

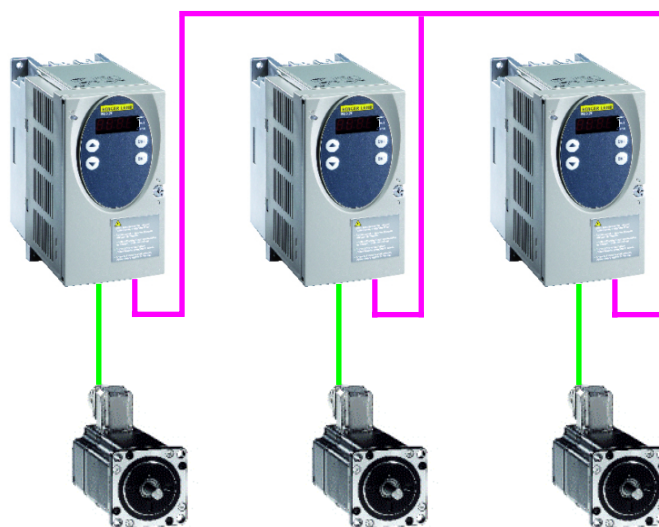
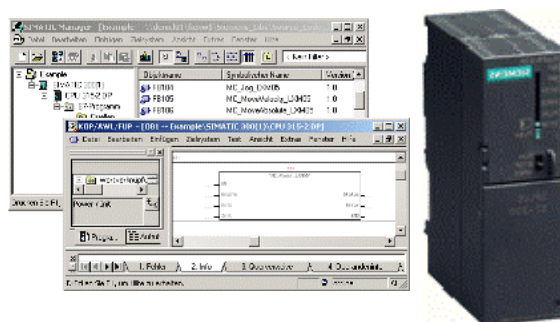
## Technical documentation

---

Manual

**Step 7 library for  
Lexium 05B, SD 328B**

Edition: V1.01, 07.2007



Berger Lahr GmbH & Co. KG  
Breslauer Str. 7  
D-77933 Lahr

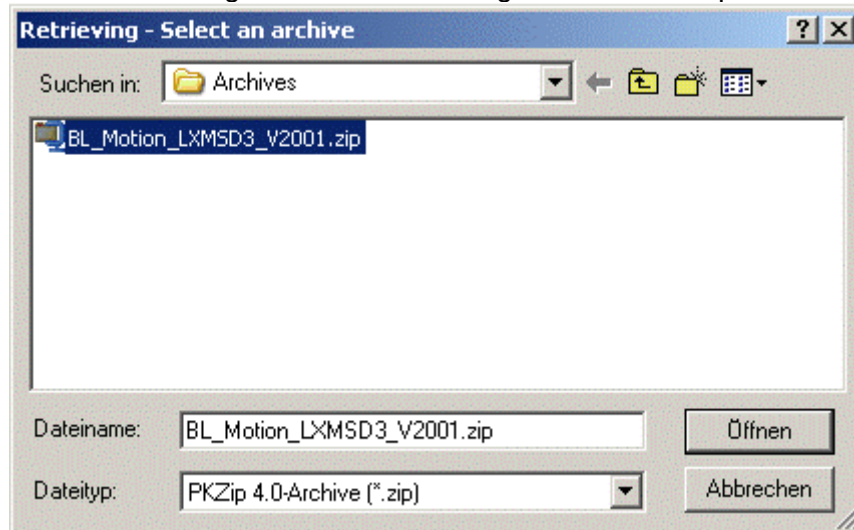
## Contents

1	Extracting the library .....	3
2	Starting a new project .....	5
3	Configuring the hardware .....	6
4	Installing the GSD .....	7
5	Linking the drive into the PB network .....	9
6	Assigning the I/O addresses .....	12
6.1	Parameter data channel .....	12
6.2	Process data channel .....	13
7	Description of the library blocks .....	14
7.1	Copying the axis structure into the project .....	14
7.2	Creating an axis reference .....	15
8	Library blocks .....	17
8.1	Basic calling procedures .....	17
8.2	Explanation of common parameters .....	18
8.3	Phasing diagrams .....	19
8.4	Initialisation .....	20
8.5	Jog .....	23
8.6	Homing .....	25
8.7	Current control .....	28
8.8	Speed control .....	29
8.9	Profile position mode .....	30
8.10	Velocity mode .....	34
8.11	Electronic gear .....	35
8.12	Stopping .....	38
8.13	Fast position capture .....	39
8.14	Read parameter .....	41
8.15	Write parameter .....	53
8.16	Inputs/outputs .....	58
8.17	Error handling .....	60
9	Glossary .....	62
10	List of error numbers .....	64
11	Parameter list for Up- and Download function .....	68

# 1 Extracting the library

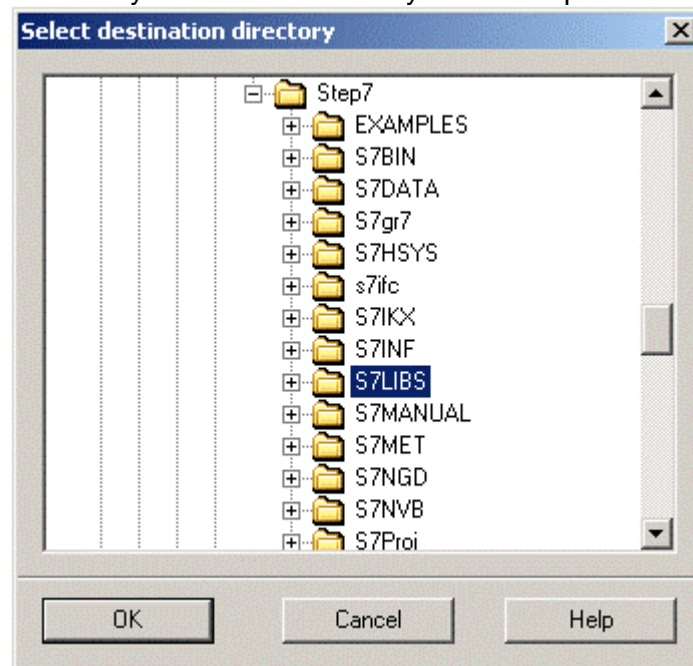
In order to use the library blocks, you must first unpack the archive "BL\_Motion\_LXMSD3\_Vxxxx.zip" with the Step7 software.  
This is done with the menu item **Retrieve** in the menu **File**.

The following window for selecting the archive is opened:



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

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

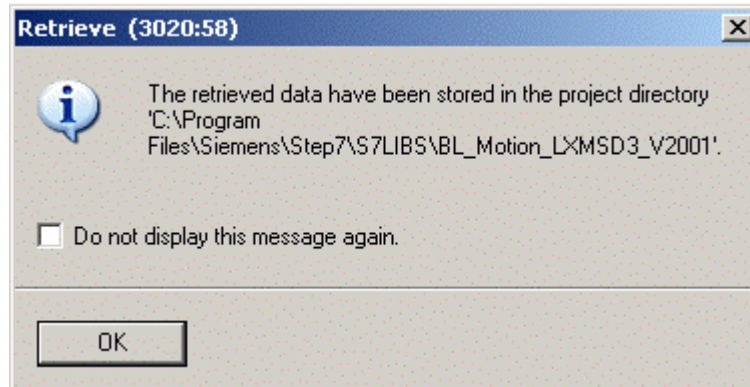


Mark the required directory, and confirm your selection with "OK".

Recommendation: <Siemens directory>\Step7\S7TMP  
Example: C:\Programs\Siemens\Step7\S7LIBS

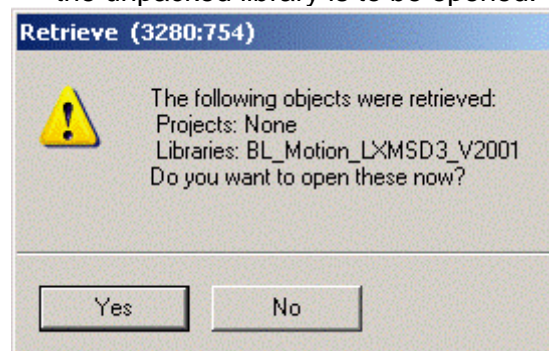
## Step 7 Motion Library for Lexium 05B and SD 328B

Depending on the configuration of your Step7 software, the successful unpacking procedure will be confirmed.



Confirm with "OK".

In a further confirmation window, you are asked whether the unpacked library is to be opened.



Deny the request with "No".

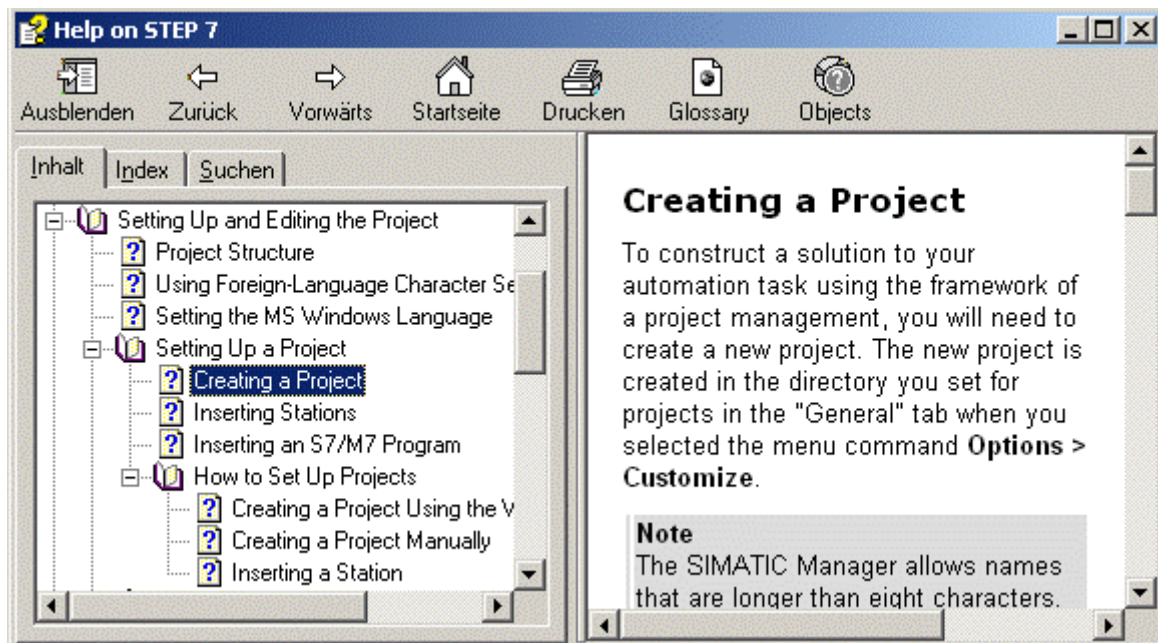
**Note:** Of course, you can open the library, and manually copy the relevant blocks into your application by means of the copying function of the Step 7 software.

You have now successfully unpacked the library, and can therefore access the blocks with the Step7 editors in order to use them in your application as described below.

For the library's function, it is essential that you use the associated Device Master File of the relevant drive (GSD **BLS70977.GSD** for LXM05, and GSD **BLS70A19.GSD** for SD328). But first, the corresponding GSD must be installed so that it is available in the Hardware Manager. For this purpose, you must start a new project and start the Hardware Manager.

## 2 Starting a new project

Create a new project. For this, you open the menu **File**, and select the menu item **New** or **Assistant 'New project'**. Hereby, it is assumed that you know how to create a new project, so that reference is made here to the online Help and to the documentation of Step7 and Siemens.

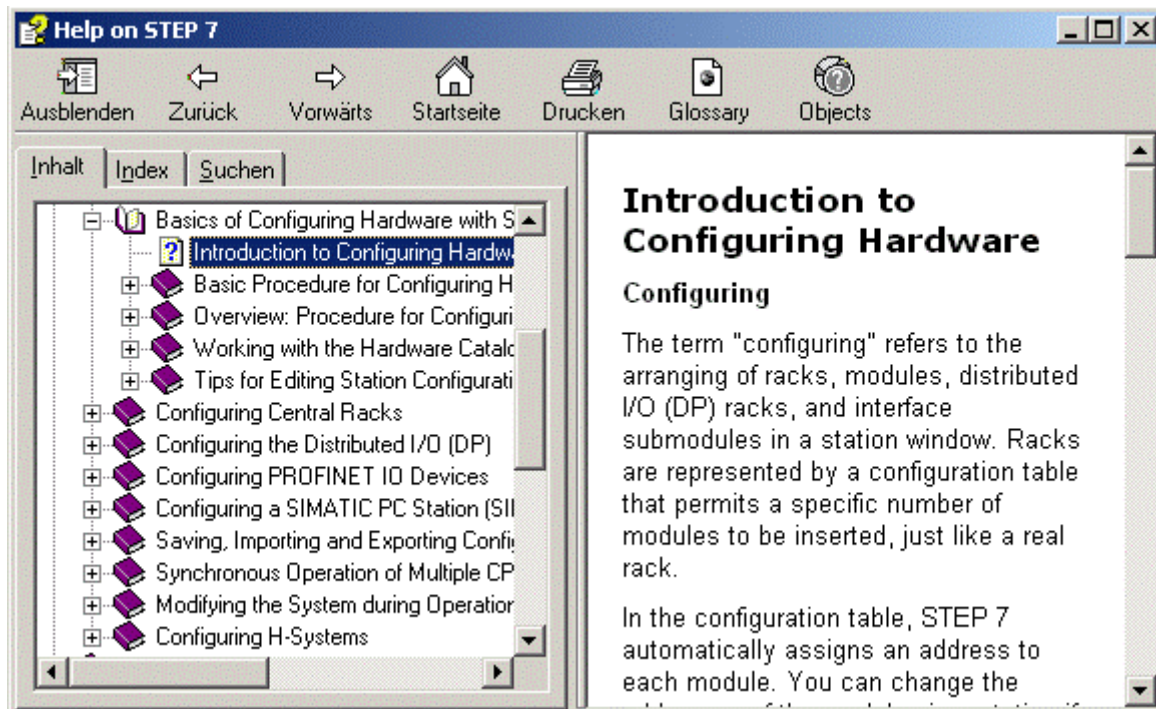




### 3 Configuring the hardware

When you have created a new project, you must define the hardware that is to be used. For this, you select the menu **Insert**, and insert a station by means of the menu item **Station**. Subsequently, you mark the inserted station, and start the hardware configurator via the menu **Edit** and the sub-menu item **Open Object**.

Hereby, it is assumed that you know how to configure the hardware, so that reference is made here to the online Help and to the documentation of Step7 and Siemens.



In order to link the drive into the Profibus network, you must first install the GSD associated to the corresponding drive, as described in the following section. If this has already been done, you can proceed with the Chapter [Linking the drive into the PB network](#).

