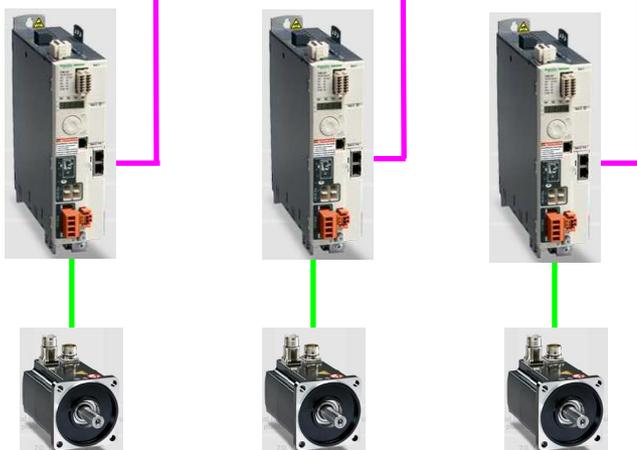
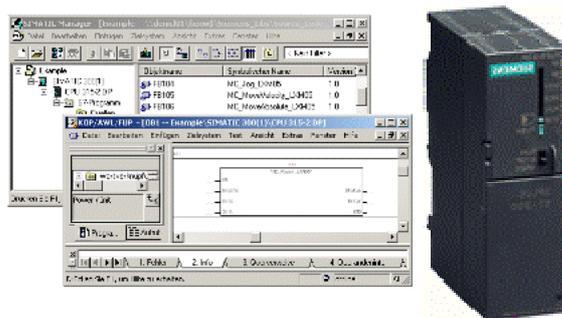


Manual Step 7 Motion Library for Lexium 32M via Profibus DP

Edition: V1.00, 08.2009



Schneider Electric Motion
Deutschland GmbH
Breslauer Str. 7
D-77933 Lahr

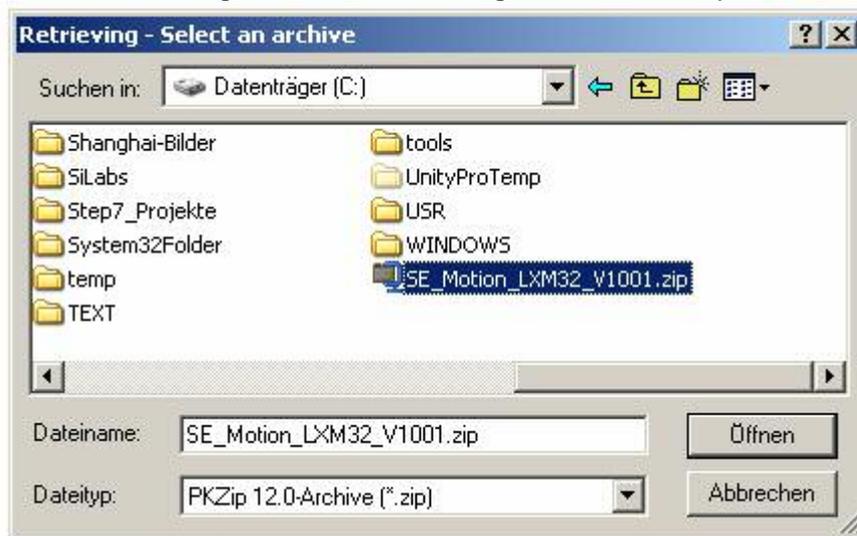
Contents

| | | |
|------|--|----|
| 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 |
| 7 | Diagnostic interrupt | 13 |
| 8 | Description of the library blocks | 14 |
| 8.1 | Copying the axis structure into the project | 14 |
| 8.2 | Creating an axis reference | 15 |
| 8.3 | Creating a block-DB for “Parameter_UP and Download” | 17 |
| 8.4 | Creating a block-DB for “Dataset_Read and Dataset_Write” | 17 |
| 9 | Library blocks..... | 18 |
| 9.1 | Basic calling procedures | 18 |
| 9.2 | Explanation of common parameters..... | 19 |
| 9.3 | Phasing diagrams | 20 |
| 9.4 | Initialisation | 21 |
| 9.5 | Jog..... | 24 |
| 9.6 | Homing..... | 27 |
| 9.7 | Torque control..... | 31 |
| 9.8 | Velocity control | 32 |
| 9.9 | Profile position mode | 34 |
| 9.10 | Electronic gear | 41 |
| 9.11 | Stopping | 47 |
| 9.12 | Fast position capture..... | 49 |
| 9.13 | Read parameter..... | 52 |
| 9.14 | Write parameter..... | 62 |
| 9.15 | Inputs/outputs | 70 |
| 9.16 | Error handling | 75 |
| 10 | Glossary | 80 |
| 11 | List of error numbers | 82 |

