC in this case. Omitting the acknowledgment signal will not lead to an error message in this case.

Undefined tool turret position

The CST axis must not be moved if the turret is in a mechanical intermediate position after a control reset. The system does not disable a movement that is caused by the execution of a spindle command and/or the execution of an MRF command. Thus, the programmer must set the "AxxC.MHOLD" axis command in this case.

Measure for defined tool turret position

To ensure that the tool turret is in a defined position after a control reset, process parameter Bxx.032 "Home position required" in the machine parameters must be set to "yes" and "MRF" must be programmed in the reverse program. This ensures that the CST axis is homed after the reverse program has been executed and that the mechanical system is in a defined position (the NC requests spindle mode after reference point return). Using the corresponding acknowledgment signal, the NC monitors any turret mode after the "MRF" command has been executed.

Note: Prior to each changeover to turret mode, the NC issues a spindle positioning command with a corresponding changeover request to the PLC. This ensures that a defined switching position is undertaken before the coupling procedure. In other words, "M19 S0" is internally added to command "MRF", as is the case for all other magazine commands, as long as this has not already been programmed in the NC user program.

Behavior upon power activation

If the mechanical turret system is in a defined operating state (turret mode or spindle mode) before power is activated (PxxS.POWER=0), and if this state is acknowledged by the PLC, the NC transfers this operating state as a request to the PLC. The request remains when power is applied (PxxS.POWER=1). A control reset (PxxC.CLEAR=1) can be used to clear the pending request after power has been applied (PxxS.POWER=0).

Acknowledging requests

Once the corresponding operating state (turret or spindle mode) has been acknowledged for the first time, the acknowledgment signal must remain applied as long as the operating state is requested by the NC if the NC requests spindle or turret mode after power has been applied (PxxS.POWER=1). Canceling the acknowledgment signal will cause process error #321 "CST acknowledgment is missing" to be issued in this case.

The request is reset when a control reset is executed. If the "Bxx.036 Manual axis jogging causes reset" parameter has been set to "Yes", the corresponding request will also be reset when a spindle jog command is activated.

Home

If the combined spindle/turret axis has not been homed, command "MRF" (Tool magazine homing) will always be performed before a tool magazine motion command is executed. Once homing has been completed, the required turret movement is carried out immediately.

If the spindle has not yet been oriented, a spindle orientation with "M19 S0" is performed internally before an MRF is executed.

Aborting a changeover process

If a tool magazine movement command is being processed in the NC, the NC sends a request to switch the turret mechanical system from spindle mode to turret mode to the PLC. At the same time, the axis is switched to turret mode within the NC (axis processor). If the PLC does not acknowledge turret mode, a control reset (PxxC.CLEAR=1 and PxxS.ERROR=0) can be used for aborting the changeover process. After Control Reset has been completed, the NC requests spindle mode again. The axis is also switched back to spindle mode in the axis processor.

Primary mode of operation

Spindle mode is the main mode of the CST axis. As a result, a switch to turret mode occurs only by executing tool magazine movement commands (e.g. MTP in the NC program or PxxC.MGPOS via PLC). Once this command has been completed, the NC sends the PLC a request for switching the mechanical turret system from turret mode to spindle mode. At the same time, the axis is switched back to spindle mode. If the PLC does not acknowledge spindle mode, a control reset can be used for aborting the changeover process. The spindle mode request from the NC is cleared during this process.

Executing a control reset resets the pending request (e.g. spindle request). If process parameter "Bxx.036 Manual axis jogging causes reset" has been set to "Yes", the spindle request will also be canceled when the spindle is jogged.

Interrogating and Acknowledging Turret Mode Preselection

The control preselects turret mode whenever a command is programmed in the NC program that initiates a turret movement. The "MAG_ACT" standard function can be used in the PLC for interrogating the mode.

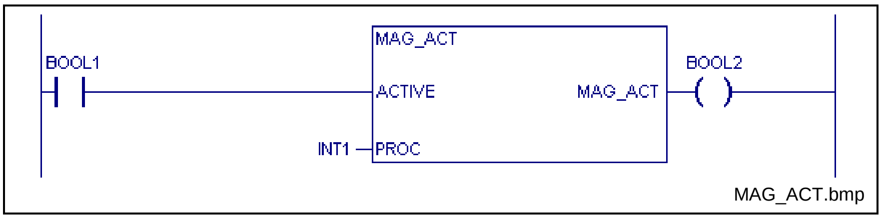


Fig. 2-63: Actively interrogating a magazine

ACTIVE: 0 Function not active
1 Interrogation of turret mode preselection active
PROC: Process number

Function result

0 Function is not active or turret mode preselection is not active
1 Turret mode preselection is active

The "MAG_Q" standard function is used for acknowledging that turret mode selection has been performed. Once turret mode has been acknowledged for the first time, the acknowledgment signal must be active as long as turret mode is selected. Canceling the acknowledgment during turret mode will cause a process error to be generated.

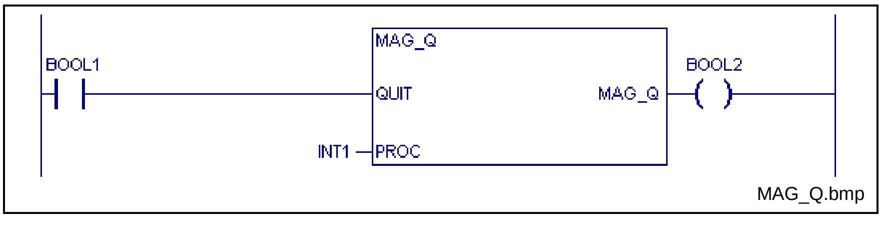


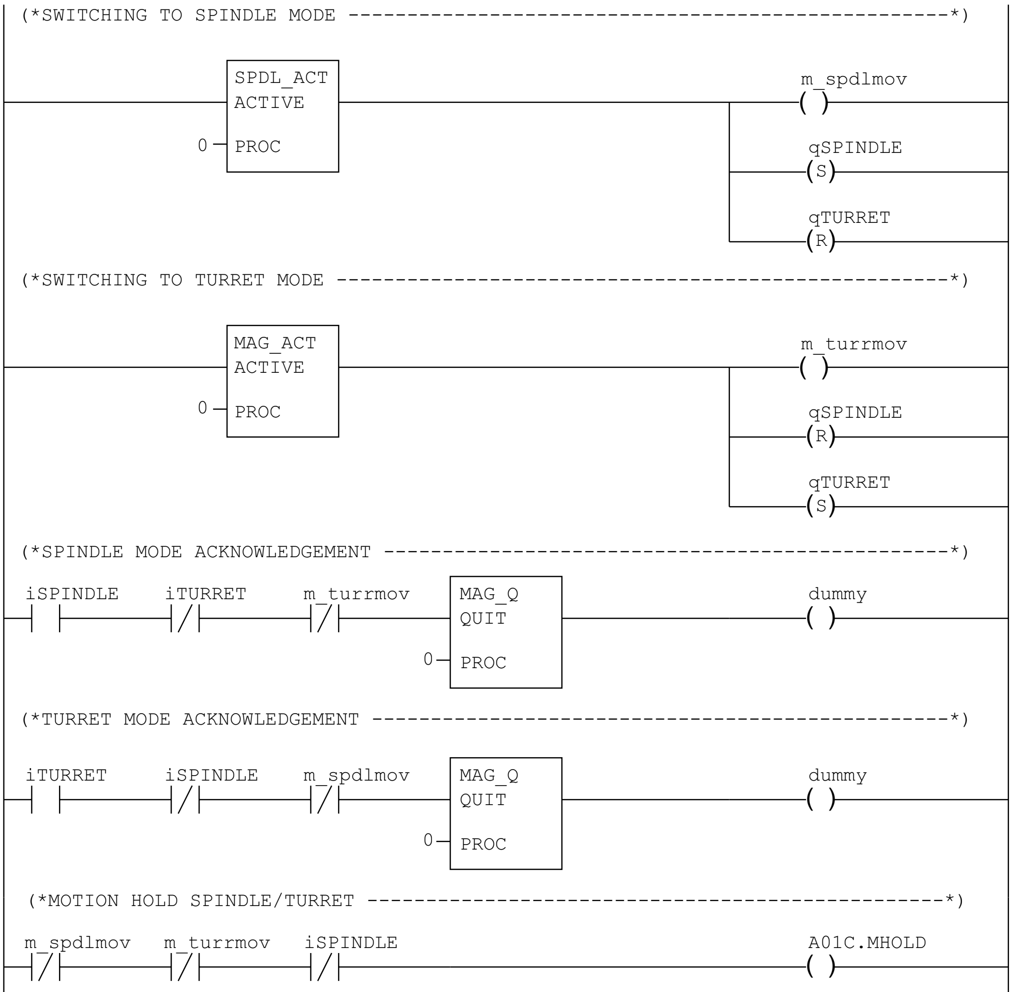
Fig. 2-64: Acknowledge magazine

QUIT: 0 Function not active
1 Acknowledgment of turret mode preselection (must be active the whole time)
PROC: Process number

Function result

The QUIT input is passed on.

Turret mode preselection procedure:



SPDL_ACT			SPDL_ACT
m_spdlmov	FLAG:SPINDLE MODE REQUEST		BOOL
qSPINDLE	SET SPINDLE MODE	%Q0.905.7	BOOL
qTURRET	SET TURRET MODE	%Q9.905.6	BOOL
MAG_ACT			MAG_ACT
m_turrmov	FLAG:TURRET MODE REQUEST		
iSPINDLE	SPINDLE MODE FEEDBACK	%Q0.916.0	BOOL
iTURRET	TURRET MODE FEEDBACK	%Q0.916.1	BOOL
SPDL_Q			SPDL_Q
dummy	AUXILIARY FLAG		BOOL
MAG_Q			MAG_Q
A01C.MHOLD	MOTION HOLD		BOOL

list021.FH7

Fig. 2-65: Preselection of turret mode

Interrogating and Acknowledging Spindle Mode Preselection

The controller preselects spindle mode whenever the NC program does not execute a command that initiates a turret movement. The "SPDL_ACT" standard function can be used in the PLC for interrogating this mode.

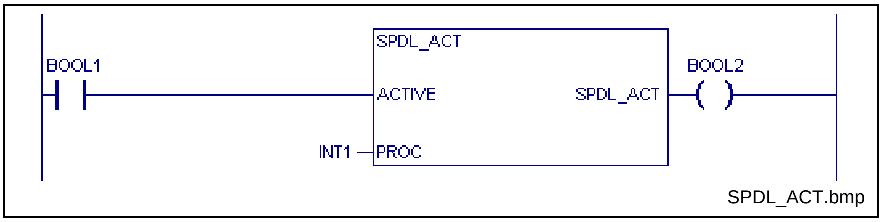


Fig. 2-66: Actively interrogating a spindle

ACTIVE: 0 Function not active
1 Interrogation of spindle mode preselection is active
PROC: Process number

Function result

0 Function is not active or spindle mode preselection is not active
1 Spindle mode preselection is active

The "SPDL_Q" standard function is used for acknowledging that spindle mode selection has been performed. After the control has been switched on, spindle mode will be selected even if there is no acknowledgment. Once spindle mode has been acknowledged for the first time, the acknowledgment signal must be active as long as spindle mode is selected. Canceling the acknowledgment during spindle mode will cause a process error to be generated.

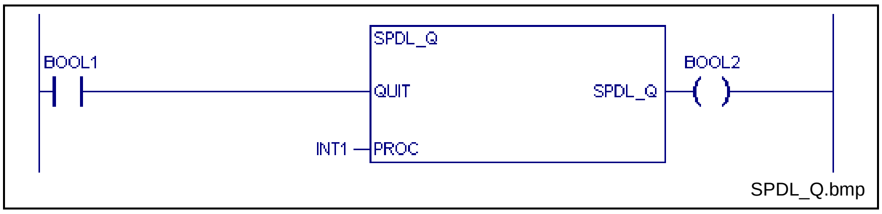


Fig. 2-67: Acknowledging spindle mode

QUIT: 0 Function not active
1 Acknowledgment of spindle mode preselection (must be active the whole time)
PROC: Process number

Function result

The QUIT input is passed on.

Spindle mode preselection procedure:

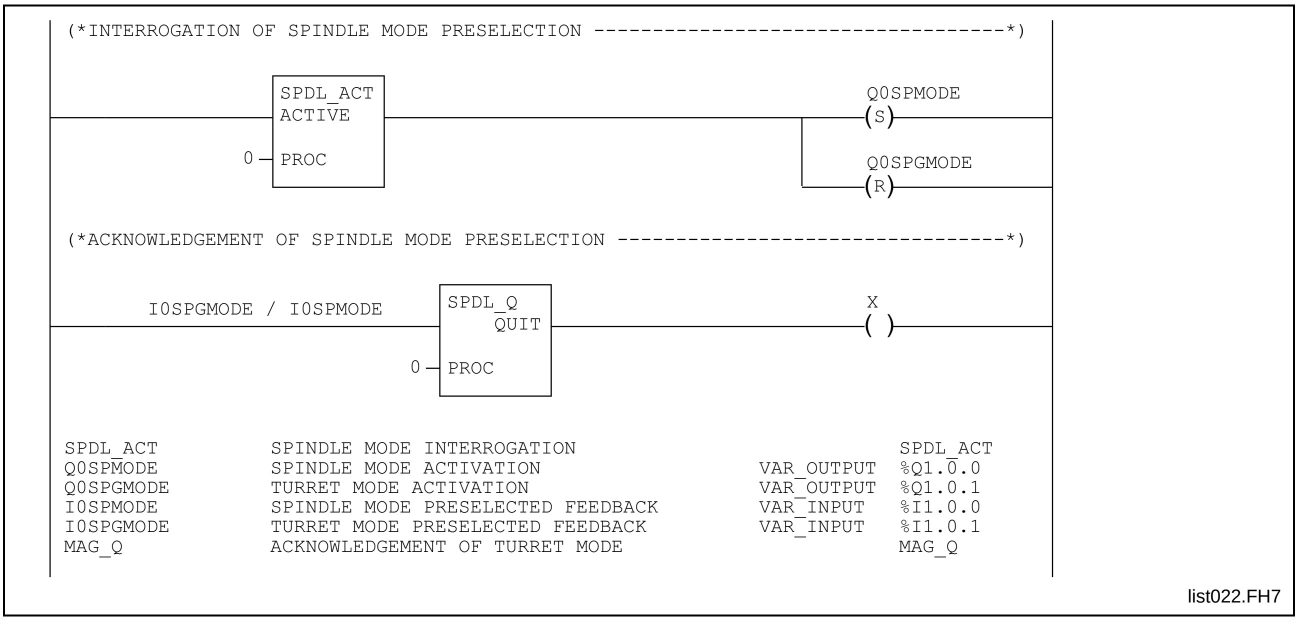


Fig. 2-68: Preselection of spindle mode

2.8 Selectable Main Spindle / Rotary Axis Mode

The *interrogation* of the mode is controlled by two standard functions:

- Interrogating the rotary axis mode preselection: "ROTMOD" and
- interrogating the spindle mode preselection: "SPMOD".

There are two more standard functions for *acknowledging* the mode selection:

- Acknowledging the rotary axis mode preselection: "ROTMOD_Q" and
- acknowledging the spindle mode preselection: "SPMOD_Q".

Interrogating and acknowledging rotary axis mode preselection

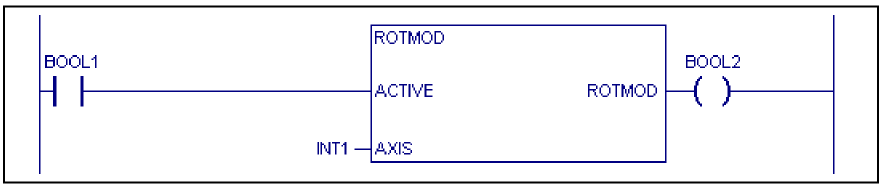


Fig. 2-69: Preselection of rotary axis mode

ACTIVE: 0 Function not active
1 Interrogation of rotary axis mode preselection is active
AXIS: Axis number

Function result

0 Function is not activated or rotary axis mode preselection is not active
1 Rotary axis mode preselection is active

The "ROTMOD_Q" standard function is used for acknowledging rotary axis mode. Once rotary axis mode has been acknowledged for the first time, the acknowledgment signal must be present as long as rotary axis mode is preselected (even after a Control Reset, or after the controller has been switched on). Canceling the acknowledgment during rotary axis mode will cause a process error to be generated.

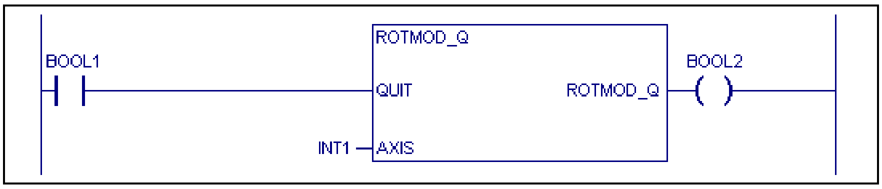


Fig. 2-70: Acknowledging rotary axis mode

ACTIVE: 0 Function not active
1 Acknowledgment of rotary axis mode preselection (must be active the whole time)
AXIS: Axis number

Function result The "QUIT" input is passed on.

Interrogating and acknowledging spindle mode preselection

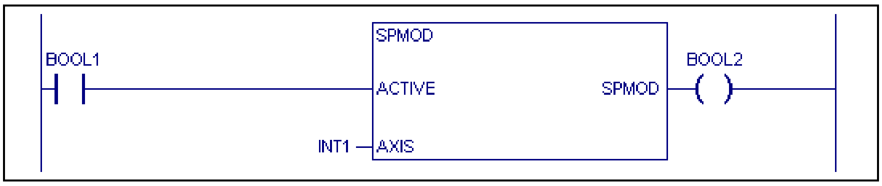


Fig. 2-71: Preselection of spindle mode

ACTIVE: 0 Function not active
1 Interrogation of spindle mode preselection is active
AXIS: Axis number

Function result
0 Function is not active or spindle mode preselection is not active
1 Spindle mode preselection is active

The "SPMOD_Q" standard function is used for acknowledging spindle mode preselection. After the controller has been switched on, spindle mode will be selected even if there is no acknowledgment. Once spindle mode has been acknowledged for the first time, the acknowledgment signal must be present as long as spindle mode is preselected (even after a Control Reset, or after the controller has been switched on). Canceling the acknowledgment during spindle mode will cause a process error to be generated.

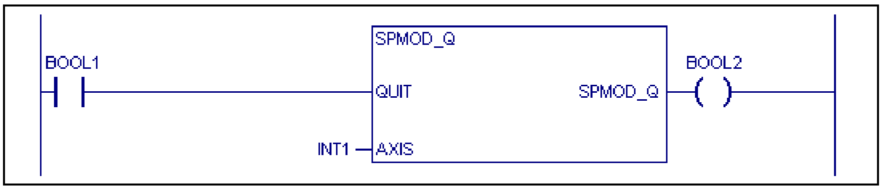
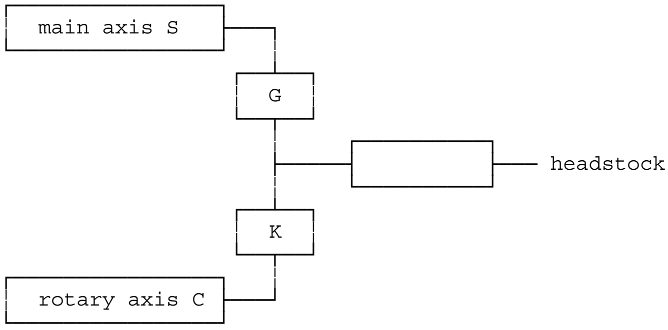


Fig. 2-72: Acknowledging spindle mode

ACTIVE: 0 Function not active
1 Acknowledgment of spindle mode preselection (must be active the whole time)
AXIS: Axis number

Function result The "QUIT" input is passed on.

Example: C Axis mode with main and rotary axis



In main axis mode, the main axis "S" drives the headstock via a gearbox "G". The clutch "K" is open in this case (i.e. the rotary axis "C" is not involved in this movement).

In rotary axis mode, the rotary axis "C" is connected via a clutch. Activating the clutch de-activates the spindle's controller enabling signal so that the spindle idles. The controller enable signal is activated when main spindle mode is selected.

PLC program sequence:

The spindle is initialized in the PLC program when the controller is switched on. The "spindle speed", "spindle acceleration" and "spindle position" parameters are set via the SERCOS required data channel.

"Main spindle/rotary axis mode" selection is initiated by the NC.

Rotary axis mode is selected by:

- Jogging C axis:
  The first job command produces a selection request. There is no movement until the second jog command is issued.
- NC command with C axis

The "ROTMOD" standard function in the PLC interprets rotary axis mode selection. The PLC activates the clutch. The "ROTMOD_Q" function acknowledges the execution of the action. The "ROTMOD_Q" standard function must be provided with data until main spindle mode is selected.

Spindle mode is selected by:

- - main spindle jogging,
- - NC command with spindle.

The "SPMOD" standard function in the PLC interprets main axis mode selection. The PLC releases the clutch. The "SPMOD_Q" function acknowledges the execution of the action. The "SPMOD_Q" standard function must be provided with data until rotary axis mode is selected.

Initializing the main axis

Simple solution: Writing to the parameters without safety prompt

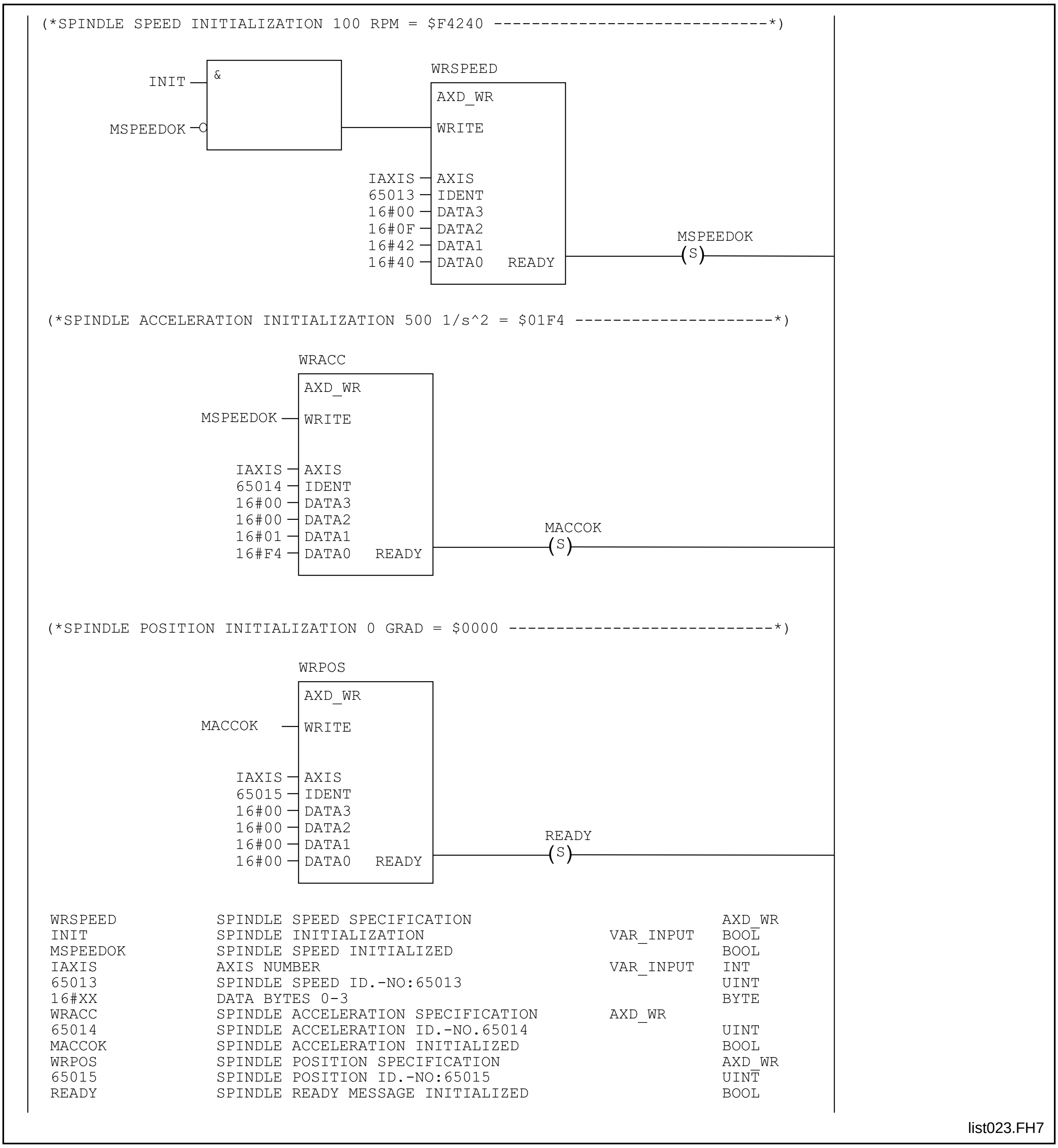


Fig. 2-73: Initialization of the main axis

Better solution: SFC structure for initialization with safety prompt

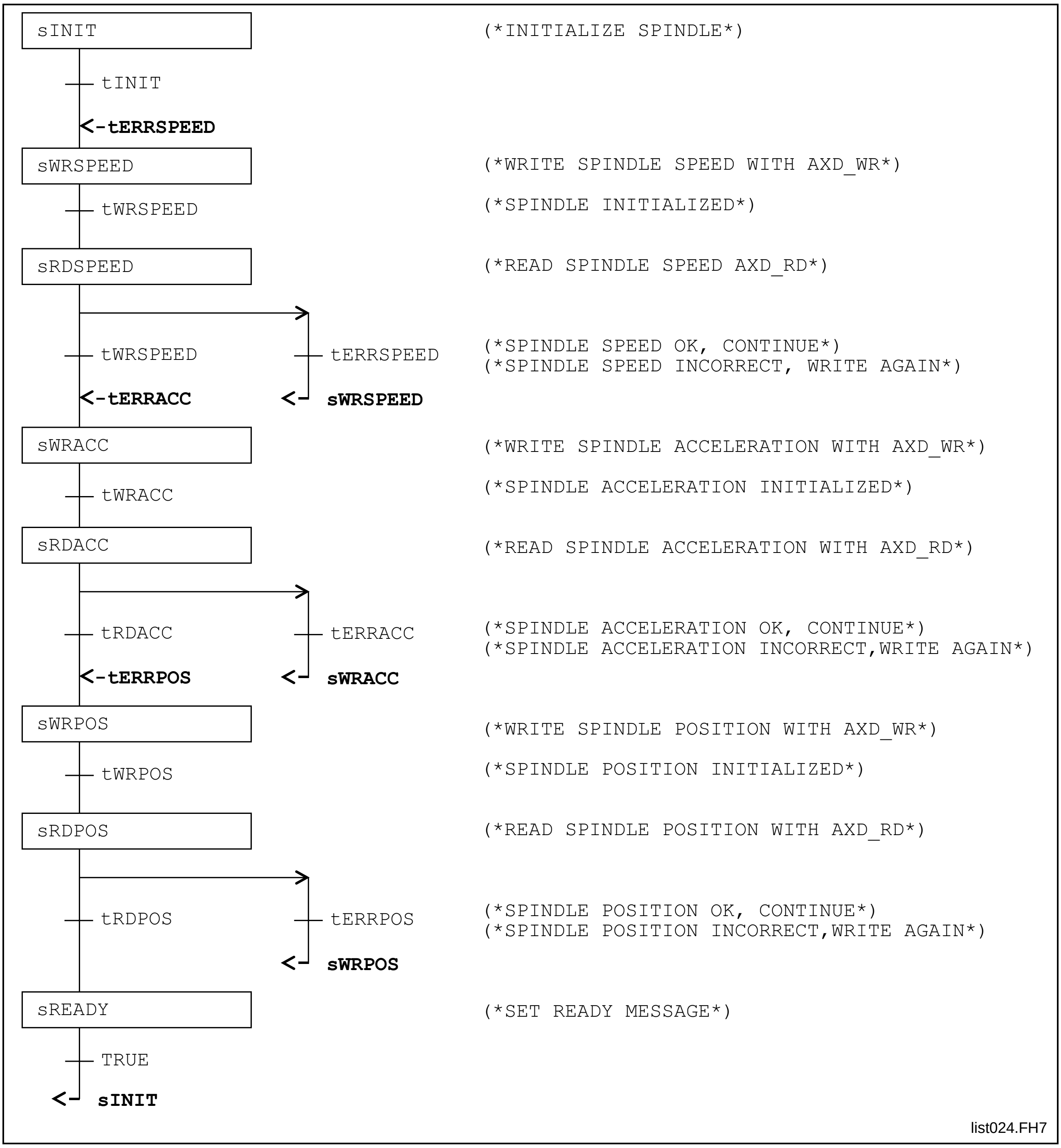


Fig. 2-74: SFC structure for initialization with safety prompt

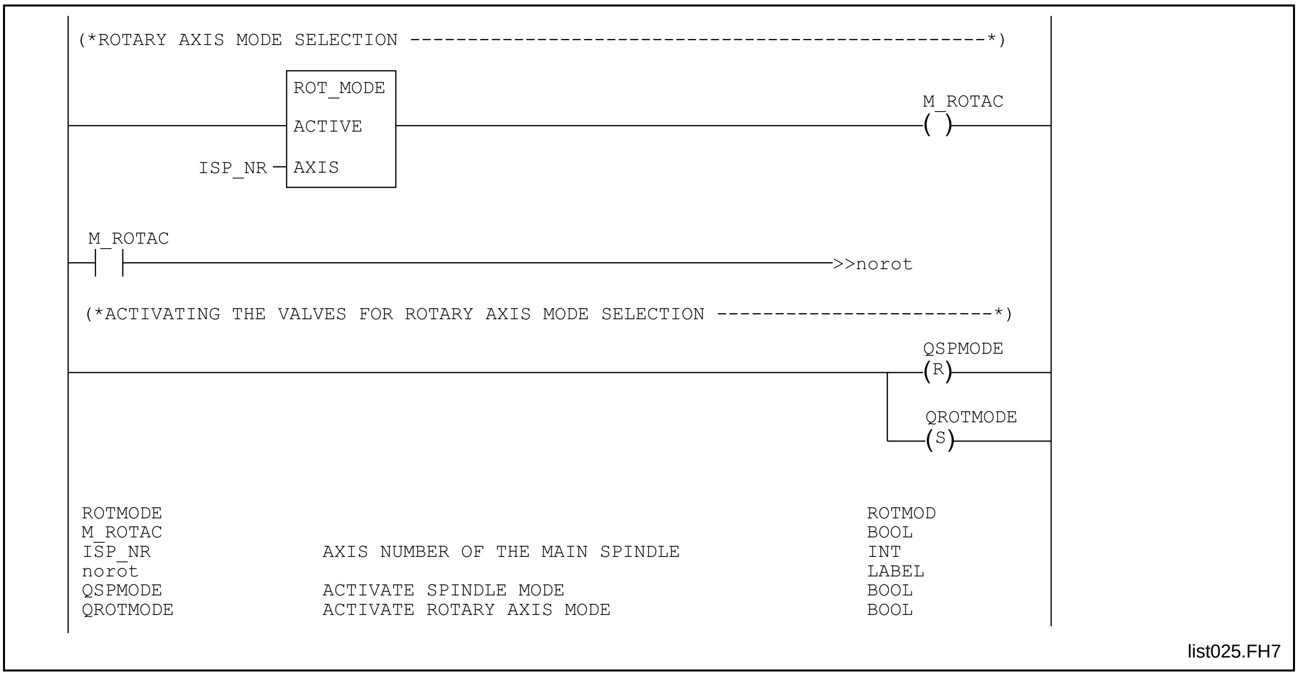


Fig. 2-75: Switching from rotary axis to spindle mode

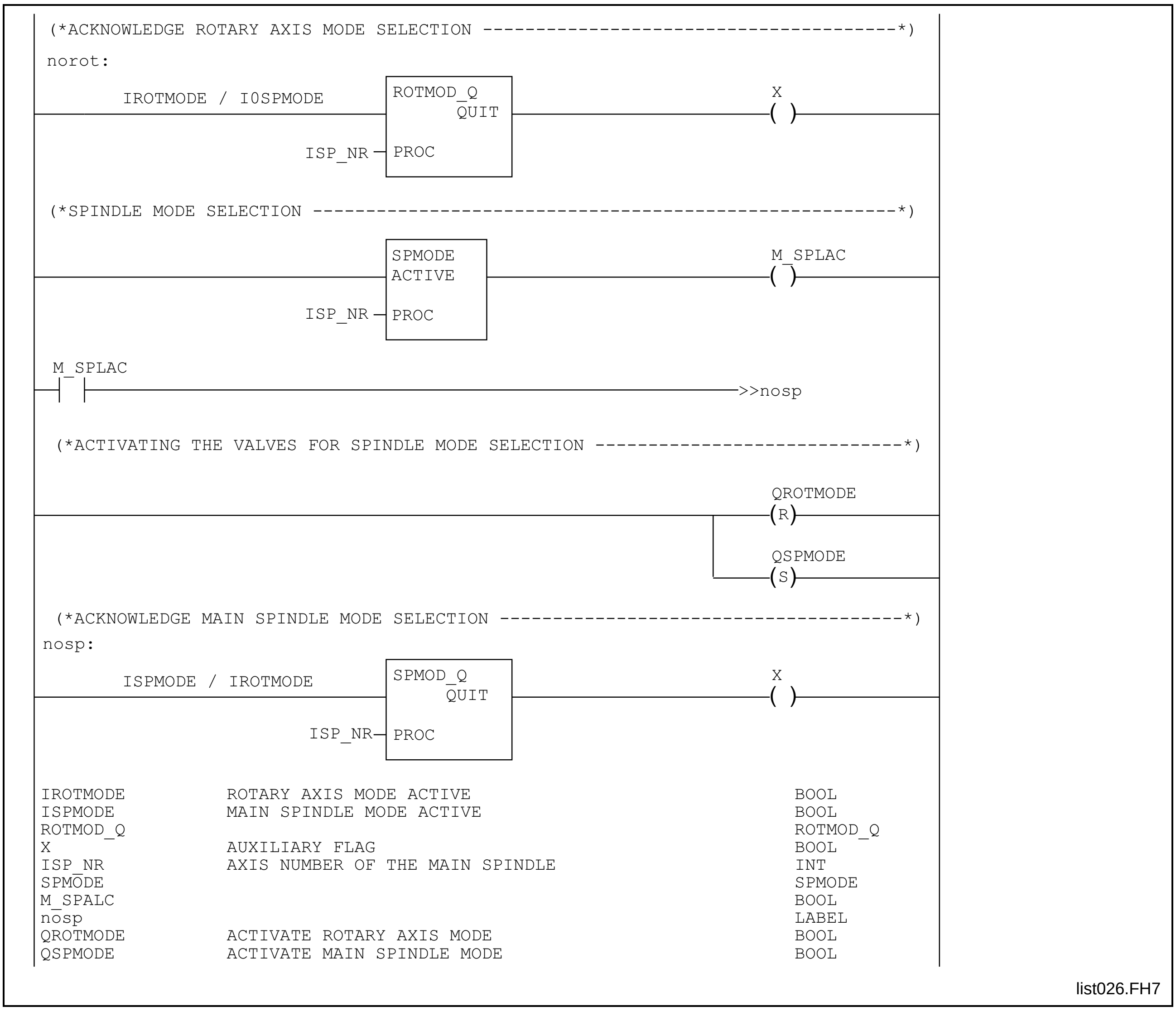


Fig. 2-76: Switching from rotary axis to spindle mode (continued)

2.9 Hand Wheel Functions

General

The BTM unit enables one or more handwheels to be used on a machine. The NC permits handwheel operation in "Setup" mode only.

Motion behaviour: The axes respond without delay to each change of the handwheel. The axis that is linked with the handwheel follows exactly the path the operator specifies by turning the handwheel. Very fast rotations do not lead to incremental losses, drive errors, or incorrect axis positions.

Resolution: Handwheel mode and axis jogging employ the same resolution. Unlike other distances, the "variable jog distance" value from the axis parameters is related to a full handwheel revolution, not to a graduation mark.

Different distance values can be set for the individual handwheels that are used in a process (necessary for grinding machines).

Handwheel – technical data:

- Pulse rate per rotation: 100 increments / rotation
- Pulse number per scale line: 1 increment / scale line
- Scale lines rate per rotation: 100 scale lines / rotation
- max./min. sensor value: +/-2^{15} = +/-32768 (2 byte value)

Handwheel position transfer "HNDWHEEL"

The "HNDWHEEL" function is used for controlling the handwheel in the PLC program. The function must be invoked once for each axis that is to be moved via a handwheel.

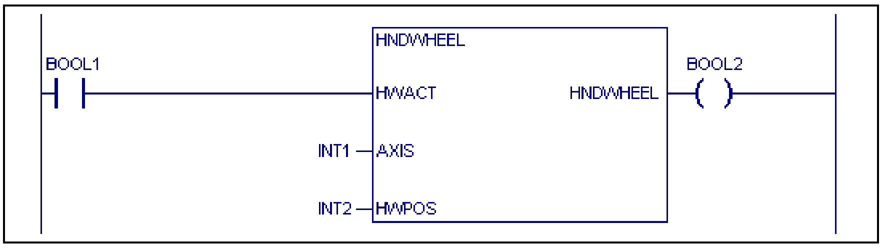


Fig. 2-77: Activating handwheel

HWACT: 0 Handwheel is not active
1 Handwheel is active
AXIS: Axis number (1 ... 32)
HWPOS: Relative handwheel position (-32768 ... 32767)

Function result The "HWACT" input is passed on.

As soon as the PLC sets the "HWACT" function input (handwheel active), the NC interprets the impulses from the handwheel and moves the related "AXIS" (axis number) axis, taking the specified weighting into account.

The PLC reports the handwheel position to the NC as an INTEGER value.

Value range:

$$+/-2^{15} = +/-32768\,(\text{2byte value})$$

Fig. 2-78: Value range handwheel position

As long as handwheel mode is active and the related handwheel is switched on, the PLC user program transfers the values from the handwheel sensor to the "HWPOS" function input (relative handwheel position) in each program cycle.

Note: Handwheel sensor overrange conditions result in a position jump of +32768 and vice versa. The Rexroth MTC 200 takes such an overflow automatically into account.

Prior to switching to MDI operation or changing mode, all handwheels of the relevant process must be deactivated. Otherwise switching is not accepted.

The PLC employs the "PxxCJOGM0", "PxxCJOGM1" and "PxxCJOGM2" process control signals for weighting the jog mode. For each graduation mark, the unit is moved by the preselected display units of the jog mode. The distance that is to be moved is also influenced by the programmed fractional part digits for distances in the process parameters and by the base programming unit.

Process control signals			Distance per graduation mark as a function of the programmed decimal point for path dimension		Unit	Resolution unit (AE)
PxxCJOGM2	PxxCJOGM1	PxxCJOGM0	4	5		
0	0	0	0	0	mm or inches	-
0	0	1	0.0001	0.00001	mm or inches	1 AE
0	1	0	0.001	0.0001	mm or inches	10 AE
0	1	1	0.01	0.001	mm or inches	100 AE
1	0	0	0.1	0.01	mm or inches	1000 AE
1	0	1	1	0.1	mm or inches	10000 AE
1	1	0	10	1	mm or inches	100000 AE
1	1	1	Par.value/100	Par.value/100	mm or inches	Variable Jog distance

Fig. 2-79: Process control signals

The "HNDWHEEL" function issues an error message in the case of the following axis numbers:

- Negative axis number
- Axis number = 0
- Axis number > 32

In the event of an error, the "HNDWHEEL" function returns the error type "-158" and the error number "1".

The time constant of the jerk limitation filter enables the acceleration ramp to be influenced. Small values cause hard and swift reactions to handwheel movements. Higher values cause softer reactions to a change of the handwheel.

The time constant of the jerk limitation filter is calculated as follows:

$$\text{Time constant} = \text{value} * 2\text{ms}$$

Fig. 2-80: Time constant of the jerk limitation filter

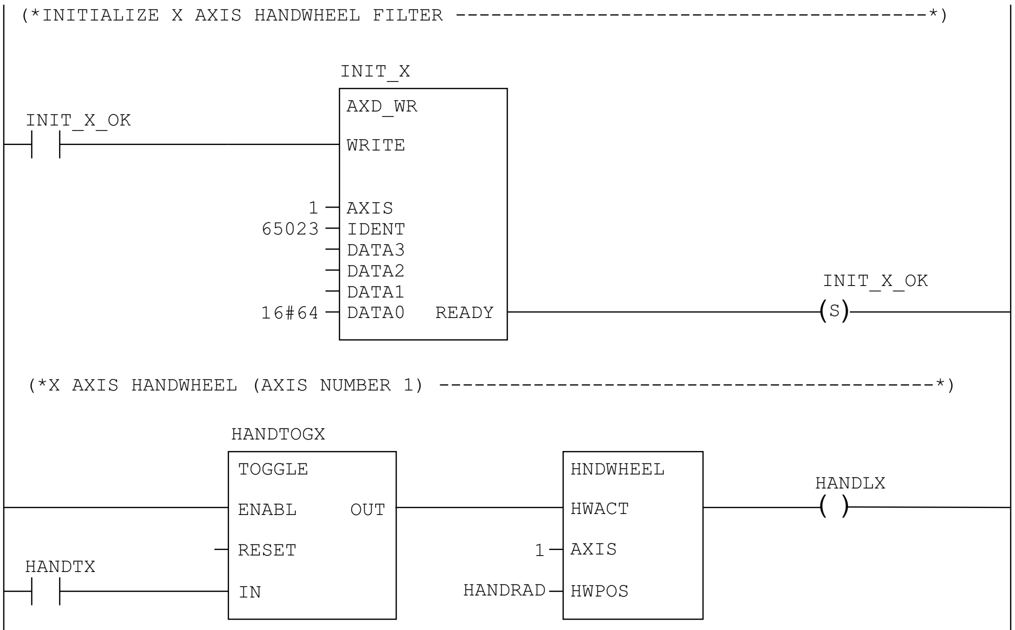
Suggested value: 100 … 200 (hex 16#64 ... 16#C8)

Maximum value: 500

The jerk limitation filter is addressed by the ident number P-7-3583 (for "AXD_WR" PLC function block: 65023).

The jerk limitation filter has an effect only in handwheel mode.

