

Manual
PLCopen Motion Library
for S7-1200/1500

Lexium 32M – PROFIBUS DP
Lexium 32M – PROFINET IO
Lexium32intergrated – PROFINET IO

Edition: V2.1, 02.2017



Schneider Electric
Automation GmbH
Breslauer Str. 7
D-77933 Lahr

Contents

1	<i>Disclaimer</i>	5
2	<i>Safety information</i>	6
2.1	Hazard categories.....	6
2.2	Qualification of personnel	7
2.3	Intended use	7
2.4	Basic Information	8
2.5	Standards and terminology.....	9
3	<i>Changes in Version 2.1</i>	10
3.1	PROFIBUS/PROFINET IO Data	10
3.1.1	ReadInputData_1200_1500	10
3.1.2	WriteOutputData_1200_1500	10
3.2	Support of Lexium32 integrated.....	10
4	<i>Extracting the library</i>	11
4.1	Group “00 Mandatory Basic Functions”	13
4.2	Group “01 Administrative”	13
4.3	Group “02 Single Axis”	14
4.4	Group “03 Multi Axis”	14
4.5	Group “04 Extended IO Module”	15
4.6	Group “05 MotionSequenceMode”	15
5	<i>Installing the GSD</i>	16
5.1	Profibus (GSD).....	16
5.2	Profinet (GSD)	18
6	<i>Starting a new project</i>	20
6.1	Create a newproject	20
6.2	Add PLC	20
6.3	Add LXM32 to the network.....	22
6.3.1	Profibus	22
6.3.2	Profinet.....	22
6.4	HW Identifier of DriveProfileLexium 1.....	24
6.4.1	Profibus:	24
6.4.2	Profinet:	25
6.5	Copy library elements into the project	26
6.6	Create Axis Reference Structure	27
6.7	Assign HW-ID to the AxisRef (Init_LXM32_1200_1500)	27
6.8	Call MC_ function blocks.....	28
7	<i>Library blocks</i>	32
7.1	Explanation of common parameters	32

7.2	Behavior of function blocks with the input Enable.....	33
7.3	Behavior of function blocks with the input Execute	35
7.4	PLCopen state diagram.....	37
7.5	Initialisation.....	38
7.5.1	Init_LXM32_1200_1500.....	38
7.5.2	ReadInputData_1200_1500 [FC1].....	39
7.5.3	WriteOutputData_1200_1500.....	39
7.5.4	MC_Power_LXM32_1200_1500.....	40
7.6	Jog.....	41
7.6.1	MC_Jog_LXM32_1200_1500.....	41
7.7	Homing.....	43
7.7.1	MC_SetPosition_LXM32_1200_1500.....	43
7.7.2	MC_Home_LXM32_1200_1500.....	44
7.8	Torque control	46
7.8.1	MC_TorqueControl_LXM32_1200_1500	46
7.9	Profile Velocity mode	48
7.9.1	MC_Move_Velocity_LXM32_1200_1500.....	48
7.10	Profile position mode	50
7.10.1	MC_MoveAbsolute_LXM32_1200_1500.....	50
7.10.2	MC_MoveAdditive_LXM32_1200_1500.....	52
7.10.3	MC_MoveRelative_LXM32_1200_1500.....	53
7.11	Electronic gear	54
7.11.1	MC_GearIn_LXM32_1200_1500	54
7.11.2	GearInSync_LXM32_1200_1500	56
7.11.3	MC_GearOut_LXM32_1200_1500	58
7.12	Stopping	59
7.12.1	MC_Stop_LXM32_1200_1500.....	59
7.12.2	MC_Halt_LXM32_1200_1500.....	60
7.13	Fast position capture.....	61
7.13.1	MC_TouchProbe_LXM32_1200_1500.....	61
7.13.2	MC_AbortTrigger_LXM32_1200_1500.....	63
7.14	Read parameter	64
7.14.1	MC_ReadPar_LXM32_1200_1500.....	64
7.14.2	MC_ReadStatus_LXM32_1200_1500	66
7.14.3	MC_ReadActualPos_LXM32_1200_1500	68
7.14.4	MC_ReadActualVel_LXM32_1200_1500.....	69
7.14.5	MC_RdActualTorque_LXM32_1200_1500	70
7.14.6	UploadParameter_LXM32_1200/1500	71
7.15	Write parameter	73
7.15.1	MC_WritePar_LXM32_1200_1500.....	73
7.15.2	SetLimitSwitch_LXM32_1200_1500.....	75
7.15.3	ResetParameters_LXM32_1200_1500	76
7.15.4	StorePar_LXM32_1200_1500.....	77
7.15.5	DownloadParameter_LXM32_1200_1500	78
7.16	Inputs/outputs.....	80
7.16.1	MC_ReadDigInput_LXM32_1200_1500	80
7.16.2	MC_ReadDigOutput_LXM32_1200_1500	81
7.16.3	MC_WriteDigitalOutput_LXM32_1200_1500	82

7.17	Error handling	83
7.17.1	MC_ReadAxisError_LXM32_1200_1500	83
7.17.2	ReadAxisWarning_LXM32_1200_1500	86
7.17.3	MC_Reset_LXM32_1200_1500	87
7.18	Extended IO module	88
7.18.1	ConfAnalogOutputs_LXM32_1200_1500	88
7.18.2	ReadAnalogInput_LXM32_1200_1500	89
7.18.3	ReadAnalogOutput_LXM32_1200_1500	90
7.18.4	SetAnalogOutputMod_LXM32_1200_1500	91
7.18.5	WriteAnalogOutput_LXM32_1200_1500	92
7.19	Motion Sequence Mode	93
7.19.1	AbortMotionSeq_LXM32_1200_1500	93
7.19.2	DownlDataSets_LXM32_1200_1500	94
7.19.3	RdMotionSeqStatus_LXM32_1200_1500	95
7.19.4	ReadDataSet_LXM32_1200_1500	96
7.19.5	StartMotionSeq_LXM32_1200_1500	97
7.19.6	UpIDataSets_LXM32_1200/1500	98
7.19.7	WriteDataSet_LXM32_1200_1500	99
7.19.8	WriteTransCond_LXM32_1200_1500	100
8	Glossary	101
9	List of error numbers	103

1 Disclaimer

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2016 Schneider Electric. All rights reserved.

2 Safety information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a Danger safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

2.1 Hazard categories

Safety instructions to the user are highlighted by safety alert symbols in the manual. In addition, labels with symbols and/or instructions are attached to the product that alert you to potential hazards.

Depending on the seriousness of the hazard, the safety instructions are divided into 4 hazard categories.

DANGER

DANGER indicates an imminently hazardous situation, which, if not avoided, **will result** in death or serious injury.

WARNING

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

NOTICE

NOTICE indicates a potentially hazardous situation, which, if not avoided, **can result** in equipment damage.

2.2 Qualification of personnel

Only appropriately trained persons who are familiar with and understand the contents of this manual and all other pertinent product documentation are authorized to work on and with this product. In addition, these persons must have received safety training to recognize and avoid hazards involved. These persons must have sufficient technical training, knowledge and experience and be able to foresee and detect potential hazards that may be caused by using the product, by changing the settings and by the mechanical, electrical and electronic equipment of the entire system in which the product is used.

All persons working on and with the product must be fully familiar with all applicable standards, directives, and accident prevention regulations when performing such work.

2.3 Intended use

This product is a library for industrial use with the appropriate controllers, drives and motors.

The product may only be used in compliance with all applicable safety regulations and directives, the specified requirements and the technical data.


Prior to using the product, you must perform a risk assessment in view of the planned application. Based on the results, the appropriate safety measures must be implemented.

Since the product is used as a component in an entire system, you must ensure the safety of persons by means of the design of this entire system (for example, machine design).

Any use other than the use explicitly permitted is prohibited and can result in hazards.


Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel.

2.4 Basic Information

 WARNING
<p>LOSS OF CONTROL</p> <ul style="list-style-type: none">• The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop, overtravel stop, power outage and restart.• Separate or redundant control paths must be provided for critical functions.• System control paths may include communication links. Consideration must be given to the implication of unanticipated transmission delays or failures of the link.• Observe all accident prevention regulations and local safety guidelines.¹⁾• Each implementation of the product must be individually and thoroughly tested for proper operation before being placed into service. <p>Failure to follow these instructions can result in death or serious injury.</p>

1) For USA: Additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems".

Improper error handling can provoke unintended movements or signals or deactivate monitoring functions. You must create effective error handling routines within your application and provide for safe control algorithms to respond appropriately to any detected error.

 WARNING
<p>UNINTENDED OPERATION DUE TO IMPROPER ERROR HANDLING</p> <ul style="list-style-type: none">• Carefully program the error handling routines.• Verify the effectiveness of all error handling routines. <p>Failure to follow these instructions can result in death, serious injury or equipment damage.</p>

WARNING

UNINTENDED OPERATION DUE TO MODIFICATIONS TO THE LIBRARY

Do not modify the library in any way whatsoever.

Failure to follow these instructions can result in death, serious injury or equipment damage.

2.5 Standards and terminology

Technical terms, terminology and the corresponding descriptions in this manual are intended to use the terms or definitions of the pertinent standards.

In the area of drive systems, this includes, but is not limited to, terms such as "safety function", "safe state", "fault", "fault reset", "failure", "error", "error message", "warning", "warning message", etc.

Among others, these standards include:

- IEC 61800 series: "Adjustable speed electrical power drive systems"
- IEC 61158 series: "Digital data communications for measurement and control – Fieldbus for use in industrial control systems"
- IEC 61784 series: "Industrial communication networks – Profiles"
- IEC 61508 series: "Functional safety of electrical/electronic/ programmable electronic safety-related systems"

Also see the glossary at the end of this manual.

3 Changes in Version 2.1

3.1 PROFIBUS/PROFINET IO Data

Reading and writing the fieldbus IO data was done in earlier versions of the library inside each function block automatically.

To improve the performance and reduce the PLC cycle time this has changed.

There are two new functions which must be called once each PLC cycle by the user.

3.1.1 ReadInputData_1200_1500

Reading the input data of the “DriveProfileLexium1”.

This function must be called at the beginning of each cycle before the library function blocks are called.

3.1.2 WriteOutputData_1200_1500

Writing the output data of the “DriveProfileLexium1”.

This function must be called at the end of each cycle after the library function blocks are called.

3.2 Support of Lexium32 integrated

The LXM32i (PROFINET) is supported since this library version.

The following Operating modes are not supported by the LXM32i.

- Electronic Gear
- MotionSequenceMode

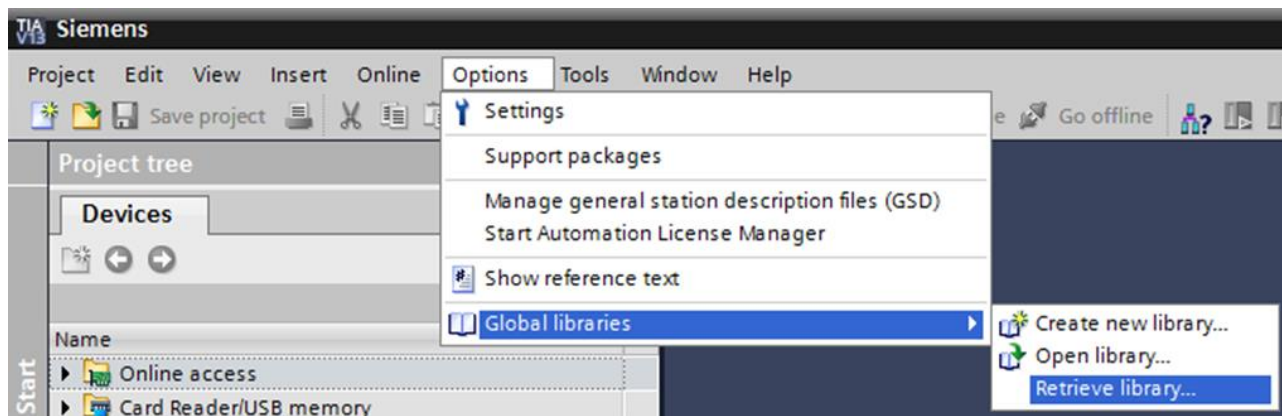
The following function are not supported by the LXM32i

- Functions with the extended IO Module

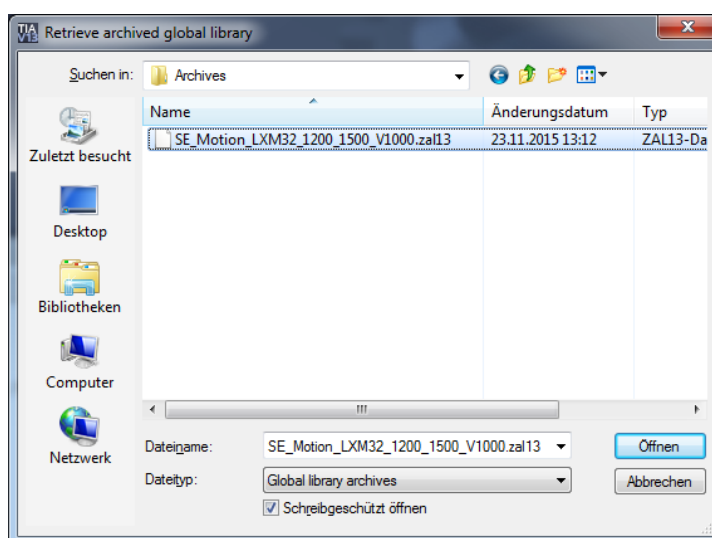
4 Extracting the library

In order to use the library blocks, you must first unpack the archive “SE_Motion_LXM32_1200_1500_Vxxxx.zal13” with the TIA Portal software. This is done with the menu item “**Retrieve Library...**” in the menu

“Options->Global Libraries”.

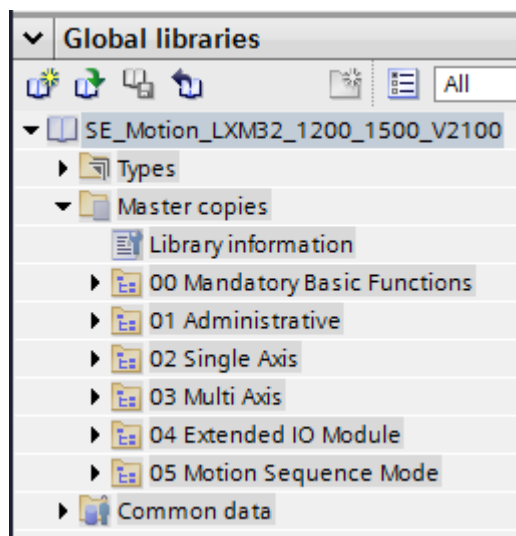


Browse to the directory of the library archive, and mark the library. Confirm your selection with “Open”.



In the window shown below, you select the target directory into which the library is to be unpacked.





You have now successfully unpacked the library. The library items are available in the window Libraries -> GlobalLibraries -> SE_Motion_LXM32_1200_1500.





















The Library is structured in 6 groups.

4.1 Group “00 Mandatory Basic Functions”














The data structure “Axis_Ref_LXM32_1200_1500” and the functions “Init_LXM32_1200_1500”, “ReadInputData_1200_1500” and “WriteOutputData_1200_1500” are mandatory when using any function block of this library.

	Name	Modified
	WriteOutputData_1200_1500	1/26/2017 10:31 AM
	Axis_Ref_LXM32_1200_1500	1/26/2017 10:31 AM
	ReadInputData_1200_1500	1/26/2017 10:31 AM
	Init_LXM32_1200_1500	1/26/2017 10:31 AM




4.2 Group “01 Administrative”

	Name ▲	Modified
	DownlDrivePar_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_RdActualTorque_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_ReadActualPos_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_ReadActualVel_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_ReadAxisError_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_ReadDigInput_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_ReadDigOutput_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_ReadPar_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_ReadStatus_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_Reset_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_WriteDigOutput_LXM32_1200_1500	1/26/2017 10:31 AM
	MC_WritePar_LXM32_1200_1500	1/26/2017 10:31 AM
	ReadAxisWarning_LXM32_1200_1500	1/26/2017 10:31 AM
	ResetPar_LXM32_1200_1500	1/26/2017 10:31 AM
	SetLimitSwitch_LXM32_1200_1500	1/26/2017 10:31 AM
	StorePar_LXM32_1200_1500	1/26/2017 10:31 AM
	UplDrivePar_LXM32_1200	1/26/2017 10:31 AM
	UplDrivePar_LXM32_1500	1/26/2017 10:31 AM






4.3 Group “02 Single Axis”

Name ▲	Modified
 MC_AbortTrigger_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_Halt_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_Home_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_Jog_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_MoveAbsolute_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_MoveAdditive_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_MoveRelative_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_MoveVelocity_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_Power_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_SetPosition_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_Stop_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_TorqueControl_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_TouchProbe_LXM32_1200_1500	1/26/2017 10:31 AM











4.4 Group “03 Multi Axis”

Name ▲	Modified
 GearInSync_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_GearIn_LXM32_1200_1500	1/26/2017 10:31 AM
 MC_GearOut_LXM32_1200_1500	1/26/2017 10:31 AM

4.5 Group “04 Extended IO Module”

Name ▲	Modified
 ConfAnalogOutputs_LXM32_1200_1500	1/26/2017 10:31 AM
 ReadAnalogInput_LXM32_1200_1500	1/26/2017 10:31 AM
 ReadAnalogOutput_LXM32_1200_1500	1/26/2017 10:31 AM
 SetAnalogOutputMod_LXM32_1200_1500	1/26/2017 10:31 AM
 WriteAnalogOutput_LXM32_1200_1500	1/26/2017 10:31 AM

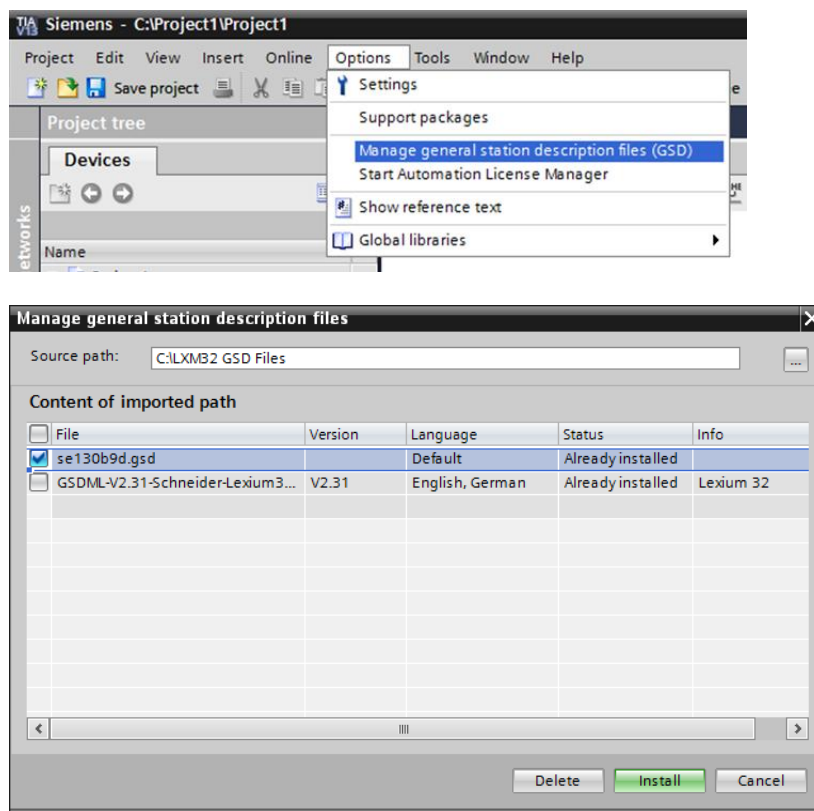
4.6 Group “05 MotionSequenceMode”

Name ▲	Modified
 AbortMotionSeq_LXM32_1200_1500	1/26/2017 10:31 AM
 DataSet_LXM32_1200_1500	1/26/2017 10:31 AM
 DownlDataSets_LXM32_1200_1500	1/26/2017 10:31 AM
 RdMotionSeqStatus_LXM32_1200_1500	1/26/2017 10:31 AM
 ReadDataSet_LXM32_1200_1500	1/26/2017 10:31 AM
 StartMotionSeq_LXM32_1200_1500	1/26/2017 10:31 AM
 UpIDataSets_LXM32_1200	1/26/2017 10:31 AM
 UpIDataSets_LXM32_1500	1/26/2017 10:31 AM
 WriteDataSet_LXM32_1200_1500	1/26/2017 10:31 AM
 WriteTransCond_LXM32_1200_1500	1/26/2017 10:31 AM

5 Installing the GSD

5.1 Profibus (GSD)

The library uses the GSD (Generic Station Description) “**SE100B9D.GSD**” for LXM32.