1 Extracting the library

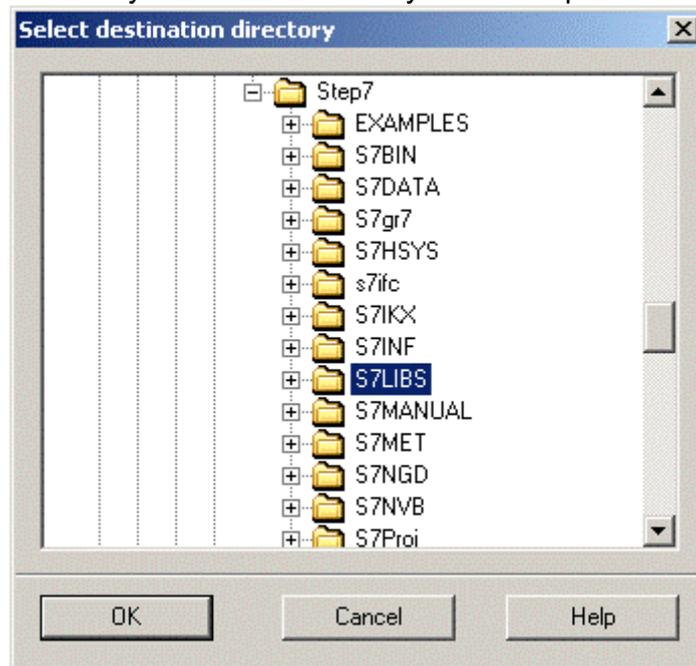
In order to use the library blocks, you must first unpack the archive “SE_Motion_LXM32_Vxxx.zip” with the Step7 software. This is done with the menu item **Retrieve** in the menu **File**.

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”.

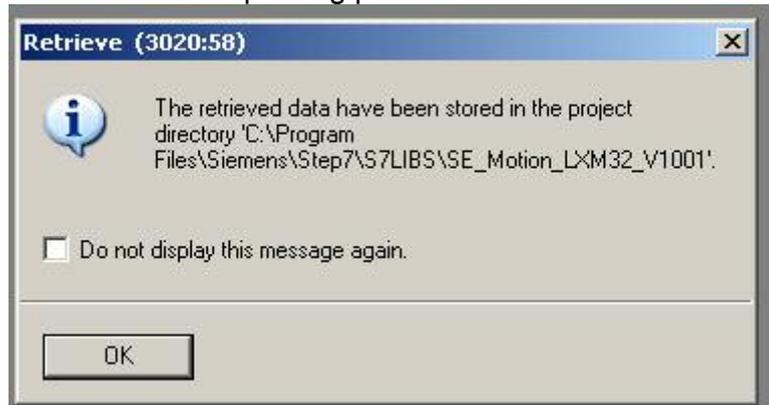
In the window shown below, you select the target directory into which the library is to be unpacked.



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

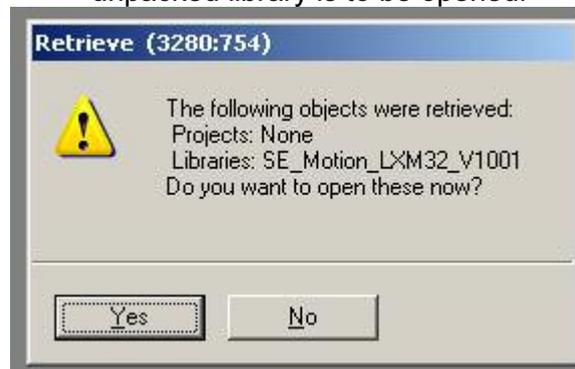
Recommendation: <Siemens directory>\Step7\S7TMP
Example: C:\Programs\Siemens\Step7\S7LIBS

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".

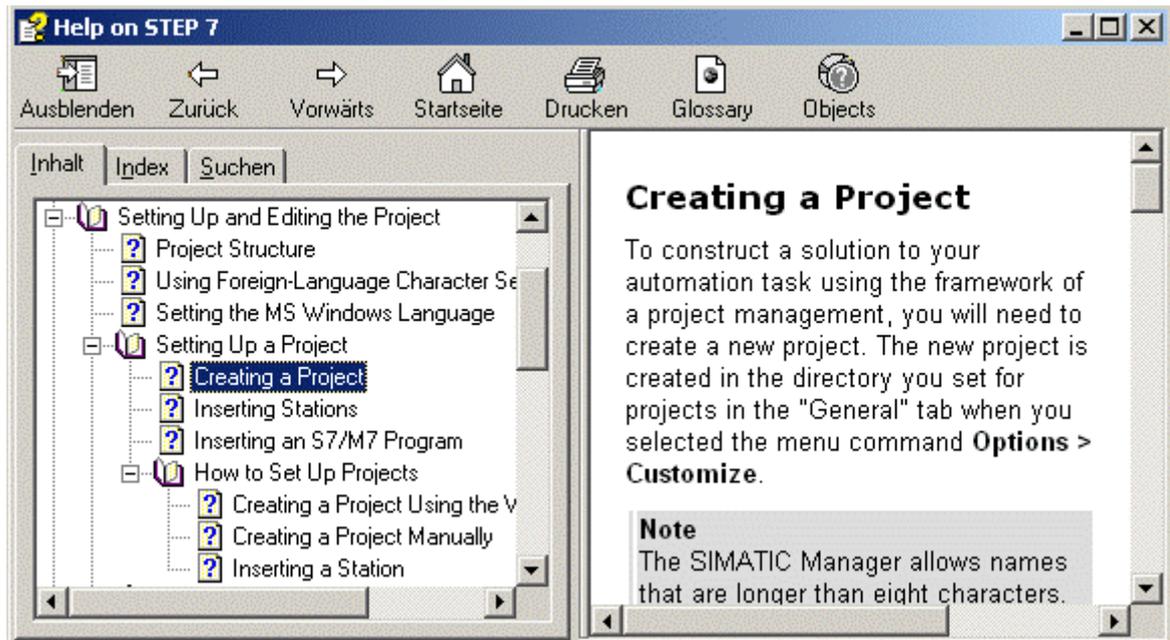
Note: Of course, you can open the library, and manually copy the relevant blocks into your application by means of the copying function of the Step 7 software.