Example: Handwheel initialization and operation



INIT_X_OK	INIT. X AXIS HANDWHEEL FILTER		BOOL
INIT_X	WRITE X AXIS HANDWHEEL FILTER VALUE FB		AXD_WR
1	AXIS NUMBER		ANY_INT
65023	HANDWHEEL FILTER		ANY_INT
16#64			ANY_INT
HANDTOGX	X AXIS HANDWHEEL ON/OFF FB		TOGGLE
HANDTX	X AXIS HANDWHEEL ACTIVE KEY ON/OFF	I2.21.0	BOOL
HANDRAD	RELATIVE HANDWHEEL POSITION	IW2.22	WORD
HANDLX	X AXIS HANDWHEEL ACTIVE LED	Q2.20.0	BOOL

list027.FH7

Fig. 2-81: Initialization and operation of the handwheel

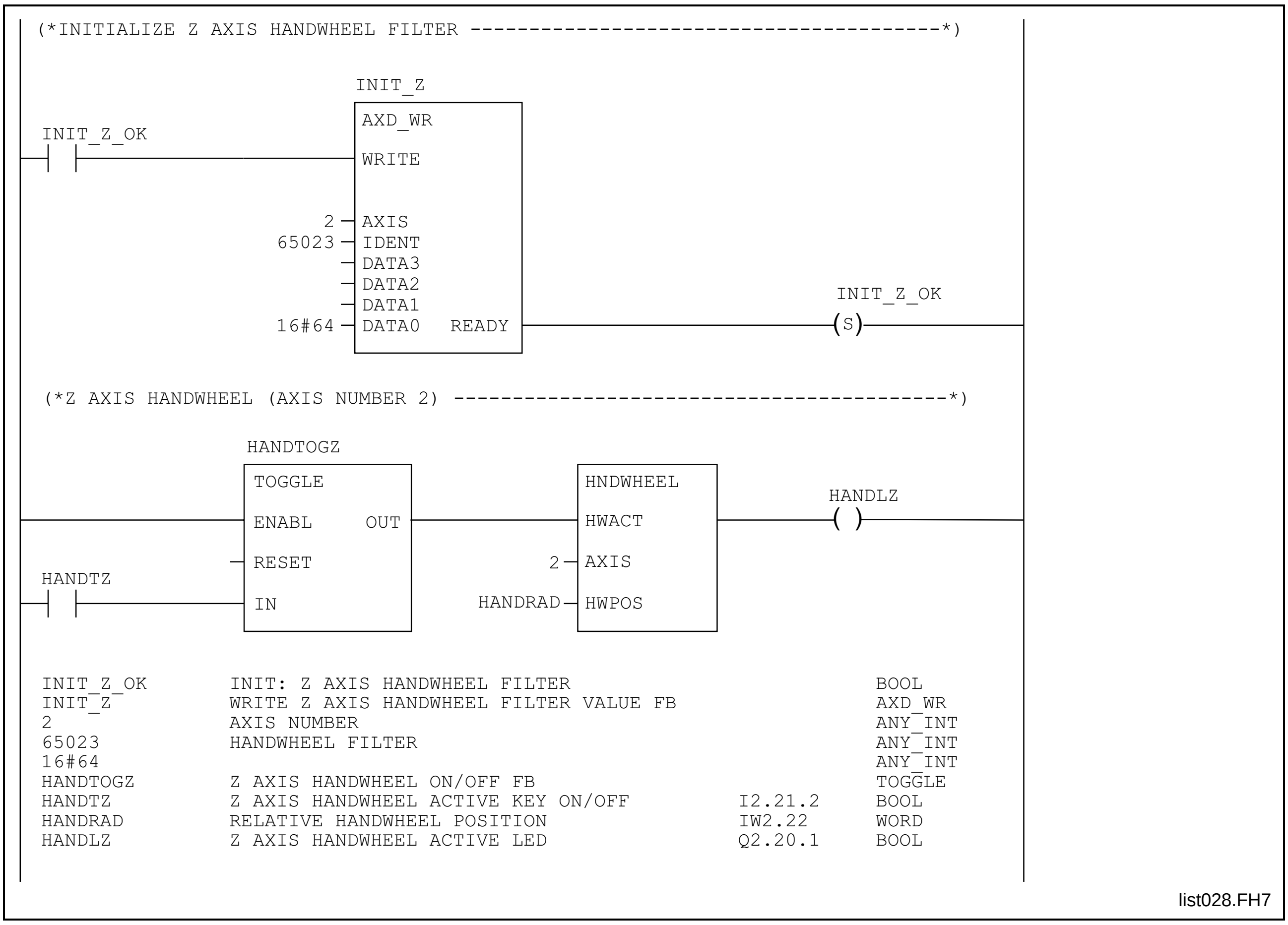


Fig. 2-82: Initialization and operation of the handwheel (continued)

2.10 Functions for Asynchronous Turret Mode

General

A turret can move *synchronously* or *asynchronously* with NC blocks. After the tool management has been activated and the turret selected as the tool magazine type, the parameter Bxx.044 "Asynchronous turret movement" is available in the process parameters. The movement of a turret is defined in that parameter.

If synchronous turret movement has been defined in the process parameters, the "REV_SYNC" function does not influence the type of the turret movement.

When asynchronous turret movement has been defined, the "REV_SYNC" function may be activated for forcing a synchronous turret movement.

Overview: turret motion types

		"Asynchronous turret movement" parameter	
		No	Yes
Function "REV_SYNC"	not active	synchronous	asynchronous
	Active	synchronous	synchronous

Fig. 2-83: Overview of possible turret motion types

Interface of the REV_SYNC Function

Synchronous swiveling of the turret in the NC block:

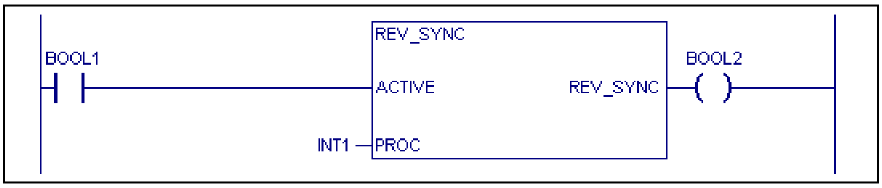


Fig. 2-84: Force synchronous turret movement

ACTIVE: 0 Turret is swiveled asynchronously
1 Turret is swiveled synchronously
PROC: Process number [0..6]

Function result

The ACTIVE input is looped through.

Programming errors prevent the "REV_SYNC" function from being executed correctly. In such a case, error handling reports the cause of the error.

Error type for the functions

REV_SYNC: -226

Error numbers:

1 - Invalid input parameter

- The value of the "PROC" input is negative.
- The PROC input is greater than 6.

2.11 Functions for Single Dimension Collision Monitoring

General

Purpose

The function "Single dimension collision monitoring" is used to prevent collisions between two linear axes which

- travel in the same or the opposite direction, and
- might crash because of their spatial closeness.

The "single dimension collision monitoring" function is for example used in machine tools where two machining slides travel on one guide rail. It can also be used for portals with two carriages operated on one guide rail.

Method of operation

The axes to be monitored can belong to different processes. When the control recognizes a crash risk between two collision monitoring axis, all feed axes are stopped in the respective processes. Spindles whose speed is regulated are stopped only if the corresponding PLC interface signal AxxC.SPSTP "Spindle stop" is set.

Parameter value assignment

Up to 8 collision monitoring groups each consisting of 2 linear axes, can be defined and parameterized in the machine data page 13 ‚Single dimension collision monitoring‘.

Activation

The single dimension collision monitoring is enabled and acknowledged via the two PLC interface functions COL_CTRL and COL_CTRL_S.

Note: For detailed documentation of this function, please refer to the functional description "Single dimension collision monitoring".

Interface of function COL_CTRL

By means of the COL_CTRL (collision control) PLC function, the collision monitoring function in a monitoring group is enabled and disabled.

Rexroth
Bosch Group

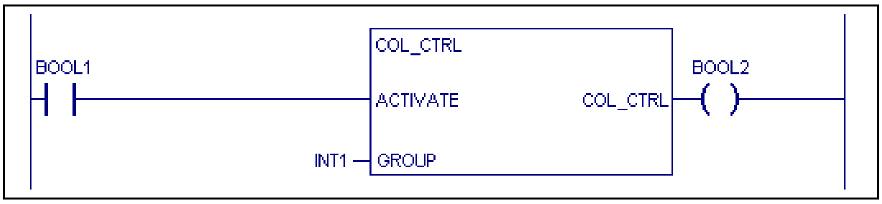


Fig. 2-85: Enable/disable single dimension collision monitoring function

ACTIVATE BOOL 0 - Disable collision monitoring in the monitoring group
1 - Enable collision monitoring in the monitoring group
GROUP: INT Monitoring group (1.0.8)

Function result COL_CTRL: BOOL The function input ACTIVATE is passed through to the output.

Interface of function COL_CTRL_S

The PLC function COL_CTRL_S (collision control set) is used for the acknowledgment dialog for enabling/disabling of the single dimension collision monitoring in a monitoring group.

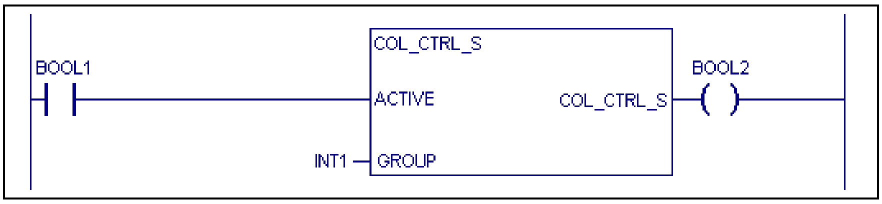


Fig. 2 -86: Acknowledgment dialog for the single dimension collision monitoring function

ACTIVE BOOL 0 - The COL_CTRL_S function is not enabled
1 - The COL_CTRL_S function is enabled
GROUP: INT Monitoring group (1.0.8)

Function result COL_CTRL_S: BOOL 0 - The COL_CTRL_S function is not enabled, or the COL_CTRL_S function is enabled, and in the NC, the collision monitoring function in the monitoring group is not (yet) enabled
1 - The acknowledgment dialog is enabled, and the NC has enabled the collision monitoring function in the monitoring group

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the PLC functions ‘COL_CTRL‘ and ‘COL_CTRL_S‘. In this case, the PLC will generate a corresponding error message.

Error type for the functions	
COL_CTRL: -211	COL_CTRL_S: -212

Fig. 2-87: Error types of functions for the single dimension collision monitoring

Error numbers:

1 - Invalid input parameter

- The value of the "GROUP" input is smaller than 1.
- The "GROUP" input is greater than 8.

2 - Internal transfer error

- general message error
- The "GROUP" input is invalid.

2.12 Enabling the NC

In the PLC program, the NC must be enabled through the "NC_ENABLE" function. Before enabling is executed, the PLC user program must have completed all steps and links required for initialization. Thus, the "enabling the NC" function should be set in the step enabling condition following the Init step. As long as the PLC has not released the NC, the NC generates the diagnosis ”Process not activated” for all parameterized processes.

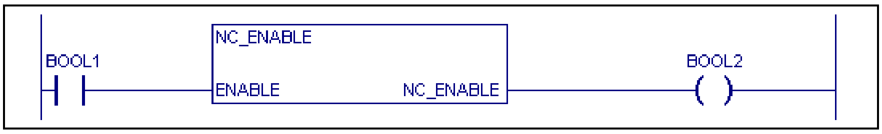


Fig. 2-88: Enable NC process

ENABLE: 0 Not enabled
1 Enabled

Function result

0 Enabling has not been executed.
1 Enabling has been executed.

When the ENABLE input variables are set to 1, the PLC function "NC_ENABLE" is called. As soon as the NC has actually been released by "NC_ENABLE", this is acknowledged by "NC_ENABLE" by setting output variable to 1.

3 Function Block Diagrams Rexroth MTC 200

3.1 Standard Function Block Diagrams

General Comments on NC Data Access Procedures

Process Data Channel

The process data channel is used for exchanging data between the NC and the PLC.

The following NC data is accessed via the process data channel:

• NC variables	NCVAR_RD, NCVAR_WR
• Tool data	TLD_RD, TLD_WR, TLBD_RD, TLBD_WR, TLED_RD, TLED_WR, TLG_RD, TLG_WR, TL_ENABLE, TL_MOVE, TL_RESET, TL_DELETE
• D corrections	DCD_RD, DCD_WR
• Zero offset	OTD_RD, OTD_WR
• Machine data	MTD_RD, MTD_WR

The data is transferred to the NC processes on a per-process basis. Exceptions are the data access processes via the MTD_RD, MTD_WR, TLBD_RD, TLBD_WR, TLED_RD, TLED_WR, TLG_RD, TLG_WR, TL_ENABLE, TL_MOVE, TL_RESET and TL_DELETE function blocks that are always handled via process 0.

The error flag is set if more than eight of the above-mentioned function blocks are used simultaneously (READ /WRITE input active). The error number 236 (process data channel overflow) is output in this case.

Required data channel

The required data channel is used for exchanging data between axis processor and drives and the PLC. The “AXD_RD” and “AXD_WR” function blocks are used for accessing the axis data.

Any number of data items can be requested at the same time. With APR or drive, data is exchanged sequentially. The speed at which SERCOS data is displayed on the user interface is reduced if the PLC utilizes the required data channel intensively (e.g. continuous reading of spindle torque value).

Function blocks for NC variables

The standard function blocks "NCVAR_RD" and "NCVAR_WR" enable NC variables to be read and written .

The variables can be transferred as

- INTEGER number,
- DOUBLE INTEGER number, or
- REAL number.

The function block interfaces are exactly defined. When a function block is invoked, the programmer "connects" the individual signals.

The time required for variable transfer results as follows:

$$\boxed{(\text{Number of variables}) * (\text{PLC cycle time}) = \text{max request time}}$$

Fig. 3-1: Time required for variablen transfer

Interfaces of the function blocks “NCVAR_RD” and “NCVAR_WR”

- Writing an NC variable – “NCVAR_WR”:

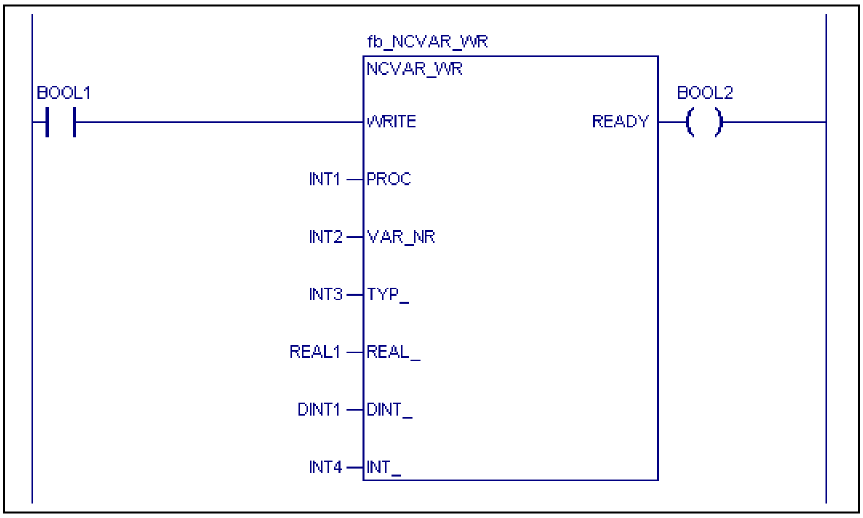


Fig. 3-2: Write NC variable

- Reading an NC variable - ‘NCVAR_RD’:

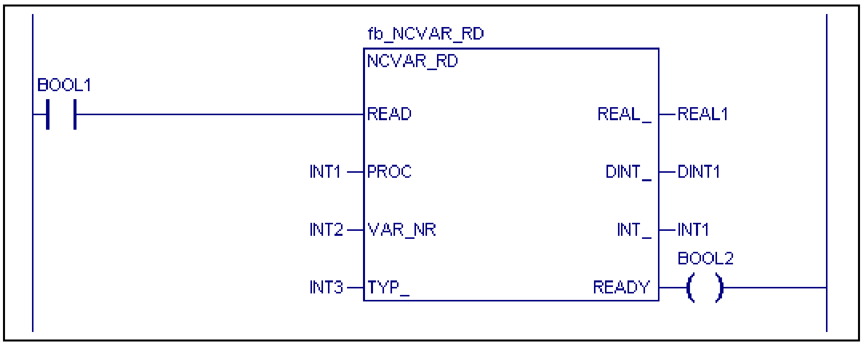


Fig. 3-3: Read NC variable

READ/WRITE:	0 FB not active 1 Initiate reading/writing NC variables
PROC:	Process number (0-6)
VAR_NR:	Variable number (0...255)
TYP_:	Data type that is to be read/written 7 - INT 9 - DINT 10 - REAL In "read NC variables", this corresponds to the FB output (INT_, DINT_, REAL_) that is to be selected.
INT_:	INTEGER data input/output
DINT_:	DOUBLE INTEGER data input/output
REAL_:	REAL data input/output
READY:	0 Data transfer active, or FB not active 1 Data was read and/or written

Method of operation of the NC variables "NCVAR_RD" and "NCVAR_WR""

Timing of writing NC variables:

```
                +-----------+      +---------------------------
WRITE       ----+           +------+

                +-----------+      +---------------------------
DATA        ----+           +------+

                           +-+                 +-+         +-+
READY       ---------------+ +-----------------+ +---------+ +--
```

(1) Setting the "WRITE" input initiates the data transfer of the NC variables. The NC variable that is to be written to is defined by the "VAR_NR" input. The "TYP_" input selects the "INT_", "DINT", or "REAL_" data input. The data must be available when the "WRITE" input is set.

(2) The active "READY" output indicates that the transfer of the NC variable is completed. The variable value is not available in the NC.

(3) In a single data exchange, the "WRITE" input may now be cleared.

(4) Clearing the "WRITE" input also clears the "READY" output of the function block.

(5) If the "WRITE" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables NC data to be written to cyclically.

Timing of reading NC variables:

```
                (1)        (2)(3)(4)                (5)
                +-----------+      +---------------------------
READ        ----+           +------+

                           +-+                 +-+         +-+
DATA        ---------------+ +-----------------+ +---------+ +--

                           +-+                 +-+         +-+
READY       ---------------+ +-----------------+ +---------+ +--
```

(1) Setting the READ input initiates the data transfer of the NC variables. The NC variable that is to be read is defined by the "NR" input. The "TYP_" output selects the "INT_", "DINT_", or "REAL_" data input.

(2) The active "READY" output indicates that the transfer of the NC variable is completed. The variable value is not available at the function block's "INT_", "DINT_", or "REAL_" output.

(3) In a single data exchange, the "READ" input may now be cleared.

(4) Clearing the "READ" input also clears the "READY" output of the function block.

(5) If the READ input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables NC data to be read cyclically.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "NCVAR_RD" and "NCVAR_WR". In such a case, error handling reports the cause of the error.

Error type of the "NCVAR_RD" and "NCVAR_WR" function blocks	
Reading NC variables	*Writing NC variables*
NCVAR_RD: -211	NCVAR_WR: -212

Fig. 3-4: Error type: NCVAR_RD, NCVAR_WR

Error numbers:

1 - Invalid input parameter

- The value of the "PROC", "VAR_NR" or "TYP_" input is negative.
- The PROC input is greater than 6.
- The PROC input is greater than 255.
- The value of the "TYP_" input is different than 7, 9 or 10

Examples of variable programming

Controlled by the state of the "IVARSTART" input, the "IWERT" input word is to be transferred to the NC in a variable. Reading the same variable ensures that the value in the NC is correct.

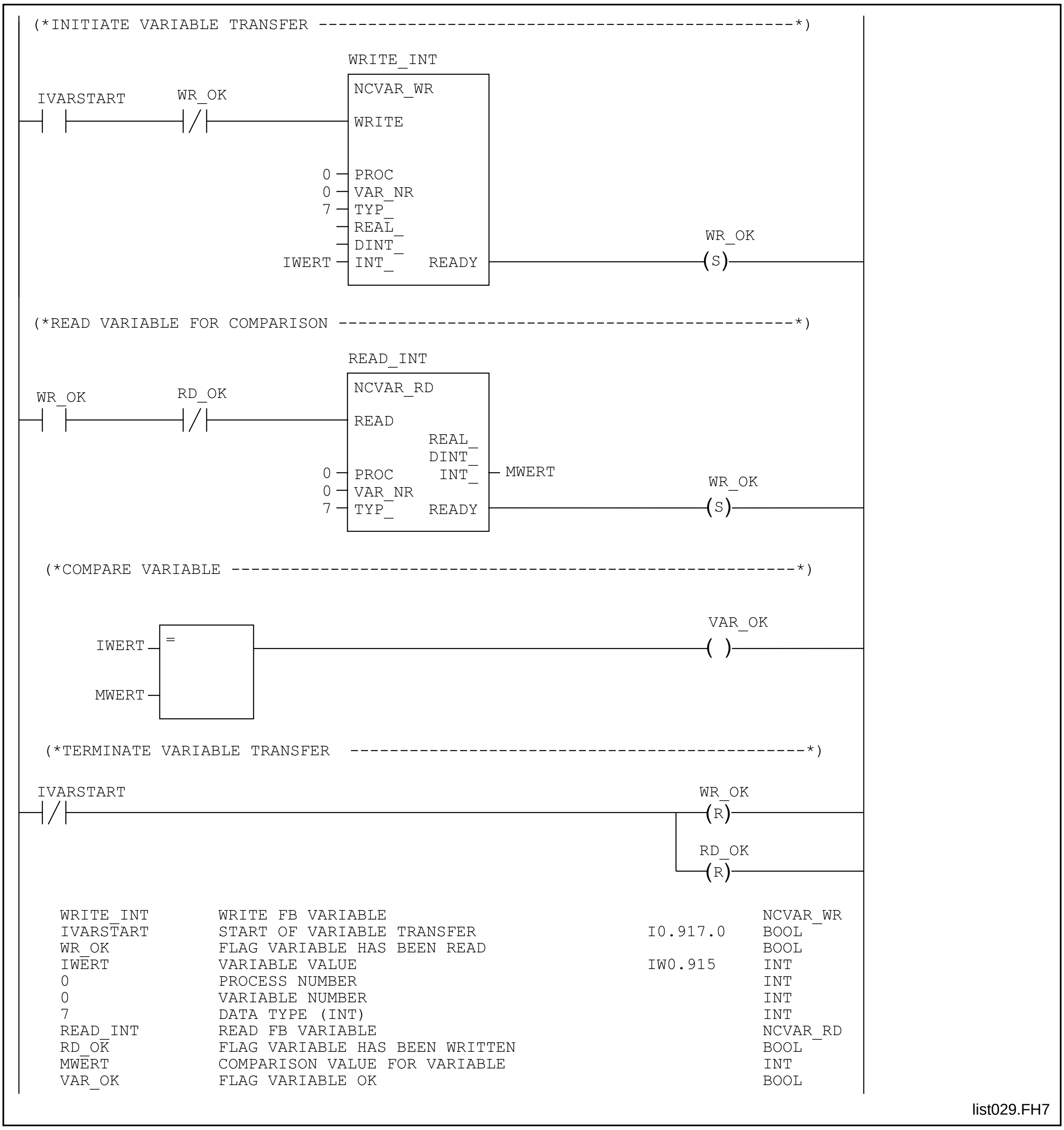


Fig. 3-5: Programming variables

Rexroth Bosch Group

The PLC employs the "M500" auxiliary function to read the current part type "PART_TYP" and to save it in the variable "50". During this time, the NC program waits for the acknowledgment of the auxiliary function. The PLC acknowledges the auxiliary function with the "READY" message:

"Variable has been written".

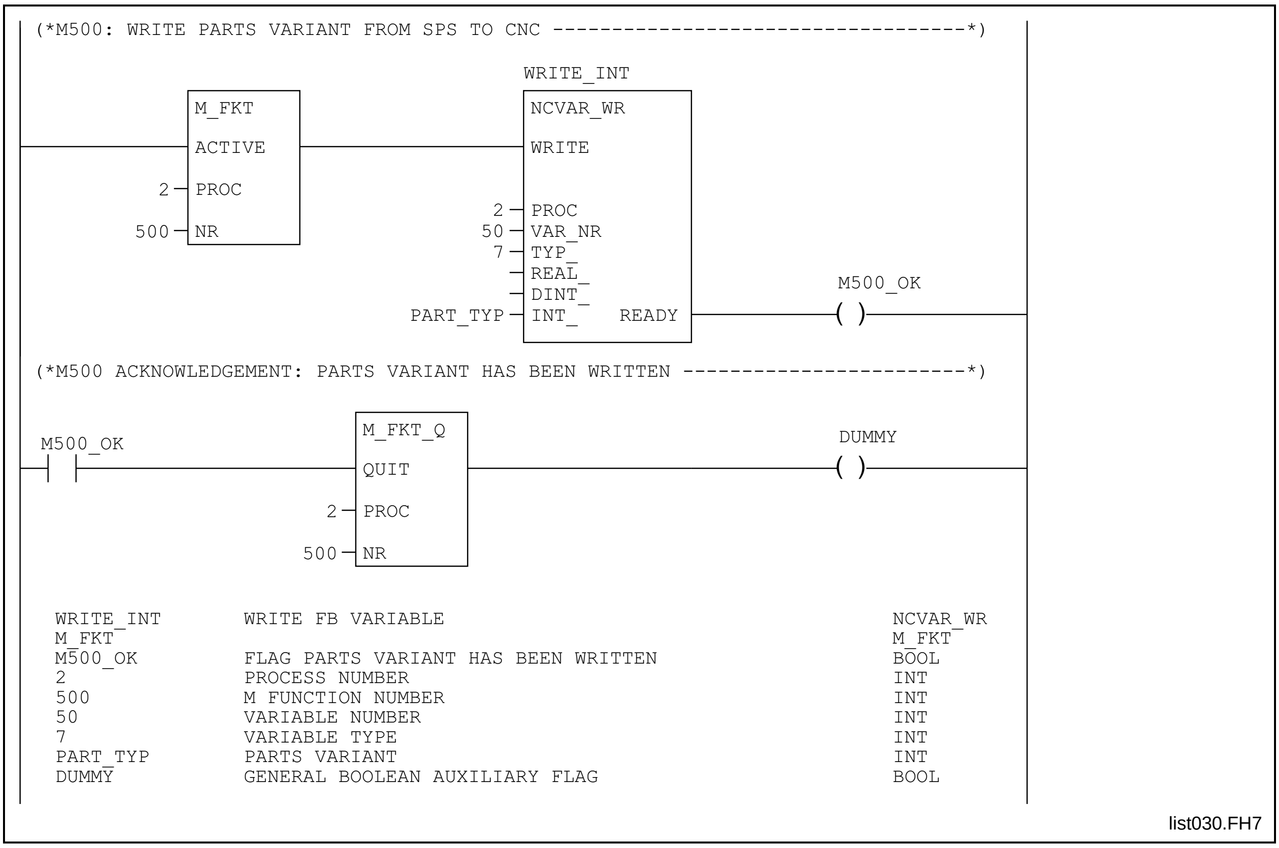


Fig. 3-6: Programming variables with auxiliary function M500

Function Blocks for NC Program Memory

The "NC program memory" can be influenced via the user interface and via the PLC.

The PLC can employ the "SEL_MEM" and "ACT_MEM" function blocks for processing the NC program memory. The "SEL_MEM" function block is used for selecting one of the two NC program memories. The "ACT_MEM" function block is used for interrogating the active NC program memory.

Selecting the NC program memory "SEL_MEM"

The "SEL_MEM" function block is used for selecting the NC program memory.

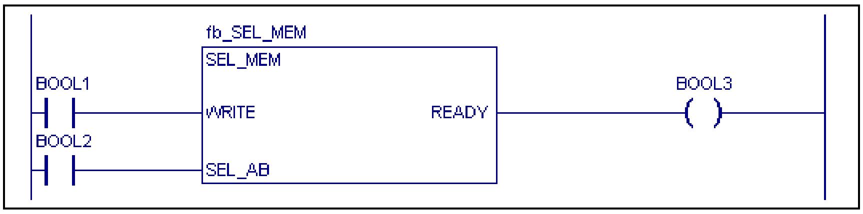


Fig. 3-7: Select NC program memory

WRITE: NC program memory at SEL_AB is transferred to NC
SEL_AB: 0 Selection of NC program memory A
1 Selection of NC program memory B

READY: 0 Selection not yet started or still active
1 NC program memory selection terminated

<u>Timing of the NC program memory selection:</u>

```
                    (1)  (2)                                              (3)  (4)
FB input               +-----------------------------------------------------┐
SEL_AB             ----+                                                     └--------
FB input                   +-------------------------------------------------┐
WRITE              --------+                                                 └--------
FB input                                                               +-----┐
READY              ----------------------------------------------------+     └-----
```

(1) The binary information for NC program memory selection is applied to the "SEL_AB" FB input. NC program memory "A" is selected if a signal "0" is applied to input "SEL_AB". NC program memory "B" is selected if a signal "1" is applied to input "SEL_AB" during writing.

(2) Activating the "WRITE" input starts the selection of the NC program memory. The signal must be applied until the feedback message from the "READY" output has been received. Writing is aborted if the "WRITE" signal is canceled during the request of the active NC program memory. The selection of the NC program memory is undefined in this case.

(3) The "READY" output is set to indicate the transfer of the NC program memory selection.

(4) The activation of the NC program memory selection can now be removed.

Example: Selecting the NC program memory

A switch on the main control panel is used for selecting the NC program memory. Selection shall always be made.

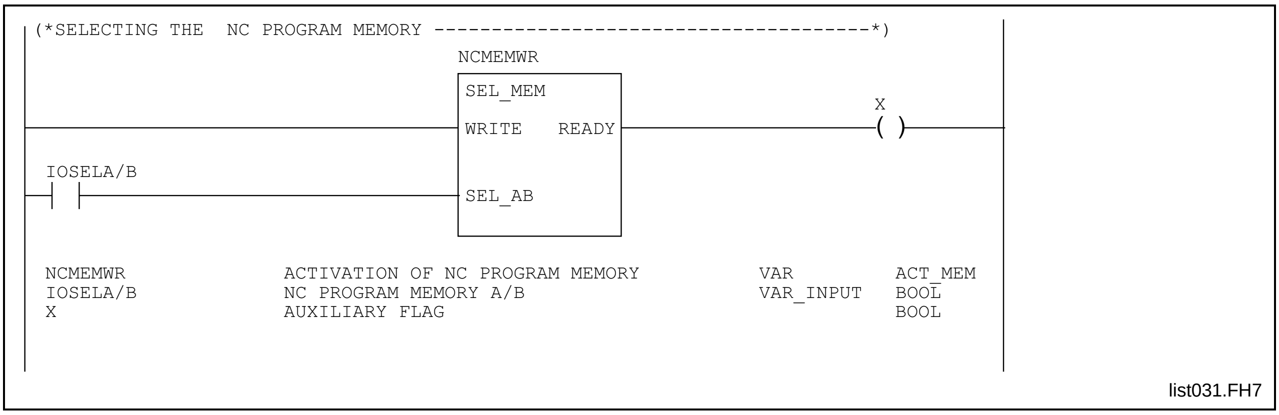


Fig. 3-8: NC program memory selection

Reading the active NC program memory "ACT_MEM""

The "ACT_MEM" function block permits the active NC program memory to be read.

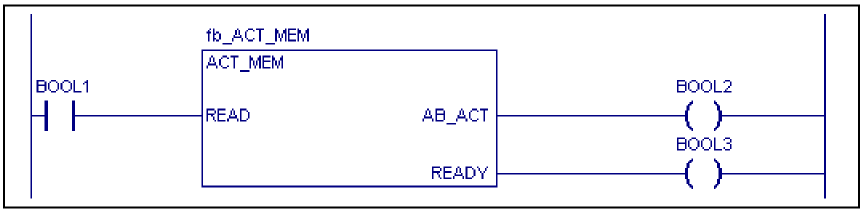


Fig. 3-9: Reading the active program memory

READ: The active NC program memory is read
AB_ACT: 0 NC program memory A is active
1 NC program memory B is active

READY: 0 The active NC program memory is requested, or FB is not active
1 The active program memory is selected.

<u>Timing of reading the active NC program memory:</u>

```
(1)                                                         (2)    (3)
FB input        +-----------------------------------------------------┐
READ        ----+                                                      └--------

FB input                                                          +-----┐
AB_ACT      ------------------------------------------------------+      └-----

FB input                                                          +-----┐
READY       ------------------------------------------------------+      └-----
```

(1) Activating the "READ" input initiates the interrogation of the active NC program memory. The signal must be applied until the feedback message from the "READY" output has been received. Reading is aborted if the "READ" signal is canceled during the request of the active NC program memory.

(2) The active NC program memory can be evaluated after the feedback from the "READY" output has been received. The "AB_ACT" output is cleared if the NC program memory "A" is active.

(3) The "AB_ACT" output is set if the NC program memory "B" is active.

(4) The interrogation of the active NC program memory can now be cleared.

Example: Interrogating the active NC program memory

A switch on the main control panel is used for selecting the NC program memory. Selection shall always be made. The active NC program memory is indicated by two lamps.

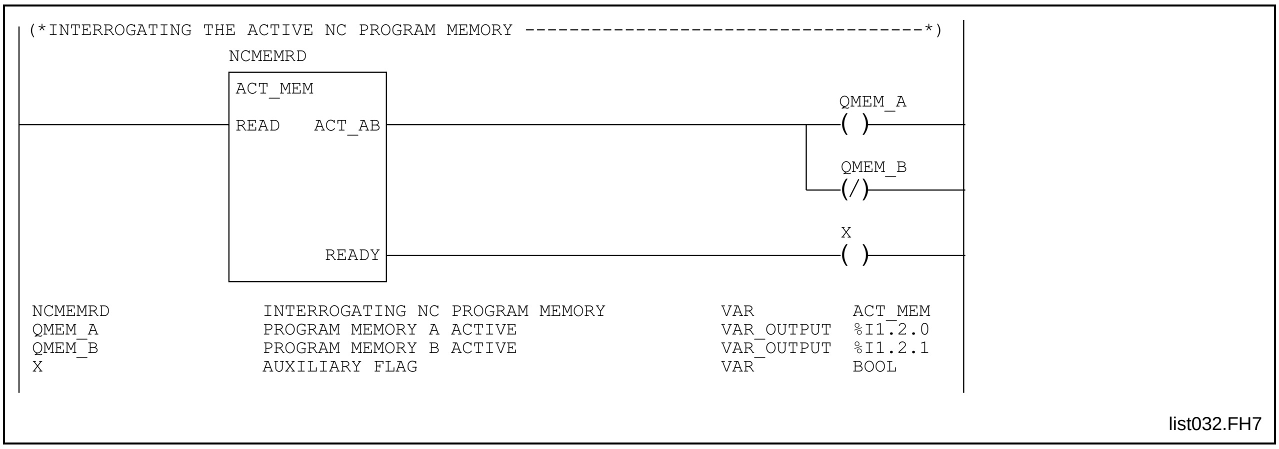


Fig. 3-10: Interrogation of the active NC program memory

Function blocks for the SERCOS required data channel

The "SERCOS required data channel" is used for the data exchange between the Rexroth MTC 200 and the digital drives.

The SERCOS required data channel is used by the user interface, the NC, and the PLC.

Access priority is of the following structure:

- highest priority - axis processor and CPU - PLC
- lowest priority: - operator interface

The standard function blocks "AXD_RD" and "AXD_WR" in the PLC program permit data in the digital drive to be read or written.

Application- or drive-related functions can manually be triggered via the user interface, or initiation can be program-controlled by the NC program or the PLC. A modification is only possible if an online change of this operating data is supported.

Note: For more information, see the parameter documentation of the drives.

The function block interfaces are exactly defined. When a function block is invoked, the programmer merely "connects" the individual signals.

Interfaces of the function blocks "AXD_RD" and "AXD_WR"

- Writing drive data "AXD_WR":

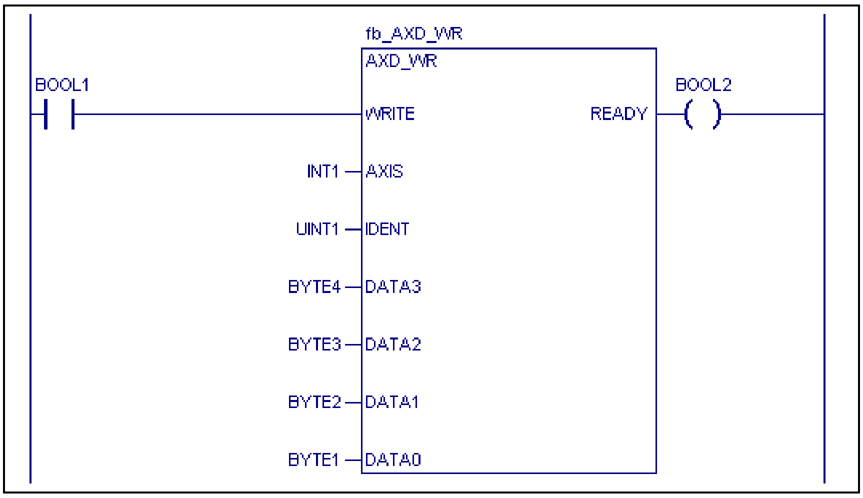


Fig. 3-11: Write drive data

After each change, the parameters that are contained in the list of operating data that is to be saved (ident number S-0-0192) are automatically transferred via the required data channel from the temporary memory (RAM) to the drive's retentive EPROM.

If a parameter is cyclically changed in the PLC program or through an "AXD" command in the NC program, the life span of 10000 write cycles that is specified for an EPROM can quickly be reached. Writing data to these memory cells would then no longer be possible.

To counteract this behavior, there is a storage mode parameter with the ident number S-0-00269. The storage mode parameter must be changed to "1" if a parameter in the above-mentioned list is changed cyclically. The changed data will then only be stored temporarily (i.e. not each time in the EPROM). The temporary data will be lost when the unit is switched off.

When the unit is switched on, the retentive initial values will be written to the temporary memory, and storage mode will be initialized with "0".

Note: In the PLC program, two AXD functions must never be active at the same time.

To save the EPROM's service life, storage mode must be set to "1" before one of the parameters from the S-0-0192 list is cyclically changed via the required data channel.

- Reading drive data "AXD_RD":

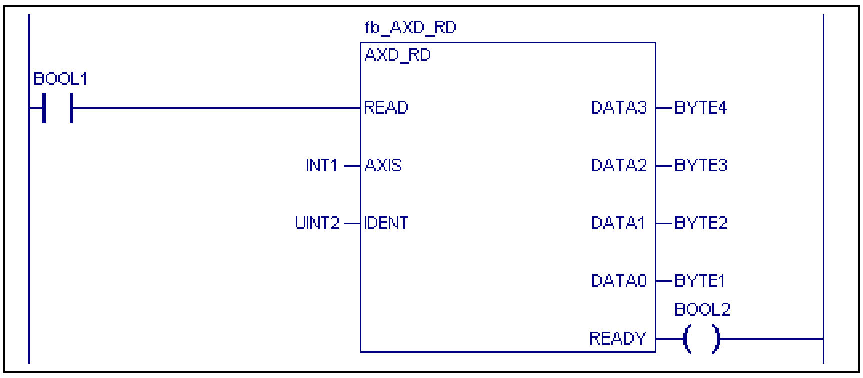


Fig. 3-12: Read drive data

READ/WRITE: 0 FB not active

	1 * Initiate reading/writing required data
DATA0…3:	Data (DATA0 - low byte; DATA3 - high byte)
AXIS:	Axis number (1...32)
IDENT:	ID number
READY:	0 Data transfer active, or FB not active
	1 Data was read and/or written

Calculating the Ident number

The PLC specifies the ident number as an UNSIGNED INTEGER number. This requires a conversion of the SERCOS parameters to be performed.

```
15    12 11     |              0
 X X X X X X X X|X X X X X X X X
 | +---+ +---------------------+
 |    |             |
 |    |             +----- Parameter number 0...4095
 |    +------------------- Parameter record 0...7
 +------------------------ 0 - Standard parameter (S)
                           1 - Product parameter (P)
```

Example

Conversion of SERCOS ID numbers → ID numbers

P-7-3560 → 65000

Definition of a part of the SERCOS Ident number (each separated by "-"):

P := 0x8000 (Hexadecimal) = 32768 (Decimal)

S := 0x8000 (Hexadecimal) = 0 (Decimal)

7: = 0x7000 (hexadecimal) = 28672 (decimal)

P-7 := 0x8000 + 0x7000 = 0xF000 := 32768 + 28672 = 61440

Conversion of SERCOS Ident numbers -> Ident numbers

P-7-nnnn := 61440 + nnnn

Method of operation

Timing of writing drive data:

```
                1          234                 5
                +-----------+    +---------------------------
WRITE       ----+            +-----+

                +-----------+    +---------------------------
DATA        ----+            +-----+

                           +-+              +-+         +-+
READY       ---------------+ +---------------+ +----------+ +--
```

(1) Setting the "WRITE" input initiates the transfer of the drive data. The "IDENT" input addresses the drive data that is to be written to. The "DATA0 ... DATA3" data must be available when the "WRITE" input is set.

(2) The active "READY" output indicates that the transfer of the drive data is completed. The data is now available in the digital drive.

(3) In a single data exchange, the "WRITE" input may now be cleared.

(4) Clearing the "WRITE" input also clears the "READY" output of the function block.

(5) If the "WRITE" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables drive data to be written to cyclically.

Timing of reading drive data:

```
                1          234                 5
                +------------+    +----------------------------
READ        ----+            +----+

                           +-+                +-+         +-+
DATA        ---------------+ +----------------+ +---------+ +--

                           +-+                +-+         +-+
READY       ---------------+ +----------------+ +---------+ +--
```

(1) Setting the "READ" input initiates the transfer of the drive data. The "IDENT" input addresses the drive data that is to be read.

(2) The active "READY" output indicates that the transfer of the drive data is completed. The data is now available at the function block's "DATA0 ... DATA3" output.

(3) In a single data exchange, the "READ" input may now be cleared.

(4) Clearing the "READ" input also clears the "READY" output of the function block.

(5) If the READ input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables drive data to be read cyclically.

Note: Cyclic utilization of the required data channel in the PLC program significantly slows down the access from the user interface to the drive data. This delays the screen updating.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "AXD_RD" and "AXD_WR". In such a case, error handling reports the cause of the error.

Error type of the function blocks	
Reading drive data	*Writing drive data*
AXD_RD: 133	AXD_WR: 132

Fig. 3-13: Error type of the function blocks AXD_RD and AXD_WR

Error numbers:

1 - Invalid input parameter

- The value of the "AXIS" or "IDENT" input is negative.
- The "AXIS" input is greater than 20.

List of the APR-SERCOS parameters

The SERCOS ident number range from 65000 (P-7-3560) onwards is reserved for parameters on the "APR". These parameters can only be addressed through the "AXD" command or via the PLC required data channel.

Required data requests in this ident number range are processed directly by the "ARP". The SERCOS required data channel to the digital drives is not affected. This means that required data requests from user interface or from NC for special functions (e.g. homing) can run simultaneously.

The following parameters are currently available:

The access options (read, write, read and write) of the individual parameters are specified. The drives which support the parameter are listed too.

