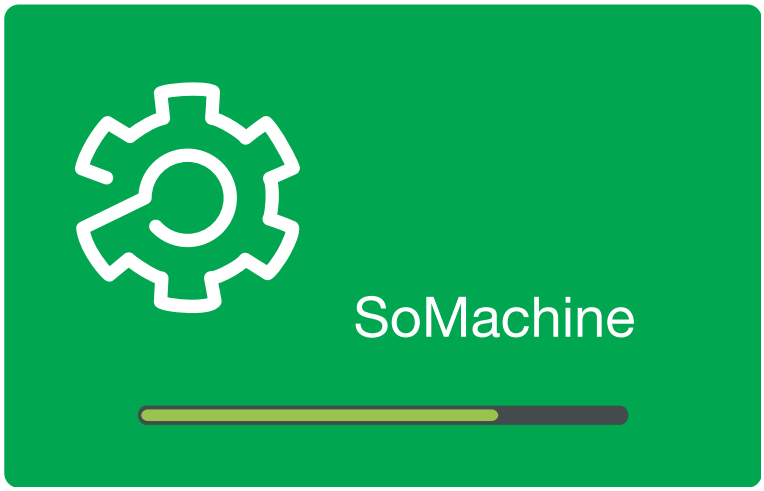
Lexium Library

Function blocks

Software manual

V2.09, 04.2012



0198441113892, V2.09, 04.2012

www.schneider-electric.com

Schneider Electric

Important information

This manual is part of the product.

Carefully read this manual and observe all instructions.

Keep this manual for future reference.

Hand this manual and all other pertinent product documentation over to all users of the product.

Carefully read and observe all safety instructions and the chapter "Before you begin - safety information".

Some products are not available in all countries.
For information on the availability of products, please consult the catalog.

Subject to technical modifications without notice.

All details provided are technical data which do not constitute warranted qualities.

Most of the product designations are registered trademarks of their respective owners, even if this is not explicitly indicated.

0198441113892, V2.09, 04.2012

Table of contents

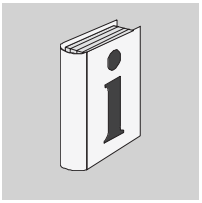
0198441113892, V2.09, 04.2012

0198441113892, V2.09, 04.2012

0198441113892, V2.09, 04.2012

0198441113892, V2.09, 04.2012

About this manual



This manual is an extract of the SoMachine Online Help. Fully read and understand all manuals of the SoMachine Online Help and of the products used.

Purpose of this document

This document describes the functions of the Lexium Library.

Software environment	Devices	Fieldbus
Device Descriptions of version 3.2 and higher are supported.	LXM05 SD328 LXM32	CANopen

Validity note

This document is valid for SoMachine as of Version 2.0.

Source manuals The latest versions of the manuals can be downloaded from the Internet at:

http://www.schneider-electric.com

Corrections and suggestions We always try to further optimize our manuals. We welcome your suggestions and corrections.

Please get in touch with us by e-mail:
techcomm@schneider-electric.com.

Work steps If work steps must be performed consecutively, this sequence of steps is represented as follows:

- ■ Special prerequisites for the following work steps
- ▶ Step 1
- ◁ Specific response to this work step
- ▶ Step 2

If a response to a work step is indicated, this allows you to verify that the work step has been performed correctly.

Unless otherwise stated, the individual steps must be performed in the specified sequence.

SI units SI units are the original values. Converted units are shown in brackets behind the original value; they may be rounded.

Example:
Minimum conductor cross section: 1.5 mm^2 (AWG 14)

Glossary Explanations of special technical terms and abbreviations.

Index List of keywords with references to the corresponding page numbers.

0198441113892, V2.09, 04.2012

Disclaimer

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products described here. 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 user or integrator to perform the appropriate and fully comprehensive risk analyses, evaluation and testing of the products with respect to the relevant specific application or use of the products. Neither Schneider Electric nor any of its affiliate 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.

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.

0198441113892, V2.09, 04.2012

1 Before you begin - safety information

1

1.1 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.

1.2 Intended use

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

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.

0198441113892, V2.09, 04.2012

1.3 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.

CAUTION

CAUTION used without the safety alert symbol, is used to address practices not related to personal injury (e.g. **can result** in equipment damage).

0198441113892, V2.09, 04.2012

1.4 Basic information

⚠ WARNING

LOSS OF CONTROL

- 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.

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

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”.

⚠ WARNING

UNINTENDED BEHAVIOR DUE TO IMPROPER ERROR HANDLING

Improper error handling can change movements or signals or deactivate monitoring functions.

- Carefully program the error handling routines.
- Verify the effectiveness of error handling.

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

⚠ WARNING

UNINTENDED BEHAVIOR DUE TO CHANGES TO THE LIBRARY

- Do not change or manipulate the library in any way whatsoever.

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

0198441113892, V2.09, 04.2012

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

Also see the glossary at the end of this manual.

0198441113892, V2.09, 04.2012

2 Lexium Library Guide

2

Library name

Lexium Library (LXM)

Software environment	Devices	Fieldbus
Device Descriptions of version 3.2 and higher are supported.	LXM05 SD328 LXM32	CANopen

The function blocks described here are used to control LXM32, LXM05 and SD328 drives in CANopen fieldbuses under the SoMachine software environment. The function blocks are compliant with the IEC 61131-3 standard.

Naming conventions

- Function blocks with the prefix MC_ ("Motion Control") are compliant with the PLCopen specifications. They conform to a global standard for programming motion control applications.
- Function blocks without a prefix are vendor-specific (Schneider Electric); however, they comply with the general PLC open rules.

Simple application

- The function blocks are used in the same way.
- The function blocks comply with the PLCopen state diagram.
- The function blocks feature a visualization that can be easily integrated into the application.

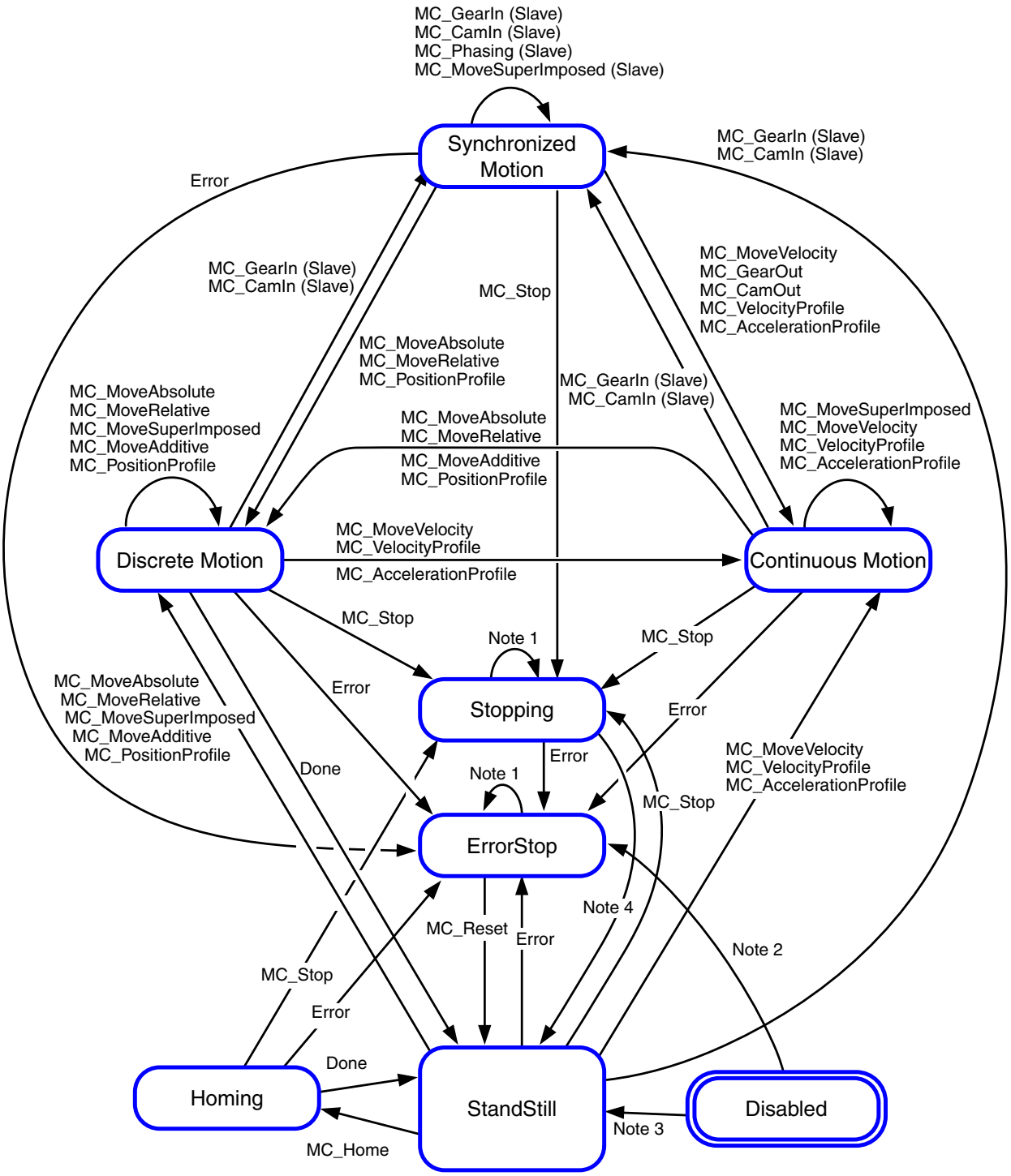
Categorization of the function blocks

- Single axis: These function blocks are used for movements or functions of a single, independent axis.
- Multi axis: These function blocks are used for coordinated movements of several axes (for example, Electronic Gear).
- Motion Sequence: These function blocks are used for movements whose target values are contained in parameterizable data sets.
- Administrative: These function blocks are used for configuration tasks (such as reading and writing of parameters, restoring a device configuration, etc.).
- Device Function: These function blocks support you in commissioning a drive at a controller. Before these function blocks can be used, you must correctly set the communication parameters baud rate and node address.

0198441113892, V2.09, 04.2012

2.1 PLCopen state diagram

At any given point in time, the drive is exactly in one state. If a function block is executed or an error occurs, this may cause a state transition. The function block "2.10.1.4 MC_ReadStatus_LXM" is used to read the current status of the drive.



Note 1: In the states ErrorStop or Stopping, all function blocks can be called, but none of them will be executed, except for

0198441113892, V2.09, 04.2012

"2.10.5.3 MC_Reset_LXM" and Error. Calling the function block "2.10.5.3 MC_Reset_LXM" will cause a transition to the states Stand-Still or ErrorStop, respectively.

Note 2: Power.Enable = TRUE and there is an error

Note 3: Power.Enable = TRUE and there is no error

Note 4: "2.7.9.1 MC_Stop_LXM". Done AND NOT "2.7.9.1 MC_Stop_LXM" Execute.

2.2 List of the function blocks

Category	Subcategory	Function block	Type	LXM05	SD328	LXM32
Single axis						
	Initialization	"2.7.1.1 MC_Power_LXM"	PLCopen	X	X	X
	Operating mode Jog	"2.7.2.1 MC_Jog_LXM"	PLCopen	X	X	X
	Operating mode Current Control	"2.7.3.1 Current-Control_LXM"	Vendor-specific	X	-	-
	Operating mode Profile Torque	"2.7.4.1 MC_Tor-queControl_LXM"	PLCopen	-	-	X
	Operating mode Speed Control / Oscillator	"2.7.5.1 Velocity-Control_LXM"	Vendor-specific	X	X	-
	Operating mode Profile Velocity	"2.7.6.1 MC_MoveVeloc-ity_LXM"	PLCopen	X	X	X
	Operating mode Profile Position	"2.7.7.1 MC_MoveAbso-lute_LXM"	PLCopen	X	X	X
		"2.7.7.2 MC_MoveAddi-tive_LXM"	PLCopen	X	X	X
		"2.7.7.3 MC_Mov-eRelative_LXM"	PLCopen	X	X	X
	Operating mode Homing	"2.7.8.1 MC_Home_LXM"	PLCopen	X	X	X
		"2.7.8.2 MC_Set-Position_LXM"	PLCopen	X	X	X
	Stopping	"2.7.9.1 MC_Stop_LXM"	PLCopen	X	X	X
		"2.7.9.2 MC_Halt_LXM"	PLCopen	-	-	X
	Position capture via signal input	"2.7.10.1 MC_TouchP-robe_LXM"	PLCopen	X	X	X
		"2.7.10.2 MC_AbortTrig-ger_LXM"	PLCopen	X	X	X

0198441113892, V2.09, 04.2012

Category	Subcategory	Function block	Type	LXM05	SD328	LXM32
Multi axis						
	Operating mode Electronic Gear	"2.8.1.1 GearIn-Sync_LXM"	Vendor-specific	X	X	X
		"2.8.1.2 MC_GearIn_LXM"	PLCopen	-	-	X
		"2.8.1.3 MC_Gear-Out_LXM"	PLCopen	X	X	X

Category	Subcategory	Function block	Type	LXM05	SD328	LXM32
Motion Sequence						
	Operating mode Motion Sequence	"2.9.1.1 ReadMo-tionSequenceSta-tus_LXM"	Vendor-specific	-	-	X
		"2.9.1.2 ReadDa-taSet_LXM"	Vendor-specific	-	-	X
		"2.9.1.3 WriteDa-taSet_LXM"	Vendor-specific	-	-	X
		"2.9.1.4 WriteTran-sitionCondi-tion_LXM"	Vendor-specific	-	-	X
		"2.9.1.5 StartMo-tionSe-quence_LXM"	Vendor-specific	-	-	X
		"2.9.1.6 AbortMo-tionSe-quence_LXM"	Vendor-specific	-	-	X

0198441113892, V2.09, 04.2012

Category	Subcategory	Function block	Type	LXM05	SD328	LXM32
Administrative						
	Reading a parameter	"2.10.1.1 MC_ReadActualTorque_LXM"	PLCopen	-	-	X
		"2.10.1.2 MC_ReadActualVelocity_LXM"	PLCopen	X	X	X
		"2.10.1.3 MC_ReadActualPosition_LXM"	PLCopen	X	X	X
		"2.10.1.4 MC_ReadStatus_LXM"	PLCopen	X	X	X
		"2.10.1.5 MC_ReadParameter_LXM"	PLCopen	X	X	X
		"2.10.1.6 GetSupplierVersion"	Vendor-specific	X	X	X
	Writing a parameter	"2.10.2.1 MC_WriteParameter_LXM"	PLCopen	X	X	X
		"2.10.2.2 SetDriveRamp_LXM"	Vendor-specific	X	X	X
		"2.10.2.3 SetStopRamp_LXM"	Vendor-specific	-	-	X
		"2.10.2.4 SetLimitSwitch_LXM"	Vendor-specific	X	X	X
		"2.10.2.5 ResetParameters_LXM"	Vendor-specific	X	X	X
		"2.10.2.6 StoreParameters_LXM"	Vendor-specific	X	X	X
	Saving and restoring device configuration	"2.10.3.1 UploadDriveParameter_LXM"	Vendor-specific	X	X	X
		"2.10.3.2 DownloadDriveParameter_LXM"	Vendor-specific	X	X	X
	Inputs and outputs	"2.10.4.1 ReadAnalogInputs_LXM"	Vendor-specific	X	X	-
		"2.10.4.2 MC_ReadDigitalInput_LXM"	PLCopen	X	X	X
		"2.10.4.3 MC_ReadDigitalOutput_LXM"	PLCopen	X	X	X
		"2.10.4.4 MC_WriteDigitalOutput_LXM"	PLCopen	-	-	X
	Error handling	"2.10.5.1 ReadAxisWarning_LXM"	Vendor-specific	-	-	X
		"2.10.5.2 MC_ReadAxisError_LXM"	PLCopen	X	X	X

0198441113892, V2.09, 04.2012

Category	Subcategory	Function block	Type	LXM05	SD328	LXM32
		"2.10.5.3 MC_Reset_LXM"	PLCopen	X	X	X

Category	Subcategory	Function block	Type	LXM05	SD328	LXM32
Device Function						
	Startup	"2.11.1.1 Servo_Startup"	Vendor-specific	X	-	X
		"2.11.1.2 Stepper_Startup"	Vendor-specific	-	X	-

0198441113892, V2.09, 04.2012

2.3 Compatibility of the function blocks

The table below shows the main differences between the previous library BL_Motion_CAN_Vyyy.lib (for LXM05 and SD328) and the Lexium Library (for LXM05, SD328 and LXM32).