1. Select the directory with the original GSD file
2. Select the GSD file which should be installed
3. Select “Install” to execute the operation



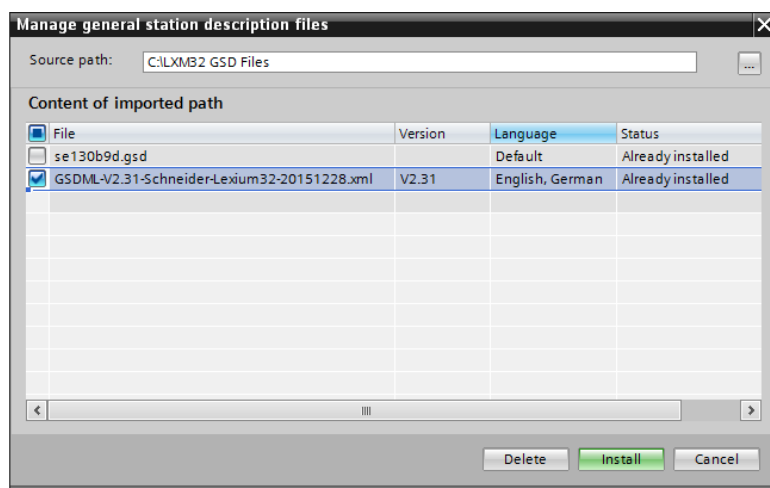
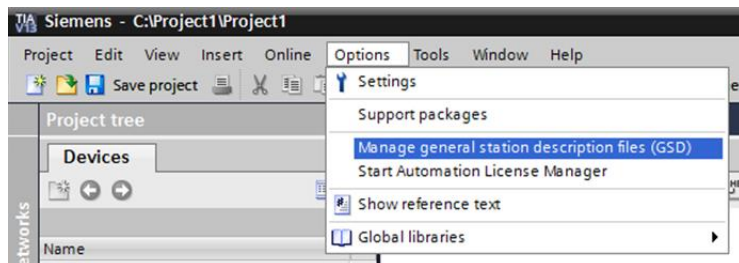
5.2 Profinet (GSD)

The library uses the GSD (Generic Station Description)

“**GSDML-V2.31-Schneider-Lexium32-xxxxxxx.xml**” für LXM32M (PROFINET)

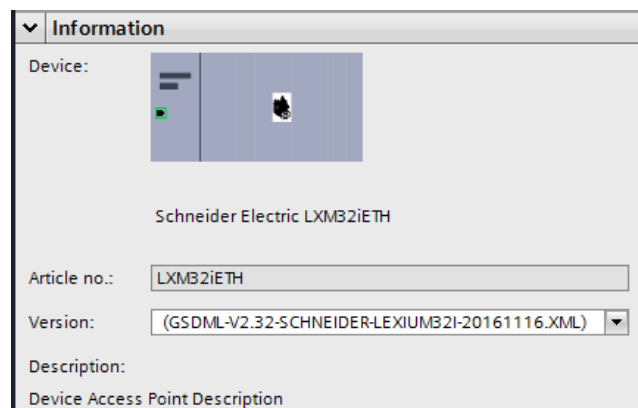
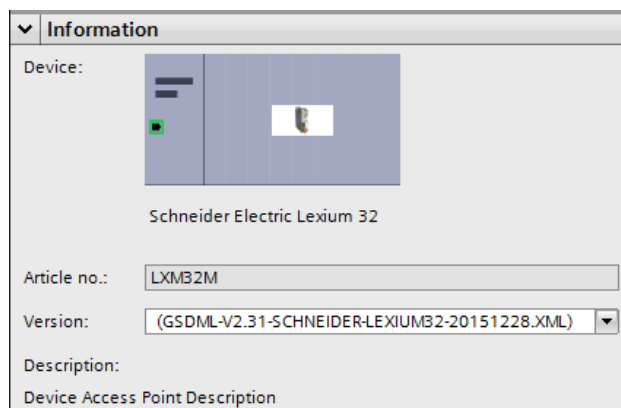
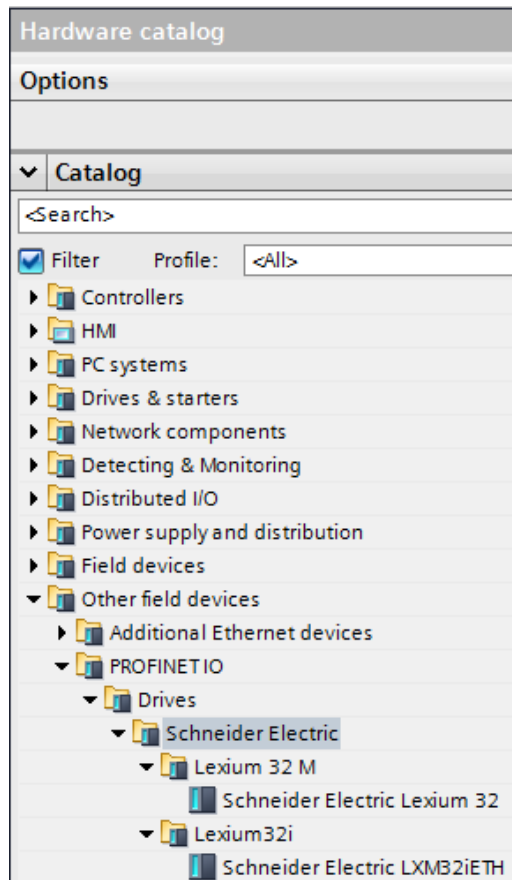
or

„**GSDML-V2.32-Schneider-Lexium32i-xxxxxxx.xml**“ für LXM32i (PROFINET)



1. Select the directory with the original GSD file
2. Select the GSD file which should be installed
3. Select “Install” to execute the operation

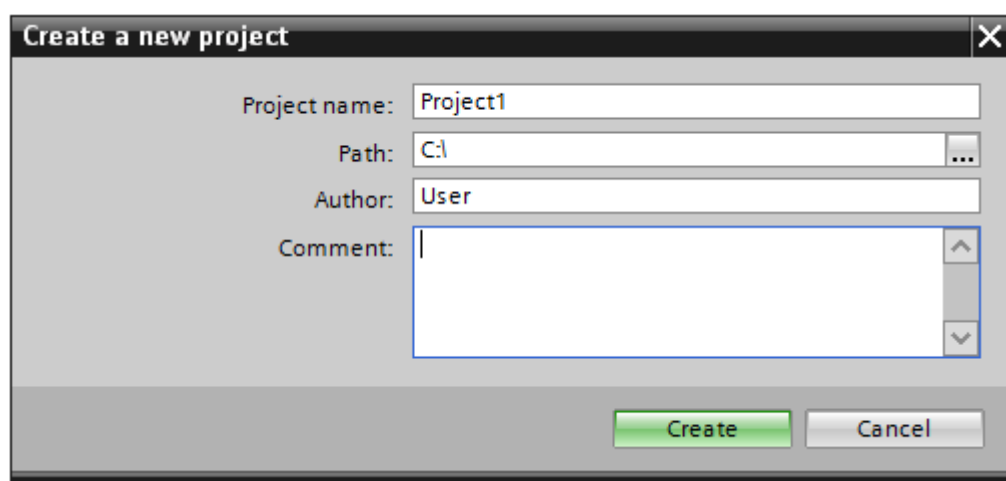
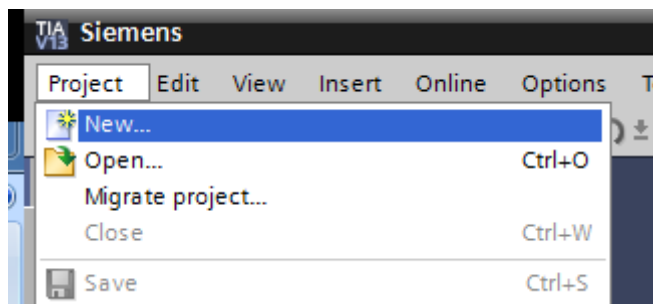
After a successful installation the LXM32 Profinet device is available in the TIA Portal Hardware Catalog.



6 Starting a new project

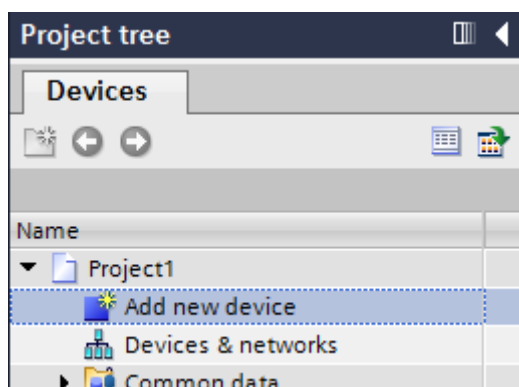
6.1 Create a newproject

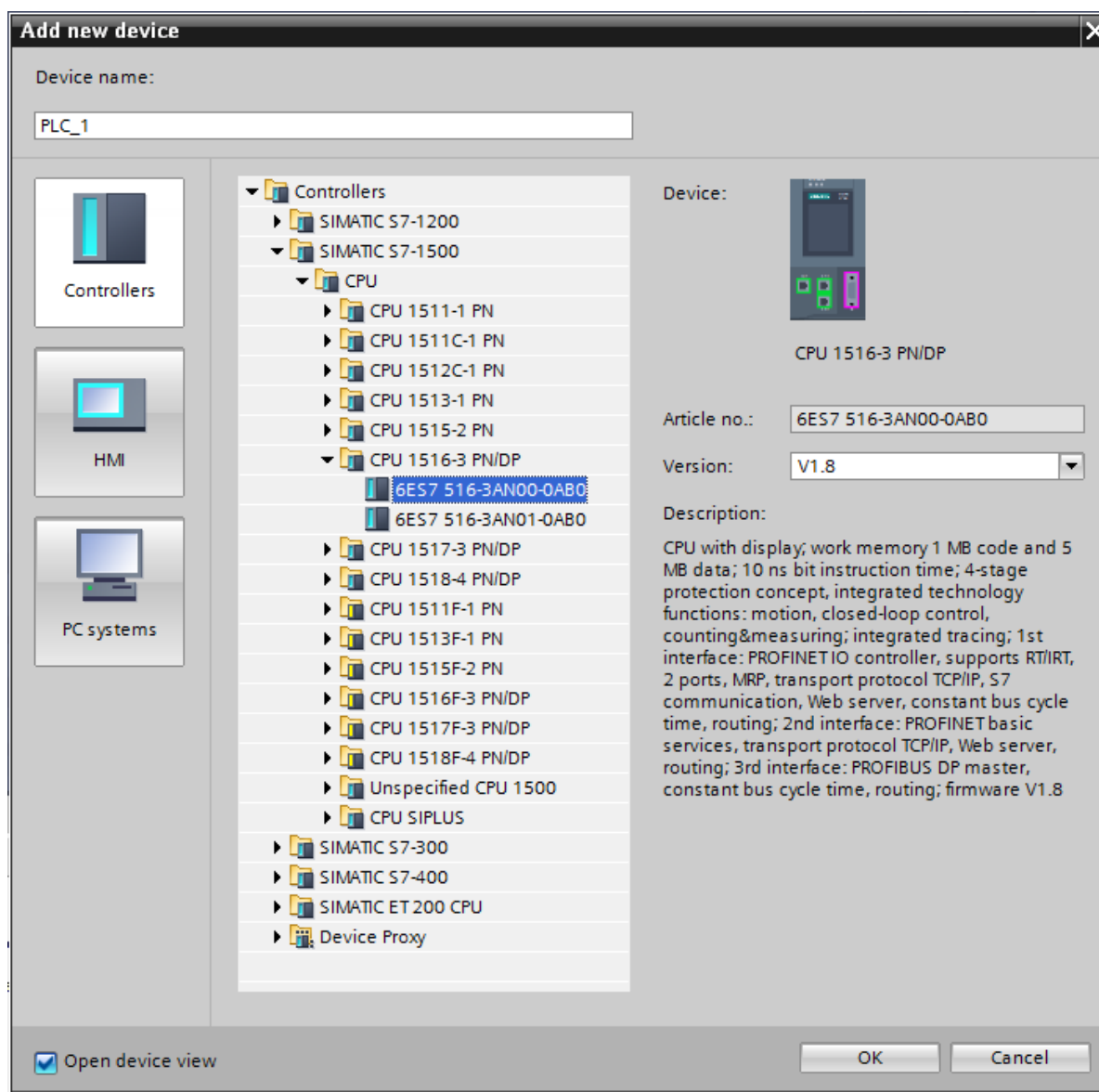
Create a new project. For this, you open the menu **Project**, and select the menu item **New**.



6.2 Add PLC

When you have created a new project select “Add new device”

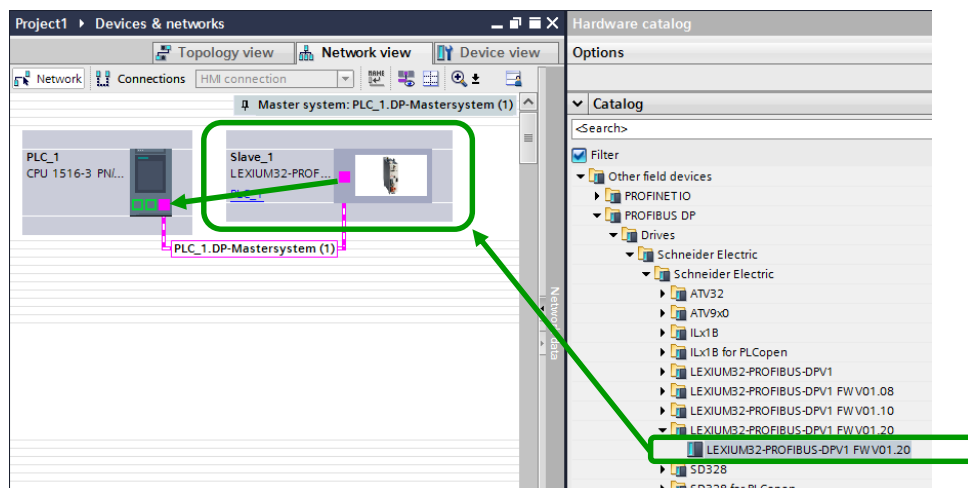




6.3 Add LXM32 to the network

6.3.1 Profibus

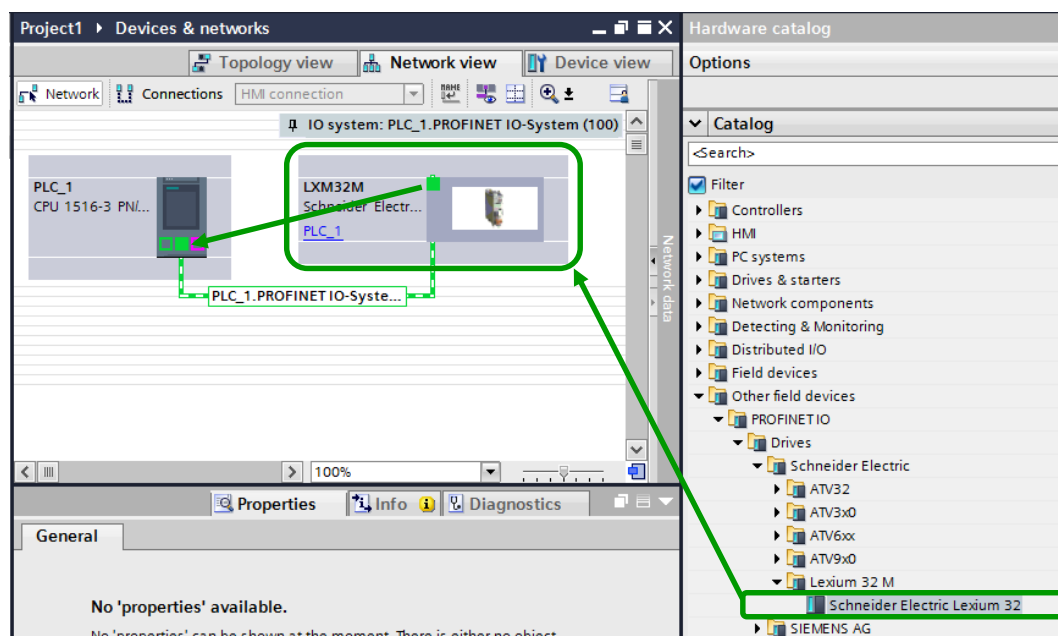
1. Insert the LXM32 Profibus Device to the Network view
2. Link the LXM32 PB interface to the PLC Master Interface



You have now linked the drive into the network as a Profibus Slave.

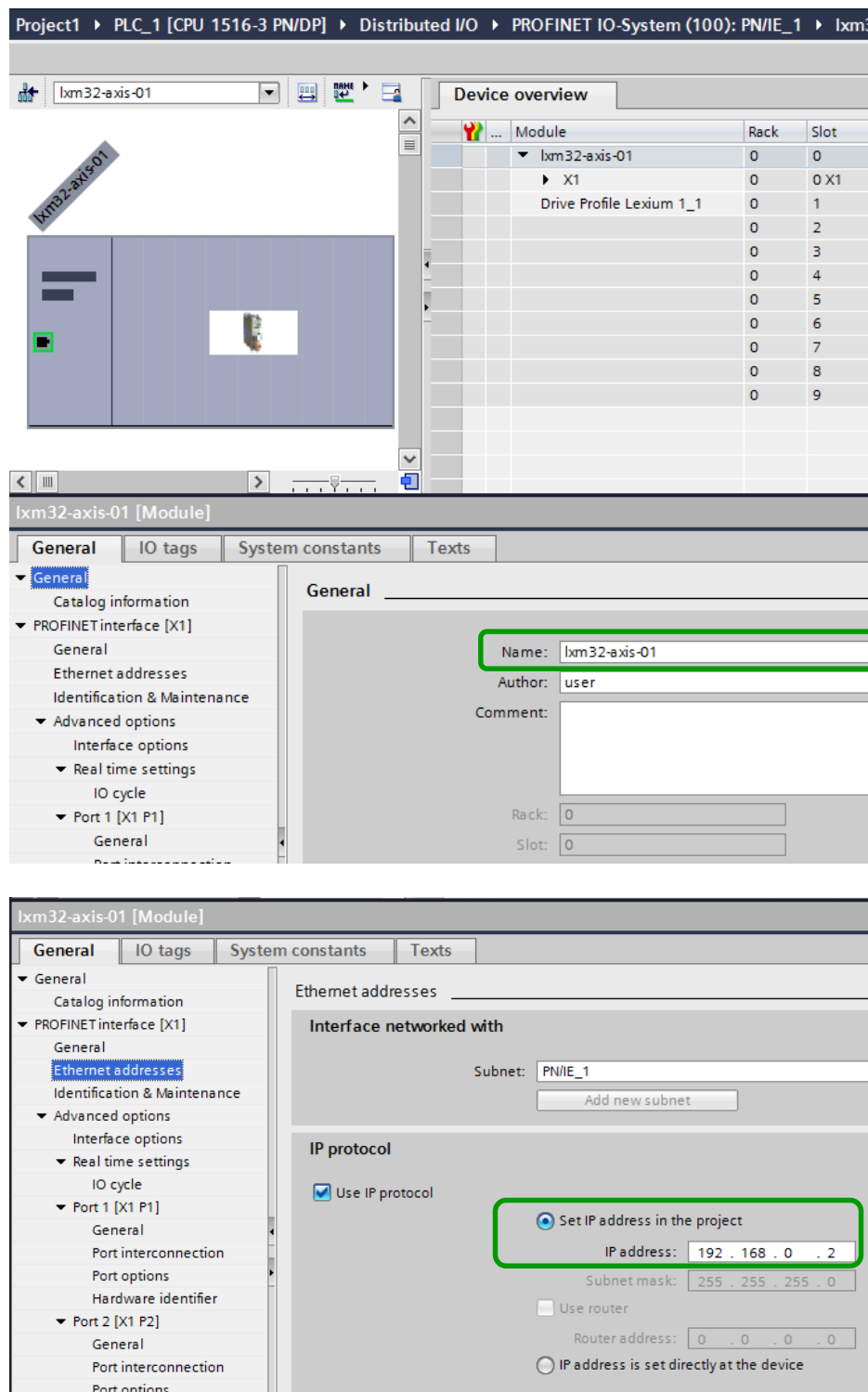
6.3.2 Profinet

1. Insert the LXM32 Profinet Device to the Network view
2. Link the LXM32 Profinet interface to the PLC Master Interface



You have now linked the drive into the network as a Profinet IO-Device.

Setup PROFINET station name and IP address (Properties of LXM32 Device)



The screenshot shows the SIMATIC Manager interface for configuring a PROFINET station. The top bar indicates the project path: Project1 > PLC_1 [CPU 1516-3 PN/DP] > Distributed I/O > PROFINET IO-System (100): PN/IE_1 > lxm32-axis-01.

The left sidebar shows the project tree with the following structure:

- lxm32-axis-01
 - General
 - PROFINET interface [X1]
 - General
 - Ethernet addresses
 - Identification & Maintenance
 - Advanced options
 - Interface options
 - Real time settings
 - IO cycle
 - Port 1 [X1 P1]
 - General

The main window displays the 'Device overview' table:

Module	Rack	Slot
lxm32-axis-01	0	0
X1	0	0 X1
Drive Profile Lexium 1_1	0	1
	0	2
	0	3
	0	4
	0	5
	0	6
	0	7
	0	8
	0	9

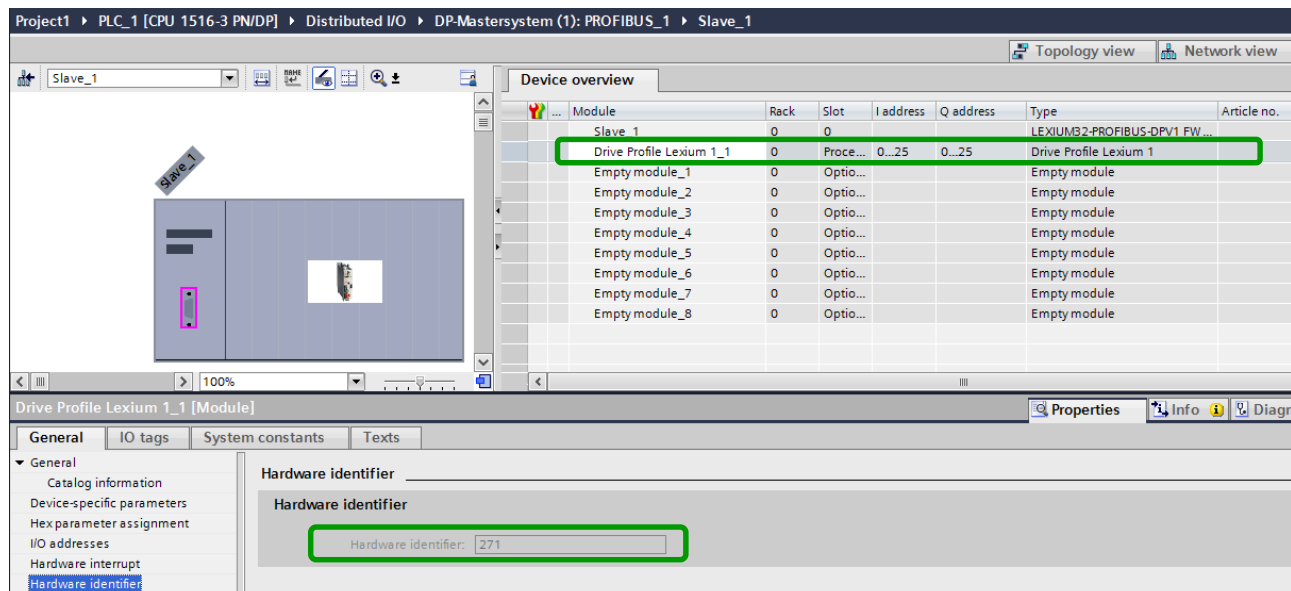
The 'lxm32-axis-01 [Module]' properties window is open, showing the 'General' tab. The 'Name' field is highlighted with a green box and contains the text 'lxm32-axis-01'. The 'Author' field contains 'user'.

The 'Ethernet addresses' tab is also shown, with the 'Interface networked with' section displaying 'Subnet: PN/IE_1'. The 'IP protocol' section is checked, and the 'Set IP address in the project' radio button is selected, with the IP address field highlighted by a green box and containing '192.168.0.2'.

6.4 HW Identifier of DriveProfileLexium 1

The PLCopen open library does only use the IO data structure of DriveProfileLexium1. DriveProfileLexium1 is implemented in the default configuration in Profibus and Profinet.