Ident number	SERCOS Ident number	MTC 200	TRANS 200	Description
65005	P-7-3565	X	-	Actual torque value of the drive
65008	P-7-3568	X	-	Servo Drive as Spindle Drive
65009	P-7-3569	X	-	Reference angle position "DDS as main spindle"
65010	P-7-3570	X	-	Jerk Limiting
65011	P-7-3571	X	-	Spindle function parameter
65012	P-7-3572	X	-	Spindle Function During Gear Change
65013	P-7-3573	X	-	Set Spindle Speed for Spindle Function
65014	P-7-3574	X	-	Spindle Acceleration
65015	P-7-3575	X	-	Spindle Angle
65016	P-7-3576	X	X	Reduced Motor Torque
65017	P-7-3577	X	X	Reduced motor torque when driving to positive stop
65018	P-7-3578	X	X	Reduced Motor Torque at Hard Stop
65022	P-7-3582	X	-	Command Filter Command Position Value
65023	P-7-3583	X	-	Filter Jerk Limiting
65024	P-7-3584	X	-	Filter Acceleration Override
65025	P-7-3585	X	-	Polygon Mode
65026	P-7-3586	X	-	Triggering Oscilloscope Function
65027	P-7-3587	X	-	Virtual Axis
65029	P-7-3589	X	-	Offset angle of a slave spindle
65030	P-7-3590	X	-	Switching Torque of Position Control in Velocity Loop
65031	P-7-3591	X	X	Static monitoring window for control-internal drive monitoring of the drives
65032	P-7-3592	X	-	Moving the reference cam of a combined spindle / turret axis
65033	P-7-3593	X	-	Activation of the Simulation Mode
65034	P-7-3594	X	-	Switch on/off Droop Compensation
65035	P-7-3595	X	-	Activate Measuring Function
65036	P-7-3596	X	-	Measured Torque Value
65037	P-7-3597	X	-	Maximum Torque Peak for Tool Breakage Monitoring
65038	P-7-3598	X	-	Maximum Measured Torque Peak
65040	P-7-3600	X	-	Measured Torque Distribution
65041	P-7-3601	X	-	Measured Maximum Lag Error
65042	P-7-3602	X	-	P factor for spindle synchronization compensation controller
65043	P-7-3603	X	-	Number of Turret Locations
65044	P-7-3604	X	-	Homing of the Spindle Necessary
65045	P-7-3605	X	-	Actual axis position
65046	P-7-3606	X	-	Current position of 2nd encoder system
65047	P-7-3607	X	X	Active Encoder System
65048	P-7-3608	X	X	Positive Limit of Dynamic Limit Surveillance

Ident number	SERCOS Ident number	MTC 200	TRANS 200	Description
65049	P-7-3609	X	X	Negative Limit of Dynamic Limit Surveillance
65050	P-7-3610	X	X	Filter Time Constant for 2nd Encoder System
65051	P-7-3611	X	-	Axis command position
65052	P-7-3612	X	-	Maximum spindle speed
65053	P-7-3613	X	-	Coupling Type for Axis Coupling on APR, Interpolator 1
65054	P-7-3614	X	-	Length of the APR Table of Values 1 (Interpolator 1)
65055	P-7-3615	X	-	Transfer of Table Values for Table 1 (Interpolator 1)
65056	P-7-3616	X	-	Tables Factor for Interpolator 1
65057	P-7-3617	X	-	Table 1 Start Position
65058	P-7-3618	X	-	Support Point Distances of Table 1 (Interpolator 1)
65059	P-7-3619	X	-	Multi-Function Interpolator Mode (Coupling Type 2 Only) Interpolator 1
65060	P-7-3620	X	-	Number of the Reference Axis (Coupling Types 2 or 4) of the Interpolator 1
65061	P-7-3621	X	-	Activate multi-function interpolator axis coupling
65062	P-7-3622	X	-	Activating the Encoder of a Digital Drive
65063	P-7-3623	X	-	Switching off Speed Reduction when Using the Transmit Function or Triaglide/Rod Kinematics
65064	P-7-3624	X	-	Adaptation of third-party SERCOS drives
65065	P-7-3625	X	X	Sensitivity of Dynamic Limit Monitoring
65066	P-7-3626	X	-	Additional Reference Offset of the Feed Axis with Active Transmit Function
65067	P-7-3627	X	-	Angular deviation of the synchronous spindle from the master spindle with active spindle synchronization
65068	P-7-3628	X	X	SERCOS Test Mode
65069	P-7-3629	X	X	SERCOS drive telegram error counter. Number of drive telegrams
65078	P-7-3638	X	X	Length of the SERCOS Fiber Optic
65080	P-7-3640	X	-	Actual spindle speed
65081	P-7-3641	X	-	Position Window for Block Enabling in Polygon Mode 4
65082	P-7-3642	X	-	Activate Parking Axis
65084	P-7-3644	X	-	Definition of Forced Dynamization
65085	P-7-3645	X	-	Acknowledgment Forced Dynamization
65086	P-7-3646	X	-	Switching speed for switching to curved acceleration characteristic of a spindle in position control
65087	P-7-3647	X	-	Safe Reference
65088	P-7-3648	X	-	Number of Data Points for 3D Compensation
65089	P-7-3649	X	X	Measuring Period for Idling Torque Measuring
65090	P-7-3650	X	X	Measured Idling Torque
65091	P-7-3651	X	X	Measuring Period for Standstill Torque Measuring
65092	P-7-3652	X	X	Measured Standstill Torque
65093	P-7-3653	X	-	Coupling Type for Axis Coupling on APR, Table 2 of Multi-Function Interpolator
65094	P-7-3654	X	-	Length of APR Values Table (Interpolator 2)

Ident number	SERCOS Ident number	MTC 200	TRANS 200	Description
65095	P-7-3655	X	-	Transfer of Table Values for Table 2 (Interpolator 2)
65096	P-7-3656	X	-	Tables Factor for Interpolator 2
65097	P-7-3657	X	-	Table 2 Start Position
65098	P-7-3658	X	-	Support Point Distances of Table 2 (Interpolator 2)
65099	P-7-3659	X	-	Multi-Function Interpolator Mode (Coupling Type 2 Only) Interpolator 2
65100	P-7-3660	X	-	Reference Axis Number (Coupling Type 2) of Interpolator 2
65101	P-7-3661	X	-	Coupling Type for Axis Coupling on APR, Table 3 of Multi-Function Interpolator
65102	P-7-3662	X	-	Length of APR Values Table (Interpolator 3)
65103	P-7-3663	X	-	Transfer of Table Values for Table 3 (Interpolator 3)
65104	P-7-3664	X	-	Tables Factor for Interpolator 3
65105	P-7-3665	X	-	Table 3 Start Position
65106	P-7-3666	X	-	Support Point Distances of Table 3 (Interpolator 3)
65107	P-7-3667	X	-	Multi-Function Interpolator Mode (Coupling Type 2 Only) Interpolator 3
65108	P-7-3668	X	-	Reference Axis Number (Coupling Type 2) of Interpolator 3
65109	P-7-3669	X	-	Coupling Type for Axis Coupling on APR, Table 4 of Multi-Function Interpolator
65110	P-7-3670	X	-	Length of APR Values Table (Interpolator 4)
65111	P-7-3671	X	-	Transfer of Table Values for Table 4 (Interpolator 4)
65112	P-7-3672	X	-	Tables Factor for Interpolator 4
65113	P-7-3673	X	-	Table 4 Start Position
65114	P-7-3674	X	-	Support Point Distances of Table 4 (Interpolator 4)
65115	P-7-3675	X	-	Multi-Function Interpolator Mode (Coupling Type 2 Only) Interpolator 4
65116	P-7-3676	X	-	Reference Axis Number (Coupling Type 2) of Interpolator 4
65118	P-7-3678	X	X	Switching Between Position Contacts and Waypoints
65119	P-7-3679	X	X	Preselection of the Position Contact Data to be Transferred
65120	P-7-3680	X	X	Transferring Position Contact Data
65121	P-7-3681	X	X	Transferring Position Contact Data
65122	P-7-3682	X	X	Transferring Position Contact Data
65123	P-7-3683	X	X	Transferring Position Contact Data
65124	P-7-3684	X	X	Transferring Position Contact Data
65125	P-7-3685	X	X	Transferring Position Contact Data
65126	P-7-3686	X	X	Transferring Position Contact Data
65127	P-7-3687	X	X	Transferring Position Contact Data

Fig. 3-14: List of the APR-SERCOS parameters

Access to drive data - examples

Controlled by the state of the "I_RDMOTLD" input, the current load of the drive is to be read and saved in the "MLOAD" variable.

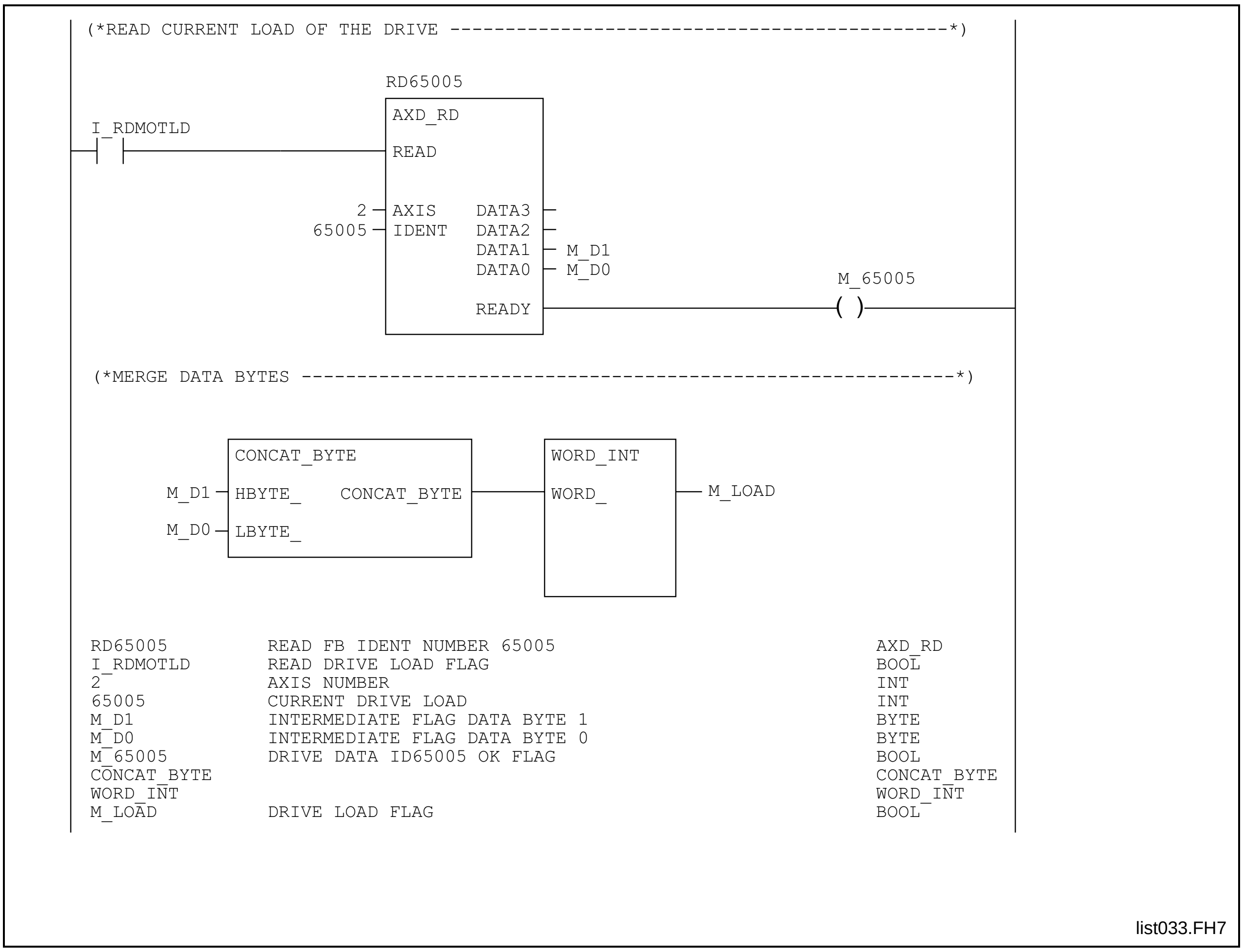


Fig. 3-15: Access to drive data

Time required for a transfer

Once data transfer has been activated, up to 60 ms may elapse until the data is available in the PLC.

Note: The required data channel is also needed for transferring drive data to the user interface and to the NC. The PLC must therefore not obstruct that channel by cyclic data transfer.

Please refer to the individual application descriptions for a specification of the data that may expediently be influenced via the PLC.

Function Blocks for Accessing Machine Data

The "MTD_RD" function block enables the PLC program to read machine data. Provided that the PLC program has enabled the access, the "MTD_WR" function block can be used for writing to the machine data. The machine data is kept in the NC. The PLC must therefore access this information as required, i.e. serially. Thus, the machine data will be available after one PLC cycle at the earliest.

The function block interfaces are exactly defined. When a function block is invoked, the programmer merely "connects" the individual signals.

Interfaces of the function blocks "TD_RD" and "MTD_WR""

- Reading machine data "MTD_RD":

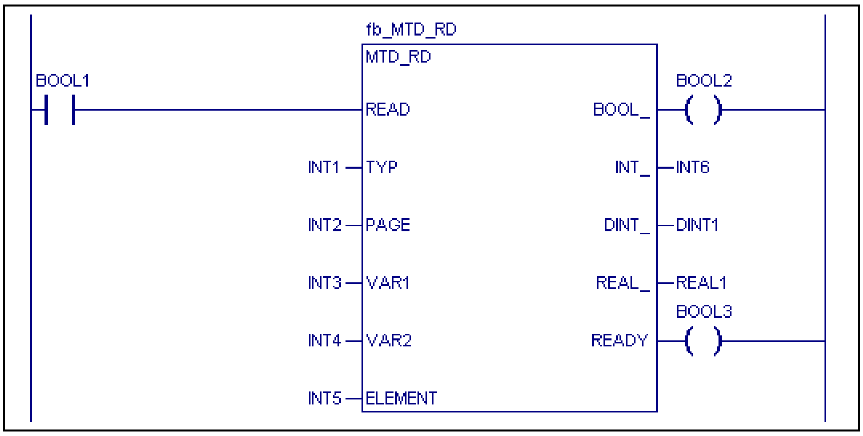


Fig. 3-16: Read machine data

- Writing machine data "MTD_WR":

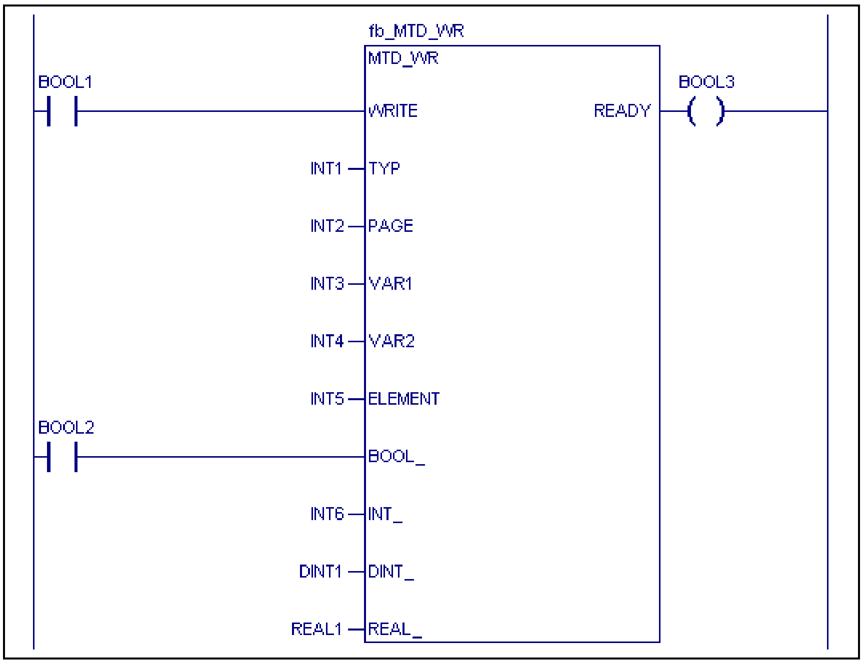


Fig. 3-17: Write machine data

READ:	0 FB not active 1 Initiate the reading of machine data
WRITE:	0 FB not active 1 Initiate the writing of machine data

TYPE:	Data type that is to be read/written 0 - BOOL 7 - INT 9 - DINT 10 - REAL
PAGE:	Page number (1-299)
VAR1:	Control variable 1 (min. value - max. value according to page definition)
VAR2:	Control variable 2 (min. value - max. value according to page definition)
ELEMENT:	Data element (1-110)
BOOL_:	Machine data that is to be read or to be written
INT_:	Machine data that is to be read or to be written
DINT_:	Machine data that is to be read or to be written
REAL_:	Machine data that is to be read or to be written
READY:	When reading machine data 0 - Data invalid 1 - Data valid When writing machine data 0 - Data transfer active or FM not active 1 - Data transferred

Method of operation

Timing of Writing Machine Data:

```
                1          234                5
                +-----------+     +----------------------------
WRITE       ----+           +-----+
                +-----------+     +----------------------------
M data      ----+           +-----+
                           +-+               +-+          +-+
READY       ---------------+ +---------------+ +----------+ +--
```

(1) Setting the "WRITE" input initiates the transfer of the machine data. The machine data that is to be accessed is defined by the 'PAGE', 'VAR1', 'VAR2' and 'ELEMENT' inputs. The "TYP" input defines the type of the machine data. Automatic conversion is performed if the specified data does not exist in the specified format.

(2) The active "READY" output indicates that the transfer of the machine data is completed. In a write process, the "READY" output shows that data transfer has been completed.

(3) In a single data exchange, the "WRITE" input may now be cleared.

(4) Clearing the "WRITE" input also clears the "READY" output of the function block.

(5) If the "WRITE" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables tool data to be written cyclically.

Timing of Reading Machine Data:

```
                1          234                5
                +-----------+     +----------------------------
READ        ----+           +-----+
                           +-+               +-+          +-+
M data      ---------------+ +---------------+ +----------+ +--
                           +-+               +-+          +-+
READY       ---------------+ +---------------+ +----------+ +--
```

(1) Setting the "READ" input initiates the transfer of the machine data. The machine data that is to be accessed is defined by the 'PAGE', 'VAR1', 'VAR2' and 'ELEMENT' inputs. The "TYP" input defines the type of the machine data. Automatic conversion is performed if the specified data does not exist in the specified format.

(2) The active "READY" output indicates that the transfer of the machine data is completed. In a reading process, the 'READY' output shows that the requested data are available.

(3) In a single data exchange, the "READ" input may now be cleared.

(4) Clearing the "READ" input also clears the "READY" output of the function block.

(5) If the "READ" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables machine data to be read cyclically.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "MTD_RD" and "MTD_WR". In such a case, error handling reports the cause of the error.

Error type of the function blocks	
Reading machine data	*Writing machine data*
MTD_RD: 178	MTD_RD: 177

Fig. 3-18: Error type "Reading and writing of machine data"

Error numbers:

1 - Invalid input parameter

- The value of the "PAGE", "VAR1", "VAR2" or "ELEMENT" input is negative.
- The value of the "PAGE" input is outside the range 1 ... 299.
- The value of the "VAR1" input is outside the range -1000 ... 1000.
- The value of the "VAR2" input is outside the range -1000 ... 1000.
- The value of the "ELEMENT" input is outside the range 1 ... 110.
- The "TYP_" input is different than 0, 7, 9 or 10.

6 - Internal transfer error

- CPU reports error (e.g. machine data cannot be found; data is write-protected)

Function Blocks for Accessing Zero Point Data

The "OTD_RD" function block permits zero point data to be read from the PLC program. The "OTD_WR" function block writes to the zero point data. The zero point data is kept in the NC. The PLC must therefore access this information as required, i.e. serially. Thus, the zero point data will be available after one PLC cycle at the earliest.

The function block interfaces are exactly defined. When a function block is invoked, the programmer merely "connects" the individual signals.

Interfaces of the function blocks for the access to zero point data

- Reading zero point data "OTD_RD":

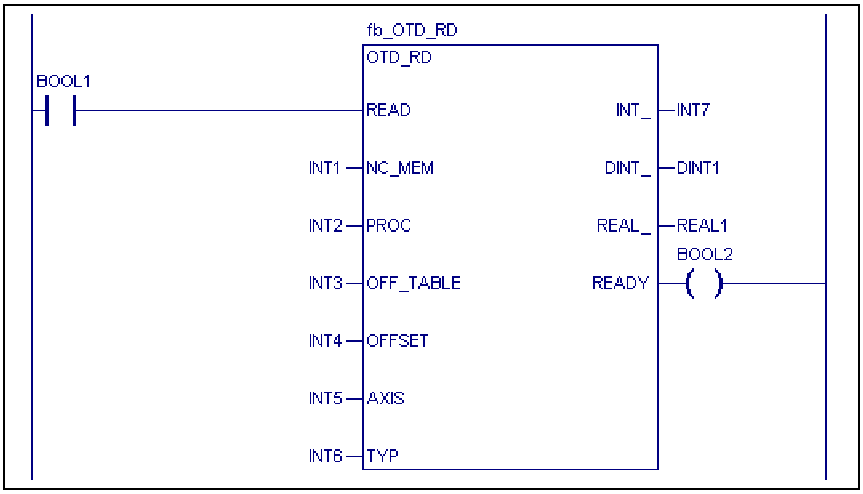


Fig. 3-19: Read zero offset data

- Writing zero offset data "OTD_WR":

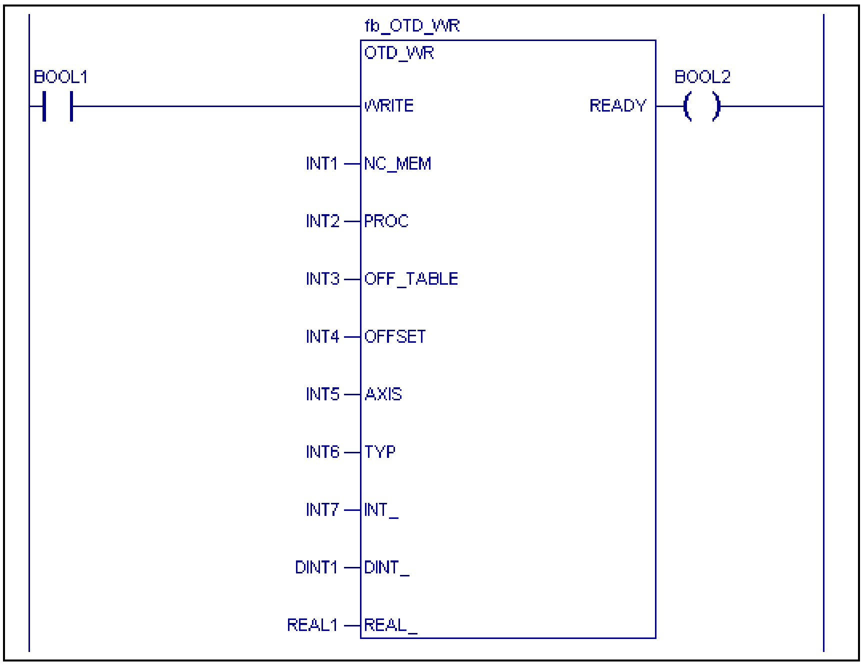


Fig. 3-20: Write zero offset data

READ: 0 FB not active
1 Initiate the reading of zero point data

WRITE:	0 FB not active 1 Initiate the writing of zero point data
NC_MEM:	NC memory 0 - NC memory A 2 - NC memory B -1 - Current NC memory
PROC:	Process number (0-6)
OFF_TABLE:	Zero offset table (0..9) 1 - Current zero offset table
OFFSET:	Zero offset 0 - Active offset 1 - Offset value according to G50/G51 2 - Offset value according to G52 3 - General offset 4 - Offset G54 5 - Offset G55 6 - Offset G56 7 - Offset G57 8 - Offset G58 9 - Offset G59
AXIS:	Axis denomination 1 - Value of the X axis (according to axis meaning) 2 - Value of the Y axis (according to axis meaning) 3 - Value of the Z axis (according to axis meaning) 4 - Value of the U axis (according to axis meaning) 5 - Value of the V axis (according to axis meaning) 6 - Value of the W axis (according to axis meaning) 7 - Value of the A axis (according to axis meaning) 8 - Value of the B axis (according to axis meaning) 9 - Value of the C axis (according to axis meaning) 10 - Value of the rotation (angle F)
TYPE:	Data type that is to be read/written 7 - INT 9 - DINT 10 - REAL
INT_:	Zero point data
DINT_:	Zero point data
REAL_:	Zero point data
READY:	When reading zero point data 0 - Data invalid 1 - Data valid When writing zero point data 0 - Data transfer active or FM not active 1 - Data transferred

The "OTD_WR" function block can not be used for writing to the zero offset values for "G50/G51", "G52" and to the active zero offset value.

Method of operation

Timing of writing zero point data:

```
                1          234                5
                +-----------+    +----------------------------
WRITE       ----+           +----+
                +-----------+    +----------------------------
NP data     ----+           +----+
                           +-+               +-+         +-+
READY       ---------------+ +---------------+ +---------+ +--
```

(1) Setting the "WRITE" input initiates the transfer of the zero point offset data. The inputs "NC_MEM", "PROC", "OFF_TABLE", "OFFSET" and "AXIS" define the zero point data that is to be accessed. The "TYP" input defines the type of the zero point data. Automatic conversion is performed if the specified data does not exist in the specified format.

(2) The active "READY" output indicates that the transfer of the zero point data data is completed. In a write process, the "READY" output shows that data transfer has been completed.

(3) In a single data exchange, the "WRITE" input may now be cleared.

(4) Clearing the "WRITE" input also clears the "READY" output of the function block.

(5) If the "WRITE" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed.

Timing of reading zero point data:

```
                1          234                5
                +-----------+    +----------------------------
READ        ----+           +----+
                           +-+               +-+         +-+
ZP data     ---------------+ +---------------+ +---------+ +--
                           +-+               +-+         +-+
READY       ---------------+ +---------------+ +---------+ +--
```

(1) Setting the "READ" input initiates the transfer of the zero point data. The inputs "NC_MEM", "PROC", "OFF_TABLE", "OFFSET" and "AXIS" define the zero point data that is to be accessed. The "TYP" input defines the type of the zero point data. Automatic conversion is performed if the specified data does not exist in the specified format.

(2) The active "READY" output indicates that the transfer of the zero point data data is completed. In a reading process, the 'READY' output shows that the requested data are available.

(3) In a single data exchange, the "READ" input may now be cleared.

(4) Clearing the "READ" input also clears the "READY" output of the function block.

(5) If the "READ" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "OTD_RD" and "OTD_WR". In such a case, error handling reports the cause of the error.

Error type of the function blocks	
Reading zero offset data	*Writing zero offset data*
OTD_RD: 176	OTD_WR: 175

Fig. 3-21: Error type "Reading and writing of zero point data"

Error numbers:

1 - Invalid input parameter

- The value of the "PROC" and "AXIS" input is negative.
- The value of the "NC_MEM" input is 0 or smaller than -1.
- The value of the "OFF_TABLE" or "OFFSET" input is smaller than -1.
- The PROC input is greater than 6.
- The value of the "OFF_TABLE" input is greater than 9.
- The "OFFSET" input is greater than 9.
- The value of the "TYP_" input is different than 7, 9 or 10

6 - Internal transfer error

- CPU reports error (e.g. machine data cannot be found; data is not in zero point offset table)

Function blocks for serial interfaces

OPEN_COM; Initializing a general data channel

CLOS_COM: Terminating the data transfer of a general data channel

WR_BYTE: Writing a BYTE on the general transmission channel

RD_BYTE: Reading a BYTE on a general transfer channel

CTRL_COM: Interrogating the status of a serial interface

Note: Refer to documentation "PLC operations set".

Function blocks for the graphical user interface

The "GUI_SK16" Bosch Rexroth standard function block permits the graphical user interface to be linked with the PLC. Via the "GUI_SK16" function block, the graphical user interface influences – through the configured M keys – user-defined flags or outputs that are related to the selected screen.

The function block interface is exactly defined. When a function block is invoked, the programmer "connects" the individual signals.

Transferring softkeys to the graphical user interface "GUI_SK16"

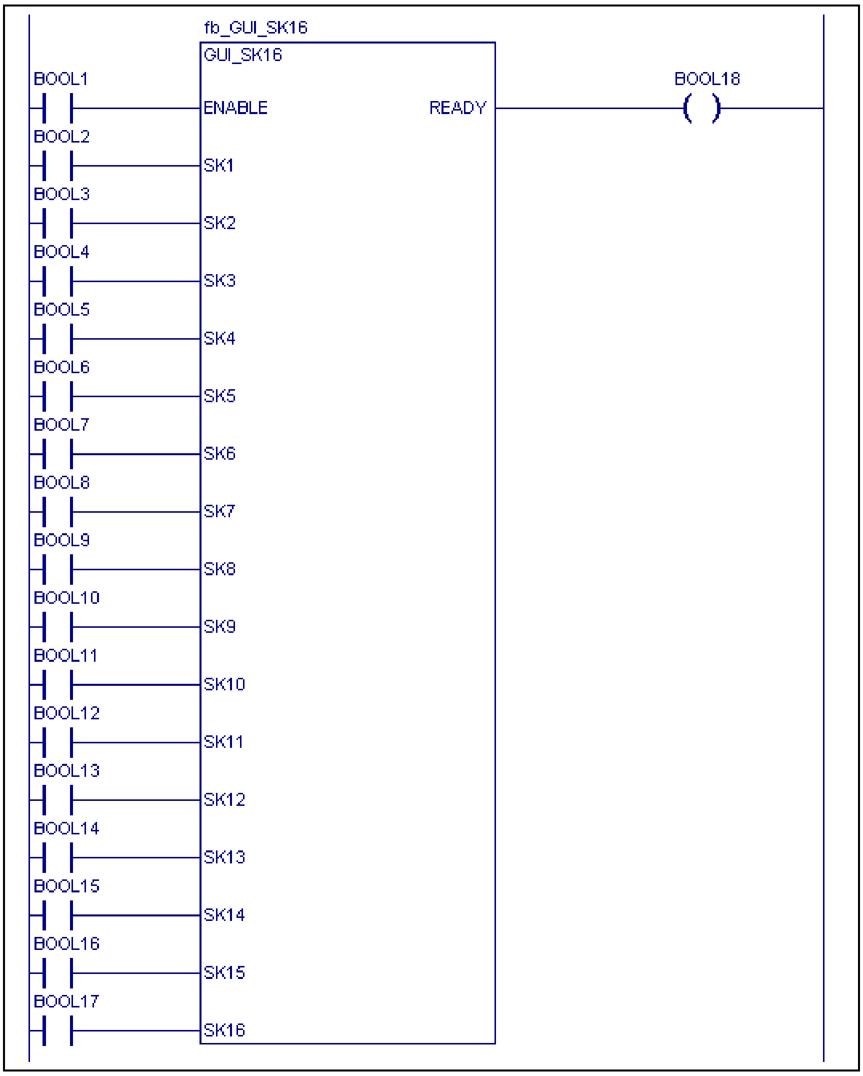


Fig. 3-22: Softkeys transmission to the user interface

ENABLE: 0 No function block processing; i.e. outputs and/or flags are not influenced.
1 Function blocks are processed; i.e. outputs and/or flags assume the states of the softkeys.

SK1:	Softkey 1
SK2:	Softkey 2
SK3:	Softkey 3
SK4:	Softkey 4
SK5:	Softkey 5
SK6:	Softkey 6
SK7:	Softkey 7
SK8:	Softkey 8
SK9:	Softkey 9
SK10:	Softkey 10
SK11:	Softkey 11
SK12:	Softkey 12
SK1:	Softkey 13
SK1:	Softkey 14
SK1:	Softkey 15
SK1:	Softkey 16

Skxx =0: Outputs and/or flags are reset.
Skxx =1: Outputs and/or flags are set.

READY: 0 FB is not processed
1 FB is processed

Method of operation

Each image has a softkey bar assigned in the "graphical user interface".

Each softkey has three items allocated:

- Input, flag, or output that is influenced when the related softkey is selected.
- Input, flag, or output whose status information causes the softkey to appear released or depressed.
- Input, flag, or output whose status information causes the softkey to appear dark or illuminated.

Depending on the active image, the graphical user interface provides the PLC with the addresses of the outputs and/or flags that are to be influenced when the softkeys are selected. When the "GUI_SK16" function block is processed in the PLC program, these outputs and/or flags are set if the related softkey has been selected. The flags and/or outputs are cleared when the softkey is de-selected or the image is changed. The PLC program processes the outputs and/or flags and performs machine functions according to the related states.

The "GUI_SK16" function block is used for transferring the signal states of the "SK1 ... SK8" softkeys. SK16". A softkey can be derived from any input.

Note: The "GUI_SK16" function block may not be programmed together with the "USERBOF" function block (mutual interference).

Rexroth
Bosch Group

Application rules

The function block may only be processed once. To ensure that the influenced outputs and/or flags remain the same throughout the PLC program, the function block must be inserted at the beginning of the PLC program.

Invocation in a structured PLC program:

In a structured PLC program, the function block is programmed as the first network in the initialization step in the first action block. This function block must not be invoked again in the further sequence of the PLC program.

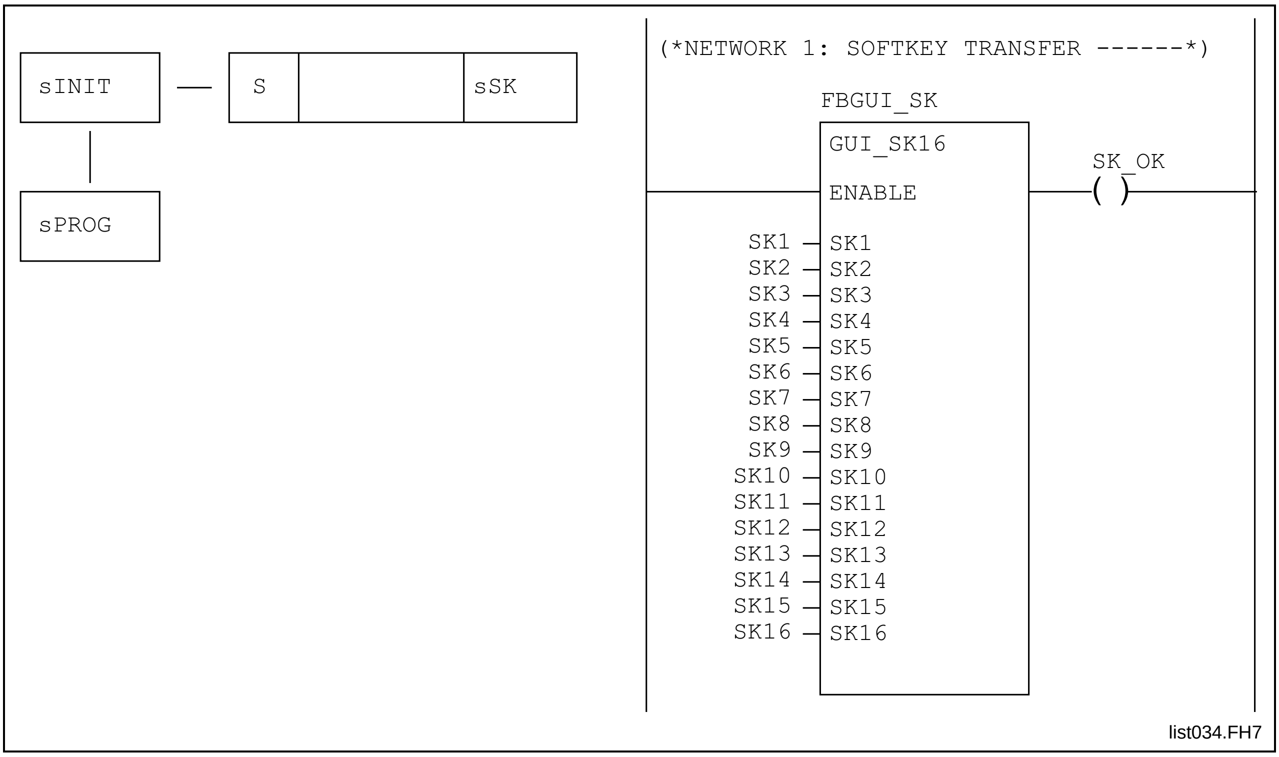


Fig. 3-23: Invocation in structured PLC programs

Invocation in a nonstructured PLC program:

In an unstructured PLC program, the function block is programmed as the first network. There is no further invocation of the function block.

```
(*NETWORK 1: SOFTKEY TRANSFER -------------------------------------------------*)
                   FBGUI_SK
                  +--------------+
                  | GUI_SK16     |                                SK_OK
------------------| ENABLE       |----------------------------------( )--------
                  |              |
           SK1 ---| SK1          |
           SK2 ---| SK2          |
           SK3 ---| SK3          |
           SK4 ---| SK4          |
           SK5 ---| SK5          |
           SK6 ---| SK6          |
           SK7 ---| SK7          |
           SK8 ---| SK8          |
           SK9 ---| SK9          |
          SK10 ---| SK10         |
          SK11 ---| SK11         |
          SK12 ---| SK12         |
          SK13 ---| SK13         |
          SK14 ---| SK14         |
          SK15 ---| SK15         |
          SK16 ---| SK16         |
                  +--------------+

SK1              MACHINE FUNCTION KEY SOFTKEY 1         %I1.3.0        BOOL
SK2              MACHINE FUNCTION KEY SOFTKEY 2         %I1.3.1        BOOL
SK3              MACHINE FUNCTION KEY SOFTKEY 3         %I1.3.2        BOOL
SK4              MACHINE FUNCTION KEY SOFTKEY 4         %I1.3.3        BOOL
SK5              MACHINE FUNCTION KEY SOFTKEY 5         %I1.3.4        BOOL
SK6              MACHINE FUNCTION KEY SOFTKEY 6         %I1.3.5        BOOL
SK7              MACHINE FUNCTION KEY SOFTKEY 7         %I1.3.6        BOOL
SK8              MACHINE FUNCTION KEY SOFTKEY 8         %I1.3.7        BOOL
SK9              MACHINE FUNCTION KEY SOFTKEY 9         %I1.4.0        BOOL
SK10             MACHINE FUNCTION KEY SOFTKEY 10        %I1.4.1        BOOL
SK11             MACHINE FUNCTION KEY SOFTKEY 11        %I1.4.2        BOOL
SK12             MACHINE FUNCTION KEY SOFTKEY 12        %I1.4.3        BOOL
SK13             MACHINE FUNCTION KEY SOFTKEY 13        %I1.4.4        BOOL
SK14             MACHINE FUNCTION KEY SOFTKEY 14        %I1.4.5        BOOL
SK15             MACHINE FUNCTION KEY SOFTKEY 15        %I1.4.6        BOOL
SK16             MACHINE FUNCTION KEY SOFTKEY 16        %I1.4.7        BOOL
SK               GRAPHICAL USER INTERFACE FB                           GIU_SK16
SK_OK            FLAG 'FB SK EXECUTED'                                 BOOL
```

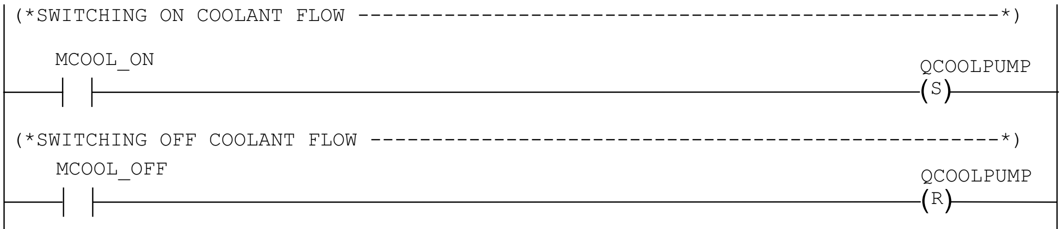
list035.FH7

Fig. 3-24: Invocation in unstructured PLC programs

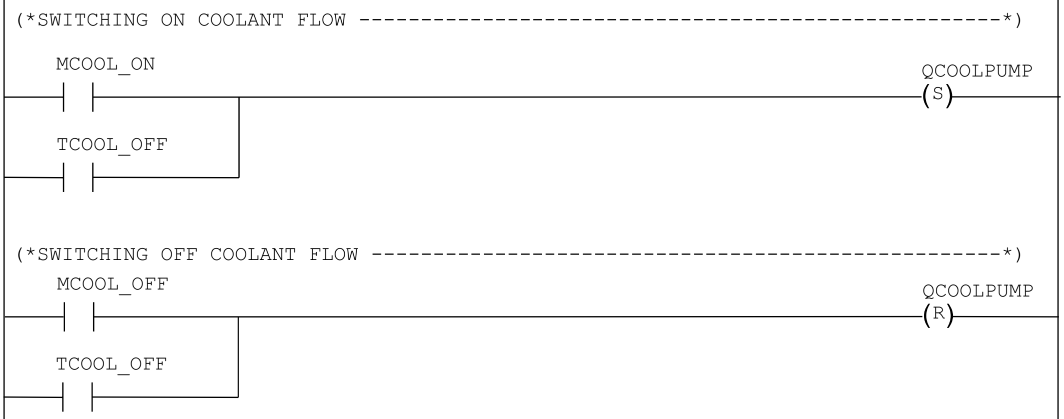
Rexroth
Bosch Group

Manipulating flags/outputs:

Manipulating an output/flag from the "graphical user interface":



Manipulating an output/flag from graphical user interface and via keys:



MCOOL_ON	SWITCH ON COOLANT FLAG	VAR	BOOL
QCOOLPUMP	COOLANT PUMP VALVE	Q1.0.0	BOOL
MCOOLOFF	SWITCH OFF COOLANT FLAG	VAR	BOOL
TCOOL_ON	KEY SWITCH ON COOLANT	I1.2.1	BOOL
TCOOLOFF	KEY SWITCH OFF COOLANT	I1.2.0	BOOL

list036.FH7

Fig. 3-25: Manipulation of output/flags

Function blocks for accessing tool data

The "TLD_RD" function block enables tool data to be read from the PLC program. With certain restrictions, the "TLD_WR" function block can be used for writing to tool data. Since the tool data is maintained in the NC, the PLC must access this information as required (i.e. serially). Thus, the tool data will be available after one PLC cycle at the earliest.

With the function blocks TLG_RD and TLG_WR, the tool group status bits can be accessed reading and partly writing from the PLC.

The function block interfaces are exactly defined. When a function block is invoked, the programmer merely "connects" the individual signals.