0198441113892, V2.09, 04.2012

Old function block	New function block	LXM05	SD328	LXM32	Remarks
MC_Power_CAN	"2.7.1.1 MC_Power_LXM"	X	X	X	
MC_Jog_CAN	"2.7.2.1 MC_Jog_LXM"	X	X	X	
CurrentControl_CAN	"2.7.3.1 CurrentControl_LXM"	X	X	-	
No equivalent	"2.7.4.1 MC_TorqueControl_LXM"	-	-	X	
VelocityControl_CAN	"2.7.5.1 VelocityControl_LXM"	X	X	-	
MC_MoveVelocity_CAN	"2.7.6.1 MC_MoveVelocity_LXM"	X	X	X	
MC_MoveAbsolute_CAN	"2.7.7.1 MC_MoveAbsolute_LXM"	X	X	X	
MC_MoveAdditive_CAN	"2.7.7.2 MC_MoveAdditive_LXM"	X	X	X	
MC_MoveRelative_CAN	"2.7.7.3 MC_MoveRelative_LXM"	X	X	X	
MC_Home_CAN	"2.7.8.1 MC_Home_LXM"	X	X	X	
MC_SetPosition_CAN	"2.7.8.2 MC_SetPosition_LXM"	X	X	X	
MC_Stop_CAN	"2.7.9.1 MC_Stop_LXM"	X	X	X	
No equivalent	"2.7.9.2 MC_Halt_LXM"	-	-	X	
MC_TouchProbe_CAN	"2.7.10.1 MC_TouchProbe_LXM"	X	X	X	
MC_AbortTrigger_CAN	"2.7.10.2 MC_AbortTrigger_LXM"	X	X	X	
MC_GearIn_CAN	"2.8.1.1 GearInSync_LXM"	X	X	X	This function block replaces the function block MC_GearIn_CAN. The functionality remains the same (position synchronization).
No equivalent	"2.8.1.2 MC_GearIn_LXM"	-	-	X	This new function block complies with the PLCopen specification (velocity synchronization).
MC_GearOut_CAN	"2.8.1.3 MC_GearOut_LXM"	X	X	X	
No equivalent	"2.9.1.1 ReadMotionSequenceStatus_LXM"	-	-	X	
No equivalent	"2.9.1.2 ReadDataSet_LXM"	-	-	X	
No equivalent	"2.9.1.3 WriteDataSet_LXM"	-	-	X	
No equivalent	"2.9.1.4 WriteTransitionCondition_LXM"	-	-	X	
No equivalent	"2.9.1.5 StartMotionSequence_LXM"	-	-	X	
No equivalent	"2.9.1.6 AbortMotionSequence_LXM"	-	-	X	

0198441113892, V2.09, 04.2012

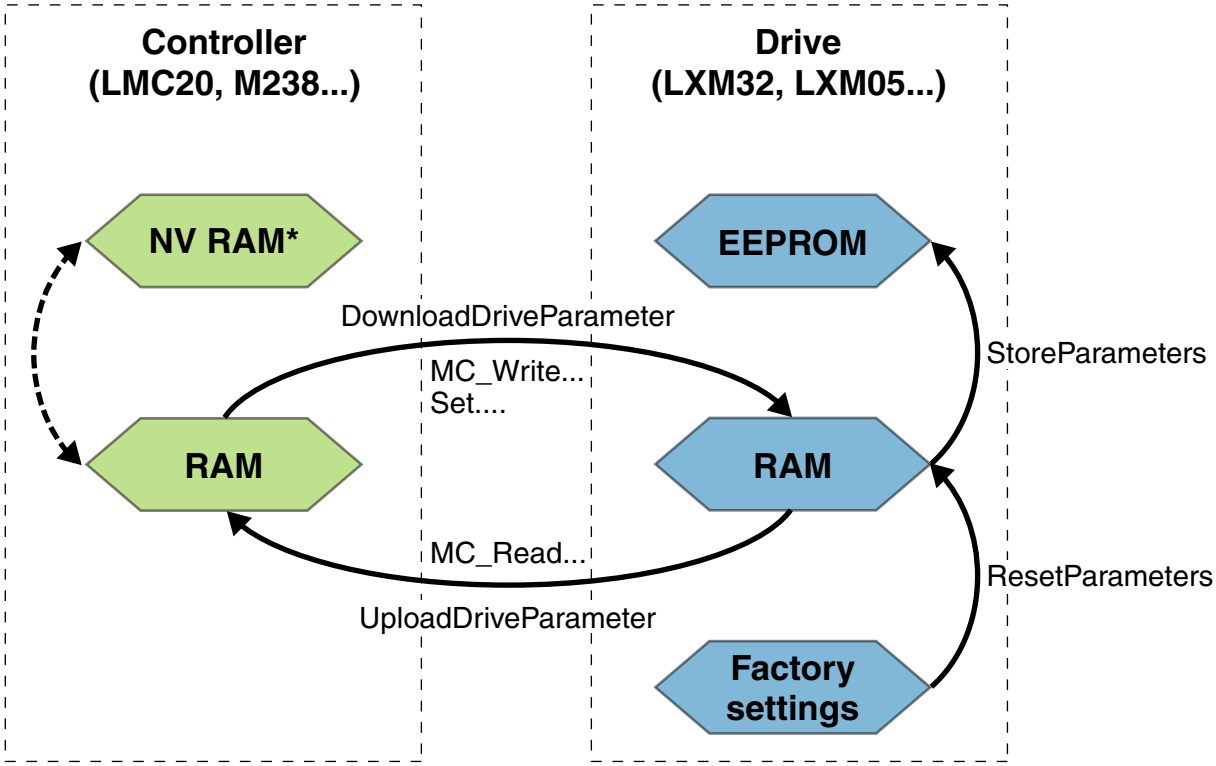
Old function block	New function block	LXM05	SD328	LXM32	Remarks
No equivalent	"2.10.1.1 MC_ReadActualTorque_LXM"	-	-	X	
MC_ReadActualVelocity_CAN ReadRefActualVelocity_CAN ReadActualMasterVelocity_CAN	"2.10.1.2 MC_ReadActualVelocity_LXM"	X	X	X	The new function block allows you to read four different types of velocities. The type is selected via the new input `VelocityType`.
MC_ReadActualPosition_CAN ReadActualPositionInc_CAN ReadRefPosition_CAN ReadRefPositionInc_CAN ReadActualMasterPosition_CAN	"2.10.1.3 MC_ReadActualPosition_LXM"	X	X	X	The new function block allows you to read eight different types of positions. The type is selected via the new input `PositionType`.
MC_ReadStatus_CAN	"2.10.1.4 MC_ReadStatus_LXM"	X	X	X	
MC_ReadParameter_CAN	"2.10.1.5 MC_ReadParameter_LXM"	X	X	X	
GetVersion_CAN	"2.10.1.6 GetSupplierVersion"	X	X	X	
MC_WriteParameter_CAN	"2.10.2.1 MC_WriteParameter_LXM"	X	X	X	
SetDriveRamp_CAN	"2.10.2.2 SetDriveRamp_LXM"	X	X	X	New inputs: `Type`, `Torque`
No equivalent	"2.10.2.3 SetStopRamp_LXM"	-	-	X	
SetLimitSwitch_CAN	"2.10.2.4 SetLimitSwitch_LXM"	X	X	X	
ResetParameters_CAN	"2.10.2.5 ResetParameters_LXM"	X	X	X	
StoreParameters_CAN	"2.10.2.6 StoreParameters_LXM"	X	X	X	
UploadDriveParameter_CAN	"2.10.3.1 UploadDriveParameter_LXM"	X	X	X	The interface (parameter structure) of the function block has changed. Output: The size has been suppressed.
DownloadDriveParameter_CAN	"2.10.3.2 DownloadDriveParameter_LXM"	X	X	X	The interface (parameter structure) of the function block has changed. Outputs: Index and subindex have been suppressed.
ReadAnalogInputs_CAN	"2.10.4.1 ReadAnalogInputs_LXM"	X	X	-	
MC_ReadDigitalInput_CAN	"2.10.4.2 MC_ReadDigitalInput_LXM"	X	X	X	
MC_ReadDigitalOutput_CAN	"2.10.4.3 MC_ReadDigitalOutput_LXM"	X	X	X	

0198441113892, V2.09, 04.2012

Old function block	New function block	LXM05	SD328	LXM32	Remarks
No equivalent	"2.10.4.4 MC_WriteDigitalOutput_LXM"	-	-	X	
No equivalent	"2.10.5.1 ReadAxisWarning_LXM"	-	-	X	
MC_ReadAxisError_CAN	"2.10.5.2 MC_ReadAxisError_LXM"	X	X	X	New output: `AxisErrorID`
MC_Reset_CAN	"2.10.5.3 MC_Reset_LXM"	X	X	X	
No equivalent	"2.11.1.1 Servo_Startup"	X	-	X	
No equivalent	"2.11.1.2 Stepper_Startup"	-	X	-	

0198441113892, V2.09, 04.2012

2.4 Memory structure during parameter transmission



0198441113892, V2.09, 04.2012

2.5 Transitions between function blocks

The table below shows how the execution of a function block (function block 1) can be terminated by another function block (function block 2) in the case of LXM32.

	Function block 2				
Function block 1	**MC_Jog**	**MC_Home**	**MC_MoveAbsolute**	**MC_MoveAdditive**	**MC_MoveRelative**
MC_Jog	On the fly	Motor standstill	Motor standstill 1)	Motor standstill 1)	Motor standstill 1)
MC_Home	Not permitted	Not permitted	Not permitted	Not permitted	Not permitted
MC_MoveAbsolute	Motor standstill	Motor standstill	On the fly	On the fly	On the fly
MC_MoveAdditive	Motor standstill	Motor standstill	On the fly	On the fly	On the fly
MC_MoveRelative	Motor standstill	Motor standstill	On the fly	On the fly	On the fly
MC_MoveVelocity	Motor standstill	Motor standstill	Motor standstill 1)	Motor standstill 1)	Motor standstill 1)
MC_TorqueControl	Motor standstill	Motor standstill	Motor standstill 1)	Motor standstill 1)	Motor standstill 1)
MC_Stop	Not permitted	Not permitted	Not permitted	Not permitted	Not permitted
MC_Halt	Not permitted	Not permitted	Not permitted	Not permitted	Not permitted
GearInSync	Motor standstill	Motor standstill	Motor standstill 1)	Motor standstill 1)	Motor standstill 1)
MC_GearIn	Motor standstill	Motor standstill	Motor standstill 1)	Motor standstill 1)	Motor standstill 1)

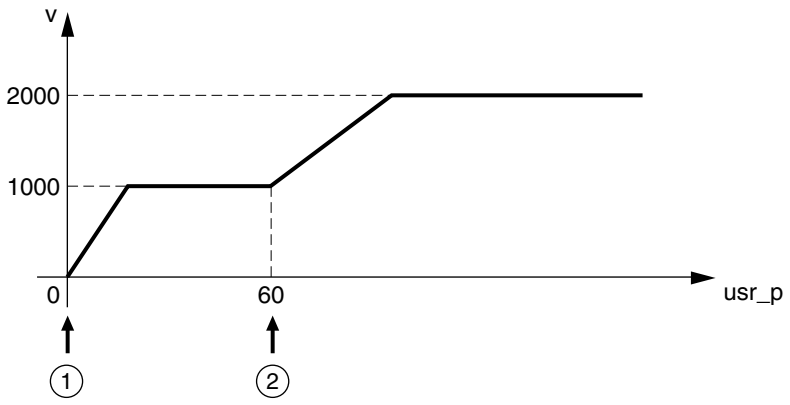
1) Firmware version <V01.04: Only if motor is at a standstill. Firmware version >V01.04: Depends on the setting in parameter `PP_OpmChgType`.

0198441113892, V2.09, 04.2012

	Function block 2					
Function block 1	**MC_MoveVelocity**	**MC_TorqueControl**	**MC_Stop**	**MC_Halt**	**GearInSync**	**MC_GearIn**
MC_Jog	On the fly	On the fly	On the fly	On the fly	Motor standstill	On the fly
MC_Home	Not permitted	Not permitted	On the fly	On the fly	Not permitted	Not permitted
MC_MoveAbsolute	On the fly	On the fly	On the fly	On the fly	Motor standstill	On the fly
MC_MoveAdditive	On the fly	On the fly	On the fly	On the fly	Motor standstill	On the fly
MC_MoveRelative	On the fly	On the fly	On the fly	On the fly	Motor standstill	On the fly
MC_MoveVelocity	On the fly	On the fly	On the fly	On the fly	Motor standstill	On the fly
MC_TorqueControl	On the fly	On the fly	On the fly	On the fly	Motor standstill	On the fly
MC_Stop	Not permitted	Not permitted	Not permitted	Not permitted	Not permitted	Not permitted
MC_Halt	Not permitted	Not permitted	On the fly	Not permitted	Not permitted	Not permitted
GearInSync	On the fly	On the fly	On the fly	On the fly	On the fly	On the fly
MC_GearIn	On the fly	On the fly	On the fly	On the fly	Motor standstill	On the fly

On the fly The execution of function block 2 is started on the fly, i.e. immediately without any delay. The execution of function block 1 is aborted.

Example of On the fly

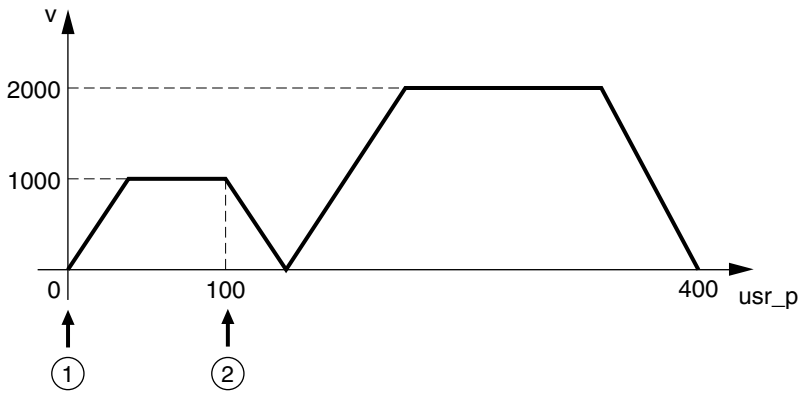


Running function block	MC_MoveAbsolute_LXM Position = 100 usr_p, velocity = 1000 usr_v
New function block	MC_MoveVelocity_LXM Velocity = 2000 usr_v

Motor standstill The execution of function block 2 first decelerates the motor to a standstill with the adjusted deceleration ramp. The movement as per function block 2 starts as soon as the motor has come to a standstill. The execution of function block 1 is aborted.

0198441113892, V2.09, 04.2012

Examples of motor standstill



Running function block	MC_MoveVelocity_LXM Velocity = 1000 usr_v
New function block	MC_MoveAbsolute_LXM Position = 400 usr_p, velocity = 2000 usr_v

Not permitted Function block 1 cannot be aborted by the new function block. Function block 2 will not be executed.

0198441113892, V2.09, 04.2012

2.6 Basic inputs and outputs

Input/output	Data type	Description
Axis	Axis_Ref_LXM	Name of the axis (instance) for which the function block is to be executed. The name must be declared in the PLC configuration. The name of the axis can be found to the left in the tree structure of your software.
Input	Input_Ref_LXM	Input is a special data type for digital and analog inputs. The data type corresponds to the name of the axis (instance) to which the inputs belong (similar to Axis). In the case of function blocks specifically provided for reading analog and digital inputs, Input replaces the input Axis.
Output	Output_Ref_LXM	Output is a special data type for digital and outputs. The data type corresponds to the name of the axis (instance) to which the outputs belong (similar to Axis). In the case of function blocks specifically provided for writing and reading analog and digital inputs, Output replaces the input Axis.

Input	Data type	Description
Enable	BOOL	Value range: TRUE, FALSE Initial value: FALSE The input Enable starts or terminates the execution of a function block. (exception "2.7.1.1 MC_Power_LXM") 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.
Execute	BOOL	Value range: TRUE, FALSE Initial value: FALSE The input Execute starts the execution of a function block in the case of a rising edge (FALSE->TRUE). If a second rising edge is detected during the execution of the function block, the current execution is aborted and the function block is executed again. Execution is terminated as soon as the output Busy is FALSE. FALSE and, at the same time, Busy = FALSE: Either Done, Error or CommandAborted are set to TRUE for one call. TRUE and, at the same time, Busy = FALSE: Either Done, Error or CommandAborted are set to TRUE and remain TRUE until Execute is set to FALSE.

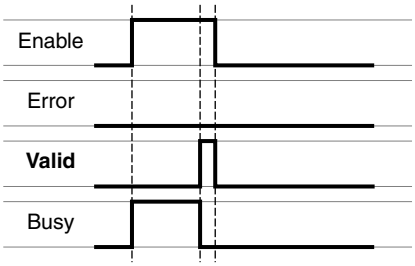
0198441113892, V2.09, 04.2012

Output	Data type	Description
Done	BOOL	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.
Valid	BOOL	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.
Busy	BOOL	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.
CommandAborted	BOOL	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.
Error	BOOL	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.

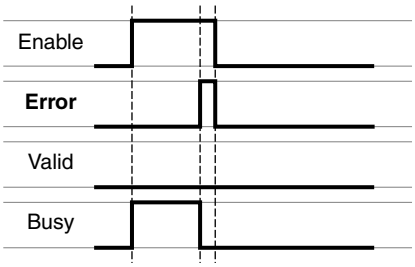
0198441113892, V2.09, 04.2012

2.6.1 Signal behavior of function blocks with the input `Enable`

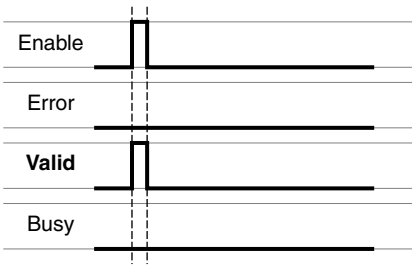
Example 1 Single execution without error (execution requires more than one call).



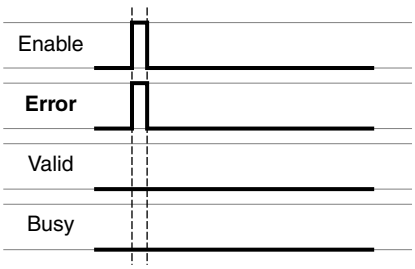
Example 2 Single execution with error (execution requires more than one call).



Example 3 Single execution without error (execution requires only one call).

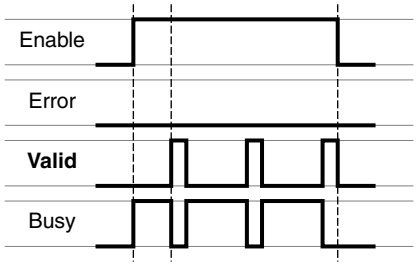


Example 4 Single execution with error (execution requires only one call).