6.4.1 Profibus:



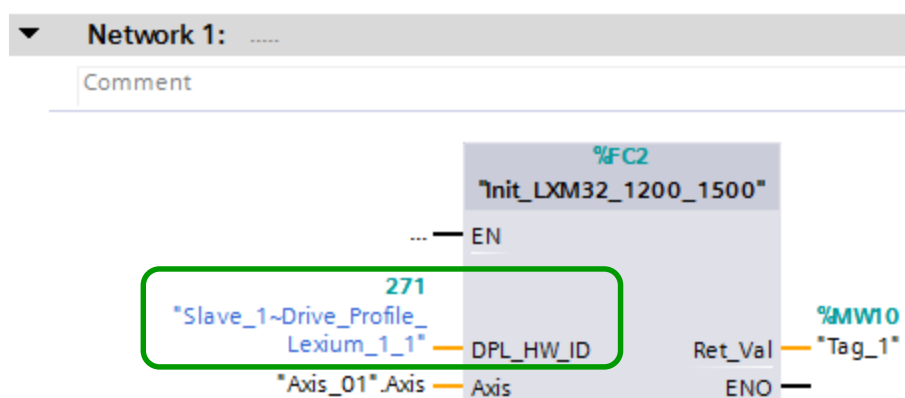
The screenshot shows the HW Config interface for a Profibus network. The 'Device overview' table lists the modules in the rack:

Module	Rack	Slot	I address	Q address	Type	Article no.
Slave 1	0	0			LEXIUM32-PROFIBUS-DPV1 FW...	
Drive Profile Lexium 1_1	0	Proce...	0...25	0...25	Drive Profile Lexium 1	
Empty module_1	0	Optio...			Empty module	
Empty module_2	0	Optio...			Empty module	
Empty module_3	0	Optio...			Empty module	
Empty module_4	0	Optio...			Empty module	
Empty module_5	0	Optio...			Empty module	
Empty module_6	0	Optio...			Empty module	
Empty module_7	0	Optio...			Empty module	
Empty module_8	0	Optio...			Empty module	

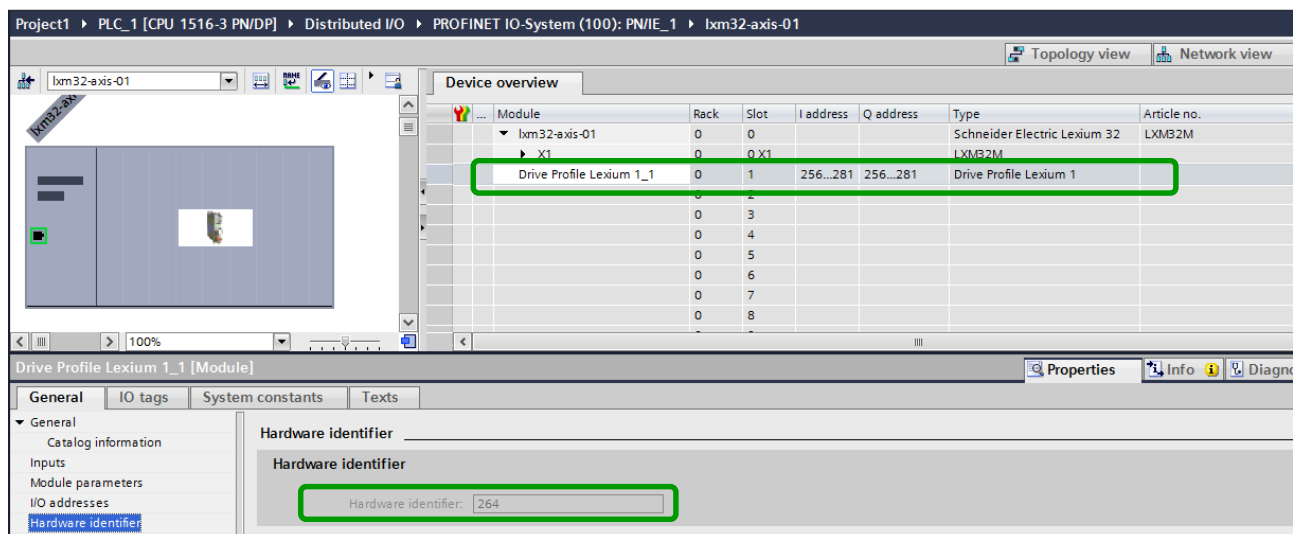
The 'Properties' window for 'Drive Profile Lexium 1_1 [Module]' is shown, with the 'Hardware identifier' set to 271.

Note: The Hardware identifier is used in the application to address the IO data of the DriveProfileLexium1. The Hardware identifier must be declared at the input "DPL_HW_ID" of the function [Init LXM32 1200 1500](#).

Example:



6.4.2 Profinet:



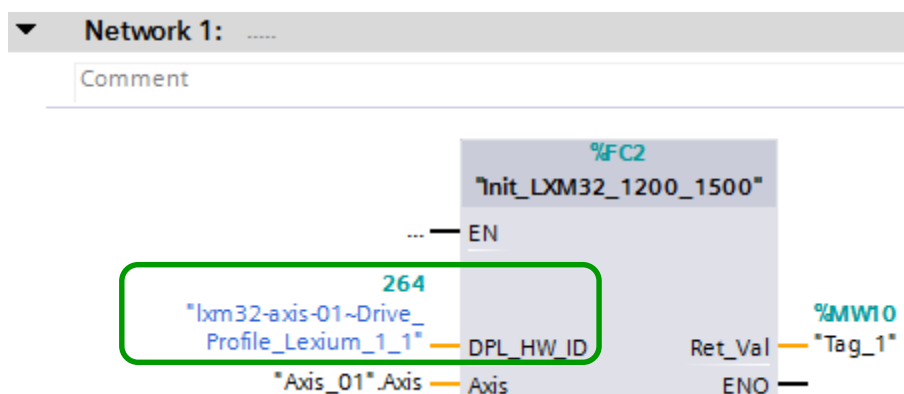
The screenshot shows the HW Config interface for a project named 'Project1'. The breadcrumb path is: PLC_1 [CPU 1516-3 PN/DP] > Distributed I/O > PROFINET IO-System (100): PN/IE_1 > lxm32-axis-01. The 'Device overview' table lists the modules:

Module	Rack	Slot	I address	Q address	Type	Article no.
lxm32-axis-01	0	0			Schneider Electric Lexium 32	LXMB2M
X1	0	0	X1		LXM32M	
Drive Profile Lexium 1_1	0	1	256...281	256...281	Drive Profile Lexium 1	

The 'Drive Profile Lexium 1_1' module is highlighted with a green box. Below the table, the 'Hardware identifier' field is set to '264', also highlighted with a green box.

Note: The Hardware identifier is used in the application to address the IO data of the DriveProfileLexium1.
The Hardware identifier must be declared at the input "DPL_HW_ID" of the function [Init_LXM32_1200_1500](#).

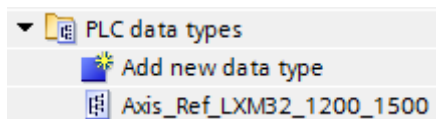
Example:



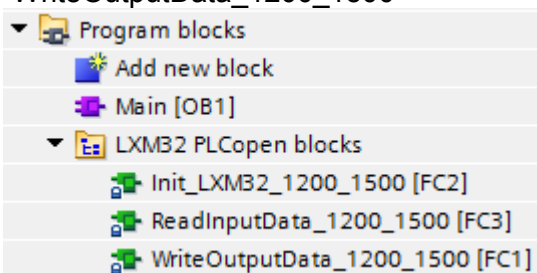
6.5 Copy library elements into the project

At first you have to copy the mandatory items into the new project.

1. PLC data type “Axis_Ref_LXM32_1200_1500”

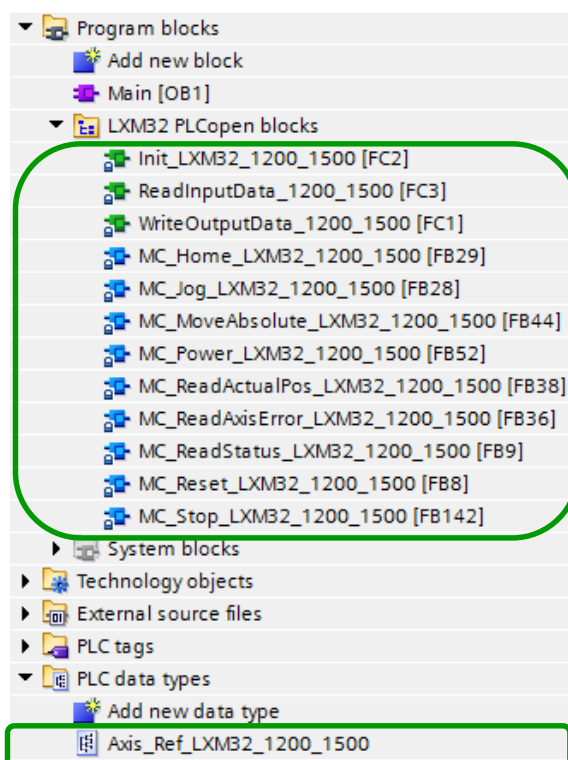


2. Functions: “Init_LXM32_1200_1500”, “ReadInputData_1200_1500” and “WriteOutputData_1200_1500”



3. At last you have to copy all function blocks which are needed in the application.
Example setup for a pic and place application

MC_Power:	Enable/Disable the drive power stage
MC_ReadStatus	Read the drive status
MC_ReadAxisError	Read drive errors and application errors
MC_Reset	Error reset
MC_Home	Reference movement
MC_Jog	Manual movement
MC_MoveAbsolute	Point to point movement
MC_Stop	Stop an active movement
MC_ReadActualPosition	Read the actual drive position

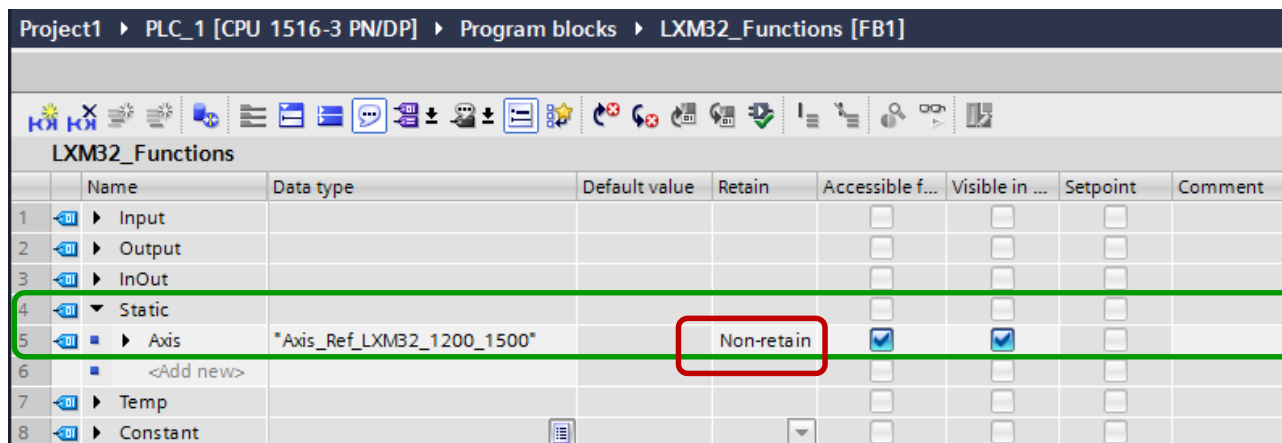


6.6 Create Axis Reference Structure

Each LXM32 needs an own axis reference structure of the PLC data type “Axis_Ref_LXM32_1200_1500”.

This structure can be defined in a global data block or in the static variables of an function block.

In this example you can see how to declare the axis reference structure in the stat variables of a function block.



	Name	Data type	Default value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
3	InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
4	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
5	Axis	*Axis_Ref_LXM32_1200_1500		Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
7	Temp				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
8	Constant				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

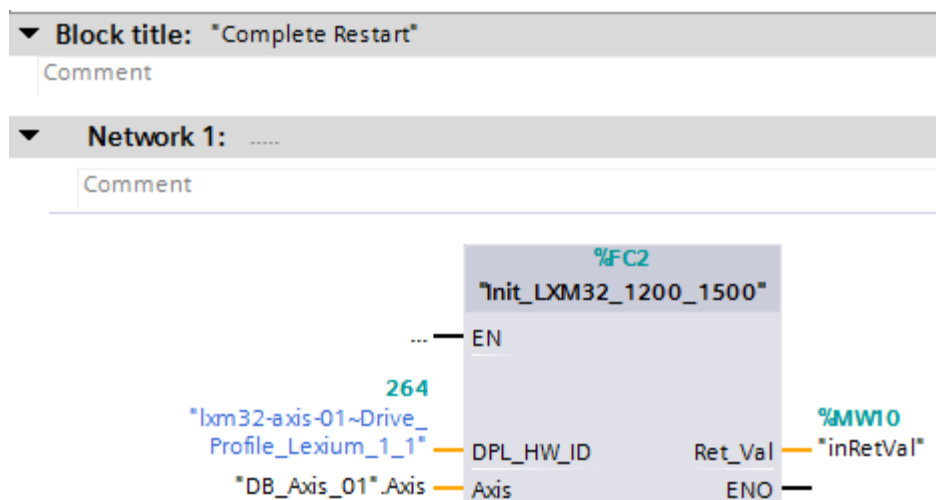
Note: The data structure “Axis_Ref_1200_1500” should be set to Non-retain because the data has to be initialized at PLC restart.

6.7 Assign HW-ID to the AxisRef (Init_LXM32_1200_1500)

At PLC restart the HW-ID of LXM32 (DriveProfileLexium1) has to be assigned to the AxisReference Structure.

For this the function Init_LXM32_1200_1500 must be called for one cycle at PLC startup. Usually this is done in the OB 100 (Startup OB).

In this Example the Axis reference structure is defined in the instance data block “DB_Axis_01” with the symbolic name “Axis” and the Init_LXM32_1200_1500 is called in OB100. The LXM32 Profinet station name is defined as “lxm32-axis-01”. The hardware ID of the “DriveProfileLexium1” is 264. (See HW configuration)

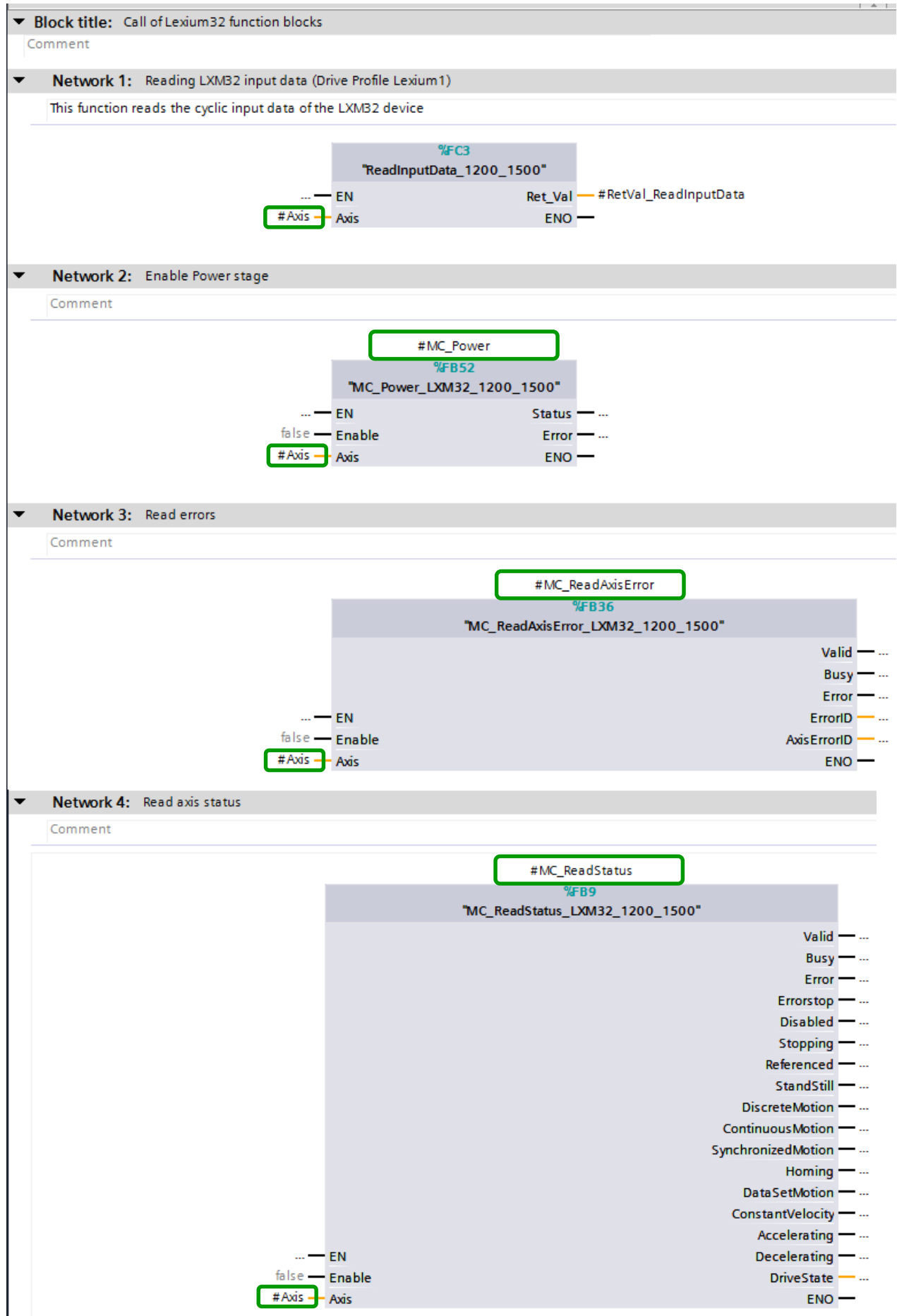


6.8 Call MC_ function blocks

The library function blocks can be called separately with an own instance DB.
It is also possible to define multi instances in the static variables of a function block.

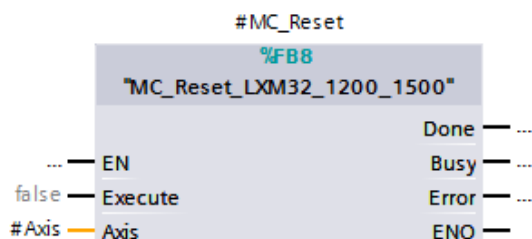
In this example the motion function blocks are called in one multi instance function block.

LXM32_Blocks				
	Name	Data type	...	Retain
1	Input			
2	Output			
3	InOut			
4	Static			
5	Axis	"Axis_Ref_LXM32_1200_1500"		Non-retain
6	MC_Power	"MC_Power_LXM32_1200_1500"		
7	MC_ReadAxisError	"MC_ReadAxisError_LXM32_1200_1500"		
8	MC_ReadStatus	"MC_ReadStatus_LXM32_1200_1500"		
9	MC_Reset	"MC_Reset_LXM32_1200_1500"		
10	MC_Stop	"MC_Stop_LXM32_1200_1500"		
11	MC_Home	"MC_Home_LXM32_1200_1500"		
12	MC_MoveAbsolute	"MC_MoveAbsolute_LXM32_1200_1500"		
13	MC_Jog	"MC_Jog_LXM32_1200_1500"		
14	MC_ReadActualPosition	"MC_ReadActualPos_LXM32_1200_1500"		
15	Temp			
16	RetVal_ReadInputData	Int		
17	RetVal_WriteOutputData	Int		



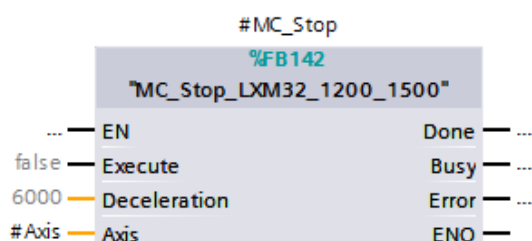
Network 5: Error Reset

Comment



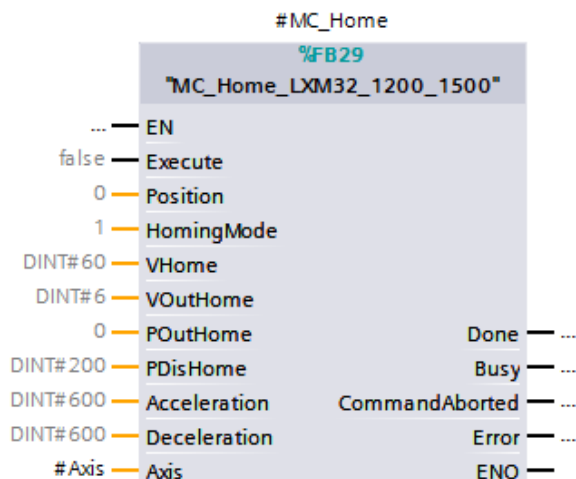
Network 6: Stop active movement

Comment



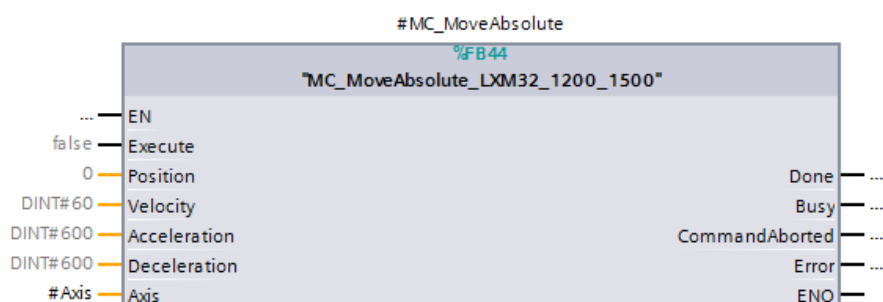
Network 7: Reference movement

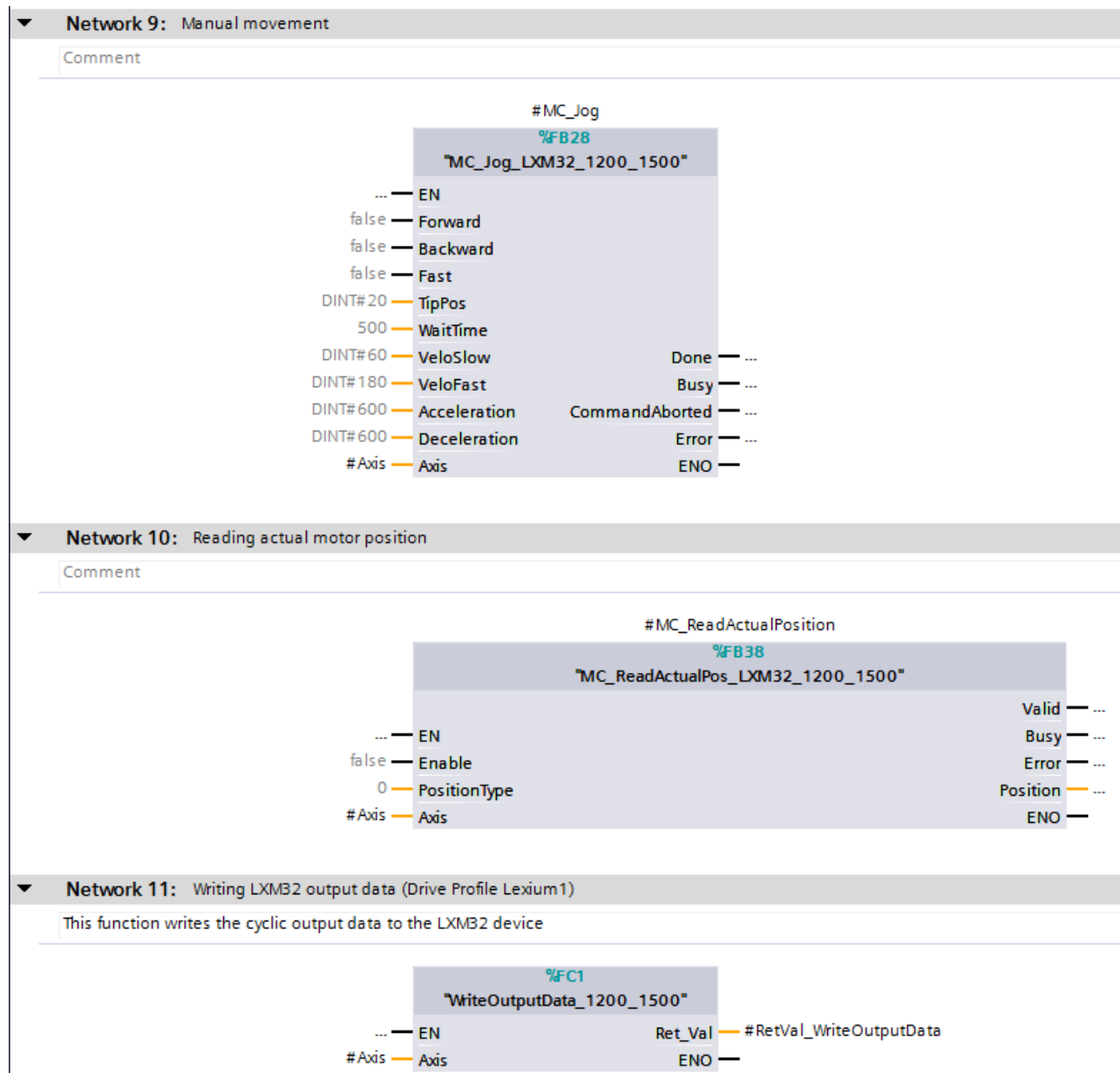
Comment



Network 8: Point to point movement (absolute)

Comment





The function „WriteOutputData_1200_1500“ must be called always at last after the call of the motion function blocks.