Interfaces of the function blocks "TLD_RD" and "TLD_WR"

- Reading tool data "TLD_RD":

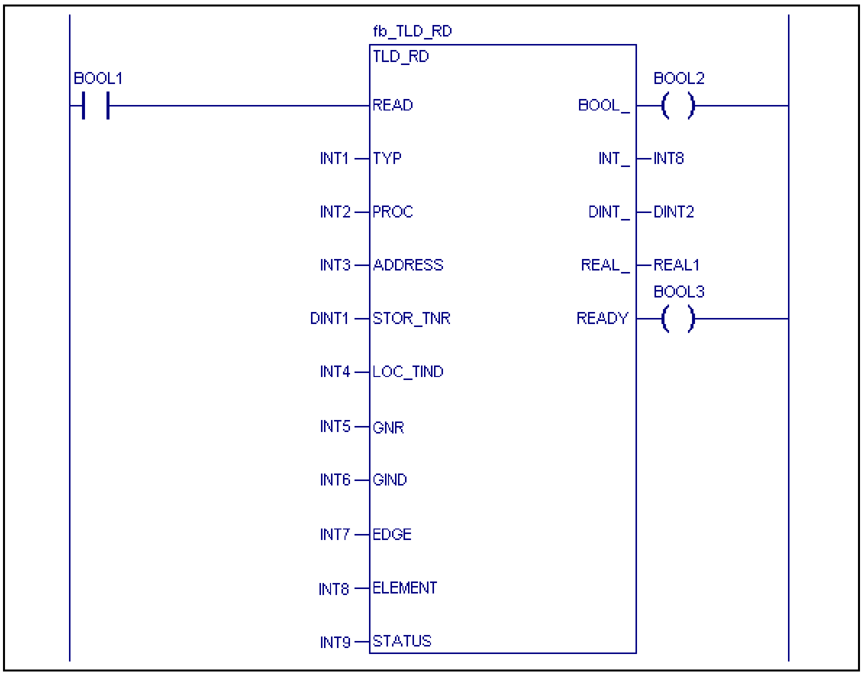


Fig. 3-26: Read tool data

- Writing tool data "TLD_WR:

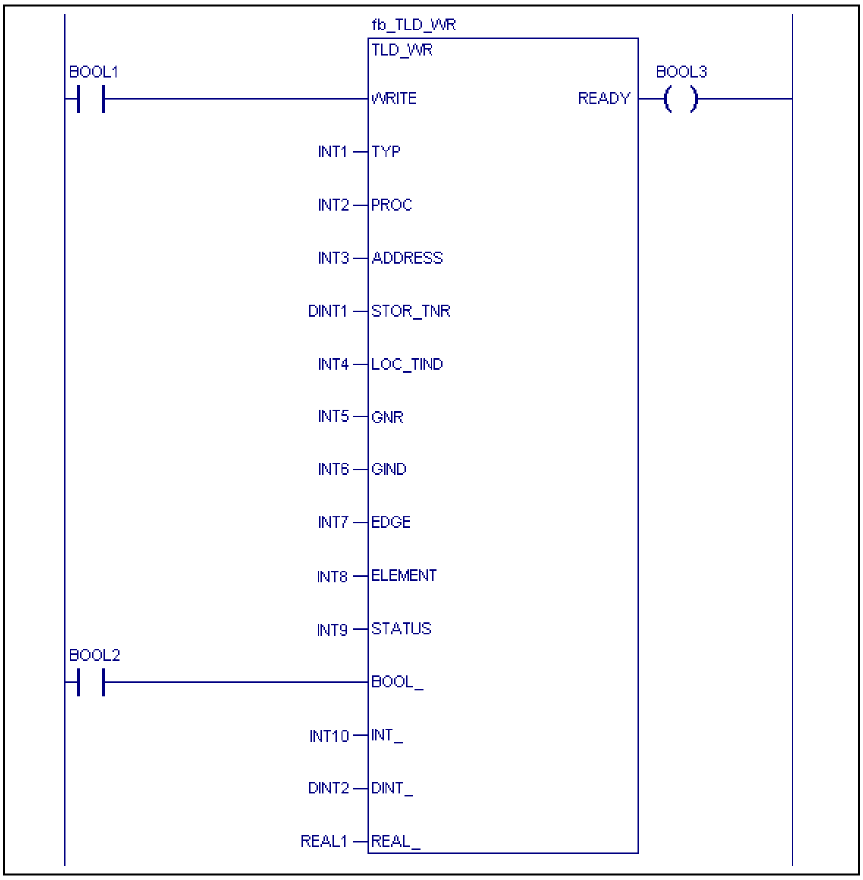


Fig. 3-27: Write tool data

READ:	0 FB not active 1 Initiate reading tool data
WRITE:	0 FB not active 1 Initiate writing tool data
TYPE:	Data type that is to be read/written 0 - BOOL 7 - INT 9 - DINT 10 - REAL With "Read tool", this corresponds to the FB output (BOOL, INT, DINT, REAL) that is to be selected.
PROC	: Number of the process to which the tool data is allocated (0...6)
ADDRESS:	0 Addressing via magazine/location 1 Addressing via tool/duplo number
STOR_TNR:	Magazine or tool number During addressing memory/location: (0…3, 7) 0 - Magazine/turret 1 - Spindle 2 - Gripper 3 - Position 7 – Active tool With tool/duplo number addressing: (1..00.9999999)
LOC_TIND:	Location or index number With magazine/location addressing: (0…999) With tool/duplo number addressing: (1..0.9999)
GNR:	Tool group number (0...99) Insignificant with magazine/location addressing: (0) With tool/duplo number addressing: (0..99)
GIND:	Tool group duplo number (0...99) Insignificant with magazine/location addressing: (0) With tool/duplo number addressing: (0..99)
EDGE:	Edge (0...9) 0 – base tool data 1...9 – edge data
ELEMENT:	Data element With edge 0: (3...26) →Base tool data With edge 1...9: (1...36) →edge data
STATUS	: Number of status bit (not applicable for Boolean tool data) With edge 0: (0...31) With edge 1…9: (0...15)
BOOL_:	Tool data that is to be read or to be written
INT_:	Tool data that is to be read or to be written
DINT_:	Tool data that is to be read or to be written
REAL_:	Tool data that is to be read or to be written
READY:	When reading tool data 0 - Data invalid 1 - Data valid With writing tool data 0 - Data transfer active or FM not active 1 - Data transferred

Note: For addressing, see *"Rexroth MTC 200 Tool management"* description (project planning manual).

Tool list data:

Base tool data, tool status bits, tool group status bits

Basic tool data (per tool)						V23_20030319		
DESIGNATION	**RANGE**	**DATA TYPE in the PLC**	**UNIT**	**DE**		**OPT.**	**SL**	**TL**
Tool identification								
Index address	hexadecimal double		-	01			X	X
ID (tool NAME)	up to 28 characters	STRG28	-	02				X
Memory	0 - 2		-	03				X
Location	0 - 999		-	04				X
Tool number	1 - 9999999	DINT	-	05			X	X
Tool duplo number	1 - 9999	INT	-	06				X
Correction type	1 - 5	USINT	-	07			X	X
Number of tool edges	1 - 9	USINT	-	08			X	X
Tool Status	0/1 (32 status bits)	USINT	-	09				X
Location data								
Free half-locations	0 - 4	USINT	-	10				X
Old pocket	1 - 999	INT	-	11				X
Stor. of next setup tool	0 - 2	INT	-	12				X
Loc. of next setup tool	1 - 999	INT	-	13				X
Stor. of prev. setup tool	0 - 2	INT	-	14				X
Loc. of prev. setup tool	1 - 999	INT	-	15				X
Units								
Time unit	0/1 (0: min, 1: cycl.)	USINT	-	16				X
Unit of length	0/1 (0: mm, 1: inches)	USINT	-	17			X	X
Technology data								
Tool code	0 - 9	USINT	-	18			X	X
Representation type	0 - 65535	INT	-	19			X	X
User data								
User data 1	+/- 1.2 * 10-38 - +/- 3.4 * 10+38 and 0 (9 significant digits)	REAL	any	20		**A00.061**		X
User data 2		REAL		21		**A00.062**		X
User data 3		REAL		22		**A00.063**		X
User data 4		REAL		23		**A00.064**		X
User data 5		REAL		24		**A00.065**		X
User data 6		REAL		25		**A00.066**		X
User data 7		REAL		26		**A00.067**		X
User data 8		REAL		27		**A00.068**		X
User data 9		REAL		28		**A00.069**		X
Group data								
				29				
Group number	0 - 99	BYTE	-	30				X
Group duplo number	0 - 99	BYTE	-	31				X
Group status	0/1 (16 status bits)	WORD	-	32				X
Comment	up to 5 x 76 alphanumeric		-	99		**A00.057**	X	

WGD_all_V23_20030319.xls

Data element 99 "Comment" is not loaded in the control.

DE ... Data element
SL ... Setup list-specific datum
R.TL ... Replacement tool
TL ... Tool list-specific datum
CTRL29 ... STRING28
OPT ... Optional datum

Tool status bits 1 - 16 from basic tool data element 09

V22_20030519

Group designation	Group information	Symbol	Value	Bit	Byte	Word	Write access			Type			Comment
							TM	OP	ASP	SL	TL	PL	
Presence	Tool not available	!	1	1	LOW - BYTE 0 ... 7	LOW – WORD 0 ... 15	X			X	X		Tool is missing
	Tool available		0										
	Tool is not required	?	1	2			X			X	X		Tool not required for machining
	Tool required		0										
Error correction type	Correction type wrong	t	1	3			X			X	X		Correction type does not comply with requirements
	correction type not faulty		0										
Error tool edge number	Incorrect number of tool edges	e	1	4			X			X	X		Number of tool edges does not comply with requirements
	Correct number of tool edges		0										
Error tool edge	Tool edge(s) incorrect	f	1	5			X			X	X		Tool edge data does not comply with requirements
	Edge(s) not faulty		0										
Error tool code	Tool code incorrect	$	1	6			X			X	X		Tool data do not correspond to the definition
	Tool code not incorrect		0										
Reserved for extension				7									
Reserved for extension				8									
Location locking	Location locked	B	1	9	HIGH – BYTE 0 ... 7		X	X	X		X	X	ASP/OP: e.g. location is damaged TM: tool is entered
	Location not locked		0										
Reserved for extension	Upper half-location locking			10									
Reserved for extension	Lower half-location locking			11									
Upper half-location reserved	Upper half-location reserved	)	1	12				X	X		X	X	Reserved for temp. moved tools
	Upper half-location not reserved		0										
Lower half-location reserved	Lower half-location reserved	(	1	13				X	X		X	X	Reserved for temp. moved tools
	Lower half-location not reserved		0										
Reserved for extension	Upper half-location covering			14									
Reserved for extension	Lower half-location covering			15									
Location reservation	Location assigned	+	1	16			X				X	X	There is a tool in the location
	Location not assigned		0										

WSB_all_V22_20030519_1.xls

TM ... Tool management
OP ... Bediener
ASP ... Application-specific programs in PLC or NC
LL ... Location-specific status bit

SL ... Setup list-specific status bit
WL ... Tool list-specific status bit
OPT ... Optional datum

Tool status bits 17 - 32 aus
Base tool data - Data element 09

V22_20030519

Group designation	Group information	Symbol	Value	Bit	Byte	Word	Write access TM	Write access OP	Write access ASP	Type SL	Type TL	Type PL	Comment
Wear status	Tool is worn out		1		LOW BYTE 0 - 7	HIGH - WORD 0 ... 15							
	tool is not worn out	d	0	17			X				X		The remaining lifetime of the tool has elapsed (replace)
	Warning limit is reached		1										
	Warning limit not reached	w	0	18			X				X		The remaining lifetime is about to expire (replace)
Alternate identification	Machining tool		1										
	No machining tool	p	0	19			X				X		There is one machining tool for each alternate tool group
	Replacement tool		1										
	no spare tool	s	0	20			X				X		A spare tool is a tool that can still be used, no machining tool
Tool coding	Tool with fixed location coding		1										
	Tool without fixed location coding	C	0	21			X	X	X		X		The tool must only be replaced onto the predef. tool location
Tool block	Tool locked		1										
	Tool is not locked	L	0	22				X	X		X		Tool must not be used
Tool breakage	Tool broken		1										
	Tool is not broken	D	0	23				X	X		X		Tool is damaged, e.g. a tool edge has broken off
Reserved for extension				24									
User tool status 1	User tool status bit 1 Axx.075	any	1 / 0	25	HIGH - BYTE 0 ... 7			X	X		X		Any meaning
User tool status 2	User tool status bit 2 Axx.076		1 / 0	26				X	X		X		Any meaning
User tool status 3	User tool status bit 3 Axx.077		1 / 0	27				X	X		X		Any meaning
User tool status 4	User tool status bit 4 Axx.078		1 / 0	28				X	X		X		Any meaning
User tool status 5	User tool status bit 5 Axx.079		1 / 0	29				X	X		X		Any meaning
User tool status 6	User tool status bit 6 Axx.080		1 / 0	30				X	X		X		Any meaning
User tool status 7	User tool status bit 7 Axx.081		1 / 0	31				X	X		X		Any meaning
User tool status 8	User tool status bit 8 Axx.082		1 / 0	32				X	X		X		Any meaning

WSB_all_V22_20030519_2.xls

TM ... Tool management
OP ... Operator
ASP ... Application-specific programs in PLC or NC
LL ... Location-specific status bit

SL ... Setup list-specific status bit
TL ... Werkzeuglistenspezifisches Statusbit
OPT ... Optional datum

Tool Groups: Group Status Data Element 32									V23_20021112
Status	**xxx**	**Symbol**	**Value**	**Bit**	**Writing Access**			**Type**	**Comment**
					TM	**OP**	**USP**	**GL**	
Presence	Group not available	!	1	1	X			X	Tool in this group is missing
	Group available		0						
	Group not available	?	1	2	X			X	No tool in this group required
	Group available		0						
Group status	Group enabled	L	1	3		X	X	X	Programmable from the user
	Group disabled		0						
Wear state	Group worn out	d	1	4	X			X	At least one tool alternate chain is worn out.
	Group not worn out		0						
	Warning limit is reached	w	1	5	X			X	At least one tool alternate chain of group has reached the warning limit
	Warning limit not reached		0						
Name of alternate	Machining group	p	1	6	X			X	Group is machining group
	No machining group		0						
	Spare group	s	1	7	X			X	Group is alternate group
	No spare group		0						
Reserved for extension				8					
User group status 1	User group status bit 1	any	1 0	9		X	X	X	Any meaning
User group status 2	User group status bit 2	any	1 0	10		X	X	X	Any meaning
User group status 3	User group status bit 3	any	1 0	11		X	X	X	Any meaning
User group status 4	User group status bit 4	any	1 0	12		X	X	X	Any meaning
User group status 5	User group status bit 5	any	1 0	13		X	X	X	Any meaning
User group status 6	User group status bit 6	any	1 0	14		X	X	X	Any meaning
User group status 7	User group status bit 7	any	1 0	15		X	X	X	Any meaning
User group status 8	User group status bit 8	any	1 0	16		X	X	X	Any meaning

WZG_all_V23_20021112.xls

TM ... Tool management
OP ... Operator
USP ... User-specific
LL ... Location-specific status bit

SL ... Setup list-specific status bit
TL ... Tool list-specific status bit
OPT ...Optional date

Tool list data:

Tool edge data, tool edge status bit

Tool edge data (per tool edge) V22_20030317							
DESIGNATION	RANGE	DATA TYPE in PLC	UNIT	DE	OPT.	SL	TL
Tool edge identification							
Tool edge position	0 - 8	USINT		01		X	X
Tool edge status	0; 1 (16 status bits)	WORD		02			X
Tool life data							
Remaining tool life	-99.9 - +100.00	REAL	%	03	A00.059		X
Warning limit	+0.1 - +100.00	REAL		04	A00.059		X
max. utilization time	0 - 9999999 (0: tool life recognition switched off)	REAL	min. or cycles	05	A00.059		X
Time used	0 - 9999.999	REAL		06	A00.059	X	
Geometry Data							
Length L1	-99999.9999 - +99999.9999 or -9999.99999 - +9999.99999	DINT	mm or inches	07			X
Length L2		DINT		08			X
Length L2		DINT		09			X
Radius R		DINT		10			X
Wear L1		DINT		11	A00.055		X
Wear L2		DINT		12	A00.055		X
Wear L3		DINT		13	A00.055		X
Wear R		DINT		14	A00.055		X
Offset L1		DINT		15	A00.056		X
Offset L2		DINT		16	A00.056		X
Offset L3		DINT		17	A00.056		X
Offset R		DINT		18	A00.056		X
Geometry limit values							
L1_min	-99999.9999 - +99999.9999 or -9999.99999 - +9999.99999	DINT	mm or inches	19	A00.060	X	
L1_max		DINT		20	A00.060	X	
L2_min		DINT		21	A00.060	X	
L2_max		DINT		22	A00.060	X	
L3_min		DINT		23	A00.060	X	
L3_max		DINT		24	A00.060	X	
R_min		DINT		25	A00.060	X	
R_max		DINT		26	A00.060	X	
Wear factors							
Wear factor L1	-99999.9999 - +99999.9999 or -9999.99999 - +9999.99999	DINT	mm/min or inch/min or cycles	27	A00.058		X
Wear factor L2		DINT		28	A00.058		X
Wear factor L3		DINT		29	A00.058		X
Wear factor R		DINT		30	A00.058		X
User data							
User data 1	+/- 1.2 * 10-38 - +/- 3.4 * 10+38 and 0 (9 significant digits)	REAL	any	31	A00.070		X
User data 2		REAL	any	32	A00.071		X

User data 3		REAL	any	33	**A00.072**		**X**
User data 4		REAL	any	34	**A00.073**		**X**
User data 5		REAL	any	35	**A00.074**		**X**
User data 6	-99999.9999 - +99999.9999 or -9999.99999 - +9999.99999	DINT	any	36	**A00.092**		**X**
User data 7		DINT	any	37	**A00.093**		**X**
User data 8		DINT	any	38	**A00.094**		**X**
User data 9		DINT	any	39	**A00.095**		**X**
User data 10		DINT	any	40	**A00.096**		**X**

SD_all_V22_20030317.xls

DE ... Data element

SL ... Setup list-specific datum

OPT ... Optional datum

TL ... Tool list-specific datum

Tool edge status bit from tool edge data element 02

V22_20030317

Group name	Group information	Symbol	Value	Bit	Write access			Type		Comment
					TM	OP	ASP	SL	TL	
Incorrect tool edge position	Incorrect tool edge orientation	o	1	1	X			X		Tool edge data do not correspond to the definition
	Tool edge orientation is not incorrect		0							
L1 incorrect	L1 incorrect	1	1	2	X			X		Tool edge data do not correspond to the definition
	L1 not incorrect		0							
L2 incorrect	L2 incorrect	2	1	3	X			X		Tool edge data do not correspond to the definition
	L2 not incorrect		0							
L3 incorrect	L3 incorrect	3	1	4	X			X		Tool edge data do not correspond to the definition
	L3 not incorrect		0							
R incorrect	R incorrect	r	1	5	X			X		Tool edge data do not correspond to the definition
	R not incorrect		0							
Reserved for extensions				6						
Reserved for extensions				7						
Reserved for extensions				8						
Wear state	Tool edge worn out	d	1	9	X				X	The tool edge can no longer be used (replace)
	Tool not worn out		0							
	Warning limit is reached	w	1	10	X				X	The remaining tool life of the tool edge is near its end (replace)
	Warning limit not reached		0							
Reserved for extensions				11						
Reserved for extensions				12						
User tool edge status 1	User tool edge status bit 1 Axx.083	any	1	13		X	X		X	Any meaning
			0							
User tool edge status 2	User tool edge status bit 2 Axx.084	any	1	14		X	X		X	Any meaning
			0							
User tool edge status 3	User tool edge status bit 3 Axx.085	any	1	15		X	X		X	Any meaning
			0							
User tool edge status 4	User tool edge status bit 4 Axx.086	any	1	16		X	X		X	Any meaning
			0							

SSB_all_V22_20030317.xls

TM ... Tool Management

OP ... Operator

ASP ... Application-spec. programs on the PLC or NC

SL ... Setup list-specific status bit

TL ... Tool list-specific status bit

OPT ... Optional datum

Method of operation

Timing of writing tool data:

```
                1          234                 5
                +-----------+    +---------------------------
WRITE       ----+            +-----+
                +-----------+    +---------------------------
Tool data   ----+            +-----+
                           +-+               +-+          +-+
READY       ---------------+ +---------------+ +----------+ +--
```

(1) Setting the WRITE input initiates the transfer of the tool data. The inputs "PROC", "ADDRESS", "STOR_TNR", "LOC_TIND", "EDGE", "ELEMENT" and "STATUS" define the tool data that is to be accessed. The "TYP" input defines the type of the tool data. Automatic conversion is performed if the specified data does not exist in the specified format.

(2) The active "READY" output indicates that the transfer of the tool data is completed. In a write process, the "READY" output shows that data transfer has been completed.

(3) In a single data exchange, the "WRITE" input may now be cleared.

(4) Clearing the "WRITE" input also clears the "READY" output of the function block.

(5) If the "WRITE" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables tool data to be written cyclically.

Timing of reading tool data:

```
                1          234                 5
                +-----------+    +---------------------------
READ        ----+            +-----+

                           +-+               +-+          +-+
Tool data   ---------------+ +---------------+ +----------+ +--

                           +-+               +-+          +-+
READY       ---------------+ +---------------+ +----------+ +--
```

(1) Setting the READ input initiates the transfer of the tool data. The inputs "PROC", "ADDRESS", "STOR_TNR", "LOC_TIND", "EDGE", "ELEMENT" and "STATUS" define the tool data that is to be accessed. The "TYP" input defines the type of the tool data. Automatic conversion is performed if the specified data does not exist in the specified format.

(2) The active "READY" output indicates that the transfer of the tool data is completed. In a reading process, the 'READY' output shows that the requested data are available.

(3) In a single data exchange, the "READ" input may now be cleared.

(4) Clearing the "READ" input also clears the "READY" output of the function block.

(5) If the "READ" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables tool data to be read cyclically.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "TLD_RD" and "TLD_WR". In such a case, error handling reports the cause of the error.

Error type of the function blocks	
Reading Tool Data	*Writing tool data*
TLD_RD: 170	TLD_WR: 169

Fig 3-28: Error type "Reading and writing tool data"

Error numbers:

1 - Invalid input parameter

- The value of the "PROC", "ADDRESS", "STOR_TNR", "LOC_TIND", "GNR", "GIND", "EDGE", "ELEMENT" or "STATUS" input is negative.
- The PROC input is greater than 6.
- The value of the "ADDRESS" input is greater than 1.
- The value of the "STOR_TNR" input is greater than 2.
- The value of the "LOC_TIND" input is greater than 999.
- The "GNR" input is greater than 99.
- The "GIND" input is greater than 99.
- The "EDGE" input is greater than 9.
- The "ELEMENT" input is greater than 32.
- The "STATUS" input is greater than 32.
- The "TYP_" input is different than 0, 7, 9 or 10.

6 - internal transmission error with following possible causes:

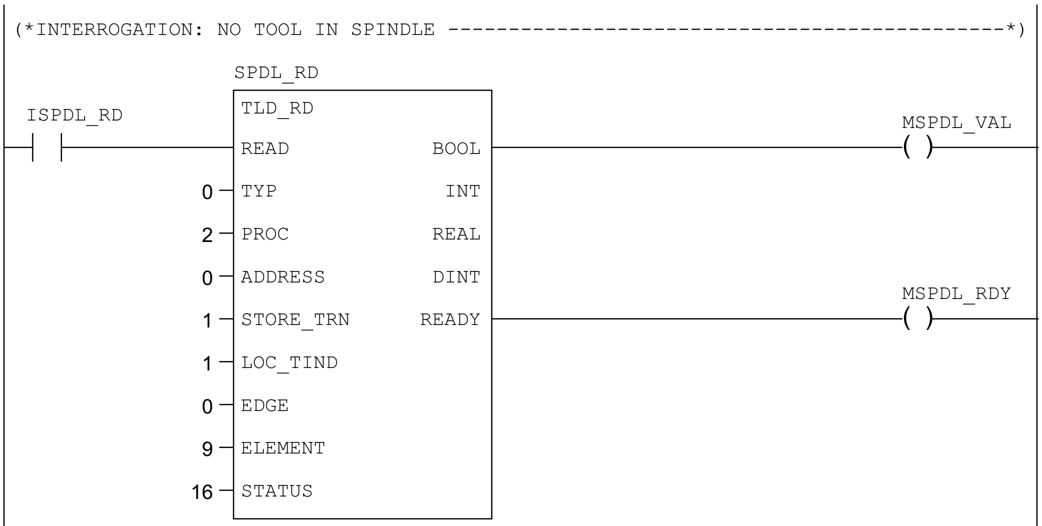
- general message error
- invalid location number
- invalid tool name
- invalid process number
- process still active
- invalid data directory
- no magazine list available
- tool data faulty
- tool data cannot be changed
- location is not empty
- invalid tool group number
- invalid tool group duplo number

Examples of accessing tool data

Reading the tool data type:

The PLC program is to check whether there is a tool in the spindle. The "TLD_RD" function block interrogates the "location occupied" status bit. The "ISPDL_RD" input activates the interrogation. Setting the "READY" function block output makes the result available as "MSPDL_VAL" flag for further processing in the PLC program.

Example Spindle 1 of process 2 is interrogated



SPDL_RD	READ SPINDLE OCCUPANCY FB		TLD_RD
ISPDL_RD	ACTIVATION OF READ SPINDLE OCCUPANCY	%I0.915.0	BOOL
MSPDL_VAL	SPINDLE OCCUPIED FLAG		BOOL
0	BOOLEAN DATA TYPE TO BE READ		ANY_INT
2	PROCESS NUMBER		ANY_INT
0	ADDRESSING USING MAGAZINE/LOCATION		ANY_INT
MSPDL_RDY	READ SPINDLE OCCUPANCY FLAG		BOOL
1	SPINDLE MAGAZINE		ANY_INT
1	SPINDLE NUMBER 1		ANY_INT
0	BASE TOOL DATA		ANY_INT
9	TOOL STATUS BITS		ANY_INT
16	NUMBER OF THE TOOL STATUS BIT		ANY_INT

list037.FH7

Fig. 3-29: Reading tool data: No tool in spindle

Interfaces of the function blocks "TLG_RD" and "TLG_WR"

- Reading tool group status "TLG_RD":

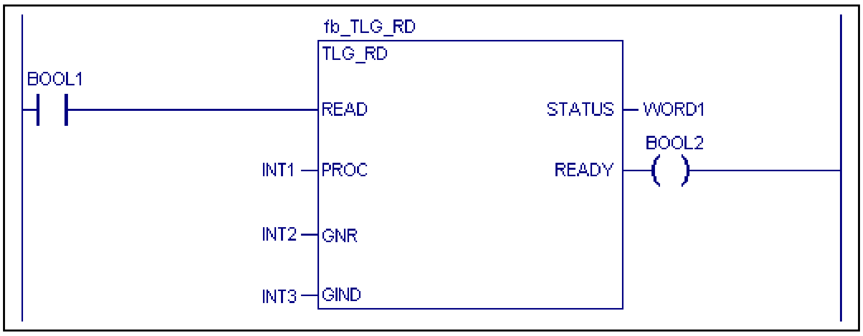


Fig. 3-30: Read tool group status

- Writing tool group status "TLG_WR;
  Write access is valid to status bit, "enable group / not enabled" and to "User group status bit".

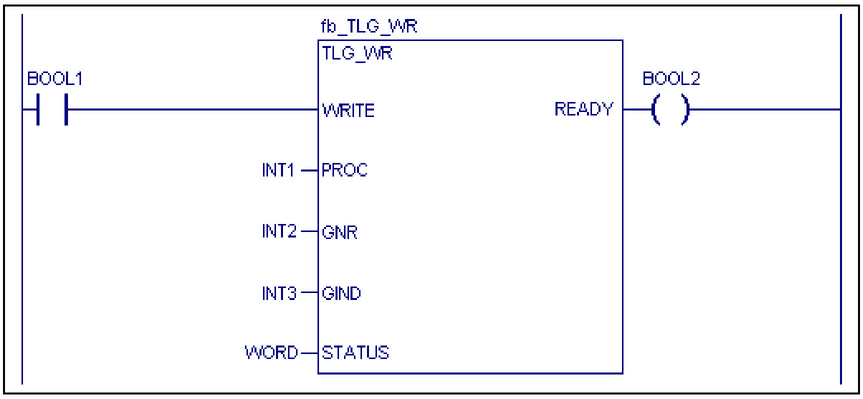


Fig. 3-31: Write tool group status

READ:	0 – FM not active		
	1 - Initiate the reading of group status		
WRITE:	0 – FM not active		
	1 – Initiate the writing of group status		
PROC	: 0..6 Process to which the tool data is allocated		
GNR:	0..99	Tool group number	
GIND:	0..99	Tool group duplo number	
STATUS:	Bit 0:	1	Group does not exist
		0	Group exists
	Bit 1:	1	Group not required
		0	Group required
	Bit 2:	0	Group not disabled
		1	Group disabled
	Bit 3:	0	Group not worn
		1	Group worn
	Bit 4:	0	Warning limit not reached
		1	Warning limit is reached
	Bit 5:	0	No machining group
		1	Machining group
	Bit 6:	0	No spare group
		1	Spare group
	Bit 7:		Reserved
	Bit 8..15:		User group status bits
READY	:when reading tool data		
	0 - Data invalid		
	1 - Data valid		
	When writing group status		
	0 - Data transfer active or FM not active		
	1 - Data transferred		

Rexroth Bosch Group

Note: For a detailed description of tool group management, see documentation "Rexroth MTC 200 Tool management (project planning manual)".

Method of operation

Timing during writing tool group status:

```
                1          234                5
                +-----------+    +---------------------------
READ        ----+           +-----+

                           +-+              +-+         +-+
STATUS      ---------------+ +--------------+ +---------+ +--

                           +-+              +-+         +-+
READY       ---------------+ +--------------+ +---------+ +--
```

(1) Setting the 'READ' input initiates the transfer of the group status. The ‘PROC’, ‘GNR’ and ‘GIND’ inputs define the group status location that is to be read.

(2) The active 'READY' output indicates that the transfer of the tool data is completed.

(3) In a single data exchange, the 'READ' input may now be cleared.

(4) Clearing the 'READ' input also clears the 'READY' output of the function block.

(5) If the 'READ' input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables group status to be read cyclically.

Timing during reading tool group status:

```
                1          234                5
                +-----------+    +---------------------------
WRITE       ----+           +-----+

                +-----------+    +---------------------------
STATUS      ----+           +-----+

                           +-+              +-+         +-+
READY       ---------------+ +--------------+ +---------+ +--
```

(1) Setting the 'WRITE' input initiates the transfer of the group status. The ‘PROC’, ‘GNR’ and ‘GIND’ inputs define the group status location that is to be modified.

(2) The active 'READY' output indicates that the transfer of group status is completed.

(3) In a single data exchange, the 'WRITE' input may now be cleared.

(4) Clearing the 'WRITE' input also clears the 'READY' output of the function block.

(5) If the 'WRITE' input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This enables group status to be written cyclically.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "TLG_RD" and "TLG_WR" In such a case, error handling reports the cause of the error.

Error type of the function blocks	
Reading tool group status	*Writing tool group status*
TLG_RD: 213	TLG_WR: 214

Fig. 3-32: Error type "Reading and writing tool group status"

<u>Error numbers:</u>

1 - Invalid input parameter

- The value of the "PROC", "GNR" or "GIND" input is negative.
- The PROC input is greater than 6.
- The "GNR" input is greater than 99.
- The "GIND" input is greater than 99.

6 - internal transmission error with following possible causes:

- general message error
- invalid process number
- invalid tool group number
- invalid tool group duplo number
- Group status bit can not be modified

Tool Functions in the PLC

Access to tool data via PLC. The size corresponds to the functions already available in the user interface.

Tool functions in the PLC:

- Enter tool data record
- Enable tool data record
- Read tool data record
- Activate tool corrections
- Clear tool data record
- Reset tool data
- Moving tool

Access to the individual tool data has already been provided with the "TLD_RD" and "TLD_WR" function blocks.

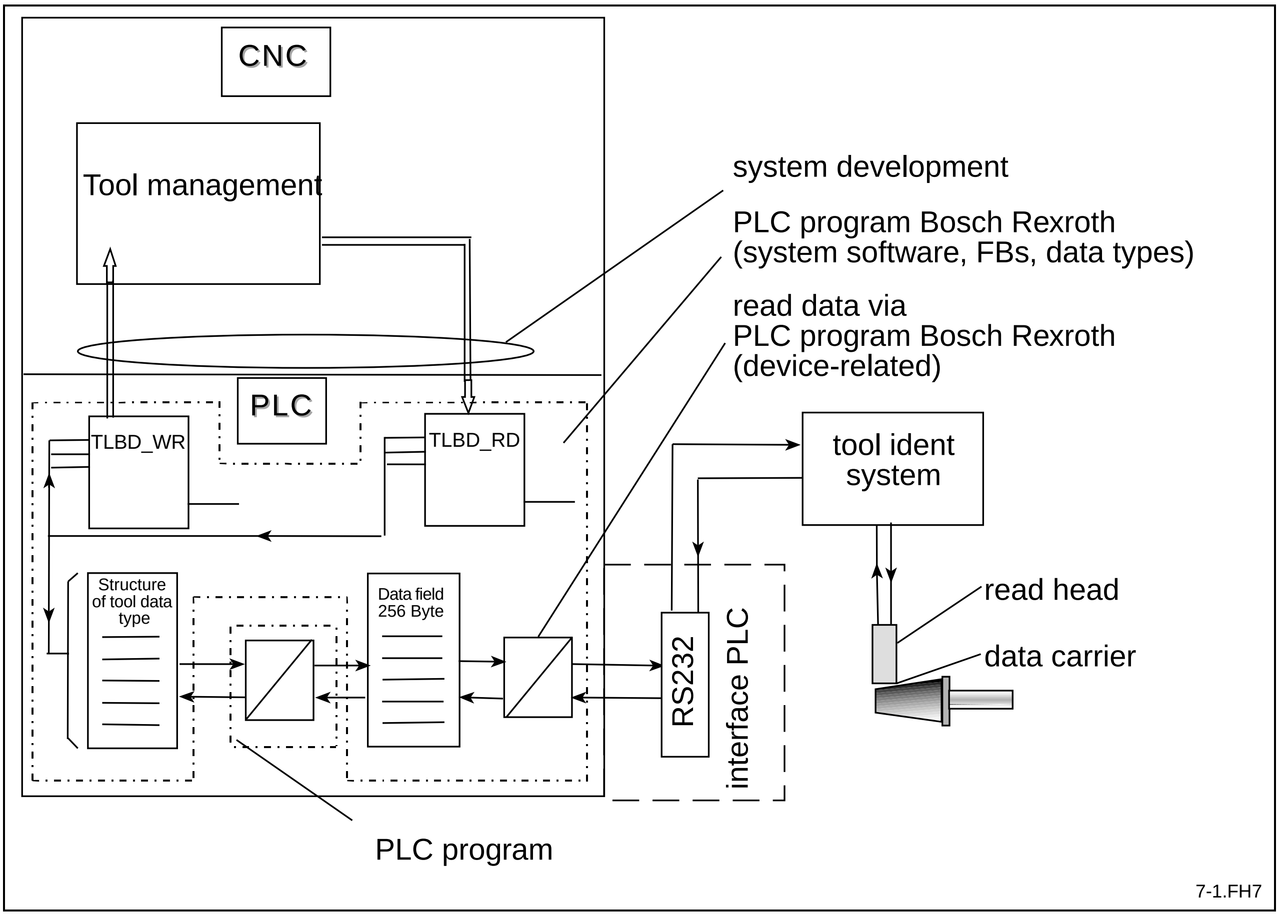


Fig. 3-33: Tool functions

Structure of a tool data block

A tool data record in the PLC program consists of the base tool data and the tool edge data. The tool edge data exist once for each tool edge.

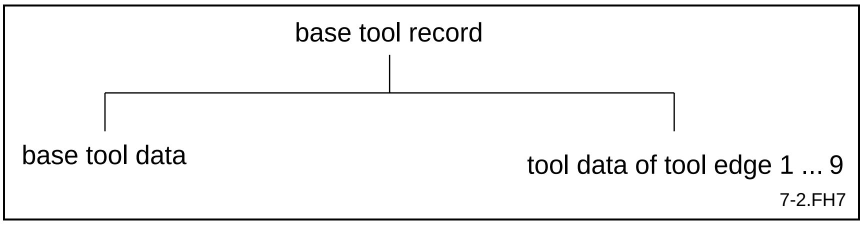


Fig. 3-34: Structure of a tool data record

Structured data are used for programming the tool base data and tool edge data in the PLC (cf. process or axis data types).

<u>Data record of base tool data:</u>

```
STRUCT TLBD
        TOOL_NAME:    CHAR28;    (*En.D02: ID (Tool name)*)
        T_NR:         DINT;      (*En.D05: Tool number*)
        INDEX_NR:     INT;       (*En.D06: Index number*)
        CORR_TYP:     USINT;     (*En.D07: Correction type*)
        EDGES:        USINT;     (*En.D08: No. of tool edges*)
        STATUS:       DWORD;     (*En.D09: Status bits*)
        HALF_POCK:    USINT;     (*En.D10: Free half-locations*)
        G_NR:         USINT;     (*En.D30: Tool group number*)
        OLD_PLACE:    INT;       (*En.D11: Old location*)
        TIME_UNIT:    USINT;     (*En.D16: Time unit*)
        LEN_UNIT:     USINT;     (*En.D17: Linear unit*)
        CODE:         USINT;     (*En.D18: Tool code*)
        G_INDEX:      USINT;     (*EN.D31: Tool group duplo number*)
        DISPLTYP:     INT;       (*En.D19: Display type*)
        USERDAT1:     REAL;      (*En.D20: User data 1*)
        USERDAT2:     REAL;      (*En.D21: User data 2*)
        USERDAT3:     REAL;      (*En.D22: User data 3*)
        USERDAT4:     REAL;      (*En.D232: User data 4*)
        USERDAT5:     REAL;      (*En.D24: User data 5*)
        USERDAT6:     REAL;      (*En.D25: User data 6*)
        USERDAT7:     REAL;      (*En.D26: User data 7*)
        USERDAT8:     REAL;      (*En.D27: User data 8*)
        USERDAT9:     REAL;      (*En.D28: User data 9*)
END_STRUCT
```

Data record of tool edge data:

```
STRUCT TLED
          ORIENT:       USINT;         (*En.D01: Tool edge orientation*)
          STATUS:       WORD;          (*En.D02: Status bits*)
          F1:           USINT;         (*reserved*)
          REM_LIFE:     REAL;          (*En.D03: Remaining tool life*)
          WARN_LIM:     REAL;          (*En.D04: Warning limit*)
          MAX_LIFE:     REAL;          (*En.D05: Max. useful life*)
          LENGTH_L1:    DINT;          (*En.D07: Length L1*)
          LENGTH_L2:    DINT;          (*En.D08: Length L2*)
          LENGTH_L3:    DINT;          (*En.D09: Length L3*)
          RADIUS_R:     DINT;          (*En.D10: Radius R*)
          WEAR_L1:      DINT;          (*En.D11: Wear L1*)
          WEAR_L2:      DINT;          (*En.D12: Wear L2*)
          WEAR_L3:      DINT;          (*En.D13: Wear L3*)
          WEAR_R:       DINT;          (*En.D14: Wear R*)
          OFFSET_L1:    DINT;          (*En.D15: Offset L1*)
          OFFSET_L2:    DINT;          (*En.D16: Offset L2*)
          OFFSET_L3:    DINT;          (*En.D17: Offset L3*)
          OFFSET_R:     DINT;          (*En.D27: Wear factor L1*)
          WEARFCTL2:    DINT;          (*En.D28: Wear factor L2*)
          WEARFCTL3:    DINT;          (*En.D29: Wear factor L3*)
          WEARFCTR:     DINT;          (*En.D30: Wear factor R*)
          USERDAT1:     REAL;          (*En.D31: User data 1*)
          USERDAT2:     REAL;          (*En.D32: User data 2*)
          USERDAT3:     REAL;          (*En.D33: User data 3*)
          USERDAT4:     REAL;          (*En.D34: User data 4*)
          USERDAT5:     REAL;          (*En.D35: User data 5*)
          USERDAT6:     DINT;          (*En.D36: User data 6*)
          USERDAT7:     DINT;          (*En.D37: User data 7*)
          USERDAT8:     DINT;          (*En.D38: User data 8*)
          USERDAT9:     DINT;          (*En.D39: User data 9*)
          USERDAT10:    DINT;          (*En.D40: User data 10*)
END_STRUCT
```

Function blocks for the tool functions

The tool functions

- enter,
- Enable
- Remove
- reset, and
- Move

are initiated by function blocks. Addressing the tool data that is to be accessed is similar to addressing the "TLD_RD" or "TLD_WR" function blocks.

- Entering a Tool (Base Tool Data "TLBD_WR" and Tool Edge Data "TLED_WR" separately)

The PLC can use the "TLBD_WR" and "TLED_WR" function blocks for entering tool data records in the magazine list. The target magazine must not be occupied in this process. Data transfer is performed in individual steps. This means that the base tool data and the tool edge data is written individually.

Note: After the tool data has been written, the tool must be enabled by the "TL_ENABLE" function block.

Note: The function blocks „TLD_RD“, „TLD_WR“, „TLBD_RD“ XE "TLBD_RD" , „TLED_RD“ XE "TLED_RD" , „TL_DELETE“ XE "Werkzeug löschen - TL_DELETE" XE "TL_DELETE" , „TL_MOVE“ XE "TL_MOVE" and „TL_RESET“ contain additional input parameter "Tool group number" and "Tool group duplo number" as of Firmware version 23VRS. The parameter will be required because of the new NC function "Tool group management". For a detailed description of tool group management, see documentation "Rexroth MTC 200 Tool management (project planning manual)".

- Enter base tool data - "TLBD_WR":

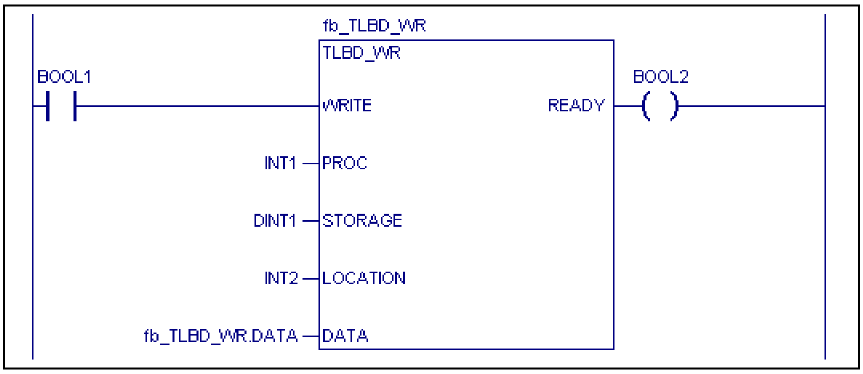


Fig. 3-35: Write base tool data

- Enter tool edge data - "TLED_WR":

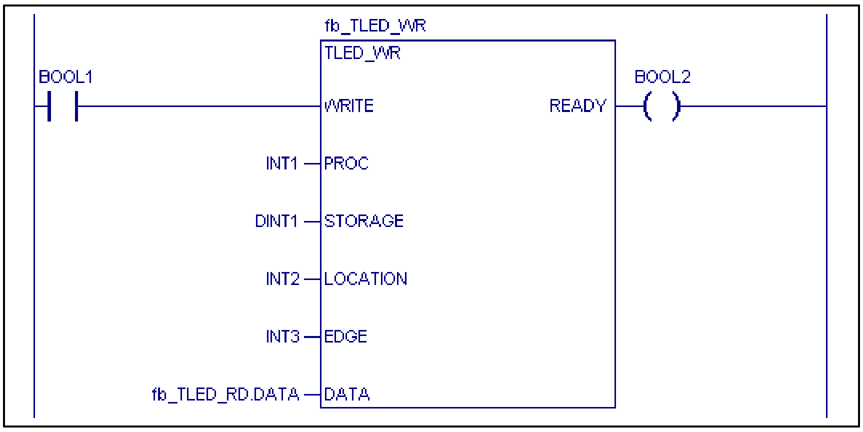


Fig. 3-36: Write tool edge data

WRITE: 0 FB not active
1 Initiate the writing of tool data
PROC: Process number (0-6)
STORAGE: Memory(0…2)
0 - Magazine/turret
1 - Spindle
2 - Gripper
LOCATION:Tool location number (0...999)
EDGE: Tool edge (1...9)
DATA: Data structure of base tool data:

READY: When writing tool data
0 - Data transfer active or FM not active
1 - Data transferred

Method of operation

Timing of entering tool data:

```
                1          234                5
                +-----------+    +---------------------------
WRITE       ----+            +----+
                +-----------+    +---------------------------
Tool data   ----+            +----+
                           +-+                +-+        +-+
READY       ---------------+ +----------------+ +--------+ +--
```

(1) Setting the "WRITE" input initiates the transfer of the base tool data and/or the tool edge data. The PROC, STORAGE, LOCATION and EDGE inputs define the location and/or the tool that is to be entered. The base tool data and/or the tool edge data is transferred as a data structure at the "DATA" input.

(2) The active "READY" output indicates that the entry of a tool is completed. The data has now completely been transferred to the NC.

(3) In a single data exchange, the "WRITE" input may now be cleared.

(4) Clearing the "WRITE" input also clears the "READY" output of the function block.

(5) If the "WRITE" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This causes an attempt to be made to enter a further tool in the previously occupied location. The NC issues an error message for acknowledgment.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "TLBD_WR" and "TLED_WR". In such a case, error handling reports the cause of the error.