## 4 Installing the GSD

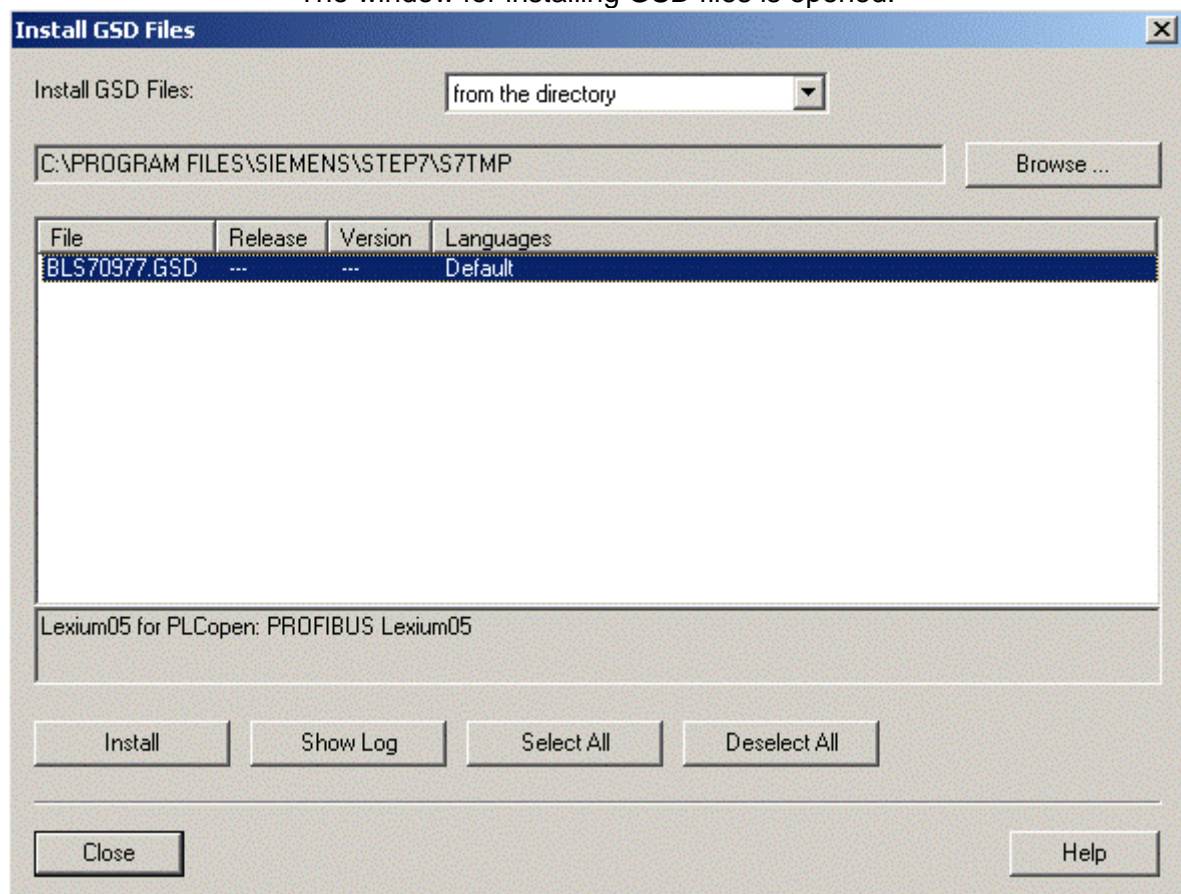
**Note:** The library may only be used with the associated GSD (data master file)  
**BLS70977.GSD** for LXM05 and **BLS70A19.GSD** for SD328.

**The library will not work with the standard GSD.**

Copy the GSD into any directory on your hard disk.  
 Recommendation: <Siemens directory>\Step7\S7TMP  
 Example: C:\Programs\Siemens\Step7\S7LIBS

Next, you open the dialogue box in the hardware configurator for installing GSD files. This is done via the menu **Extras** and the menu item **Installing GSD files....**

The window for installing GSD files is opened.

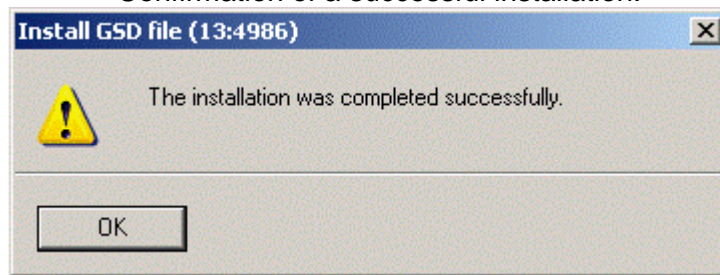


Browse to the directory with the GSD, and mark it. Confirm your selection with "Install".

**Step 7 Motion Library for Lexium 05B and SD 328B**

---

Confirmation of a successful installation.



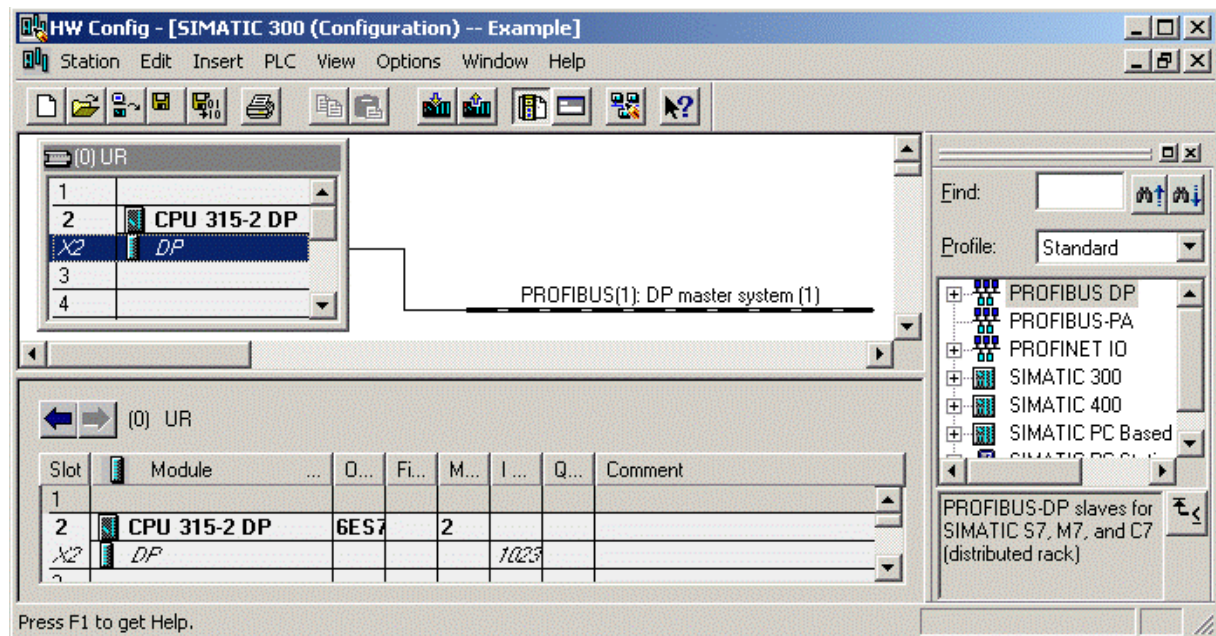
Close the confirmation message with "OK".

Now also close the window for installing GSDs with "Close", which returns you to the hardware configurator for the remaining hardware installation steps.

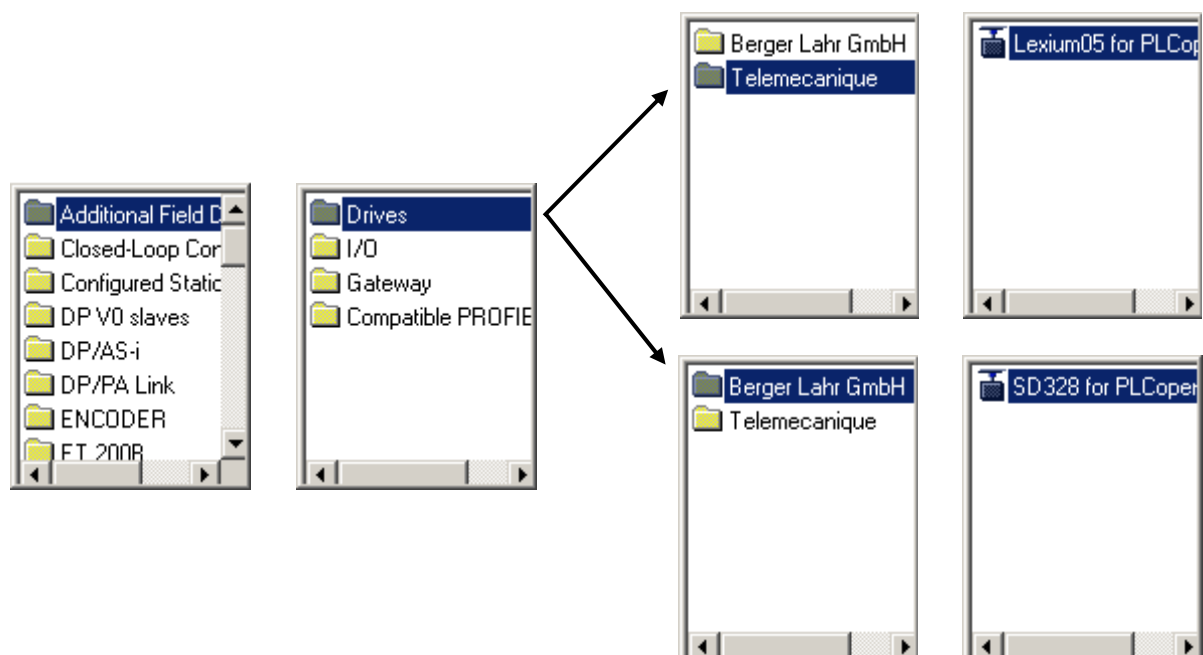


## 5 Linking the drive into the PB network

A prerequisite for linking the drives into the network is that you have included a module rack, a CPU, and a DP master system in your hardware configuration.

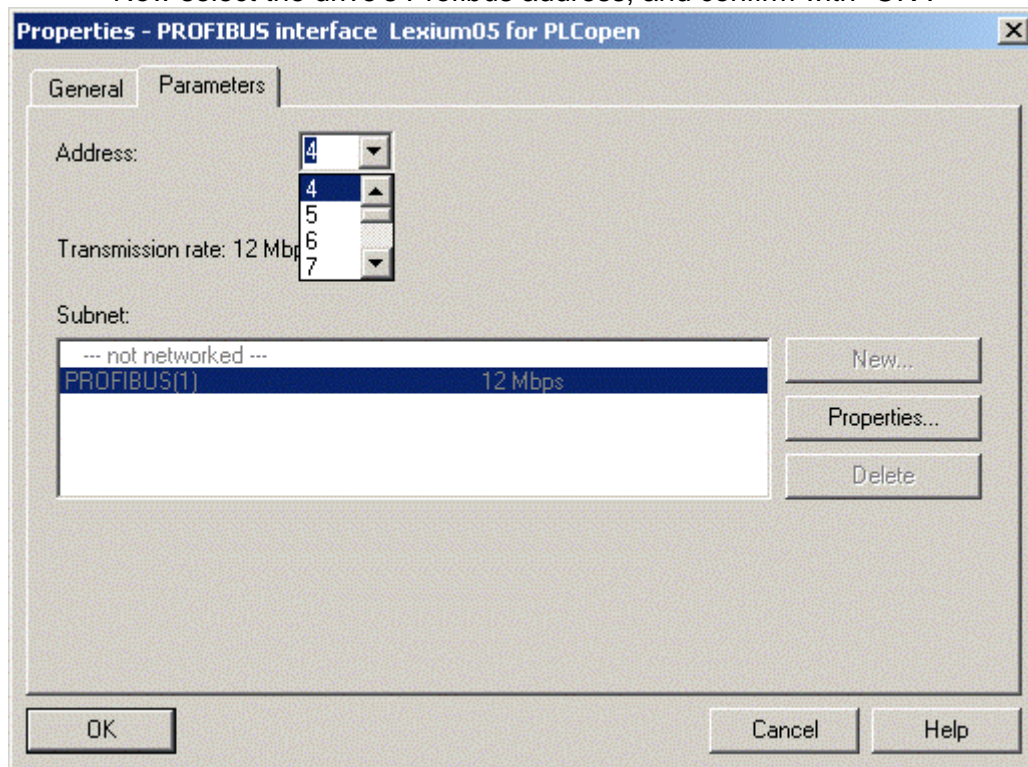


Now mark the master system in the hardware configurator, and via the menu **Insert/Insert Object...** you select the item **ICIA for PLCopen** after clicking through the sub-menus **Additional Field Devices**, **Drives**, and **ICIA**. For **Lexium05** you select the entry **Lexium05 for PLCopen** in the menu **Telemecanique**, and for **SD328** you select the entry **SD328 for PLCopen** in the menu **Berger Lahr GmbH**.

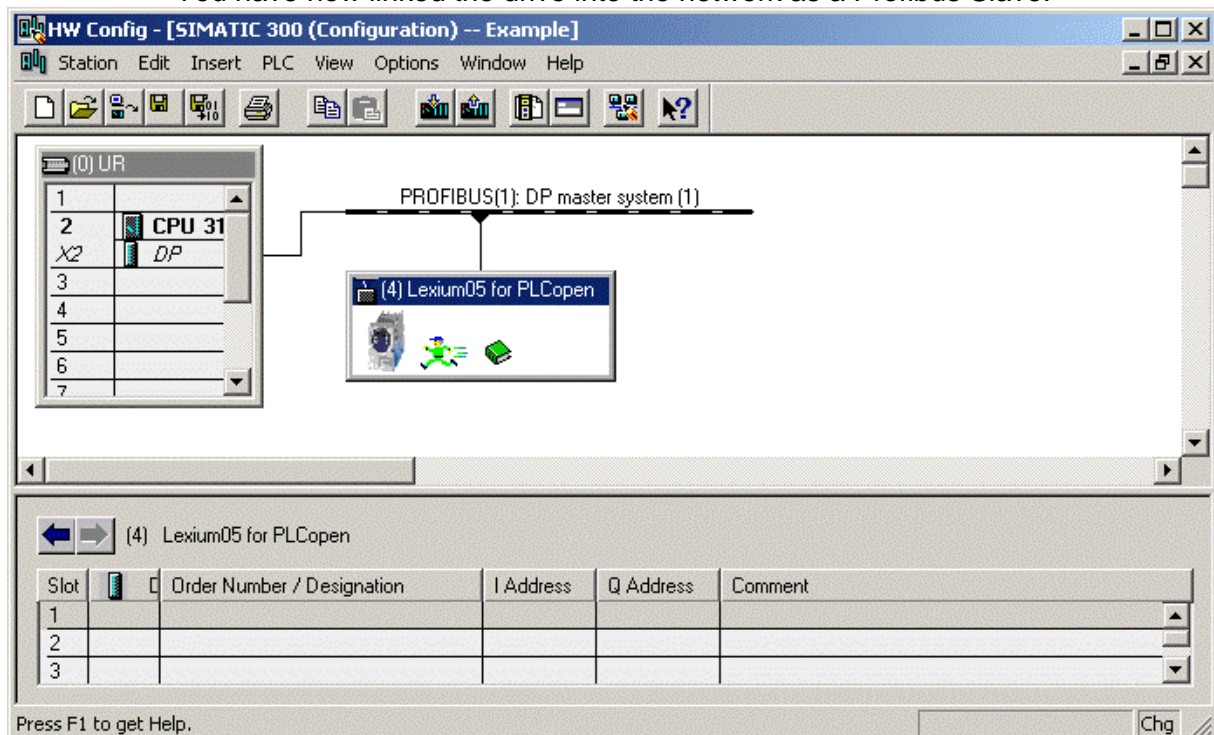


## Step 7 Motion Library for Lexium 05B and SD 328B

Now select the drive's Profibus address, and confirm with "OK".

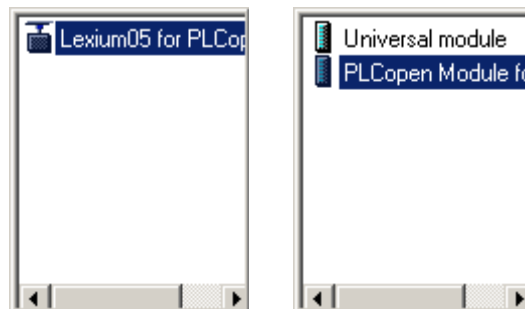


You have now linked the drive into the network as a Profibus Slave.