7 Library blocks

7.1 Explanation of common parameters

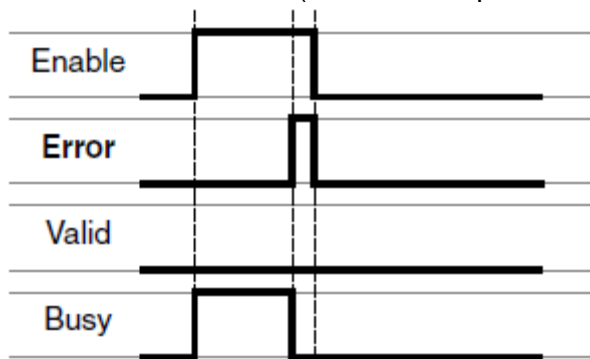
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	Starts (=TRUE) and stops (=FALSE) the block's execution. The block continues to be executed as long as TRUE is returned (level-sensitive).
	Execute	BOOL	<p>The block is executed once by a rising edge. With all motion blocks (except MC_Home) the input parameters are read after a rising edge during execution, and the movement is then continued with the new parameters.</p> <p>When the block has been executed (Busy = FALSE), the output parameters are maintained until FALSE is returned. The falling edge deletes the output parameters. If the input is already FALSE when execution is completed, the output parameters are issued during precisely one block call, and then deleted (edge-sensitive).</p>
OUT	Valid	BOOL	<p>Value range: TRUE, FALSE Initial value: FALSE FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.</p>
	Done	BOOL	<p>Value range: TRUE, FALSE Initial value: FALSE FALSE: Execution has not (yet) been terminated without an error. TRUE: Execution has been completed without an error.</p>
	Busy	BOOL	<p>Value range: TRUE, FALSE Initial value: FALSE FALSE: Execution of the function block has been terminated. TRUE: Function block is being executed. NOTE: In the operating mode Profile Velocity, the output remains TRUE even when the target velocity has been reached or Execute becomes FALSE. The output Busy is set to FALSE as soon as another function block such as MC_Stop is executed.</p>
	Error	BOOL	<p>Value range: TRUE, FALSE Initial value: FALSE FALSE: Execution of the function block is running, nor error has occurred up until now. TRUE: An error has occurred in the execution of the function block.</p>
	CommandAborted	BOOL	<p>Value range: TRUE, FALSE Initial value: FALSE FALSE: Execution has not (yet) been canceled without an error. TRUE: Execution has been aborted by another function block.</p>
INOUT	Axis	STRUCT	<p>Axis reference structure (Axis_Ref_LXM32_1200_1500) Name of the axis (instance) for which the function block is to be executed.</p>

7.2 Behavior of function blocks with the input Enable

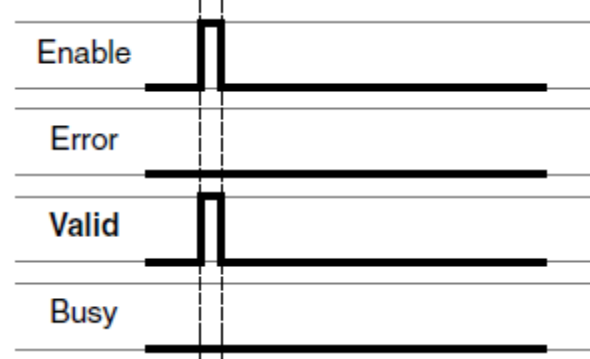
Example 1 Single execution without a detected error (execution requires more than one Cycle).



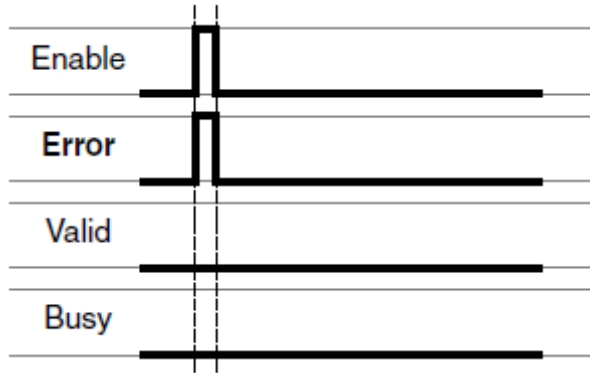
Example 2 Single execution with a detected error (execution requires more than one Cycle).



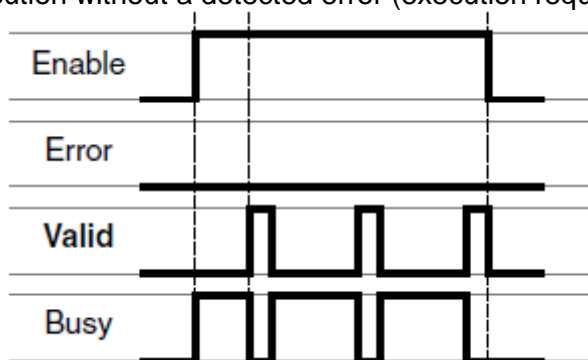
Example 3 Single execution without a detected error (execution requires only one Cycle).



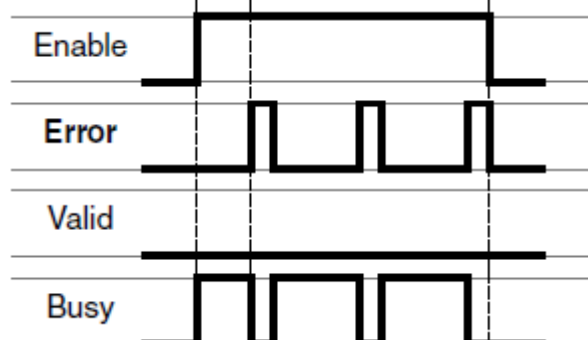
Example 4 Single execution with a detected error (execution requires only one Cycle).



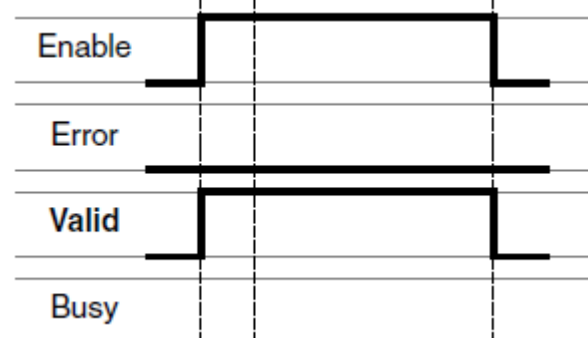
Example 5 Repeated execution without a detected error (execution requires more than one Cycle).



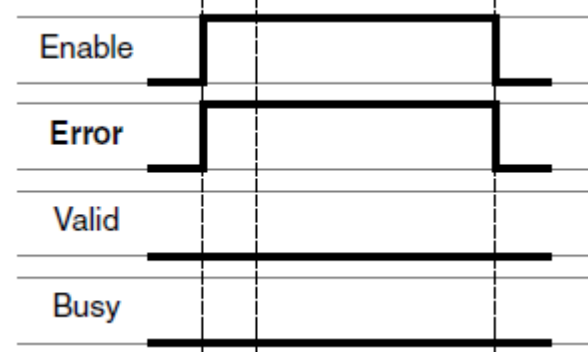
Example 6 Repeated execution with a detected error (execution requires more than one Cycle).



Example 7 Repeated execution without a detected error (execution requires only one Cycle).

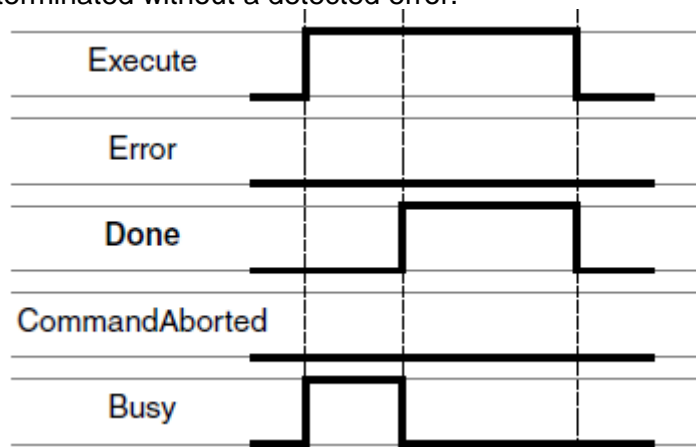


Example 8 Repeated execution with a detected error (execution requires only one Cycle).

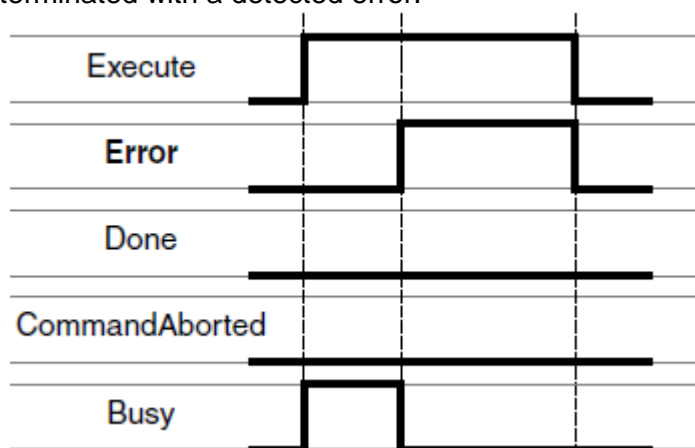


7.3 Behavior of function blocks with the input Execute

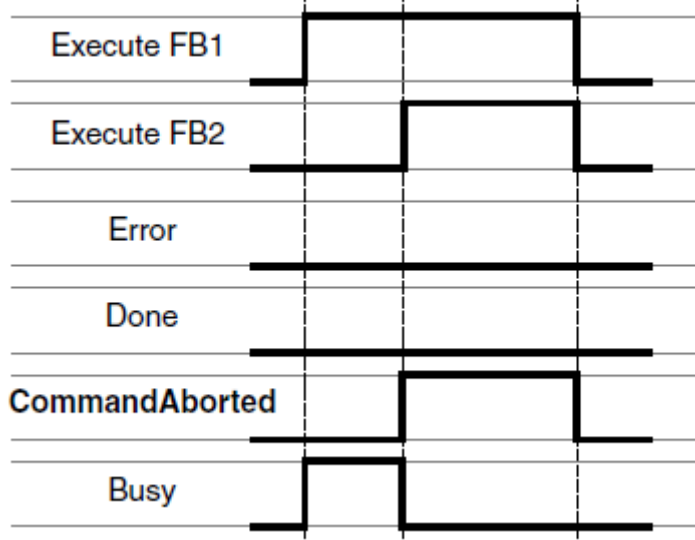
Example 1 Execution terminated without a detected error.



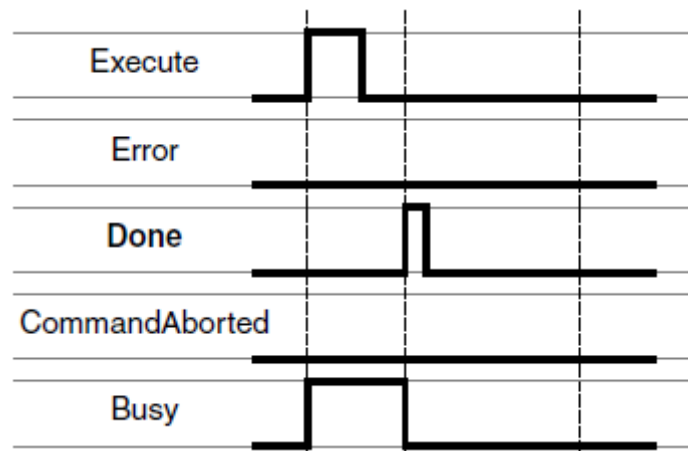
Example 2 Execution terminated with a detected error.



Example 3 Abortion of the execution because another function block has been started.

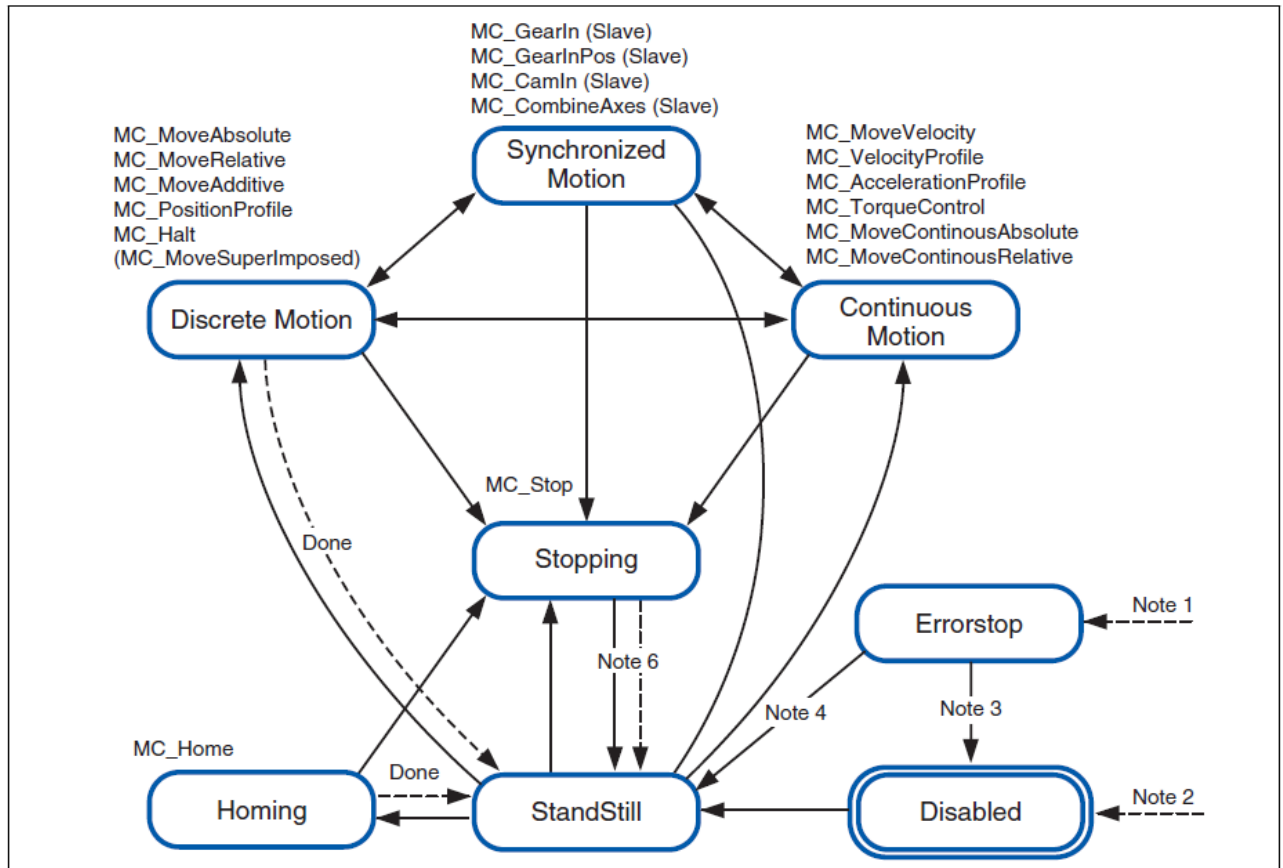


Example 4 After the input Execute was set to FALSE during a Cycle, the execution was terminated and no error was detected. The output Done is set to TRUE only for a single Cycle.



7.4 PLCopen state diagram

The state diagram represents the axis in terms of PLCopen. At any given point in time, the axis is exactly in one state. If a function block is executed or an error occurs, this may cause a state transition. The function block "MC_ReadStatus" delivers the current status of the axis.



Note 1: An error has been detected. (Transition from any state).

Note 2: The input Enable of the function block MC_Power is set to FALSE and no error has been detected (transition from any state).

Note 3: MC_Reset and MC_Power.Status = FALSE

Note 4: MC_Reset and MC_Power.Status = TRUE and MC_Power.Enable = True

Note 5: MC_Power.Enable = TRUE and MC_Power.Status = TRUE

Note 6: MC_Stop.Done = TRUE and MC_Stop.Execute = TRUE

7.5 Initialisation

7.5.1 Init_LXM32_1200_1500

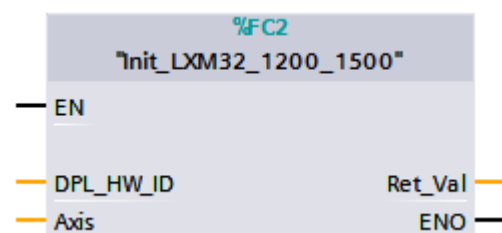
Function description

Initialisation of the “axis reference structure” after PLC startup.

The function must be called after each PLC startup. (e.g. in Startup OB 100)

The Hardware ID is stored in the axis reference structure. This is necessary to address the input and output data of the LXM32 (DriveProfileLexium1 – 26 Bytes in/out).

Graphical representation

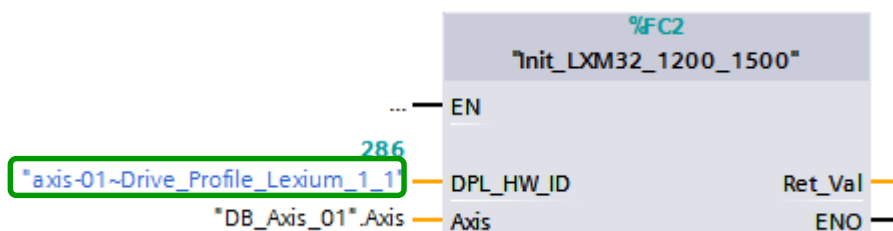


Parameter description:

Par. type	Parameter	Data type	Description
IN	DPL_HW_ID	HW_IO	Hardware ID of DriveProfileLexium1 (see HW Configuration)
IN_OUT	Axis	Axis_Ref_LXM32_1200_1500	Axis reference structure
OUT	Ret_Val	INT	Error number (value <>0 = error).

Note:

We suggest to use always symboliv names insatead of absolute addresses for the hardware ID of the DriveProfileLexium1.



7.5.2 ReadInputData_1200_1500 [FC1]

Function description

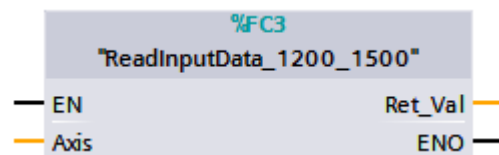
This function is used to read the 26 bytes input data of the DriveProfileLexium1.

This function calls the Siemens system functions DPRD_DAT to read the 26 bytes input data of the DriveProfileLexium1.

Since library version 2.0 this function must be called by the user in the application.

In earlier version is was called inside the motion function blocks.

To optimize the PLC cycle performance this function must now be called once by the user application.



Calling:

This function must be called each PLC cycle once before the motion function blocks are called.

7.5.3 WriteOutputData_1200_1500

Function description

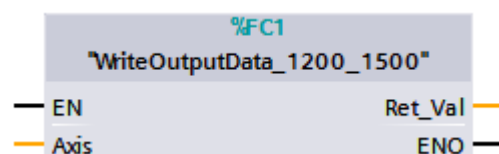
This function is used to write the 26 bytes output data of the DriveProfileLexium1.

This function calls the Siemens system functions DPWR_DAT to write the 26 bytes output data of the DriveProfileLexium1.

Since library version 2.0 this function must be called by the user in the application.

In earlier version is was called inside the motion function blocks.

To optimize the PLC cycle performance this function must now be called once by the user application.



Calling:

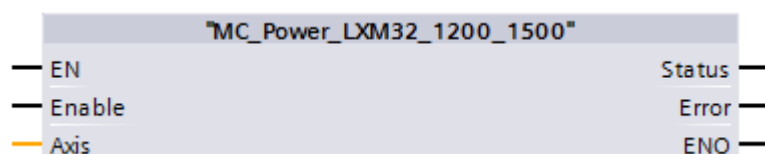
This function must be called each PLC cycle once after the motion function blocks are called.

7.5.4 MC_Power_LXM32_1200_1500

Function description

The function block enables or disables the power stage. TRUE at the input Enable enables the power stage. Once the power stage is enabled, the output Status is set. FALSE at the input Enable disables the power stage. Once the power stage is disabled, the output Status is reset. If errors are detected during execution, the output Error is set.

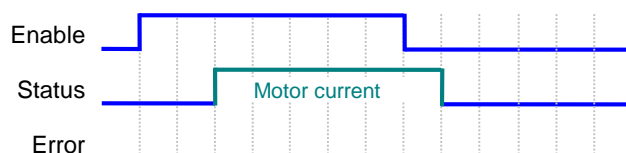
Graphical representation



Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: disable power stage TRUE: enable power stage
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Status	BOOL	Indicates the status of the motor current. FALSE: Power stage is disabled. TRUE: Power stage is enabled.
	Error	BOOL	TRUE: function finished with error

Phase diagram



7.6 Jog

In the operating mode Jog, a movement is made from the actual motor position in the desired direction. The velocity can be set. As long as the signal for the direction is available, a continuous movement is made in the desired direction.