Error type of the function blocks	
Entering the base tool data	*Entering tool edge data*
TLBD_WR: 186	TLED_WR: 187

Fig. 3-37: Error types "Base tool data and tool edge data"

Error numbers:

1 - Invalid input parameter

- The value of the PROC, STORAGE, LOCATION or EDGE input is negative.
- The PROC input is greater than 6.
- The value of the STORAGE input is greater than 2.
- The value of the LOCATION input is greater than 999.
- The EDGE input is greater than 9.

6 - internal transmission error with following possible causes:

- general message error
- invalid location number
- invalid tool name
- invalid process number
- process still active
- invalid data directory

- no magazine list available
- tool data faulty
- tool data cannot be changed
- location is not empty

Enabling a tool "TL_ENABLE"

The ""TL_ENABLE" function block is used for enabling a tool data record for further utilization. Before, the tool data record was inhibited by the process of entering a tool.

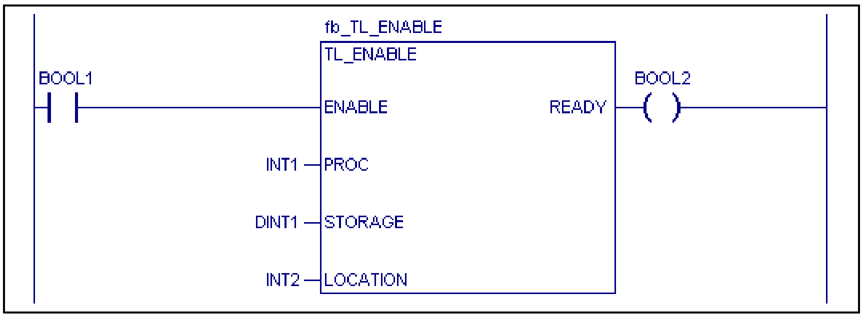


Fig. 3-38: Enable tool

ENABLE: 0 FB not active
1 Enable tool data
PROC: Process number (0-6)
STORAGE: Memory(0…2)
0 - Magazine/turret
1 - Spindle
2 - Gripper
LOCATION:Tool location number (0...999)

READY: 0 Enabling active or FB not active
1 Tool enabled

Timing of enabling a tool:

```
                1          234                5
                +-----------+    +----------------------------
ENABLE      ----+           +----+
                           +-+               +-+         +-+
READY       ---------------+ +---------------+ +---------+ +--
```

(1) Setting the "ENABLE" input initiates the process of enabling the tool. The PROC, STORAGE and LOCATION inputs define the location that is to be enabled.

(2) The active "READY" output indicates that the tool has been enabled.

(3) In a single enabling process, the "ENABLE" input may now be cleared.

(4) Clearing the "ENABLE" input also clears the "READY" output of the function block.

(5) If the "ENABLE" input remains statically ON, a new enabling request is automatically initiated once the old enabling process has been completed. This causes unnecessary loading of the internal data channel.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "TL_ENABLE". In such a case, error handling reports the cause of the error.

Error type of the function block
Enabling tools
TL_ENABLE: 188

Fig. 3-39: Error type "Enabling a tool"

Error numbers:

1 - Invalid input parameter

- The value of the PROC, STORAGE or LOCATION input is negative.
- The PROC input is greater than 6.
- The value of the STORAGE input is greater than 2.
- The value of the LOCATION input is greater than 999.

6 - internal transmission error with following possible causes:

- general message error
- invalid location number
- invalid tool name
- invalid process number
- process still active
- invalid data directory
- no magazine list available
- tool data faulty
- tool data cannot be changed
- location is not empty

Reading tool data "TLBD_RD" and/or "TLED_RD"

The "TLBD_RD" and "TLED_RD" function blocks are used for transferring tool data records from the magazine list to the PLC. The source memory is not cleared when it is read (no destructive reading). Data transfer is performed in individual steps. This means that the base tool data and "n" cutting data items are requested individually.

- Reading base tool data - "TLBD_RD":

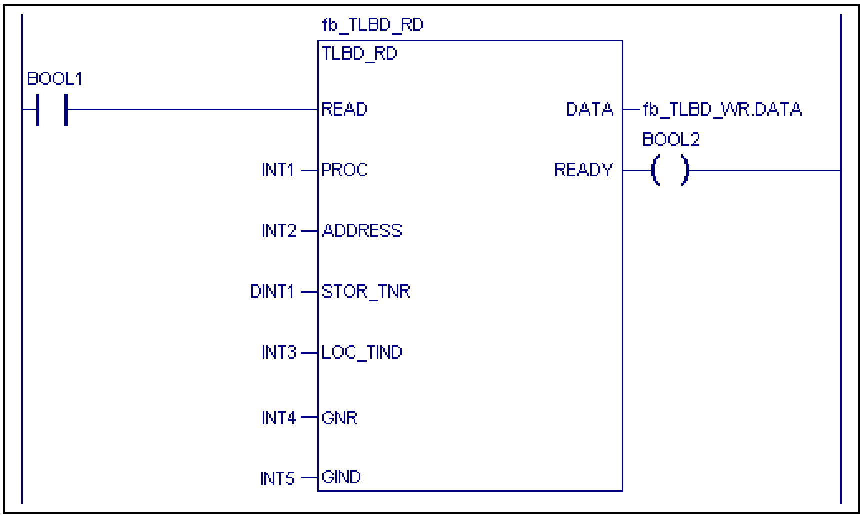


Fig. 3-40: Read base tool data

READ: 0 FB not active
1 Initiate reading tool data

PROC: Process number (0-6)

ADDRESS: 0 Addressing via magazine/location
1 Addressing via tool/duplo number

STOR_TNR: Magazine or tool number
With magazine/location addressing: (0…2)
0 - Magazine/turret
1 - Spindle
2 - Gripper
With tool/duplo number addressing: (1..00.9999999)

LOC_TIND: Location or index number
With magazine/location addressing: (0…999)
With tool/duplo number addressing: (1..0.9999)

GNR: Tool group number (0...99)
Insignificant with magazine/location addressing: (0)
With tool/duplo number addressing: (0..99)

GIND: Tool group duplo number (0...99)
Insignificant with magazine/location addressing: (0)
With tool/duplo number addressing: (0..99)

DATA: Data structure of base tool data:

READY: When reading tool data
0 - Data invalid
1 - Data valid

- Reading tool edge data - "TLED_RD":

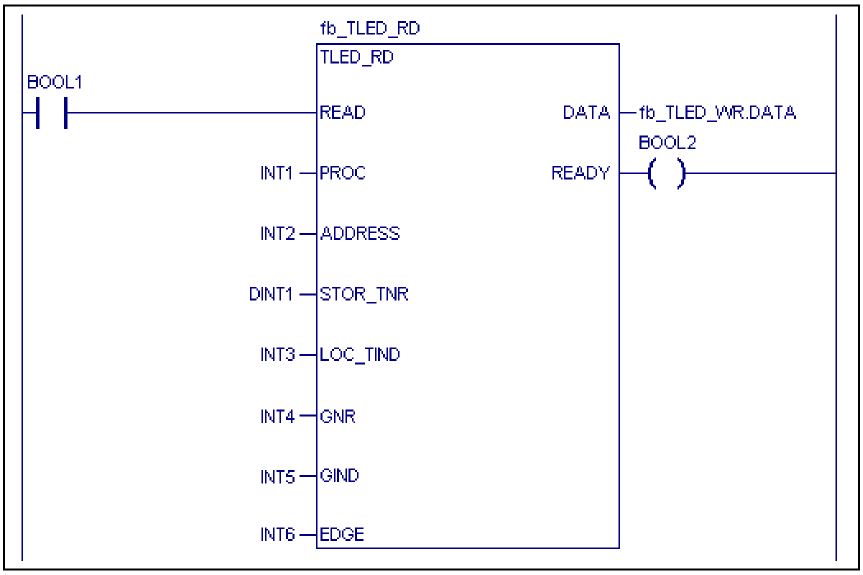


Fig. 3-41: Read tool edge data

READ:	0 FB not active
	1 Initiate reading tool data
PROC:	Process number (0-6)
ADDRESS:	0 Addressing via magazine/location
	1 Addressing via tool/duplo number
STOR_TNR:	Magazine or tool number
	With magazine/location addressing: (0…2)
	0 - Magazine/turret
	1 - Spindle
	2 - Gripper
	With tool/duplo number addressing: (1..00.9999999)
LOC_TIND:	Location or index number
	With magazine/location addressing: (0…999)
	With tool/duplo number addressing: (1..0.9999)
GNR:	Tool group number (0...99)
	Insignificant with magazine/location addressing: (0)
	With tool/duplo number addressing: (0..99)
GIND:	Tool group duplo number (0...99)
	Insignificant with magazine/location addressing: (0)
	With tool/duplo number addressing: (0..99)
EDGE:	Tool edge (1...9)
DATA:	Data structure of tool edge data:
READY:	When reading tool data
	0 - Data invalid
	1 - Data valid

Method of operation

Timing of reading tool data:

```
                1          234                5
                +-----------+    +-----------------------------
READ        ----+           +-----+
                           +-+               +-+          +-+
Tool data   ---------------+ +---------------+ +----------+ +--
                           +-+               +-+          +-+
READY       ---------------+ +---------------+ +----------+ +--
```

(1) Setting the "READ" input initiates the transfer of the base tool data and/or the tool edge data. The PROC, ADDRESS, STOR_TNR, LOC_TIND and EDGE inputs define the tool data that is to be read.

(2) The active "READY" output indicates that the tool data is present at the "DATA" data output.

(3) In a single data exchange, the "READ" input may now be cleared.

(4) Clearing the "READ" input also clears the "READY" output of the function block.

(5) If the "READ" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed. This causes unnecessary loading of the NC PLC data channel. Cyclic reading of tool data records should be avoided.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "TLBD_RD" and "TLED_RD". In such a case, error handling reports the cause of the error.

Error type of the function blocks	
Reading base tool data	*Reading tool edge data*
TLBD_RD: 189	TLED_RD: 190

Fig. 3-42: Error types "Reading base tool data and/or tool edge data"

Error numbers:

1 - Invalid input parameter

- The value of PROC, ADDRESS, STOR_TNR, LOC_TIND, GNR, GIND or EDGE input is negative.
- The PROC input is greater than 6.
- The value of the ADDRESS input is greater than 1.
- The value of the STOR_TNR input is greater than 2.
- The value of the LOC_TIND input is greater than 999.
- The GNR input is greater than 99.
- The GIND input is greater than 99.
- The EDGE input is greater than 9.

6 - internal transmission error with following possible causes:

- general message error
- invalid location number
- invalid tool name
- invalid process number
- process still active
- invalid data directory
- no magazine list available
- tool data faulty
- tool data cannot be changed
- location is not empty
- invalid tool group number
- invalid tool group duplo number

Tool reset "TL_RESET"

The "TL_RESET" tool reset function block sets the tool's life count to the maximum value of 100%. All wear-related corrections are set to "0".

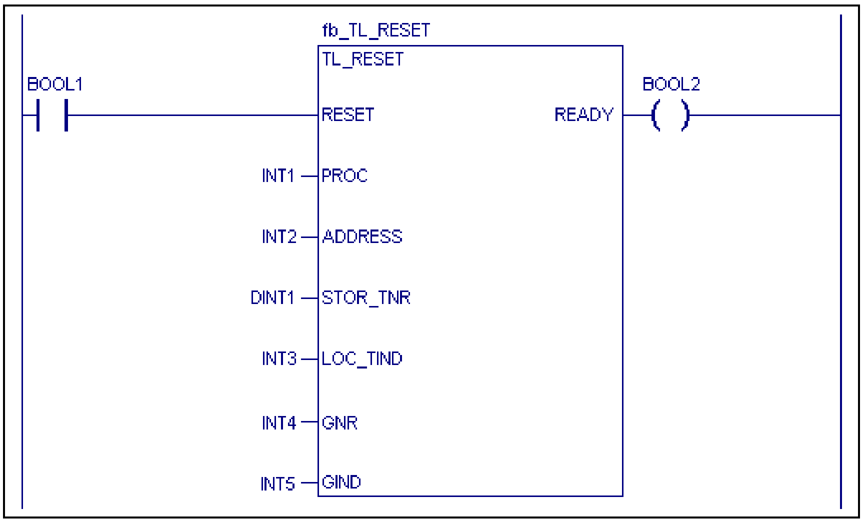


Fig. 3-43: Reset tool

RESET:	0 FB not active 1 Reset tool life
PROC:	Process number (0-6)
ADDRESS:	0 Addressing via magazine/location 1 Addressing via tool/duplo number
STOR_TNR:	Magazine or tool number With magazine/location addressing: (0…2) 0 - Magazine/turret 1 - Spindle 2 - Gripper With tool/duplo number addressing: (1..00.9999999)
LOC_TIND:	Location or index number With magazine/location addressing: (0…999) With tool/duplo number addressing: (1..0.9999)
GNR:	Tool group number (0...99) Insignificant with magazine/location addressing: (0) With tool/duplo number addressing: (0..99)
GIND:	Tool group duplo number (0...99) Insignificant with magazine/location addressing: (0) With tool/duplo number addressing: (0..99)
READY:	0 FB not active or reset not yet performed 1 Tool has been reset

Method of operation

Timing of resetting a tool:

```
               1          234                  5
               +-----------+    +---------------------------
RESET      ----+           +-----+
                          +-+                 +-+         +-+
READY      ---------------+ +----------------+ +----------+ +--
```

(1) Setting the "RESET" input initiates the resetting of the tool data. The PROC, ADDRESS, STOR_TNR and LOC_TIND inputs define the location and/or tool that shall be reset.

(2) The active "READY" output indicates that resetting the tool has been completed.

(3) In a single reset process, the "RESET" input may now be cleared.

(4) Clearing the "WRITE" input also clears the "READY" output of the function block.

(5) If the "RESET" input remains statically ON, a new reset is automatically initiated once the old reset has been completed. This causes unnecessary loading of the NC PLC data channel. Cyclic resetting of tools should be avoided.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "TL_RESET". In such a case, error handling reports the cause of the error.

Error type of the function block
Resetting tools
TL_RESET: 191

Fig. 3-44: Error type "Tool reset"

Error numbers:

1 - Invalid input parameter

- The value of PROC, ADDRESS, STOR_TNR, LOC_TIND, GNR or GIND input is negative.
- The PROC input is greater than 6.
- The value of the ADDRESS input is greater than 1.
- The value of the STOR_TNR input is greater than 2.
- The value of the LOC_TIND input is greater than 999.
- The GNR input is greater than 99.
- The GIND input is greater than 99.

6 - internal transmission error with following possible causes:

- general message error
- invalid location number
- invalid tool name
- invalid process number
- process still active
- invalid data directory
- no magazine list available
- tool data faulty
- tool data cannot be changed
- location is not empty
- invalid tool group number
- invalid tool group duplo number

Delete tool "TL_DELETE"

The 'TL_DELETE' function block deletes a tool from the magazine list. The tool data is irrevocably lost.

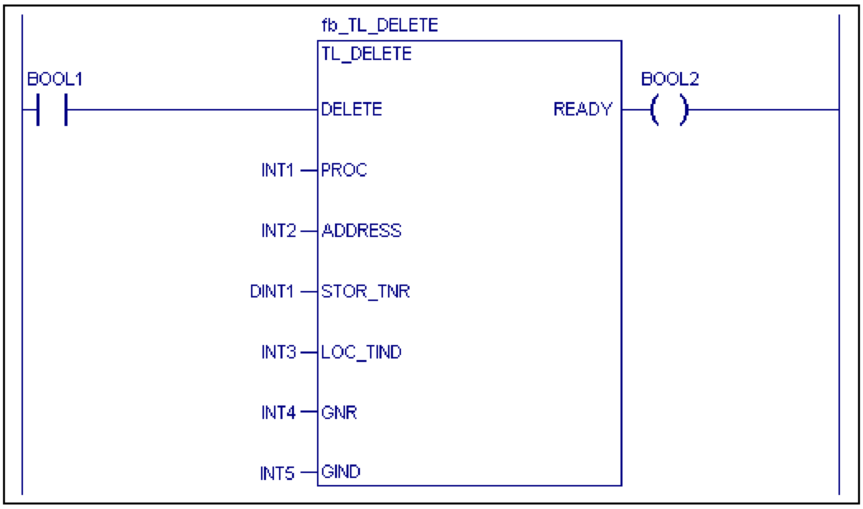


Fig. 3-45: Delete tool

DELETE:	0 FB not active 1 Tool is deleted
PROC:	Process number (0-6)
ADDRESS:	0 Addressing via magazine/location 1 Addressing via tool/duplo number
STOR_TNR:	Magazine or tool number With magazine/location addressing: (0…2) 0 - Magazine/turret 1 - Spindle 2 - Gripper With tool/duplo number addressing: (1..00.9999999)
LOC_TIND:	Location or index number With magazine/location addressing: (0…999) With tool/duplo number addressing: (1..0.9999)
GNR:	Tool group number (0...99) Insignificant with magazine/location addressing: (0) With tool/duplo number addressing: (0..99)
GIND:	Tool group duplo number (0...99) Insignificant with magazine/location addressing: (0) With tool/duplo number addressing: (0..99)
READY:	0 FB not active or tool not yet deleted 1 Tool has been deleted

Method of operation

Timing of deleting a tool:

```
                1          234                5
                +-----------+    +-----------------------------
DELETE      ----+           +----+
                           +-+               +-+         +-+
READY       ---------------+ +---------------+ +---------+ +---
```

(1) Setting the "DELETE" input initiates the deletion of the tool data record. The PROC, ADDRESS, STOR_TNR and LOC_TIND inputs define the location and/or tool that shall be deleted.

(2) The active "READY" output indicates that the tool has been deleted.

(3) In a single deletion, the "DELETE" input may now be cleared.

(4) Clearing the "DELETE" input also clears the "READY" output of the function block.

(5) If the "DELETE" input remains statically ON, a new deletion is automatically initiated once the tool has been deleted. This causes unnecessary loading of the NC PLC data channel, and the second deletion will lead to an error because the tool no longer exists. Cyclic deletion of tools should therefore be avoided.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "TL_DELETE". In such a case, error handling reports the cause of the error.

Error type of the function block
Deleting tools
TL_DELETE: 192

Fig. 3-46: Error type "Deleting tool data"

Error numbers:

1 - Invalid input parameter

- The value of PROC, ADDRESS, STOR_TNR, LOC_TIND, GNR or GIND input is negative.
- The PROC input is greater than 6.
- The value of the ADDRESS input is greater than 1.
- The value of the STOR_TNR input is greater than 2.
- The value of the LOC_TIND input is greater than 999.
- The GNR input is greater than 99.
- The GIND input is greater than 99.

6 - internal transmission error with following possible causes:

- general message error
- invalid location number
- invalid tool name
- invalid process number
- process still active
- invalid data directory
- no magazine list available
- tool data faulty
- tool data cannot be changed
- location is not empty
- invalid tool group number
- invalid tool group duplo number

Moving tool "TL_MOVE"

The "TL_MOVE" function block is used for moving a tool data record to a new tool magazine location.

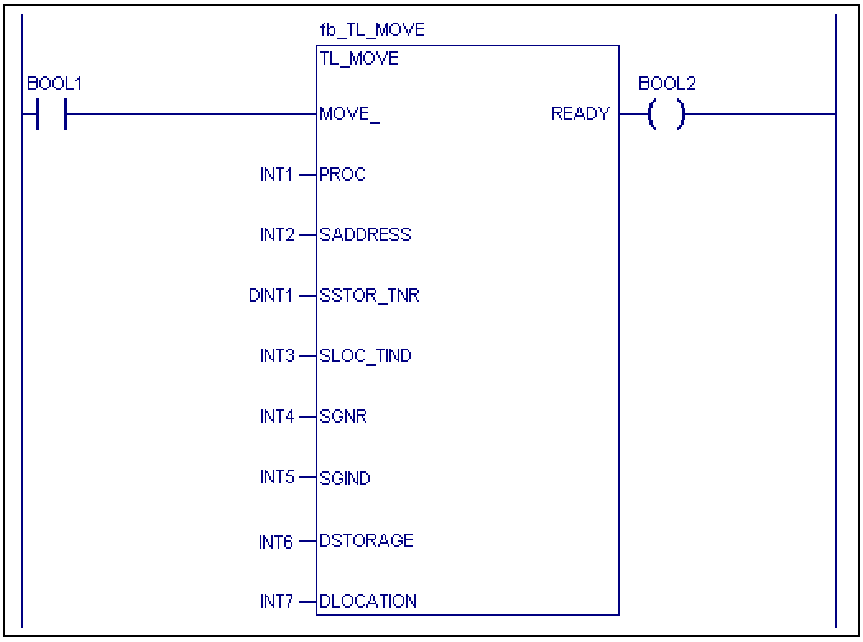


Fig. 3-47: Move tool

MOVE: 0 FB not active
1 Move tool to new location

PROC: Process number (0-6)

SADDRESS: Source address
- 0 - Addressing via magazine/location
- 1 - Addressing via tool/duplo number

SSTOR_TNR: Source magazine and/or tool number
- With magazine/location addressing: (0…2)
  - 0 - Magazine/turret
  - 1 - Spindle
  - 2 - Gripper
- With tool/duplo number addressing: (1..00.9999999)

SLOC_TIND: Source location or index number
- With magazine/location addressing: (0…999)
- With tool/duplo number addressing: (1..0.9999)

SGNR: Tool group number (0...99)
- Insignificant with magazine/location addressing: (0)
- With tool/duplo number addressing: (0..99)

SGIND: Tool group duplo number (0...99)
- Insignificant with magazine/location addressing: (0)
- With tool/duplo number addressing: (0..99)

DSTORAGE: Designed magazine: (0...2)
- 0 - Magazine/turret
- 1 - Spindle
- 2 - Gripper

DLOCATION: Designed location: (0...999)

READY: 0 FB not active or tool not yet moved
1 Tool has been moved.

Method of operation

Timing of moving a tool:

```
                    1           234                  5
                    +-----------+    +---------------------------
MOVE           ----+            +----+
                               +-+                +-+         +-+
READY          ---------------+ +---------------+ +----------+ +--
```

(1) Setting the "MOVE" input initiates the move of the tool data. The PROC, SADDRESS, SSTOR_TNR and SLOC_TIND inputs define the location and/or tool that shall be moved. The DSTORAGE and DLOCATION inputs define the location to which the tool data is moved.

(2) The active "READY" output indicates that moving the tool has been completed.

(3) Once the tool has been moved, the MOVE input may be cleared.

(4) Clearing the "MOVE" input also clears the "READY" output of the function block.

(5) If the "MOVE" input remains statically ON, a new move with the same source and target addresses will automatically be initiated once the old move has been completed. This causes an error "location occupied" to be issued.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "TL_MOVE". In such a case, error handling reports the cause of the error.

Error type of the function block
Moving tools
TL_MOVE: 193

Fig. 3-48: Error type "Moving a tool"

Error numbers:

1 - Invalid input parameter

- The values of the inputs PROC, SADDRESS, SSTOR_TNR, SLOC_TIND, SGNR, SGIND, DSTORAGE or DLOCATION are negative.
- The PROC input is greater than 6.
- The value of the "SADDRESS" input is greater than 1.
- The value of the "SSTOR_TNR" input is greater than 2.
- The value of the "SLOC_TIND" input is greater than 999.
- The SGNR input is greater than 99.
- The SGIND input is greater than 99.
- The value of the DSTORAGE input is greater than 2.
- The value of the DLOCATION input is greater than 999.

6 - internal transmission error with following possible causes:

- general message error
- invalid location number
- invalid tool name

- invalid process number
- process still active
- invalid data directory
- no magazine list available
- tool data faulty
- tool data cannot be changed
- location is not empty
- invalid tool group number
- invalid tool group duplo number

Function blocks for accessing D corrections "DCD_RD" and "DCD_WR""

The "DCD_RD" function block permits D correction values to be read from the PLC program. The "DCD_WR" function block is used for writing the correction values. The D corrections are maintained in the NC. The PLC must therefore access this information as required (i.e. serially). Thus, the D corrections will be available after one PLC cycle at the earliest.

Interfaces of the function blocks "DCD_RD" and "DCD_WR"

- Reading D corrections - "DCD_RD:

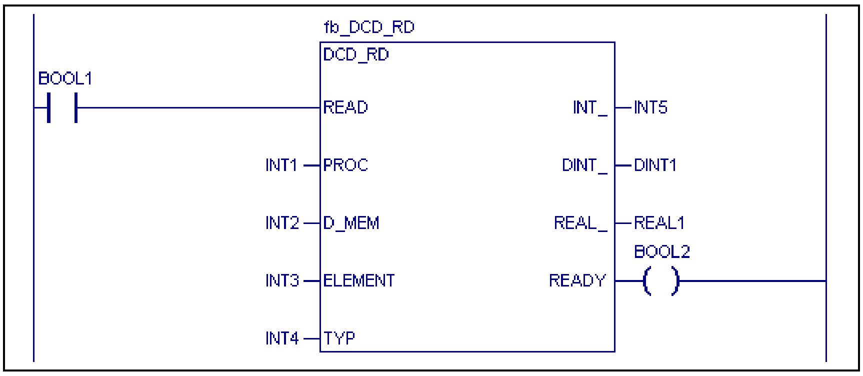


Fig. 3-49: Read D correction

- Writing D corrections - "DCD_WR:

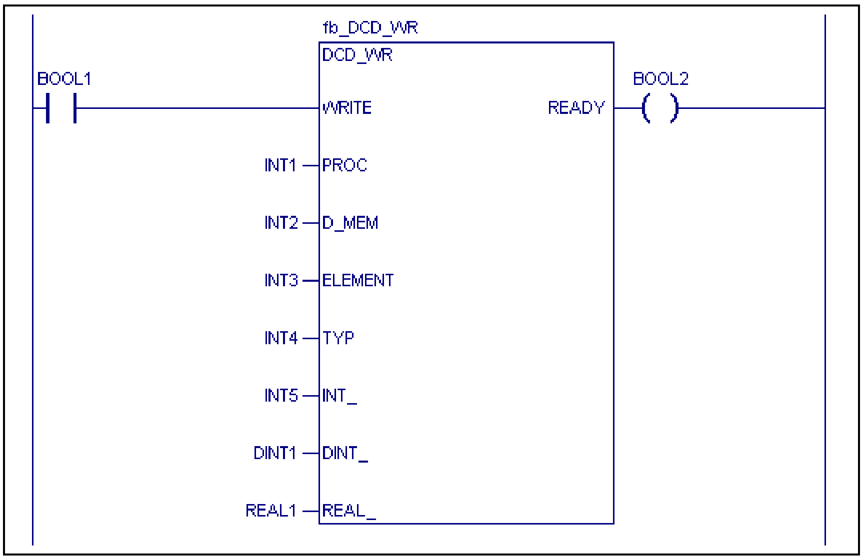


Fig. 3-50: Write D correction

READ:	0 FB not active 1 Initiate reading D corrections
WRITE:	0 FB not active 1 Initiate writing D corrections
PROC:	Process number (0-6)
D_MEM:	D memory number (1...99)
ELEMENT:	Element number (1...4) 1 - Value for L1 2 - Value for L2 3 - Value for L3 4 - Value for R
TYPE:	Data type that is to be read/written 7 - INT 9 - DINT 10 - REAL
INT_:	D corrections
DINT_:	D corrections
REAL_:	D corrections
READY:	Reading D corrections 0 - Data invalid 1 - Data valid Writing D corrections 0 - Data transfer active or FM not active 1 - Data transferred

Rexroth
Bosch Group

Method of operation

Timing of writing D corrections:

```
                  1         234              5
                  +-----------+    +----------------------------
WRITE         ----+           +----+
                  +-----------+    +----------------------------
D corrections ----+           +----+
                             +-+               +-+         +-+
READY         ---------------+ +---------------+ +---------+ +--
```

(1) Setting the "WRITE" input initiates the data transfer of the D corrections. The PROC, D_MEM and ELEMENT inputs define the D corrections that are to be accessed. The "TYP" input defines the type of the D corrections data. Automatic conversion is performed if the specified data does not exist in the specified format.

(2) The active "READY" output indicates that the transfer of the D corrections is completed. In a write process, the "READY" output shows that data transfer has been completed.

(3) In a single data exchange, the "WRITE" input may now be cleared.

(4) Clearing the "WRITE" input also clears the "READY" output of the function block.

(5) If the "WRITE" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed.

Timing of reading D corrections:

```
                   1         234              5
                   +-----------+    +----------------------------
READ           ----+           +----+
                              +-+               +-+         +-+
D corrections  ---------------+ +---------------+ +---------+ +--
                              +-+               +-+         +-+
READY          ---------------+ +---------------+ +---------+ +--
```

(1) Setting the "READ" input initiates the data transfer of the D corrections. The PROC, D_MEM and ELEMENT inputs define the D corrections that are to be accessed. The "TYP" input defines the type of the D corrections data. Automatic conversion is performed if the specified data does not exist in the specified format.

(2) The active "READY" output indicates that the transfer of the D corrections is completed. In a reading process, the 'READY' output shows that the requested data are available.

(3) In a single data exchange, the "READ" input may now be cleared.

(4) Clearing the "READ" input also clears the "READY" output of the function block.

(5) If the "READ" input remains statically ON, a new data transfer is automatically initiated once the old data transfer has been completed.

Error Handling

Any programming errors in the PLC program may cause incorrect execution of the new function blocks "DCD_RD" and "DCD_WR". In such a case, error handling reports the cause of the error.

Error type of the function blocks	
Reading D corrections	*Writing D corrections*
DCD_RD: 221	DCD_WR: 222

Fig. 3-51: Error type "Reading and writing D corrections"

Error numbers:

1 - Invalid input parameter

- The value of the PROC, D_MEM or ELEMENT input is negative.
- The PROC input is greater than 6.
- The value of the D_MEM input is 0 or greater than 99.
- The value of the ELEMENT input is 0 or greater than 4.
- The value of the "TYP_" input is different than 7, 9 or 10

6 - internal transmission error with following possible causes:

- general message error
- invalid location number
- invalid tool name
- invalid process number
- process still active
- invalid data directory
- no magazine list available
- tool data faulty
- tool data cannot be changed
- target location is not empty

4 Error Handling MTC200

4.1 Errors in Functions and Function Blocks

Explanation:

S#ErrorTyp, indicates the fb/fn that initiated the error.

fn – firmware function

*fn – standard function

fb – firmware function block

*fb – standard function block

S#ErrorTyp	Type	Name	Comment
-1	fn	M_FKT	Polling of M help functions with indication of the help function number
-2	fn	M_FKT_Q	Acknowledgement of M help functions with indication of the help function number
-3	fn	S_FKT	Polling of S help functions with indication of the help function number
-4	fn	S_FKT_Q	Acknowledgement of S help functions with indication of the help function number
-5	fn	T_FKT	Polling of T help functions with indication of the help function number
-6	fn	T_FKT_Q	Acknowledgement of T help functions with indication of the help function number
-7	fn	Q_FKT	Polling of Q help functions with indication of the help function number
-8	fn	Q_FKT_Q	Acknowledgement of Q help functions with indication of the help function number
-9	fn	EVENT	Polling of events
-10	fn	EV_ST	Value transmission to events
-11	fn	EV_SET	Conditional setting of events
-12	fn	EV_RES	Conditional resetting of events
-13	fn	MSG_WR	Diagnosis output, message number directly defined
-14	fn	MSG_RD	Read-in of CNC message numbers
-15	fn	MRF	Request for Magazine reference run
-16	fn	MRF_Q	Acknowledgement of Magazine reference run
-17	fn	MMV	Request for Magazine on new position
-18	fn	MMV_Q	Acknowledgement of Magazine on new position
-19	fn	TCH	Request for General tool change
-20	fn	TCH_Q	Acknowledgement of General tool change
-21	fn	TMS	Request for Tool change magazine / spindle
-22	fn	TMS_Q	Acknowledgement of Tool change magazine / spindle
-23	fn	TSM	Request for Tool change spindle / magazine
-24	fn	TSM_Q	Acknowledgement of Tool change spindle / magazine
-25	fn	XMS	Initialization of Tool transfer magazine / spindle
-26	fn	XMS_PA	Tool transfer magazine / spindle allowed
-27	fn	XMS_NA	Tool transfer magazine / spindle not allowed
-28	fn	XMS_Q	Acknowledgement of Tool transfer magazine / spindle
-29	fn	XSM	Initialization of Tool transfer spindle / magazine

-30	fn	XSM_PA	Tool transfer spindle / magazine allowed
-31	fn	XSM_NA	Tool transfer spindle / magazine not allowed
-32	fn	XSM_Q	Acknowledgement of Tool transfer spindle / magazine
-33	fn	XMG	Initialization of Tool transfer magazine / gripper
-34	fn	XMG_PA	Tool transfer magazine / gripper allowed
-35	fn	XMG_NA	Tool transfer magazine / gripper not allowed
-36	fn	XMG_Q	Acknowledgement of Tool transfer magazine / gripper
-37	fn	XSG	Initialization of Tool transfer spindle / gripper
-38	fn	XSG_PA	Tool transfer spindle / gripper allowed
-39	fn	XSG_NA	Tool transfer spindle / gripper not allowed
-40	fn	XSG_Q	Acknowledgement of Tool transfer spindle / gripper
-41	fn	XGS	Initialization of Tool transfer gripper / spindle
-42	fn	XGS_PA	Tool transfer gripper / spindle allowed
-43	fn	XGS_NA	Tool transfer gripper / spindle not allowed
-44	fn	XGS_Q	Acknowledgement of Tool transfer gripper / spindle
-45	fn	XGM	Initialization of Tool transfer gripper / magazine
-46	fn	XGM_PA	Tool transfer gripper / magazine allowed
-47	fn	XGM_NA	Tool transfer gripper / magazine not allowed
-48	fn	XGM_Q	Acknowledgement of Tool transfer gripper / magazine
-49	*fn	GRAY_TO_BYTE	Type conversion of graycode -> BYTE
-50	*fn	BYTE_TO_GRAY	Type conversion of BYTE -> graycode
-51	*fn	BYTE_BCD_TO_INT	Type conversion of BCD code, byte, 2 digits-> INTEGER
-52	*fn	WORD_BCD_TO_INT	Type conversion of BCD code, word, 4 digits -> INTEGER
-53	*fn	BYTE_TO_INT	Type conversion of BYTE -> INTEGER
-54	*fn	WORD_TO_INT	Type conversion of WORD -> INTEGER
-55	*fn	INT_TO_BYTE	Type conversion of integer number -> BYTE
-56	*fn	INT_TO_WORD	Type conversion of integer number -> word
-57	*fn	INT_TO_BCD_WORD	Type conversion of integer number -> 4 digit BCD-coded word
-58	*fn	USINT_TO_INT	Type conversion of UNSIGNED SHORT INTEGER -> INTEGER
-59	*fn	INT_TO_USINT	Type conversion of INTEGER -> UNSIGNED SHORT INTEGER
-60	*fn	USINT_TO_BYTE	Type conversion of UNSIGNED SHORT INTEGER -> BYTE
-61	*fn	BYTE_TO_USINT	Type conversion of BYTE -> UNSIGNED SHORT INTEGER
-62	*fn	CONCAT_BYTE	Attachment of low byte to high byte
-63	*fn	CONCAT_WORD	Attachment of low word to high word
-64	*fn	HIGH_BYTE	Taking the high byte from the word
-65	*fn	LOW_BYTE	Taking the low byte from the word
-66	*fn	HIGH_WORD	Taking the high word from DWORD
-67	*fn	LOW_WORD	Taking the low word from DWORD
-68	*fn	SIGN_INT	Sign of an integer number
-69	*fn	ABS_INT	Absolute value of an integer number
-70	*fn	SHL_BYTE	Move BYTE by n digits to the left
-71	*fn	SHL_WORD	Move WORD by n digits to the left
-72	*fn	SHR_BYTE	Move BYTE by n digits to the right

-73	*fn	SHR_WORD	Move WORD by n digits to the right
-74	*fn	ROL_BYTE	Rotate BYTE by n digits to the left
-75	*fn	ROL_WORD	Rotate WORD by n digits to the left
-76	*fn	ROR_BYTE	Rotate BYTE by n digits to the right
-77	*fn	ROR_WORD	Rotate WORD by n digits to the right
-78	*fb	SR	FLIP_FLOP, dominating setting
-79	*fb	RS	FLIP_FLOP, dominating resetting
-80	*fb	R_TRIG	Identification of a rising edge
-81	*fb	F_TRIG	Identification of a falling edge
-82	*fb	CTUD_USINT_INDR	Up-down counter, value range UNSIGNED SHORT INTEGER
-83	*fb	CTUD_UINT_INDR	Up-down counter, value range UNSIGNED INTEGER
-84	*fb	CTUD_INT_INDR	Up-down counter, value range INTEGER
-85	*fb	TP	Timer pulse
-86	*fb	TON	On-delay timer function block
-87	*fb	TOFF	Off-delay timer function block
-88	fb	SC_WRITE	Function no longer supported
-89	fb	SC_READ	Function no longer supported
-90	fb	VAR_WR	Function no longer supported
-91	fb	VAR_RD	Function no longer supported
-92	fb	SEL_MEM	Selection of the NC program memory
-93	fb	ACT_MEM	Polling of the active NC program memory
-94	fn	XMS_CA	Cancel tool transfer from magazine to spindle
-95	fn	XSM_CA	Cancel tool transfer from spindle to magazine
-96	fn	XMG_CA	Cancel tool transfer from magazine to gripper
-97	fn	XSG_CA	Cancel tool transfer from spindle to gripper
-98	fn	XGS_CA	Cancel tool transfer from gripper to spindle
-99	fn	XGM_CA	Cancel tool transfer from gripper to magazine
-100	fn	MHP	Function no longer supported
-101	fn	MHP_Q	Function no longer supported
-102	fn	GRP	Function no longer supported
-103	fn	GRP_Q	Function no longer supported
-104	fn	REL	Function no longer supported
-105	fn	REL_Q	Function no longer supported
-106	fb	OPEN_COM	Initialization of a general data channel
-107	fb	CLOS_COM	Close data transmission of a general data channel
-108	fb	OPEN_SOT	Function no longer supported
-109	fb	CLOS_SOT	Function no longer supported
-110	fb	WR_BYTE	Write a byte to the transmit buffer
-111	fb	RD_BYTE	Read a byte to general transmission channel
-112	fb	CTRL_COM	Request status of a serial interface
-113	fn	MAG_ACT	Polling of selected magazine axis for combined spindle / turret axis
-114	fn	MAG_Q	Acknowledgement of selected magazine axis for combined spindle / revolving axis

-115	fn	SPDL_ACT	Polling of selected spindle for combined spindle / turret axis
-116	fn	SPDL_Q	Acknowledgement of selected spindle for combined spindle / turret axis
-117	fn	M_ALL	Polling of M help functions without indication of the help function number
-118	fn	M_ALL_Q	Acknowledgement of M help functions without indication of the help function number
-119	fn	S_ALL	Polling of S help functions without indication of the help function number
-120	fn	S_ALL_Q	Acknowledgement of S help functions without indication of the help function number
-121	fn	T_ALL	Polling of T help functions without indication of the help function number
-122	fn	T_ALL_Q	Acknowledgement of T help functions without indication of the help function number
-123	fn	Q_ALL	Polling of Q help functions without indication of the help function number
-124	fn	Q_ALL_Q	Acknowledgement of Q help functions without indication of the help function number
-125	fb	USERBOF	Function no longer supported
-126	fn	M_NR	Reading of the M help function number
-127	fn	S_NR	Reading of the S help function number
-128	fn	Q_NR	Reading of the Q help function number
-129	fn	XFER_CHK	Deactivate check of the tool transfer
-130	fn	MSG_WR_N	Message output with additional information as number
-131	fn	MSG_WR_A	Message output with additional information as axis identification
-132	fb	AXD_WR	Writing of demand data
-133	fb	AXD_RD	Reading of demand data
-134	fn	SPMOD	Request for preselection of spindle mode for rotary-axis-capable main spindle
-135	fn	SPMOD_Q	Acknowledgement of preselection of spindle mode for rotary-axis-capable main spindle
-136	fn	ROTMOD	Request for preselection of rotary axis mode for rotary-axis-capable main spindle
-137	fn	ROTMOD_Q	Acknowledgement of preselection of rotary axis mode for rotary-axis-capable main spindle
-138	*fn	CHAR_TO_BYTE	Type conversion of CHAR -> BYTE
-139	*fn	BYTE_TO_CHAR	Type conversion of BYTE -> CHAR
-140	*fn	INT_TO_STRING	Type conversion of INTEGER -> STRING
-141	*fn	STRING_TO_INT	Type conversion of STRING -> INTEGER
-142	*fn	LEN	Length of a STRING
-143	*fn	LEFT	Leftmost L_ character of a STRING
-144	*fn	RIGHT	Rightmost L_ character of a STRING
-145	*fn	MID	L_ character of a STRING, from the p^{th} character
-146	*fn	CONCAT_S	Combination of two STRINGS
-147	*fn	INSERT	Insert of a STRING after the L^{th} character
-148	*fn	DELETE	Delete L_ character of a STRING from p^{th} character
-149	*fn	REPLACE	Replace L_ character of a STRING from p^{th} character

-150	*fn	FIND	Find character string IN2_ in IN1_
-151	fb	GUI_SK	Function no longer supported
-152	*fn	DINT_TO_DWORD	Type conversion of DOUBLE INTEGER -> DOUBLE WORD
-153	*fn	DWORD_TO_DINT	Type conversion of DOUBLE WORD -> DOUBLE INTEGER
-154	*fn	DINT_TO_INT	Type conversion of DOUBLE INTEGER -> INTEGER
-155	*fn	INT_TO_DINT	Type conversion of INTEGER -> DOUBLE INTEGER
-156	*fn	DINT_TO_TIME	Type conversion of DOUBLE INTEGER -> Time
-157	*fn	TIME_TO_DINT	Type conversion of Time -> DOUBLE INTEGER
-158	fn	HNDWHEEL	Transmission of handwheel position
-159	*fn	SHL_DW	Move DOUBLE WORD by n digits to the left
-160	*fn	SHR_DW	Move DOUBLE WORD by n digits to the right
-161	*fn	ROL_DW	Rotate DOUBLE WORD by n digits to the left
-162	*fn	ROR_DW	Rotate DOUBLE WORD by n digits to the right
-163	fb	RLVAR_WR	Function no longer supported
-164	fb	RLVAR_RD	Function no longer supported
-165	*fn	DINT_TO_REAL	Type conversion of DOUBLE INTEGER -> REAL
-166	*fn	REAL_TO_DINT	Type conversion of REAL -> DOUBLE INTEGER
-167	*fn	STRING_TO_REAL	Type conversion of STRING -> REAL
-168	*fn	REAL_TO_STRING	Type conversion of REAL -> STRING
-169	fb	TLD_WR	Writing to tool data
-170	fb	TLD_RD	Reading of tool data
-171	*fn	DINT_TO_UDINT	Type conversion of DOUBLE INTEGER -> UNSIGNED DOUBLE INTEGER
-172	*fn	UDINT_TO_DINT	Type conversion of UNSIGNED DOUBLE INTEGER -> DOUBLE INTEGER
-173	fb	DATE_RD	Reading of the date
-174	fb	TOD_RD	Reading of the time
-175	fb	OTD_WR	Writing to zero point data
-176	fb	OTD_RD	Reading of zero point data
-177	fb	MTD_WR	Writing to machine data
-178	fb	MTD_RD	Reading of machine data
-179	fb	NETIO_RD	Reading of realtime bits
-180	fn	T_NR	Reading of T help function number for process (PROC)
-181	fn	E_FKT	Reading of E help functions (EDGE) for process (PROC)
-182	fn	E_FKT_Q	Acknowledgement of E help functions (EDGE) for process (PROC)
-183	fn	E_ALL	Polling of any E help function for process (PROC)
-184	fn	E_ALL_Q	Acknowledgement of any E help function for process (PROC)
-185	fn	E_NR	Reading of E help function number for process (PROC)
-186	fb	TLBD_WR	Writing of basic tool data
-187	fb	TLED_WR	Writing of tool tip data
-188	fb	TL_ENABLE	Enable of tool data
-189	fb	TLBD_RD	Reading of basic tool data
-190	fb	TLED_RD	Reading of tool tip data
-191	fb	TL_RESET	Reset tool