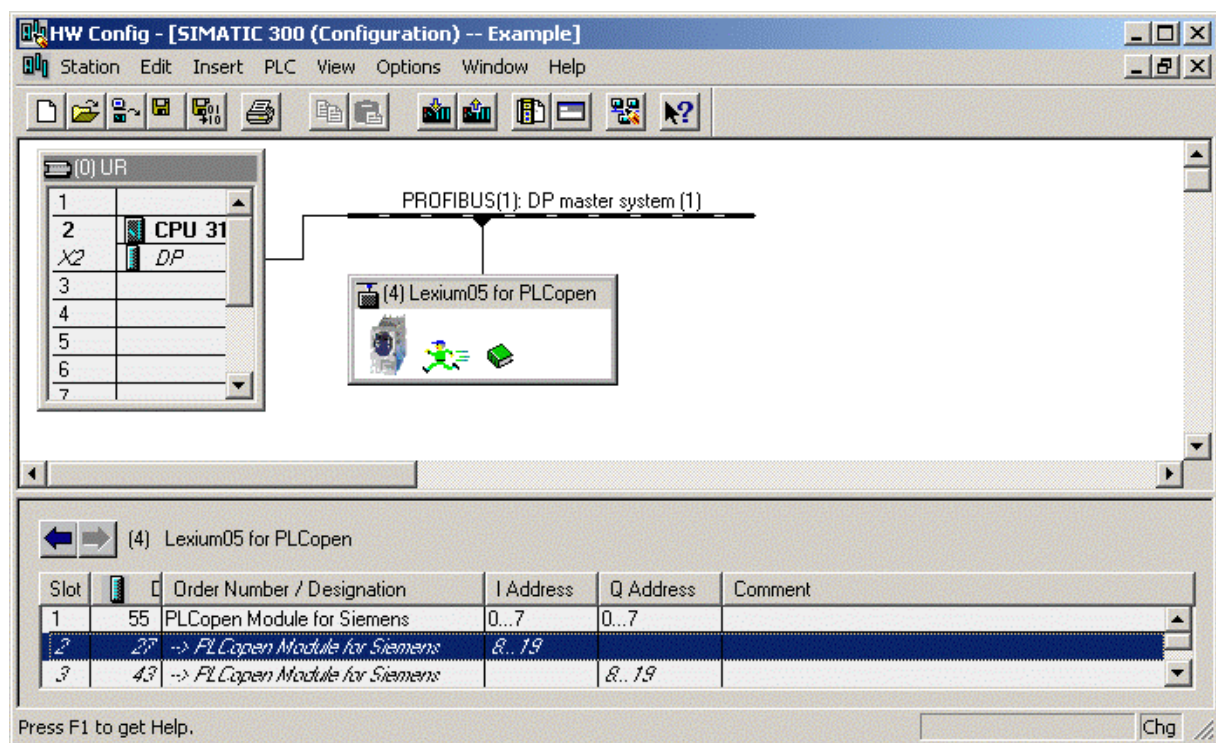


## Step 7 Motion Library for Lexium 05B and SD 328B

Next, you must insert the communication block, in order to define the input and output addresses for the parameter and process data channels. For this, you mark Slot 1 of the PB slave, and select the block **PLCopen Block for Siemens** in the menu item **Lexium05** of the menu **Insert/Insert Object...**



The drive is now linked into the DP master system. The library uses two communication channels for communication with the drive:  
The parameter data channel (8 bytes) in Slot 1, and the process data channel (12 bytes) in Slots 2 and 3.



Slot	Order Number / Designation	I Address	Q Address	Comment
1	55 PLCopen Module for Siemens	0...7	0...7	
2	27 -> PLCopen Module for Siemens	8...19		
3	43 -> PLCopen Module for Siemens		8...19	

Finally, the I/O addresses for the communication channels must be defined, as described in the next Chapter.



## 6 Assigning the I/O addresses

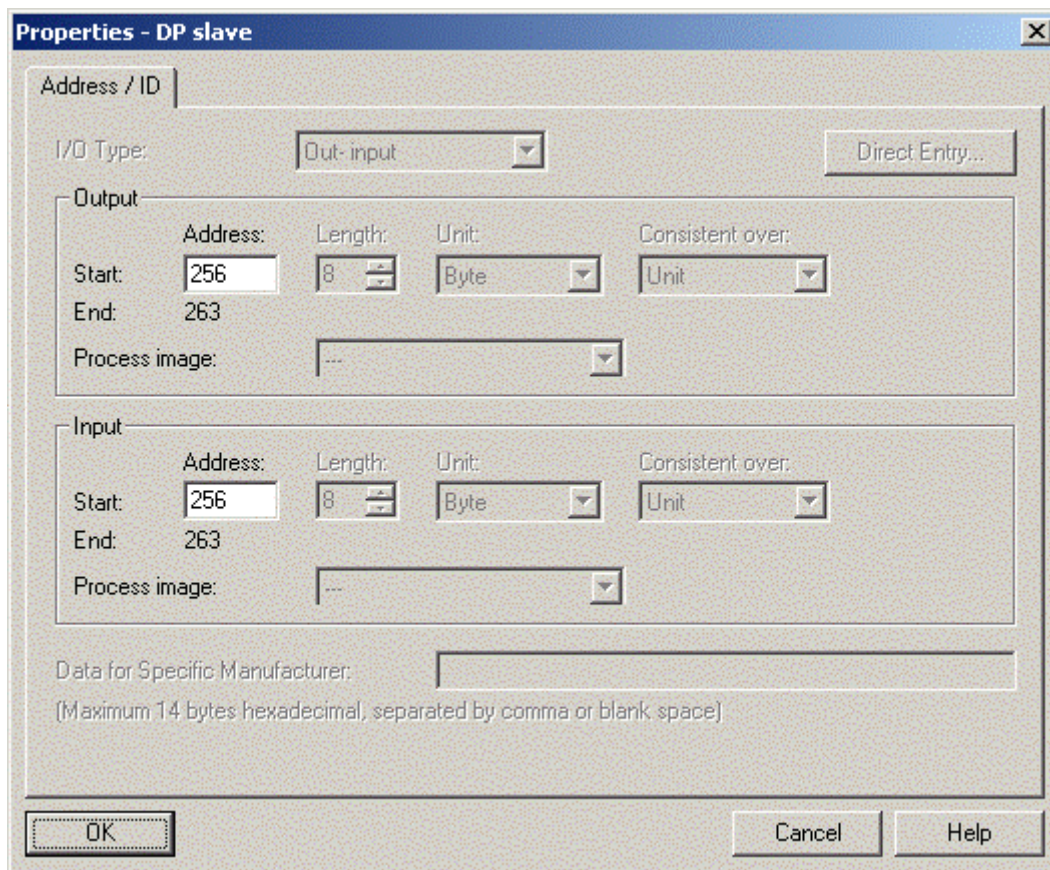
The last adjustment in the hardware configurator involves assigning the input and output addresses of the communication channels. These addresses depend on the projected CPU and on the PLC's configuration.

### 6.1 Parameter data channel

In order to assign the addresses for the parameter data channel, you must mark Slot 1, and select the menu item **Object Properties...** in the menu **Edit**.

Now assign a free address space for output and input data respectively, as described below.

**Note:** These addresses must be made known to the library.  
The starting address of the output range must be transferred to the function [MC\\_Init\\_LXM05](#) at the input "AdrParameterOut", and the starting address of the input range at the input "AdrParameterIn".



**Properties - DP slave**

Address / ID

I/O Type: Out-input Direct Entry...

**Output**

	Address:	Length:	Unit:	Consistent over:
Start:	256	8	Byte	Unit
End:	263			
Process image:	---			

**Input**

	Address:	Length:	Unit:	Consistent over:
Start:	256	8	Byte	Unit
End:	263			
Process image:	---			

Data for Specific Manufacturer:

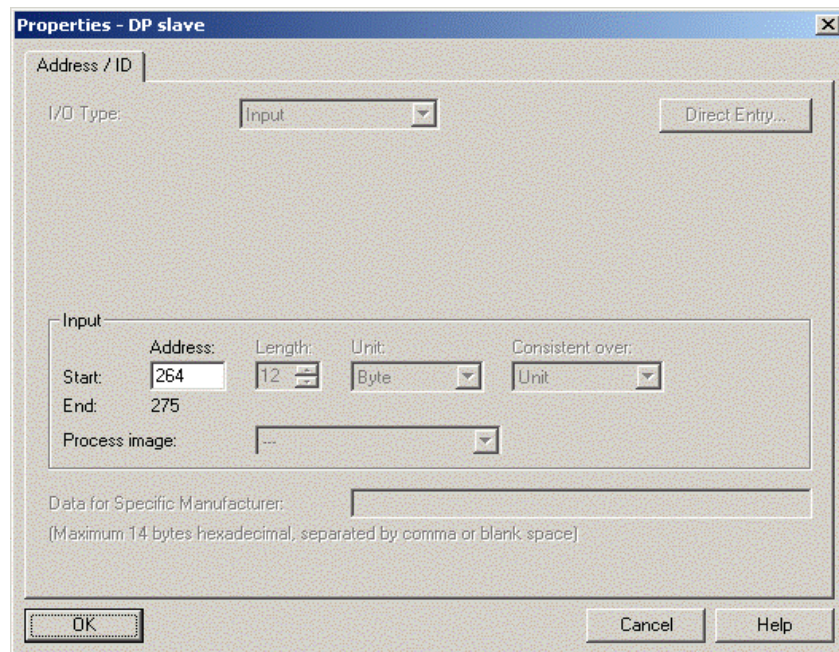
(Maximum 14 bytes hexadecimal, separated by comma or blank space)

OK Cancel Help

## 6.2 Process data channel

The addresses of the process data channel are assigned in the same way as the addresses of the parameter data channel. The only difference is that the input and output ranges are assigned to other slots.

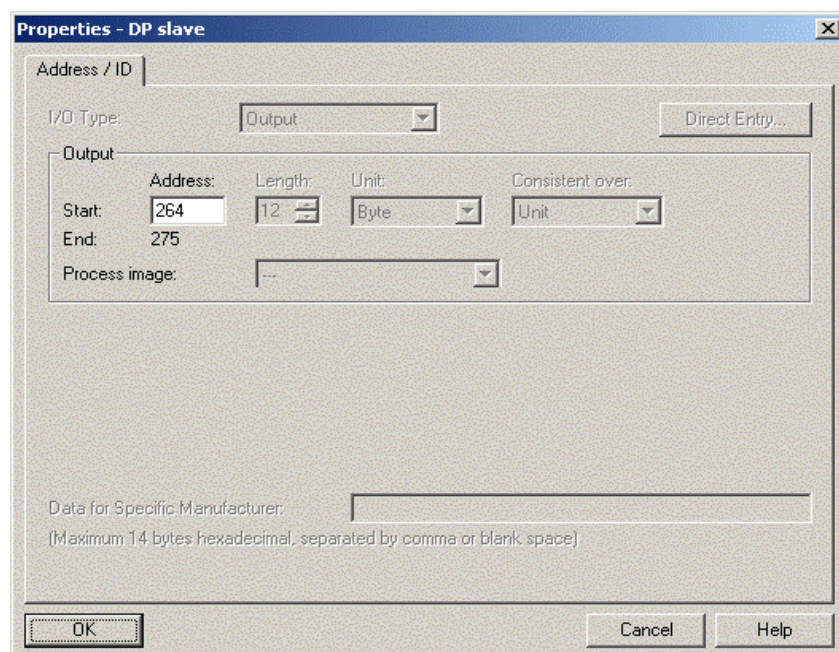
**Note:** These addresses must be made known to the library. The starting address of the output range must be transferred to the function [MC\\_Init\\_LXM05](#) at the input “AdrProcessdataOut”, and the starting address of the input range at the input “AdrProcessdataIn”.



The dialog box titled "Properties - DP slave" shows the "Input" configuration. The "I/O Type" is set to "Input". The "Address / ID" field is empty. The "Direct Entry..." button is visible. The "Input" section contains the following fields:

	Address:	Length:	Unit:	Consistent over:
Start:	264	12	Byte	Unit
End:	275			
Process image:	---			

Below the input section is a text field for "Data for Specific Manufacturer:" with a note: "(Maximum 14 bytes hexadecimal, separated by comma or blank space)". At the bottom are "OK", "Cancel", and "Help" buttons.



The dialog box titled "Properties - DP slave" shows the "Output" configuration. The "I/O Type" is set to "Output". The "Address / ID" field is empty. The "Direct Entry..." button is visible. The "Output" section contains the following fields:

	Address:	Length:	Unit:	Consistent over:
Start:	264	12	Byte	Unit
End:	275			
Process image:	---			

Below the output section is a text field for "Data for Specific Manufacturer:" with a note: "(Maximum 14 bytes hexadecimal, separated by comma or blank space)". At the bottom are "OK", "Cancel", and "Help" buttons.

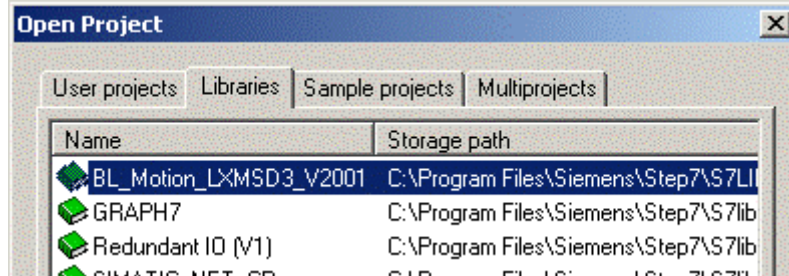
To conclude the configuration, save and compile the settings by means of the menu item **Save and Compile** in the menu **Station** of the hardware configurator.



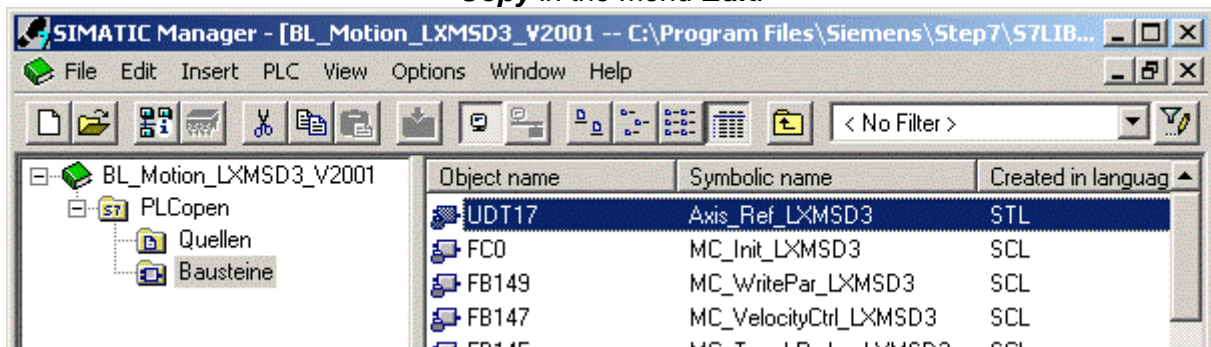
## 7 Description of the library blocks

### 7.1 Copying the axis structure into the project

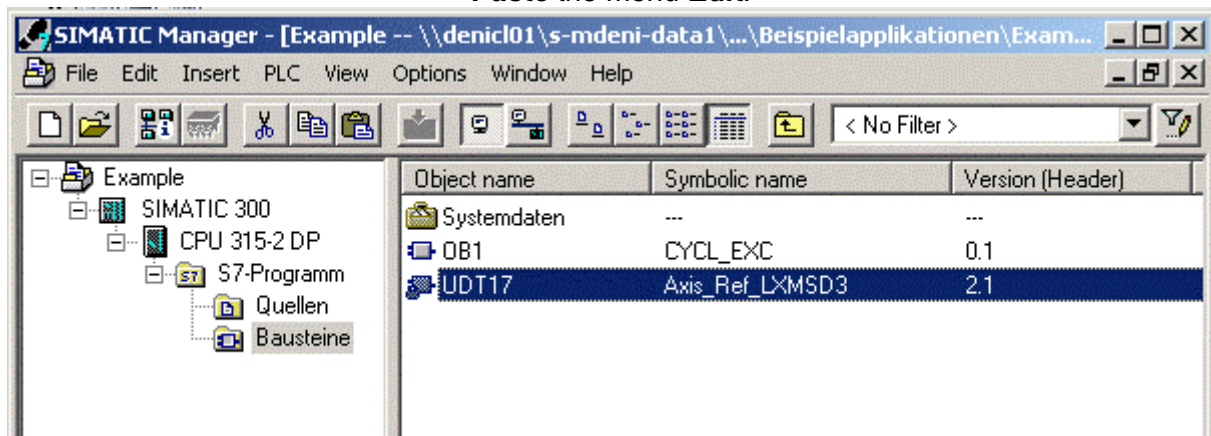
Open the previously extracted library in the SIMATIC Manager.



Next, mark the block UDT18, and copy it into the clipboard with the function **Copy** in the menu **Edit**.



Close the library, and mark the block folder in your project.  
Now insert the block into your project from the clipboard by means of the function **Paste** the menu **Edit**.