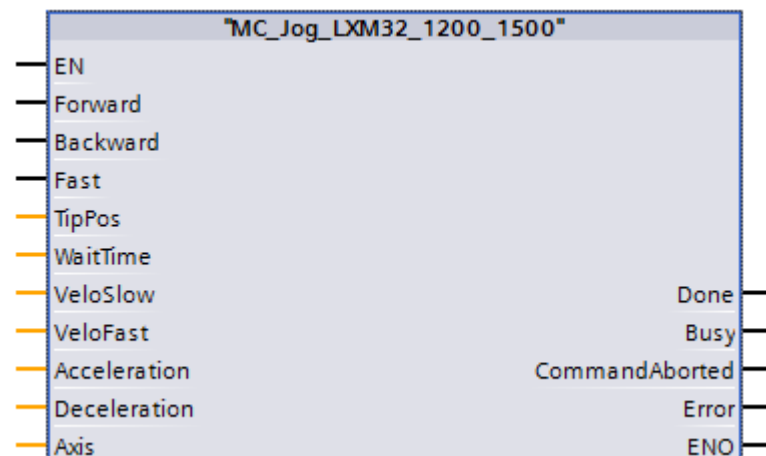
If movements in positive and negative directions are requested at the same time, there is no motor movement.

7.6.1 MC_Jog_LXM32_1200_1500

Function description

The function block starts the operating mode Jog. TRUE at the input Forward or the input Backward starts the jog movement. If both the inputs Forward and Backward are FALSE, the operating mode is terminated and the output Done is set. If both the inputs Forward and Backward are TRUE, the operating mode remains active, the jog movement is stopped and the output Busy remains set.

Graphical representation

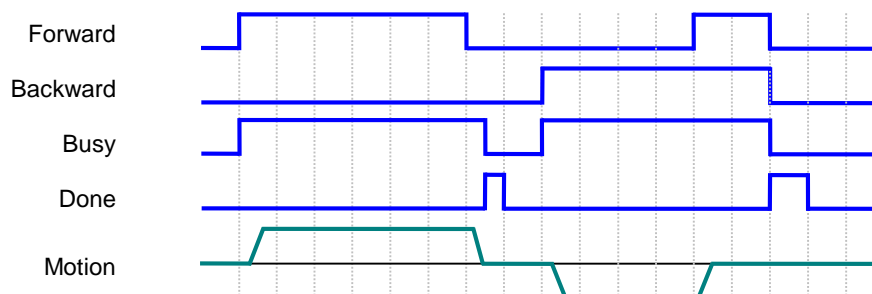


Parameter description

Par. type	Parameter	Data type	Description
IN	Forward	BOOL	FALSE: No movement in positive direction TRUE: Movement in positive direction is started.
	Backward	BOOL	FALSE: No movement in negative direction TRUE: Movement in negative direction is started.
	Fast	BOOL	The velocity can be changed during the movement. FALSE: Movement at the velocity set in VeloSlow. TRUE: Movement at the velocity set in VeloFast.
	TipPos	DINT	Value range: 0 ... 2147483647 Initial value: 20 0: Continuous movement is started immediately. >0: Movement by this distance value in usr units [usr_p]. The movement is stopped, the waiting time WaitTime starts. After the waiting time WaitTime has elapsed, a continuous movement is started.

IN	WaitTime	INT	Value range: 0 ... 65535 Initial value: 500 Waiting time in [ms]. If TipPos is >0, the waiting time WaitTime starts as soon as the adjusted distance has been covered. After the waiting time WaitTime has elapsed, a continuous movement is started.
	VeloSlow	DINT	Value range: Initial value: 60 Velocity in user units [usr_v]. If Fast = FALSE, the movement is made at this velocity.
	VeloFast	DINT	Value range: Initial value: 180 Velocity in user units [usr_v]. If Fast = TRUE, the movement is made at this velocity.
	Acceleration	DINT	Value for the acceleration ramp gradient in user units [usr_a] Value range: 1...2147483647; Initial value: 600.
	Deceleration	DINT	Value for the deceleration ramp gradient in user units [usr_a] Value range: 1...2147483647; Initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: Jog movement has stopped without error
	Busy	BOOL	TRUE: Jog movement is active
	CommandAborted	BOOL	TRUE: Jog movement aborted
	Error	BOOL	TRUE: Jog movement has stopped with error

Fehler! Verweisquelle konnte nicht gefunden werden.



7.7 Homing

The operating mode Homing is used to define a reference point. The reference point establishes an absolute position reference between the motor position and a defined axis position. The reference point can be defined by means of a reference movement or by means of position setting.

- Reference movement: Movement to a limit switch, a reference switch or the index pulse of the motor encoder. When the position is reached, a position reference is automatically created. This position becomes the absolute user-defined position.
- Position setting: The current motor position is set to a desired position value. The zero point is defined by the position value. Position setting is only possible when the motor is at a standstill.

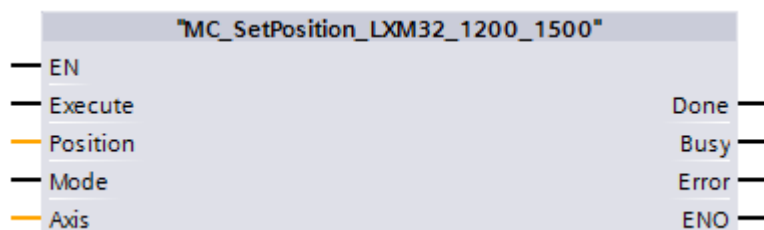
The operating mode Homing must be completed without an error for the new reference point to be valid.

7.7.1 MC_SetPosition_LXM32_1200_1500

Function description

This function block sets a position value at the actual position of the motor. The zero point is defined by the position value. The function block can only be used when the motor is at a standstill.

Graphical representation



Parameter description

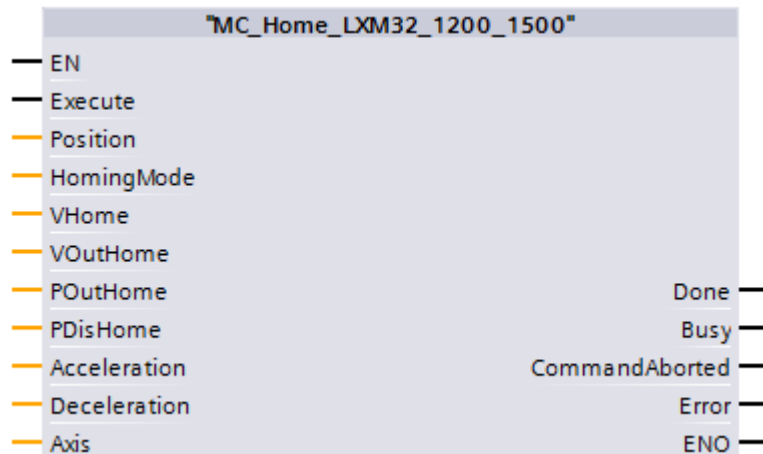
Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Position	DINT	Dimension setting position in user units [usr_p] Value range: - 2147483648..2147483647, initial value: 0.
	Mode	BOOL	FALSE: Set current motor position as <i>Position</i> . TRUE: Add <i>Position</i> to current motor position.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: function finished without error
	Busy	BOOL	TRUE: function is active
	Error	BOOL	TRUE: function finished with error

7.7.2 MC_Home_LXM32_1200_1500

Function description

The function block configures and starts a reference movement.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Position	DINT	Position is set as current motor position after successful reference movement [usr_p]. Value range: depends on scaling factor, initial value: 0.
	HomeMode	INT	LIMN with index pulse 1 : LIMP with index pulse 2: 7 = REF+ with index pulse, beyond REF, in direction of LIMN 8 = REF+ with index pulse, within REF, in direction of LIMN 9 = REF+ with index pulse, within REF, in direction of LIMP 10 = REF+ with index pulse, beyond REF, in direction of LIMP 11 = REF- with index pulse, beyond REF, in direction of LIMN 12 = REF- with index pulse, within REF, in direction of LIMN 13 = REF- with index pulse, within REF, in direction of LIMP 14 = REF- with index pulse, beyond REF, in direction of LIMP 17 = LIMN 18 = LIMP 23 = REF+, beyond REF, in direction of LIMN 24 = REF+, within REF, in direction of LIMN 25 = REF+, within REF, in direction of LIMP 26 = REF+, beyond REF, in direction of LIMP 27 = REF-, beyond REF, in direction of LIMN 28 = REF-, within REF, in direction of LIMN 29 = REF-, within REF, in direction of LIMP 30 = REF-, beyond REF, in direction of LIMP 33 = on index pulse, in direction of LIMN 34 = on index pulse, in direction of LIMP
	VHome	DINT	Speed for searching the limit or reference switch [usr_v]. Drive stops when switching edge has been detected. Value range: 1...2147483647; Initial value: 60.

Par. type	Parameter	Data type	Description
IN	VOutHome	INT	Speed for clearance movement back to the switching edge [usr_v]. The max. travel distance when searching for the switching edge can be restricted with the parameter <i>POutHome</i> . Value range: 1...2147483647; Initial value: 6.
	POutHome	DINT	0: Clearing monitor switched off. >0: Run-off [usr], i.e. max. travel distance when searching for the switching edge. If the switching edge is not found in this distance, the reference movement is interrupted with an error. Value range: 0..2147483647, initial value: 0.
	PDisHome	DINT	Distance between the switching edge and the reference point [usr]. At end of movement, the drive moves back towards switching edge until the distance has been reached. Value range: 1..2147483647, initial value: 200.
	Acceleration	DINT	Value for the acceleration ramp gradient [user units] Value range: 1...2147483647; Initial value: 600.
	Deceleration	DINT	Value for the acceleration ramp gradient [user units] Value range: 1...2147483647; Initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

7.8 Torque control

⚠ WARNING

EXCESSIVELY HIGH VELOCITY DUE TO INCORRECT LIMIT VALUE

Without a proper limit value, the motor can reach a very high velocity in this operating mode.

- Check the parameterized velocity limitation.

Failure to follow these instructions can result in death, serious injury or equipment damage.

You can set a target torque in the operating mode Profile Torque. The movement is made with this target torque in the operating mode Profile Torque.

7.8.1 MC_TorqueControl_LXM32_1200_1500

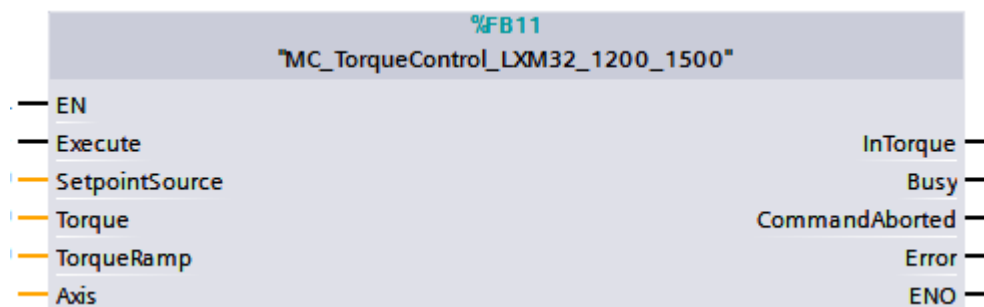
Function description

The function block starts the operating mode Profile Torque. In the operating mode Profile Torque, a movement is made with a desired target torque.

The source of the target torque can be selected by the input “SetpointSource”.

When the target torque is reached, the output InTorque is set. The input TorqueRamp lets you set the slope of the motion profile for the torque.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	SetpointSource	UINT	Target value source: 0: Input Torque (LXM32M + LXM32i) 1: Analogue input (LXM32M – IO Modul Slot 1) 2: PTI Interface (LXM32M)
	Torque	INT	Value range: -30000 ... 30000 Initial value: 0 Target torque The value corresponds to 0.1% of the nominal torque of the motor. Example: Torque = 300 corresponds to 30% of the nominal torque of the motor. Use the parameter _M_M_O to get the nominal torque of the motor.
	TorqueRamp	DINT	Value range: Initial value: 100000 The input TorqueRamp is used in the operating mode Profile Torque. The value corresponds to 0.1% per second of the nominal torque of the motor. Example: If TorqueRamp = 1000, then 100% of the nominal

			torque of the motor is reached in one second. Use the parameter _M_M_O to get the nominal torque of the motor.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	InTorque	BOOL	FALSE: Target torque is not yet reached. TRUE: Target torque reached.
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

Notes

In the operating mode Profile Torque, a position overtravel does not trigger an error. In the case of a position overtravel, the drive loses its reference position. After a position overtravel, an absolute positioning movement is still performed, but it is no longer consistent with the original homing. You can use the function block MC_ReadStatus_LXM32 to check whether the drive is still homed.

7.9 Profile Velocity mode

You can set a target velocity in the operating mode Profile Velocity. The movement is performed with this target velocity in the operating mode Profile Velocity. The movement continues until a new target velocity is set or until the operating mode is aborted.

Transitions between two target velocities are performed on the basis of a motion profile. The motion profile is determined by the profile generator in the drive on the basis of the actual velocity, the target velocity and the acceleration and deceleration ramps.

7.9.1 MC_Move_Velocity_LXM32_1200_1500

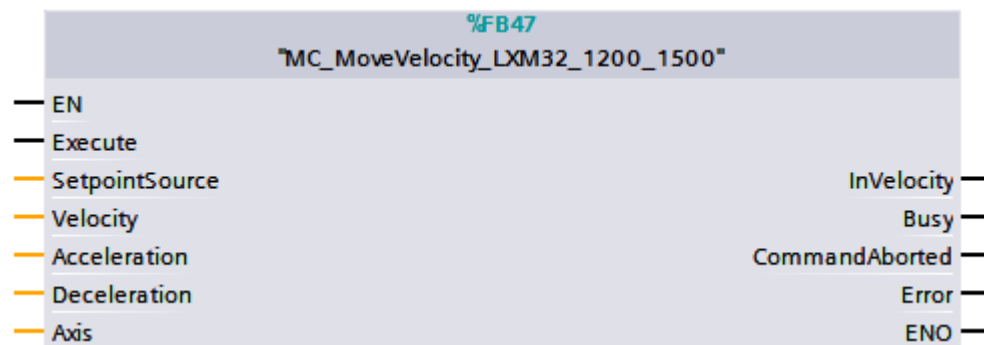
Function description

The function block starts the operating mode Profile Velocity.

The source of the target torque can be selected by the input "SetpointSource".

When the target velocity is reached, InVelocity is set.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	SetpointSource	UINT	Target value source: 0: Input Torque (LXM32M + LXM32i) 1: Analogue input (LXM32M – IO Modul Slot 1)
	Velocity	DINT	Target velocity [user units]. Value range: -2147483647...2147483647; Initial value: 0
	Acceleration	DINT	Value for the acceleration ramp gradient [user units] Value range: 1...2147483647; Initial value: 600.
	Deceleration	DINT	Value for the acceleration ramp gradient [user units] Value range: 1...2147483647; Initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference structure

OUT	InVelocity	BOOL	FALSE: Target velocity is not yet reached. TRUE: Target velocity reached.
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

Note

In the operating mode Profile Velocity, a position overtravel does not trigger an error. A position overtravel results in a loss of the zero point.

7.10 Profile position mode

The following settings can be made in the operating mode Profile Position:

- Target position
- Type of movement (relative movement or absolute movement)
- Target velocity
- Acceleration and deceleration ramps

The movement to the target position is made on the basis of a motion profile. The motion profile is calculated by the profile generator in the drive. The calculation is performed on the basis of the actual position and the target position, the actual velocity and the target velocity and the acceleration and deceleration ramps.

In the operating mode Profile Position, absolute movements, relative movements and additive movements are possible.

- Absolute movement with reference to the zero point
- Relative movement with reference to the actual position
- Additive movement with reference to the previous target position

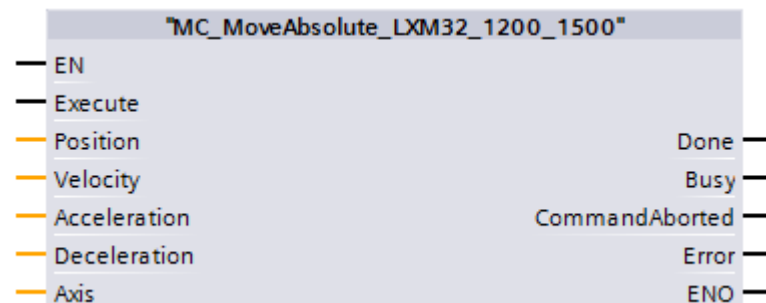
A zero point must be defined with the operating mode Homing prior to the first absolute movement.

7.10.1 MC_MoveAbsolute_LXM32_1200_1500

Function description

The function block starts a movement to the absolute target position Position at velocity.

Graphical representation



Parameter description

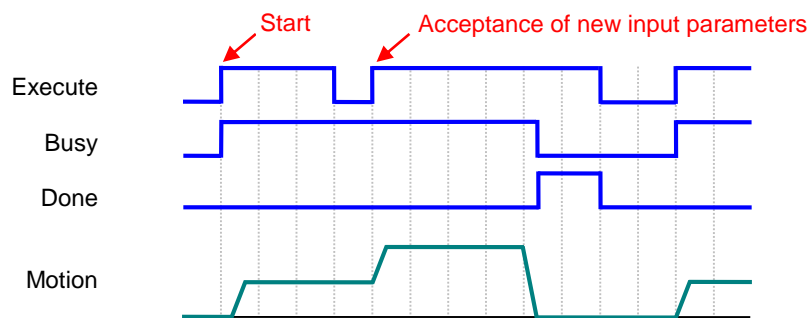
Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Position	DINT	Value for the absolute target position [usr]. Value range: depends on scaling factor, initial value: 0.
	Velocity	DINT	Value for the set speed of the movement [usr]. Value range: 1...2147483647; Initial value: 60.
	Acceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
	Deceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active

	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

Note

Absolute positioning requires a valid zero point. You can use the function block MC_ReadStatus_LXM32 to check for a valid zero point.

Fehler! Verweisquelle konnte nicht gefunden werden.

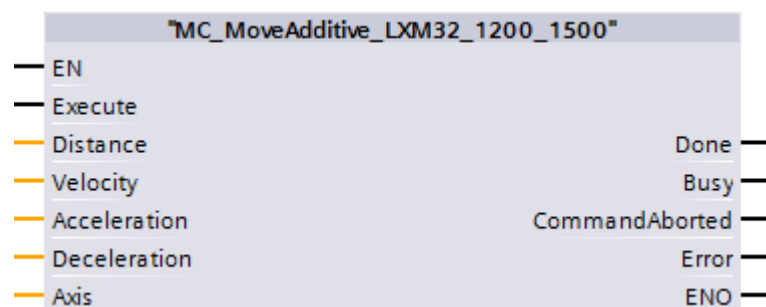


7.10.2 MC_MoveAdditive_LXM32_1200_1500

Function description

The function block starts a movement to the original target position plus distance at velocity.

Graphical representation



Parameter description

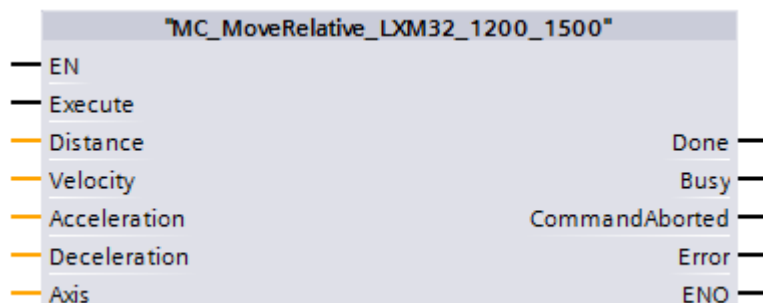
Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Distance	DINT	Value for the travel distance that is added to the current target position, and thus determines the new target position [usr]. Value range: depends on scaling factor, initial value: 0.
	Velocity	DINT	Value for the set speed of the movement [usr]. Value range: 1...2147483647; Initial value: 60.
	Acceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
	Deceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

7.10.3 MC_MoveRelative_LXM32_1200_1500

Function description

The function block starts a movement by distance at velocity.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Distance	DINT	Value for the travel distance that is added to the current motor position, and thus determines the new target position [usr]. Value range: depends on scaling factor, initial value: 0.
	Velocity	DINT	Value for the set speed of the movement [usr]. Value range: 1...2147483647; Initial value: 60.
	Acceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
	Deceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

7.11 Electronic gear

In the operating mode Electronic Gear, movements are carried out according to externally supplied reference value signals. A new position reference value is calculated on the basis of these reference value signals plus an adjustable gear ratio.

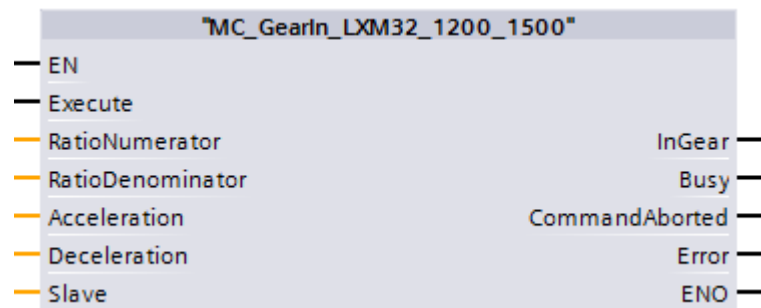
This operating mode is not supported by LXM32i.

7.11.1 MC_GearIn_LXM32_1200_1500

Function description

The function block starts the operating mode Electronic Gear with the method Velocity Synchronization. In the operating mode Electronic Gear, movements are carried out according to externally supplied reference value signals. A new velocity value is calculated on the basis of these reference value signals plus an adjustable gear ratio. Reference value signals supplied during an interruption caused by Halt or by an error of error class 1 are not considered.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	RatioNumerator	DINT	Gear ratio numerator. Value range: -2147483648 .. 2147483647, initial value: 1.
	RatioDenominator	DINT	Gear ratio denominator. Value range: 1 .. 2147483647, initial value: 1.
	Acceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
	Deceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
IN_OUT	Slave	STRUCT	Axis reference structure
OUT	InGear	BOOL	FALSE: The electronic gear is disabled. TRUE: The electronic gear is enabled.
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

Notes

This requires the parameter GEARratio to be 0 (see product manual). This way, RatioNumerator and RatioDenominator are used to calculate the gear ratio

The enabled direction of movement in the operating mode Electronic Gear is set via the parameter GEARdir_enabl.

The operating mode Electronic Gear with the method Position Synchronization is started with the function block GearInSync_LXM32 .

In the operating mode Electronic Gear with the method Velocity Synchronization, a position overtravel does not trigger an error. A position overtravel results in a loss of the zero point.