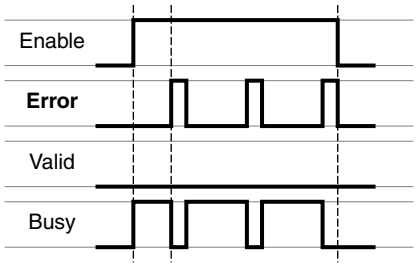


0198441113892, V2.09, 04.2012

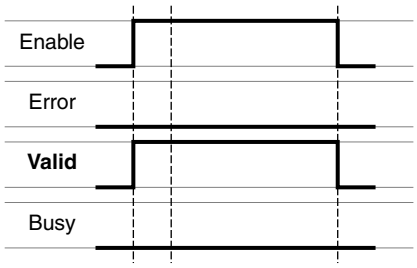
Example 5 Repeated execution without error (execution requires more than one call).



Example 6 Repeated execution with error (execution requires more than one call).



Example 7 Repeated execution without error (execution requires only one call).

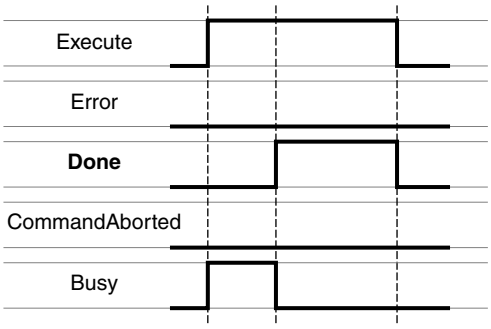


Example 8 Repeated execution with error (execution requires only one call).

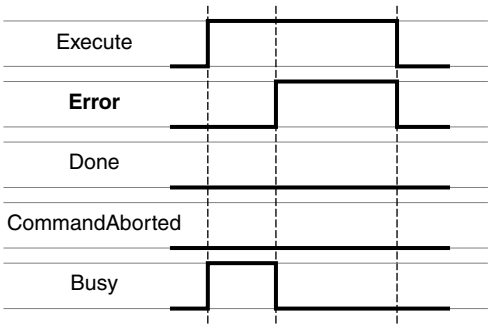
0198441113892, V2.09, 04.2012

2.6.2 Signal behavior of function blocks with the input `Execute`

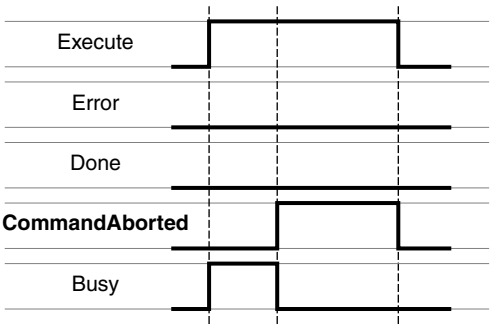
Example 1 Execution terminated without error.



Example 2 Execution terminated with error.

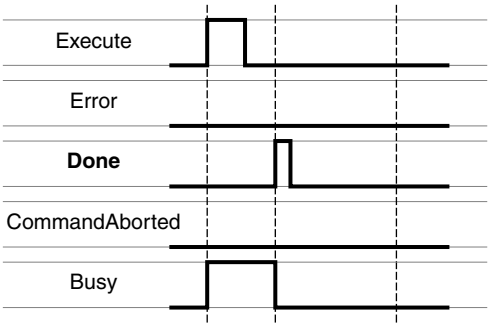


Example 3 Abortion of the execution because another function block takes over control.



0198441113892, V2.09, 04.2012

Example 4 Execution completed without error after Execute has been set to FALSE during execution.



0198441113892, V2.09, 04.2012

2.7 Single axis

2.7.1 Initialization

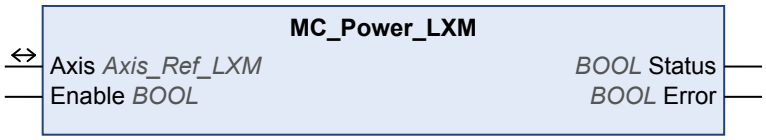
The initialization function block enables or disables the power stage. Other function blocks can only be used when the power stage is enabled.

2.7.1.1 MC_Power_LXM

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 occur during execution, the output `Error` is set.

Graphical representation



Compatible devices

LXM05, SD3 and LXM32

Inputs/outputs

The table below shows the outputs.

Output	Data type	Description
`Status`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE FALSE: Power stage is disabled. TRUE: Power stage is enabled.

"2.6 Basic inputs and outputs"

Notes

In the case of a Node Guarding error, the error memory must be reset by means of the function block "2.10.5.3 MC_Reset_LXM" before the power stage can be enabled again.

If the input Enable = TRUE, an error is signaled if the power supply is lost.

The output `Status` is set to FALSE and the output `Error` to TRUE. Once the power supply is available again, the output `Status` is set back to TRUE.

0198441113892, V2.09, 04.2012

2.7.2 Operating mode 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.

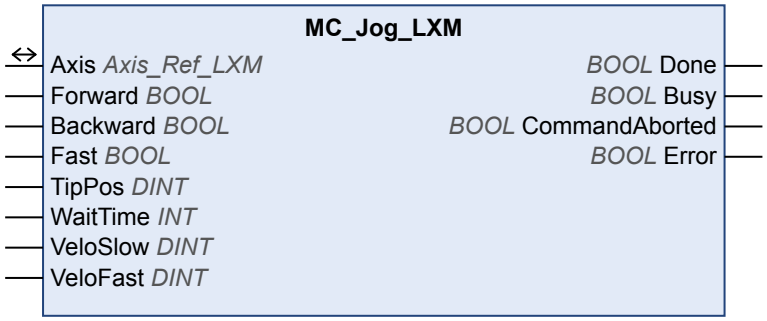
If movements in positive and negative directions are requested at the same time, there is no motor movement.

2.7.2.1 MC_Jog_LXM

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



Compatible devices

LXM05, SD3 and LXM32

0198441113892, V2.09, 04.2012

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
Forward	BOOL	Value range: FALSE, TRUE Initial value: FALSE FALSE: No movement in positive direction TRUE: Movement in positive direction is started.
Backward	BOOL	Value range: FALSE, TRUE Initial value: FALSE FALSE: No movement in negative direction TRUE: Movement in negative direction is started.
Fast	BOOL	Value range: FALSE, TRUE Initial value: FALSE 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 increments [inc]. The movement is stopped, the waiting time WaitTime starts. After the waiting time WaitTime has elapsed, a continuous movement is started.
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 [min^{-1}]. If Fast = FALSE, the movement is made at this velocity.
VeloFast	DINT	Value range: Initial value: 180 Velocity in [min^{-1}]. If Fast = TRUE, the movement is made at this velocity.

"2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

2.7.3 Operating mode Current 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.

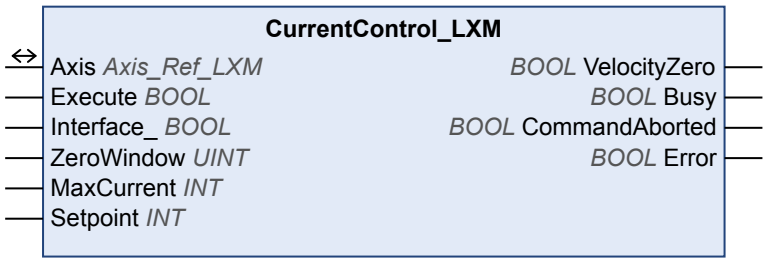
In the operating mode Current Control, you can set a reference value for the motor current. On the basis of this reference value, the device calculates a current with which the motor accelerates to a velocity limited by the load torque. Therefore, without a load, the motor accelerates up to the adjustable velocity limit.

2.7.3.1 CurrentControl_LXM

Function description

The function block starts the operating mode Current Control. In the operating mode Current Control, a reference value for the motor current is supplied. The input `Interface` lets you set the source of the reference value. The reference value can be supplied either via the input `Setpoint` or via the analog input ANA1.

Graphical representation



Compatible devices

LXM05

Inputs/outputs

The table below shows the inputs.

0198441113892, V2.09, 04.2012

Input	Data type	Description
Interface_	BOOL	Value range: FALSE, TRUE Initial value: FALSE FALSE: The reference value is supplied via the 10V analog input. TRUE: The reference value is supplied via the input Setpoint.
ZeroWindow	UINT	Value range: 0 ... 1000 Initial value: 0 This input is effective only if the input Interface_ has the value FALSE. If the reference value is supplied via an analog input, you can specify a zero voltage window as an absolute value in millivolts. Example: ZeroWindow = 20: The voltage range -20 mV ... +20 mV is interpreted as 0 mV.
MaxCurrent	INT	Value range: -30000 ... +30000 Initial value: 300 This input is effective only if the input Interface_ has the value FALSE. A current limitation [Apk x 100] is set according to the voltage at the 10V analog input.
Setpoint	INT	Value range: -30000 ... +30000 Initial value: 0 This input is only effective if the input Interface_ has the value TRUE. Current limitation value [Apk x 100].

The table below shows the outputs.

Output	Data type	Description
VelocityZero	BOOL	Value range: FALSE, TRUE Initial value: FALSE FALSE: Velocity > 0. TRUE: Velocity = 0, motor standstill.

"2.6 Basic inputs and outputs"

Notes This function block is not available for LXM32. Use the function block "2.7.4.1 MC_TorqueControl_LXM" for LXM32.

0198441113892, V2.09, 04.2012

2.7.4 Operating mode Profile Torque

⚠ 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.

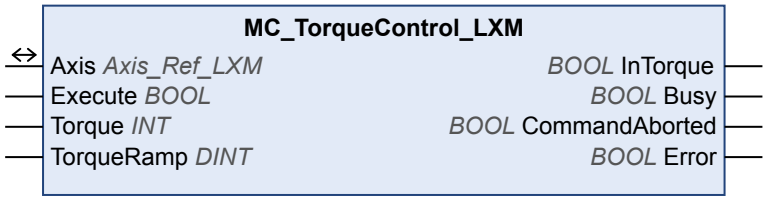
You can set a target torque in the operating mode Profile Torque. The movement is made with this target torque in the operating mode Profile Torque.

2.7.4.1 MC_TorqueControl_LXM

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 reference value for the target torque is supplied via the input `Torque`. 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



Compatible devices LXM32

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
`Torque`	`INT`	Value range: -30000 ... 30000 Initial value: 0 Target torque The value corresponds to 0.1% of 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.

The table below shows the outputs.

0198441113892, V2.09, 04.2012

Output	Data type	Description
InTorque	BOOL	Value range: FALSE, TRUE Initial value: FALSE FALSE: Target torque not yet reached TRUE: Target torque reached

"2.6 Basic inputs and outputs"

Notes

- This function block is not available for LXM05. Use the function block "2.7.3.1 CurrentControl_LXM" for LXM05.
- In the operating mode Profile Torque, a position overtravel does not trigger an error. A position overtravel results in a loss of the zero point. 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 "2.10.1.4 MC_ReadStatus_LXM" to check whether the drive is still homed.

0198441113892, V2.09, 04.2012

2.7.5 Operating mode Speed Control / Oscillator

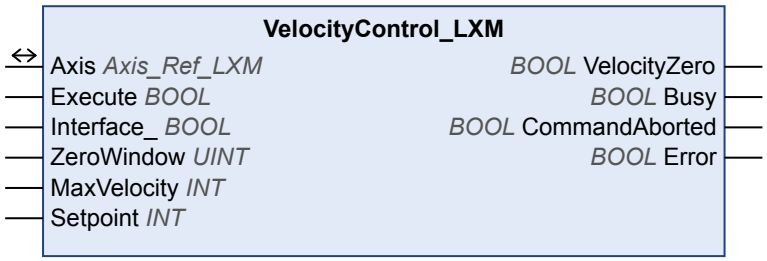
You can set a reference velocity in the operating mode Speed Control / Oscillator. The movement is performed with this reference velocity in the operating mode Speed Control.

2.7.5.1 VelocityControl_LXM

Function description

The function block starts the operating mode Speed Control / Oscillator. In the operating mode Speed Control, a movement is made with a set target velocity. The target velocity is set at the input `Velocity`. When the motor is at a standstill, `VelocityZero` is set.

Graphical representation



Compatible devices

LXM05 and SD3

Inputs/outputs

The table below shows the inputs.

Input	Data type	Description
`Interface_`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE: The reference value is supplied via the analog input. TRUE: Reference value is supplied via the input `Setpoint`.
`ZeroWindow`	`UINT`	Value range: 0 ... 1000 Initial value: 0 Zero voltage window at analog input in mV. Example: If the value is 20, the range -20 ... 20 mV is interpreted as 0.
`MaxVelocity`	`INT`	Value range: -30000 ... 30000 Initial value: 3000 The value is only considered if the input `Interface` is FALSE. The value specifies the maximum speed of rotation (ANA1 = 10V) in min^{-1}. Adapt the value to your motor and the mechanical situation.
`Setpoint`	`INT`	Value range: -30000 ... 30000 Initial value: 0 The value is only considered if the input `Interface` is TRUE. The value specifies the reference value for the speed of rotation in min^{-1}.

The table below shows the outputs.

0198441113892, V2.09, 04.2012

Output	Data type	Description
VelocityZero	BOOL	Value range: FALSE, TRUE Initial value: FALSE: The motor is not at a standstill. TRUE: The motor is at a standstill.

"2.6 Basic inputs and outputs"

Notes In the operating mode Speed Control / Oscillator, a position overtravel does not trigger an error. A position overtravel results in a loss of the zero point.

0198441113892, V2.09, 04.2012

2.7.6 Operating mode Profile Velocity

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.

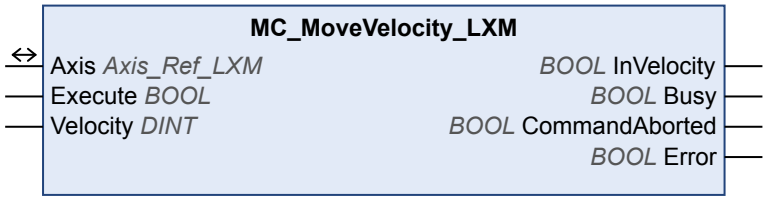
Transitions between two target velocities are performed on the basis of a motion profile. The motion profile is determined by the profile generator in the drive on the basis of the actual velocity, the target velocity and the acceleration and deceleration ramps.

2.7.6.1 MC_MoveVelocity_LXM

Function description

The function block starts the operating mode Profile Velocity with the velocity `Velocity`. When the target velocity is reached, `InVelocity` is set.

Graphical representation



Compatible devices

LXM05, SD3 and LXM32

Inputs/outputs

The table below shows the inputs.

Input	Data type	Description
`Velocity`	`DINT`	Value range: Initial value: 0 Target velocity

The table below shows the outputs.

Output	Data type	Description
`InVelocity`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE FALSE: Target velocity not yet reached. TRUE: Target velocity reached.

"2.6 Basic inputs and outputs"

Notes

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.

0198441113892, V2.09, 04.2012

2.7.7 Operating mode Profile Position

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

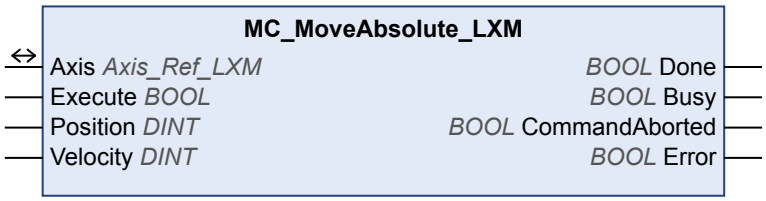
A zero point must be defined with the operating mode Homing prior to the first absolute movement.

2.7.7.1 MC_MoveAbsolute_LXM

Function description

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

Graphical representation



Compatible devices

LXM05, SD3 and LXM32

Inputs/outputs

The table below shows the inputs.

Input	Data type	Description
`Position`	`DINT`	Value range: Initial value: Target position absolute:
`Velocity`	`DINT`	Value range: Initial value: 60 Target velocity value LXM32: 1 ... 2147483647 [usr v] LXM05 and SD328: 1 ... 13200 [min^{-1}]

"2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

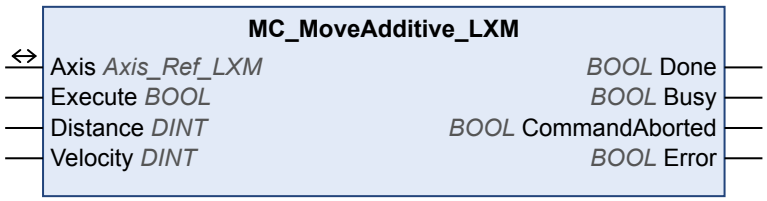
Notes

- Absolute positioning requires a valid zero point. You can use the function block "2.10.1.4 MC_ReadStatus_LXM" to check for a valid zero point.

2.7.7.2 MC_MoveAdditive_LXM

Function description The function block starts a movement to the original target position plus distance `Distance` at velocity `Velocity`.

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

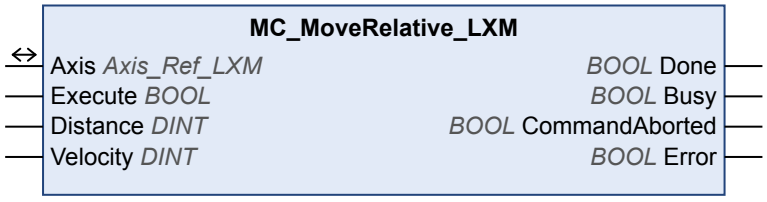
Input	Data type	Description
`Distance`	`DINT`	Value range: Initial value: Target position relative with reference to the previous target position
`Velocity`	`DINT`	Value range: Initial value: 60 Target velocity

"2.6 Basic inputs and outputs"

2.7.7.3 MC_MoveRelative_LXM

Function description The function block starts a movement by distance `Distance` at velocity `Velocity`.