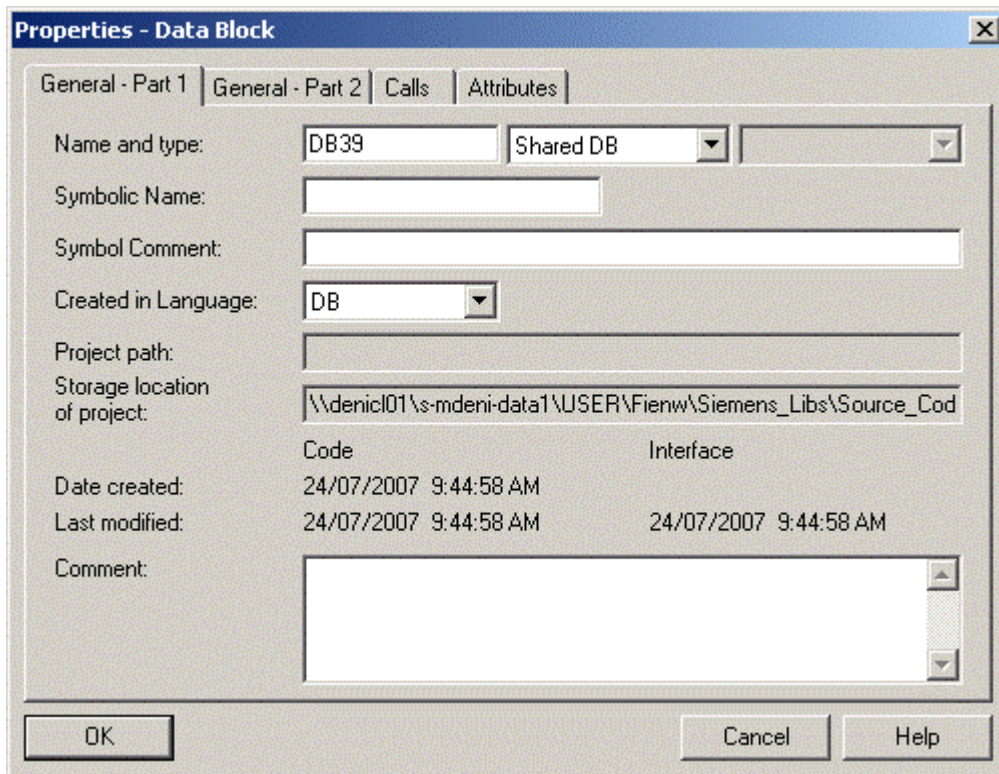
**Note:** Of course, you can also change the number of the UDT.  
This is done with the function **Rename** in the menu **Edit**.

## 7.2 Creating an axis reference

Create a global data block (Axis DB) in the block folder as follows:

Select **Insert / S7-Block / Data Block** in the Simatic Manager.

Edit the block's properties according to your requirements. Hereby, you must keep in mind that the block is a global DB.



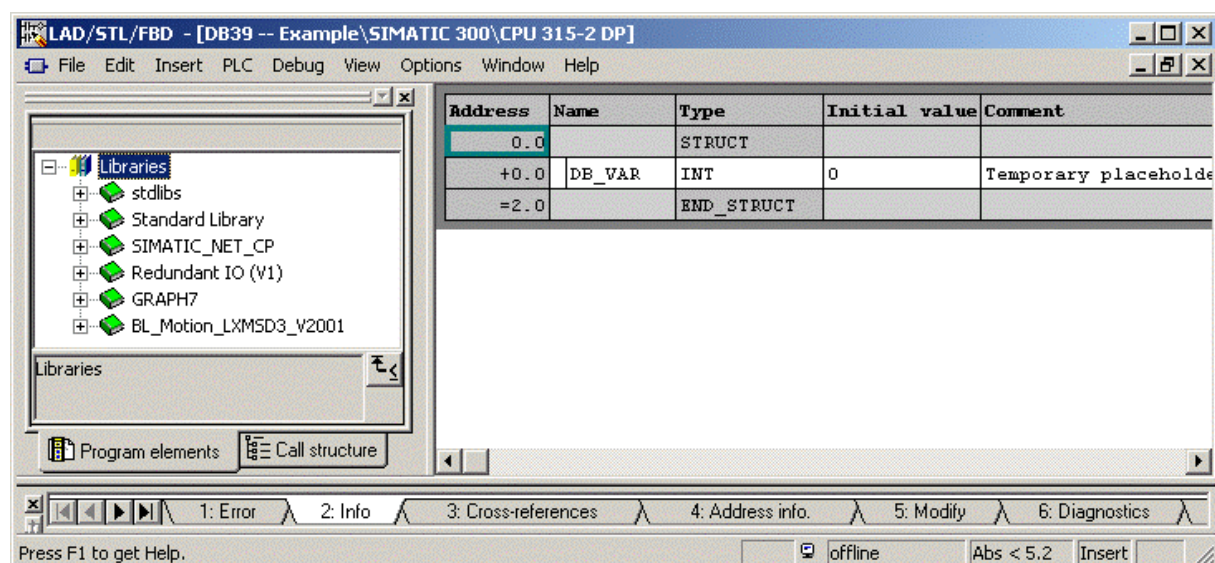
The 'Properties - Data Block' dialog box is shown with the 'General - Part 1' tab selected. The fields are as follows:

- Name and type: DB39, Shared DB
- Symbolic Name: (empty)
- Symbol Comment: (empty)
- Created in Language: DB
- Project path: (empty)
- Storage location of project: \\denic01\s-mdeni-data1\USER\Fienw\Siemens\_Libs\Source\_Cod
- Date created: 24/07/2007 9:44:58 AM
- Last modified: 24/07/2007 9:44:58 AM
- Comment: (empty text area)

Buttons at the bottom: OK, Cancel, Help.

Confirm your entries with OK.

Next, open the block by marking it and selecting the sub-menu item **Open Object** in the menu **Edit**. This starts the LAD/STL/FBD editor, with which you can edit the block.



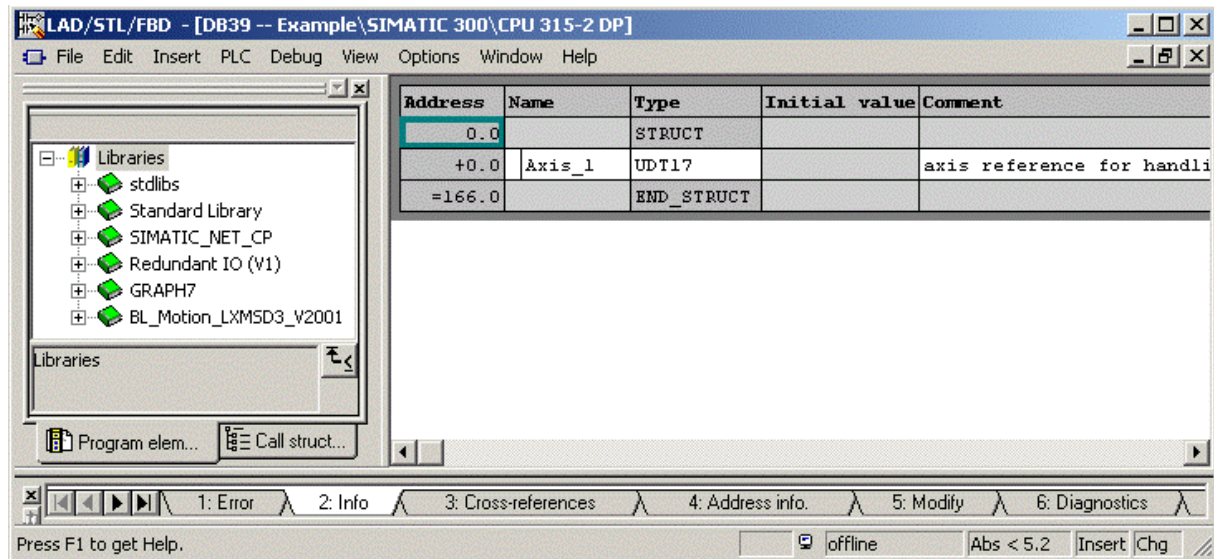
The 'LAD/STL/FBD' editor window is shown for the block 'DB39 -- Example\SIMATIC 300\CPU 315-2 DP'. The left pane shows the 'Libraries' tree with 'BL\_Motion\_LXM5D3\_V2001' selected. The right pane shows a table of variables:

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DB_VAR	INT	0	Temporary placeholder
=2.0		END_STRUCT		

The bottom status bar shows 'Press F1 to get Help.' and 'offline'.

## Step 7 Motion Library for Lexium 05B and SD 328B

You can now create the axis reference by defining a variable of the type UDT17. In case you have renamed the UDT, you must also rename the type.



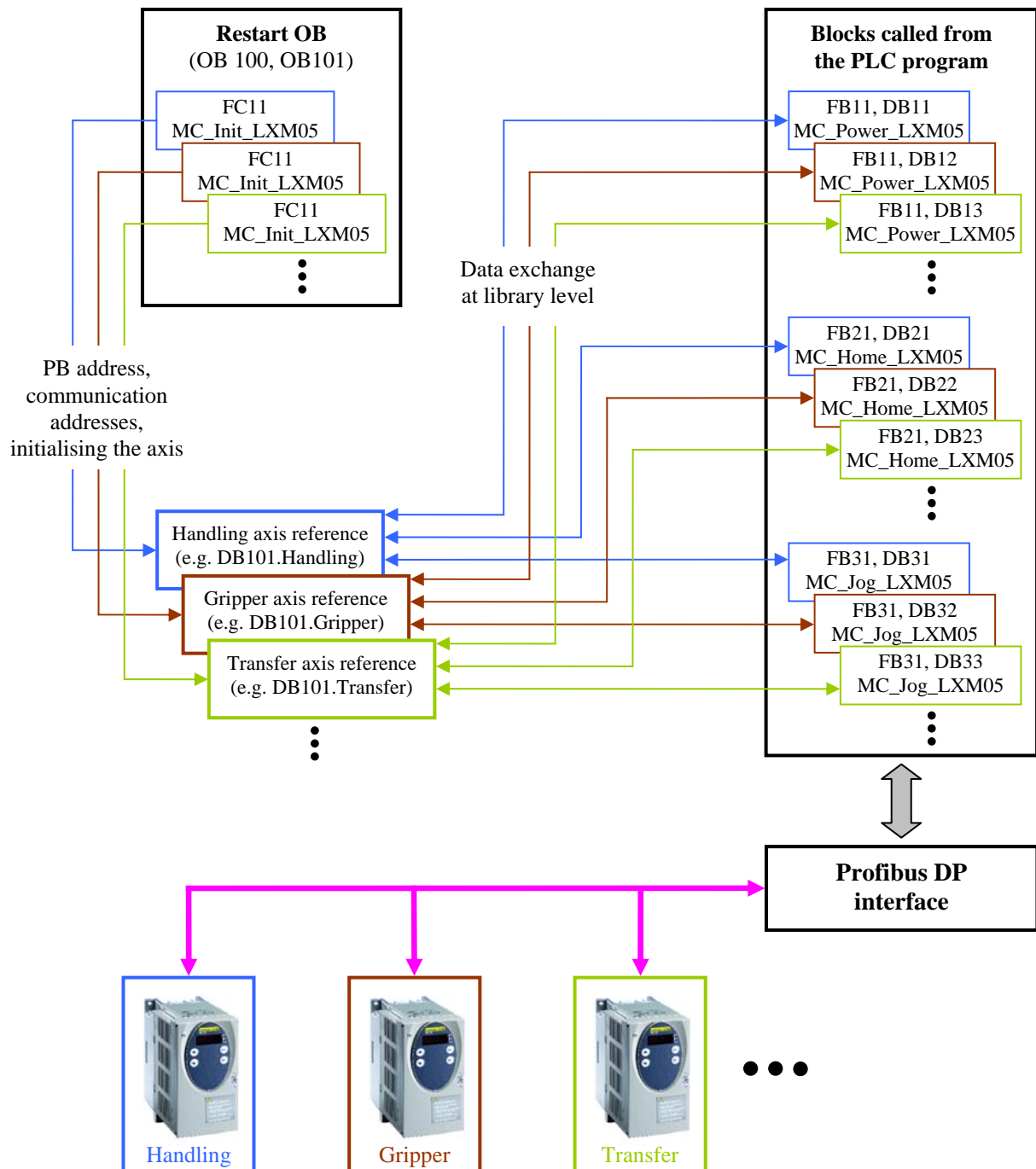
**Note:** If you are using several drives, you can create all the axis references in one block.

This method of defining the axis reference only represents one of several possibilities. Of course, other concepts can be applied, whereby it must only be ensured that all the blocks of an axis use the same structure.

You have now established the basic conditions to start with the actual programming of your application.

## 8 Library blocks

### 8.1 Basic calling procedures





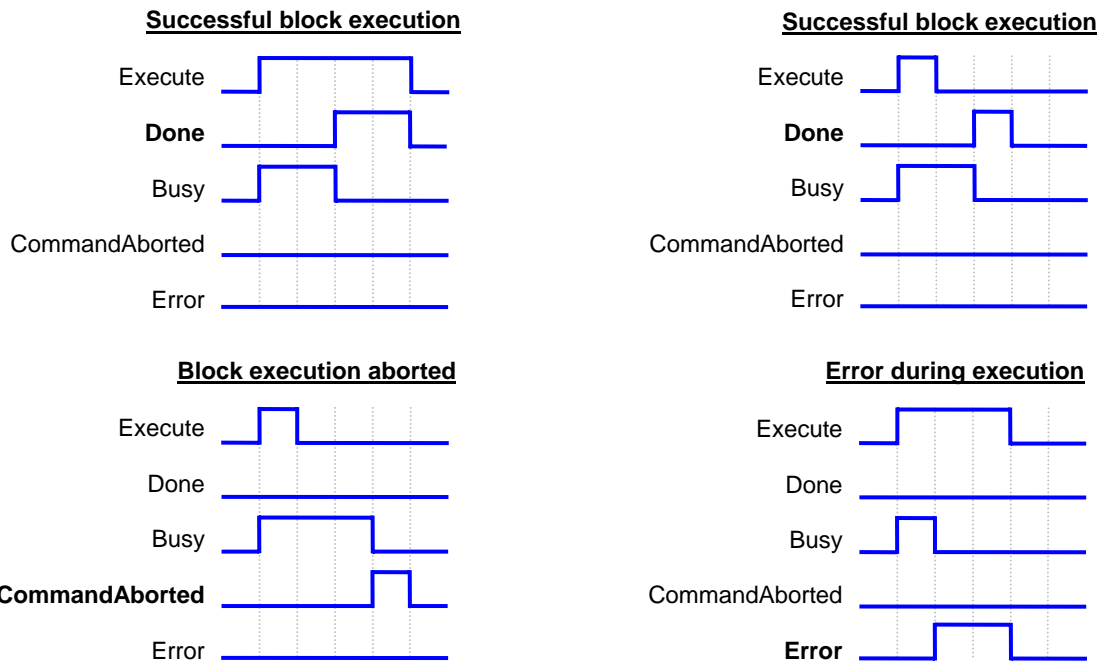
## 8.2 Explanation of common parameters

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	Starts (=TRUE) and stops (=FALSE) the block's execution. The block continues to be executed as long as TRUE is returned (level-sensitive).
	Execute	BOOL	<p>The block is executed once by a rising edge. With all motion blocks (except MC_Home) the input parameters are read after a rising edge during execution, and the movement is then continued with the new parameters.</p> <p>When the block has been executed (Busy = FALSE), the output parameters are maintained until FALSE is returned. The falling edge deletes the output parameters. If the input is already FALSE when execution is completed, the output parameters are issued during precisely one block call, and then deleted (edge-sensitive).</p>
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
INOUT	Axis	STRUCT	This parameter will be transferred to the AxisDB. Example: Axis := DBname.Axisname
	Init	BOOL	<p>An initialising bit in the axis DB, which is not used by any other FB, is transferred to this parameter ("Init.I0" .. "Init.I63", see bit field for the initialising function). With Init (=TRUE), the block executes its initialisation and subsequently resets the bit.</p> <p>Example: Init := DBname.Axisname.Init.Ix with Ix = I0 .. I63.</p>

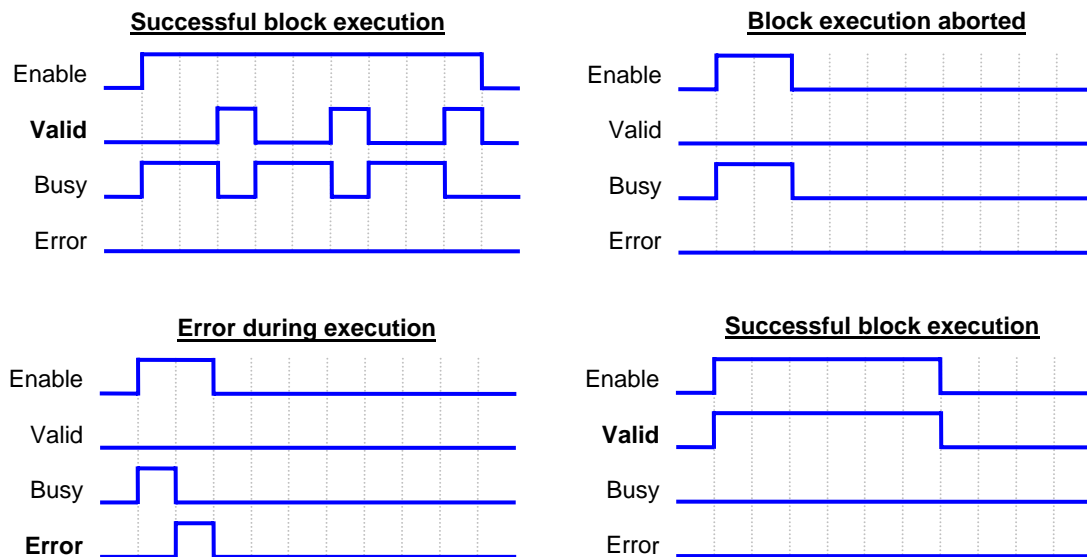


## 8.3 Phasing diagrams

### Phasing diagrams with Execute



### Phasing diagrams with Enable



## 8.4 Initialisation

After every restart (hot or cold) of the PLC, it is necessary for the library blocks to be re-initialised, in order to set the block's local data into a defined original state (initial value).