7.11.2 GearInSync_LXM32_1200_1500

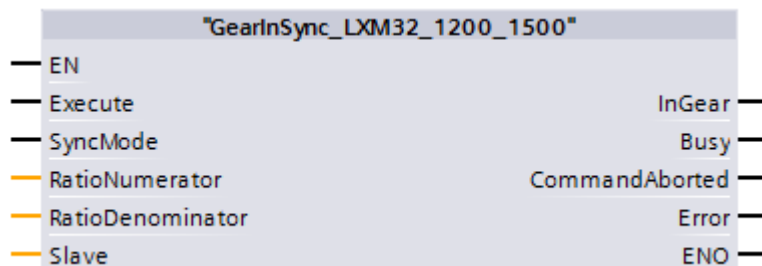
Function description

The function block starts the operating mode Electronic Gear with the method Position Synchronization. In the operating mode Electronic Gear, movements are carried out according to externally supplied reference value signals. A new position value is calculated on the basis of these reference value signals plus an adjustable gear ratio.

In the case of position synchronization without compensation movement, the movement is made synchronously (position synchronicity) with the supplied reference value signals. Reference value signals supplied during an interruption caused by Halt or by an error of error class 1 are not considered.

In the case of position synchronization with compensation movement, the movement is made synchronously (position synchronicity) with the supplied reference value signals. Reference value signals supplied during an interruption caused by Halt or by an error of error class 1 are considered and compensated for. See the product manual for additional information on compensation movements.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	SyncMode	BOOL	Value range: FALSE, TRUE Initial value: TRUE FALSE: Position synchronization without compensation movement. TRUE: Position synchronization with compensation movement
	RatioNumerator	DINT	Gear ratio numerator. Value range: -2147483648 .. 2147483647, initial value: 1.
	RatioDenominator	DINT	Gear ratio denominator. Value range: 1 .. 2147483647, initial value: 1.
IN_OUT	Slave	STRUCT	Axis reference structure
OUT	InGear	BOOL	TRUE: The Gear ratio is reached.
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

Notes

This requires the parameter GEARratio to be 0 (see product manual). This way, RatioNumerator and RatioDenominator are used to calculate the gear ratio.

The velocity of the compensation movement (SyncMode = TRUE) is limited by:

the maximum current (parameter CTRL_I_max).

Maximum velocity of the motor.

The enabled direction of movement in the operating mode Electronic Gear is set via the parameter GEARdir_enabl.

The operating mode Electronic Gear with the method Velocity Synchronization is started with the function block MC_GearIn_LXM32 .

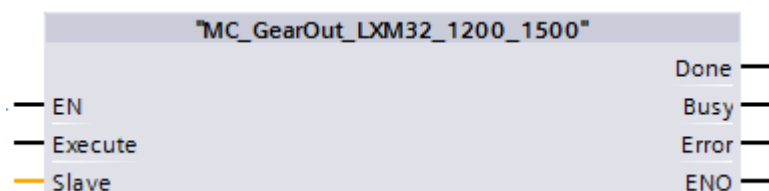
Once the operating mode is active, the compensation movement must not exceed the maximum permissible position deviation. If the required compensation movement exceeds the maximum permissible position deviation, a following error is signaled.

7.11.3 MC_GearOut_LXM32_1200_1500

Function description

The function block terminates the operating mode Electronic Gear.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
IN_OUT	Slave	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

7.12 Stopping

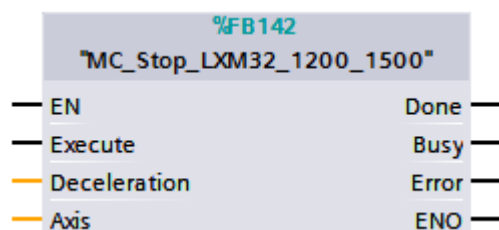
Each operating mode can be canceled by stopping. Stopping the operating mode does not generate an error.

7.12.1 MC_Stop_LXM32_1200_1500

Function description

The function block is used to stop the current movement with the set deceleration ramp.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Deceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

Note

- The function block can only be interrupted by disabling the power stage via the function block MC_Power_LXM32 .
- As long as the input Execute is TRUE, no other function block with the exception of MC_Power_LXM32 can be started.

7.12.2 MC_Halt_LXM32_1200_1500

Function description

The function block is used to stop the motor under normal operating conditions. The current movement is interrupted; it can be resumed. If a Halt is triggered, there is a transition of the PLCopen state to "DiscreteMotion" until the motor has reached a standstill. Once the motor has reached a standstill, the output Done is set and the state transitions to "StandStill".

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Deceleration	DINT	Value for the acceleration ramp gradient [usr] Value range: 1...2147483647; Initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error

7.13 Fast position capture

Position capture via a signal input captures the current position at the point in time at which an edge is detected at one of the digital Capture inputs.

Settings:

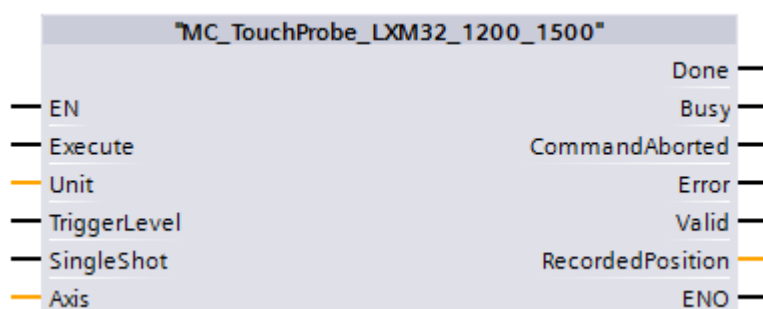
- Position capture can be triggered by a rising edge or a falling edge at the signal input.
- It is possible to use one-time or continuous position capture.

7.13.1 MC_TouchProbe_LXM32_1200_1500

Function description

The function block configures and starts position capture.

Graphical representation



Parameter description

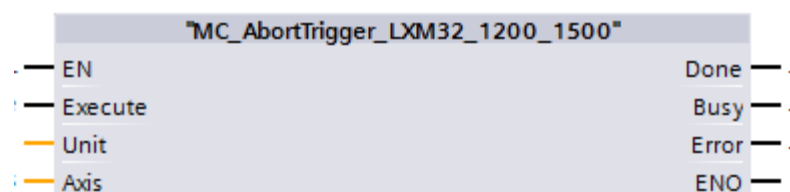
Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Unit	INT	Value range: 1 ... 2 Initial value: 1 1: Start position capture via Capture input 1. 2: Start position capture via Capture input 2.
	TriggerLevel	BOOL	Value range: FALSE, TRUE Initial value: FALSE FALSE: Start position capture at falling edge. TRUE: Start position capture at rising edge.
	SingleShot	BOOL	Value range: FALSE, TRUE Initial value: TRUE FALSE: Set continuous position capture. Continuous capture means that the motor position is captured anew at every edge. The the previously captured value is lost. TRUE: Sets one-time position capture. One-time capture means that the position is captured at the first edge. The capture value is not overwritten by a new edge.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: block execution aborted
	Error	BOOL	TRUE: block finished with error
	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	RecordedPosition	BOOL	Value range: -2147483648 ... 2147483647 Initial value: 0 Captured motor position

7.13.2 MC_AbortTrigger_LXM32_1200_1500

Function description

The function block is used to terminate position capture.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Unit	INT	Value range: 1 ... 2 Initial value: 1 1: Stop position capture via Capture input 1. 2: Stop position capture via Capture input 2.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

7.14 Read parameter

The following functions blocks allow you to read drive parameters such as the actual position or the actual velocity.

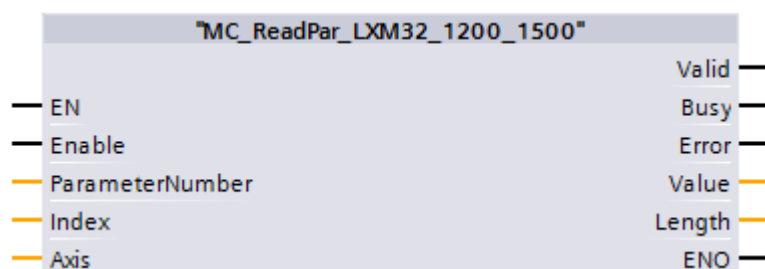
An additional function block provides read access to individual parameters of the device. See the product manual for a description of the parameters.

7.14.1 MC_ReadPar_LXM32_1200_1500

Function description

The function block reads an object from the device parameter list.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
	ParameterNumber	INT	Value range: 0 ... 32767 Initial value: 1000 Number of the parameter: 1: Target position of profile generator _RAMP_p_target (7938) [usr_p] 2: Positive position limit of software limit switch MON_swLimP (1544) [usr_p] 3: Negative position limit of software limit switch MON_swLimN (1546) [usr_p] 4: Monitoring of the positive software limit switch (Aktiviert: Bit 0 = 0. Deaktiviert: Bit 0 = 1) 5: Monitoring of the negative software limit switch (Aktiviert: Bit 0 = 0. Deaktiviert: Bit 0 = 1) 10: Actual velocity. _n_act (7696) [min-1] 11: Target velocity. _RAMP_v_target (7946) [usr_v] 1000: Selection via index (Profibus Parameter Address)
	Index	INT	Value range: 0 ... 32767 Initial value: 0 Index (Profibus parameter address) of parameter to be read. Only valid if ParameterNumber = 1000. See the product manual for an overview of the parameters.
IN_OUT	Axis	STRUCT	Axis reference structure

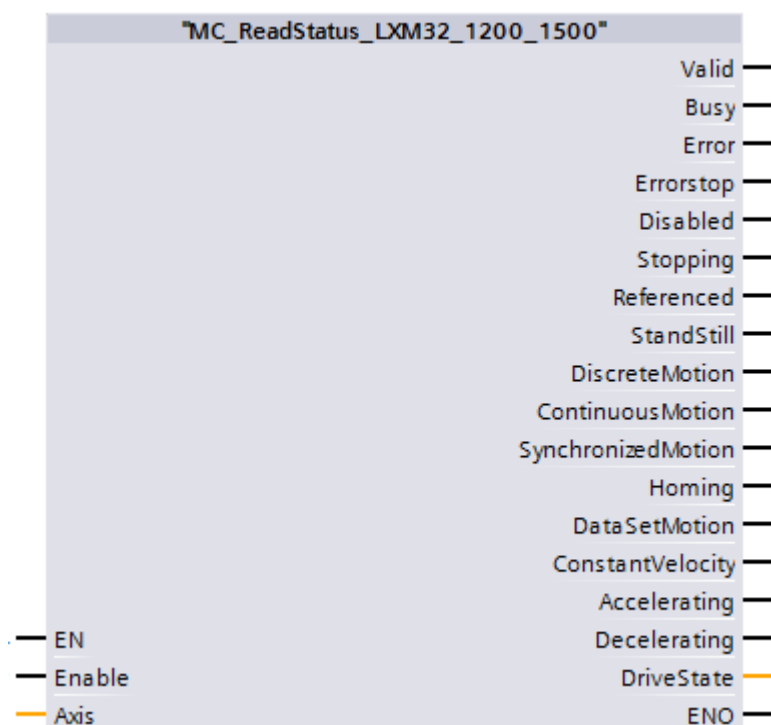
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	Value	DINT	Value range: -2147483648 ... 2147483647 Initial value: 0 Value of the parameter.
	Length	INT	Value range: 0 ... 32767 Initial value: 0 Length of the parameter in bytes.

7.14.2 MC_ReadStatus_LXM32_1200_1500

Function description

The function block is used to read the current status of the device.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
IN_OUT	Axis	STRUCT	Axis reference structure

OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	Errorstop	BOOL	TRUE: The movement has been interrupted by an error.
	Disabled	BOOL	FALSE: Power stage is enabled. TRUE: Power stage is disabled
	Stopping	BOOL	TRUE: The function block MC_Stop_LXM32 is being executed or the movement is being stopped.
	Referenced	BOOL	TRUE: The zero point (reference point) is valid.
	StandStill	BOOL	TRUE: The movement has been stopped.
	DiscreteMotion	BOOL	TRUE: The operating mode Profile Position has been started.
	ContinuousMotion	BOOL	TRUE: The operating mode Profile Velocity has been started.
	SynchronizedMotion	BOOL	TRUE: A synchronized movement at a constant velocity is performed. (for example, in the operating mode Electronic Gear)
	Homing	BOOL	TRUE: The operating mode Homing has been started.
	ConstantVelocity	BOOL	TRUE: A movement at a constant velocity is performed.
	Accelerating	BOOL	TRUE: The motor accelerates.
	Decelerating	BOOL	TRUE: The motor decelerates.
	DriveState	INT	Drive operating state 2 Not Ready To Switch On 3 Switch On Disabled 4 Ready To Switch On 6 Operation Enabled 7 Quick Stop Active 8 Fault Reaction Active 9 Fault

Notes

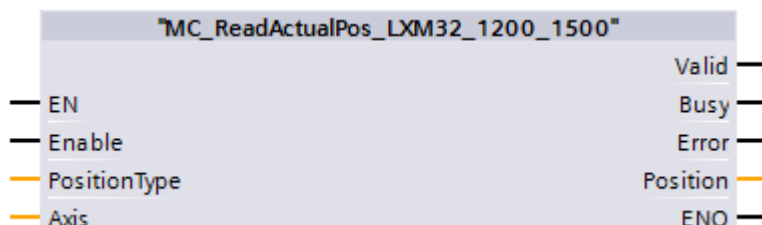
At any given point in time, the drive is exactly in one of the states: StandStill, DiscreteMotion, ContinuousMotion, Stopping, Disabled or ErrorStop. The corresponding output is then TRUE.

7.14.3 MC_ReadActualPos_LXM32_1200_1500

Function description

The function block is used to read the actual velocity of the motor.

Graphical representation



Parameter description

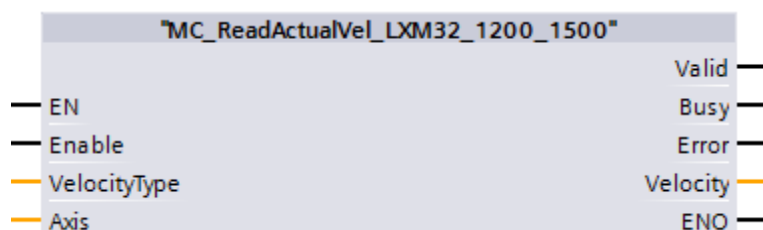
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
	PositionType	INT	Value range: 0...7, Initial value: 0 Selection of the position to be read from the drive: 0: Actual motor position in user units (7706) [usr_p] 1: Actual motor position in internal units (7700) [inc] 2: Target position of profile generator in user units (7704) [usr_p] 3: Target position of profile generator in internal units (7698) [inc] 4: Actual position of Encoder 2 (Modul) in user units (7732) [usr_p] 5: Actual position of Encoder 2 (Modul) in internal units (7730) [inc] 6: Actual position of profile generator in user units (7940) [usr_p] 7: Actual position at position interface (PTI) in internal units (2058) [inc]
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	Position	DINT	Motor's current actual position In usr or Inc depending on the selected position to be read with <i>PositionType</i> input.

7.14.4 MC_ReadActualVel_LXM32_1200_1500

Function description

The function block is used to read the actual velocity of the motor.

Graphical representation



Parameter description

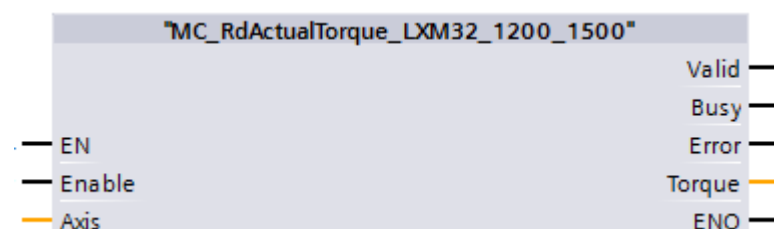
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
	VelocityType	INT	Value range: 0 ... 3 Initial value: 0 Specification of the source of the velocity: 0: Actual velocity of the motor in user units [usr_v] 1: Reference speed of rotation _n_ref (7694) [min ⁻¹] 2: Actual velocity of profile generator in user units (_RAMP_v_act 7948) [usr_v] 3: Actual velocity at PTI interface (_v_PTI_act 2060) [inc/s]
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	Velocity	DINT	Velocity value of the source selected for the input VelocityType.

7.14.5 MC_RdActualTorque_LXM32_1200_1500

Function description

The function block is used to read the actual torque of the motor.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	Torque	INT	Actual torque value. 100.0 % correspond to the continuous stall torque _M_M_0. The read value is indicated in increments of 0.1 %. Example: The value Torque = 300 is read. This means that currently effective torque amounts to 30 % of the nominal torque of the motor.

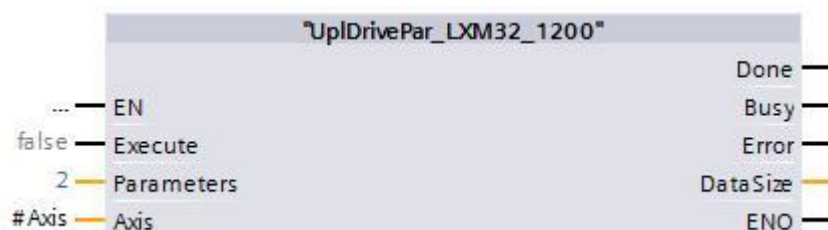
7.14.6 UploadParameter_LXM32_1200/1500

Function description

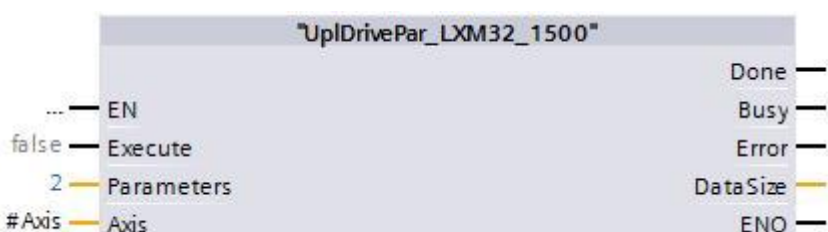
Stores all the drive parameters in a DB. The DB requires 5900 words (Firmware PR912.00 V1.22.00). The size depends on the LXM32 Firmware Revision. It is possible that the size will increase with a newer revision because of enhancements.

The memory size relates to the firmware version PR912.00 V1.22.00.

Graphical representation



This block must be used with a [S7-1200](#) PLC



This block must be used with a [S7-1500](#) PLC

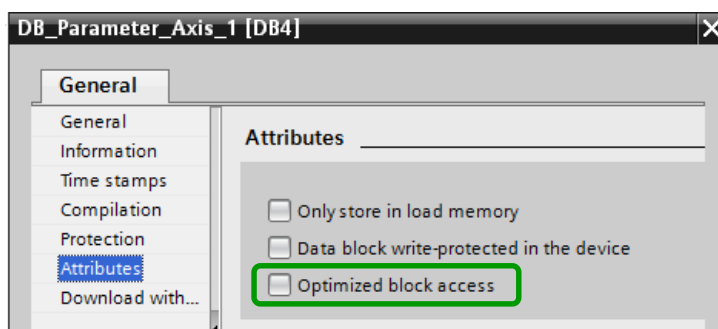
Data Block Example:

Project1 ▸ PLC_1 [CPU 1516-3 PN/DP] ▸ Program blocks ▸ DB_DriveBackup [DB2]

	Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Parameter	Array[0..5900] of Word		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Note:

In the properties of this data block the “optimized block access” must be switched off.



Parameter description

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Parameters	DB_ANY	DB no (data block with a size of 5900 words is required in Firmware version V1.22.00)
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	DataSize	INT	Number of WORDS which are needed. (The value could be different caused by the firmware revision)

Note:

The function block can only be executed if the drive is in the operating state 3 Switch On Disabled (operating state of drive). To transition to this state, disable the power stage with the function block MC_Power_LXM32 .

The two function blocks DownloadDriveParameter_LXM32 and UploadDriveParameter_LXM32 allow you to save the parameters stored in a device to an identical device without using the commissioning software.

7.15 Write parameter

The following function blocks allow you to write drive parameters, for example the values for the acceleration and deceleration ramps.

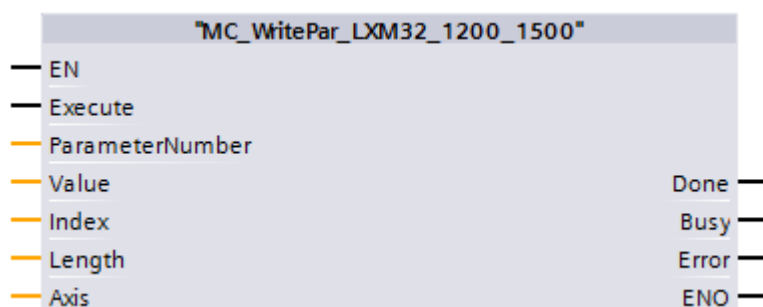
An additional function block provides write access to individual parameters of the device. See the product manual for a description of the parameters.

7.15.1 MC_WritePar_LXM32_1200_1500

Function description

The function block is used to write a value to a specific parameter.

Graphical representation



Parameter description

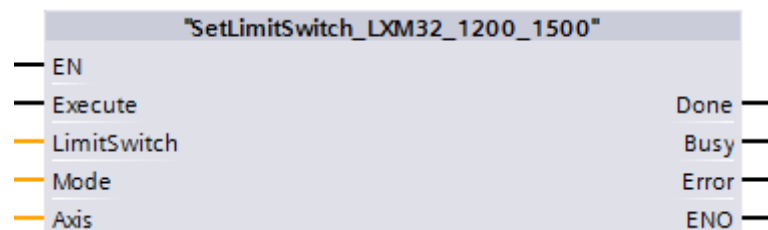
Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	ParameterNumber	UINT	Value range: 0 ... 32767 Initial value: 1000 2: Position of the software limit switch in positive direction (Value in usr) 3: Position of the software limit switch in negative direction (Value in usr) 4: Activate (Value = 1) or deactivate (Value = 2) software limit switch in positive direction 5: Activate (Value = 1) or deactivate (Value = 2) software limit switch in negative direction 1000: The parameter to be written is set via the input Index. See the product manual for a list of the parameters with the corresponding Profibus/Profinet parameter address.
	Value	DINT	Value range: -2147483648...2147483647 Initial value: 0 New value to be written to the parameter. The units of the values depend on the parameter.
	Index	UINT	Value range: 0 ... 32767 Initial value: 0 Index of the parameter to be written. See the product manual for a list of the parameters with index and subindex. Can only be used if the input ParameterNumber = 1000. See the product manual for a list of the parameters with the corresponding Profibus/Profinet address.
	Length	UINT	Value range: 0 ... 32767 Initial value: 0 Length of the parameter to be written in bytes.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

7.15.2 SetLimitSwitch_LXM32_1200_1500

Function description

Enabling / disabling the limit switches, and adjusting their operating sense.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	LimitSwitch	INT	1: Positive limit switch LIMP 2: Negative limit switch LIMN Value range: 1..2, initial value: 1.
	Mode	BOOL	0: Deactivate limit switch 1: Enable limit switch for normally-open operation 2: Enable limit switch for normally-closed operation Value range: 0..2, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

Notes

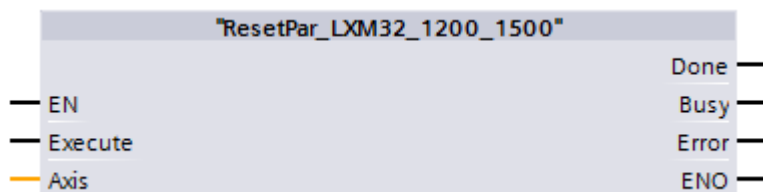
The function block can only be executed if the drive is in the operating state 4 Ready To Switch On (operating state of drive).