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

0198441113892, V2.09, 04.2012

Input	Data type	Description
Distance	DINT	Value range: Initial value: Target position relative with reference to actual position
Velocity	DINT	Value range: Initial value: 60 Target velocity

"2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

2.7.8 Operating mode 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.

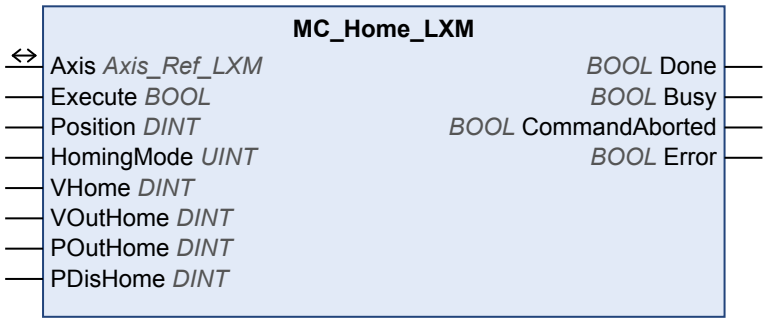
- Reference movement: Movement to a limit switch, a reference switch or the index pulse of the motor encoder. When the position is reached, a position reference is automatically created. This position becomes the absolute user-defined position.
- Position setting: The current motor position is set to 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.

2.7.8.1 MC_Home_LXM

Function description The function block configures and starts a reference movement.

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

0198441113892, V2.09, 04.2012

Input	Data type	Description
Position	DINT	Value range: -2147483648 ... 2147483647 Initial value: 0 HomingMode 1 ... 8: Position at reference point HomingMode 9: Position for Position Setting
HomingMode	UINT	Value range: Initial value: 1 1: LIMN with index pulse 2: LIMP with index pulse 7: REF+ with index pulse, inverted, outside 8: REF+ with index pulse, inv., inside 9: REF+ with index pulse, not inv., inside 10: REF+ with index pulse, not inv., outside 11: REF- with index pulse, inv., outside 12: REF- with index pulse, inv., inside 13: REF- with index pulse, not inv., inside 14: REF- with index pulse, not inv., outside 17: LIMN 18: LIMP 23: REF+, inv., outside 24: REF+, inv., inside 25: REF+, not inv., inside 26: REF+, not inverted, outside 27: REF-, inv., outside 28: REF-, inv., inside 29: REF-, not inv., inside 30: REF-, not inverted, outside 33: Index pulse in negative direction 34: Index pulse in positive direction 35: Position setting
VHome	DINT	Value range: Initial value: 60 Target velocity for searching the switch. HomingMode 1 ... 34 only.
VOutHome	DINT	Value range: Initial value: 6 Target velocity for moving away from switch.
POutHome	DINT	Value range: 0 ... 2147483647 Initial value: 0 Maximum distance for search for switching point. 0: Search distance monitoring disabled >0: Maximum distance After detection of the switch, the drive starts to search for the defined switching point. If the defined switching point is not found within the distance defined here, the reference movement is canceled with an error.
PDisHome	DINT	Value range: 0 ... 2147483647 Initial value: 200 Maximum search distance after overtravel of switch. 0: Search distance monitoring disabled >0: Search distance The switch must be activated again within this search distance, otherwise the reference movement is cancelled.

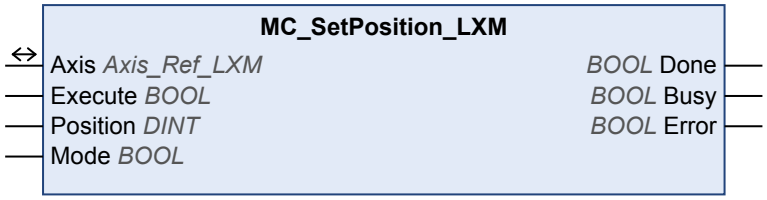
"2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

2.7.8.2 MC_SetPosition_LXM

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



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
`Position`	`DINT`	Value range: -2147483648 ... 2147483647 Initial value: 0 Value for position setting in [usr].
`Mode`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE: The actual position is set to the value of the input `Position`. TRUE: The value of the input `Position` is added to the actual position.

"2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

2.7.9 Stopping

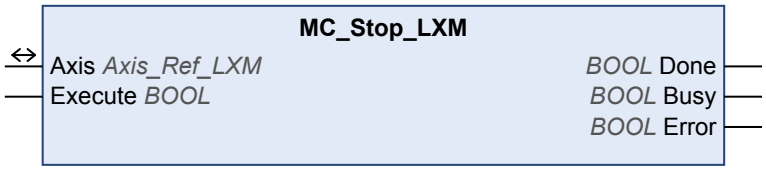
Each operating mode can be canceled by stopping. Stopping the operating mode does not generate an error.

2.7.9.1 MC_Stop_LXM

Function description

The function block is used to stop the current movement. The operating mode is stopped by the function block.

Graphical representation



Compatible devices

LXM05, SD3 and LXM32

Inputs/outputs

"2.6 Basic inputs and outputs"

Notes

- LXM05 and SD328:
  The deceleration ramp is set with the function block "2.10.2.2 SetDriveRamp_LXM".

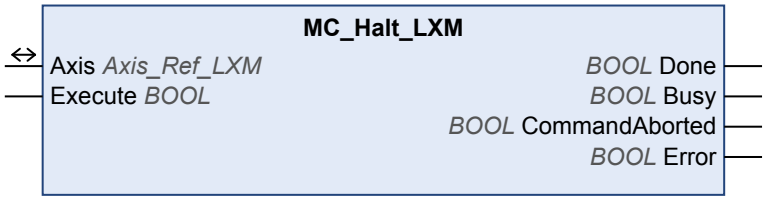
  LXM32:
  The deceleration ramp is set with the function block "2.10.2.3 SetStopRamp_LXM".
- The function block can only be interrupted by disabling the power stage via the function block "2.7.1.1 MC_Power_LXM".
- As long as the input `Execute` is TRUE, no other function block with the exception of "2.7.1.1 MC_Power_LXM" can be started.

2.7.9.2 MC_Halt_LXM

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



Compatible devices

LXM32

Inputs/outputs

"2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

Notes

- The deceleration ramp is set in the function block "2.10.2.2 SetDriveRamp_LXM".

0198441113892, V2.09, 04.2012

2.7.10 Position capture via signal input

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.

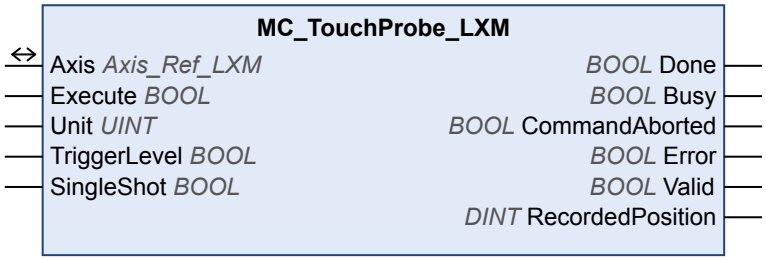
Settings:

- 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.

2.7.10.1 MC_TouchProbe_LXM

Function description The function block configures and starts position capture.

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
`Unit`	`UINT`	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 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.

The table below shows the outputs.

Output	Data type	Description
`RecordedPosition`	`DINT`	Value range: -2147483648 ... 2147483647 Initial value: 0 Captured motor position

"2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

2.7.10.2 MC_AbortTrigger_LXM

Function description The function block is used to terminate position capture.

Graphical representation

Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
Unit	UINT	Value range: 1 ... 2 Initial value: 1 1: Cancel position capture via Capture input 1. 2: Cancel position capture via Capture input 2.

"2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

2.8 Multi axis

2.8.1 Operating mode Electronic Gear

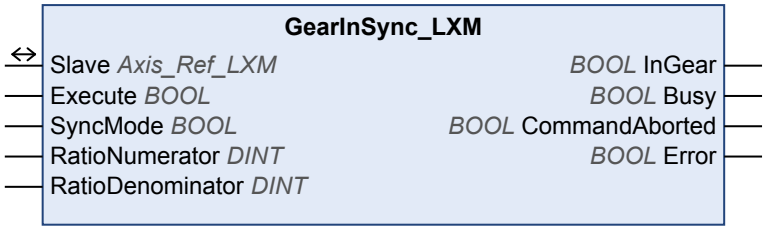
In the operating mode Electronic Gear, movements are carried out according to externally supplied reference value signals. A new position reference value is calculated on the basis of these reference value signals plus an adjustable gear ratio.

2.8.1.1 GearInSync_LXM

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



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs/outputs.

Input/output	Data type	Description
`Slave`	`Axis_Ref_LXM`	Value range: Initial value: Name of the slave axis

The table below shows the inputs.

0198441113892, V2.09, 04.2012

Input	Data type	Description
SyncMode	BOOL	Value range: FALSE, TRUE Initial value: TRUE FALSE: Position synchronization without compensation movement. TRUE: Position synchronization with compensation movement
RatioNumerator	DINT	Value range: -2147483648 ... 2147483647 Initial value: 1 Numerator of gear ratio
RatioDenominator	DINT	Value range: 1 ... 32767 Initial value: 1 Denominator of gear ratio

The table below shows the outputs.

Output	Data type	Description
InGear	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The output is set to TRUE if the adjusted gear ratio is reached for the first time in the operating mode Electronic Gear.

"2.6 Basic inputs and outputs"

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 "2.8.1.2 MC_GearIn_LXM".
- 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.

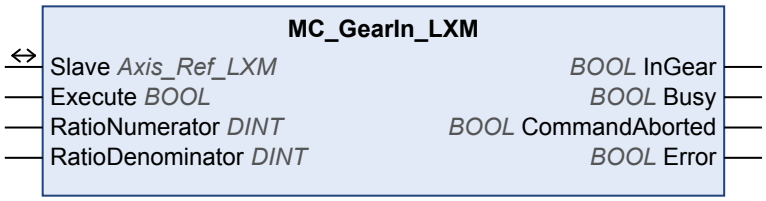
2.8.1.2 MC_GearIn_LXM

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.

0198441113892, V2.09, 04.2012

Graphical representation



Compatible devices LXM32

Inputs/outputs The table below shows the inputs/outputs.

Input/output	Data type	Description
`Slave`	`Axis_Ref_LXM`	Value range: Initial value: Name of the slave axis for which this function block is to be executed. This name must be declared in the PLC configuration. In the configuration, the name can be found next to the appropriate CANopen node.

The table below shows the inputs.

Input	Data type	Description
`RatioNumerator`	`DINT`	Value range: -2147483648 ... 2147483647 Initial value: 1 Gear ratio: Numerator of gear ratio
`RatioDenominator`	`DINT`	Value range: 1 ... 32767 Initial value: 1 Gear ratio: Denominator of gear ratio

The table below shows the outputs.

Output	Data type	Description
`InGear`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE The output is set to TRUE if the adjusted gear ratio is reached for the first time.

"2.6 Basic inputs and outputs"

Notes

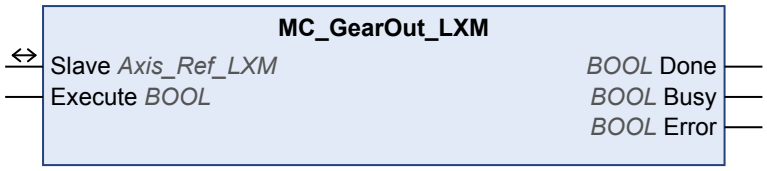
- This requires the parameter `GEARratio` to be 0 (see product manual). This way, `RatioNumerator` and `RatioDenominator` are used to calculate the gear ratio
- The enabled direction of movement in the operating mode Electronic Gear is set via the parameter `GEARdir_enabl`.
- The operating mode Electronic Gear with the method Position Synchronization is started with the function block "2.8.1.1 GearInSync_LXM".
- 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.

0198441113892, V2.09, 04.2012

2.8.1.3 MC_GearOut_LXM

Function description The function block terminates the operating mode Electronic Gear.

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs/outputs.

Input/output	Data type	Description
Slave	Axis_Ref_LXM	Value range: Initial value: Name of the slave axis

"2.6 Basic inputs and outputs"

Notes

- The function block terminates the operating mode Electronic Gear.
- The motor decelerates with the deceleration ramp "2.10.2.2 SetDriveRamp_LXM".

0198441113892, V2.09, 04.2012

2.9 Motion Sequence

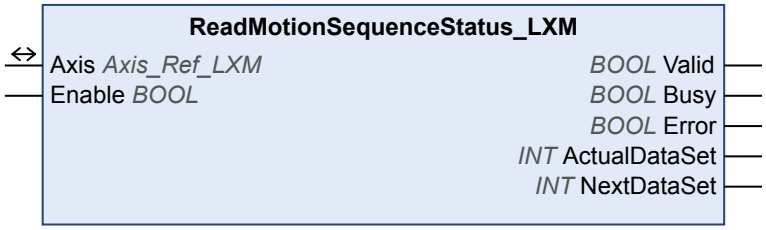
2.9.1 Operating mode Motion Sequence

In the operating mode Motion Sequence, movements are performed via parameterizable data sets. These parameterizable data sets can be started individually or as a sequence. A parameterizable data set contains settings on the type of data set, the appropriate target values and the setting for the subsequent data set.

2.9.1.1 ReadMotionSequenceStatus_LXM

Function description The function block returns the current status of the operating mode Motion Sequence.

Graphical representation



Compatible devices LXM32

Inputs/outputs The table below shows the outputs.

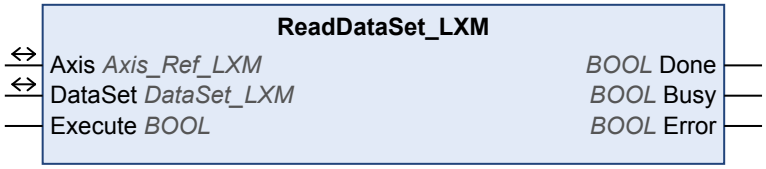
Output	Data type	Description
`ActualDataSet`	`INT`	The output indicates the data set number of the active data set.
`NextDataSet`	`INT`	The output indicates the number of the next data set to be processed in the sequence.

"2.6 Basic inputs and outputs"

2.9.1.2 ReadDataSet_LXM

Function description The function block reads the current data set in the operating mode Motion Sequence.

Graphical representation



Compatible devices LXM32

Inputs/outputs The table below shows the inputs/outputs.

0198441113892, V2.09, 04.2012

Input/output	Data type	Description
DataSet	DataSet_LXM	The data type has the following structure: DataSetNumber, DataSetType, Value1, Value2, Value3, Value4, Transition, NextCondition, WaitTime and NextDataSet.

"2.6 Basic inputs and outputs"

2.9.1.3 WriteDataSet_LXM

Function description The function block configures a data set for the operating mode Motion Sequence.

Graphical representation

Compatible devices LXM32

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
DataSet	DataSet_LXM	The data type has the following structure: DataSetNumber, DataSetType, Value1, Value2, Value3, Value4, Transition, NextCondition, WaitTime and NextDataSet.

"2.6 Basic inputs and outputs"

2.9.1.4 WriteTransitionCondition_LXM

Function description The function block meets the transition condition (NextCondition = 2) for the next data set in the operating mode Motion Sequence.

Graphical representation

Compatible devices LXM32

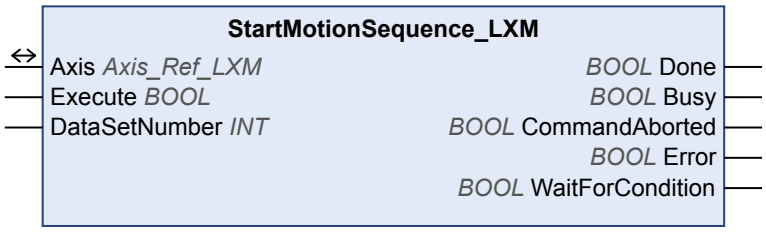
Inputs/outputs "2.6 Basic inputs and outputs"

2.9.1.5 StartMotionSequence_LXM

Function description The function block starts and monitors the operating mode Motion Sequence.

0198441113892, V2.09, 04.2012

Graphical representation



Compatible devices LXM32

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
DataSetNumber	INT	Value range: Initial value: 0 The input specifies the number of the data set to be used for starting a sequence.

The table below shows the outputs.

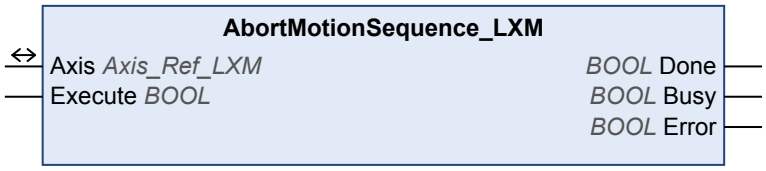
Output	Data type	Description
WaitForCondition	BOOL	Value range: FALSE, TRUE Initial value: FALSE The output indicates that the data set has been completed and that the sequence is waiting for the transition condition to be met. Once the transition condition is met, the subsequent data set started.

"2.6 Basic inputs and outputs"

2.9.1.6 AbortMotionSequence_LXM

Function description The function block cancels a sequence in the operating mode Motion Sequence.

Graphical representation



Compatible devices LXM32

Inputs/outputs "2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

2.10 Administrative

2.10.1 Reading a parameter

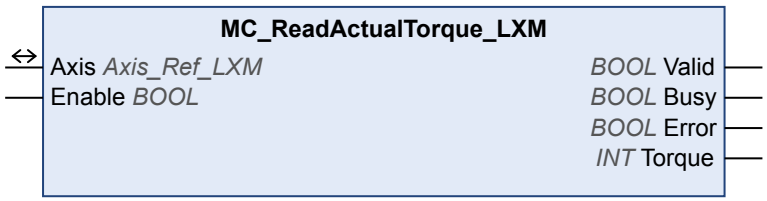
The following functions blocks allow you to read drive parameters such as the actual position or the actual velocity.