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 (SE100B9D.GSD for LXM32). 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

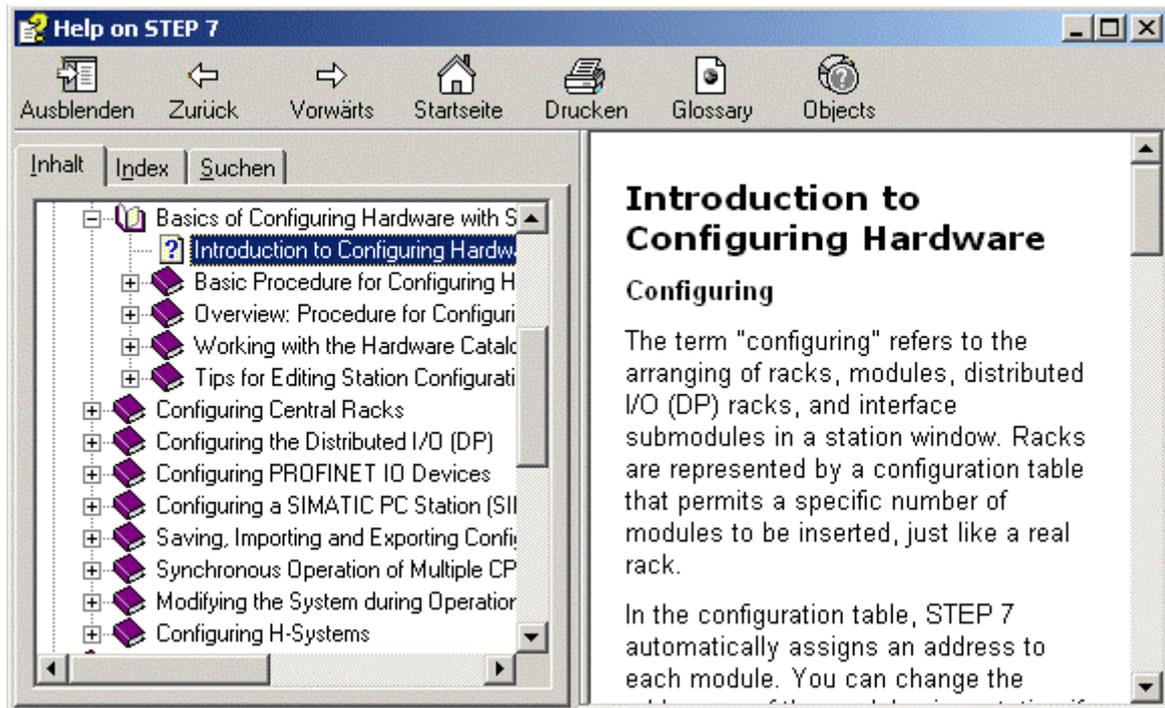
Create a new project. For this, you open the menu **File**, and select the menu item **New** or **Assistant 'New project'**. Hereby, it is assumed that you know how to create a new project, so that reference is made here to the online Help and to the documentation of Step7 and Siemens.



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

The library uses the GSD (data master file) **SE100B9D.GSD** for LXM32.