For this purpose, the bit field "Init.Ix" is defined in the axis DB, and every block has the in/out parameter "Init". The blocks execute their initialisation routine once, if their parameter "Init" is set, and subsequently reset the transferred initialisation bit automatically. Hereby, the function MC\_Init\_LXM05 also handles initialisation of the axis DB, as well as setting the initialisation bits in the axis DB for initialising the library blocks.

Therefore, every library block called in the user program must be given an initialisation bit from the axis DB via the in/out parameter "Init".

Important: Every initialisation bit may only be used by one library block. For this purpose, a total of 64 initialisation bits (AxisDB.Init.I0.. AxisDB.Init.I63) is available per axis.

By means of initialisation, it is ensured that no erratic functions and dangerous, unexpected motor movements are caused by old, invalid data.

### Note:

By using the library specific GSD, the PZD5 and PZD6 in the process data channel are automatically mapped during the profibus initialisation for send and receive direction.

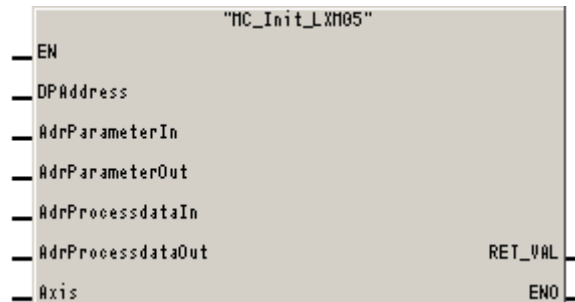
**It is not allowed to change this mapping, otherwise the functionality of the library is not guaranteed !**

### 8.4.1 MC\_Init\_LXM05

#### Task:

Initialisation of an axis.

#### Calling:



Call the block MC\_Init\_LXM05 once for every axis after every start of the CPU (OB100 and OB101). Depending on your program structure, you can also call the block directly in the corresponding restart OB.

#### Parameter description:

Par. type	Parameter	Data type	Description
IN	DPAddress	INT	Profibus address of the axis
	AdrParameterIn	INT	Input address of the <a href="#">Parameterdatachannel</a> .
	AdrParameterOut	INT	Output address of the <a href="#">Parameter data channel</a> .
	AdrProcessdataIn	INT	Input address of the <a href="#">Process data channel</a> .
	AdrProcessdataOut	INT	Output address of the <a href="#">Process data channel</a> .
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
OUT	Ret_Val	INT	Error number (value <>0 = error).

#### Operating principle:

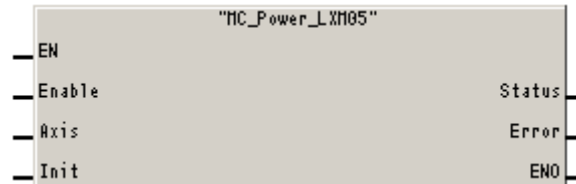
As far as possible, the transferred addresses are checked for plausibility, and entered into the axis structure of the parameter *Axis*. In addition, the initialisation bits are set in order to prepare for initialisation of the library blocks.

## 8.4.2 MC\_Power\_LXM05

### Task:

Switching the motor current on/off.

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Switches the motor current off. TRUE: Switches the motor current on.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Status	BOOL	Indicates the status of the motor current. FALSE: Motor current is 'off'. TRUE: Motor current is 'on'.
	Error	BOOL	TRUE: An error has occurred during execution.

### Operating principle:

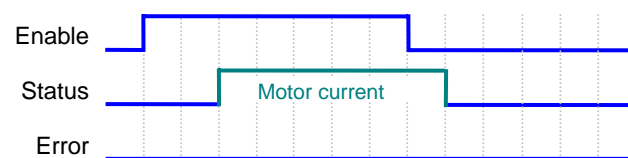
With TRUE at input *Enable*, the motor current is switched 'on'. As soon as the motor current is switched on, the output *Status* is set.

With FALSE at input *Enable*, the motor current is switched 'off'. As soon as the motor current is switched off, the output *Status* is reset.

If an error occurs during execution, the output *Error* is set.

The motor current can be switched off from any status. Any motion block that is active at this point will be aborted.

### Phase diagram:



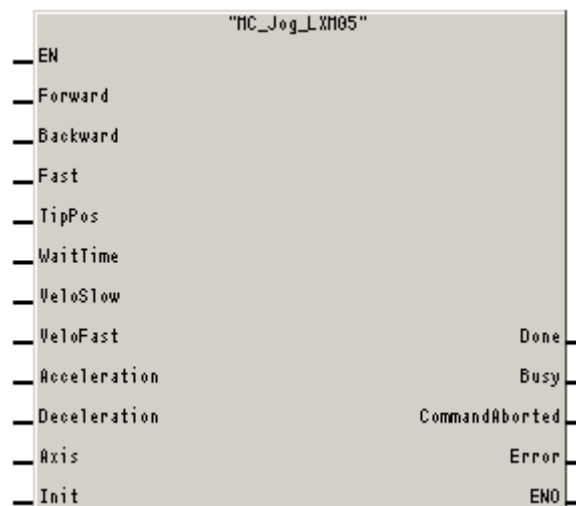
## 8.5 Jog

### 8.5.1 MC\_Jog\_LXM05

#### Task:

Jog is carried out in the “classical manual mode”, i.e. with the inputs *Forward* or *Backward* active for a longer period, the motor changes to continuous operation.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Forward	BOOL	FALSE: Stops the movement. TRUE: The axis moves in the clockwise direction.
	Backward	BOOL	FALSE: Stops the movement. TRUE: The axis moves in the counter-clockwise direction.
	Fast	BOOL	Speed switchover is also possible during operation: FALSE: Speed <i>VeloSlow</i> is selected. TRUE: Speed <i>VeloFast</i> is selected.
	TipPos	DINT	0: Infinite, i.e. the motor switches to continuous operation immediately. >0: Distance [usr] travelled by the motor after start, before it switches to continuous operation after the delay time ( <i>WaitTime</i> ) has elapsed. Value range: depends on scaling factor, initial value: 20.
	WaitTime	INT	Delay time [ms], which starts when the motor has travelled a defined distance ( <i>TipPos</i> ), and after which the motor switches to continuous operation. Value range: 1..32767, initial value: 500.
	VeloSlow	INT	Speed [rpm] for movement if Fast = FALSE. Value range: 1..3000, initial value: 60.



IN	VeloFast	INT	Speed [rpm] for movement if Fast = TRUE. Value range: 1..3000, initial value: 180.
	Acceleration	INT	Value for the acceleration ramp gradient [(10 rpm/s²)] Value range: 30..65535, initial value: 600.
	Deceleration	INT	Value for the deceleration ramp gradient [(10 rpm/s²)] Value range: 750..65535, initial value: 750.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

## Operating principle:

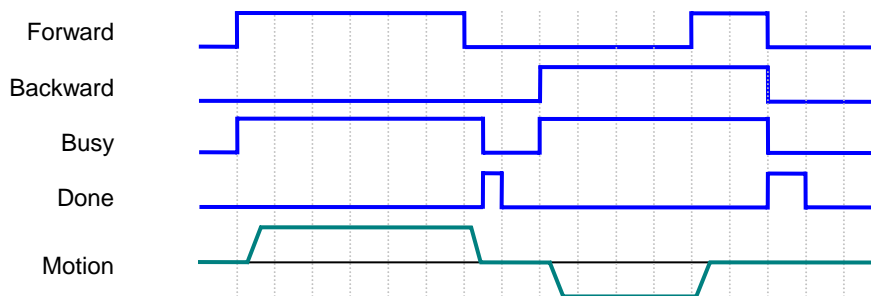
With TRUE at the *Forward* or *Backward* input, jog is started.

Depending on the parameter *Fast*, operation is either with the slow (*VeloSlow*) or with the fast (*VeloFast*) speed. The speed can also be changed during active jog. The parameters *TipPos* and *WaitTime* are used to determine the conditions for switching from the jogging mode to continuous operation.

If *Forward* and *Backward* = FALSE, the operating mode is terminated, and *Done* is set.

If *Forward* and *Backward* = TRUE, the operating mode remains active, the jogging mode is stopped, and *Busy* remains set.

## Phase diagram:



## 8.6 Homing

In homing mode, an absolute scale reference of the motor position at a defined axis position is established. Homing can be executed by means of the two blocks described below.

### 8.6.1 MC\_SetPosition\_LXM05

#### Task:

Absolute and relative set dimensions.

#### Calling:

Set dimensions can only be carried out while the drive is at standstill.



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Position	DINT	Dimension setting position [usr] Value range: - 2147483648..2147483647, initial value: 0.
	Mode	BOOL	FALSE: Set current motor position as <i>Position</i> . TRUE: Add <i>Position</i> to current motor position.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

#### Operating principle:

Specifying the dimension reference relative to the current motor position Set dimensions allows the current motor position to be defined as the new axis reference point to which all subsequent position data relate. Set dimensions shifts the reference point for setpoint positions to the new dimension setting position.

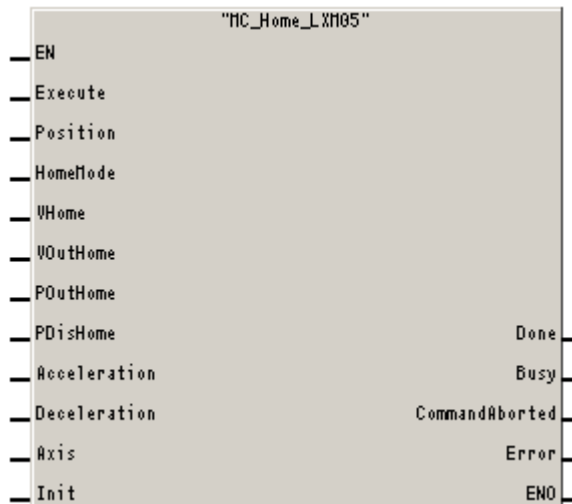
Set dimensions can be used to carry out a continuous absolute positioning without exceeding the positioning limits.

## 8.6.2 MC\_Home\_LXM05

### Task:

Executing the reference movement.

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Position	DINT	Position is set as current motor position after successful reference movement [usr]. Value range: depends on scaling factor, initial value: 0.
	HomeMode	INT	LIMN with index pulse 1 : LIMP with index pulse 2: 7 = REF+ with index pulse, beyond REF, in direction of LIMN 8 = REF+ with index pulse, within REF, in direction of LIMN 9 = REF+ with index pulse, within REF, in direction of LIMP 10 = REF+ with index pulse, beyond REF, in direction of LIMP 11 = REF- with index pulse, beyond REF, in direction of LIMN 12 = REF- with index pulse, within REF, in direction of LIMN 13 = REF- with index pulse, within REF, in direction of LIMP 14 = REF- with index pulse, beyond REF, in direction of LIMP 17 = LIMN 18 = LIMP 23 = REF+, beyond REF, in direction of LIMN 24 = REF+, within REF, in direction of LIMN 25 = REF+, within REF, in direction of LIMP 26 = REF+, beyond REF, in direction of LIMP 27 = REF-, beyond REF, in direction of LIMN 28 = REF-, within REF, in direction of LIMN 29 = REF-, within REF, in direction of LIMP 30 = REF-, beyond REF, in direction of LIMP 33 = on index pulse, in direction of LIMN 34 = on index pulse, in direction of LIMP
	VHome	INT	Speed for searching the limit or reference switch [rpm]. Drive stops when switching edge has been detected. Value range: 1..13200, initial value: 60.

Par. type	Parameter	Data type	Description
IN	VOutHome	INT	Speed for clearance movement back to the switching edge [rpm]. The max. travel distance when searching for the switching edge can be restricted with the parameter <i>POutHome</i> . Value range: 1..3000, initial value: 6.
	POutHome	DINT	0: Clearing monitor switched off. >0: Run-off [usr], i.e. max. travel distance when searching for the switching edge. If the switching edge is not found in this distance, the reference movement is interrupted with an error. Value range: 0..2147483647, initial value: 0.
	PDisHome	DINT	Distance between the switching edge and the reference point [usr]. At end of movement, the drive moves back towards switching edge until the distance has been reached. Value range: 1..2147483647, initial value: 200.
	Acceleration	INT	Value for the acceleration ramp gradient [(10 rpm/s <sup>2</sup> )] Value range: 30..65535, initial value: 600.
	Deceleration	INT	Value for the deceleration ramp gradient [(10 rpm/s <sup>2</sup> )] Value range: 750..65535, initial value: 750.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

### Operating principle:

In the reference movement a defined position on the axis is approached. The defined position is specified by a mechanical switch: Limit switch, reference switch.

There are four standard reference movements:

1. Movement to positive limit switch LIMP.
2. Movement to negative limit switch LIMN.
3. Search for the reference switch REF using clockwise rotation.
4. Search for the reference switch REF using counter-clockwise rotation.

A reference movement can be executed with or without an index pulse.

- Reference movement without index pulse  
Movement from switching edge to a parametrisable distance from the switching edge.
- Reference movement with index pulse  
Movement from the switching edge to the next index pulse of the encoder.

For the reference movement, search speed (*VHome*), clearance speed (*VOutHome*), safety distance (*PDisHome*) and the clearance path (*POutHome*) are adjustable. A reference movement must be completed for the new reference point to be valid. If a reference movement has been aborted, it must be restarted.

The motor moves as a function of these parameters until it reaches its target or the operating mode is interrupted by the execution of another block (e.g. MC\_Stop).

Any attempt to accept changed parameters by means of a rising edge at *Execute* during a homing movement is not allowed, and results in an error.

After successful completion of the reference movement, a reference position is created automatically. In this way, the reached position is defined as the absolute reference position, and is set as the value of *Position*.

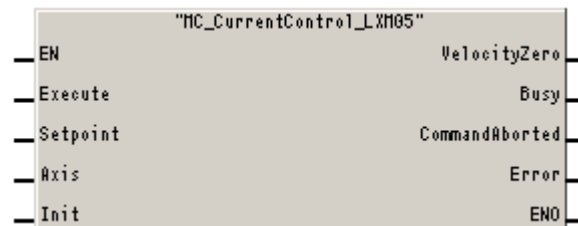
## 8.7 Current control

### 8.7.1 MC\_CurrentControl\_LXM05

#### Task:

Starting and monitoring the current control operating mode.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Setpoint	DINT	Set current [Apk x 0.01]. Value range: -30000..+30000, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	VelocityZero	BOOL	FALSE: The motor is running. TRUE: The motor is at standstill.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

#### Operating principle:

In the current control operating mode, the reference value for the motor current is preset directly via the parameter *Setpoint*, and a movement without target position is started. The motor moves as a function of this setpoint until a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC\_Stop).