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

2.10.1.1 MC_ReadActualTorque_LXM

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

Graphical representation



Compatible devices LXM32

Inputs/outputs The table below shows the outputs.

Output	Data type	Description
`Torque`	`INT`	Value range: Initial value: 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.

"2.6 Basic inputs and outputs"

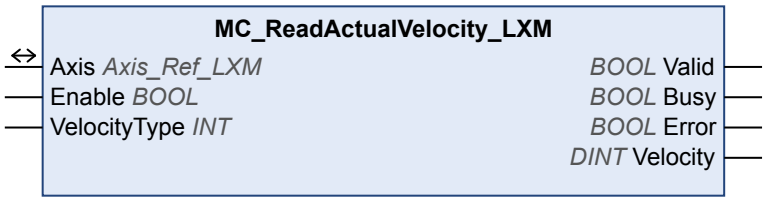
Notes The function block uses Service Data Objects (SDO) to read the parameter. Therefore, it is strongly recommended not to permanently set the input `Enable` to TRUE. This may cause overload on the fieldbus. It is recommended to deactivate the function block when the input `Busy` is set to FALSE.

2.10.1.2 MC_ReadActualVelocity_LXM

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

0198441113892, V2.09, 04.2012

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
`VelocityType`	`INT`	Value range: 0 ... 3 Initial value: 0 Specification of the source of the velocity: • 0: Actual velocity of the motor [min^{-1}] • 1: Reference velocity (from profile generator) [min^{-1}] • 2: Actual velocity (from profile generator) [min^{-1}] • 3: Actual velocity at PTI interface [inc/s] NOTE: The PTI interface (Pulse Train In) is also referred to as RS422 interface in the case of some products. LXM05 and SD328: the PTI interface is CN5.

The table below shows the outputs.

Output	Data type	Description
`Velocity`	`DINT`	Value range: Initial value: Velocity value of the source selected for the input `VelocityType`.

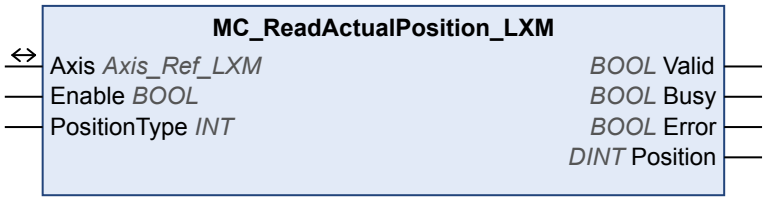
"2.6 Basic inputs and outputs"

Notes The function block uses Service Data Objects (SDO) to read the parameter from the device. Therefore, it is strongly recommended not to permanently set the input `Enable` to TRUE. This may cause overload on the fieldbus. It is recommended to deactivate the function block when the output `Busy` is set to FALSE.

2.10.1.3 MC_ReadActualPosition_LXM

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

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

0198441113892, V2.09, 04.2012

Input	Data type	Description
PositionType	INT	Value range: 0 ... 7 Initial value: 0 Specification of the source of the position: • 0: Actual position of motor [usr] • 1: Actual position of motor [inc] • 2: Reference position (from profile generator) [usr] • 3: Reference position (from profile generator) [inc] • 4: Actual position of an external encoder [usr] • 5: Actual position of an external encoder [inc] • 6: Actual position (from profile generator) [usr] • 7: Actual position (from profile generator) [inc]

The table below shows the outputs.

Output	Data type	Description
Position	DINT	Value range: Initial value: Position value of the source selected for the input PositionType.

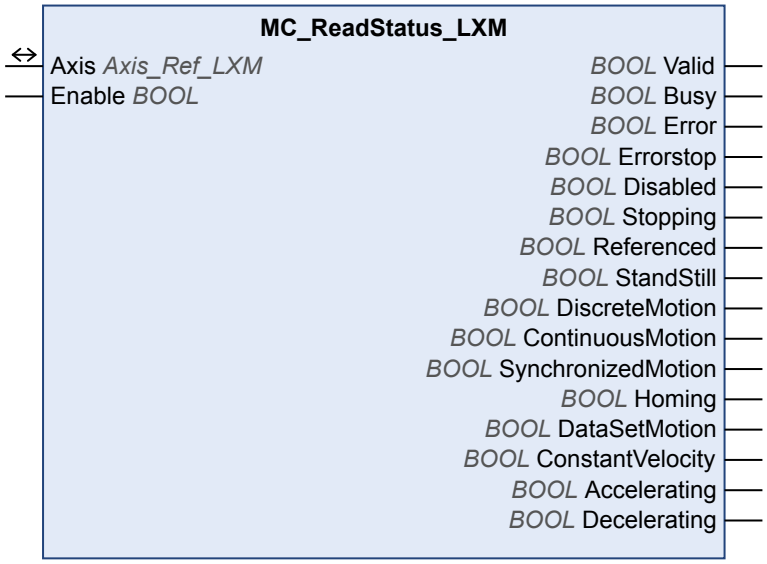
"2.6 Basic inputs and outputs"

Notes The function block uses Service Data Objects (SDO) to read the parameter from the device. Therefore, it is strongly recommended not to permanently set the input Enable to TRUE. This may cause overload on the fieldbus. It is recommended to deactivate the function block when the input Busy is set to FALSE.

2.10.1.4 MC_ReadStatus_LXM

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

Graphical representation



Compatible devices LXM05, SD3 and LXM32

0198441113892, V2.09, 04.2012

Inputs/outputs The table below shows the outputs.

Output	Data type	Description
Errorstop	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The movement has been interrupted by an error.
Disabled	BOOL	Value range: FALSE, TRUE Initial value: FALSE FALSE: Power stage is enabled. TRUE: Power stage is disabled
Stopping	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The function block "2.7.9.1 MC_Stop_LXM" is being executed or the movement is being stopped.
Referenced	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The zero point (reference point) is valid.
StandStill	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The movement hast been stopped.
DiscreteMotion	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The operating mode Profile Position has been started.
ContinuousMotion	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The operating mode Profile Velocity has been started.
SynchronizedMotion	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: A synchronized movement at a constant velocity is performed. (for example, in the operating mode Electronic Gear)
Homing	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The operating mode Homing has been started.
DataSetMotion	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The operating mode Motion Sequence has been started.
ConstantVelocity	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: A movement at a constant velocity is performed.
Accelerating	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The motor accelerates.
Decelerating	BOOL	Value range: FALSE, TRUE Initial value: FALSE TRUE: The motor decelerates.

"2.6 Basic inputs and outputs"

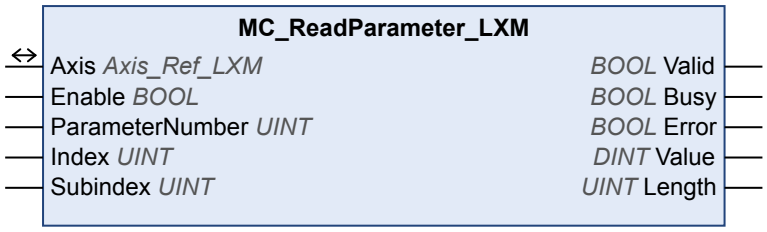
Notes At any given point in time, the drive is in one of the states: StandStill, DiscreteMotion, ContinuousMotion, Stopping, Disabled or ErrorStop. The corresponding output is then TRUE.

0198441113892, V2.09, 04.2012

2.10.1.5 MC_ReadParameter_LXM

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

Graphical representation



Compatible devices LXM05, SD3 and LXM32

0198441113892, V2.09, 04.2012

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
ParameterNumber	UINT	Value range: 0 ... 65535 Initial value: 1000 Number of the parameter: 1: Reference position (from profile generator). LXM32: _p_tarRAMPusr LXM05, SD328: _p_tarRAMPusr 2: Positive position limit of software limit switch. LXM32: MON_swLimPusr LXM05, SD328: SPVswLimPusr 3: Negative position limit of software limit switch. LXM32: MON_swLimNusr LXM05, SD328: SPVswLimNusr 4: Monitoring of the positive software limit switch. (Activated: Bit 0 = 0. Deactivated: Bit 0 = 1) LXM32: MON_SW_Limits LXM05, SD328: SPV_SW_Limits 5: Monitoring of the negative software limit switch. (Activated: Bit 0 = 0. Deactivated: Bit 0 = 1). LXM32: MON_SW_Limits LXM05, SD328: SPV_SW_Limits 10: Actual velocity. LXM32: _n_act LXM05, SD328: _n_act 11: Target velocity. LXM32: _v_targetRAMP LXM05, SD328: _n_targetRAMP 1000: Selection via index and subindex.
Index	UINT	Value range: 0 ... 65535 Initial value: 0 Index of parameter to be read. Only valid if ParameterNumber = 1000. See the product manual for an overview of the parameters.
Subindex	UINT	Value range: 0 ... 255 Initial value: 0 Subindex of parameter to be read. Only valid if ParameterNumber = 1000. See the product manual for an overview of the parameters.

The table below shows the outputs.

Output	Data type	Description
Value	DINT	Value range: -2147483648 ... 2147483647 Initial value: 0 Value of the parameter.
Length	INT	Value range: 0 ... 65535 Initial value: 0 Length of the parameter in bytes.

"2.6 Basic inputs and outputs"

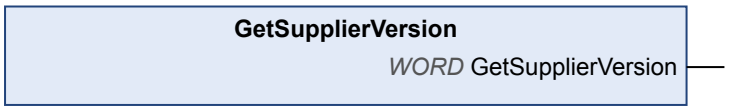
0198441113892, V2.09, 04.2012

Notes The function block uses Service Data Objects (SDO) to read the parameter. Therefore, it is strongly recommended not to permanently set the input `Enable` to TRUE. This may cause overload on the field-bus. It is recommended to deactivate the function block when the input `Busy` is set to FALSE.

2.10.1.6 GetSupplierVersion

Function description The function returns the version of the library of the device.

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the outputs.

Output	Data type	Description
`GetSupplierVersion`	`WORD`	The output provides the version number of the library. Convert the decimal value to hex. Example: GetSupplierVersion = 12368 = 3050_h = Version 3.0.5.0

0198441113892, V2.09, 04.2012

2.10.2 Writing a parameter

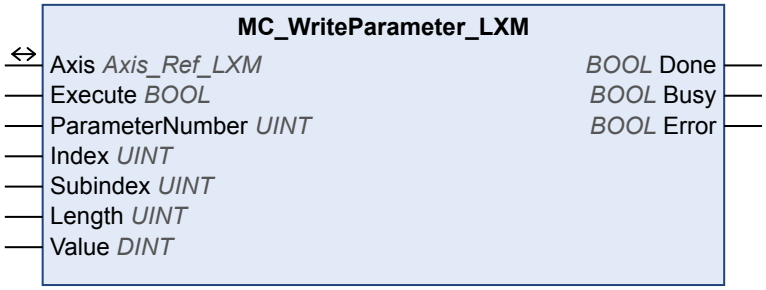
The following function bocks allow you to write drive parameters, for example the values for the acceleration and deceleration ramps.

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

2.10.2.1 MC_WriteParameter_LXM

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

Graphical representation



Compatible devices LXM05, SD3 and LXM32

0198441113892, V2.09, 04.2012

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
ParameterNumber	UINT	Value range: 0 ... 65535 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 inputs Index and SubIndex. See the product manual for a list of the parameters with the corresponding CANopen address.
Index	UINT	Value range: 0 ... 65535 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 CANopen address.
Subindex	UINT	Value range: 0 ... 255 Initial value: 0 Subindex 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 CANopen address.
Length	UINT	Value range: 0 ... 65535 Initial value: 0 Length of the parameter to be written in bytes.
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.

"2.6 Basic inputs and outputs"

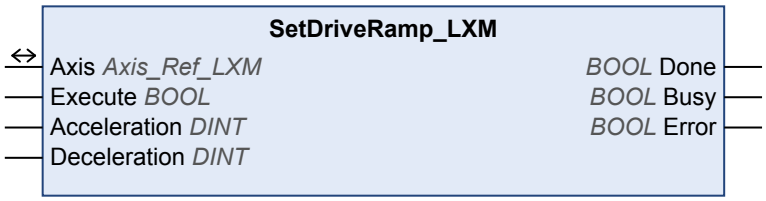
Notes If the inputs ParameterNumber, Index or Subindex are changed while Busy is TRUE, the function block uses the previous values. The next time the function block is executed, the new values will be used.

2.10.2.2 SetDriveRamp_LXM

Function description The function block configures the acceleration ramp and the deceleration ramp of the device.

0198441113892, V2.09, 04.2012

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
Acceleration	DINT	LXM05: Value range: 30 ... 3000000 Initial value: 600 SD328: Value range: 1 ... 3000000 Initial value: 600 LXM32: Value range: 1 ... 2147483647 Initial value: 600 LXM05 and SD328: Acceleration ramp in min^{-1}/s. LXM32: Acceleration ramp in user-defined units (usr_a).
Deceleration	DINT	LXM05: Value range: 750 ... 3000000 Initial value: 750 SD328: Value range: 1 ... 3000000 Initial value: 750 LXM32: Value range: 1 ... 2147483647 Initial value: 600 LXM05 and SD328: Deceleration ramp in min^{-1}/s. LXM32: Deceleration ramp in user-defined units (usr_a).

"2.6 Basic inputs and outputs"

Notes Note the following for drives with high external moments of inertia or for highly dynamic applications: The motors regenerate energy during deceleration. The DC bus can absorb a limited amount of energy in the capacitors. Connecting additional capacitors to the DC bus increases the amount of energy that can be absorbed. If the capacity of the capacitors is exceeded, the excess energy must be discharged via internal or external braking resistors. If the energy is not discharged, an overvoltage monitor will shut off the power stage. Overvoltages can be limited by adding a braking resistor with a corresponding braking resistor controller. This converts the regenerated energy to heat energy during deceleration.

2.10.2.3 SetStopRamp_LXM

Function description This function block is used to set the deceleration ramp for the function block "2.7.9.1 MC_Stop_LXM".

0198441113892, V2.09, 04.2012

Graphical representation

Compatible devices LXM32

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
`RampType`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE: MC_Stop_LXM is decelerated with a deceleration ramp. TRUE: MC_Stop_LXM is decelerated with a torque ramp.
`Value`	`DINT`	The value depends on the adjusted ramp (`RampType`). `RampType` = FALSE: Value range: 200 ... 2147483647 Initial value: 6000 The value corresponds to the device parameter `RAMPquickstop`; it is specified in the unit [usr_a]. See the product manual for additional information on this parameter. `RampType` = TRUE: Value range: 1 ... 30000 Initial value: The value corresponds to the device parameter `LIM_I_maxQSTP`; it is specified in the unit [A_{rms}]. See the product manual for additional information on this parameter.

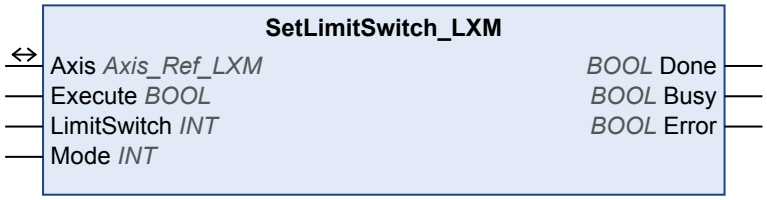
"2.6 Basic inputs and outputs"

Notes "2.10.2.3 SetStopRamp_LXM" also modifies the deceleration ramp for "Quick Stop".

2.10.2.4 SetLimitSwitch_LXM

Function description This function block is used to parameterize the positive limit switch (LIMP) and the negative limit switch (LIMN).

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the inputs.

0198441113892, V2.09, 04.2012

Input	Data type	Description
LimitSwitch	INT	Value range: 1 ... 2 Initial value: 1 1: Limit switch in positive direction of movement LIMP 2: Limit switch in negative direction of movement LIMN
Mode	INT	Value range: 0 ... 2 Initial value: 0 0: Deactivate limit switch 1: Activate limit switch as normally closed contact (NCC) 2: Activate limit switch as normally open contact (NOC)

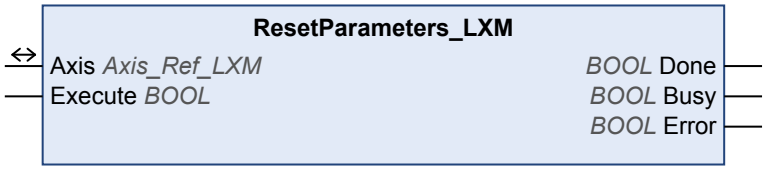
"2.6 Basic inputs and outputs"

Notes The function block can only be executed if the drive is in the operating state **3** Switch On Disabled (operating state of drive).

2.10.2.5 ResetParameters_LXM

Function description This function block restores all parameters to the factory settings.

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs "2.6 Basic inputs and outputs"

Notes Observe the information provided in chapter .

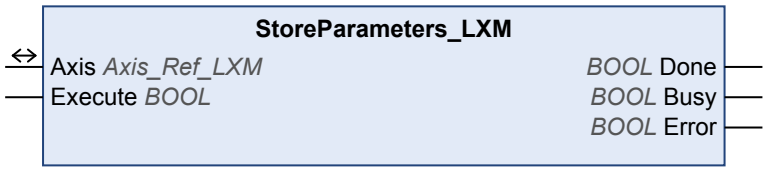
- The function block can only be executed if the drive is in the operating state **3** Switch On Disabled (operating state of drive). To transition to this state, disable the power stage with the function block "2.7.1.1 MC_Power_LXM".
- All parameters are restored to the factory settings with the exception of:
  - Communication parameters
  - Inversion of direction
  - Signal type PTI interface
  - Start-up operating mode
  - Encoder simulation settings
  - Functions of the digital inputs and outputs
- The new settings are not saved to the EEPROM. Use "2.10.2.6 StoreParameters_LXM" to save the new settings to the EEPROM.