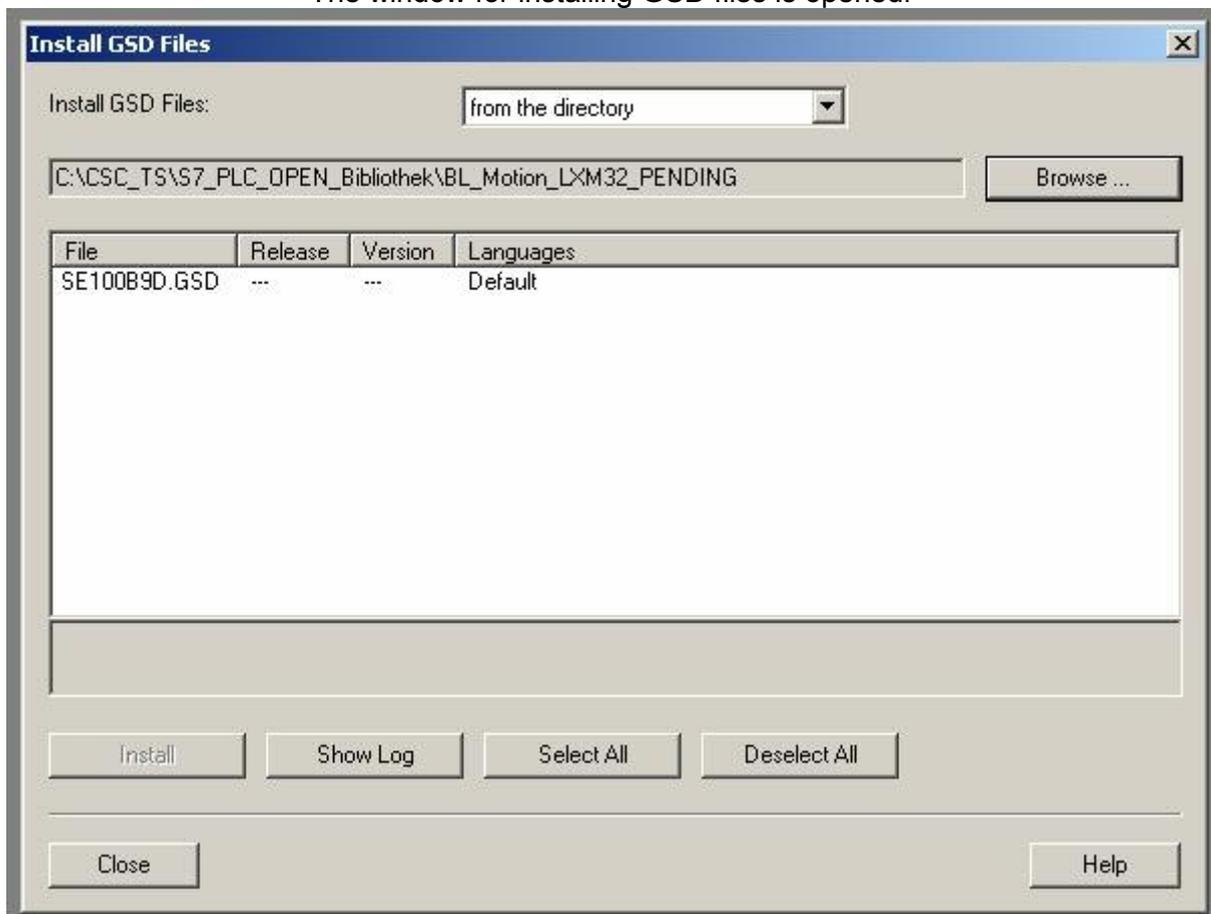
Copy the GSD and bmp-files into any directory on your hard disk.

Recommendation: <Siemens directory>\Step7\S7TMP

Example: C:\Programs\Siemens\Step7\S7LIBS

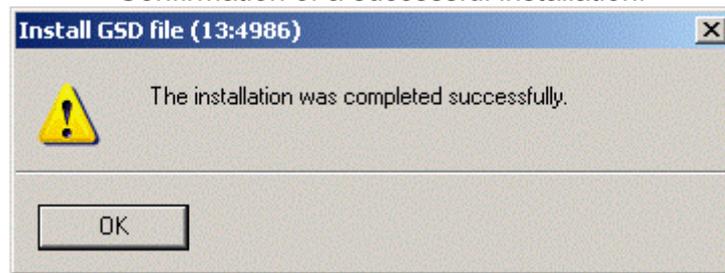
Next, you open the dialogue box in the hardware configurator for installing GSD files. This is done via the menu **Extras** and the menu item **Installing GSD files...**

The window for installing GSD files is opened.



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

Confirmation of a successful installation.

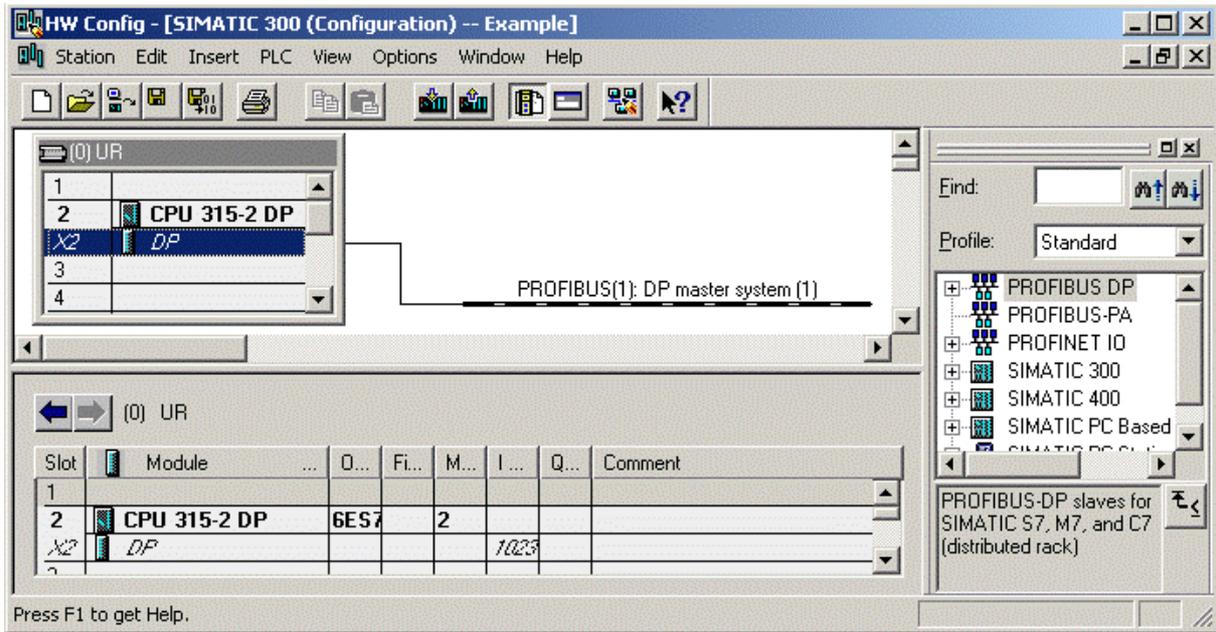


Close the confirmation message with "OK".