#### Note:

In the current control mode, the drive can reach extreme speeds when operated without limits or load. By means of the device parameter CTRL\_n\_max (see manual), the max. speed can be limited to protect the drive system.



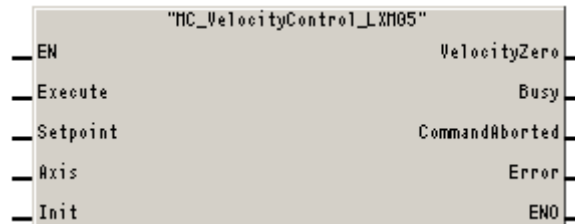
## 8.8 Speed control

### 8.8.1 MC\_VelocityControl\_LXM05

#### Task:

Starting and monitoring the speed control operating mode.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Setpoint	DINT	Set speed [rpm]. Value range: -30000..+30000, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	VelocityZero	BOOL	FALSE: The motor is running. TRUE: The motor is at standstill.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

#### Operating principle:

In the speed control operating mode, the reference value for motor speed is preset directly via the parameter *Setpoint*, and a movement without target position is started. The motor moves as a function of this setpoint until a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC\_Stop).

#### Note:

In the device documentation, this operating mode is described as "Oscillator mode". Transitions between two speeds are executed only as a function of the adjusted control parameters, compare speed mode where the transitions are defined via a profile generator.

## 8.9 Profile position mode

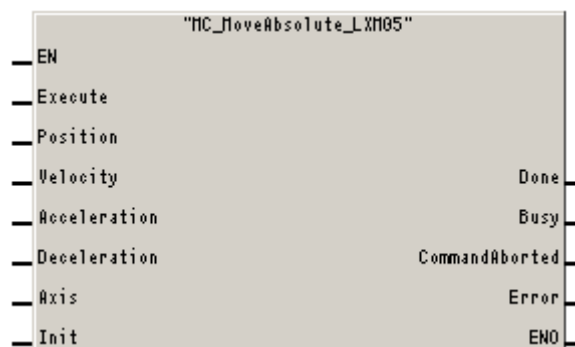
In the profile position operating mode, the motor is positioned from a point A to a point B by means of a positioning block.

### 8.9.1 MC\_MoveAbsolute\_LXM05

#### Task:

Starting and monitoring the profile position operating mode with an absolute target position.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Position	DINT	Value for the absolute target position [usr]. Value range: depends on scaling factor, initial value: 0.
	Velocity	INT	Value for the set speed of the movement [rpm]. Value range: 1..13200, initial value: 60.
	Acceleration	INT	Value for the acceleration ramp gradient [(10 rpm/s²)] Value range: 30..65535, initial value: 600.
	Deceleration	INT	Value for the deceleration ramp gradient [(10 rpm/s²)] Value range: 750..65535, initial value: 750.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

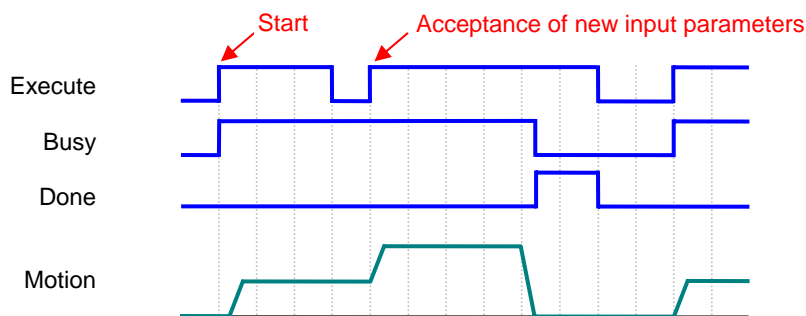
## Operating principle:

Positioning on the target position *Position* at speed *Velocity*, and with absolute reference to the axis zero point. The motor moves as a function of these parameters until it reaches its target, a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC\_Stop).

### Note:

Before an absolute positioning, the reference point must be defined by homing.

### Phase diagram:

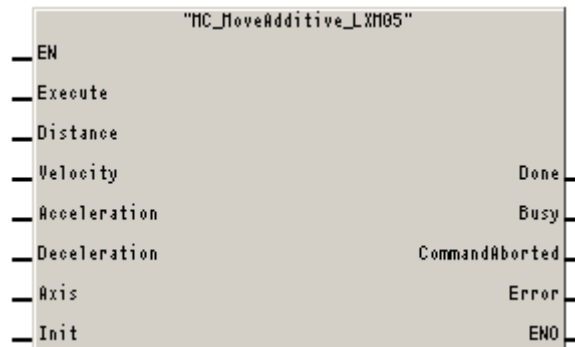


## 8.9.2 MC\_MoveAdditive\_LXM05

### Task:

Starting and monitoring the profile position operating mode with a target position relative to the current target position.

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Distance	DINT	Value for the travel distance that is added to the current target position, and thus determines the new target position [usr]. Value range: depends on scaling factor, initial value: 0.
	Velocity	INT	Value for the set speed of the movement [rpm]. Value range: 1..13200, initial value: 60.
	Acceleration	INT	Value for the acceleration ramp gradient [(10 rpm/s²)] Value range: 30..65535, initial value: 600.
	Deceleration	INT	Value for the deceleration ramp gradient [(10 rpm/s²)] Value range: 750..65535, initial value: 750.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

### Operating principle:

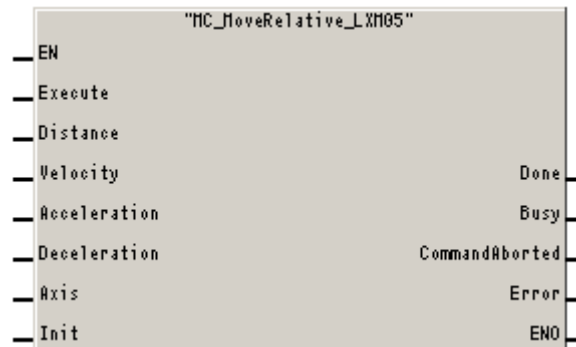
Positioning with a travel distance *Distance* referred to the current target position at speed *Velocity*. The motor moves as a function of these parameters until it reaches its target, a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC\_Stop).

### 8.9.3 MC\_MoveRelative\_LXM05

#### Task:

Starting and monitoring the profile position operating mode with a target position relative to the current motor position.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Distance	DINT	Value for the travel distance that is added to the current motor position, and thus determines the new target position [usr]. Value range: depends on scaling factor, initial value: 0.
	Velocity	INT	Value for the set speed of the movement [rpm]. Value range: 1..13200, initial value: 60.
	Acceleration	INT	Value for the acceleration ramp gradient [(10 rpm/s²)] Value range: 30..65535, initial value: 600.
	Deceleration	INT	Value for the deceleration ramp gradient [(10 rpm/s²)] Value range: 750..65535, initial value: 750.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

#### Operating principle:

Positioning with a travel distance *Distance* referred to the current motor position at speed *Velocity*. The motor moves as a function of these parameters until it reaches its target, a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC\_Stop).

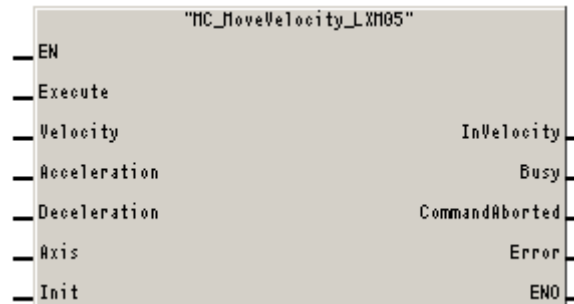
## 8.10 Velocity mode

### 8.10.1 MC\_MoveVelocity\_LXM05

#### Task:

Starting and monitoring the velocity mode.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Velocity	INT	Value for the set speed of the movement [rpm]. Value range: -13200..13200, initial value: 0.
	Acceleration	INT	Value for the acceleration ramp gradient [(10 rpm/s²)] Value range: 30..65535, initial value: 600.
	Deceleration	INT	Value for the deceleration ramp gradient [(10 rpm/s²)] Value range: 750..65535, initial value: 750.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	InVelocity	BOOL	TRUE: Set speed reached
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

#### Operating principle:

In the velocity operating mode, the motor receives a set speed via the parameter *Velocity*, and a movement without a target position is started.

The motor moves as a function of this setpoint until a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC\_Stop).

## 8.11 Electronic gear

In the electronic gear mode the positioning controller calculates a new position preset for the motor movement from a position preset and an adjustable gear ratio. This mode is used if one or more motors is to follow the reference signal from a NC controller or an encoder.

The reference signals for the electronic gear operating mode must be applied at the RS 422 interface. If reference pulses are applied, the positioning controller offsets them with the gear ratio, and positions the motor at the new setpoint position.

Position values are given in internal increments. If the values change, the positioning controller follows immediately. Electronic gear mode is not limited by the positioning area boundaries.

The gear ratio is the relationship between the motor increments and the externally applied reference pulses for motor movement. The gear ratio is determined with numerator and denominator. A negative numerator reverses the direction of rotation. The resulting positioning path is dependent upon the current motor resolution. It amounts to 131072 motor increments per revolution.

### Notes

**Prerequisite:** The device parameter *GEARratio* (see manual) must be set to 0 (default), for the input parameters *RatioNumerator* and *RatioDenominator* to be effective.

**Synchronisation:** Before the electronic gear operating mode is started, there is no synchronisation between reference pulses and motor.  
In case of a compensation movement (*SyncMode* = TRUE), motor movement is only limited by the max. current (device parameter *CTRL\_I\_max*, see manual) and the drive's max. speed (device parameter *CTRL\_n\_max*, see manual).  
In case of immediate synchronisation (*SyncMode* = FALSE) the motor will follow the reference pulses, starting from the time when gear processing is enabled in the drive.

**Direction enable:** The direction enabling allows restriction of the movement to clockwise or counterclockwise rotation. The direction enable is activated with the device parameter *GEARdir\_enabl* (see manual).

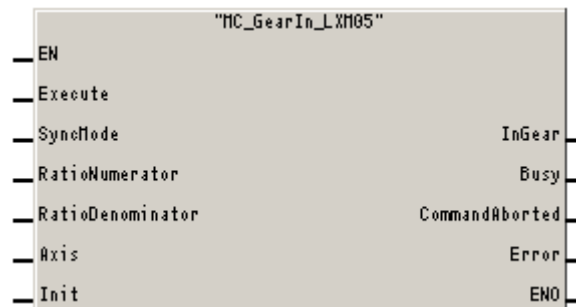


### 8.11.1 MC\_GearIn\_LXM05

#### Task:

Starting and monitoring the electronic gear operating mode with a gear ratio.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	SyncMode	BOOL	FALSE: Real-time synchronisation. The positioning controller follows reference pulses from the time at which the gear processing is activated. Reference pulses that occur before the operating mode is started, are ignored. TRUE: Synchronisation with compensation movement. After gear processing has been enabled, the motor attempts to execute the accumulated reference pulses.
	RatioNumerator	DINT	Gear ratio numerator. Value range: -2147483648 .. 2147483647, initial value: 1.
	RatioDenominator	DINT	Gear ratio denominator. Value range: 1 .. 2147483647, initial value: 1.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	InGear	BOOL	FALSE: The electronic gear is disabled. TRUE: The electronic gear is enabled.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

#### Operating principle:

In the electronic gear operating mode, the motor is supplied continuously with calculated position presets in the form of reference pulses at the encoder input in combination with the gear ratio (parameters *Numerator* and *Denominator*).

The motor moves as a function of this setpoint until it receives a new gear ratio. Operation is terminated by execution of the block MC\_GearOut\_LXM05 or by execution of another block (e.g. MC\_Stop).

## 8.11.2 MC\_GearOut\_LXM05

### Task:

Switch-off of the electronic gear operating mode.

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Done	BOOL	TRUE: Block execution was completed successfully.
OUT	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

### Operating principle:

With enabled electronic gear operating mode the motor is directly uncoupled from the gear master, and braked to a standstill by means of a torque ramp. Hereby, the parameter LIM\_I\_maxHalt (see manual) specifies the torque ramp current.

This interrupts the active block MC\_GearIn\_LXM05, which in turn signals CommandAborted = TRUE.

## 8.12 Stopping

### 8.12.1 MC\_Stop\_LXM05

#### Task:

Stopping the drive with a torque ramp.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Done	BOOL	TRUE: Block execution was completed successfully.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

#### Operating principle:

Every operating mode can be terminated by stopping the drive. This does not generate an error. The interrupted movement block terminates its execution with *CommandAborted* = TRUE, and the drive changes to the status “Stopping”. This status remains active until the drive is at a standstill and the block's input *Execute* has been reset. The status then changes to “Standstill”, and movement blocks can be started again.

#### Note:

This function cannot be interrupted by other movement block. As long as *Execute* = TRUE, no other movement block can be started. Also after standstill the drive remains blocked.

The block brakes the motor with a torque ramp. The parameter LIM\_I\_maxHalt (see manual) determines the current for the torque ramp. After drive standstill an internal position compensation is run, the position control is enabled and the motor is stopped with the power amplifier active.

## 8.13 Fast position capture

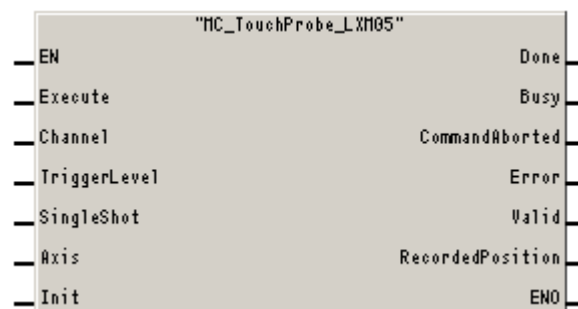
The motor position can be captured with a precision of 2  $\mu$ s delay by means of 2 parametrisable channels. Two signal inputs CAP1 and CPA2 are available for capturing the trigger signals.

### 8.13.1 MC\_TouchProbe\_LXM05

#### Task:

Adjusting, starting, and monitoring the fast position capture.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Channel	INT	Channel number: Selection of the channel to which the other parameters are referred (1 = CAP1, 2 = CAP2). Value range: 1 .. 2, initial value: 1.
	TriggerLevel	BOOL	Triggering signal edge. FALSE: falling edge: TRUE: rising edge
	SingleShot	BOOL	FALSE: If the triggering event occurs repetitively, the recorded position is overwritten with the most recent position. TRUE: Position capture is disabled after the triggering event has occurred, so that the recorded position cannot be overwritten. Initial value: TRUE.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.
	Valid	BOOL	TRUE: The value to be read is available.
	RecordedPosition	BOOL	Recorded motor position when the triggering event occurs [usr]. Value range: -2147483648 .. 2147483647, initial value: 0.

### Operating principle:

Fast position capture serves to detect the current motor position at the time when a digital 24V signal appears at one of the two capture inputs (LIMP = CAP1, LIMN = CAP2).

The motor position can be detected here by means of a falling edge at the corresponding input (*Channel*).

Moreover, the parameter *SingleShot* can be used to determine whether position capture is to be executed once or continuously.

In case of a single position capture, the block is terminated with *Done* = TRUE as soon as the adjusted edge (*TriggerLevel*) occurs, and signals the detected position (*RecordedPosition*).

In case of continuous position capture, the block signals a *Valid* = TRUE and the detected position (*RecordedPosition*) with every occurring edge (*TriggerLevel*), whereby the previous position is overwritten. The block does not terminate itself – it can only be interrupted with MC\_AbortTrigger\_LXM05.

### 8.13.2 MC\_AbortTrigger\_LXM05

#### Task:

Terminating an active position capture.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Channel	INT	1: Terminates position capture on channel 1 (CAP1). 2: Terminates position capture on channel 2 (CAP2). Value range: 1 .. 2, initial value: 1.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

### Operating principle:

The active position capture is disabled for the corresponding *Channel*.

For this channel, the block MC\_TouchProbe\_LXM05 signals *CommandAborted* = TRUE.