-192	fb	TL_DELETE	Delete tool
-193	fb	TL_MOVE	Move tool
-195	*fb	BOOL_BYTE	Conversion of 8-bit -> byte
-196	*fb	BYTE_BOOL	Conversion of Byte -> 8-bit
-197	*fb	BOOL_WORD	Conversion of 16-bit -> word
-198	*fb	WORD_BOOL	Conversion of word -> 16-bit
-199	*fb	BOOL_DW	Conversion of 32-bit -> doubleword
-200	*fb	DW_BOOL	Conversion of doubleword -> 32-bit
-201	fb	FLASH	Pulse generator
-202	fb	TOGGLE	Toggling of a bit
-203	fb	RD_STR	Reading a STRING via the serial interface
-204	fb	WR_STR	Writing a STRING to the transmitter buffer
-205	fn	TIME_DAY	Conversion of TIME_ to the numerical value of day
-206	fn	TIME_HOUR	Conversion of TIME_ to the numerical value of hour
-207	fn	TIME_MIN	Conversion of TIME_ to the numerical value of minute
-208	fn	TIME_SEC	Conversion of TIME_ to the numerical value of second
-209	fn	TIME_MS	Conversion of TIME_ to the numerical value of milliseconds
-210	fn	MAKETIME	Conversion of the FN inputs day 'D', hour 'H', minute 'M', second 'S', and millisecond 'MS' to a time value
-211	fn	COL_CTRL	Switch on and off the one-dimensional approach monitoring
-212	fn	COL_CTRL_S	Request the status of the approach monitoring
-213	fb	TLG_RD	Read the group status information of the PLC user program's tool group
-214	fb	TLG_WR	Modification of the group status data by the PLC user
-215	fb	IB_GROFF	Function no longer supported
-216	fb	MODBUS	Function no longer supported
-217	fn	BT_START	Starts BT Bus
-218	fn	BT_STOP	Stops BT Bus
-219	fn	BT_STATUS	Status information on BT Bus process data exchange
-220			
-221	fb	DCD_RD	Reading of D-corrections
-222	fb	DCD_WR	Writing of D-corrections
-223	fb	NCVAR_RD	Reading of NC variables
-224	fb	NCVAR_WR	Writing of NC variables
-225	fb	GUI_SK16	Enable of (16) machine function keys GUI / menu9
-226	fn	REV_SYNC	Synchronous swiveling of the revolver in the NC set
-227	fb	CLR_COM	Clearing the receiver and transmitter buffers of a serial interface
-228	*fn	SINT_TO_INT	Conversion of SINT number into INT number
-229	*fn	INT_TO_SINT	Conversion of INT number into SINT number
-230	*fn	SINT_TO_BYTE	Conversion of SINT number to BYTE
-231	*fn	BYTE_TO_SINT	Conversion of BYTE to SINT number
-232	*fn	UINT_TO_INT	Type conversion of UNSIGNED INTEGER -> INTEGER
-233	*fn	INT_TO_UINT	Type conversion of INTEGER UNSIGNED -> INTEGER
-234	*fn	UINT_TO_WORD	Type conversion of UNSIGNED INTEGER -> WORD

-235	*fn	WORD_TO_UINT	Type conversion of WORD -> UNSIGNED INTEGER
-236	*fn	REAL_TO_DWORD	Type conversion of REAL -> DWORD
-237	*fn	DWORD_TO_REAL	Type conversion of DWORD -> REAL
-238	Fb	BTXX	Communication between PLC and HMI operating panels of BTV04, BTV05 and BTC06 via a serial interface
-239	fb	DPM_SLDIAG	Single diagnosis of a PROFIBUS slave
-240	fn	VLT_MEAS	In connection with the analog module RMC12.2-2E-1A, it is possible to measure voltages of up to ±10 V.
-241	fn	SAVE_IO	Safety function for projected axis
-242	fb	DPM_STATE	Status information on the PROFIBUS master:
-243	fn	DPM_STOP	Stopping the bus communication
-244	fn	DPM_START	Starting the bus communication
-245	fn	DPM_EXCHG	Status information on PROFIBUS process data exchange
-246	fn	AMP_MEAS	In connection with the analog module RMC12.2-2E-1A, it is possible to measure currents of up to ±20 mA.
-247	fn	RES_MEAS	In connection with the analog module RMC12.2-2E-1A, it is possible to measure resistances of up to 2000 Ω
-248	fn	TMP1MEAS	In connection with the analog module RMC12.2-2E-1A, it is possible to measure temperature ranging from -100 °C to +850 °C.
-249	fn	AN_OUT	In connection with the analog module RMC12.2-2E-1A, it is possible to provide voltages of up to ±10 V and currents of up to +20 mA at the analog output.
-250	fn	DIAG_WORD	Diagnosis functions (hidden to the user)
-251	fn	DIAG_UINT	Diagnosis functions (hidden to the user)
-252	fn	DIAG_INT	Diagnosis functions (hidden to the user)
-253			
-254	fn	BTXX2	Communication block for manual device BTC06 with 64 IO.
-255	fn	RT_DATA	Function block for quick access to NC signal values
-256	fn	DIAG_BYTE	Diagnosis functions (hidden to the user)
-257	fn	DIAG_CHAR	Diagnosis functions (hidden to the user)
-258	fn	DIAG_SINT	Diagnosis functions (hidden to the user)
-259	fn	DIAG_USINT	Diagnosis functions (hidden to the user)
-260	fn	DIAG_BOOL32	Diagnosis functions (hidden to the user)
-261	fn	DIAG_DWORD	Diagnosis functions (hidden to the user)
-262	fn	DIAG_BOOL0	Diagnosis functions (hidden to the user)
-263	fn	DIAG_BOOL4	Diagnosis functions (hidden to the user)
-264	fn	DIAG_BOOL8	Diagnosis functions (hidden to the user)
-265	fn	DIAG_BOOL16	Diagnosis functions (hidden to the user)
-266	fn	DIAG_DINT	Diagnosis functions (hidden to the user)
-267	fn	DIAG_UDINT	Diagnosis functions (hidden to the user)
-268	fn	DIAG_REAL	Diagnosis functions (hidden to the user)
-269	fn	DIAG_TIME	Diagnosis functions (hidden to the user)
-270	fn	SQRT_REAL	Root
-271	fn	LN_REAL	Natural logarithm LN
-272	fn	LOG_REAL	Common logarithm LOG

-273	fn	EXP_REAL	Exponential function
-274	fn	SIN_REAL	Sinusoidal function
-275	fn	COS_REAL	Cosinoidal function
-276	fn	TAN_REAL	Tangential function
-277	fn	ASIN_REAL	Arc sinusoidal function
-278	fn	ACOS_REAL	Arc cosinoidal function
-279	fn	ATAN_REAL	Arc tangential function
-280	fb	MC_INITIALIZATION	Initialization of the DP-RAM interface MC-PLC
-281	fb	MC_CHANGE_ PHASE	Writing of the SERCOS communication phase (phase switchover)
-282	fb	MC_RD_PARA METER	Reading of an MC single parameter
-283	fb	MC_WR_PARA METER	Writing of an MC single parameter
-284	fb	MC_WR_LISTDATA	Writing of an MC list parameter
-285	fn	MC_DIAGNOSIS	Reading of an MC-SIS diagnosis
-286	fb	MC_RD_LISTDATA	Reading of an MC list parameter
-287	fb	MC_RD_DATASTA TUS	Reading of data state of an MC parameter
-288	fb	MC_ABORT_TRANS MISSION	Abortion of an MC parameter transmission
-289	fb	MC_RW_PTR_TLG	MC communication block
-290	fb	MC_RD_PHASE	Reading of SERCOS communication phase
-291	fb	MC_RD_ATTRIBUTE	Reading of attribute of an MC parameter
-292	fb	MC_RD_NAME	Reading of an MC parameter name
-293	fb	MC_RD_UNIT	Reading of an MC parameter unit
-294	fb	MC_RD_MIN_VALUE	Reading of minimum value of an MC parameter
-295	fb	MC_RD_MAX_VALUE	Reading of maximum value of an MC parameter
-296	fb	MC_RD_ELEMENT	Reading of an MC parameter element
-297	fb	MC_WR_ELEMENT	Writing of an MC parameter element
-298	fn	MC_CONCAT_TO_ IDENT_NO	Creating an MC parameter identification number
-299	fn	MC_CONVERT_TO_ IDENT_NO	Converting of an MC parameter identification number
-300	fb	MC_TYP01_CAM_ TABLE	Calculation of a cam (cross cutter) for short formats (Typ 01 – consistency up to the speed)
-301	fb	MC_TYP02_CAM_ TABLE	Calculation of a cam (cross cutter) for short formats (Typ 02 – consistency up to the acceleration)
-302	fb	MC_TYP03_CAM_ TABLE	Calculation of a cam (cross cutter) for long formats (Typ 03 – consistency up to the acceleration)
-303	fb	CALC_LINEAR_Y	calculates linearized for a default X value of a predetermined X-Y value table the respective Y value.
-304	fb	PID_CONTROL	Depending on the wiring of the input variables FB provides "P", "PI", "PD", "I" and "PID" functionalities.
-305	fb	AVERAGE_REAL	calculates the floating average value form maximum 64 REAL values.
-306	fb	AVERAGE_DINT	calculates the floating average value from maximum 1023 DINT values.
-307	fb	MC_RD_ARRAY	Reading of the operating data of a list parameter (4096 bytes, array)

-308	fb	MC_WR_ARRAY	Writing of the operating data of a list parameter (4096 bytes, array)
-309	fb	MC_RW_ARRAY_ TLG	Transmission of any telegram (263 bytes, array) as e.g. subsequently added SIS services or other transmission protocols
-310	fb	PT2_FILTER	serves for low pass filtering of a signal.
-311	fb	MC_TYP04_CAM_TA BLE	Calculation of a cam for pilgrim step mode (TYP04)
-312	fb	MC_TYP05_CAM_TA BLE	Calculation of a cam for a feed movement (TYP05)
-320	fb	CTUD_INT	UP / DOWN Counter INT corresponding IEC
-321	fb	CTUD_UINT	UP / DOWN Counter UINT corresponding IEC
-322	fb	CTUD_USINT	UP / DOWN Counter USINT corresponding IEC
-323	fn	NC_ENABLE	Synchronization of AXD and NC Initialization
-324	fb	PCP_INITIATE	Make a connection to a PCP-Slave
-325	fb	PCP_ABORT	Open a connection
-326	fb	PCP_READ	Read Object values
-327	fb	PCP_WRITE	Modify device parameters
-328	fb	PCP_IDENTIFY	Reading "Type designation plate"
-329	fb	PCP_GET_OD	Read several object discriptions
-330	fn	IB_STATE	Determines the status of the 1st INTERBUS
-331	fn	ASIM_START	ASI-Bus, starts IO data exchange
-332	fn	ASIM_STOP	ASI-Bus, stopps IO data exchange
-333	fn	ASIM_STATE_CH1	ASI-Bus, diagnosis of channel 1
-334	fn	ASIM_RESET	ASI-Bus reset
-335	fn	ASIM_STATE_CH2	ASI-Bus, diagnosis of channel 2
-336	fb	ASIM_SLDIAG	ASI-Bus, diagnosis of slaves

Fig. 4-1: Errors in functions and function blocks

4.2 Error Numbers

ErrorNr: Datentyp USINT

Nummer	Kommentar
0	No error
1	Invalid input parameter The operation, function, function block is not executed. Feed back of unreasonable results possible
2	Range exceeded Invalid result.
3	Range fallen below
4	Conversion error The input parameter cannot be converted correctly. Conversion is done with an internally modified input parameter. Feed back of unreasonable results possible
5	Division by zero Invalid result.
6	Internal transmission error An error occurred during an internal data request from / to the CNC.
7	Subscript error, range exceeded The operation is not executed.
8	Operation not defined
9	Pointer error, invalid address
10	Error during activation of action blocks
233	General SYNAX error
234	Memory not available
235	Addressed PC104 module not available Unable to serve this bus connection, the addressed PC104 module is not available.
236	Process data channel overflow More than eight TLD, OTD, MTD, NC_VAR, TLED, TLBD, DCD programmed in parallel.
237	Too many accesses to variables More than 100 NC variables have been programmed.
238	Interface not open A serial interface, which is not yet open, is accessed by WR_STRING or RD_STRING.
239	STRING overflow processing When using STRING functions, a STRING with more than 255 characters occurred.
240	Invalid input parameter DEVICE A negative device number or an excessive DEVICE number was transmitted during parameterization of the serial interface.
241	Invalid input parameter SERNR A negative number or an excessive number for the serial interface SERNR was transmitted during parameterization of the serial interface.
242	Invalid input parameter BAUD A negative number or an excessive number for the baud rate BAUD was transmitted during parameterization of the serial interface.

243	Invalid input parameter DATA A negative number or an excessive number for the number of data bits DATA was transmitted during parameterization of the serial interface.
244	Invalid input parameter PARITY A negative number or an excessive number for the evaluation of the PARITY bit was transmitted during parameterization of the serial interface.
245	Invalid input parameter STOP A negative number or an excessive number for the number of STOP bits was transmitted during parameterization of the serial interface.
246	Invalid input parameter PROTOKOL A negative number or an excessive number for the type of serial interface PROTOKOL was transmitted during parameterization of the serial interface.
247	Invalid input parameter HANDSH A negative number or an excessive number for the type of handshake HANDSH was transmitted during parameterization of the serial interface.
248	Interface not available
249	All COM interfaces already open
250	Not used any longer
251	Not used any longer
252	General interface error Parity, frame, overrun
253	Transmitter buffer overflow
254	Receiver buffer overflow
255	Timeout acknowledgement telegram

Fig. 4-1: Overview of error numbers

5 Index

;

;Test mode is active - PxxS.DRYRN 1-12, 1-60

6

65005 3-13
65008 1-126, 3-13
65009 3-13
65010 3-13
65011 3-13
65012 1-121, 3-13
65013 1-121, 3-13
65014 1-121, 3-13
65015 1-121, 3-13
65016 3-13
65017 3-13
65018 3-13
65022 3-13
65023 2-54, 3-13
65024 3-13
65025 3-13
65026 3-13
65027 3-13
65029 3-13
65030 3-13
65031 3-13
65032 3-13
65033 3-13
65034 3-13
65035 3-13
65036 3-13
65037 3-13
65038 3-13
65040 3-13
65041 3-13
65042 3-13
65043 3-13
65044 3-13
65045 3-13
65046 3-13
65047 3-13
65048 3-13
65049 3-14
65050 3-14
65051 3-14
65052 3-14
65053 3-14
65054 3-14
65055 3-14
65056 3-14
65057 3-14
65058 3-14
65059 3-14
65060 3-14
65061 3-14
65062 3-14
65063 3-14
65064 3-14
65065 3-14
65066 3-14
65067 3-14
65068 3-14
65069 3-14
65078 3-14
65080 3-14

65081 3-14
65082 3-14
65084 3-14
65085 3-14
65086 3-14
65087 3-14
65088 3-14
65089 3-14
65090 3-14
65091 3-14
65092 3-14
65093 3-14
65094 3-14
65095 3-15
65096 3-15
65097 3-15
65098 3-15
65099 3-15
65100 3-15
65101 3-15
65102 3-15
65103 3-15
65104 3-15
65105 3-15
65106 3-15
65107 3-15
65108 3-15
65109 3-15
65110 3-15
65111 3-15
65112 3-15
65113 3-15
65114 3-15
65115 3-15
65116 3-15
65118 3-15
65119 3-15
65120 3-15
65121 3-15
65122 3-15
65123 3-15
65124 3-15
65125 3-15
65126 3-15
65127 3-15

A

A00.055 3-35
A00.056 3-35
A00.057 3-31
A00.058 3-36
A00.059 3-35
A00.060 3-35, 3-36
A00.061 3-31
A00.062 3-31
A00.063 3-31
A00.064 3-31
A00.065 3-31
A00.066 3-31
A00.067 3-31
A00.068 3-31
A00.069 3-31
A00.070 3-36
A00.071 3-36
A00.072 3-36
A00.073 3-36
A00.074 3-36
A00.092 3-36
A00.093 3-36

A00.094 3-36
A00.095 3-36
A00.096 3-36
ABS_INT 4-2
Absolute value of an INT number - ABS_INT 4-2
Acceleration, programmable - MMA 1-84, 1-93
Accepting NC program number - PxxC.SP 1-9, 1-39, 1-40, 1-98, 1-99
Access to tool data from NC program - TLD 1-95, 1-96, 3-1, 3-28, 3-29, 3-38, 3-39, 3-40, 3-44, 3-46, 3-47, 4-5, 4-10
Acknowledge machining of mechanism MxxC.QP 1-15, 1-104
Acknowledge process - PxxC.QP 1-9, 1-74
Acknowledge process - PxxS.QP 1-11, 1-44, 1-70, 1-73, 1-74, 1-104
Acknowledgement of magazine to spindle tool change - TMS_Q 2-22, 2-31, 2-32, 2-33, 2-35, 4-1
Acknowledgement of spindle to magazine tool change - TSM_Q 2-22, 2-32, 2-34, 2-35, 4-1
ACOS_REAL 4-8
ACT_MEM 3-6, 3-8, 4-3
Activate button - AxxC.SAFAG 1-5, 1-137
Activate spindle CCW rotation (Mx4) - AxxC.M4 1-4, 1-124, 1-125, 1-126
Activate spindle CW rotation (Mx3) - AxxC.M3 1-4, 1-124, 1-125, 1-126
Activate spindle synchronization - PxxC.SSnON 1-101
Activate starting lockout - AxxS.SAFSL 1-7, 1-137
Activate synchronous axes compound <xx> - PxxC.SCONn 1-100
Activating safe operational stop - AxxC.SAFOS 1-5, 1-137
Activating safe stop - AxxC.SAFSS 1-5, 1-137
Collision monitoring, single dimension 2-57, 2-58, 4-6
Activation of save-reduced velocity with safe-limited absolute position 1 - AxxC.SAFA1 1-5, 1-137
Activation of save-reduced velocity with safe-limited absolute position 2 - AxxC.SAFA2 1-5, 1-137
Active homing of feed axes using G74 - PxxS.G74 1-12, 1-61
Active plane rotation together with G50, G51, G54 .. G59 - P 1-6, 1-136, 3-11, 3-13, 4-8
Actual position - PxxC.MGAP 1-10, 1-80, 1-89, 1-93, 1-94
Actual spindle torque value comparison (Md >= Mdx) - AxxS.MD_DX 1-6, 1-136
Adaptive feed control thrust missing - PxxS.THMIS 1-12, 1-60
Adaptive feed regulation excessive thrust - PxxS.EXCTH 1-12, 1-60
Adjustable zero offset - G54 3-21
Adjustable zero offset - G55 3-21
Adjustable zero offset - G56 3-21
Adjustable zero offset - G57 3-21
Adjustable zero offset - G58 3-21
Adjustable zero offset - G59 3-21
Advance program enabling - PxxS.AP 1-11, 1-44, 1-67, 1-69, 1-70, 1-105
Advance program start - MxxS.AP 1-15, 1-105
Advance program start - PxxC.ADV 1-9, 1-30, 1-31, 1-32, 1-41, 1-42, 1-45, 1-46, 1-51, 1-57, 1-69
AMP_MEAS 4-7
AN_OUT 4-7
Analog module RMC12.2-2E-1A, currents up to ±20mA measured. - AMP_MEAS 4-7
Analog module RMC12.2-2E-1A, measure resistors up to 2000O - RES_MEAS 4-7
Analog module RMC12.2-2E-1A, measure temperatures of -100°C up to +850°C - TMP1MEAS 4-7
Analog module RMC12.2-2E-1A, provide voltages up to ±10V and currents up to +20mA at the analog output - AN_OUT 4-7
Analog module RMC12.2-2E-1A, voltages up to ±10V measured - VLT_MEAS 4-7
Announcement of motion in negative direction - AxxS.MCNEG 1-6, 1-133
Announcement of motion in positive direction - AxxS.MCPOS 1-6, 1-132
AP 1-35, 1-44, 1-50, 1-67, 1-105
Appending low-byte to high-byte - CONCAT_BYTE 4-2
Appending low-word to high-word - CONCAT_WORD 4-2
ASIN_REAL 4-8
ATAN_REAL 4-8
Auxiliary line voltage - PxxC.LINE 1-9, 1-18, 1-27
AXD 1-102, 2-54, 3-1, 3-9, 3-10, 3-12, 4-4, 4-9
AXD_RD 3-1, 3-9, 3-10, 3-12, 4-4

Rexroth
Bosch Group

AXD_WR 2-54, 3-1, 3-9, 3-12, 4-4
Axes / spindle lock - AxxC.LOCK 1-6, 1-139
Axis / spindle lock - AxxC.LOCK 1-4, 1-127
Axis enable - AxxC.ENABL 1-4, 1-110, 1-114, 1-115, 1-124, 1-125
Axis functions, activate parking axis - 65082 3-14
Axis functions, activate parking axis - P-7-3642 3-14
Axis functions, activation of simulation mode - 65033 3-13
Axis functions, activation of simulation mode - P-7-3593 3-13
Axis functions, active encoder system - 65047 3-13
Axis functions, active encoder system - P-7-3607 3-13
Axis functions, actual position of axis - 65045 3-13
Axis functions, actual position of axis - P-7-3605 3-13
Axis functions, actual torque value of drive - 65005 3-13
Axis functions, actual torque value of drive - P-7-3565 3-13
Axis functions, address time of dynamic limit monitoring - 65065 3-14
Axis functions, address time of dynamic limit monitoring - P-7-3625 3-14
Axis functions, command position of axis - P-7-3611 3-14
Axis functions, command position of axis -65051 3-14
Axis functions, current position of 2nd encoder system - 65046 3-13
Axis functions, current position of 2nd encoder system - P-7-3606 3-13
Axis functions, negative limit of dynamic limit monitoring - 65049 3-14
Axis functions, negative limit of dynamic limit surveillance - P-7-3609 3-14
Axis functions, positive limit of dynamic limit surveillance - 65048 3-13
Axis functions, positive limit of dynamic limit surveillance - P-7-3608 3-13
Axis functions, reduced motor torque - 65016 3-13
Axis functions, reduced motor torque - P-7-3576 3-13
Axis functions, reduced motor torque on positive stop - 65018 3-13
Axis functions, reduced motor torque upon driving against positive stop - 65017 3-13
Axis functions, reduced motor torque upon driving against positive stop - P-7-3577 3-13
Axis functions, smoothing time constant for 2nd encoder system - P-7-3610 3-14
Axis functions, smoothing time constant for 2nd sensor system - 65050 3-14
Axis functions, static monitoring window for controller-internal drive monitoring function - 65031 3-13
Axis functions, static monitoring window for controller-internal drive monitoring function - P-7-3591 3-13
Axis functions, virtual axis - 65027 3-13
Axis functions, virtual axis - P-7-3587 3-13
Axis homed - AxxS.HOMED 1-6, 1-113, 1-130, 1-131
Axis in positioning window - AxxS.POSWN 1-6, 1-133
Axis movement in negative direction - AxxS.MVNEG 1-6, 1-130
Axis movement in positive direction - AxxS.MVPOS 1-6, 1-130
Axis operative - AxxC.READY 1-4, 1-109, 1-110, 1-114, 1-115, 1-124, 1-125, 1-128, 1-129
AxxC.ENABL 1-4, 1-110, 1-114, 1-115, 1-124, 1-125
AxxC.GEARn 1-118
AxxC.HOME 1-4, 1-34, 1-113, 1-114, 1-115, 1-130
AxxC.HOMLS 1-4, 1-111
AxxC.JGNEG 1-4, 1-30, 1-34, 1-114, 1-115
AxxC.JGPOS 1-4, 1-30, 1-34, 1-114, 1-115
AxxC.LIMIT 1-4, 1-126
AxxC.LOCK 1-4, 1-127
AxxC.M19 1-4, 1-125
AxxC.M3 1-4, 1-124, 1-125, 1-126
AxxC.M4 1-4, 1-124, 1-125, 1-126
AxxC.M5 1-4, 1-124, 1-125
AxxC.MHOLD 1-4, 1-117, 1-124, 1-125, 1-139, 2-43
AxxC.MTAS 1-4, 1-27, 1-108
AxxC.N_CMD 1-4, 1-99, 1-122
AxxC.OTRVL 1-4, 1-27, 1-107
AxxC.OVRD 1-5, 1-62, 1-123
AxxC.QDDS 1-4, 1-116
AxxC.RAPID 1-3, 1-4, 1-126
AxxC.READY 1-4, 1-109, 1-110, 1-114, 1-115, 1-124, 1-125, 1-128, 1-129
AxxC.SAFA1 1-5, 1-137
AxxC.SAFA2 1-5, 1-137
AxxC.SAFAG 1-5, 1-137
AxxC.SAFOS 1-5, 1-137
AxxC.SAFRS 1-5, 1-137

AxxC.SAFSS 1-5, 1-137
AxxC.SPEED 1-4, 1-127
AxxC.SPHLT 1-4, 1-44, 1-99, 1-122
AxxC.SPRST 1-4, 1-123
AxxC.SPSTE 1-4, 1-122
AxxC.SPSTP 1-4, 1-98, 1-121, 2-57
AxxC.STRBP 1-4, 1-111
AxxS.BBDIG 1-6, 1-110, 1-129
AxxS.HOMED 1-6, 1-113, 1-130, 1-131
AxxS.IDDS 1-6, 1-131, 1-132
AxxS.INPOS 1-6, 1-136
AxxS.LD90 1-6, 1-133
AxxS.LIMIT 1-6, 1-139
AxxS.LOCK 1-6, 1-139
AxxS.MCNEG 1-6, 1-133
AxxS.MCPOS 1-6, 1-132
AxxS.MD_DX 1-6, 1-136
AxxS.MHOLD 1-6, 1-139
AxxS.MVNEG 1-6, 1-130
AxxS.MVPOS 1-6, 1-130
AxxS.N_CMD 1-6, 1-134
AxxS.N_MAX 1-6, 1-135
AxxS.N_MIN 1-6, 1-135
AxxS.P_PX 1-6, 1-136, 1-137
AxxS.POSWN 1-6, 1-133
AxxS.RF 1-6, 1-109, 1-114, 1-115, 1-124, 1-125, 1-128, 1-129
AxxS.SAFAC 1-7, 1-137
AxxS.SAFEN 1-7, 1-137
AxxS.SAFP1 1-7, 1-137
AxxS.SAFP2 1-7, 1-137
AxxS.SAFP3 1-7, 1-137
AxxS.SAFP4 1-7, 1-137
AxxS.SAFRY 1-7, 1-137
AxxS.SAFSL 1-7, 1-137
AxxS.SPEED 1-6, 1-138
AxxS.SYNC 1-6, 1-134
AxxS.WPn 1-131

B

BES 2-17
Bit, switch - TOGGLE 4-6
Block is active - PxxS.RUN 1-11, 1-35, 1-44, 1-45, 1-54
Block transition, velocity-optimal - G08 1-12, 1-61, 1-117
BOOL_BYTE 4-6
BOOL_DW 4-6
BOOL_WORD 4-6
Branch command if NC event is set - BES 2-17
Branch command, with stop - BST 1-53, 1-54, 1-55, 1-56, 1-64, 1-66, 1-67, 1-74, 1-84, 1-92, 1-97, 1-101, 1-123
Tool, selected 1-84, 1-94, 2-24, 2-25, 2-26, 2-44
Bring selected tool into position - MTP 1-84, 1-94, 2-24, 2-25, 2-26, 2-44
BST 1-53, 1-54, 1-55, 1-56, 1-64, 1-66, 1-67, 1-74, 1-84, 1-92, 1-97, 1-101, 1-123
BTXX 4-7
BTXX2 4-7
Bxx.021 1-86, 1-88, 1-89
Bxx.024 1-86, 1-88, 1-89
Bxx.032 2-43
Bxx.036 2-43, 2-44
Bxx.044 2-56
Bxx.072 1-87, 1-88, 1-89
BYTE_BCD_TO_INT 4-2
BYTE_BOOL 4-6
BYTE_TO_CHAR 4-4
BYTE_TO_GRAY 4-2
BYTE_TO_INT 4-2
BYTE_TO_SINT 4-6
BYTE_TO_USINT 4-2

C

Cancel tool transfer from gripper to magazine - XGM_CA 2-41, 4-3
Cancel tool transfer from gripper to spindle - XGS_CA 2-41, 4-3
Cancel tool transfer from magazine to gripper - XMG_CA 2-41, 4-3
Cancel tool transfer from magazine to spindle - XMS_CA 2-36, 2-39, 2-40, 2-41, 4-3
Cancel tool transfer from spindle to gripper - XSG_CA 2-41, 4-3
Cancel tool transfer from spindle to magazine - XSM_CA 2-41, 4-3
Change tool from spindle into magazine - TSM 2-22, 2-28, 2-32, 2-33, 2-34, 2-35, 4-1
Changeable software limits - AxxC.LIMIT 1-4, 1-126
CHAR_TO_BYTE 4-4
Clock generator - FLASH 4-6
CLOS_COM 3-23, 4-3
CLR_COM 4-6
Collision monitoring, single dimension, reading the activation state in the NC - COL_CTRL_S 2-57, 2-58, 4-6
Combined spindle / turret axis, number of turret locations - 65043 3-13
Combined spindle / turret axis, number of turret locations - P-7-3603 3-13
Combined spindle / turret axis, homing spindle required - P-7-3604 3-13
Combined spindle / turret axis, homing spindle required -65044 3-13
Combined spindle / turret axis, move reference switch of a Sauter turret - 65032 3-13
Combined spindle / turret axis, move reference switch of a Sauter turret - P-7-3592 3-13
Command position - PxxS.MGCP 1-12, 1-89, 1-90, 1-93, 1-94
Command spindle speed reached - AxxC.N_CMD 1-4, 1-99, 1-122
Command spindle speed reached - PxxC.N_CMD 1-99, 1-122
Comment (assembly instructions) - A00.057 3-31
Communication block for handheld terminal BTC06 with 64 IO - BTXX2 4-7
Communication block, between PLC and HMI control panel with serial port - BTXX 4-7
Communication function for safety function (IST) - SAVE_IO 4-7
Communication module, deleting receive and transmission buffer of a serial interface - CLR_COM 4-6
CONCAT_BYTE 4-2
CONCAT_S 4-4
CONCAT_WORD 4-2
Conditional stop - PxxC.M001 1-9, 1-44, 1-45
Constant cutting speed- G96 1-62, 1-139
Controller enabling signal - AxxS.RF 1-6, 1-109, 1-114, 1-115, 1-124, 1-125, 1-128, 1-129
COS_REAL 4-8
Counter, INT after IEC UP / DOWN - CTUD_INT 4-3, 4-9
Counter, UINT after IEC UP / DOWN - CTUD_UINT 4-3, 4-9
Counter, USINT after IEC UP / DOWN - CTUD_USINT 4-3, 4-9
CTRL_COM 3-23, 4-3
CTUD_INT 4-3, 4-9
CTUD_INT_INDR 4-3
CTUD_UINT 4-3, 4-9
CTUD_UINT_INDR 4-3
CTUD_USINT 4-3, 4-9
CTUD_USINT_INDR 4-3
Cutting speed, constant - G96 1-62, 1-139
Cxx.013 1-85
Cxx.059 1-126

D

D corrections, read - DCD_RD 3-1, 3-60, 3-63, 4-6
D corrections, write - DCD_WR 3-1, 3-60, 3-61, 3-63, 4-6
Data channel, finish data transmission - CLOS_COM 3-23, 4-3
Data channel, initialization - OPEN_COM 3-23, 4-3
Data exchange with drives - AXD 1-102, 2-54, 3-1, 3-9, 3-10, 3-12, 4-4, 4-9
DATE_RD 4-5
DC bus voltage OK - PxxC.UDRDY 1-9, 1-22, 1-29
DCD 3-1, 3-60, 3-61, 3-63, 4-6, 4-10
DCD_RD 3-1, 3-60, 3-63, 4-6
DCD_WR 3-1, 3-60, 3-61, 3-63, 4-6

Define mechanism - MxxS.DP 1-15, 1-64, 1-104
Define process - PxxC.DP 1-9, 1-63, 1-64
Define process - PxxS.DP 1-11, 1-63, 1-64, 1-104
DELETE 3-47, 3-56, 3-57, 4-4
Delete tool - TL_DELETE 1-95, 1-96, 3-1, 3-47, 3-56, 3-57, 4-6
DIAG_BOOL0 4-7
DIAG_BOOL16 4-7
DIAG_BOOL32 4-7
DIAG_BOOL4 4-7
DIAG_BOOL8 4-7
DIAG_BYTE 4-7
DIAG_CHAR 4-7
DIAG_DINT 4-7
DIAG_DWORD 4-7
DIAG_INT 4-7
DIAG_REAL 4-7
DIAG_SINT 4-7
DIAG_TIME 4-7
DIAG_UDINT 4-7
DIAG_UINT 4-7
DIAG_USINT 4-7
DIAG_WORD 4-7
Diagnosis functions (skipped for user) - DIAG_BOOL0 4-7
Diagnosis functions (skipped for user) - DIAG_BOOL16 4-7
Diagnosis functions (skipped for user) - DIAG_BOOL32 4-7
Diagnosis functions (skipped for user) - DIAG_BOOL4 4-7
Diagnosis functions (skipped for user) - DIAG_BOOL8 4-7
Diagnosis functions (skipped for user) - DIAG_BYTE 4-7
Diagnosis functions (skipped for user) - DIAG_CHAR 4-7
Diagnosis functions (skipped for user) - DIAG_DINT 4-7
Diagnosis functions (skipped for user) - DIAG_DWORD 4-7
Diagnosis functions (skipped for user) - DIAG_INT 4-7
Diagnosis functions (skipped for user) - DIAG_REAL 4-7
Diagnosis functions (skipped for user) - DIAG_SINT 4-7
Diagnosis functions (skipped for user) - DIAG_TIME 4-7
Diagnosis functions (skipped for user) - DIAG_UDINT 4-7
Diagnosis functions (skipped for user) - DIAG_UINT 4-7
Diagnosis functions (skipped for user) - DIAG_USINT 4-7
Diagnosis functions (skipped for user) - DIAG_WORD 4-7
Digital drive ready for operation - AxxS.BBDIG 1-6, 1-110, 1-129
DINT_TO_DWORD 4-5
DINT_TO_INT 4-5
DINT_TO_REAL 4-5
DINT_TO_TIME 4-5
DINT_TO_UDINT 4-5
DP 1-35, 1-50, 1-64
DPM_SLDIAG 4-7
DPM_START 4-7
DPM_STATE 4-7
DPM_STOP 4-7
Dry run mode - PxxC.DRYRN 1-10, 1-52
Dry run mode - PxxS.DRYRN 1-12, 1-60
DW_BOOL 4-6
DWORD_TO_DINT 4-5
DWORD_TO_REAL 4-7

E

E 1-1, 1-75, 1-94, 1-131, 2-10, 2-11, 2-12, 2-13, 3-25, 3-44, 3-47, 4-5
E auxiliary function number, reading the - E_NR 2-12, 2-13, 4-5
E auxiliary functions, acknowledging with specifying the auxiliary function number- E_FKT_Q 2-11, 2-13, 4-5
E auxiliary functions, acknowledging without specifying the auxiliary function number- E_ALL 2-12, 2-13, 4-5
E auxiliary functions, requesting with specifying the auxiliary function number- E_FKT 2-10, 2-11, 2-13, 4-5
E auxiliary functions, requesting without specifying the auxiliary function number- E_ALL 2-11, 2-12, 2-13, 4-5
E_ALL 2-11, 2-12, 2-13, 4-5
E_ALL_Q 2-12, 2-13, 4-5

E_FKT 2-10, 2-11, 2-13, 4-5
E_FKT_Q 2-11, 2-13, 4-5
E_NR 2-12, 2-13, 4-5
EMERGENCY STOP checking - machine - PxxC.EMACH 1-9, 1-19, 1-27
EMERGENCY STOP checking - station - PxxC.ESTAT 1-9, 1-19, 1-27
Enable movement - AxxS.SAFEN 1-7, 1-137
Enable tool storage for manual mode - MEN 1-84, 1-92
Enabling the tool magazine - PxxC.MGENA 1-10, 1-80, 1-81, 1-90
Enabling the tool magazine - PxxS.MGENA 1-11, 1-81, 1-90, 1-91
Equipment check ignores tool wear - PxxC.MGITW 1-10, 1-80, 1-83
Equipment check of tool management before version 19 PxxC.MGMTC - PxxC.MGNSL 1-10, 1-80, 1-82
Equipment check, deactivation of automatic - PxxC.MGWTC 1-10, 1-52, 1-80, 1-82
Error - PxxS.ERROR 1-11, 1-27, 1-49, 1-52, 1-53, 1-97, 2-43
Error clearing - PxxC.CLEAR 1-8, 1-9, 1-49, 2-43
Error handling MTC200 4-1
Error numbers
Data type USINT 4-10
EV_RES 2-13, 2-14, 2-15, 2-16, 4-1
EV_SET 2-13, 2-14, 2-15, 2-16, 4-1
EV_ST 2-13, 2-14, 2-15, 2-16, 4-1
EVENT 2-13, 2-14, 2-15, 2-16, 4-1
Events, conditional resetting of - EV_RES 2-13, 2-14, 2-15, 2-16, 4-1
Events, conditional setting of - EV_SET 2-13, 2-14, 2-15, 2-16, 4-1
Events, requests of - EVENT 2-13, 2-14, 2-15, 2-16, 4-1
Events, value transmission to - EV_ST 2-13, 2-14, 2-15, 2-16, 4-1
EXP_REAL 4-8
External 24 volt power supply - PxxC.EXT24 1-9, 1-18, 1-27

F

F_TRIG 4-3
Feed and speed limiting - AxxS.SPEED 1-6, 1-138
Feed override - AxxC.OVRD 1-5, 1-62, 1-123
Feed override - PxxC.FOVRD 1-10, 1-62, 1-63
Feed per rotation - G95 1-138
Feed, linear interpolation - G01 1-112
Filter functions, filter acceleration override - P-7-3584 3-13
Filter functions, filter acceleration override -65024 3-13
Filter functions, filter jerk limitation - P-7-3583 2-54, 3-13
Filter functions, filter jerk limiting - 65023 2-54, 3-13
Filter functions, master filter command position value - 65022 3-13
Filter functions, master filter command position value - P-7-3582 3-13
FIND 4-5
FLASH 4-6
Offset database, selection of 1-1, 1-50, 1-75, 1-103, 1-104, 1-108, 1-109
D correction, access to 3-1, 3-60, 3-61, 3-63, 4-6, 4-10
Tool data, access to 1-95, 1-96, 3-1, 3-28, 3-29, 3-38, 3-39, 3-40, 3-44, 3-46, 3-47, 4-5, 4-10
Function block for fast access to NC signal values - RT_DATA 4-7

G

G code for constant cutting code active - PxxS.G96 1-12, 1-62
G00 1-12, 1-61, 1-63
G01 1-112
G08 1-12, 1-61, 1-117
G26 1-60
G33 1-61, 1-138
G50 3-21
G51 3-21
G52 3-21
G54 3-21
G55 3-21
G56 3-21
G57 3-21
G58 3-21
G59 3-21
G63 1-61, 1-138

G64 1-61, 1-138
G65 1-61, 1-138
G74 1-12, 1-61
G95 1-138
G96 1-62, 1-139
Gear function, spindle angle - 65015 1-121, 3-13
Gear functions, spindle acceleration - 65014 1-121, 3-13
Gear selection - AxxC.GEARn 1-118
Gearbox functions, gear shifting - 65012 1-121, 3-13
Gearbox functions, gear shifting - P-7-3572 3-13
Gearbox functions, spindle acceleration - P-7-3574 3-13
Gearbox functions, spindle angle - P-7-3575 3-13
Gearbox functions, spindle speed spectification for spindle function - 65013 1-121, 3-13
Gearbox functions, spindle speed spectification for spindle function - P-7-3573 3-13
Geometry limit value - A00.060 3-35, 3-36
GRAY_TO_BYTE 4-2
Guard closed - PxxC.ESTPn 1-20
GUI_SK16 3-24, 3-25, 4-6

H

Handwheel position, transfer - HNDWHEEL 2-53, 2-54, 4-5
HIGH_BYTE 4-2
HIGH_WORD 4-2
HLT 1-56
HNDWHEEL 2-53, 2-54, 4-5
Home position required - Bxx.032 2-43
Homing of axis - AxxC.HOME 1-4, 1-34, 1-113, 1-114, 1-115, 1-130
Homing of tool magazine - PxxC.MGHOM 1-10, 1-80, 1-85
Homing switch - AxxC.HOMLS 1-4, 1-111
Homing, acknowledgment magazine - MRF_Q 2-20, 2-22, 2-23, 4-1
Homing, query magazine - MRF 1-84, 1-85, 2-20, 2-22, 2-23, 2-43, 4-1
Homing, tool memory - MRF 1-84, 1-85, 2-20, 2-22, 2-23, 2-43, 4-1

I

INSERT 4-4
INT, sign of an INT number - SIGN_INT 4-2
INT_TO_BCD_WORD 4-2
INT_TO_BYTE 4-2
INT_TO_DINT 4-5
INT_TO_SINT 4-6
INT_TO_STRING 4-4
INT_TO_UINT 4-6
INT_TO_USINT 4-2
INT_TO_WORD 4-2
Interpolation function, jerk limitation - 65010 3-13
Interpolation function, jerk limitation - P-7-3570 3-13
Interpolation function, mode for polygon curve - 65025 3-13
Interpolation function, mode for polygon curves - P-7-3585 3-13
Interpolation function, position window for block enabling in polygon mode 4 - 65081 3-14
Interpolation function, position window for block enabling in polygon mode 4 - P-7-3641 3-14
Interpolation function, switch droop compensation on/off - 65034 3-13
Interpolation function, switch droop compensation on/off - P-7-3594 3-13
Interruption of control voltage - PxxS.POWIN 1-11, 1-50, 1-57
IST functions, acknowledgement forced dynamization - P-7-3645 3-14
IST functions, default forced dynamization - 65084 3-14
IST functions, default forced dynamization - P-7-3644 3-14
IST functions, feedback forced dynamization -65085 3-14
IST functions, safe reference - 65087 3-14
IST functions, safe reference - P-7-3647 3-14