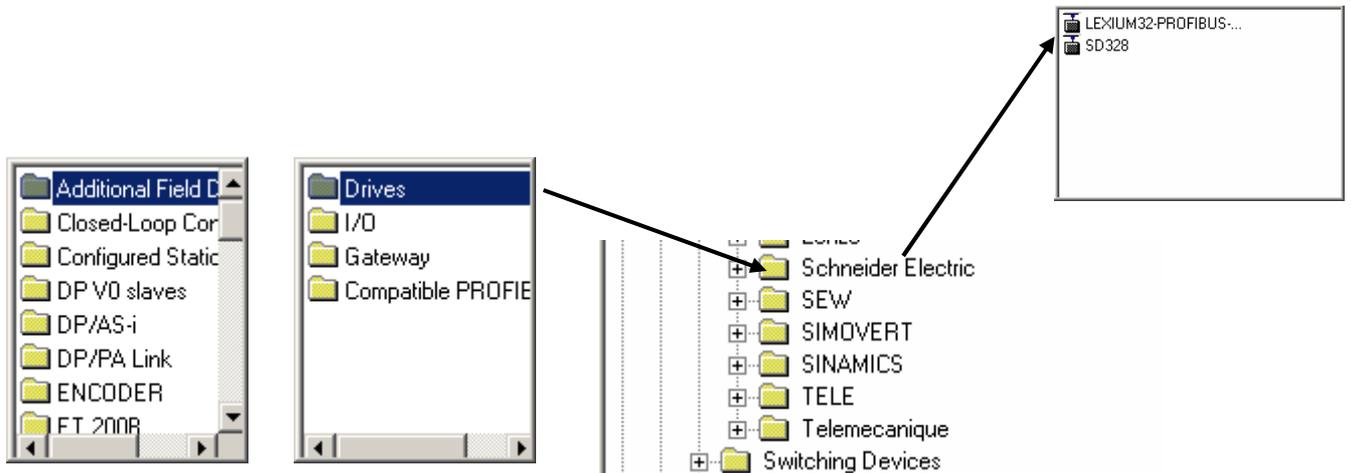
Now also close the window for installing GSDs with "Close", which returns you to the hardware configurator for the remaining hardware installation steps.

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.

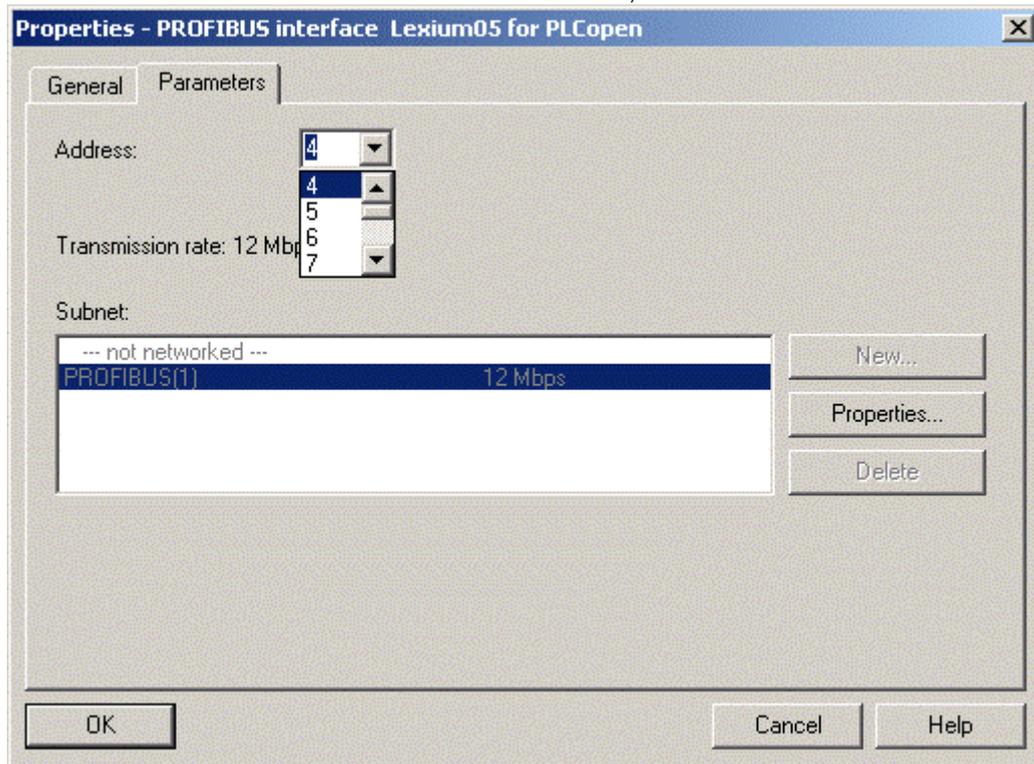


Now mark the master system in the hardware configurator, and via the menu **Insert/Insert Object...** you select the item **LEXIUM32-PROFIBUS** after clicking through the sub-menus **Additional Field Devices**, **Drives**, and **Schneider Electric**.