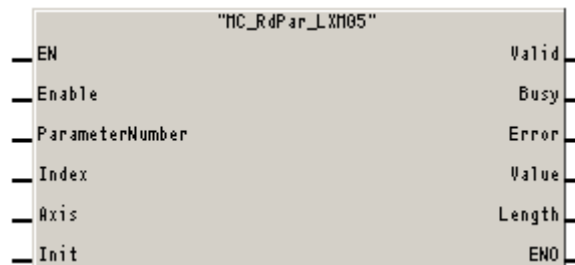
## 8.14 Read parameter

### 8.14.1 MC\_ReadParameter\_LXM05

#### Task:

Reading an Object from the [Deviceparameter list](#).

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	ParameterNumber	INT	0: The parameter is selected with <i>Index</i> . 1: Current setpoint position of the profile generator [usr]. 2: Position of the positive software limit switch [usr] 3: Position of the negative software limit switch [usr] 4: Enable (Bit0=1) or Inhibit (Bit0=0) of the positive software limit switch 5: Enable (Bit0=1) or Inhibit (Bit0=0) of the negative software limit switch 10: Current actual speed [rpm]. 11: Current set speed [rpm]. Other numbers are not supported. Value range: 0..32767, initial value: 0.
	Index	INT	Index of the Object to be read – the Objects are listed in the manual with their index. Only valid with <i>ParameterNumber</i> = 0. Value range: 0..32767, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Value	DINT	Value of the parameter to be read. Value range: -2147483648..2147483647, initial value: 0.
	Length	INT	Length of the parameter to be read [bytes]. Value range: 1..4, initial value: 0.

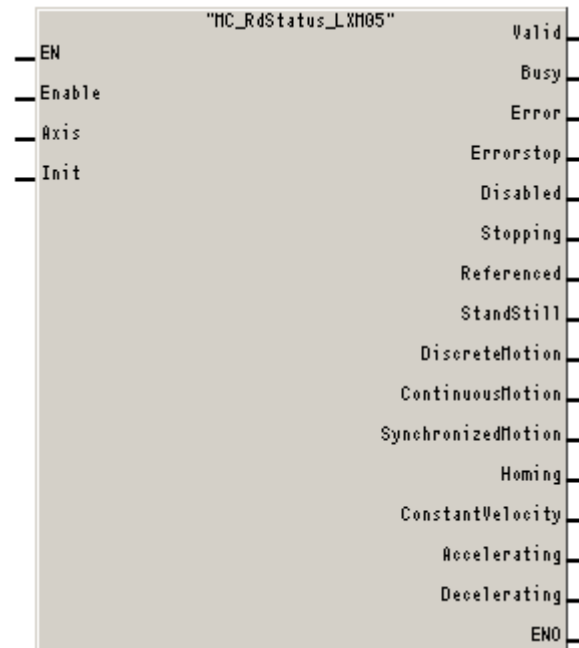


## 8.14.2 MC\_ReadStatus\_LXM05

### Task:

Reading the drive's current status.

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The read status is valid. FALSE: The status is not (yet) valid.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Errorstop	BOOL	TRUE: The axis is in a fault condition.
	Disabled	BOOL	TRUE: Motor current is 'off'.
	Stopping	BOOL	TRUE: The axis has been stopped and is still being blocked by the block MC_Stop_LXM05.
	Referenced	BOOL	TRUE: The drive is ready.
	StandStill	BOOL	TRUE: The drive is stopped.
	DiscreteMotion	BOOL	TRUE: The drive is in a profile position operating mode.

Par. type	Parameter	Data type	Description
OUT	ContinuousMotion	BOOL	TRUE: The drive is in an operating mode without a defined target position (e.g. MC_MoveVelocity_LXM05, MC_Jog_LXM05).
	SynchronizedMotion	BOOL	The drive is in the electronic gear operating mode.
	Homing	BOOL	TRUE: The drive is in the homing operating mode.
	ConstantVelocity	BOOL	TRUE: The drive is running at a constant speed.
	Accelerating	BOOL	TRUE: The drive is accelerating.
	Decelerating	BOOL	TRUE: The drive is slowing down.

### Operating principle:

The drive's current status information is being read and output. These are only valid in connection with the parameter *Valid*.

### Note:

At any time, the drive is in only one of the states *StandStill*, *Homing*, *DiscreteMotion*, *ContinuousMotion*, *SynchronizedMotion*, *Stopping*, *Disabled* or *Errorstop*.

The correspondingly named output of the block is then TRUE.

The same applies for the movement conditions *ConstantVelocity*, *Accelerating*, and *Decelerating*.

### 8.14.3 MC\_ReadActualPosition\_LXM05

#### Task:

Reading the motor's actual position in user-defined units.

#### Calling:



#### Parameter description:

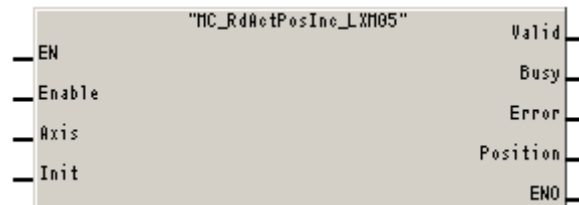
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
OUT	Position	DINT	Motor's current actual position [usr]. Value range: depends on scaling factor, initial value: 0.

## 8.14.4 MC\_ReadActualPositionInc\_LXM05

### Task:

Reading the motor's actual position in increments.

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Position	DINT	Motor's current actual position [Inc]. Value range: -2147483648..2147483647, initial value: 0.

## 8.14.5 MC\_ReadActualVelocity\_LXM05

### Task:

Reading the motor's current speed in rpm.

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Velocity	INT	Motor's current speed [rpm]. Value range: -13200..13200, initial value: 0.

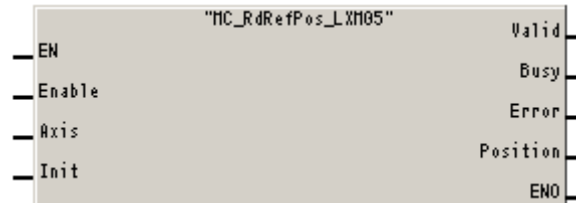


## 8.14.6 MC\_ReadRefPosition\_LXM05

### Task:

Reading the movement profile generator's current position in user-defined units.

### Calling:



### Parameter description:

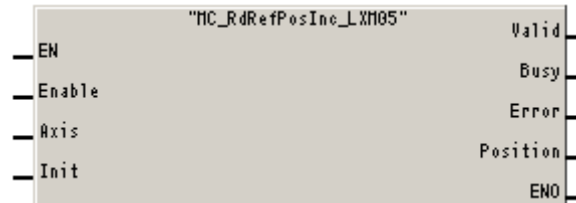
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Position	DINT	Actual position of the movement profile generator [usr]. Value range: depends on scaling factor, initial value: 0.

### 8.14.7 MC\_ReadRefPositionInc\_LXM05

#### Task:

Reading the movement profile generator's current position in increments.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Position	DINT	Actual position of movement profile generator [Inc]. Value range: -2147483648..2147483647, initial value: 0.

## 8.14.8 MC\_ReadRefVelocity\_LXM05

### Task:

Reading the movement profile generator's current speed in rpm.

### Calling:



### Parameter description:

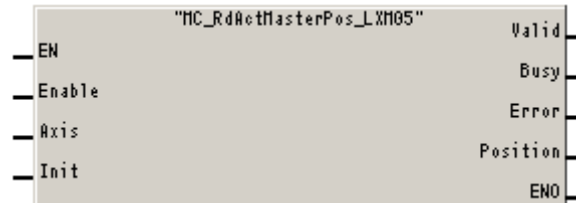
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Velocity	DINT	Current speed of the movement profile generator [rpm]. Value range: -13200..13200, initial value: 0.

## 8.14.9 MC\_ReadActualMasterPosition\_LXM05

### Task:

Reading the encoder's current position in increments.

### Calling:



### Parameter description:

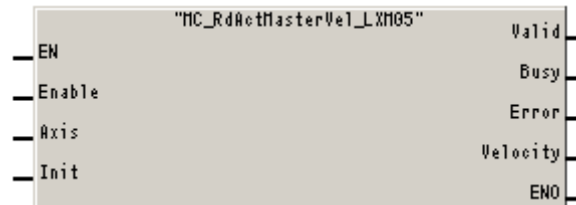
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
OUT	Position	DINT	Current position of the external encoder [Inc]. Value range: -2147483648..2147483647, initial value: 0.

### 8.14.10 MC\_ReadActualMasterVelocity\_LXM05

#### Task:

Reading the external encoder's actual speed in increments per second.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Velocity	INT	Actual speed of the external encoder [Inc/s]. Value range: -13200..13200, initial value: 0.



### 8.14.11 MC\_UploadParameter\_LXM05

#### Task:

Reading all variable parameters and store them in the [parameter list](#).

#### Calling:



#### Parameter description:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Data	ANY	Structure for the read data.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Errorinfo	DWORD	Additional error information, address of the error. High Word: DB-Number, Low Word: parameter address in the DB Value range: 16#0..16#FFFFFFF, initial value: 16#0.

#### Operating principle:

The list of the variable parameters is defined in the library in the UDT12 by the structure Data. For uploading the parameters a structure from the UDT12 has to be created in a Shared DB and this structure has to be hand over to the input Data of the function block. The read parameters will be written into this structure.

#### Note:

With the two blocks MC\_UploadParameter\_LXM05 and [MC\\_DownloadParameter\\_LXM05](#), a defective device can be exchanged without a special tool to parameterize the device.

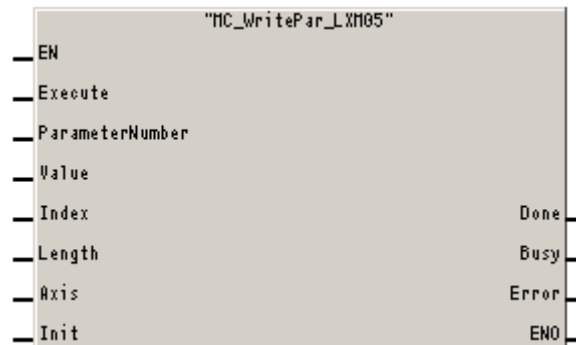
## 8.15 Write parameter

### 8.15.1 MC\_WriteParameter\_LXM05

#### Task:

Writing an Object from the [Deviceparameter list](#).

#### Calling:



#### Parameter description:

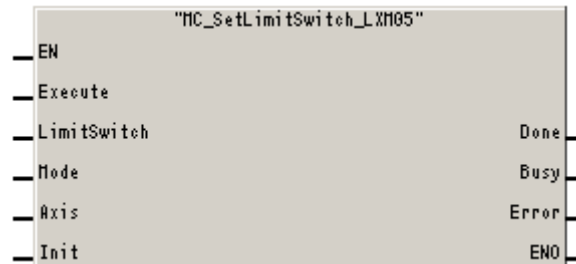
Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	ParameterNumber	INT	0: The parameter is selected with <i>Index</i> . >0: Number of the parameter that is to be written: 2: Position of the positive software limit switch [usr] 3: Position of the negative software limit switch [usr] 4: Enable (Bit0=1) or Inhibit (Bit0=0) of the positive software limit switch 5: Enable (Bit0=1) or Inhibit (Bit0=0) of the negative software limit switch Other numbers are not supported. Value range: 0..32767, initial value: 0.
	Value	DINT	Value of the parameter to be written. Value range: -2147483648..2147483647, initial value: 0.
	Index	INT	Index of the Object to be written – the Objects are listed in the manual with their index. Only valid with <i>ParameterNumber</i> = 0. Value range: 0..32767, initial value: 0.
	Length	INT	Length of the parameter to be written [bytes]. Value range: 1..4, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

## 8.15.2 MC\_SetLimitSwitch\_LXM05

### Task:

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

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	LimitSwitch	INT	1: Positive limit switch LIMP 2: Negative limit switch LIMN Value range: 1..2, initial value: 1.
	Mode	BOOL	0: Deactivate limit switch 1: Enable limit switch for normally-open operation 2: Enable limit switch for normally-closed operation Value range: 0..2, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

### 8.15.3 MC\_ResetParameters\_LXM05

#### Task:

Resetting the parameters to their status after “First Setup” (see manual).

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Done	BOOL	TRUE: Block execution was completed successfully.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

#### Operating principle:

All parameter values are reset to default values, with the exception of the communication parameters, the control mode and the logic type (“source” or “sink” of the inputs/outputs).

#### Note:

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

## 8.15.4 MC\_StoreParameters\_LXM05

### Task:

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

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Done	BOOL	TRUE: Block execution was completed successfully.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

## 8.15.5 MC\_DownloadParameter\_LXM05

### Task:

Writing all variable parameters out of the [parameter list](#) to the drive.

### Calling:



### Parameter description:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Data	ANY	Structure with the writing data.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Errorinfo	DWORD	Additional error information, address of the error. High Word: DB-Number, Low Word: parameter address in the DB Value range: 16#0..16#FFFFFFF, initial value: 16#0.

### Operating principle:

The list of the variable parameters is defined in the library by the UDT12. For downloading the parameters a structure from the UDT12 has to be created in a Shared DB and this structure has to be hand over to the input Data of the function block. The parameters to be written will be taken out of this structure.

### Note:

It is recommended to make an upload of all parameters with the function block [MC UploadParameters\\_LXM05](#) before downloading the parameters within this function block. The initialisation values of the parameter list may not be equal to the default values of those in the drive. After uploading the parameters it is possible to change the data in the defined structure and then execute the download.



## 8.16 Inputs/outputs

Apart from the process image, in which the digital inputs and outputs of the target system are displayed, other blocks are available, which provide access to the digital inputs/outputs of every drive in the system. The 24V signal interface of the drive provides 6 inputs and 2 outputs, which are assigned to functions such as limit switches.

### 8.16.1 MC\_ReadDigitalInput\_LXM05

#### Task:

Reading the drive's current input status.

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	InputNumber	INT	Number of the input that is to be read: 0: I0/REF 1: I1/LIMN 2: I2/LIMP 3: I3/HALT 4: I4/SAVE_DISABLE_B 5: I5/SAVE_DISABLE_A Value range: 0..5, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Value	BOOL	TRUE: The read input (InputNumber) has a 24V signal level. FALSE: The read input (InputNumber) has a 0V signal level.
	Inputs	WORD	Overall input status (regardless of InputNumber): I0 = Bit 0, I1 = Bit 1, I2 = Bit 2, I3 = Bit 3, I4 = Bit 4, I5 = Bit 5 Value range: 00h..3Fh, initial value: 00h.

## 8.16.2 MC\_ReadDigitalOutput\_LXM05

### Task:

Reading the drive's current output status.

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	OutputNumber	INT	Number of the output that is to be read: 0: Output 0 /NO_FAULT_OUT 1: Output 1 /ACTIVE1_OUT Value range: 0..1, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Value	BOOL	TRUE: The read output (OutputNumber) has a 24V signal level. FALSE: The read output (OutputNumber) has a 0V signal level.
	Outputs	WORD	Overall output status (regardless of OutputNumber): Output 0 = Bit 0, Output 1 = Bit 1 Value range: 00h..03Fh, initial value: 00h.

## 8.17 Error handling

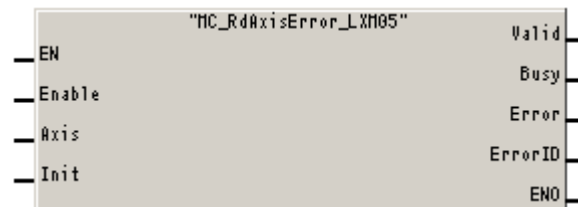
For the purpose of error handling, every block as an output *Error*, which is set if a [synchronous](#) or [asynchronous](#) error occurs. For a more detailed analysis of the error's cause, the block MC\_ReadAxisError\_LXM05 is called. By means of MC\_Reset\_LXM05, the error cell is cleared to make it available for future error messages.

### 8.17.1 MC\_ReadAxisError\_LXM05

#### Task:

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

#### Calling:



#### Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	ErrorID	INT	0: No error message in the error cell >0: Error number (see list of error numbers). Value range: 0..65535, initial value: 0.

## 8.17.2 MC\_Reset\_LXM05

### Task:

Error acknowledgement.