2.10.2.6 StoreParameters_LXM

Function description The function block saves the parameter values to the EEPROM.

0198441113892, V2.09, 04.2012

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs "2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

2.10.3 Saving and restoring device configuration

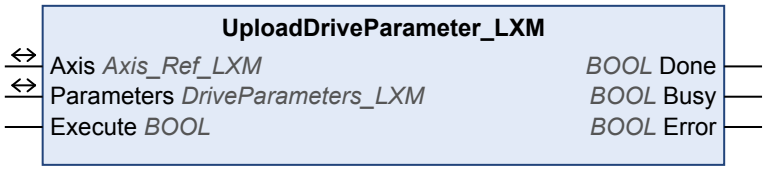
Using a function block, you can upload the device configuration from the drive to the controller. A further function block lets you download a device configuration stored on the controller to a drive.

2.10.3.1 UploadDriveParameter_LXM

Function description

The function blocks reads the parameter values that can be modified from the device. See also "2.10.3.2 DownloadDriveParameter_LXM".

Graphical representation



Compatible devices

LXM05, SD3 and LXM32

Inputs/outputs

The table below shows the inputs/outputs.

Input/output	Data type	Description
`Parameters`	`DriveParameters_LXM`	Value range: Initial value: List of the device parameters.

"2.6 Basic inputs and outputs"

Notes

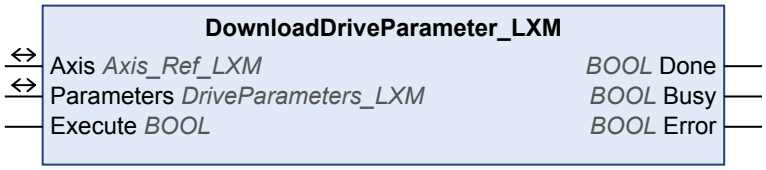
- The function block can only be executed if the drive is in the operating state **3** Switch On Disabled (operating state of drive). To transition to this state, disable the power stage with the function block "2.7.1.1 MC_Power_LXM".
- The two function blocks "2.10.3.2 DownloadDriveParameter_LXM" and "2.10.3.1 UploadDriveParameter_LXM" allow you to save the parameters stored in a device to an identical device without using the commissioning software.

2.10.3.2 DownloadDriveParameter_LXM

Function description

The function blocks writes the parameter values that can be modified to the device. Before calling the function block, you must execute "2.10.3.1 UploadDriveParameter_LXM". If not, an error message will be generated.

Graphical representation



Compatible devices

LXM05, SD3 and LXM32

0198441113892, V2.09, 04.2012

Inputs/outputs The table below shows the inputs/outputs.

Input/output	Data type	Description
Parameters	DriveParameters_LXM	List of device parameters

"2.6 Basic inputs and outputs"

Notes

- 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 "2.7.1.1 MC_Power_LXM".
- In order to permanently store the parameters, you must save them to the EEPROM using the function block "2.10.2.6 StoreParameters_LXM".
- The two function blocks "2.10.3.2 DownloadDriveParameter_LXM" and "2.10.3.1 UploadDriveParameter_LXM" allow you to save the parameters stored in a device to an identical device without using the commissioning software.

0198441113892, V2.09, 04.2012

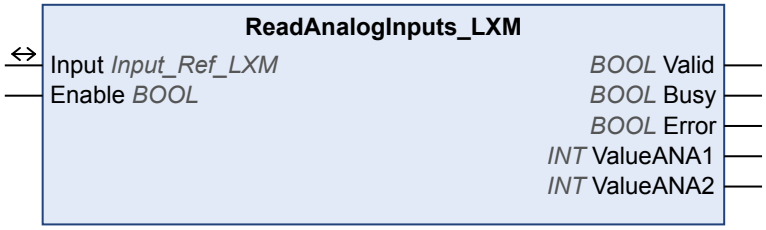
2.10.4 Inputs and outputs

The following function blocks allow you to access the digital and analog inputs and outputs of all CAN nodes in the system..

2.10.4.1 ReadAnalogInputs_LXM

Function description The function block reads the current values of the analog inputs.

Graphical representation



Compatible devices LXM05 and SD3

Inputs/outputs The table below shows the outputs.

Output	Data type	Description
`ValueANA1`	`INT`	Value range: -10000 ... 10000 Initial value: 0 Corresponds to the input voltage in [mV] at the analog input ANA1.
`ValueANA2`	`INT`	Value range: Initial value: Corresponds to the input voltage in [mV] at the analog input ANA2.

"2.6 Basic inputs and outputs"

Notes

- The function block uses Service Data Objects (SDO) to read the parameter from the device. Therefore, it is strongly recommended not to permanently set the input `Enable` to TRUE. This may cause overload on the fieldbus. It is recommended to deactivate the function block when the input `Busy` is set to FALSE.
- SD328: ANA2 is not available.

2.10.4.2 MC_ReadDigitalInput_LXM

Function description Reads the current state of the digital inputs of the drive.

Graphical representation

Compatible devices LXM05, SD3 and LXM32

0198441113892, V2.09, 04.2012

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
`InputNumber`	`INT`	Value range: 0 ... 5 Initial value: 0 Number of the input to be read. Input IO: Bit number DI0: 0 DI1: 1 DI2: 2 DI3: 3 DI4: 4 DI5: 5

The table below shows the outputs.

Output	Data type	Description
`Value`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE FALSE: Level at selected input is 0 V. TRUE: Level at selected input is 24 V.
`Inputs`	`WORD`	Value range: 00_h ... $3F_h$ Initial value: 00_h Image of the inputs as a bit pattern. Bit 0 = first input. Input IO: Bit number DI0: 0 DI1: 1 DI2: 2 DI3: 3 DI4: 4 DI5: 5

"2.6 Basic inputs and outputs"

Notes See the product manual for a description of the digital inputs.

2.10.4.3 MC_ReadDigitalOutput_LXM

Function description The function block is used to get the current state of the digital outputs.

Graphical representation

Compatible devices LXM05, SD3 and LXM32

0198441113892, V2.09, 04.2012

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
`OutputNumber`	`INT`	Value range: 0 ... 2 Initial value: 0 Number of the output to be read. Output IO: Bit number DQ0: 0 DQ1: 1 DQ2: 2 (LXM05, SD328 and LXM32M only)

The table below shows the outputs.

Output	Data type	Description
`Value`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE FALSE: Level at selected output is 0 V. TRUE: Level at selected output is 24 V.
`Outputs`	`WORD`	Value range: 00_h ... 03_h Initial value: 00_h Image of the outputs as a bit pattern. Bit 0 = first output. Output IO: Bit number DQ0: 0 DQ1: 1 DQ2: 2 (LXM05, SD328 and LXM32M only)

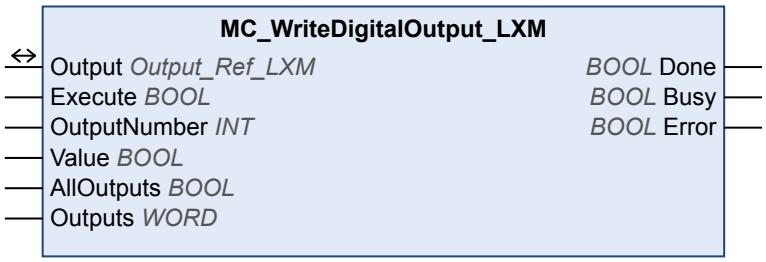
"2.6 Basic inputs and outputs"

Notes See the product manual for a description of the digital outputs.

2.10.4.4 MC_WriteDigitalOutput_LXM

Function description The function blocks writes values to the digital outputs.

Graphical representation



Compatible devices LXM32

0198441113892, V2.09, 04.2012

Inputs/outputs The table below shows the inputs.

Input	Data type	Description
`OutputNumber`	`INT`	Value range: 0 ... 2 Initial value: 0 Signal output to which to write. 0 = DQ0 1 = DQ1 2 = DQ2
`Value`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE FALSE: 0V is written to the selected signal output. TRUE: 24V is written to the selected signal output.
`AllOutputs`	`BOOL`	Value range: FALSE, TRUE Initial value: FALSE: The signal output to be written to is set via input `OutputNumber`. TRUE: The signal outputs to be written to are set via input `Output`.
`Outputs`	`WORD`	Value range: 0000_h ... 0007_h Initial value: 0 0: 0 V is written to the selected signal output. 1: 24V is written to the selected signal output. $0000\ 0000\ 0000\ 0001_2$ (0000_h) = Signal output 0 (DQ0) 24V $0000\ 0000\ 0000\ 0010_2$ (0002_h) = Signal output 1 (DQ1) 24V $0000\ 0000\ 0000\ 0100_2$ (0004_h) = Signal output 2 (DQ2) 24V Example: writing 24 V to all signal outputs: $0000\ 0000\ 0000\ 0111_2$ (0007_h)

"2.6 Basic inputs and outputs"

Notes Functions may already have been assigned to the outputs. This function block checks whether or not a function is already assigned to an output when the output is written to. An error is generated if a function is assigned to the output.

0198441113892, V2.09, 04.2012

2.10.5 Error handling

For error handling, each function block has an output `Error` which is set if a synchronous or asynchronous error occurs.

The function block MC_ReadAxisError_xxx is called to analyze the cause of the error. The function block contains the stored error information.

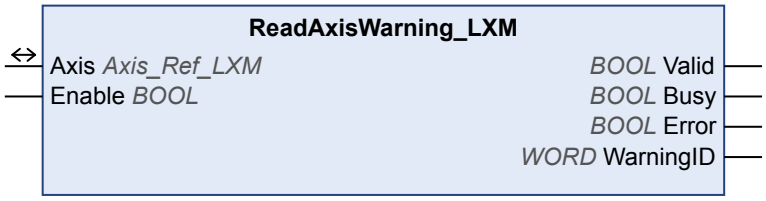
The function block MC_Reset_xxx deletes the error information entered. Future error information can now be stored.

If an additional error occurs, the error information is only stored if no stored error information already exists. If there is still information pertaining to a previous error, the new error message is ignored.

2.10.5.1 ReadAxisWarning_LXM

Function description The function block reads the most recent active warning.

Graphical representation



Compatible devices LXM32

Inputs/outputs The table below shows the outputs.

Output	Data type	Description
`WarningID`	`WORD`	Value range: Initial value: 0 Error number of the most recent warning. A warning is an error of error class 0. See the product manual for a description of the error numbers.

"2.6 Basic inputs and outputs"

Notes

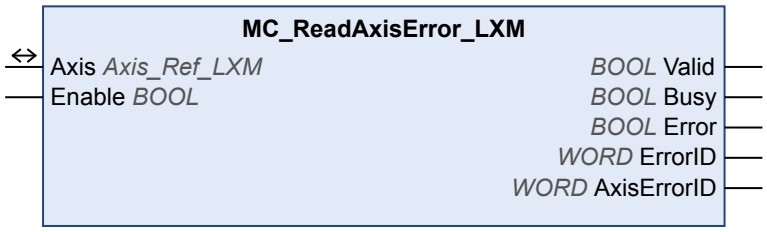
- The function block uses Service Data Objects (SDO) to read the parameter. Therefore, it is strongly recommended not to permanently set the input `Enable` to TRUE. This may cause overload on the fieldbus. It is recommended to deactivate the function block when the input `Busy` is set to FALSE.

2.10.5.2 MC_ReadAxisError_LXM

Function description The function block reads the error information pertaining to the most recent error.

0198441113892, V2.09, 04.2012

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs The table below shows the outputs.

Output	Data type	Description
`ErrorID`	`WORD`	Value range: 0000_h .... $FFFF_h$ Initial value: 0000_h 0: No error stored. >0: Stored error number. See the product manual for an overview of the error numbers.
`AxisErrorID`	`WORD`	Value range: Initial value: 0 LXM32 only: Bit 0 ... bit 7: Corresponds to the ID of the function block that has triggered the error. Bit 8 ... bit 15: Corresponds to the input that has triggered the error. The basic inputs such as `Axis`, `Input`, `Execute`, `Enable`, ... are not considered.

"2.6 Basic inputs and outputs"

Notes The function block uses Service Data Objects (SDO) to read the parameter from the device. Therefore, it is strongly recommended not to permanently set the input `Enable` to TRUE. This may cause overload on the fieldbus. It is recommended to deactivate the function block when the input `Busy` is set to FALSE.

Table of error numbers The table below shows the error numbers of the library. See the product manual for the error numbers of the drive.

0198441113892, V2.09, 04.2012

ErrorID hexadecimal	ErrorID decimal	Error class	Description
1100_h	4352		Parameter error
6108_h	24840		Function not available
8130_h	33072		Node Guarding error
$A309_h$	41737		Drive not in operating state**6** Operation Enabled
$A31B_h$	41755		"HALT" requested
$A338_h$	41784		The operating mode is not supported by this device.
$FF00_h$	65280	-	Toggle bit unchanged
$FF01_h$	65281	-	SDO timeout
$FF02_h$	65282	-	Server / client command specifier invalid or unknown
$FF06_h$	65286	-	No memory available
$FF07_h$	65287	-	Access to object not possible
$FF08_h$	65288	-	No read access, because write-only object (wo)
$FF09_h$	65289	-	No write access, because read object (ro)
$FF0A_h$	65290	-	Object does not exist in object dictionary
$FF0B_h$	65291	-	Object does not support PDO mapping
$FF0C_h$	65292	-	Number or length of objects exceed the byte length of the PDO
$FF0D_h$	65293	-	Parameters are incompatible
$FF0E_h$	65294	-	Device detects internal incompatibility
$FF0F_h$	65295	-	Hardware error, access denied
$FF10_h$	65296	-	Data type and parameter length do not match
$FF11_h$	65297	-	Data type does not match, parameter too long
$FF12_h$	65298	-	Data type does not match, parameter too short
$FF13_h$	65299	-	Subindex not supported
$FF14_h$	65300	-	Value range of parameter too large (relevant only for write access)
$FF15_h$	65301	-	Parameter values too great
$FF16_h$	65302	-	Parameter values too small
$FF17_h$	65303	-	Upper value is less than lower value
$FF18_h$	65304	-	General error
$FF19_h$	65305	-	Data can neither be transmitted to the application nor saved.
$FF1A_h$	65306	-	Local access channel is used, data can neither be transmitted nor saved.
$FF1B_h$	65307	-	Device status keeps data from being transmitted and saved.
$FF1C_h$	65308	-	Object dictionary does not exist or cannot be generated (for example, if data error occurs during generation from file)
$FF20_h$	65312	-	Unknown status
$FF21_h$	65313	-	Input variable was changed before response was received ("2.10.1.5 MC_ReadParameter_LXM", "2.10.2.1 MC_WriteParameter_LXM")
$FF22_h$	65314	-	Attempt to interrupt a non-interruptible function block ("2.7.1.1 MC_Power_LXM", "2.7.9.1 MC_Stop_LXM")

0198441113892, V2.09, 04.2012

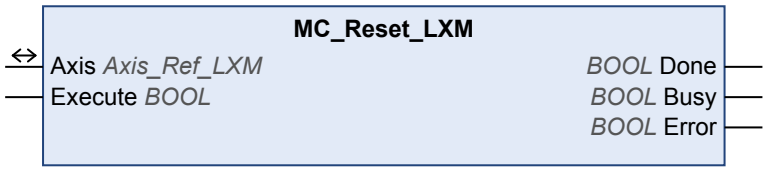
ErrorID hexadecimal	ErrorID decimal	Error class	Description
FF23h	65315	0	Trigger function is already active
FF24h	65316	-	PDO timeout
FF25h	65317	-	Operating mode Electronic Gear not active
FF27h	65319	-	Drive not in state "StandStill"
FF2Ah	65322	-	Trigger event lost
FF34h	65332	-	Power stage does not switch to operating state **6** Operation Enabled
FF35h	65333	-	Incorrect program number
FF36h	65334	-	Operating mode or function is not supported
FF38h	65336	-	Parameter list has not yet been read via "2.10.3.1 UploadDriveParameter_LXM".
FF39h	65337	-	Parameter list is not supported
FF3Bh	65339	-	Power stage is not in operating state **4** Ready To Switch On
FF3Dh	65341	-	Operating mode Motion Sequence is not active
FF3Eh	65342	-	No transition condition requested
FF3Fh	65343	-	Size of parameter list is not supported
FF50h	65360	-	Initialization error of function block
FF51h	65361	-	The function block cannot be controlled via the application since `i_iControlMode` = 1.
FF52h	65362	-	The function block cannot be controlled via the visualization since `i_iControlMode` = 0.
FF53h	65363	-	The value at the input `i_iControlMode` is outside of the valid value range.
FF54h	65364	-	The value at the input `iq_iCmd` is outside of the valid value range.
FF55h	65365	-	The function block and the connected device are incompatible.

2.10.5.3 MC_Reset_LXM

Function description

The function block is used to acknowledge an error. The error memory is cleared so that it is available for future error messages. If the power stage has been disabled by the automatic error response, it can be enabled again, provided that the cause of the error has been rectified when the error message is acknowledged.

Graphical representation



Compatible devices LXM05, SD3 and LXM32

Inputs/outputs "2.6 Basic inputs and outputs"

0198441113892, V2.09, 04.2012

2.11 Device Function

2.11.1 Startup

These function blocks "Startup" support you in commissioning a drive system at a controller. Before these function blocks can be used, you must set the communication parameters baud rate and node address in the drive and in the controller. Function blocks and the visualization cannot be used simultaneously.