Step 7 Motion Library for Lexium 32

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



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

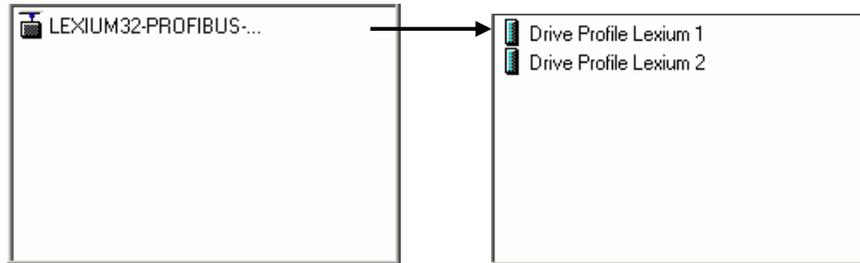


| | Order number | Firmware | Diagnostic address | Comment |
|-------------|--------------|----------|--------------------|---------|
| IBUS | | | 2047 | |

Step 7 Motion Library for Lexium 32

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 **“DriveProfileLexium1”** menu **Insert/Insert Object...**

The library only works with “DriveProfileLexium1” !



The drive is now linked into the DP master system. The library uses two communication channels for communication with the drive:

The parameter data's and the process data's in Slots 1.
 (Complete: 26 Byte Input data's and 26 Byte Output data's).

The screenshot shows the SIMATIC Manager interface. At the top, a rack configuration window shows slots 1 through 7. Slot 2 contains a "CPU 315F-2 DP" and slot X2 contains a "DP" module. A horizontal line labeled "PROFIBUS_1_Axis: DP master system (1)" connects to a slave rack labeled "(10) LEXIUM32-PROFIBUS". Below this, a detailed view of the slave rack shows a table of modules.

| Slot | DP ID | ... | Order Number / Designation | I Address | Q Address | Comment |
|------|-------|-----|----------------------------|-----------|-----------|---------|
| 1 | 195 | | Drive Profile Lexium 1 | 0..25 | 0..25 | |
| 2 | 0 | | Empty module | | | |
| 3 | 0 | | Empty module | | | |
| 4 | 0 | | Empty module | | | |
| 5 | 0 | | Empty module | | | |
| 6 | 0 | | Empty module | | | |
| 7 | 0 | | Empty module | | | |
| 8 | 0 | | Empty module | | | |
| 9 | 0 | | Empty module | | | |

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

6 Assigning the I/O addresses

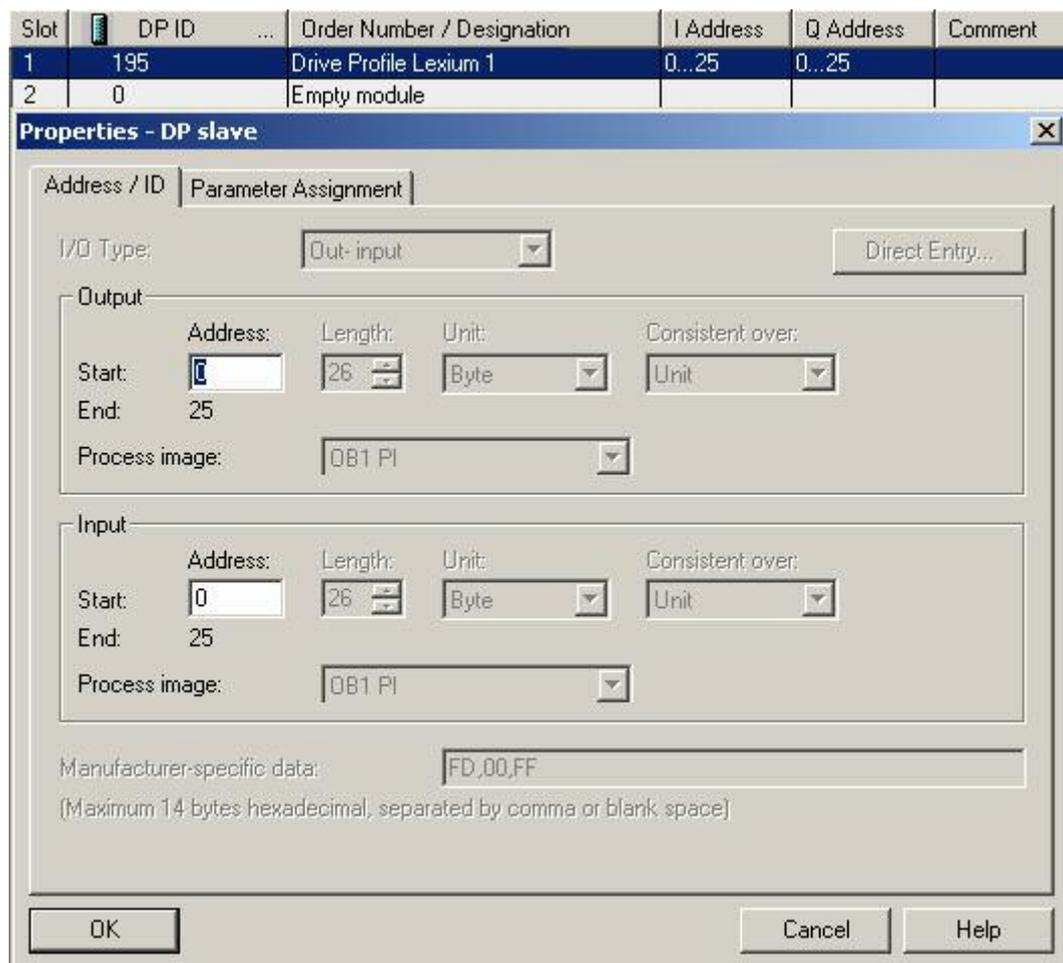
The last adjustment in the hardware configurator involves assigning the input and output addresses of the communication channels. These addresses depend on the projected CPU and on the PLC's configuration.