### Calling:



### Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Done	BOOL	TRUE: Block execution was completed successfully.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

### Operating principle:

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

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

### Note

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

## 9 Glossary

### Asynchronous errors

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

[Errorhandling](#)

### Movement profile generator

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

### Error class

The device response depends on the severity of an error:

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

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

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

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

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

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

### Error cell

The error cell contains the error code and the error class of an error that has occurred. A newly occurred error will be entered, provided that the error cell is free (i.e. equal to zero). If the error cell is occupied (i.e. not equal to zero), the previous error message will not be overwritten – instead, the new error message is ignored. The error cell is cleared by executing the block MC\_Reset\_LXM05, provided that the cause of the error has been rectified.

### Device parameter list or Object list

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

[MC\\_WriteParameter\\_LXM05](#) [MC\\_ReadParameter\\_LXM05](#)

## Inc, Inc/s

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

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

Resolution of the power amplifier: 131072 increments per revolution

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

## Scaling

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

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

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

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

Also see the device manual.

## Synchronous errors

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

[Errorhandling](#)

## usr

stands for “user-defined unit”.

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



## 10 List of error numbers

ErrorID hex	ErrorID dec	Error class	Drive error messages
16#1100	4352	0	Parameter out of permissible range
16#1101	4353	0	Parameter does not exist (index)
16#1102	4354	0	Parameter does not exist (sub-index)
16#1103	4355	0	Parameter write not permissible (READ only)
16#1104	4356	0	Write access denied (no access authorisation)
16#1106	4358	0	Command not allowed when drive is active
16#1107	4359	0	Access via other interface blocked
16#1108	4360	0	Parameter not readable (Block Upload)
16#1109	4360	0	Power fail data invalid
16#110A	4362	0	Boot loader not present
16#110B	4363	3	Initialisation error
16#1300	4864	3	Safe Standstill triggered (SAFE_DISABLE_A, SAFE_DISABLE_B)
16#1301	4865	4	SAFE_DISABLE_A and SAFE_DISABLE_B different level
16#1310	4880	3	Reference signal frequency too high
16#1603	5635	0	Capture memory occupied by other function
16#1606	5638	0	Recording still active
16#1607	5639	0	Trigger parameter for capture not defined
16#1608	5640	0	Trigger option for trigger parameter not permitted
16#1609	5641	0	No capture channel defined
16#160A	5642	0	No recorded data present
16#160B	5643	0	Parameter not recordable
16#160C	5644	1	Autotuning: Moment of inertia outside permissible range
16#160E	5646	1	Autotuning: Test movement could not be started
16#160F	5647	1	Autotuning: Power amplifier cannot be enabled
16#1610	6548	1	Autotuning: Processing discontinued
16#1611	5649	1	System error: Autotuning internal write access
16#1612	5650	1	System error: Autotuning internal read access
16#1613	5651	1	Autotuning: Max. permissible positioning range exceeded
16#1614	5652	0	Autotuning: already active
16#1617	5655	1	Autotuning: Friction or load torque too high
16#1618	5656	1	Autotuning: Optimisation aborted
16#1A00	6656	0	System error: FIFO memory overflow
16#1A01	6657	3	Motor has been changed
16#1A02	6658	3	Motor has been changed
16#1B00	6912	4	System error: Faulty parameter in motor or power amplifier
16#1B01	6913	3	User parameter for max. speed too high
16#1B02	6914	3	User parameter for max. current, holding current or Quick Stop current too high
16#2300	8960	3	Power amplifier overcurrent
16#2301	8961	3	Overcurrent in ballast resistor
16#3100	12544	par.	Phase error in mains supply
16#3200	12800	3	DC busovervoltage
16#3201	12801	3	DC busundervoltage (switch-off threshold)
16#3202	12802	2	DC busundervoltage (Quick Stop threshold)
16#3203	12803	4	Motor encoder supply voltage
16#3206	12806	0	DC busundervoltage (warning)
16#4100	16640	3	Power amplifier overtemperature
16#4101	16641	0	Warning power amplifier overtemperature

## Step 7 Motion Library for Lexium 05B and SD 328B

16#4102	16642	0	Power amplifier overload (I <sup>2</sup> t) warning
16#4200	16896	3	Device overtemperature
16#4300	17152	3	Motor overtemperature
16#4301	17153	0	Warning motor overtemperature
16#4302	17154	0	Motor overload (I <sup>2</sup> t) warning
16#4402	17410	0	Warning of overload (I <sup>2</sup> t) in ballast resistor
16#5200	20992	3	No connection to the motor encoder
16#5201	20993	4	Errors in motor encoder communication
16#5202	20994	4	Motor encoder is not supported
16#5203	20995	4	No connection to the motor encoder
16#5204	20996	3	Connection to motor encoder lost
16#5430	21552	0	System error: EEPROM read error
16#5431	21553	0	System error: EEPROM write error
16#5435	21557	0	System error: EEPROM not formatted
16#5437	21559	0	System error: EEPROM checksum error in manufacturer data
16#5438	21560	0	System error: EEPROM checksum error in user-defined parameter
16#5439	21561	0	System error: EEPROM checksum error in CAN parameter
16#543A	21562	0	System error: EEPROM hardware info invalid
16#543B	21563	0	System error: EEPROM manufacturer data invalid
16#543C	21564	0	System error: EEPROM CAN data invalid
16#543D	21565	0	System error: EEPROM user parameter invalid
16#5600	22016	3	Motor connection phase fault
16#5601	22017	4	Interruption or faulty motor encoder signals
16#5602	22018	4	Interruption or faulty motor encoder signals
16#5603	22019	4	Commutation error
16#6107	24839	0	Parameters outside value range (calculation error)
16#6108	24840	0	Function not available
16#610D	24845	0	Error in selection parameter
16#610F	24847	4	Internal timebase fault (timer 0)
16#7120	28960	4	Invalid motor data
16#7121	28961	2	System error: Errors in motor encoder communication
16#7122	28962	4	Motor data not acceptable
16#7123	28963	4	Motor current offset outside permissible range
16#7124	28964	4	System error: Motor encoder faulty
16#7200	29184	4	System error: Calibration of analog/digital converter
16#7201	29185	4	System error: Motor encoder initialising (quadrant evaluation)
16#7327	29479	4	System error: Position sensor not ready
16#7328	29480	4	Motor encoder sends: Position capture errors
16#7329	29481	0	Motor encoder sends: Warning
16#7330	29482	4	System error: Motor encoder (Hiperface)
16#7331	29483	4	System error: Motor encoder initialisation
16#7333	29485	4	System error: Discrepancy during calibration of analog/digital converter
16#7334	29486	3	System error: Analog/digital converter offset too big
16#7335	29487	0	Communication to motor encoder occupied
16#7336	29488	3	Offset with Sincos drift compensation too high
16#7337	29489	1	Offset could not be successfully written
16#7400	29696	0	System error: Illegal interrupt (XINT2)
16#7500	29952	0	RS 485/Modbus: Overrun error
16#7501	29953	0	RS 485/Modbus: Framing error
16#7502	29954	0	RS 485/Modbus: Parity error
16#7503	29955	0	RS 485/Modbus: receive error

## Step 7 Motion Library for Lexium 05B and SD 328B

16#A060	41056	2	Calculation error with electronic gearbox
16#A061	41057	2	Change in reference value with electronic gearbox too large
16#A300	41728	0	Torque ramp with HALT current active
16#A301	41729	0	Drive in status 'QuickStopActive'
16#A302	41730	1	Interruption via LIMP
16#A303	41731	1	Interruption via LIMN
16#A304	41732	1	Interruption via REF
16#A306	41734	1	Interruption by user-initiated software stop
16#A307	41735	0	Interruption by internal software stop
16#A308	41736	0	Drive in 'FAULT' status
16#A309	41737	0	Drive not in 'OperationEnable' status
16#A310	41744	0	Power amplifier not active
16#A312	41746	0	Profile generating interrupted
16#A313	41747	0	Position overrun present (pos_over=1), therefore reference point no longer defined (ref_ok=0)
16#A314	41748	0	No reference position
16#A315	41749	0	Homing active
16#A316	41750	0	Overflow on acceleration calculation
16#A317	41751	0	Drive not at standstill
16#A318	41752	0	Operating mode active ( x_end = 0)
16#A319	41753	1	Manual/Autotuning: Distance range overflow
16#A31A	41754	0	Manual/Autotuning: Amplitude/ offset set too high
16#A31B	41755	0	HALT requested
16#A31C	41756	0	Illegal position setting with software limit switch
16#A31D	41757	0	Speed range overflow (CTRL_n_max)
16#A31E	41758	1	Interruption via positive software limit switch
16#A31F	41759	1	Interruption via negativesoftware limit switch
16#A320	41760	par.	Position tracking error
16#A321	41761	0	RS 422 position interface not defined as input
16#A324	41764	1	Error when homing (additional info = detailed error number)
16#A325	41765	1	Approach limit switch not enabled
16#A326	41766	1	REF switch not found between LIMP and LIMN
16#A327	41767	1	Reference movement to REF without direction reversal, impermissible limit switch LIM activated
16#A328	41768	1	Reference movement to REF without direction reversal, overrun of LIM or REF not permissible
16#A329	41769	1	More than one signal LIMP, LIMN, REF active
16#A32A	41770	1	Ext. monitoring signal LIMP with counterclockwise rotation
16#A32B	41771	1	Ext. monitoring signal LIMN with clockwise rotation
16#A32C	41772	1	Reference movement error at REF (e.g. by impact)
16#A32D	41773	1	Reference movement error at LIMP (e.g. by impact)
16#A32E	41774	1	Reference movement error at LIMN (e.g. by impact)
16#A32F	41775	1	Index pulse not found
16#A330	41776	0	Reproducibility of the index pulse movement uncertain, index pulse motion too close to the switch
16#A331	41777	3	No run-up operating mode with local control selected
16#A332	41778	1	Error with jog (additional info = detailed error number)
16#A334	41780	2	Timeout at Standstill window monitor
16#A335	41781	1	Processing only possible in fieldbus mode
16#B100	45312	0	RS 485/Modbus: Unknown service
16#B200	45568	0	RS 485/Modbus: Protocol error
16#B201	45569	2	RS 485/Modbus: Nodeguarderror

## Step 7 Motion Library for Lexium 05B and SD 328B

16#B202	45570	0	RS 485/Modbus: Nodeguardwarning
16#B203	45571	0	RS 485/Modbus: Number of monitor objects incorrect
16#B204	45572	0	RS 485/Modbus: service too long
16#B300	45824	4	Profibus: Initialising failed
16#B301	45825	4	Profibus: Initialising failed
16#B302	45826	0	Profibus: Write access denied (incorrect job identification)
16#B303	45827	par.	Profibus: Faulty processing of process data channel
16#B304	45828	par.	Profibus: Faulty processing of process data channel
16#B305	45829	par.	Profibus: Parameter cannot be mapped to the output data frame
16#B306	45830	par.	Profibus: Faulty processing of process data channel
16#B307	45831	par.	Profibus: Faulty processing of process data channel
16#B308	45832	par.	Profibus: Parameter cannot be read
16#B309	45833	0	Profibus: Sub-index not equal to zero
16#B30A	45834	0	Profibus: Parameter does not exist
16#B30B	45835	1	Profibus: Watchdog
16#B30C	45836	1	Profibus: Motor stop via clear command of master
16#B30D	45837	0	Profibus: Parameter cannot be mapped
<b>Library error messages</b>			
16#FF00	65280		Wrong_AxisRef
16#FF01	65281		Initialization_Failed
16#FF02	65282		Wrong_Data_Length
16#FF03	65283		Timeout
16#FF04	65284		Axis_Busy
16#FF05	65285		Invalid_Parameter_Number
16#FF06	65286		Unknown_State
16#FF07	65287		Capture_Busy
16#FF08	65288		Trigger_Event_Lost
16#FF09	65289		Axis_Not_In_Standstill
16#FF0A	65290		Unknown_Device_Type
16#FF0B	65291		Wrong_Data_Struct

# 11 Parameter list for Up- and Download function

[MC UploadParameter LXM05](#)
[MC DownloadParameter LXM05](#)

Name of parameter	Profibus address	Drive
password	1034	LXM05B, SD328B
DEVcmdinterf	1282	LXM05B, SD328B
IOposInterfac	1284	LXM05B, SD328B
SPVcommutat	1290	LXM05B
BRK_trelease	1294	LXM05B, SD328B
BRK_tclose	1296	LXM05B, SD328B
RESint_ext	1298	LXM05B
SPV_Flt_AC	1300	LXM05B
SPV_Flt_pDiff	1302	LXM05B, SD328B
M_Type_Config	1306	LXM05B
PWM_fChop	1308	LXM05B
SPV_MainsVolt	1310	LXM05B
SPV_EarthFlt	1312	LXM05B
RESext_ton	1314	LXM05B
RESext_P	1316	LXM05B
RESext_R	1318	LXM05B
ESIMscale	1322	LXM05B
SPV_Flt_AC_time	1330	LXM05B
SPV_SW_Limits	1542	LXM05B, SD328B
SPVswLimPusr	1544	LXM05B, SD328B
SPVswLimNusr	1546	LXM05B, SD328B
POSscaleDenom	1550	LXM05B, SD328B
POSscaleNum	1552	LXM05B, SD328B
RAMPn_max	1554	LXM05B, SD328B
RAMPacc	1556	LXM05B, SD328B
RAMPdecel	1558	LXM05B, SD328B
POSdirOfRotat	1560	LXM05B, SD328B
RAMP_TAUjerk	1562	LXM05B, SD328B
IOsigRef	1564	LXM05B, SD328B
IOsigLimN	1566	LXM05B, SD328B
IOsigLimP	1568	LXM05B, SD328B
RAMPnstart0	1570	SD328B
RAMPquickstop	1572	SD328B
RAMP_TypeSel	1574	SD328B
SPVn_win	1576	LXM05B, SD328B
SPVn_winTime	1578	LXM05B, SD328B
AbsHomeRequest	1580	SD328B
IODirPosintf	2062	LXM05B, SD328B
SM_Type	1034	SD328B
LIM_I_maxQSTP	4362	LXM05B
LIM_I_maxHalt	4364	LXM05B
STANDp_win	4370	LXM05B

## Step 7 Motion Library for Lexium 05B and SD 328B

STANDpwinTime	4372	LXM05B
STANDpwinTout	4374	LXM05B
CTRL_I_max_fw	4376	LXM05B
CTRL_I_max	4610	LXM05B
CTRL_n_max	4612	LXM05B
CTRL_KPn	4614	LXM05B
CTRL_TNn	4616	LXM05B
CTRL_KFDn	4618	LXM05B
CTRL_KPp	4620	LXM05B
CTRL_KFPp	4624	LXM05B
CTRL_TAUref	4626	LXM05B
SPV_p_maxDiff	4636	LXM05B
CTRL_TAUiref	4640	LXM05B
CTRL_Nffreq	4642	LXM05B
CTRL_Nfdamp	4644	LXM05B
CTRL_Nfbandw	4646	LXM05B
CTRL_Pcdamp	4648	LXM05B
CTRL_Pcdelay	4650	LXM05B
CTRLS_toggle	5136	SD328B
CTRLS_MotEnc	5138	SD328B
CTRLS_I_Stand	5140	SD328B
CTRLS_I_Ramp	5142	SD328B
CTRLS_I_Const	5144	SD328B
SERPROT	5634	LXM05B, SD328B
MBbaud	5638	LXM05B, SD328B
MBPB-Adr	5640	LXM05B, SD328B
MBformat	5642	LXM05B, SD328B
MBdword_order	5646	LXM05B, SD328B
PBSafeState	6154	LXM05B, SD328B
PBPB-Adr	6156	LXM05B, SD328B
PBFItPpo	6158	LXM05B, SD328B
GEARdir_enabl	9738	LXM05B, SD328B
GEARratio	9740	LXM05B, SD328B
GEARcontrol	9744	SD328B
GEAR_n_max	9746	SD328B
GEARramp	9748	SD328B
GEARposChgMode	9750	LXM05B, SD328B
HMSrchdisusr	10266	LXM05B, SD328B
AT_J	12056	LXM05B
HMIlocked	14850	LXM05B, SD328B
HMIDispPara	14852	LXM05B, SD328B