The function blocks "Startup" with visualization elements have the following functions:

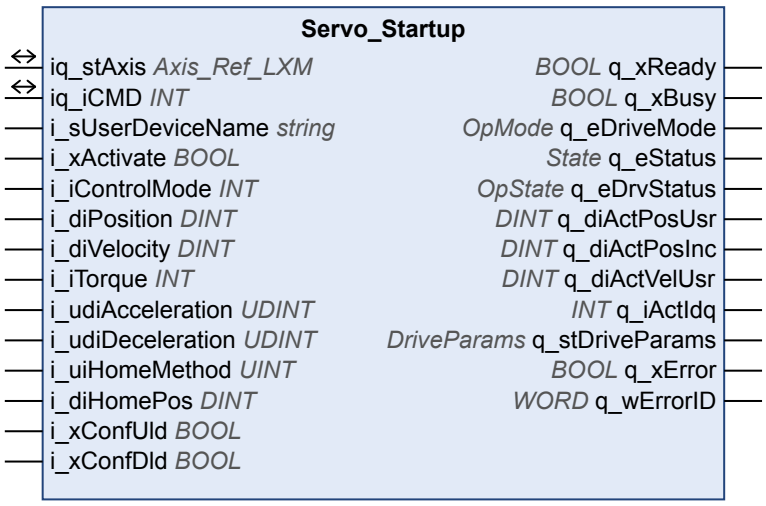
- Switching on the drive system.
- Displaying the status of the drive system.
- Fast access to frequently used parameters.
- The parameters are accessed via their index and subindex.
- Transmitting a device parameter list from the drive to the controller and from the controller to the drive (upload and download).
- Using the operating mode Homing.
- Using the operating mode Jog.
- Using the operating mode Profile Velocity (movement at defined velocity).
- Using the operating mode Profile Position (movement to defined position)
- Displaying and acknowledging error messages.

2.11.1.1 Servo_Startup

Function description

This function block supports you in commissioning a Lexium servo drive system for the first time. The function block comprises two visualizations to facilitate usage of the function block. Function blocks and the visualization cannot be used simultaneously.

Graphical representation



Compatible devices LXM05 and LXM32

Inputs/outputs The table below shows the inputs/outputs.

0198441113892, V2.09, 04.2012

Input/output	Data type	Description
iq_stAxis	Axis_Ref_LXM	Value range: Initial value: Axis structure
iq_iCMD	INT	Value range: Initial value: Commands: -1: command is active 0: no ongoing command 1: ENABLE (enable power stage) 2: DISABLE (disable power stage) 3: Reset 4: Stop 5: SetPos 6: Inc + 7: Inc – 8: MoveAbs 9: MoveVel 10: Homing 11: TorqueMode The function to be executed is written by the application as a command and overwritten by the function block when it is processed. Condition: The input is only effective if the value of ControlMode is 1. To start the selected function, the value in the parameter CMD must be written once. As soon as the command is executed, the value is overwritten by -1. When the execution of the command is terminated, the value is overwritten by 0.

The table below shows the inputs.

0198441113892, V2.09, 04.2012

Input	Data type	Description
i_sUserDeviceName	string	Value range: Initial value: Name of the axis. The name is defined by the user. If no name is entered, the node ID is displayed.
i_xActivate	BOOL	Value range: FALSE, TRUE Initial value: The selected ControlMode is activated with a rising edge. If all requirements for the selected ControlMode are met, the selected ControlMode is started. If the requirements are not met, the selection is canceled with an error message.
i_iControlMode	INT	Value range: Initial value: ControlMode = 0: The functions are controlled via the visualization. ControlMode = 1: The functions are controlled via the application. The visualization is deactivated.
i_diPosition	DINT	Value range: Initial value: Target position in [usr]
i_diVelocity	DINT	Value range: Initial value: Target velocity in [usr]
i_iTorque	INT	Value range: Initial value: Target torque The value corresponds to 0.1% of the nominal torque of the motor.
i_udiAcceleration	UDINT	Value range: Initial value: 600 Value for acceleration ramp in [usr].
i_udiDeceleration	UDINT	Value range: Initial value: 600 Value for deceleration ramp in [usr].
i_uiHomeMethod	UINT	Value range: Initial value: 1 Selected Homing method.
i_diHomePos	DINT	Value range: Initial value: Position of the reference point for position setting (Homing method).
i_xConfUld	BOOL	Value range: FALSE, TRUE Initial value: A rising edge triggers an upload (saving parameters from device to controller).
i_xConfDld	BOOL	Value range: FALSE, TRUE Initial value: A rising edge triggers a download (stored parameters from controller to device).

The table below shows the outputs.

0198441113892, V2.09, 04.2012

Output	Data type	Description
q_xReady	BOOL	Value range: FALSE, TRUE Initial value: Function block has been activated and is ready for operation.
q_xBusy	BOOL	Value range: FALSE, TRUE Initial value: A function is being performed via the function block. If a new function is started, the currently active function is canceled.
q_eDriveMode	OpMode	Value range: Initial value: Active operating mode (also see the state machine in the product manual): 0: No operating mode active 1: Operating mode Jog 2: Operating mode Homing 3: Operating mode Profile Position 4: Operating mode Profile Velocity 5: Operating mode Electronic Gear 6: Operating mode Current Control 7: Operating mode Speed Control 8: AutoTuning 9: Operating mode Profile Torque
q_eStatus	State	Value range: Initial value: State as per PLCopen state diagram: 0: Undefined 1: Errorstop 2: Disabled 3: Stopping 4: StandStill 5: DiscreteMotion 6: ContinuousMotion 7: SynchronizedMotion 8: Homing
q_eDrvStatus	OpState	Value range: Initial value: Operating state of the drive: 1: **1** Start 2: **2** Not Ready To Switch On 3: **3** Switch On Disabled 4: **4** Ready To Switch On 5: **5** Switched On 6: **6** Operation Enabled 7: **7** Quick Stop Active 8: **8** Fault Reaction Active
q_diActPosUsr	DINT	Value range: Initial value: Actual position in [usr]
q_diActPosInc	DINT	Value range: Initial value: Actual position in [inc]
q_diActVelUsr	DINT	Value range: Initial value: Actual velocity in [usr]

0198441113892, V2.09, 04.2012

Output	Data type	Description
q_iActIdq	INT	Value range: Initial value: Actual current in [A_{rms}]
q_stDriveParams	DriveParams	Value range: Initial value: Data structure, consisting of STRING: Device identification REAL: Firmware version of the device
q_xError	BOOL	Value range: FALSE, TRUE Initial value: FALSE: No error has been detected. TRUE: An error has been detected.
q_wErrorID	WORD	Value range: Initial value: Error number.

Notes

⚠ WARNING

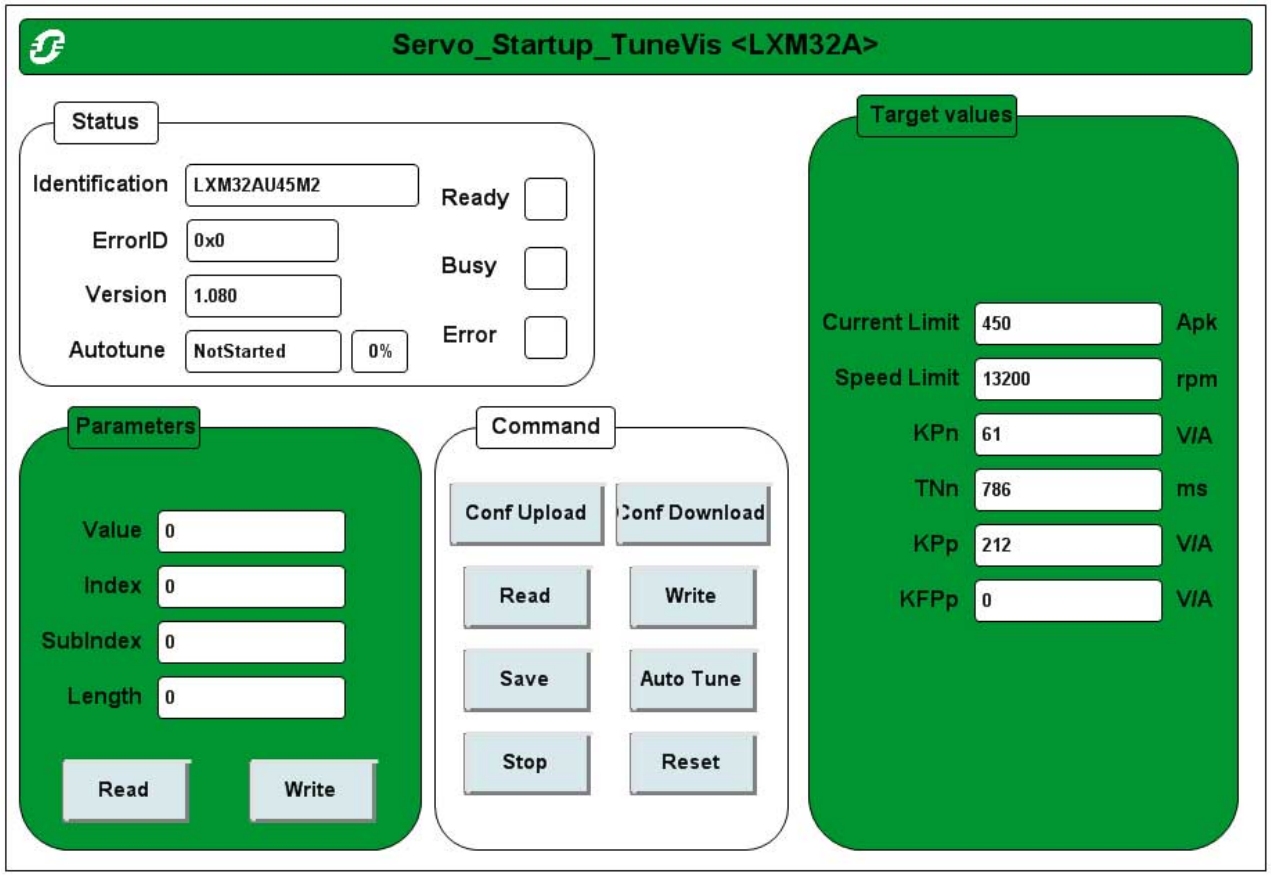
UNINTENDED BEHAVIOR DUE TO INCONSISTENT COMMANDS

If you have activated this function block, simultaneous use of other function blocks of the library leads to unintended behavior.

- Only activate this function block when all other function blocks of the library are inactive.
- Deactivate this function block before activating any other function block of the library.

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

0198441113892, V2.09, 04.2012



The visualization `Servo_Startup_TuneVis` provides direct access to many of the parameters of the drive. The parameters are addressed by means of index and subindex. Parameter values can be read and written. An image of defined parameter values can be saved from the drive to the controller. The stored parameter values can also be transferred from the controller to the drive as a single unit.

Operating mode Current Control and operating mode Profile Torque

The visualization `Servo_Startup_ManVis` allows you to start the operating mode Current Control and the operating mode Profile Torque. Observe the following safety instruction for these operating mode:

⚠ WARNING

EXCESSIVELY HIGH VELOCITY DUE TO INCORRECT LIMIT VALUE

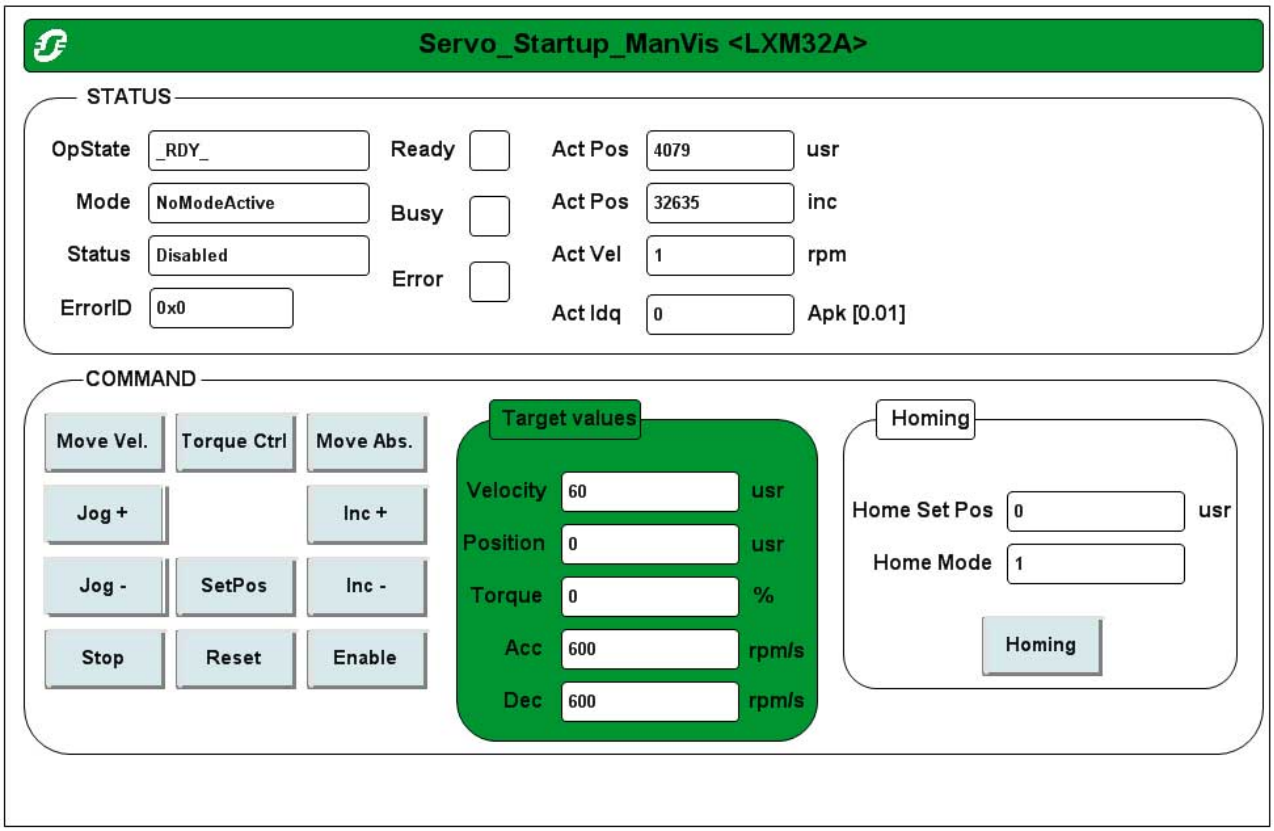
Without a proper limit value, the motor can reach a very high velocity in this operating mode.

- Check the parameterized velocity limitation.

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

Visualization `Servo_Startup_ManVis`

0198441113892, V2.09, 04.2012



The visualization `Servo_Startup_ManVis` allows you to display the status of the drive. It is possible to start and stop movements.

The following operating modes can be started:

- Operating mode Jog
- Operating mode Homing.
- Operating mode Profile Position (movement to defined position).
- Operating mode Profile Velocity (movement at defined velocity).
- Operating mode Electronic Gear.
- Operating mode Current Control.
- Operating mode Speed Control.
- Operating mode Profile Torque (LXM32 only).
- Autotuning is also possible.

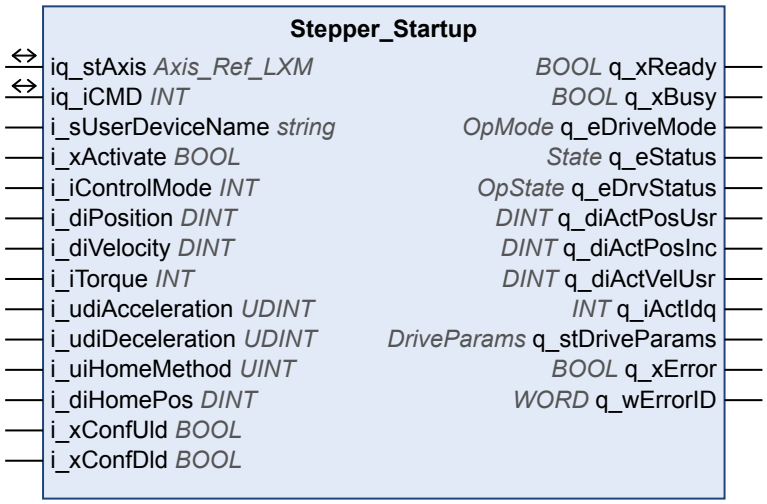
2.11.1.2 Stepper_Startup

Function description

This function block supports you in commissioning a Lexium drive system for stepper motors for the first time. The function block comprises two visualizations to facilitate usage of the function block. Function blocks and the visualization cannot be used simultaneously.

0198441113892, V2.09, 04.2012

Graphical representation



Compatible devices SD3

Inputs/outputs The table below shows the inputs/outputs.

Input/output	Data type	Description
`iq_stAxis`	`Axis_Ref_LXM`	Value range: Initial value: Axis structure
`iq_iCMD`	`INT`	Value range: Initial value: Commands: -1: command is active 0: no ongoing command 1: ENABLE (enable power stage) 2: DISABLE (disable power stage) 3: Reset 4: Stop 5: SetPos 6: Inc + 7: Inc – 8: MoveAbs 9: MoveVel 10: Homing The function to be executed is written by the application as a command and overwritten by the function block when it is processed. Condition: The input is only effective if the value of ControlMode is 1. To start the selected function, the value in the parameter CMD must be written once. As soon as the command is executed, the value is overwritten by -1. When the execution of the command is terminated, the value is overwritten by 0.

The table below shows the inputs.