In order to assign the addresses 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.

You can use the default-value 0 !

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_LXM32](#) at the input "AdrParameterOut", and the starting address of the input range at the input "AdrParameterIn".



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 Diagnostic interrupt

The Lexium 32 provides the diagnostic alarms service. The Step7 hardware configuration tool of Siemens activates this function after linking a device to the Profibus. It is possible to deactivate this function if this function is not needed.

With the right mouse-button-click on the LXM32-gsd, you have to open the "Object-properties."

→ Here you can deactivate the diagnostic alarm function (the alarm OB 82 will not be called if a stop fault occurs).

The screenshot shows the hardware configuration interface. At the top, a Profibus line is labeled "PROFIBUS_1_Axis: DP master system (1)". A device icon labeled "(10) LEXIUM32-PROFIBUS" is connected to this line. A context menu is open over the device icon, listing various actions such as Copy, Paste, Replace Object..., Delete, Move, and Go To. The "Go To" option is expanded, showing "Object Properties..." with the keyboard shortcut "Alt+Return".

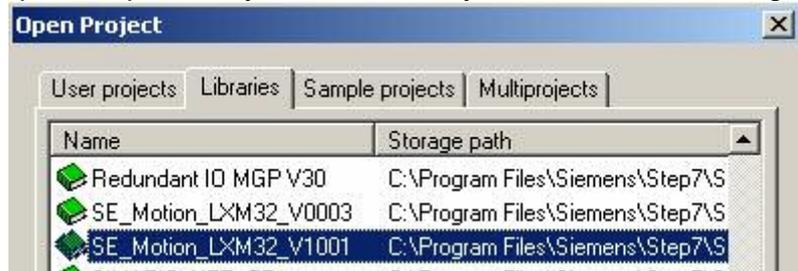
Below the context menu, the "Properties - DP slave" dialog box is open. It has two tabs: "General" and "Parameter Assignment". The "Parameter Assignment" tab is active, showing a tree view of parameters on the left and a table of values on the right. The "Diagnostic interrupt (OB82)" parameter is selected in the tree view. In the table, the value for this parameter is a checked checkbox.

| Parameters | Value |
|-----------------------------|-------------------------------------|
| Station parameters | |
| DP Interrupt Mode | DPV1 |
| DPV1 interrupts | |
| Diagnostic interrupt (OB82) | <input checked="" type="checkbox"/> |
| General DP parameters | |
| Device-specific parameters | |
| Hex parameter assignment | |

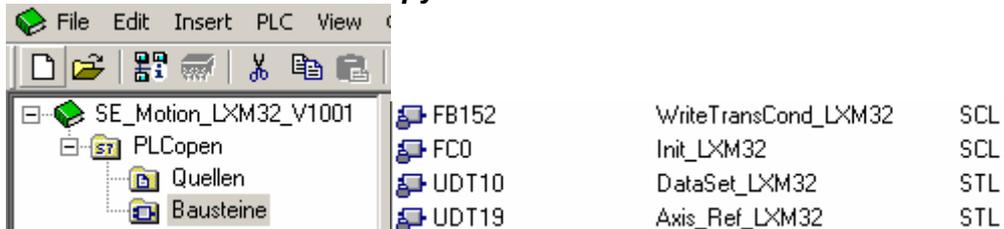
8 Description of the library blocks

8.1 Copying the axis structure into the project

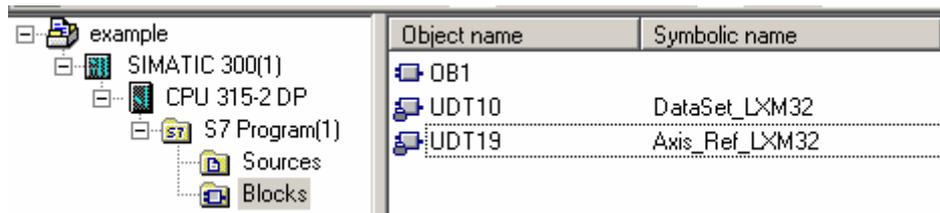
Open the previously extracted library in the SIMATIC Manager.