7.15.3 ResetParameters_LXM32_1200_1500

Function description

Set back the persistent parameters to the default values.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

Note:

The function block can only be executed if the drive is in the operating state 4 Ready To Switch On (operating state of drive). To transition to this state, disable the power stage with the function block MC_Power_LXM32 .

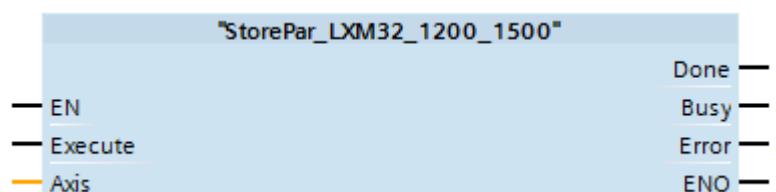
All the user set parameters will be lost if no backup has been made onto the data carrier with the commissioning software.

7.15.4 StorePar_LXM32_1200_1500

Function description

Saving all the User parameters in a non-volatile memory (EEPROM).

Graphical representation



Parameter description

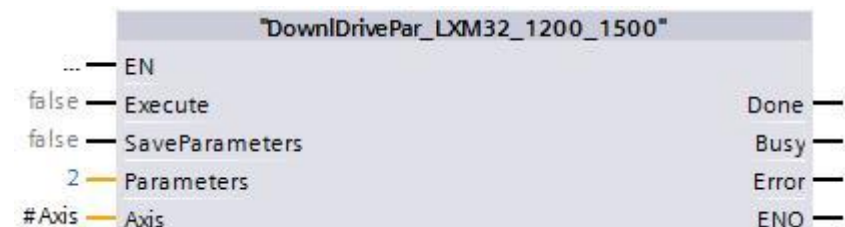
Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

7.15.5 DownloadParameter_LXM32_1200_1500

Function description

Writing all variable parameters into the drive.

Graphical representation

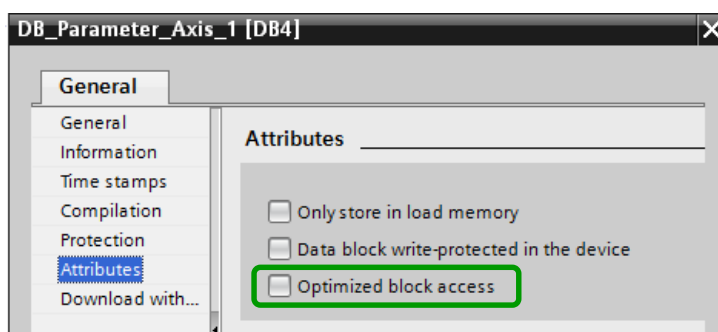


Data block example:

Project1 > PLC_1 [CPU 1516-3 PN/DP] > Program blocks > DB_DriveBackup [DB2]									
DB_DriveBackup									
	Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment	
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	Parameter	Array[0..5900] of Word		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Note:

In the properties of this data block the “optimized block access” must be switched off.



Parameter description

Par-type	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	SaveParameters	BOOL	TRUE: Store to EEPROM after download is finished
	Parameters	DB	DB no (data block with a size of 5900 words is required in Firmware version V1.22.00)
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

Note:

It is recommended to make an upload of all parameters with the function block UpIDrivePar_LXM032_1200 / UpIDrivePar_LXM032_1500 before downloading the parameters within this function block.

- The drive must be in disabled state otherwise an error will occur.
- If required, store the new drive parameters into the drive non-volatile memory (EEPROM) with the Input "SaveParameters" = TRUE. If not, the parameters will be lost if the drive is switched off.
- In case the data are not saved non-volatile, at exchanging a device the controller must remain under voltage, in order for the data be reloaded after the exchange.
- All the parameters of the Lexium 32 are written (restored).

7.16 Inputs/outputs

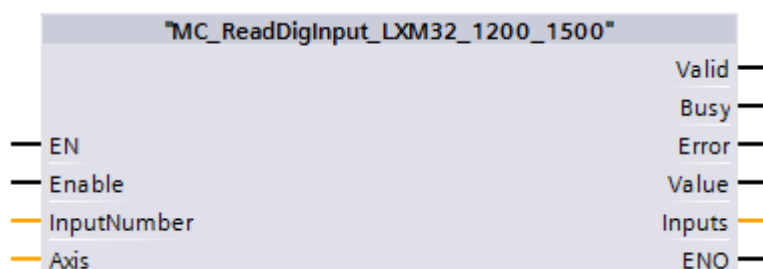
Apart from the process image, in which the digital inputs and outputs of the target system are displayed, other blocks are available, which provide access to the digital inputs/outputs of every drive in the system. The 24V signal interface of the drive provides 6 inputs and 3 outputs, which are assigned to functions such as limit switches.

7.16.1 MC_ReadDigInput_LXM32_1200_1500

Function description

Reading the drive's current input status.

Graphical representation



Parameter description

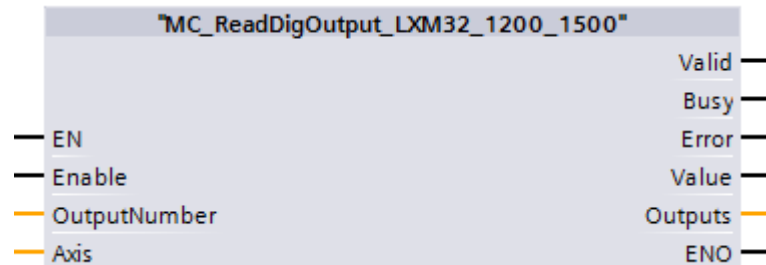
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
	InputNumber	INT	Number of the input that is to be read: 0: I0 1: I1 2: I2 3: I3 4: I4 5: I5 Value range: 0..5, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	Value	BOOL	TRUE: The read input (InputNumber) has a 24V signal level. FALSE: The read input (InputNumber) has a 0V signal level.
	Inputs	WORD	Overall input status (regardless of InputNumber): I0 = Bit 0, I1 = Bit 1, I2 = Bit 2, I3 = Bit 3, I4 = Bit 4, I5 = Bit 5 Value range: 00h..3Fh, initial value: 00h.

7.16.2 MC_ReadDigOutput_LXM32_1200_1500

Function description

Reading the drive's current output status.

Graphical representation



Parameter description

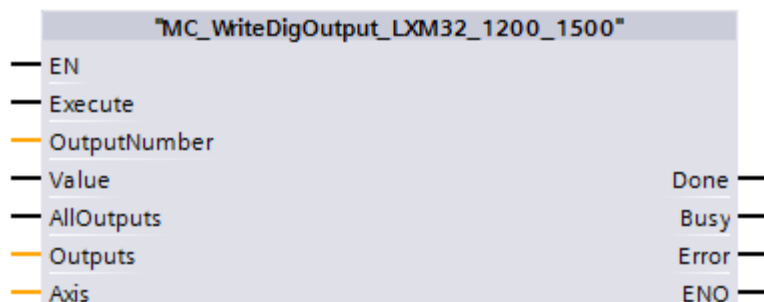
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
	OutputNumber	INT	Number of the output that is to be read: 0: Output 0 1: Output 1 2: Output 2 Value range: 0..2, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	Value	BOOL	TRUE: The read output (OutputNumber) has a 24V signal level. FALSE: The read output (OutputNumber) has a 0V signal level.
	Outputs	WORD	Overall output status (regardless of OutputNumber): Output 0 = Bit 0, Output 1 = Bit 1 ... Value range: 00h..03Fh, initial value: 00h.

7.16.3 MC_WriteDigitalOutput_LXM32_1200_1500

Function description

This function block writes a value to the output referenced by *Output*.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description					
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.					
	OutputNumber	INT	Number of the output that is to be read: 0: Output 0 1: Output 1 2: Output 2 Value range: 0..2, initial value: 0.					
	Value	BOOL	FALSE: Output is set to FALSE TRUE: Output is set to TRUE					
	AllOutputs	BOOL	FALSE: output referenced by <i>OutputNumber</i> is written to <i>Value</i> . TRUE: all outputs are written following <i>Outputs</i> value.					
	Outputs	WORD	Value of the outputs to be written when <i>AllOutputs</i> is true. <table><tr><td>Bit0</td><td>= value of DQ1</td></tr><tr><td>Bit1</td><td>= value of DQ2</td></tr><tr><td>Bit2</td><td>= value of DQ3</td></tr></table>	Bit0	= value of DQ1	Bit1	= value of DQ2	Bit2
Bit0	= value of DQ1							
Bit1	= value of DQ2							
Bit2	= value of DQ3							
IN_OUT	Axis	STRUCT	Axis reference structure					
OUT	Done	BOOL	TRUE: block finished without error					
	Busy	BOOL	TRUE: block is active					
	Error	BOOL	TRUE: block finished with error					

7.17 Error handling

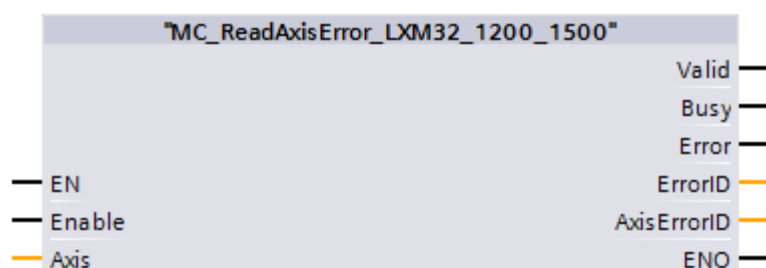
For the purpose of error handling, every block as an output *Error*, which is set if a [synchronous](#) or [asynchronous](#) error occurs. For a more detailed analysis of the error's cause, the block MC_ReadAxisError_LXM32 is called. By means of MC_Reset_LXM32, the error cell is cleared to make it available for future error messages.

7.17.1 MC_ReadAxisError_LXM32_1200_1500

Function description

Reading the last [error message](#) of a drive.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	ErrorID	WORD	0: No error message in the error cell >0: Error number (see list of error numbers). Value range: 0..65535, initial value: 0.
	AxisErrorID	WORD	Bit0...Bit7: indicates the ID of the function block which caused the error (signed value). Bit8...Bit15: indicates the input number (first input = 1, second = 2, ...) which caused the error. Control inputs are not considered (<i>Axis, Input, Execute, Enable,</i>).

Error Diagnostic Example:

Drive is enabled. MC_Jog is called with an input parameter value which will not be accepted by the drive. MC_ReadAxisError will sign the errorID, the function block instance number and the input number of the function block with the wrong value.

MC_Power:

▼ MC_Power_LXM32	*MC_Power_LXM32_1200_1500*		
▼ Input			
Enable	Bool	false	TRUE
▼ Output			
Status	Bool	false	TRUE
Error	Bool	false	FALSE

MC_Jog is called with a parameter value outside of range on the Input no. 3 (VeloSlow)

▼ MC_Jog_LXM32_120...	*MC_Jog_LXM32_1200_1500*		
▼ Input			
Forward	Bool	false	TRUE
Backward	Bool	false	FALSE
Fast	Bool	false	FALSE
TipPos	Dint	DINT#20	20
WaitTime	Int	500	500
VeloSlow	Dint	DINT#60	10000
VeloFast	Dint	DINT#180	180
Acceleration	Dint	DINT#600	600
Deceleration	Dint	DINT#600	600
▼ Output			
Done	Bool	false	FALSE
Busy	Bool	false	FALSE
CommandAbo...	Bool	false	FALSE
Error	Bool	false	TRUE

MC_ReadAxisError signs the following Information:

▼ MC_ReadAxisError_LXM32_1200_1500_Instance	*MC_ReadAxisError_LXM32_1200_1500*		
▼ Input			
Enable	Bool	false	TRUE
▼ Output			
Valid	Bool	false	TRUE
Busy	Bool	false	FALSE
Error	Bool	false	FALSE
ErrorID	Word	16#0	16#A31D
AxisErrorID	Word	16#0	16#0302

Error ID 0xA31D (see drive error list)

Error number	Error class	Description	Cause	Correctives
E A31D	-	Velocity range exceeded (parameter CTRL_v_max, M_n_max)	The velocity was set to a value greater than the maximum permissible velocity in parameter CTRL_v_max or M_n_max, whichever is lower.	If the value of parameter M_n_max is greater than the value of parameter CTRL_v_max, increase the value of parameter CTRL_v_max or reduce the velocity value.

AxisErrorID 0x0302

Error on Input number 3 at the function block with the instance access ID 2 (iInstAccessID).

Each function block instance has an own unique access ID with the symbolic name "iInstAccessID". The number is generated automatically at the first call after PLC startup and depends on the calling sequence of the function blocks in the application.

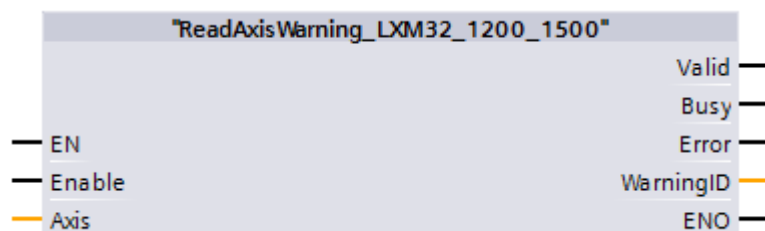
▼ MC_Jog_LXM32_1200_1500_Instance	"MC_Jog_LXM32_1200_1500"		
▼ Input			
Forward	Bool	false	TRUE
Backward	Bool	false	FALSE
Fast	Bool	false	FALSE
TipPos	DInt	DINT#20	20
WaitTime	Int	500	500
VeloSlow	DInt	DINT#60	10000
VeloFast	DInt	DINT#180	180
Acceleration	DInt	DINT#600	600
Deceleration	DInt	DINT#600	600
▼ Output			
Done	Bool	false	FALSE
Busy	Bool	false	FALSE
CommandAborted	Bool	false	FALSE
Error	Bool	false	TRUE
▼ InOut			
Axis	"Axis_Ref_LXM32_1200_1500"		
▼ Static			
bModeToggle	Bool	false	FALSE
bHlpExecute	Bool	false	FALSE
bHlpForward	Bool	false	FALSE
bHlpBackward	Bool	false	FALSE
bHlpFast	Bool	false	FALSE
wHlpCommand	Word	16#0	16#3000
iState	Int	999	100
iHlpState	Int	0	20
iHlpWaitTime	Int	0	500
iHelperErrorPar	Int	0	3
iInstAccessID	Int	0	2

7.17.2 ReadAxisWarning_LXM32_1200_1500

Function description

Returns the last active warning of the drive.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	WarningID	WORD	Last drive active warning ID.

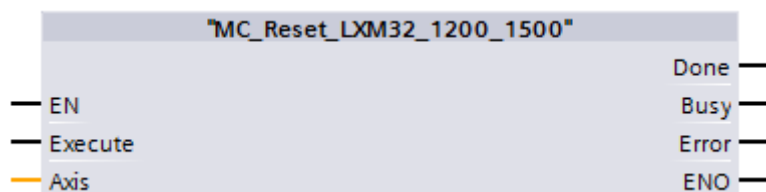
7.17.3 MC_Reset_LXM32_1200_1500

Function description

Error acknowledgement.

The error field is cleared and free for future error messages.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

The error cell is cleared to make it available for future error messages, provided that the cause of the error has been rectified.

If the motor has been stopped by the automatic error response, it will be enabled again, provided that the cause of the error has been rectified when the error message is acknowledged.

Note

Only the first occurred error is entered in the free error cell, in order to permit conclusions to be drawn about the error's cause. As long as the error cell is occupied, previous error numbers are not overwritten (also not if the cause of the error has already been rectified), so that no new errors are entered.

7.18 Extended IO module

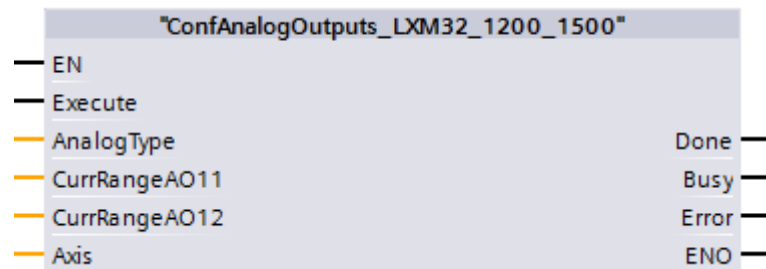
Only available on LXM32M.

7.18.1 ConfAnalogOutputs_LXM32_1200_1500

Function description

Configuration of the analogue outputs.

Graphical representation



Parameter description

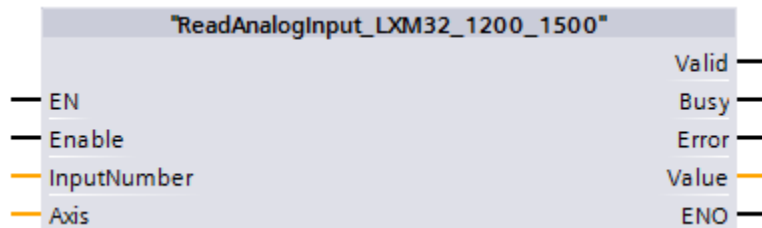
Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	AnalogType	INT	1 / Voltage - Both analog outputs are voltage outputs 2 / Current - Both analog outputs are current outputs Setting can only be changed if power stage is disabled. Value range: 1..2, initial value: 0.
	CurrRangeAO11	INT	IOM1 Range of current of analog output AO11 0: 0-20mA (0 mA correspond to 0 user-defined units) 1: 4-20mA unsigned (4 mA correspond to 0 user-defined units) 2: 4-20mA signed (12 mA correspond to 0 user-defined units) Setting can only be changed if power stage is disabled. Changed settings become active the next time the product is switched on. Value range: 0..2, initial value: 0.
	CurrRangeAO12	INT	IOM1 Range of current of analog output AO12 0: 0-20mA (0 mA correspond to 0 user-defined units) 1: 4-20mA unsigned (4 mA correspond to 0 user-defined units) 2: 4-20mA signed (12 mA correspond to 0 user-defined units) Setting can only be changed if power stage is disabled. Changed settings become active the next time the product is switched on. Value range: 0..2, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

7.18.2 ReadAnalogInput_LXM32_1200_1500

Function description

Reading analog input value of analog input 1 or 2.

Graphical representation



Parameter description

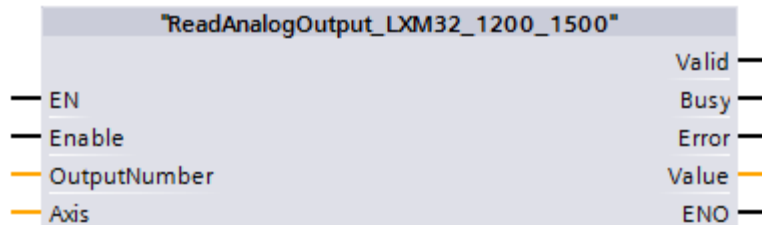
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
	InputNumber	INT	1: Analog input AI11 2: Analog input AI12 Value range: 1..2, initial value: 1.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	Value	INT	Analog value IOM1 Value of input voltage of AI11 / AI12 Value Range -10000...+10000 mV

7.18.3 ReadAnalogOutput_LXM32_1200_1500

Function description

Reading analog output value of analog output 1 or 2.

Graphical representation



Parameter description

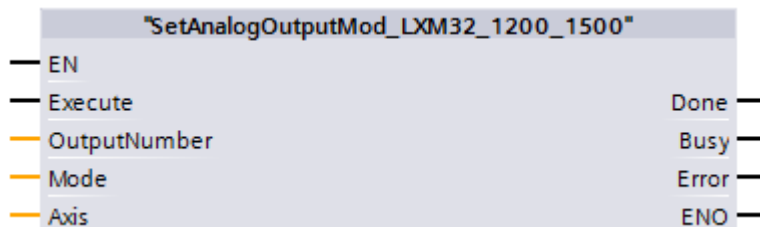
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
	OutputNumber	INT	1: Analog output AO11 2: Analog output AO12 Value range: 1..2, initial value: 1.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	Value	INT	Analog value Unit depends on setting in parameter IOM1_AQ_mode. If setting is 'Voltage': Unit: mV If setting is 'Current': Unit: µA Value range -10000..+20000

7.18.4 SetAnalogOutputMod_LXM32_1200_1500

Function description

Set mode of analog outputs.

Graphical representation



Parameter description

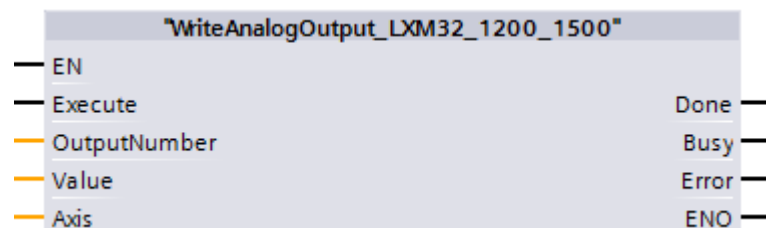
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
	OutputNumber	INT	1: Analog output AO11 2: Analog output AO12 Value range: 1..2, initial value: 1.
	Mode	INT	IOM1 Function of AQ11/AQ12 0 / None : No function 1 / Actual Velocity Actual velocity (10 V / 20 mA correspond to value in CTRL_v_max) 2 / Actual Torque Actual torque (10 V / 20 mA correspond to value in CTRL_I_max) 3 / Reference Velocity Reference velocity (10 V / 20 mA correspond to value in CTRL_v_max) 4 / Reference Torque Reference torque (10 V / 20 mA correspond to value in CTRL_I_max) 5 / Position Deviation Position deviation (10 V / 20 mA correspond to value in MON_p_dif_load_usr) 6 / Fixed Value Fixed value (setting in parameter IOM1_AQ11_FixVal) 7 / Actual Position Actual position in the modulo range (10 V / 20 mA correspond to value in MOD_Max)
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

7.18.5 WriteAnalogOutput_LXM32_1200_1500

Function description

Set analog value of analog output.
Analog output mode must be set to 6 (Fixed Value) with function block "SetAnalogOutputMod_LXM32_1200_1500".

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	OutputNumber	INT	1: Analog input AO11 2: Analog input AO12 Value range: 1..2, initial value: 1.
	Value	INT	IOM1 Fixed value for AQ11/AQ12 Only available if parameter IOM1_AQ1x_func is set to 'Fixed Value'. Unit and range depend on setting in parameter IOM1_AQ_mode. If setting is 'Voltage': Unit: mV Range: -10000 ... 10000 If setting is 'Current': Unit: µA Range: 0 ... 20000
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

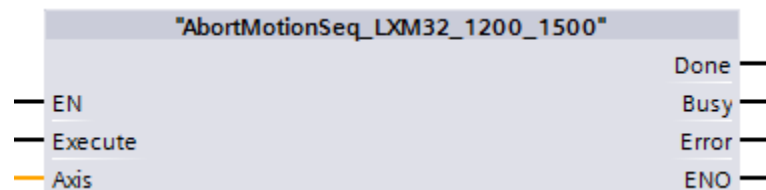
7.19 Motion Sequence Mode

7.19.1 AbortMotionSeq_LXM32_1200_1500

Function description

Abort an active motion sequence.