0198441113892, V2.09, 04.2012

Input	Data type	Description
i_sUserDeviceName	string	Value range: Initial value: Name of the axis. The name is defined by the user. If no name is entered, the node ID is displayed.
i_xActivate	BOOL	Value range: FALSE, TRUE Initial value: The selected ControlMode is activated with a rising edge. If all requirements for the selected ControlMode are met, the selected ControlMode is started. If the requirements are not met, the selection is canceled with an error message.
i_iControlMode	INT	Value range: Initial value: ControlMode = 0: The functions are controlled via the visualization. ControlMode = 1: The functions are controlled via the application. The visualization is deactivated.
i_diPosition	DINT	Value range: Initial value: Target position in [usr]
i_diVelocity	DINT	Value range: Initial value: Target velocity in [usr]
i_iTorque	INT	Value range: Initial value: Target torque The value corresponds to 0.1% of the nominal torque of the motor.
i_udiAcceleration	UDINT	Value range: Initial value: 600 Value for acceleration ramp in [usr]
i_udiDeceleration	UDINT	Value range: Initial value: 600 Value for deceleration ramp in [usr]
i_uiHomeMethod	UINT	Value range: Initial value: 1 Selected Homing method.
i_diHomePos	DINT	Value range: Initial value: Position of the reference point for position setting (Homing method).
i_xConfUld	BOOL	Value range: FALSE, TRUE Initial value: A rising edge triggers an upload (saving parameters from device to controller).
i_xConfDld	BOOL	Value range: FALSE, TRUE Initial value: A rising edge triggers a download (stored parameters from controller to device).

The table below shows the outputs.

0198441113892, V2.09, 04.2012

Output	Data type	Description
q_xReady	BOOL	Value range: FALSE, TRUE Initial value: Function block has been activated and is ready for operation.
q_xBusy	BOOL	Value range: FALSE, TRUE Initial value: A function is being performed via the function block. If a new function is started, the currently active function is canceled.
q_eDriveMode	OpMode	Value range: Initial value: Active operating mode (also see the state machine in the product manual): 0: No operating mode active 1: Operating mode Jog 2: Operating mode Homing 3: Operating mode Profile Position 4: Operating mode Profile Velocity 5: Operating mode Electronic Gear 6: Operating mode Current Control 7: Operating mode Speed Control 8: AutoTuning 9: Operating mode Profile Torque
q_eStatus	State	Value range: Initial value: State as per PLCopen state diagram: 0: Undefined 1: Errorstop 2: Disabled 3: Stopping 4: StandStill 5: DiscreteMotion 6: ContinuousMotion 7: SynchronizedMotion 8: Homing
q_eDrvStatus	OpState	Value range: Initial value: Operating state of the drive: 1: **1** Start 2: **2** Not Ready To Switch On 3: **3** Switch On Disabled 4: **4** Ready To Switch On 5: **5** Switched On 6: **6** Operation Enabled 7: **7** Quick Stop Active 8: **8** Fault Reaction Active
q_diActPosUsr	DINT	Value range: Initial value: Actual position in [usr]
q_diActPosInc	DINT	Value range: Initial value: Actual position in [inc]
q_diActVelUsr	DINT	Value range: Initial value: Actual velocity in [usr]

0198441113892, V2.09, 04.2012

Output	Data type	Description
q_iActIdq	INT	Value range: Initial value: Actual current in [A_{rms}]
q_stDriveParams	DriveParams	Value range: Initial value: Data structure, consisting of: STRING: Device identification REAL: Firmware version of the device
q_xError	BOOL	Value range: FALSE, TRUE Initial value: FALSE: No error has been detected. TRUE: An error has been detected.
q_wErrorID	WORD	Value range: Initial value: Error number.

Notes

⚠ WARNING

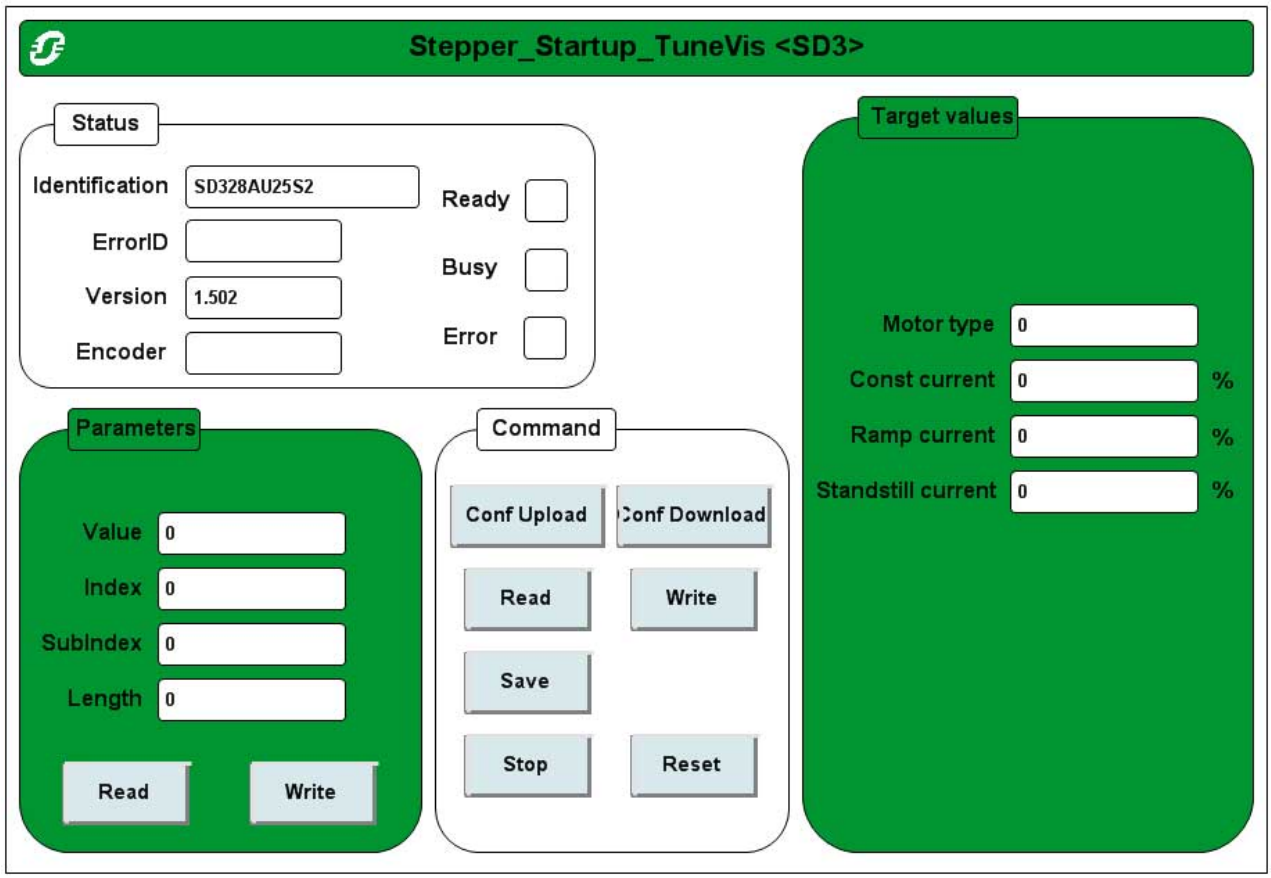
UNINTENDED BEHAVIOR DUE TO INCONSISTENT COMMANDS

If you have activated this function block, simultaneous use of other function blocks of the library leads to unintended behavior.

- Only activate this function block when all other function blocks of the library are inactive.
- Deactivate this function block before activating any other function block of the library.

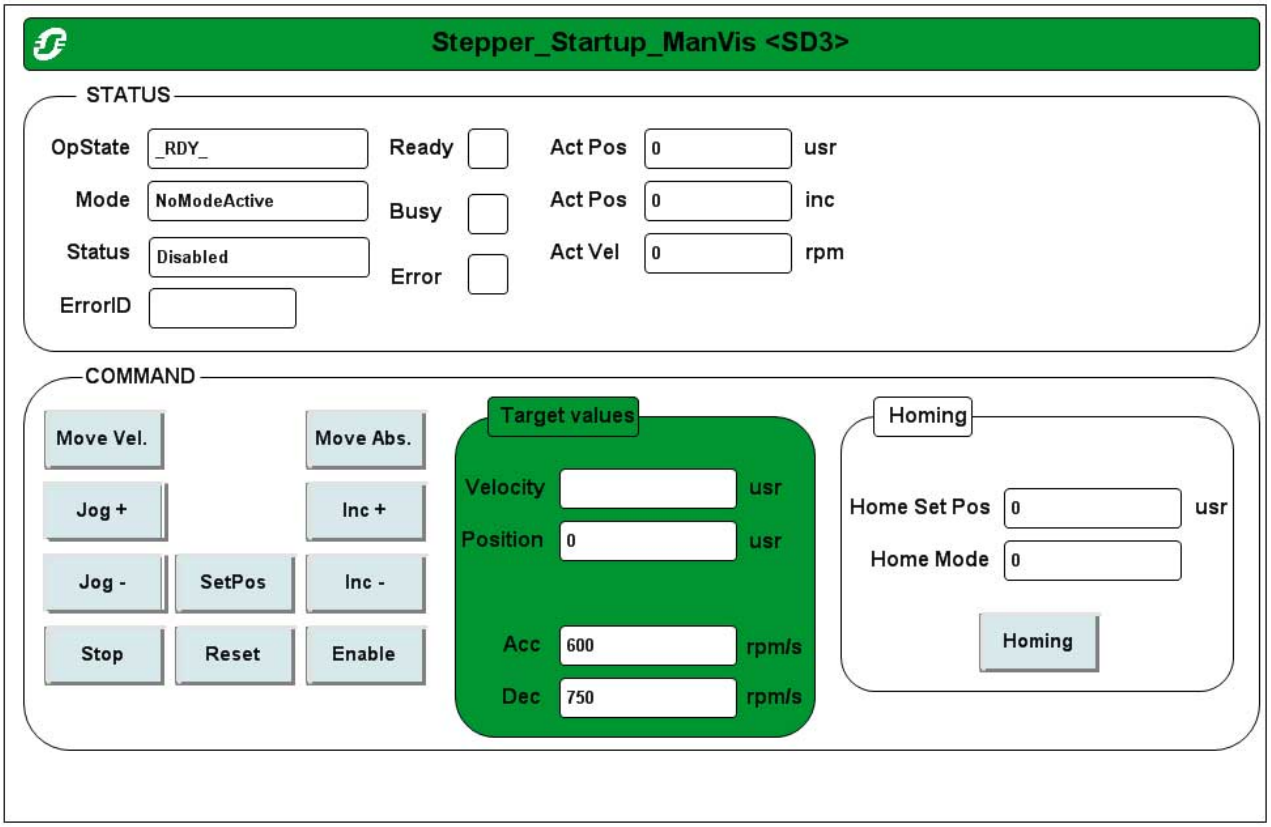
Failure to follow these instructions can result in death, serious injury or equipment damage.

0198441113892, V2.09, 04.2012



The visualization `Stepper_Startup_TuneVis` provides direct access to many of the parameters of the drive. The parameters are addressed by means of index and subindex. Parameter values can be read and written. An image of defined parameter values can be saved from the drive to the controller. The stored parameter values can also be transferred from the controller to the drive as a single unit.

0198441113892, V2.09, 04.2012



The visualization `Stepper_Startup_ManVis` allows you to display the status of the drive. It is possible to start and stop movements.

The following operating modes can be started:

- Operating mode Jog
- Operating mode Homing
- Operating mode Profile Position (movement to defined position)
- Operating mode Profile Velocity (movement at defined velocity)
- Operating mode Electronic Gear
- Operating mode Oscillator

0198441113892, V2.09, 04.2012

0198441113892, V2.09, 04.2012

3 Glossary

3

3.1 Units and conversion tables

The value in the specified unit (left column) is calculated for the desired unit (top row) with the formula (in the field).

Example: conversion of 5 meters [m] to yards [yd]
5 m / 0.9144 = 5.468 yd

3.1.1 Length

	in	ft	yd	m	cm	mm
in	-	/ 12	/ 36	* 0.0254	* 2.54	* 25.4
ft	* 12	-	/ 3	* 0.30479	* 30.479	* 304.79
yd	* 36	* 3	-	* 0.9144	* 91.44	* 914.4
m	/ 0.0254	/ 0.30479	/ 0.9144	-	* 100	* 1000
cm	/ 2.54	/ 30.479	/ 91.44	/ 100	-	* 10
mm	/ 25.4	/ 304.79	/ 914.4	/ 1000	/ 10	-

3.1.2 Mass

	lb	oz	slug	kg	g
lb	-	* 16	* 0.03108095	* 0.4535924	* 453.5924
oz	/ 16	-	* $1.942559*10^{-3}$	* 0.02834952	* 28.34952
slug	/ 0.03108095	/ $1.942559*10^{-3}$	-	* 14.5939	* 14593.9
kg	/ 0.45359237	/ 0.02834952	/ 14.5939	-	* 1000
g	/ 453.59237	/ 28.34952	/ 14593.9	/ 1000	-

3.1.3 Force

	lb	oz	p	N
lb	-	* 16	* 453.55358	* 4.448222
oz	/ 16	-	* 28.349524	* 0.27801
p	/ 453.55358	/ 28.349524	-	* $9.807*10^{-3}$
N	/ 4.448222	/ 0.27801	/ $9.807*10^{-3}$	-

3.1.4 Power

	HP	W
HP	-	* 746
W	/ 746	-

0198441113892, V2.09, 04.2012

3.1.5 Rotation

	min^{-1} (RPM)	**rad/s**	**deg./s**
min^{-1} (RPM)	-	* π / 30	* 6
rad/s	* 30 / π	-	* 57.295
deg./s	/ 6	/ 57.295	-

3.1.6 Torque

	lb·in	**lb·ft**	**oz·in**	**Nm**	**kp·m**	**kp·cm**	**dyne·cm**
lb·in	-	/ 12	* 16	* 0.112985	* 0.011521	* 1.1521	* $1.129*10^{6}$
lb·ft	* 12	-	* 192	* 1.355822	* 0.138255	* 13.8255	* $13.558*10^{6}$
oz·in	/ 16	/ 192	-	* $7.0616*10^{-3}$	* $720.07*10^{-6}$	* $72.007*10^{-3}$	* 70615.5
Nm	/ 0.112985	/ 1.355822	/ $7.0616*10^{-3}$	-	* 0.101972	* 10.1972	* $10*10^{6}$
kp·m	/ 0.011521	/ 0.138255	/ $720.07*10^{-6}$	/ 0.101972	-	* 100	* $98.066*10^{6}$
kp·cm	/ 1.1521	/ 13.8255	/ $72.007*10^{-3}$	/ 10.1972	/ 100	-	* $0.9806*10^{6}$
dyne·cm	/ $1.129*10^{6}$	/ $13.558*10^{6}$	/ 70615.5	/ $10*10^{6}$	/ $98.066*10^{6}$	/ $0.9806*10^{6}$	-

3.1.7 Moment of inertia

	lb·in^2	**lb·ft^2**	**kg·m^2**	**kg·cm^2**	**kp·cm·s^2**	**oz·in^2**
lb·in^2	-	/ 144	/ 3417.16	/ 0.341716	/ 335.109	* 16
lb·ft^2	* 144	-	* 0.04214	* 421.4	* 0.429711	* 2304
kg·m^2	* 3417.16	/ 0.04214	-	* $10*10^{3}$	* 10.1972	* 54674
kg·cm^2	* 0.341716	/ 421.4	/ $10*10^{3}$	-	/ 980.665	* 5.46
kp·cm·s^2	* 335.109	/ 0.429711	/ 10.1972	* 980.665	-	* 5361.74
oz·in^2	/ 16	/ 2304	/ 54674	/ 5.46	/ 5361.74	-

3.1.8 Temperature

	°F	**°C**	**K**
°F	-	(°F - 32) * 5/9	(°F - 32) * 5/9 + 273.15
°C	°C * 9/5 + 32	-	°C + 273.15
K	(K - 273.15) * 9/5 + 32	K - 273.15	-

3.1.9 Conductor cross section

AWG	1	2	3	4	5	6	7	8	9	10	11	12	13
mm^2	42.4	33.6	26.7	21.2	16.8	13.3	10.5	8.4	6.6	5.3	4.2	3.3	2.6

AWG	14	15	16	17	18	19	20	21	22	23	24	25	26
mm^2	2.1	1.7	1.3	1.0	0.82	0.65	0.52	0.41	0.33	0.26	0.20	0.16	0.13

0198441113892, V2.09, 04.2012

3.2 Terms and Abbreviations

	See chapter "1.5 Standards and terminology" for information on the pertinent standards on which many terms are based. Some terms and abbreviations may have specific meanings with regard to the standards.
Asynchronous error	Asynchronous errors are signaled without a request. Example of an asynchronous error: Power stage overtemperature.
Device data	The term device data refers to the parameter values of a device. The data is stored in the EEPROM of the device (persistent memory).
Error	Discrepancy between a detected (computed, measured or signaled) value or condition and the specified or theoretically correct value or condition.
Error class	Classification of errors into groups. The different error classes allow for specific responses to errors, for example by severity.
Factory setting	Factory settings when the product is shipped
Fatal error	In the case of fatal error, the product is no longer able to control the motor so that the power stage must be immediately disabled.
Fault	Fault is a state that can be caused by an error. Further information can be found in the pertinent standards such as IEC 61800-7, ODVA Common Industrial Protocol (CIP).
Fault reset	A function used to restore the drive to an operational state after a detected error is cleared by removing the cause of the error so that the error is no longer active.
LED	Light Emitting Diode
Limit switch	Switches that signal overtravel of the permissible range of travel.
Node guarding	Monitoring of the connection to the slave at an interface for cyclic data traffic.
Parameter	Device data and values that can be read and set (to a certain extent) by the user.
Power stage	The power stage controls the motor. The power stage generates current for controlling the motor on the basis of the positioning signals from the controller.
RS485	Fieldbus interface as per EIA-485 which enables serial data transmission with multiple devices.
Synchronous error	Error signaled by the controller if it is unable to execute a command received from the master.
Warning	If the term is used outside the context of safety instructions, a warning alerts to a potential problem that was detected by a monitoring function. A warning does not cause a transition of the operating state.

0198441113892, V2.09, 04.2012

0198441113892, V2.09, 04.2012

4 Index

4

0198441113892, V2.09, 04.2012

0198441113892, V2.09, 04.2012