J

Jog mode - PxxC.JOGMn 1-35
Jogging axes causes resetting - Bxx.036 2-43, 2-44

Jogging, spindle postion - Cxx.059 1-126

L

LEFT 4-4
LEN 3-45, 4-4
Limiting of feed velocity / spindle speed - AxxC.SPEED 1-4, 1-127
Linear interpolation rapid traverse - G00 1-12, 1-61, 1-63
Linear interpolation, feed - G01 1-112
LN_REAL 4-7
Load,90% exceeded - AxxS.LD90 1-6, 1-133
Location positions, page no. for variable - Bxx.072 1-87, 1-88, 1-89
Location, programmed, move into position - MMP 1-84
Lock mechanism - MxxS.LP 1-15, 1-66, 1-105
Lock process - LP 1-35, 1-50, 1-66, 1-106
Lock process - PxxS.LP 1-11, 1-66, 1-106
LOG_REAL 4-7
LOW_BYTE 4-2
LOW_WORD 4-2
LP 1-35, 1-50, 1-66, 1-106

M

M auxiliary function acknowledging without specifying the auxiliary function number- M_ALL_Q 2-1, 2-2, 2-3, 2-12, 4-4
M auxiliary function number, reading the - M_NR 2-3, 2-4, 2-12, 4-4
M auxiliary functions, acknowledging with specifying the auxiliary function number- M_FKT_Q 2-1, 2-2, 2-3, 2-12, 4-1
M auxiliary functions, requesting with specifying the auxiliary function number- M_FKT 2-1, 2-2, 2-12, 4-1
M auxiliary functions, requesting without specifying the auxiliary function number- M_ALL 2-1, 2-2, 2-3, 2-12, 4-4
M key administration between user interface and PLC - GUI_SK16 3-24, 3-25, 4-6
M_ALL 2-1, 2-2, 2-3, 2-12, 4-4
M_ALL_Q 2-1, 2-2, 2-3, 2-12, 4-4
M_FKT 2-1, 2-2, 2-12, 4-1
M_FKT_Q 2-1, 2-2, 2-3, 2-12, 4-1
M_NR 2-3, 2-4, 2-12, 4-4
Machine data elements, read and write - MTD 1-127, 3-1, 3-17, 3-19, 4-5, 4-10
Machine data, read - MTD_RD 3-1, 3-17, 3-19, 4-5
Machine data, write - MTD_WR 1-127, 3-1, 3-17, 3-19, 4-5
MAG_ACT 2-42, 2-44, 4-3
MAG_Q 2-42, 2-44, 4-3
Magazine axis, acknowledging the preselection of a comb. spindle/turret axis - MAG_Q 2-42, 2-44, 4-3
Magazine axis, interrogating the preselection of a comb. spindle/turret axis - MAG_ACT 2-42, 2-44, 4-3
Magazine position, new - acknowledgement - MMV_Q 1-89, 1-93, 1-94, 2-21, 2-25, 2-26, 4-1
Magazine position, new - interrogation - MMV 1-89, 1-93, 1-94, 2-20, 2-21, 2-24, 2-25, 2-26, 4-1
Main drive ready - PxxC.BBMSP 1-9, 1-22, 1-29
Main load contactor closed - PxxC.PWRDY 1-9, 1-21, 1-29
MAKETIME 4-6
Manual mode, enable tool storage for - MEN 1-84, 1-92
Mathematical function, arc cosine function - ACOS_REAL 4-8
Mathematical function, arc sin function - ASIN_REAL 4-8
Mathematical function, arc tangent function - ATAN_REAL 4-8
Mathematical function, cosine function - COS_REAL 4-8
Mathematical function, exponential function - EXP_REAL 4-8
Mathematical function, logarithm naturalis LN - LN_REAL 4-7
Mathematical function, root - SQRT_REAL 4-7
Mathematical function, sine function - SIN_REAL 4-8
Mathematical function, tangent function - TAN_REAL 4-8
Mathematical functions, logarithm to base 10 LOG - LOG_REAL 4-7
MDI (Manual data input) active - PxxS.MDIAC 1-11, 1-57
Measured functions, measured idling torque - P-7-3650 3-14
Measured functions, measured standstill torque - P-7-3652 3-14

Measured functions, measuring period for idling torque determination - P-7-3649 3-14
Measured functions, measuring period for standstill torque determination - P-7-3651 3-14
Measuring function, triggering of oscilloscope function - 65026 3-13
Measuring function, triggering of oscilloscope function - P-7-3586 3-13
Measuring functions, activate measuring functions - 65035 3-13
Measuring functions, activate measuring functions - P-7-3595 3-13
Measuring functions, maximum measured moment peak - 65038 3-13
Measuring functions, maximum measured torque peak - P-7-3598 3-13
Measuring functions, maximum moment peak for tool breakage monitoring - 65037 3-13
Measuring functions, maximum moment peak for tool breakage monitoring - P-7-3597 3-13
Measuring functions, measured idling torque - 65090 3-14
Measuring functions, measured maximum lag error - 65041 3-13
Measuring functions, measured maximum lag error- P-7-3601 3-13
Measuring functions, measured standstill torque -65092 3-14
Measuring functions, measured torque distribution - P-7-3600 3-13
Measuring functions, measured torque distribution -65040 3-13
Measuring functions, measured torque value - 65036 3-13
Measuring functions, measured torque value - P-7-3596 3-13
Measuring functions, measuring period for idling torque determination - 65089 3-14
Measuring functions, measuring period for standstill torque determination -65091 3-14
Mechanism number - MxxS.PROC 1-15, 1-105
Mechanism program number - MxxS.PRGNR 1-15, 1-104, 1-105
MEN 1-84, 1-92
Message number, reading - MSG_RD 1-53, 2-17, 2-19, 4-1
Message number, writing - MSG_WR 2-17, 2-18, 4-1, 4-4
Message output with additional information as axis designation - MSG_WR_A 2-17, 2-18, 4-4
Message output with additional information as number MSG_WR_N 2-17, 4-4
MFP 1-84, 2-24
MHP 1-84, 1-88, 4-3
MHP_Q 4-3
MID 4-4
Minimize torsion moment - PxxC.SSnMT 1-101
MMA 1-84, 1-93
MMP 1-84
MMV 1-89, 1-93, 1-94, 2-20, 2-21, 2-24, 2-25, 2-26, 4-1
MMV_Q 1-89, 1-93, 1-94, 2-21, 2-25, 2-26, 4-1
Mode selection - PxxC.MODEn 1-30
MOP 1-84
Motion hold - AxxC.MHOLD 1-4, 1-117, 1-124, 1-125, 1-139, 2-43
Motion hold active - AxxS.MHOLD 1-6, 1-139
Move BYTE by n digits to the left - SHL_BYTE 4-2
Move BYTE by n digits to the right - SHR_BYTE 4-2
Basic position, tool magazine 1-84, 1-88, 4-3
Move programmed position into position - MMP 1-84
Position, free 1-84, 2-24
Move to free position - MFP 1-84, 2-24
Position, old 1-84
Move to old position - MOP 1-84
Move tool magazine in basic position - MHP 1-84, 1-88, 4-3
Move WORD by n digits to the left - SHL_WORD 4-2
Move WORD by n digits to the right - SHR_WORD 4-3
Moving the tool magazine by one location in the negative direction - PxxC.MGNEG 1-9, 1-80, 1-87, 1-88
Moving the tool magazine to its base position - PxxC.MGBP 1-9, 1-80, 1-85
Moving the tool storage unit by one location in the positive direction - PxxC.MGPOS 1-9, 1-80, 1-86, 1-88, 2-44
Moving tool - TL_MOVE 3-1, 3-47, 3-58, 3-59, 4-6
MRF 1-84, 1-85, 2-20, 2-22, 2-23, 2-43, 4-1
MRF_Q 2-20, 2-22, 2-23, 4-1
MRY 2-22, 2-23, 2-25, 2-26
MSG_RD 1-53, 2-17, 2-19, 4-1
MSG_WR 2-17, 2-18, 4-1, 4-4
MSG_WR_A 2-17, 2-18, 4-4

MSG_WR_N 2-17, 4-4
MTD 1-127, 3-1, 3-17, 3-19, 4-5, 4-10
MTD_RD 3-1, 3-17, 3-19, 4-5
MTD_WR 1-127, 3-1, 3-17, 3-19, 4-5
MTP 1-84, 1-94, 2-24, 2-25, 2-26, 2-44
Multi-function interpolator, activate 2^n encoder of a digital drive - 65062 3-14
Multi-function interpolator, activate 2^n encoder of a digital drive - P-7-3622 3-14
Multi-function interpolator, activate multi-function interpolator axis coupling - 65061 3-14
Multi-function interpolator, activate multi-function interpolator axis coupling - P-7-3621 3-14
Multifunction interpolator, coupling type for axis coupling on APR, interpolator 1 - 65053 3-14
Multifunction interpolator, coupling type for axis coupling on APR, interpolator 1 - P-7-3613 3-14
Multifunction interpolator, coupling type for axis coupling on APR, table 2 of multi-function interpolator - 65093 3-14
Multifunction interpolator, coupling type for axis coupling on APR, table 2 of multi-function interpolator - P-7-3653 3-14
Multifunction interpolator, coupling type for axis coupling on APR, table 3 of multi-function interpolator - P-7-3661 3-15
Multifunction interpolator, coupling type for axis coupling on APR, table 3 of multi-function interpolator -65101 3-15
Multifunction interpolator, coupling type for axis coupling on APR, table 4 of multi-function interpolator - P-7-3669 3-15
Multifunction interpolator, coupling type for axis coupling on APR, table 4 of multi-function interpolator -65109 3-15
Multifunction interpolator, length of APR values table 1 (interpolator 1) - P-7-3614 3-14
Multifunction interpolator, length of APR values table 1 (interpolator 1) -65054 3-14
Multifunction interpolator, length of APR values table 2 (interpolator 2) - P-7-3654 3-14
Multifunction interpolator, length of APR values table 2 (interpolator 2) -65094 3-14
Multifunction interpolator, length of APR values table 3 (interpolator 3) - P-7-3662 3-15
Multifunction interpolator, length of APR values table 3 (interpolator 3) -65102 3-15
Multifunction interpolator, length of APR values table 4 (interpolator 4) - P-7-3670 3-15
Multifunction interpolator, length of APR values table 4 (interpolator 4) -65110 3-15
Multifunction interpolator, multifunction interpolator mode (only coupling type 2) interpolator 1 - P-7-3619 3-14
Multifunction interpolator, multifunction interpolator mode (only coupling type 2) interpolator 1 -65059 3-14
Multifunction interpolator, multifunction interpolator mode (only coupling type 2) interpolator 2 - P-7-3659 3-15
Multifunction interpolator, multifunction interpolator mode (only coupling type 2) interpolator 2 -65099 3-15
Multifunction interpolator, multifunction interpolator mode (only coupling type 2) interpolator 3 - P-7-3667 3-15
Multifunction interpolator, multifunction interpolator mode (only coupling type 2) interpolator 3 -65107 3-15
Multifunction interpolator, multifunction interpolator mode (only coupling type 2) interpolator 4 - P-7-3675 3-15
Multifunction interpolator, multifunction interpolator mode (only coupling type 2) interpolator 4 -65115 3-15
Multi-function interpolator, number of support points for 3-D compensation - 65088 3-14
Multi-function interpolator, number of support points for 3-D compensation - P-7-3648 3-14
Multifunction interpolator, reference axis number (coupling type 2 or 4) of interpolator 1 - 65060 3-14
Multifunction interpolator, reference axis number (coupling type 2 or 4) of interpolator 1 - P-7-3620 3-14
Multifunction interpolator, reference axis number (coupling type 2) of interpolator 2 - P-7-3660 3-15

Multifunction interpolator, reference axis number (coupling type 2) of interpolator 2 -65100 3-15
Multifunction interpolator, reference axis number (coupling type 2) of interpolator 3 - P-7-3668 3-15
Multifunction interpolator, reference axis number (coupling type 2) of interpolator 3 -65108 3-15
Multifunction interpolator, reference axis number (coupling type 2) of interpolator 4 - P-7-3676 3-15
Multifunction interpolator, reference axis number (coupling type 2) of interpolator 4 -65116 3-15
Multifunction interpolator, start position of table 1 - 65057 3-14
Multifunction interpolator, start position of table 1 - P-7-3617 3-14
Multifunction interpolator, start position of table 2 - P-7-3657 3-15
Multifunction interpolator, start position of table 2 -65097 3-15
Multifunction interpolator, start position of table 3 - P-7-3665 3-15
Multifunction interpolator, start position of table 3 -65105 3-15
Multifunction interpolator, start position of table 4 - P-7-3673 3-15
Multifunction interpolator, start position of table 4 -65113 3-15
Multifunction interpolator, support position distance of table 1 (interpolator 1) - P-7-3618 3-14
Multifunction interpolator, support position distance of table 1 (interpolator 1) - 65058 3-14
Multifunction interpolator, support position distance of table 2 (interpolator 2) - P-7-3658 3-15
Multifunction interpolator, support position distance of table 2 (interpolator 2) - 65098 3-15
Multifunction interpolator, support position distance of table 3 (interpolator 3) - P-7-3666 3-15
Multifunction interpolator, support position distance of table 3 (interpolator 3) - 65106 3-15
Multifunction interpolator, support position distance of table 4 (interpolator 4) - P-7-3674 3-15
Multifunction interpolator, support position distance of table 4 (interpolator 4) - 65114 3-15
Multifunction interpolator, table factor for interpolator 1 - 65056 3-14
Multifunction interpolator, table factor for interpolator 1 - P-7-3616 3-14
Multifunction interpolator, table factor for interpolator 2 - P-7-3656 3-15
Multifunction interpolator, table factor for interpolator 2 -65096 3-15
Multifunction interpolator, table factor for interpolator 3 - P-7-3664 3-15
Multifunction interpolator, table factor for interpolator 3 -65104 3-15
Multifunction interpolator, table factor for interpolator 4 - P-7-3672 3-15
Multifunction interpolator, table factor for interpolator 4 -65112 3-15
Multifunction interpolator, transmission of table values for table 1 (interpolator 1) - P-7-3615 3-14
Multifunction interpolator, transmission of table values for table 1 (interpolator 1) - 65055 3-14
Multifunction interpolator, transmission of table values for table 2 (interpolator 2) - P-7-3655 3-15
Multifunction interpolator, transmission of table values for table 2 (interpolator 2) - 65095 3-15
Multifunction interpolator, transmission of table values for table 3 (interpolator 3) - P-7-3663 3-15
Multifunction interpolator, transmission of table values for table 3 (interpolator 3) - 65103 3-15
Multifunction interpolator, transmission of table values for table 4 (interpolator 4) - P-7-3671 3-15
Multifunction interpolator, transmission of table values for table 4 (interpolator 4) - 65111 3-15
MxxC.POK 1-14, 1-15, 1-104
MxxC.QP 1-15, 1-104
MxxS.AP 1-15, 1-105
MxxS.DP 1-15, 1-64, 1-104
MxxS.LP 1-15, 1-66, 1-105
MxxS.PRGNR 1-15, 1-104, 1-105
MxxS.PROC 1-15, 1-105
MxxS.RP 1-15, 1-105

N

NC blocks are skipped - PxxC.BLSKP 1-9, 1-41

NC program memory, interrogating the active - ACT_MEM 3-6, 3-8, 4-3
NC program memory, selecting the - SEL_MEM 3-6, 3-7, 4-3
NC program number specification - PxxS.PRGNR 1-12, 1-39, 1-105
NC program restart is active - PxxS.BPACT 1-11, 1-52, 1-59
NC variables, read - NCVAR_RD 3-1, 3-2, 3-3, 3-4, 4-6
NC variables, write - NCVAR_RD 3-1, 3-2, 3-3, 3-4, 4-6
NC_ENABLE 2-59, 4-9
NCVAR_RD 3-1, 3-2, 3-3, 3-4, 4-6
NCVAR_WR 3-1, 3-2, 3-3, 3-4, 4-6
Negative jogging - AxxC.JGNEG 1-4, 1-30, 1-34, 1-114, 1-115

O

O 1-1, 1-50, 1-75, 1-103, 1-104, 1-108, 1-109
OFF delay timer - TOFF 4-3
Offset data, read - OTD_RD 3-1, 3-20, 3-23, 4-5
Offset data, read and write - OTD 3-1, 3-20, 3-21, 3-23, 4-5, 4-10
Offset data, write - OTD_WR 3-1, 3-20, 3-21, 3-23, 4-5
Offset register - A00.056 3-35
OPEN_COM 3-23, 4-3
OTD 3-1, 3-20, 3-21, 3-23, 4-5, 4-10
OTD_RD 3-1, 3-20, 3-23, 4-5
OTD_WR 3-1, 3-20, 3-21, 3-23, 4-5

P

P 1-6, 1-136, 3-11, 3-13, 4-8
P-7-3560 3-11, 3-12
P-7-3565 3-13
P-7-3568 3-13
P-7-3569 3-13
P-7-3570 3-13
P-7-3571 3-13
P-7-3572 3-13
P-7-3573 3-13
P-7-3574 3-13
P-7-3575 3-13
P-7-3576 3-13
P-7-3577 3-13
P-7-3578 3-13
P-7-3582 3-13
P-7-3583 2-54, 3-13
P-7-3584 3-13
P-7-3585 3-13
P-7-3586 3-13
P-7-3587 3-13
P-7-3589 3-13
P-7-3590 3-13
P-7-3591 3-13
P-7-3592 3-13
P-7-3593 3-13
P-7-3594 3-13
P-7-3595 3-13
P-7-3596 3-13
P-7-3597 3-13
P-7-3598 3-13
P-7-3600 3-13
P-7-3601 3-13
P-7-3602 3-13
P-7-3603 3-13
P-7-3604 3-13
P-7-3605 3-13
P-7-3606 3-13
P-7-3607 3-13
P-7-3608 3-13
P-7-3609 3-14
P-7-3610 3-14
P-7-3611 3-14
P-7-3612 3-14
P-7-3613 3-14

P-7-3614 3-14
P-7-3615 3-14
P-7-3616 3-14
P-7-3617 3-14
P-7-3618 3-14
P-7-3619 3-14
P-7-3620 3-14
P-7-3621 3-14
P-7-3622 3-14
P-7-3623 3-14
P-7-3624 3-14
P-7-3625 3-14
P-7-3626 3-14
P-7-3627 3-14
P-7-3628 3-14
P-7-3629 3-14
P-7-3638 3-14
P-7-3640 3-14
P-7-3641 3-14
P-7-3642 3-14
P-7-3644 3-14
P-7-3645 3-14
P-7-3646 3-14
P-7-3647 3-14
P-7-3648 3-14
P-7-3649 3-14
P-7-3650 3-14
P-7-3651 3-14
P-7-3652 3-14
P-7-3653 3-14
P-7-3654 3-14
P-7-3655 3-15
P-7-3656 3-15
P-7-3657 3-15
P-7-3658 3-15
P-7-3659 3-15
P-7-3660 3-15
P-7-3661 3-15
P-7-3662 3-15
P-7-3663 3-15
P-7-3664 3-15
P-7-3665 3-15
P-7-3666 3-15
P-7-3667 3-15
P-7-3668 3-15
P-7-3669 3-15
P-7-3670 3-15
P-7-3671 3-15
P-7-3672 3-15
P-7-3673 3-15
P-7-3674 3-15
P-7-3675 3-15
P-7-3676 3-15
P-7-3678 3-15
P-7-3679 3-15
P-7-3680 3-15
P-7-3681 3-15
P-7-3682 3-15
P-7-3683 3-15
P-7-3684 3-15
P-7-3685 3-15
P-7-3686 3-15
P-7-3687 3-15
Parameter test transfer strobe - PxxC.PARAM 1-9, 1-34
Part machined - MxxC.POK 1-14, 1-15, 1-104
Part machined - PxxC.POK 1-9, 1-74, 1-75, 1-104
Part machined - PxxS.POK 1-11, 1-50, 1-74, 1-75, 1-104
POK 1-35, 1-50, 1-64, 1-66, 1-67, 1-74, 1-75, 1-104
Position spindle - AxxC.M19 1-4, 1-125
Position strobe - AxxC.STRBP 1-4, 1-111

Positive jogging 1-4, 1-30, 1-34, 1-114, 1-115
Power enablement - PxxS.POWEN 1-11, 1-27, 1-28, 1-29
Power is on - PxxS.POWER 1-11, 1-29, 2-43
Power on - PxxS.POWON 1-11, 1-25, 1-28, 1-29
Power request - PxxC.POWON 1-9, 1-25, 1-28
Power supply unit operational - PxxC.BBSUP 1-9, 1-20, 1-21, 1-27
Power supply unit, temperature control - PxxC.TPSUP 1-9, 1-21, 1-27
Process complete - POK 1-35, 1-50, 1-64, 1-66, 1-67, 1-74, 1-75, 1-104
Process control command, define process - DP 1-35, 1-50, 1-64
Process control command, lock process - LP 1-35, 1-50, 1-66, 1-106
Process control command, process complete - POK 1-35, 1-50, 1-64, 1-66, 1-67, 1-74, 1-75, 1-104
Process control command, select NC program for process - SP 1-35
Process control command, start advance program - AP 1-35, 1-44, 1-50, 1-67, 1-105
Process control commands, start reverse program - RP 1-35, 1-48, 1-50, 1-70, 1-105
Process control commands, wait for process - WP 1-35, 1-50, 1-104
Process enablement - PxxC.ENABL 1-9, 1-40
Process number - PxxS.PROC 1-12
Process, define - DP 1-35, 1-50, 1-64
Process, lock LP 1-35, 1-50, 1-66, 1-106
Process, wait for - WP 1-35, 1-50, 1-104
PROFIBUS, bus communication, start - DPM_START 4-7
PROFIBUS, bus communication, stop - DPM_STOP 4-7
PROFIBUS, single diagnosis of a PROFIBUS slaves - DPM_SLDIAG 4-7
PROFIBUS, status information of PROFIBUS master - DPM_STATE 4-7
PROFIBUS, status information with PROFIBUS process data exchange - DPM_EXCHG 4-7
Program active - PxxS.ACTIV 1-11, 1-31, 1-32, 1-33, 1-35, 1-50, 1-51, 1-53, 1-55, 1-60, 1-69, 1-70, 1-73
Program end with reset - RET 1-53, 1-54, 1-55, 1-56, 1-66, 1-74, 1-84, 1-92, 1-97, 1-101, 1-123
Program has been stopped - PxxS.STOP 1-11, 1-32, 1-33, 1-35, 1-44, 1-50, 1-56
Program number selection - PxxC.PRGNR 1-10, 1-39, 1-40, 1-105
Program stop - PxxC.STOP 1-9, 1-48
Offset, absolute 3-21
Offset, incremental 3-21
Programmable absolute offset - G50 3-21
Programmable acceleration - MMA 1-84, 1-93
Programmable incremental zero offset - G51 3-21
Programmable position contacts, preselection of the position contact data to transfer - 65119 3-15
Programmable position contacts, preselection of the position contact data to transfer - P-7-3679 3-15
Programmable position contacts, switching between position contacts and waypoints - 65118 3-15
Programmable position contacts, switching between position contacts and waypoints - P-7-3678 3-15
Programmable position contacts, transmission of the position contact data - 65120 3-15
Programmable position contacts, transmission of the position contact data - P-7-3680 3-15
Programmable position contacts, transmission of the position contact data - P-7-3681 3-15
Programmable position contacts, transmission of the position contact data - P-7-3682 3-15
Programmable position contacts, transmission of the position contact data - P-7-3683 3-15
Programmable position contacts, transmission of the position contact data - P-7-3684 3-15
Programmable position contacts, transmission of the position contact data - P-7-3685 3-15
Programmable position contacts, transmission of the position contact data - P-7-3686 3-15
Programmable position contacts, transmission of the position contact data - P-7-3687 3-15
Programmable position contacts, transmission of the position contact data - 65121 3-15

Programmable position contacts, transmission of the position contact data - 65122 3-15
Programmable position contacts, transmission of the position contact data - 65123 3-15
Programmable position contacts, transmission of the position contact data - 65124 3-15
Programmable position contacts, transmission of the position contact data - 65125 3-15
Programmable position contacts, transmission of the position contact data - 65126 3-15
Programmable position contacts, transmission of the position contact data - 65127 3-15
Programmed halt HLT 1-56
Programmed workpiece zero point - G52 3-21
PxxC.ADV 1-9, 1-30, 1-31, 1-32, 1-41, 1-42, 1-45, 1-46, 1-51, 1-57, 1-69
PxxC.AP 1-9, 1-67, 1-70, 1-74
PxxC.BBMSP 1-9, 1-22, 1-29
PxxC.BBSUP 1-9, 1-20, 1-21, 1-27
PxxC.BLSKP 1-9, 1-41
PxxC.BPSTR 1-10, 1-52
PxxC.CLEAR 1-8, 1-9, 1-49, 2-43
PxxC.DP 1-9, 1-63, 1-64
PxxC.DRYRN 1-10, 1-52
PxxC.EMACH 1-9, 1-19, 1-27
PxxC.ENABL 1-9, 1-40
PxxC.ESTAT 1-9, 1-19, 1-27
PxxC.ESTPn 1-20
PxxC.EXT24 1-9, 1-18, 1-27
PxxC.FOVRD 1-10, 1-62, 1-63
PxxC.JOGMn 1-35
PxxC.LINE 1-9, 1-18, 1-27
PxxC.M001 1-9, 1-44, 1-45
PxxC.MGAP 1-10, 1-80, 1-89, 1-93, 1-94
PxxC.MGBP 1-9, 1-80, 1-85
PxxC.MGENA 1-10, 1-80, 1-81, 1-90
PxxC.MGHOM 1-10, 1-80, 1-85
PxxC.MGITW 1-10, 1-80, 1-83
PxxC.MGMAN 1-10, 1-80, 1-81, 1-83, 1-84, 1-93
PxxC.MGNEG 1-9, 1-80, 1-87, 1-88
PxxC.MGNSL 1-10, 1-80, 1-82
PxxC.MGNTL 1-10, 1-80, 1-83
PxxC.MGPOS 1-9, 1-80, 1-86, 1-88, 2-44
PxxC.MGWTC 1-10, 1-52, 1-80, 1-82
PxxC.MODEn 1-30
PxxC.N_CMD 1-99, 1-122
PxxC.PARAM 1-9, 1-34
PxxC.POK 1-9, 1-74, 1-75, 1-104
PxxC.POWON 1-9, 1-25, 1-28
PxxC.PRGNR 1-10, 1-39, 1-40, 1-105
PxxC.PWRDY 1-9, 1-21, 1-29
PxxC.QP 1-9, 1-74
PxxC.RAPID 1-9, 1-38
PxxC.REPOS 1-10, 1-51, 1-59
PxxC.RESTA 1-10, 1-51, 1-59
PxxC.REV 1-9, 1-30, 1-32, 1-33, 1-41, 1-45, 1-46, 1-70, 1-73
PxxC.ROVRD 1-10, 1-63
PxxC.RP 1-9, 1-70, 1-74
PxxC.SCONn 1-100
PxxC.SINGL 1-9, 1-37, 1-41
PxxC.SOVRD 1-62, 1-63
PxxC.SP 1-9, 1-39, 1-40, 1-98, 1-99
PxxC.SPHLT 1-99
PxxC.SPSTP 1-98
PxxC.SSnMT 1-101
PxxC.SSnON 1-101
PxxC.STOP 1-9, 1-48
PxxC.TPSUP 1-9, 1-21, 1-27
PxxC.UDRDY 1-9, 1-22, 1-29
PxxS.ACTIV 1-11, 1-31, 1-32, 1-33, 1-35, 1-50, 1-51, 1-53, 1-55, 1-60, 1-69, 1-70, 1-73

PxxS.AP 1-11, 1-44, 1-67, 1-69, 1-70, 1-105
PxxS.BPACT 1-11, 1-52, 1-59
PxxS.CREST 1-11, 1-59
PxxS.DP 1-11, 1-63, 1-64, 1-104
PxxS.DRYRN 1-12, 1-60
PxxS.ERROR 1-11, 1-27, 1-49, 1-52, 1-53, 1-97, 2-43
PxxS.EXCTH 1-12, 1-60
PxxS.G00 1-12, 1-61
PxxS.G08 1-12, 1-61
PxxS.G74 1-12, 1-61
PxxS.G96 1-12, 1-62
PxxS.LP 1-11, 1-66, 1-106
PxxS.MDIAC 1-11, 1-57
PxxS.MGBP 1-12, 1-86, 1-90, 1-93
PxxS.MGCP 1-12, 1-89, 1-90, 1-93, 1-94
PxxS.MGENA 1-11, 1-81, 1-90, 1-91
PxxS.MGERR 1-11, 1-50, 1-90, 1-95, 1-97
PxxS.MGMAN 1-11, 1-84, 1-90, 1-92, 1-93
PxxS.MGMOV 1-11, 1-90, 1-92
PxxS.MGREQ 1-11, 1-50, 1-90, 1-91, 1-93
PxxS.MGTWO 1-11, 1-50, 1-90, 1-94, 1-95, 1-96, 1-97
PxxS.MGWRN 1-11, 1-50, 1-90, 1-95, 1-96, 1-97
PxxS.POK 1-11, 1-50, 1-74, 1-75, 1-104
PxxS.POWEN 1-11, 1-27, 1-28, 1-29
PxxS.POWER 1-11, 1-29, 2-43
PxxS.POWIN 1-11, 1-50, 1-57
PxxS.POWON 1-11, 1-25, 1-28, 1-29
PxxS.PRGNR 1-12, 1-39, 1-105
PxxS.PROC 1-12
PxxS.QP 1-11, 1-44, 1-70, 1-73, 1-74, 1-104
PxxS.READY 1-11, 1-31, 1-32, 1-42, 1-50, 1-53, 1-59
PxxS.REPOS 1-11, 1-59
PxxS.REV 1-11, 1-31, 1-32, 1-33, 1-35, 1-50, 1-55
PxxS.RP 1-11, 1-48, 1-70, 1-72, 1-73, 1-105
PxxS.RVACT 1-11, 1-56
PxxS.SCONn 1-100
PxxS.STOP 1-11, 1-32, 1-33, 1-35, 1-44, 1-50, 1-56
PxxS.THMIS 1-12, 1-60
PxxS.TRANS 1-11, 1-53
PxxS.TREAT 1-61

Q

Q auxiliary function acknowledging without specifying the auxiliary function number - Q_ALL_Q 2-9, 2-10, 2-13, 4-4
Q auxiliary functions, acknowledgement with specifying the auxiliary function number - Q_FKT_Q 2-9, 2-10, 2-13, 4-1
Q auxiliary functions, requesting with specifying the auxiliary function number - Q_FKT 2-9, 2-10, 2-13, 4-1
Q auxiliary functions, requesting without specifying the auxiliary function number - Q_ALL 2-9, 2-10, 2-13, 4-4
Q Function number, reading the - Q_NR 2-10, 2-13, 4-4
Q_ALL 2-9, 2-10, 2-13, 4-4
Q_ALL_Q 2-9, 2-10, 2-13, 4-4
Q_FKT 2-9, 2-10, 2-13, 4-1
Q_FKT_Q 2-9, 2-10, 2-13, 4-1
Q_NR 2-10, 2-13, 4-4

R

R_TRIG 4-3
Rapid override - PxxC.ROVRD 1-10, 1-63
Rapid traverse mode - PxxC.RAPID 1-9, 1-38
Rapid traverse speed for spindle - AxxC.RAPID 1-3, 1-4, 1-126
Rapid traverse with G0 active - PxxS.G00 1-12, 1-61
Rapid traverse, linear interpolation - G00 1-12, 1-61, 1-63
RD_BYTE 3-23, 4-3
RD_STR 4-6, 4-10
Read and write machine data elements - MTD 1-127, 3-1, 3-17, 3-19, 4-5, 4-10
Read and write offset data - OTD 3-1, 3-20, 3-21, 3-23, 4-5, 4-10

Read date - DATE_RD 4-5
Readiness to start - PxxS.READY 1-11, 1-31, 1-32, 1-42, 1-50, 1-53, 1-59
Reading a SERCOS real-time status bit - AxxS.IDDS 1-6, 1-131, 1-132
Reading time of day - TOD_RD 4-5
REAL_TO_DINT 4-5
REAL_TO_DWORD 4-7
REAL_TO_STRING 4-5
Recognition of a falling edge - F_TRIG 4-3
Recognition of a rising edge - R_TRIG 4-3
Reduced motor torque on positive stop - P-7-3578 3-13
Reference cam for safe referencing - AxxC.SAFRS 1-5, 1-137
Reference marker, offset 1 to - Bxx.021 1-86, 1-88, 1-89
Reference marker, offset 4 to - Bxx.024 1-86, 1-88, 1-89
Reference position - Cxx.013 1-85
Referencing axles - G74 1-12, 1-61
Referencing, axles - G74 1-12, 1-61
Removing high bytes from WORD - HIGH_BYTE 4-2
Removing high word from DWORD - HIGH_WORD 4-2
Removing Low-Bytes from WORD - LOW_BYTE 4-2
Removing low-word from DWORD - LOW_WORD 4-2
REPLACE 4-4
Repositioning / restart active - PxxS.REPOS 1-11, 1-59
Repositioning / restart terminated - PxxS.CREST 1-11, 1-59
Repositioning to the contour - PxxC.REPOS 1-10, 1-51, 1-59
Required data, reading of - AXD_RD 3-1, 3-9, 3-10, 3-12, 4-4
Required data, writing of - AXD_WR 2-54, 3-1, 3-9, 3-12, 4-4
RES_MEAS 4-7
Reset tool - TL_RESET 1-95, 1-96, 3-1, 3-47, 3-54, 3-55, 4-5
Reset, jogging axes causes - Bxx.036 2-43, 2-44
Reset, program end - RET 1-53, 1-54, 1-55, 1-56, 1-66, 1-74, 1-84, 1-92, 1-97, 1-101, 1-123
RET 1-53, 1-54, 1-55, 1-56, 1-66, 1-74, 1-84, 1-92, 1-97, 1-101, 1-123
Returning to the contour - PxxC.RESTA 1-10, 1-51, 1-59
REV 1-56, 2-56, 2-57, 4-6
REV_SYNC 2-56, 2-57, 4-6
Reverse program active - PxxS.REV 1-31
Reverse Program Active - PxxS.REV 1-11, 1-31, 1-32, 1-33, 1-35, 1-50, 1-55
Reverse program enabling - PxxC.RP 1-9, 1-70, 1-74
Reverse program enabling - PxxS.RP 1-11, 1-48, 1-70, 1-72, 1-73, 1-105
Reverse program start - MxxS.RP 1-15, 1-105
Reverse program start - PxxC.REV 1-9, 1-30, 1-32, 1-33, 1-41, 1-45, 1-46, 1-70, 1-73
Reverse vector active - PxxS.RVACT 1-11, 1-56
Reverse vector, set - REV 1-56, 2-56, 2-57, 4-6
Revolver, synchronous swiveling - REV_SYNC 2-56, 2-57, 4-6
RIGHT 4-4
Rigid tapping - G63 1-61, 1-138
Rigid tapping - G64 1-61, 1-138
Rigid, tapping - G63 1-61, 1-138
Rigid, tapping - G64 1-61, 1-138
ROL_BYTE 4-3
ROL_WORD 4-3
ROR_BYTE 4-3
ROR_WORD 4-3
Rotary axis mode preselection, acknowledgement of a main spindle with rotary axis capability - ROTMOD_Q 2-47, 2-49, 4-4
Rotary axis mode preselection, request of a main spindle with rotary axis capability - ROTMOD 2-47, 2-49, 4-4
Rotate BYTE by n digits to the left - ROL_BYTE 4-3
Rotate BYTE by n digits to the right - ROR_BYTE 4-3
Rotate WORD by n digits to the left - ROL_WORD 4-3
Rotate WORD by n digits to the right - ROR_WORD 4-3
ROTMOD 2-47, 2-49, 4-4
ROTMOD_Q 2-47, 2-49, 4-4
RP 1-35, 1-48, 1-50, 1-70, 1-105
RS 4-3
RS_FLIP_FLOP, dominating reset - RS 4-3
RS_FLIP_FLOP, dominating setting - SR 4-3
RT_DATA 4-7

S

S auxiliary function number, reading the - S_NR 2-6, 2-12, 4-4
S auxiliary functions, acknowledgement without specifying the auxiliary function number- S_ALL_Q 2-5, 2-6, 2-13, 4-4
S auxiliary functions, acknowledging with specifying the auxiliary function number- S_FKT_Q 2-5, 2-7, 2-13, 4-1
S auxiliary functions, requesting with specifying the auxiliary function number- S_FKT 2-4, 2-5, 2-7, 2-13, 4-1
S auxiliary functions, requesting without specifying the auxiliary function number- S_ALL 2-5, 2-6, 2-13, 4-4
S_ALL 2-5, 2-6, 2-13, 4-4
S_ALL_Q 2-5, 2-6, 2-13, 4-4
S_FKT 2-4, 2-5, 2-7, 2-13, 4-1
S_FKT_Q 2-5, 2-7, 2-13, 4-1
S_NR 2-6, 2-12, 4-4
S-0-0036 1-134
S-0-0038 1-135
S-0-0039 1-135
S-0-0040 1-134
S-0-0047 1-136
S-0-0057 1-136
S-0-0084 1-136
S-0-0091 1-135
S-0-0124 1-135
S-0-0126 1-136
S-0-0157 1-134
S-0-0158 1-137
S-0-0183 1-134
S-0-0192 3-10
S-0-0228 1-134
S-0-0303 1-116
S-0-0307 1-132
S-0-0401 1-132
S-0-0405 1-116
Safe position switch 1 - AxxS.SAFP1 1-7, 1-137
Safe position switch 2 - AxxS.SAFP2 1-7, 1-137
Safe position switch 3 - AxxS.SAFP3 1-7, 1-137
Safe position switch 4 - AxxS.SAFP4 1-7, 1-137
Safety function active - AxxS.SAFAC 1-7, 1-137
Safety limit switch - AxxC.OTRVL 1-4, 1-27, 1-107
SAVE_IO 4-7
SEL_MEM 3-6, 3-7, 4-3
Select NC program for process - SP 1-35
Selecting the offset database for G54 .. G59 - O 1-1, 1-50, 1-75, 1-103, 1-104, 1-108, 1-109
SERCOS functions, adapting SERCOS third-party drives - 65064 3-14
SERCOS functions, adapting SERCOS third-party drives - P-7-3624 3-14
SERCOS functions, length of SERCOS fiber optic loop - 65078 3-14
SERCOS functions, length of SERCOS fiber optic loop - P-7-3638 3-14
SERCOS functions, SERCOS drive telegram fault counter. Number of drive telegrams - 65069 3-14
SERCOS functions, SERCOS drive telegram fault counter. Number of drive telegrams - P-7-3629 3-14
SERCOS functions, SERCOS test mode - 65068 3-14
SERCOS functions, SERCOS test mode - P-7-3628 3-14
Set reverse vector - REV 1-56, 2-56, 2-57, 4-6
SHL_BYTE 4-2
SHL_WORD 4-2
SHR_BYTE 4-2
SHR_WORD 4-3
SIGN_INT 4-2
SIN_REAL 4-8
Single step mode - PxxC.SINGL 1-9, 1-37, 1-41
SINT_TO_BYTE 4-6
SINT_TO_INT 4-6
Software limits activated - AxxS.LIMIT 1-6, 1-139
SP 1-35
SPDL_ACT 2-42, 2-46, 4-4
SPDL_Q 2-42, 2-46, 4-4

Spindle command speed reached - AxxS.N_CMD 1-6, 1-134
Spindle functions, actual spindle speed - 65080 3-14
Spindle functions, linear axis as spindle drive - 65008 1-126, 3-13
Spindle functions, linear axis as spindle drive - P-7-3568 3-13
Spindle functions, maximum spindle speed - 65052 3-14
Spindle functions, maximum spindle speed - P-7-3612 3-14
Spindle functions, position control mode switching rotary speed in speed control - 65030 3-13
Spindle functions, position control mode switching rotary speed in speed control - P-7-3590 3-13
Spindle functions, reference angle position ‘DDS as spindle drive’ - 65009 3-13
Spindle functions, reference angle position ‘DDS as spindle drive’ - P-7-3569 3-13
Spindle functions, spindle function parameter - 65011 3-13
Spindle functions, spindle function parameter - P-7-3571 3-13
Spindle functions, spindle list speed - P-7-3640 3-14
Spindle functions, switching speed for switching to curved acceleration characteristic of a spindle in position control - 65086 3-14
Spindle functions, switching speed for switching to curved acceleration characteristic of a spindle in position control - P-7-3646 3-14
Spindle is in position - AxxS.INPOS 1-6, 1-136
Spindle mode preselection, acknowledgement of a main spindle with rotary axis capability - SPMOD_Q 2-47, 2-48, 2-49, 4-4
Spindle mode preselection, request of a main spindle with rotary axis capability - SPMOD 2-47, 2-48, 2-49, 4-4
Spindle override - PxxC.SOVRD 1-62, 1-63
Spindle position jogging - Cxx.059 1-126
Spindle power output (P > Px) - AxxS.P_PX 1-6, 1-136, 1-137
Spindle speed 0 reached - AxxS.N_MIN 1-6, 1-135
Spindle stop (Mx5) - AxxC.M5 1-4, 1-124, 1-125
Spindle stop upon control reset - AxxC.SPRST 1-4, 1-123
Spindle stop upon process stop - AxxC.SPSTP 1-4, 1-98, 1-121, 2-57
Spindle stop upon process stop - PxxC.SPSTP 1-98
Spindle stop upon program end - AxxC.SPSTE 1-4, 1-122
Spindle stop upon stopped NC program - AxxC.SPHLT 1-4, 1-44, 1-99, 1-122
Spindle stop upon stopped NC program - PxxC.SPHLT 1-99
Spindle synchronous run - AxxS.SYNC 1-6, 1-134
Spindle, acknowledgement of preselection of a combined spindle/turret axis - SPDL_Q 2-42, 2-46, 4-4
Spindle, interrogating the preselection of a combined spindle/turret axis - SPDL_ACT 2-42, 2-46, 4-4
SPMOD 2-47, 2-48, 2-49, 4-4
SPMOD_Q 2-47, 2-48, 2-49, 4-4
SQRT_REAL 4-7
SR 4-3
Start advance program - AP 1-35, 1-44, 1-50, 1-67, 1-105
Start advance program - PxxC.AP 1-9, 1-67, 1-70, 1-74
Start NC program restart - PxxC.BPSTR 1-10, 1-52
Start reverse program - RP 1-35, 1-48, 1-50, 1-70, 1-105
STING, left-bound L_ character - LEFT 4-4
STING, right-bound L_ character - RIGHT 4-4
STRING, as of P_th character L_ character - MID 4-4
STRING, as of P_th delete character L_ character - DELETE 3-47, 3-56, 3-57, 4-4
STRING, as of P_th replace character L_ character by IN2_ - REPLACE 4-4
STRING, character string IN2_ in IN1_ search - FIND 4-5
STRING, inserting a character string after the L_th character - INSERT 4-4
STRING, length of a - LEN 3-45, 4-4
STRING, merging two - CONCAT_S 4-4
STRING, reading via serial interface - RD_STR 4-6, 4-10
STRING, writing via serial interface - WR_STR 4-6, 4-10
STRING_TO_INT - STRING_TO_INT 4-4
STRING_TO_REAL - STRING_TO_REAL 4-5
Switch on adaptive feed control - G26 1-60
Switching completed - AxxS.SAFRY 1-7, 1-137
Synchronization of AXD and NC initialization - NC_ENABLE 2-59, 4-9
Synchronous axes compound is activated - PxxS.SCONn 1-100
Synchronous axes, angular deviation of the synchronous spindle from the master spindle with active spindle synchronization - 65067 3-14