Graphical representation



Parameter description

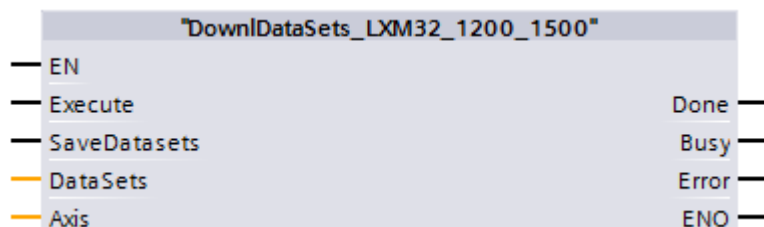
Par. Type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

7.19.2 DownlDataSets_LXM32_1200_1500

Function description

Download all motion sequence data sets.

Graphical representation



Data block Example

DB_MSM_DataSets_Axis_1				
	Name	Data type	Offset	Start value
1	Static			
2	Data	Array[0..3600] of Word	0.0	

Parameter description

Par-type	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	SaveDataSets	BOOL	TRUE: Store to EEPROM after download is finished
	DataSets	DB	DB no. size of 3600 word
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

Note:

It is recommended to make an upload the DataSets with the function block UpIDataSets_LXM032_1200 / UpIDataSets_LXM032_1500 before downloading the data sets within this function block.

- The drive must be in disabled state otherwise an error will occur.

- If required, store the data sets into the drive non-volatile memory (EEPROM) with the Input "SaveDataSets" = TRUE. If not, the parameters will be lost if the drive is switched off.

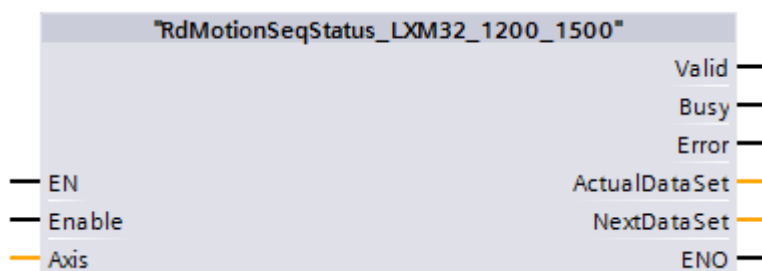
- In case the data are not saved non-volatile, at exchanging a device the controller must remain under voltage, in order for the data be reloaded after the exchange.

7.19.3 RdMotionSeqStatus_LXM32_1200_1500

Function description

Read motion sequence status

Graphical representation



Parameter description

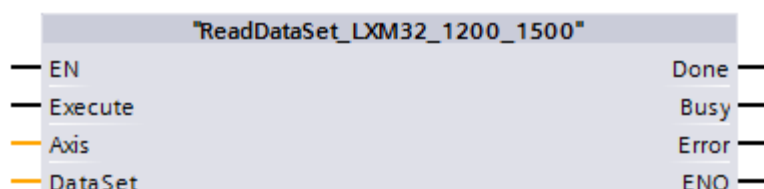
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Execution of the function block is terminated. The outputs Valid, Busy, CommandAborted and Error are set to FALSE. TRUE: The function block is executed repeatedly.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Valid	BOOL	FALSE: Execution has not (yet) been terminated without an error. The values at the outputs are not (yet) valid. TRUE: Execution has been completed without an error. The values at the outputs are valid and can be further processed.
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error
	ActualDataSet	INT	Active data set number
	NextDataSet	INT	Subsequent data set number

7.19.4 ReadDataSet_LXM32_1200_1500

Function description

Reading a single data set from LXM32.

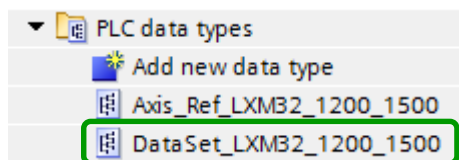
Graphical representation



Parameter description

Par-type	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
IN_OUT	Axis	STRUCT	Axis reference structure
	DataSet	STRUCT	PLC data type "DataSet_LXM32_1200_1500"
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

PLC data type "DataSet_LXM32_1200_1500"



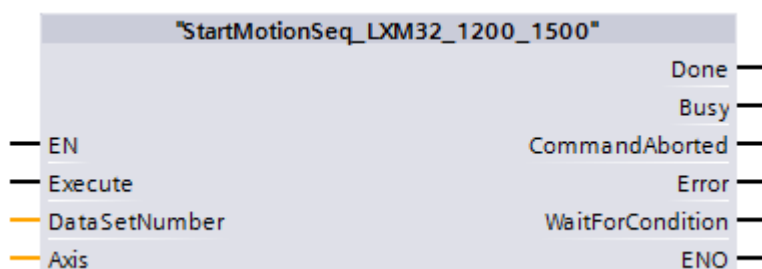
DataSet[0]		"DataSet_LXM32_1200_1500"
DataSetNumber	Int	
DataSetType	Int	
Value1	DInt	
Value2	DInt	
Value3	DInt	
Value4	DInt	
Transition	Int	
NextCondition	Int	
WaitTime	Int	
NextDataSet	Int	

7.19.5 StartMotionSeq_LXM32_1200_1500

Function description

Start motion sequence mode.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	DataSetNumber	INT	The input specifies the number of the data set to be used for starting a sequence
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	CommandAborted	BOOL	TRUE: Motion Sequence mode aborted
	Error	BOOL	TRUE: block finished with error
	WaitForCondition	BOOL	The output indicates that the data set has been completed and that the sequence is waiting for the transition condition to be met. Once the transition condition is met, the subsequent data set started.

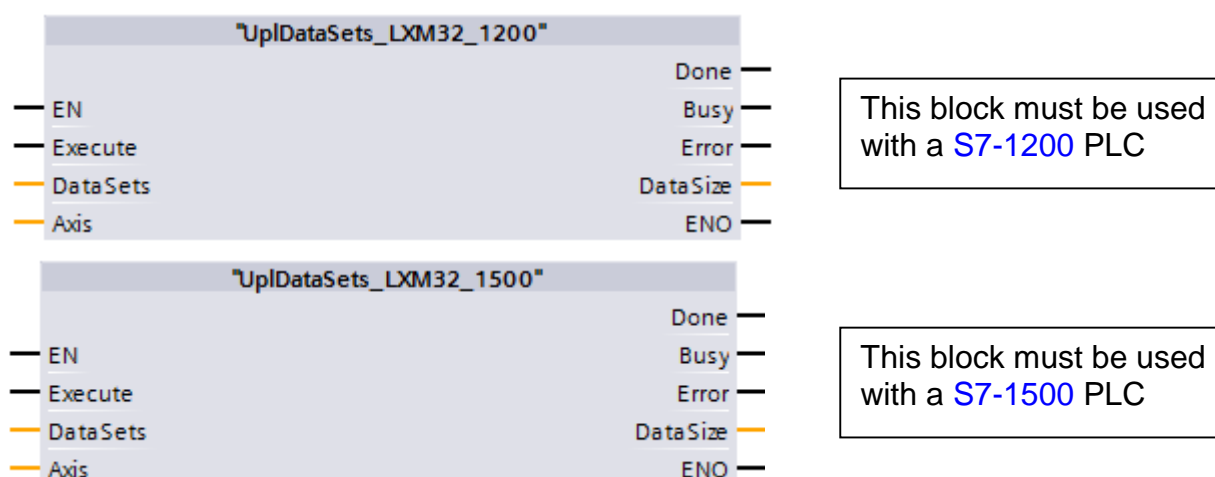
7.19.6 UpIDataSets_LXM32_1200/1500

Function description

Uploading all motion sequence mode data sets.

Stores all the motion sequence mode data sets in a DB. The DB requires 3600 words (Firmware PR912.00 V1.22.00). The size depends on the LXM32 Firmware Revision. It is possible that the size will increase with a newer revision because of enhancements.

Graphical representation



Data block Example

DB_MSM_DataSets_Axis_1				
	Name	Data type	Offset	Start value
1	Static			
2	Data	Array[0..3600] of Word	0.0	

Parameter description

Par-type	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	SaveDataSets	BOOL	TRUE: Store to EEprom after download is finished
	DataSets	DB	DB no. size of 3600 word (Example DB 3)
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

NOTE:

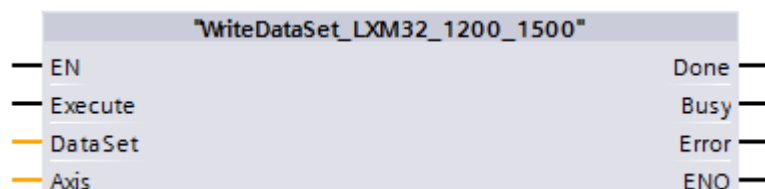
With the two blocks UpIDataSets_LXM32 and DownIDataSets_LXM032, a defective device can be exchanged without a special tool to parameterize the device.

7.19.7 WriteDataSet_LXM32_1200_1500

Function description

Write a single data set to the LXM32.

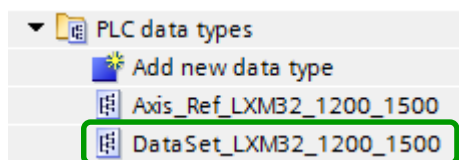
Graphical representation



Parameter description

Par-type	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
IN_OUT	Axis	STRUCT	Axis reference structure
	DataSet	STRUCT	PLC data type "DataSet_LXM32_1200_1500"
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

PLC data type "DataSet_LXM32_1200_1500"



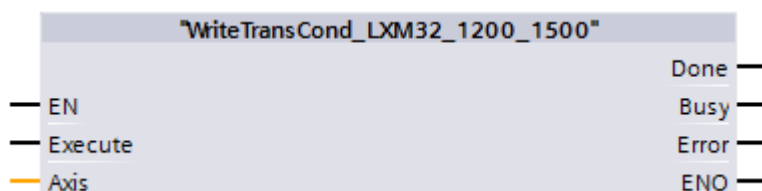
DataSet[0]		"DataSet_LXM32_1200_1500"
DataSetNumber	Int	
DataSetType	Int	
Value1	DInt	
Value2	DInt	
Value3	DInt	
Value4	DInt	
Transition	Int	
NextCondition	Int	
WaitTime	Int	
NextDataSet	Int	

7.19.8 WriteTransCond_LXM32_1200_1500

Function description

If the function block “StartMotionSeq_LXM32_1200_1500” signs the output “WaitForCondition” the sequence can be continued with this function block.

Graphical representation



Parameter description

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
IN_OUT	Axis	STRUCT	Axis reference structure
OUT	Done	BOOL	TRUE: block finished without error
	Busy	BOOL	TRUE: block is active
	Error	BOOL	TRUE: block finished with error

8 Glossary

Asynchronous errors

Asynchronous errors occur independently of the programme sequence, such as an activated limit switch or motor overtemperature, for example.

[Errorhandling](#)

Movement profile generator

From the parameters for acceleration, deceleration, set speed, set and actual position, the movement profile generator calculates a position/timing diagram that indicates the motor's setpoint position at any time of the movement. This profile is processed by the drive control during the movement.

Error class

The device response depends on the severity of an error:

Class	Response	Description
0	Warning	Only a warning, movement is not interrupted.
1	Quick Stop	Motor stops, power amplifier and control remain active.
2	Switch-off	Motor standstill, power amplifier is switched off when motor is at standstill.
3	Fatal error	Power amplifier is switched off immediately
4	Uncontrolled operation	Power amplifier is switched off immediately, device must be switched off.

For the following errors, the error class (i.e. the response to the error), is configurable:

“Phase fault in mains supply” (error number 16#3100 = 12544)

Value range: 1..3, initial value 2 (see manual, parameter SPV_Flt_AC, index 16#3005, sub-index 16#A).

“Position tracking error” (error number 16#A320 = 41760)

Value range: 1..3, initial value 3 (see manual, parameter SPV_Flt_pDiff, index 16#3005, sub-index 16#B).

Error cell

The error cell contains the error code and the error class of an error that has occurred. A newly occurred error will be entered, provided that the error cell is free (i.e. equal to zero). If the error cell is occupied (i.e. not equal to zero), the previous error message will not be overwritten – instead, the new error message is ignored. The error cell is cleared by executing the block MC_Reset_LXM32, provided that the cause of the error has been rectified.

Device parameter list or Object list

List of all the parameters in the device that can be accessed for reading or writing. The parameters are described in the device documentation.

MC_WriteParameter_LXM032 MC_ReadParameter_LXM32

Inc, Inc/s

Stands for “increments” or “increments per second”.

Referred to the motor, this represents the resolution of the power amplifier, with which the motor can be positioned (without taking any gearing into account).

Resolution of the power amplifier: 131072 increments per revolution

Drive speed results from the number of increments per second [Inc/s].

Scaling

Scaling translates the user-defined units (e.g. cm or angular degrees) into internal device units, and vice-versa. The device saves position values in user-defined units.

The scaling factor creates the relationship between the number of motor rotations and the corresponding necessary user-defined units (usr).

Number of motor revolutions = scaling factor x change of user position

During first commissioning, the scaling factor is adjusted so that one motor revolution (called 'U' in the following) corresponds to 16384 user-defined units (called 'usr' in the following): 1U = 16384 usr.

Also see the device manual.

Synchronous errors

Synchronous errors occur during writing of parameters or starting of functions, and are related to an action, for example writing an impermissible parameter value or starting a movement with disabled motor current.

[Errorhandling](#)

usr

stands for "user-defined unit".

Scaling translates the user-defined units (e.g. cm or angular degrees) into internal device units, and vice-versa. The device saves position values in user-defined units.

9 List of error numbers

The complete list of drive error numbers you can find in the LXM32M product manual.

ErrorID hex	ErrorID dec	Error class	Drive error messages
16#1100	4352	0	Parameter out of permissible range
16#1101	4353	0	Parameter does not exist (index)
16#1102	4354	0	Parameter does not exist (sub-index)
16#1103	4355	0	Parameter write not permissible (READ only)
16#1104	4356	0	Write access denied (no access authorisation)
16#1106	4358	0	Command not allowed when drive is active
16#1107	4359	0	Access via other interface blocked
16#1108	4360	0	Parameter not readable (Block Upload)
16#1109	4360	0	Power fail data invalid
16#110A	4362	0	Boot loader not present
16#110B	4363	3	Initialisation error
16#1300	4864	3	Safe Standstill triggered (SAFE_DISABLE_A, SAFE_DISABLE_B)
16#1301	4865	4	SAFE_DISABLE_A and SAFE_DISABLE_B different level
16#1310	4880	3	Reference signal frequency too high
16#1603	5635	0	Capture memory occupied by other function
16#1606	5638	0	Recording still active
16#1607	5639	0	Trigger parameter for capture not defined
16#1608	5640	0	Trigger option for trigger parameter not permitted
16#1609	5641	0	No capture channel defined
16#160A	5642	0	No recorded data present
16#160B	5643	0	Parameter not recordable
16#160C	5644	1	Autotuning: Moment of inertia outside permissible range
16#160E	5646	1	Autotuning: Test movement could not be started
16#160F	5647	1	Autotuning: Power amplifier cannot be enabled
16#1610	6548	1	Autotuning: Processing discontinued
16#1611	5649	1	System error: Autotuning internal write access
16#1613	5651	1	Autotuning: Max. permissible positioning range exceeded
16#1614	5652	0	Autotuning: already active
16#1617	5655	1	Autotuning: Friction or load torque too high
16#1618	5656	1	Autotuning: Optimisation aborted
16#1A01	6657	3	Motor has been changed
16#1B02	6914	3	User parameter for max. current, holding current or Quick Stop current too high
16#2300	8960	3	Power amplifier overcurrent
16#2301	8961	3	Overcurrent in ballast resistor
16#3100	12544	par.	Phase error in mains supply
16#3200	12800	3	DC busovervoltage
16#3201	12801	3	DC busundervoltage (switch-off threshold)
16#3202	12802	2	DC busundervoltage (Quick Stop threshold)
16#3203	12803	4	Motor encoder supply voltage
16#3206	12806	0	DC busundervoltage (warning)
16#4100	16640	3	Power amplifier overtemperature
16#4101	16641	0	Warning power amplifier overtemperature
16#4102	16642	0	Power amplifier overload (I ² t) warning
16#4200	16896	3	Device overtemperature
16#4300	17152	3	Motor overtemperature
16#4301	17153	0	Warning motor overtemperature

16#4302	17154	0	Motor overload (I ² t) warning
16#4402	17410	0	Warning of overload (I ² t) in ballast resistor
16#5200	20992	3	No connection to the motor encoder
16#5201	20993	4	Errors in motor encoder communication
16#5202	20994	4	Motor encoder is not supported
16#5203	20995	4	No connection to the motor encoder
16#5204	20996	3	Connection to motor encoder lost
16#5600	22016	3	Motor connection phase fault
16#5601	22017	4	Interruption or faulty motor encoder signals
16#5602	22018	4	Interruption or faulty motor encoder signals
16#5603	22019	4	Commutation error
16#6107	24839	0	Parameters outside value range (calculation error)
16#6108	24840	0	Function not available
16#610D	24845	0	Error in selection parameter
16#610F	24847	4	Internal timebase fault (timer 0)
16#7120	28960	4	Invalid motor data
16#7121	28961	2	System error: Errors in motor encoder communication
16#7123	28963	4	Motor current offset outside permissible range
16#7124	28964	4	System error: Motor encoder faulty
16#7329	29481	0	Motor encoder sends: Warning
16#7335	29487	0	Communication to motor encoder occupied
16#7400	29696	0	System error: Illegal interrupt (XINT2)
16#7500	29952	0	RS 485/Modbus: Overrun error
16#7501	29953	0	RS 485/Modbus: Framing error
16#7502	29954	0	RS 485/Modbus: Parity error
16#7503	29955	0	RS 485/Modbus: receive error
16#A060	41056	2	Calculation error with electronic gearbox
16#A061	41057	2	Change in reference value with electronic gearbox too large
16#A300	41728	0	Torque ramp with HALT current active
16#A301	41729	0	Drive in status 'QuickStopActive'
16#A302	41730	1	Interruption via LIMP
16#A303	41731	1	Interruption via LIMN
16#A304	41732	1	Interruption via REF
16#A306	41734	1	Interruption by user-initiated software stop
16#A307	41735	0	Interruption by internal software stop
16#A308	41736	0	Drive in 'FAULT' status
16#A309	41737	0	Drive not in 'OperationEnable' status
16#A310	41744	0	Power amplifier not active
16#A312	41746	0	Profile generating interrupted
16#A313	41747	0	Position overrun present (pos_over=1), therefore reference point no longer defined (ref_ok=0)
16#A314	41748	0	No reference position
16#A315	41749	0	Homing active
16#A316	41750	0	Overflow on acceleration calculation
16#A317	41751	0	Drive not at standstill
16#A318	41752	0	Operating mode active (x_end = 0)
16#A319	41753	1	Manual/Autotuning: Distance range overflow
16#A31A	41754	0	Manual/Autotuning: Amplitude/ offset set too high
16#A31B	41755	0	HALT requested
16#A31C	41756	0	Illegal position setting with software limit switch
16#A31D	41757	0	Speed range overflow (CTRL_n_max)
16#A31E	41758	1	Interruption via positive software limit switch
16#A31F	41759	1	Interruption via negativesoftware limit switch

16#A320	41760	par.	Position tracking error
16#A321	41761	0	RS 422 position interface not defined as input
16#A324	41764	1	Error when homing (additional info = detailed error number)
16#A325	41765	1	Approach limit switch not enabled
16#A326	41766	1	REF switch not found between LIMP and LIMN
16#A327	41767	1	Reference movement to REF without direction reversal, impermissible limit switch LIM activated
16#A328	41768	1	Reference movement to REF without direction reversal, overrun of LIM or REF not permissible
16#A329	41769	1	More than one signal LIMP, LIMN, REF active
16#A32A	41770	1	Ext. monitoring signal LIMP with counterclockwise rotation
16#A32B	41771	1	Ext. monitoring signal LIMN with clockwise rotation
16#A32C	41772	1	Reference movement error at REF (e.g. by impact)
16#A32D	41773	1	Reference movement error at LIMP (e.g. by impact)
16#A32E	41774	1	Reference movement error at LIMN (e.g. by impact)
16#A32F	41775	1	Index pulse not found
16#A330	41776	0	Reproducibility of the index pulse movement uncertain, index pulse motion too close to the switch
16#A331	41777	3	No run-up operating mode with local control selected
16#A332	41778	1	Error with jog (additional info = detailed error number)
16#A334	41780	2	Timeout at Standstill window monitor
16#A335	41781	1	Processing only possible in fieldbus mode
16#B100	45312	0	RS 485/Modbus: Unknown service
16#B200	45568	0	RS 485/Modbus: Protocol error
16#B201	45569	2	RS 485/Modbus: Nodeguarderror
16#B202	45570	0	RS 485/Modbus: Nodeguardwarning
16#B203	45571	0	RS 485/Modbus: Number of monitor objects incorrect
16#B204	45572	0	RS 485/Modbus: service too long
16#B300	45824	4	Profibus: Initialising failed
16#B301	45825	4	Profibus: Initialising failed
16#B302	45826	0	Profibus: Write access denied (incorrect job identification)
16#B303	45827	par.	Profibus: Faulty processing of process data channel
16#B304	45828	par.	Profibus: Faulty processing of process data channel
16#B305	45829	par.	Profibus: Parameter cannot be mapped to the output data frame
16#B306	45830	par.	Profibus: Faulty processing of process data channel
16#B307	45831	par.	Profibus: Faulty processing of process data channel
16#B308	45832	par.	Profibus: Parameter cannot be read
16#B309	45833	0	Profibus: Sub-index not equal to zero
16#B30A	45834	0	Profibus: Parameter does not exist
16#B30B	45835	1	Profibus: Watchdog
16#B30C	45836	1	Profibus: Motor stop via clear command of master
16#B30D	45837	0	Profibus: Parameter cannot be mapped

Library error messages			
16#FF00	65280		Wrong_AxisRef
16#FF01	65281		Initialization_Failed
16#FF02	65282		Wrong_Data_Length
16#FF03	65283		Timeout
16#FF04	65284		Axis_Busy
16#FF05	65285		Invalid_Parameter_Number
16#FF06	65286		Unknown_State
16#FF07	65287		Capture_Busy
16#FF08	65288		Trigger_Event_Lost
16#FF09	65289		Axis_Not_In_Standstill
16#FF0A	65290		Unknown_Device_Type
16#FF0B	65291		Wrong_Data_Struct
16#FF0C	65292		Motion Sequence mode not active
16#FF0D	65293		Size of parameter data block too small
16#FF0E	65294		Data block write protected