Synchronous axes, angular deviation of the synchronous spindle from the master spindle with active spindle synchronization - P-7-3627 3-14
Synchronous axes, P factor for spindle synchronization compensation controller - 65042 3-13
Synchronous axes, P factor for spindle synchronization compensation controller - P-7-3602 3-13
Synchronous axes, slave spindle offset angle - 65029 3-13
Synchronous axes, slave spindle offset angle - P-7-3589 3-13

T

T 1-61, 1-82, 1-97, 2-7, 2-8, 2-13, 3-45, 4-1, 4-4, 4-5, 4-8
T auxiliary function number, reading the - T_NR 2-8, 2-13, 3-45, 4-5
T auxiliary functions, acknowledgement without specifying the auxiliary function number- T_ALL_Q 2-8, 2-13, 4-4
T auxiliary functions, acknowledging with specifying the auxiliary function number- T_FKT_Q 2-7, 2-13, 4-1
T auxiliary functions, requesting with specifying the auxiliary function number - T_FKT 2-7, 2-13, 4-1
T auxiliary functions, requesting without specifying the auxiliary function number- T_ALL 2-7, 2-8, 2-13, 4-4
T_ALL 2-7, 2-8, 2-13, 4-4
T_ALL_Q 2-8, 2-13, 4-4
T_FKT 2-7, 2-13, 4-1
T_FKT_Q 2-7, 2-13, 4-1
T_NR 2-8, 2-13, 3-45, 4-5
TAN_REAL 4-8
Tapping spindle as lead axis - G65 1-61, 1-138
TCH 2-21, 2-28, 2-29, 2-30, 2-31, 2-35, 4-1
TCH_Q 2-21, 2-29, 2-30, 2-31, 4-1
Thermal protection switch - AxxC.MTAS 1-4, 1-27, 1-108
Thread cutting or thread tapping active - PxxS.TREAT 1-61
Threading - G33 1-61, 1-138
TIME_DAY 4-6
TIME_HOUR 4-6
TIME_MIN 4-6
TIME_MS 4-6
TIME_SEC 4-6
TIME_TO_DINT 4-5
Timer impulse - TP 4-3
Timer startup delay - TON 4-3
TL_DELETE 1-95, 1-96, 3-1, 3-47, 3-56, 3-57, 4-6
TL_ENABLE 3-1, 3-46, 3-49, 3-50, 4-5
TL_MOVE 3-1, 3-47, 3-58, 3-59, 4-6
TL_RESET 1-95, 1-96, 3-1, 3-47, 3-54, 3-55, 4-5
TLBD_RD 3-1, 3-47, 3-51, 3-53, 4-5
TLBD_WR 3-1, 3-46, 3-47, 3-48, 4-5
TLD 1-95, 1-96, 3-1, 3-28, 3-29, 3-38, 3-39, 3-40, 3-44, 3-46, 3-47, 4-5, 4-10
TLD_RD 3-1, 3-28, 3-29, 3-38, 3-39, 3-40, 3-44, 3-46, 3-47, 4-5
TLD_WR 1-95, 1-96, 3-1, 3-28, 3-29, 3-38, 3-39, 3-44, 3-46, 3-47, 4-5
TLED_RD 3-1, 3-47, 3-51, 3-52, 3-53, 4-5
TLED_WR 3-1, 3-46, 3-47, 3-48, 4-5
TMP1MEAS 4-7
TMS 2-22, 2-28, 2-31, 2-32, 2-33, 2-35, 4-1
TMS_Q 2-22, 2-31, 2-32, 2-33, 2-35, 4-1
TOD_RD 4-5
TOFF 4-3
TOGGLE 4-6
TON 4-3
Tool base data, reading - TLBD_RD 3-1, 3-47, 3-51, 3-53, 4-5
Tool base data, writing - TLBD_WR 3-1, 3-46, 3-47, 3-48, 4-5
Tool call, tool selection and- T 1-61, 1-82, 1-97, 2-7, 2-8, 2-13, 3-45, 4-1, 4-4, 4-5, 4-8
Tool change from the magazine to the spindle - TMS 2-22, 2-28, 2-31, 2-32, 2-33, 2-35, 4-1
Tool change of magazine to spindle, request - TMS 2-22, 2-28, 2-31, 2-32, 2-33, 2-35, 4-1
Tool change of spindle to magazine, request - TSM 2-22, 2-28, 2-32, 2-33, 2-34, 2-35, 4-1
Tool change, acknowledgement - TCH_Q 2-21, 2-29, 2-30, 2-31, 4-1

Tool change, complete, execute - TCH 2-21, 2-28, 2-29, 2-30, 2-31, 2-35, 4-1
Tool change, request - TCH 2-21, 2-28, 2-29, 2-30, 2-31, 2-35, 4-1
Tool data, enable - TL_ENABLE 3-1, 3-46, 3-49, 3-50, 4-5
Tool data, reading - TLD_RD 3-1, 3-28, 3-29, 3-38, 3-39, 3-40, 3-44, 3-46, 3-47, 4-5
Tool data, write - TLD_WR 1-95, 1-96, 3-1, 3-28, 3-29, 3-38, 3-39, 3-44, 3-46, 3-47, 4-5
Tool edge data, reading - TLED_RD 3-1, 3-47, 3-51, 3-52, 3-53, 4-5
Tool edge data, writing - TLBD_WR 3-1, 3-46, 3-47, 3-48, 4-5
Tool edge user data 1, designation of - A00.070 3-36
Tool edge user data 10, designation of - A00.096 3-36
Tool edge user data 2, designation of - A00.071 3-36
Tool edge user data 3, designation of - A00.072 3-36
Tool edge user data 4, designation of - A00.073 3-36
Tool edge user data 5, designation of - A00.074 3-36
Tool edge user data 6, designation of - A00.092 3-36
Tool edge user data 7, designation of - A00.093 3-36
Tool edge user data 8, designation of - A00.094 3-36
Tool edge user data 9, designation of - A00.095 3-36
Tool error status - PxxS.MGERR 1-11, 1-50, 1-90, 1-95, 1-97
Tool group status, reading - TLG_RD 3-1, 3-28, 3-41, 3-43, 4-6
Tool group status, writing - TLG_WR 3-1, 3-28, 3-41, 3-43, 4-6
Tool life count, deactivating the - PxxC.MGNTL 1-10, 1-80, 1-83
Tool life data - A00.059 3-35
Tool life monitoring
  Tool worn out - PxxS.MGTWO 1-11, 1-50, 1-90, 1-94, 1-95, 1-96, 1-97
  warn limit reached - PxxS.MGWRN 1-11, 1-50, 1-90, 1-95, 1-96, 1-97
Tool magazine homing - MRF 1-84, 1-85, 2-20, 2-22, 2-23, 2-43, 4-1
Tool magazine in motion - PxxS.MGMOV 1-11, 1-90, 1-92
Tool magazine in program or manual mode - PxxS.MGMAN 1-11, 1-84, 1-90, 1-92, 1-93
Tool magazine is in base position - PxxS.MGBP 1-12, 1-86, 1-90, 1-93
Tool magazine mode - PxxC.MGMAN 1-10, 1-80, 1-81, 1-83, 1-84, 1-93
Tool magazine request from the NC - PxxS.MGREQ 1-11, 1-50, 1-90, 1-91, 1-93
Tool selection and tool call - T 1-61, 1-82, 1-97, 2-7, 2-8, 2-13, 3-45, 4-1, 4-4, 4-5, 4-8
Tool selection- E 1-1, 1-75, 1-94, 1-131, 2-10, 2-11, 2-12, 2-13, 3-25, 3-44, 3-47, 4-5
Tool storage offset 1 to reference marker - Bxx.021 1-86, 1-88, 1-89
Tool storage offset 4 to reference marker - Bxx.024 1-86, 1-88, 1-89
Tool transfer from gripper to magazine is not permitted - XGM_NA 2-41, 4-2
Tool transfer from gripper to magazine permitted - XGM_PA 2-41, 4-2
Tool transfer from gripper to magazine, acknowledgement - XGM_Q 2-41, 4-2
Tool transfer from gripper to spindle not permitted - XGS_NA 2-41, 4-2
Tool transfer from gripper to spindle permitted - XGS_PA 2-41, 4-2
Tool transfer from gripper to spindle, acknowledgement - XGS_Q 2-41, 4-2
Tool transfer from magazine to gripper is not permitted - XMG_NA 2-41, 4-2
Tool transfer from magazine to gripper permitted - XMG_PA 2-41, 4-2
Tool transfer from magazine to gripper, acknowledgement - XMG_Q 2-41, 4-2
Tool transfer from magazine to spindle not permitted - XMS_NA 2-36, 2-38, 2-39, 2-40, 2-41, 4-1
Tool transfer from magazine to spindle permitted - XMS_PA 2-37, 2-39, 2-40, 2-41, 4-1
Tool transfer from magazine to spindle, acknowledgement - XMS_Q 2-36, 2-38, 2-39, 2-40, 2-41, 4-1
Tool transfer from spindle to gripper not permitted - XSG_NA 2-41, 4-2
Tool transfer from spindle to gripper permitted - XSG_PA 2-41, 4-2
Tool transfer from spindle to gripper, acknowledgement - XSG_Q 2-41, 4-2
Tool transfer from spindle to magazine not permitted - XSM_NA 2-41, 4-2
Tool transfer from spindle to magazine permitted - XSM_PA 2-41, 4-2
Tool transfer from spindle to magazine, acknowledgement - XSM_Q 2-41, 4-2
Tool transfer of gripper to magazine, initializing - XGM 2-36, 2-41, 4-2, 4-3
Tool transfer of gripper to spindle, initializing - XGS 2-36, 2-41, 4-2, 4-3
Tool transfer of magazine to gripper, initializing - XMG 2-36, 2-41, 4-2, 4-3
Tool transfer of magazine to spindle, initializing - XMS 2-36, 2-37, 2-38, 2-39, 2-40, 2-41, 4-1, 4-3
Tool transfer of spindle to gripper, initializing - XSG 2-36, 2-41, 4-2, 4-3
Tool transfer of spindle to magazine, initializing - XSM 2-36, 2-41, 4-1, 4-2, 4-3
Tool transfer, de-activate checking- XFER_CHK 2-41, 2-42, 4-4
Tool user data 1, designation of - A00.061 3-31

Tool user data 2, designation of - A00.062 3-31
Tool user data 3, designation of - A00.063 3-31
Tool user data 4, designation of - A00.064 3-31
Tool user data 5, designation of - A00.065 3-31
Tool user data 6, designation of - A00.066 3-31
Tool user data 7, designation of - A00.067 3-31
Tool user data 8, designation of - A00.068 3-31
Tool user data 9, designation of - A00.069 3-31
TP 4-3
Transfer channel, reading a BYTE on a general - RD_BYTE 3-23, 4-3
Transfer channel, writing a BYTE on a general - WR_BYTE 3-23, 4-3
Transformation active - PxxS.TRANS 1-11, 1-53
Transmission channel, request of error number of general - CTRL_COM 3-23, 4-3
Transmit function, additional reference offset of the feed axis with active transmit function - 65066 3-14
Transmit function, additional reference offset of the feed axis with active transmit function - P-7-3626 3-14
Transmit function, switch-off of speed reduction when using the transmit function or Triaglide/rod kinematics - 65063 3-14
Transmit function, switch-off of speed reduction when using the transmit function or Triaglide/rod kinematics - P-7-3623 3-14
TSM 2-22, 2-28, 2-32, 2-33, 2-34, 2-35, 4-1
TSM_Q 2-22, 2-32, 2-34, 2-35, 4-1
Turret movement, asynchronous - Bxx.044 2-56
Type conversion BOOL -> BYTE - BOOL_BYTE 4-6
Type conversion BOOL -> DOUBLE WORD - BOOL_DW 4-6
Type conversion BOOL -> WORD - BOOL_WORD 4-6
Type conversion BYTE -> BOOL - BYTE_BOOL 4-6
Type conversion BYTE -> CHAR - BYTE_TO_CHAR 4-4
Type conversion WORD -> BOOL - WORD_BOOL 4-6
Type conversion BCD code, BYTE, 2 digits -> INTEGER - BYTE_BCD_TO_INT 4-2
Type conversion BCD code, Word, 4 digits -> INTEGER - WORD_BCD_TO_INT 4-2
Type conversion BYTE -> INTEGER - BYTE_TO_INT 4-2
Type conversion BYTE -> UNSIGNED SHORT INTEGER - BYTE_TO_USINT 4-2
Type conversion BYTE in SINT number - BYTE_TO_SINT 4-6
Type conversion CHAR -> BYTE - CHAR_TO_BYTE 4-4
Type conversion DOUBLE INTEGER -> UNSIGNED DOUBLE INTEGER - DINT_TO_UDINT 4-5
Type conversion DOUBLE INTEGER -> DOUBLE WORD - DINT_TO_DWORD 4-5
Type conversion DOUBLE INTEGER -> INTEGER - DINT_TO_INT 4-5
Type conversion DOUBLE INTEGER -> REAL - DINT_TO_REAL 4-5
Type conversion DOUBLE INTEGER -> time - DINT_TO_TIME 4-5
Type conversion DOUBLE WORD -> BOOL - DW_BOOL 4-6
Type conversion DOUBLE WORD -> DOUBLE INTEGER - DWORD_TO_DINT 4-5
Type conversion DWORD -> REAL - DWORD_TO_REAL 4-7
Type conversion GRAY code -> hexadecimal - GRAY_TO_BYTE 4-2
Type conversion hexadecimal -> GRAY Code - BYTE_TO_GRAY 4-2
Type conversion INT in SINT number - INT_TO_SINT 4-6
Type conversion INTEGER (day, hour, minute, second, millisecond) in TIME - MAKETIME 4-6
Type conversion INTEGER -> DOUBLE INTEGER - INT_TO_DINT 4-5
Type conversion INTEGER -> STRING - INT_TO_STRING 4-4
Type conversion INTEGER -> UNSIGNED SHORT INTEGER - INT_TO_USINT 4-2
Type conversion INTEGER number -> BYTE - INT_TO_BYTE 4-2
Type conversion INTEGER number -> BYTE - INT_TO_BYTE 4-2
Type conversion INTEGER number ->4-digit BCD cod. Word - INT_TO_BCD_WORD 4-2
Type conversion INTEGER UNSIGNED ->INTEGER - INT_TO_UINT 4-6
Type conversion REAL -> STRING - REAL_TO_STRING 4-5
Type conversion REAL -> DOUBLE INTEGER - REAL_TO_DINT 4-5
Type conversion REAL -> DWORD - REAL_TO_DWORD 4-7
Type conversion SINT in BYTE number - SINT_TO_BYTE 4-6
Type conversion SINT in INT number - SINT_TO_INT 4-6

Type conversion STRING -> REAL - STRING_TO_REAL 4-5
Type conversion STRING -> INTEGER - STRING_TO_INT 4-4
Type conversion time -> DOUBLE INTEGER - TIME_TO_DINT 4-5
Type conversion TIME into INTEGER (day value) - TIME_DAY 4-6
Type conversion TIME into INTEGER (hour value) - TIME_HOUR 4-6
Type conversion TIME into INTEGER (millisecond value) - TIME_MS 4-6
Type conversion TIME into INTEGER (minute value) - TIME_MIN 4-6
Type conversion TIME into INTEGER (second value) - TIME_SEC 4-6
Type conversion UNSIGNED DOUBLE INTEGER -> DOUBLE INTEGER - UDINT_TO_DINT 4-5
Type conversion UNSIGNED INTEGER -> INTEGER - UINT_TO_INT 4-6
Type conversion UNSIGNED INTEGER -> WORD - UINT_TO_WORD 4-6
Type conversion UNSIGNED SHORT INTEGER -> BYTE - USINT_TO_BYTE 4-2
Type conversion UNSIGNED SHORT INTEGER -> INTEGER - USINT_TO_INT 4-2
Type conversion WORD -> INTEGER - WORD_TO_INT 4-2
Type conversion WORD -> UNSIGNED INTEGER - WORD_TO_UINT 4-2

U

UDINT_TO_DINT 4-5
UINT_TO_INT 4-6
UINT_TO_WORD 4-6
Up/down counter, INTEGER range - CTUD_INT_INDR 4-3
Up/down counter, UNSIGNED INTEGER range - CTUD_UINT_INDR 4-3
Up/down counter, UNSIGNED SHORT INTEGER range - CTUD_USINT_INDR 4-3
USERBOF 4-4
USINT_TO_BYTE 4-2
USINT_TO_INT 4-2

V

Velocity command value exceeding - AxxS.N_MAX 1-6, 1-135
Velocity-optimal block transition - G08 1-12, 1-61, 1-117
Velocity-optimized block transition active - PxxS.G08 1-12, 1-61
VLT_MEAS 4-7

W

Wait for process - WP 1-35, 1-50, 1-104
Wait until Position is Approached - MRY 2-22, 2-23, 2-25, 2-26
Waypoints - AxxS.WPn 1-131
Wear factors - A00.058 3-36
Wear register - A00.055 3-35
Werkzeug bewegen - TL_MOVE 3-47
Werkzeug löschen - TL_DELETE 3-47
Werkzeuggrunddaten, lesen - TLBD_RD 3-47
Werkzeugschneidendaten, lesen - TLED_RD 3-47
WORD_BCD_TO_INT 4-2
WORD_BOOL 4-6
WORD_TO_INT 4-2
Workpiece zero point, programmable - G52 3-21
WP 1-35, 1-50, 1-104
WR_BYTE 3-23, 4-3
WR_STR 4-6, 4-10
Write and read offset data - OTD 3-1, 3-20, 3-21, 3-23, 4-5, 4-10
Writing upon a SERCOS real time control bit - AxxC.QDDS 1-4, 1-116

X

XFER_CHK 2-41, 2-42, 4-4
XGM 2-36, 2-41, 4-2, 4-3
XGM_CA 2-41, 4-3
XGM_NA 2-41, 4-2
XGM_PA 2-41, 4-2
XGM_Q 2-41, 4-2
XGS 2-36, 2-41, 4-2, 4-3

XGS_CA 2-41, 4-3
XGS_NA 2-41, 4-2
XGS_PA 2-41, 4-2
XGS_Q 2-41, 4-2
XMG 2-36, 2-41, 4-2, 4-3
XMG_CA 2-41, 4-3
XMG_NA 2-41, 4-2
XMG_PA 2-41, 4-2
XMG_Q 2-41, 4-2
XMS 2-36, 2-37, 2-38, 2-39, 2-40, 2-41, 4-1, 4-3
XMS_CA 2-36, 2-39, 2-40, 2-41, 4-3
XMS_NA 2-36, 2-38, 2-39, 2-40, 2-41, 4-1
XMS_PA 2-37, 2-39, 2-40, 2-41, 4-1
XMS_Q 2-36, 2-38, 2-39, 2-40, 2-41, 4-1
XSG 2-36, 2-41, 4-2, 4-3
XSG_CA 2-41, 4-3
XSG_NA 2-41, 4-2
XSG_PA 2-41, 4-2
XSG_Q 2-41, 4-2
XSM 2-36, 2-41, 4-1, 4-2, 4-3
XSM_CA 2-41, 4-3
XSM_NA 2-41, 4-2
XSM_PA 2-41, 4-2
XSM_Q 2-41, 4-2

Z

Zero offset, adjustable - G54 3-21
Zero offset, adjustable - G55 3-21
Zero offset, adjustable - G56 3-21
Zero offset, adjustable - G57 3-21
Zero offset, adjustable - G58 3-21
Zero offset, adjustable - G59 3-21

6 Service & Support

6.1 Helpdesk

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main steht Ihnen mit Rat und Tat zur Seite. Sie erreichen uns

Our service helpdesk at our headquarters in Lohr am Main, Germany can assist you in all kinds of inquiries. Contact us

- telefonisch - by phone: **49 (0) 9352 40 50 60**
  über Service Call Entry Center Mo-Fr 07:00-18:00
  - via Service Call Entry Center Mo-Fr 7:00 am - 6:00 pm
- per Fax - by fax: **+49 (0) 9352 40 49 41**
- per e-Mail - by e-mail: **service.svc@boschrexroth.de**

6.2 Service-Hotline

Außerhalb der Helpdesk-Zeiten ist der Service direkt ansprechbar unter

After helpdesk hours, contact our service department directly at

+49 (0) 171 333 88 26

oder - or **+49 (0) 172 660 04 06**

6.3 Internet

Unter **www.boschrexroth.com** finden Sie ergänzende Hinweise zu Service, Reparatur und Training sowie die **aktuellen** Adressen *) unserer auf den folgenden Seiten aufgeführten Vertriebs- und Servicebüros.

☐ Verkaufsniederlassungen
☐ Niederlassungen mit Kundendienst

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit unserem für Sie nächstgelegenen Ansprechpartner auf.

*) Die Angaben in der vorliegenden Dokumentation können seit Drucklegung überholt sein.

At **www.boschrexroth.com** you may find additional notes about service, repairs and training in the Internet, as well as the **actual** addresses *) of our sales- and service facilities figuring on the following pages.

☐ sales agencies
☐ offices providing service

Please contact our sales / service office in your area first.

*) Data in the present documentation may have become obsolete since printing.

6.4 Vor der Kontaktaufnahme... - Before contacting us...

Wir können Ihnen schnell und effizient helfen wenn Sie folgende Informationen bereithalten:

1. detaillierte Beschreibung der Störung und der Umstände.
2. Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern.
3. Tel.-/Faxnummern und e-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind.

For quick and efficient help, please have the following information ready:

1. Detailed description of the failure and circumstances.
2. Information on the type plate of the affected products, especially type codes and serial numbers.
3. Your phone/fax numbers and e-mail address, so we can contact you in case of questions.

6.5 Kundenbetreuungsstellen - Sales & Service Facilities

Deutschland – Germany

vom Ausland: (0) nach Landeskennziffer weglassen!
from abroad: don't dial (0) after country code!

Vertriebsgebiet Mitte Germany Centre Rexroth Indramat GmbH Bgm.-Dr.-Nebel-Str. 2 / Postf. 1357 97816 Lohr am Main / 97803 Lohr **Kompetenz-Zentrum Europa** Tel.: +49 (0)9352 40-0 Fax: +49 (0)9352 40-4885	**SERVICE** **CALL ENTRY CENTER** **MO – FR** **von 07:00 - 18:00 Uhr** **from 7 am – 6 pm** **Tel. +49 (0) 9352 40 50 60** service.svc@boschrexroth.de	**SERVICE** **HOTLINE** **MO – FR** **von 17:00 - 07:00 Uhr** **from 5 pm - 7 am** **+ SA / SO** **Tel.: +49 (0)172 660 04 06** **oder / or** **Tel.: +49 (0)171 333 88 26**	**SERVICE** **ERSATZTEILE / SPARES** verlängerte Ansprechzeit - extended office time - ♦ nur an Werktagen - only on working days - ♦ von 07:00 - 18:00 Uhr - from 7 am - 6 pm - **Tel. +49 (0) 9352 40 42 22**
Vertriebsgebiet Süd Germany South Bosch Rexroth AG Landshuter Allee 8-10 80637 München Tel.: +49 (0)89 127 14-0 Fax: +49 (0)89 127 14-490	Vertriebsgebiet West Germany West Bosch Rexroth AG Regionalzentrum West Borsigstrasse 15 40880 Ratingen Tel.: +49 (0)2102 409-0 Fax: +49 (0)2102 409-406 +49 (0)2102 409-430	Gebiet Südwest Germany South-West Bosch Rexroth AG Service-Regionalzentrum Süd-West Siemensstr.1 70736 Fellbach Tel.: +49 (0)711 51046–0 Fax: +49 (0)711 51046–248	
Vertriebsgebiet Nord Germany North Bosch Rexroth AG Walsroder Str. 93 30853 Langenhagen Tel.: +49 (0) 511 72 66 57-0 Service: +49 (0) 511 72 66 57-256 Fax: +49 (0) 511 72 66 57-93 Service: +49 (0) 511 72 66 57-783	Vertriebsgebiet Mitte Germany Centre Bosch Rexroth AG Regionalzentrum Mitte Waldecker Straße 13 64546 Mörfelden-Walldorf Tel.: +49 (0) 61 05 702-3 Fax: +49 (0) 61 05 702-444	Vertriebsgebiet Ost Germany East Bosch Rexroth AG Beckerstraße 31 09120 Chemnitz Tel.: +49 (0)371 35 55-0 Fax: +49 (0)371 35 55-333	Vertriebsgebiet Ost Germany East Bosch Rexroth AG Regionalzentrum Ost Walter-Köhn-Str. 4d 04356 Leipzig Tel.: +49 (0)341 25 61-0 Fax: +49 (0)341 25 61-111

Europa (West) - Europe (West)

vom Ausland: (0) nach Landeskennziffer weglassen, from abroad: don't dial (0) after country code,

Italien: 0 nach Landeskennziffer mitwählen Italy: dial 0 after country code

Austria - Österreich	Austria – Österreich	Belgium - Belgien	Denmark - Dänemark
Bosch Rexroth GmbH Electric Drives & Controls Stachegasse 13 1120 Wien Tel.: +43 (0)1 985 25 40 Fax: +43 (0)1 985 25 40-93	Bosch Rexroth GmbH Electric Drives & Controls Industriepark 18 4061 Pasching Tel.: +43 (0)7221 605-0 Fax: +43 (0)7221 605-21	Bosch Rexroth AG Electric Drives & Controls Industrielaan 8 1740 Ternat Tel.: +32 (0)2 5830719 - service: +32 (0)2 5830717 Fax: +32 (0)2 5830731 service@boschrexroth.be	BEC A/S Zinkvej 6 8900 Randers Tel.: +45 (0)87 11 90 60 Fax: +45 (0)87 11 90 61
Great Britain – Großbritannien	Finland - Finnland	France - Frankreich	France - Frankreich
Bosch Rexroth Ltd. Electric Drives & Controls Broadway Lane, South Cerney Cirencester, Glos GL7 5UH Tel.: +44 (0)1285 863000 Fax: +44 (0)1285 863030 sales@boschrexroth.co.uk service@boschrexroth.co.uk	Bosch Rexroth Oy Electric Drives & Controls Ansatie 6 017 40 Vantaa Tel.: +358 (0)9 84 91-11 Fax: +358 (0)9 84 91-13 60	Bosch Rexroth SAS Electric Drives & Controls Avenue de la Trentaine (BP. 74) 77503 Chelles Cedex Tel.: +33 (0)164 72-70 00 Fax: +33 (0)164 72-63 00 **Hotline:** +33 (0)608 33 43 28	Bosch Rexroth SAS Electric Drives & Controls ZI de Thibaud, 20 bd. Thibaud (BP. 1751) 31084 Toulouse Tel.: +33 (0)5 61 43 61 87 Fax: +33 (0)5 61 43 94 12
France – Frankreich	Italy - Italien	Italy - Italien	Italy - Italien
Bosch Rexroth SAS Electric Drives & Controls 91, Bd. Irène Joliot-Curie 69634 Vénissieux – Cedex Tel.: +33 (0)4 78 78 53 65 Fax: +33 (0)4 78 78 53 62	Bosch Rexroth S.p.A. Via G. Di Vittoria, 1 20063 Cernusco S/N.MI Tel.: +39 02 92 365 1 +39 02 92 365 326 Fax: +39 02 92 365 500 +39 02 92 365 516378	Bosch Rexroth S.p.A. Via Paolo Veronesi, 250 10148 Torino Tel.: +39 011 224 88 11 Fax: +39 011 224 88 30	Bosch Rexroth S.p.A. Via del Progresso, 16 (Zona Ind.) 35020 Padova Tel.: +39 049 8 70 13 70 Fax: +39 049 8 70 13 77
Italy - Italien	Italy - Italien	Netherlands - Niederlande/Holland	Netherlands – Niederlande/Holland
Bosch Rexroth S.p.A. Via Mascia, 1 80053 Castellamare di Stabia NA Tel.: +39 081 8 71 57 00 Fax: +39 081 8 71 68 85	Bosch Rexroth S.p.A. Via Isonzo, 61 40033 Casalecchio di Reno (Bo) Tel.: +39 051 29 86 430 Fax: +39 051 29 86 490	Bosch Rexroth Services B.V. Technical Services Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0) 411 65 16 40 +31 (0) 411 65 17 27 Fax: +31 (0) 411 67 78 14 +31 (0) 411 68 28 60 services@boschrexroth.nl	Bosch Rexroth B.V. Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0) 411 65 19 51 Fax: +31 (0) 411 65 14 83 www.boschrexroth.nl
Norway - Norwegen	Spain - Spanien	Spain – Spanien	Sweden - Schweden
Bosch Rexroth AS Electric Drives & Controls Berghagan 1 or: Box 3007 1405 Ski-Langhus 1402 Ski Tel.: +47 (0)64 86 41 00 Fax: +47 (0)64 86 90 62 jul.ruud@rexroth.no	Bosch Rexroth S.A. Electric Drives & Controls Centro Industrial Santiga Obradors s/n 08130 Santa Perpetua de Mogoda Barcelona Tel.: +34 9 37 47 94 00 Fax: +34 9 37 47 94 01	Goimendi S.A. Electric Drives & Controls Parque Empresarial Zuatzu C/ Francisco Grandmontagne no.2 20018 San Sebastian Tel.: +34 9 43 31 84 21 - service: +34 9 43 31 84 56 Fax: +34 9 43 31 84 27 - service: +34 9 43 31 84 60 sat.indramat@goimendi.es	Bosch Rexroth AB Electric Drives & Controls - Varuvägen 7 (Service: Konsumentvägen 4, Älfsjö) 125 81 Stockholm Tel.: +46 (0)8 727 92 00 Fax: +46 (0)8 647 32 77
Sweden - Schweden	Switzerland West - Schweiz West	Switzerland East - Schweiz Ost	
Bosch Rexroth AB Electric Drives & Controls Ekvändan 7 254 67 Helsingborg Tel.: +46 (0) 42 38 88 -50 Fax: +46 (0) 42 38 88 -74	Bosch Rexroth Suisse SA Electric Drives & Controls Rue du village 1 1020 Renens Tel.: +41 (0)21 632 84 20 Fax: +41 (0)21 632 84 21	Bosch Rexroth Schweiz AG Electric Drives & Controls Hemrietstrasse 2 8863 Buttikon Tel. +41 (0) 55 46 46 111 Fax +41 (0) 55 46 46 222	

Rexroth
Bosch Group

Europa (Ost) - Europe (East)

vom Ausland: (0) nach Landeskennziffer weglassen
from abroad: don't dial (0) after country code

Czech Republic - Tschechien Bosch -Rexroth, spol.s.r.o. Hviezdoslavova 5 627 00 Brno Tel.: +420 (0)5 48 126 358 Fax: +420 (0)5 48 126 112	Czech Republic - Tschechien DEL a.s. Strojírenská 38 591 01 Zdar nad Sázavou Tel.: +420 566 64 3144 Fax: +420 566 62 1657	Hungary - Ungarn Bosch Rexroth Kft. Angol utca 34 1149 Budapest Tel.: +36 (1) 422 3200 Fax: +36 (1) 422 3201	Poland – Polen Bosch Rexroth Sp.zo.o. ul. Staszica 1 05-800 Pruszków Tel.: +48 22 738 18 00 – service: +48 22 738 18 46 Fax: +48 22 758 87 35 – service: +48 22 738 18 42
Poland – Polen Bosch Rexroth Sp.zo.o. Biuro Poznan ul. Dabrowskiego 81/85 60-529 Poznan Tel.: +48 061 847 64 62 /-63 Fax: +48 061 847 64 02	Romania - Rumänien East Electric S.R.L. Bdul Basarabia no.250, sector 3 73429 Bucuresti Tel./Fax:: +40 (0)21 255 35 07 +40 (0)21 255 77 13 Fax: +40 (0)21 725 61 21 eastel@rdsnet.ro	Romania - Rumänien Bosch Rexroth Sp.zo.o. Str. Drobety nr. 4-10, app. 14 70258 Bucuresti, Sector 2 Tel.: +40 (0)1 210 48 25 +40 (0)1 210 29 50 Fax: +40 (0)1 210 29 52	Russia - Russland Bosch Rexroth OOO Wjatskaja ul. 27/15 127015 Moskau Tel.: +7-095-785 74 78 +7-095 785 74 79 Fax: +7 095 785 74 77 laura.kanina@boschrexroth.ru
Russia - Russland ELMIS 10, Internationalnaya 246640 Gomel, Belarus Tel.: +375/ 232 53 42 70 +375/ 232 53 21 69 Fax: +375/ 232 53 37 69 elmis_ltd@yahoo.com	Turkey - Türkei Bosch Rexroth Otomasyon San & Tic. A..S. Fevzi Cakmak Cad No. 3 34295 Sefaköy - Istanbul Tel.: +90 212 413 34-00 Fax: +90 212 413 34-17	Slowenia - Slowenien DOMEL Otoki 21 64 228 Zelezniki Tel.: +386 5 5117 152 Fax: +386 5 5117 225 brane.ozebek@domel.si	

Africa, Asia, Australia – incl. Pacific Rim

Australia - Australien AIMS - Australian Industrial Machinery Services Pty. Ltd. 28 Westside Drive Laverton North Vic 3026 Melbourne Tel.: +61 3 93 59 0228 Fax: +61 3 93 59 0286 Hotline: +61 4 19 369 195 enquires@aimservices.com.au	Australia - Australien Bosch Rexroth Pty. Ltd. No. 7, Endeavour Way Braeside Victoria, 31 95 Melbourne Tel.: +61 3 95 80 39 33 Fax: +61 3 95 80 17 33 mel@rexroth.com.au	China Shanghai Bosch Rexroth Hydraulics & Automation Ltd. Waigaoqiao, Free Trade Zone No.122, Fu Te Dong Yi Road Shanghai 200131 - P.R.China Tel.: +86 21 58 66 30 30 Fax: +86 21 58 66 55 23 richard.yang_sh@boschrexroth.com.cn gf.zhu_sh@boschrexroth.com.cn	China Shanghai Bosch Rexroth Hydraulics & Automation Ltd. 4/f, Marine Tower No.1, Pudong Avenue Shanghai 200120 - P.R.China Tel: +86 21 68 86 15 88 Fax: +86 21 58 40 65 77
China Bosch Rexroth China Ltd. 15/F China World Trade Center 1, Jianguomenwai Avenue Beijing 100004, P.R.China Tel.: +86 10 65 05 03 80 Fax: +86 10 65 05 03 79	China Bosch Rexroth China Ltd. Guangzhou Repres. Office Room 1014-1016, Metro Plaza, Tian He District, 183 Tian He Bei Rd Guangzhou 510075, P.R.China Tel.: +86 20 8755-0030 +86 20 8755-0011 Fax: +86 20 8755-2387	China Bosch Rexroth (China) Ltd. A-5F., 123 Lian Shan Street Sha He Kou District Dalian 116 023, P.R.China Tel.: +86 411 46 78 930 Fax: +86 411 46 78 932	China Melchers GmbH BRC-SE, Tightening & Press-fit 13 Floor Est Ocean Centre No.588 Yanan Rd. East 65 Yanan Rd. West Shanghai 200001 Tel.: +86 21 6352 8848 Fax: +86 21 6351 3138
Hongkong Bosch Rexroth (China) Ltd. 6th Floor, Yeung Yiu Chung No.6 Ind Bldg. 19 Cheung Shun Street Cheung Sha Wan, Kowloon, Hongkong Tel.: +852 22 62 51 00 Fax: +852 27 41 33 44 alexis.siu@boschrexroth.com.hk	India - Indien Bosch Rexroth (India) Ltd. Electric Drives & Controls Plot. No.96, Phase III Peenya Industrial Area Bangalore – 560058 Tel.: +91 80 51 17 0-211...-218 Fax: +91 80 83 94 345 +91 80 83 97 374 mohanvelu.t@boschrexroth.co.in	India - Indien Bosch Rexroth (India) Ltd. Electric Drives & Controls Advance House, II Floor Ark Industrial Compound Narol Naka, Makwana Road Andheri (East), Mumbai - 400 059 Tel.: +91 22 28 56 32 90 +91 22 28 56 33 18 Fax: +91 22 28 56 32 93 singh.op@boschrexroth.co.in	India - Indien Bosch Rexroth (India) Ltd. S-10, Green Park Extension New Delhi – 110016 Tel.: +91 11 26 56 65 25 +91 11 26 56 65 27 Fax: +91 11 26 56 68 87 koul.rp@boschrexroth.co.in
Indonesia - Indonesien PT. Bosch Rexroth Building # 202, Cilandak Commercial Estate Jl. Cilandak KKO, Jakarta 12560 Tel.: +62 21 7891169 (5 lines) Fax: +62 21 7891170 - 71 rudy.karimun@boschrexroth.co.id	Japan Bosch Rexroth Automation Corp. Service Center Japan Yutakagaoka 1810, Meito-ku, NAGOYA 465-0035, Japan Tel.: +81 52 777 88 41 +81 52 777 88 53 +81 52 777 88 79 Fax: +81 52 777 89 01	Japan Bosch Rexroth Automation Corp. Electric Drives & Controls 2F, I.R. Building Nakamachidai 4-26-44, Tsuzuki-ku YOKOHAMA 224-0041, Japan Tel.: +81 45 942 72 10 Fax: +81 45 942 03 41	Korea Bosch Rexroth-Korea Ltd. Electric Drives and Controls Bongwoo Bldg. 7FL, 31-7, 1Ga Jangchoong-dong, Jung-gu Seoul, 100-391 Tel.: +82 234 061 813 Fax: +82 222 641 295
Korea Bosch Rexroth-Korea Ltd. 1515-14 Dadae-Dong, Saha-Gu Electric Drives & Controls Pusan Metropolitan City, 604-050 Tel.: +82 51 26 00 741 Fax: +82 51 26 00 747 gyhan@rexrothkorea.co.kr	Malaysia Bosch Rexroth Sdn.Bhd. 11, Jalan U8/82, Seksyen U8 40150 Shah Alam Selangor, Malaysia Tel.: +60 3 78 44 80 00 Fax: +60 3 78 45 48 00 hockhwa@hotmail.com rexroth1@tm.net.my	Singapore - Singapur Bosch Rexroth Pte Ltd 15D Tuas Road Singapore 638520 Tel.: +65 68 61 87 33 Fax: +65 68 61 18 25 sanjay.nemade @boschrexroth.com.sg	South Africa - Südafrika TECTRA Automation (Pty) Ltd. 71 Watt Street, Meadowdale Edenvale 1609 Tel.: +27 11 971 94 00 Fax: +27 11 971 94 40 Hotline: +27 82 903 29 23 georgv@tectra.co.za
Taiwan Bosch Rexroth Co., Ltd. Taichung Branch 1F., No. 29, Fu-Ann 5th Street, Xi-Tun Area, Taichung City Taiwan, R.O.C. Tel : +886 - 4 -23580400 Fax: +886 - 4 -23580402 charlie.chen@boschrexroth.com.tw jim.lin@boschrexroth.com.tw david.lai@boschrexroth.com.tw	Thailand NC Advance Technology Co. Ltd. 59/76 Moo 9 Ramintra road 34 Tharang, Bangkhen, Bangkok 10230 Tel.: +66 2 943 70 62 +66 2 943 71 21 Fax: +66 2 509 23 62 sonkawin@hotmail.com		

Nordamerika – North America

USA Headquarters - Hauptniederlassung Bosch Rexroth Corporation Electric Drives & Controls 5150 Prairie Stone Parkway Hoffman Estates, IL 60192-3707 Tel.: +1 847 6 45 36 00 Fax: +1 847 6 45 62 01 servicebrc@boschrexroth-us.com repairbrc@boschrexroth-us.com	USA Central Region - Mitte Bosch Rexroth Corporation Electric Drives & Controls Central Region Technical Center 1701 Harmon Road Auburn Hills, MI 48326 Tel.: +1 248 3 93 33 30 Fax: +1 248 3 93 29 06	USA Southeast Region - Südwest Bosch Rexroth Corporation Electric Drives & Controls Southeastern Technical Center 3625 Swiftwater Park Drive Suwanee, Georgia 30124 Tel.: +1 770 9 32 32 00 Fax: +1 770 9 32 19 03	USA SERVICE-HOTLINE - 7 days x 24hrs - **+1-800-REX-ROTH** **+1-800-739-7684**
USA East Region – Ost Bosch Rexroth Corporation Electric Drives & Controls Charlotte Regional Sales Office 14001 South Lakes Drive Charlotte, North Carolina 28273 Tel.: +1 704 5 83 97 62 +1 704 5 83 14 86	USA Northeast Region – Nordost Bosch Rexroth Corporation Electric Drives & Controls Northeastern Technical Center 99 Rainbow Road East Granby, Connecticut 06026 Tel.: +1 860 8 44 83 77 Fax: +1 860 8 44 85 95	USA West Region – West Bosch Rexroth Corporation 7901 Stoneridge Drive, Suite 220 Pleasant Hill, California 94588 Tel.: +1 925 227 10 84 Fax: +1 925 227 10 81	
Canada East - Kanada Ost Bosch Rexroth Canada Corporation Burlington Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8 Tel.: +1 905 335 55 11 Fax: +1 905 335-41 84 michael.moro@boschrexroth.ca	Canada West - Kanada West Bosch Rexroth Canada Corporation 5345 Goring St. Burnaby, British Columbia Canada V7J 1R1 Tel. +1 604 205-5777 Fax +1 604 205-6944 david.gunby@boschrexroth.ca	Mexico Bosch Rexroth Mexico S.A. de C.V. Calle Neptuno 72 Unidad Ind. Vallejo 07700 Mexico, D.F. Tel.: +52 55 57 54 17 11 Fax: +52 55 57 54 50 73 mariofelipe.hernandez@boschrexroth.com.mx	Mexico Bosch Rexroth S.A. de C.V. Calle Argentina No 3913 Fracc. las Torres 64930 Monterrey, N.L. Tel.: +52 81 83 65 22 53 +52 81 83 65 89 11 +52 81 83 49 80 91 Fax: +52 81 83 65 52 80 mario.quiroga@boschrexroth.com.mx

Südamerika – South America

Argentina - Argentinien Bosch Rexroth S.A.I.C. "The Drive & Control Company" Acassusso 48 41/47 1605 Munro Provincia de Buenos Aires Tel.: +54 11 4756 01 40 Fax: +54 11 4756 01 36 victor.jabif@boschrexroth.com.ar	Argentina - Argentinien NAKASE Servicio Tecnico CNC Calle 49, No. 5764/66 B1653AOX Villa Balester Provincia de Buenos Aires Tel.: +54 11 4768 36 43 Fax: +54 11 4768 24 13 nakase@usa.net nakase@nakase.com gerencia@nakase.com (Service)	Brazil - Brasilien Bosch Rexroth Ltda. Av. Tégula, 888 Ponte Alta, Atibaia SP CEP 12942-440 Tel.: +55 11 4414 56 92 +55 11 4414 56 84 Fax sales: +55 11 4414 57 07 Fax serv.: +55 11 4414 56 86 alexandre.wittwer@rexroth.com.br	Brazil - Brasilien Bosch Rexroth Ltda. R. Dr.Humberto Pinheiro Vieira, 100 Distrito Industrial [Caixa Postal 1273] 89220-390 Joinville - SC Tel./Fax: +55 47 473 58 33 Mobil: +55 47 9974 6645 prochnow@zaz.com.br
Columbia - Kolumbien Reflutec de Colombia Ltda. Calle 37 No. 22-31 Santafé de Bogotá, D.C. Colombia Tel.: +57 1 368 82 67 +57 1 368 02 59 Fax: +57 1 268 97 37 reflutec@neutel.com.co reflutec@007mundo.com			

Notes

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Phone +49 93 52-40-0
Fax +49 93 52-40-48 85
www.boschrexroth.de

R911296974

Printed in Germany
DOK-MTC200-SPS*GWY*V23-AW01-EN-P