Next, mark the block UDT10 and UDT19, 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**.



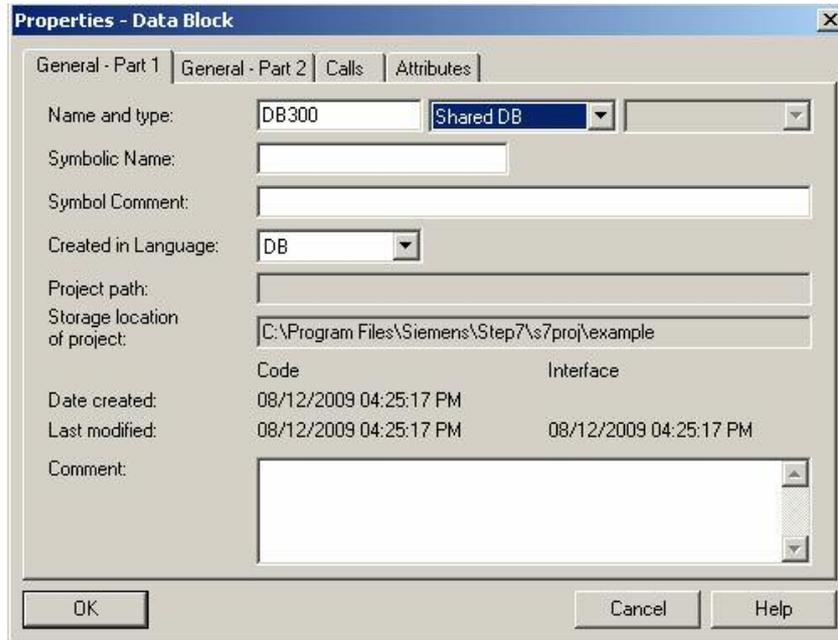
Note: Of course, you can also change the number of the UDT. This is done with the function **Rename** in the menu **Edit**.

8.2 Creating an axis reference

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

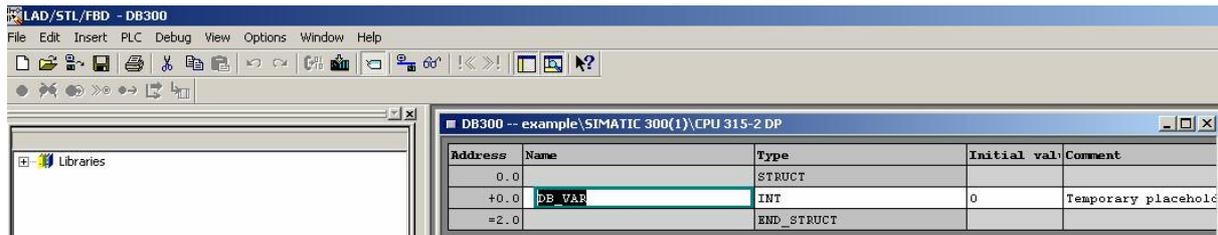
Select **Insert / S7-Block / Data Block** in the Simatic Manager.

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

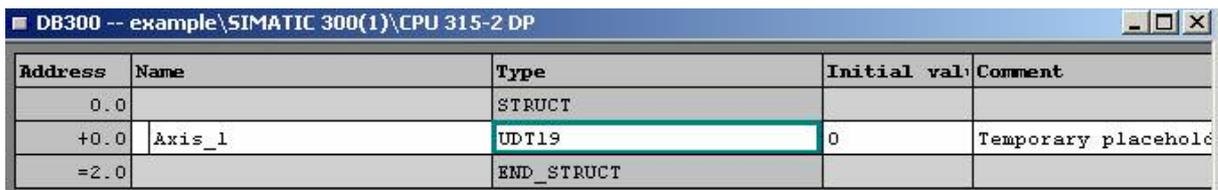


Confirm your entries with OK.

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.



You can now create the axis reference by defining a variable of the type UDT19. 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.

Step 7 Motion Library for Lexium 32

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.

In this example we created two axis-references (Axis_1, Axis_2) of type UDT19 ("Axis_ref_LXM32"):

| Address | Name | Type | Initial val. |
|---------|--------|------------------|--------------|
| 0.0 | | STRUCT | |
| +0.0 | Axis_1 | "Axis_Ref_LXM32" | |
| +180.0 | Axis_2 | "Axis_Ref_LXM32" | |
| =360.0 | | END_STRUCT | |

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

8.3 Creating a block-DB for “Parameter_UP and Download”

“FB UploadDriveParameters_LXM32” requires a Block DB of 2200 words to store the read data. This DB is then the source for the FB DownloadDriveParameter_LXM32.

| Address | Name | Type |
|---------|----------------|----------------|
| 0.0 | | STRUCT |
| +0.0 | axis_parameter | ARRAY[1..2200] |
| *2.0 | | WORD |
| =4400.0 | | END_STRUCT |

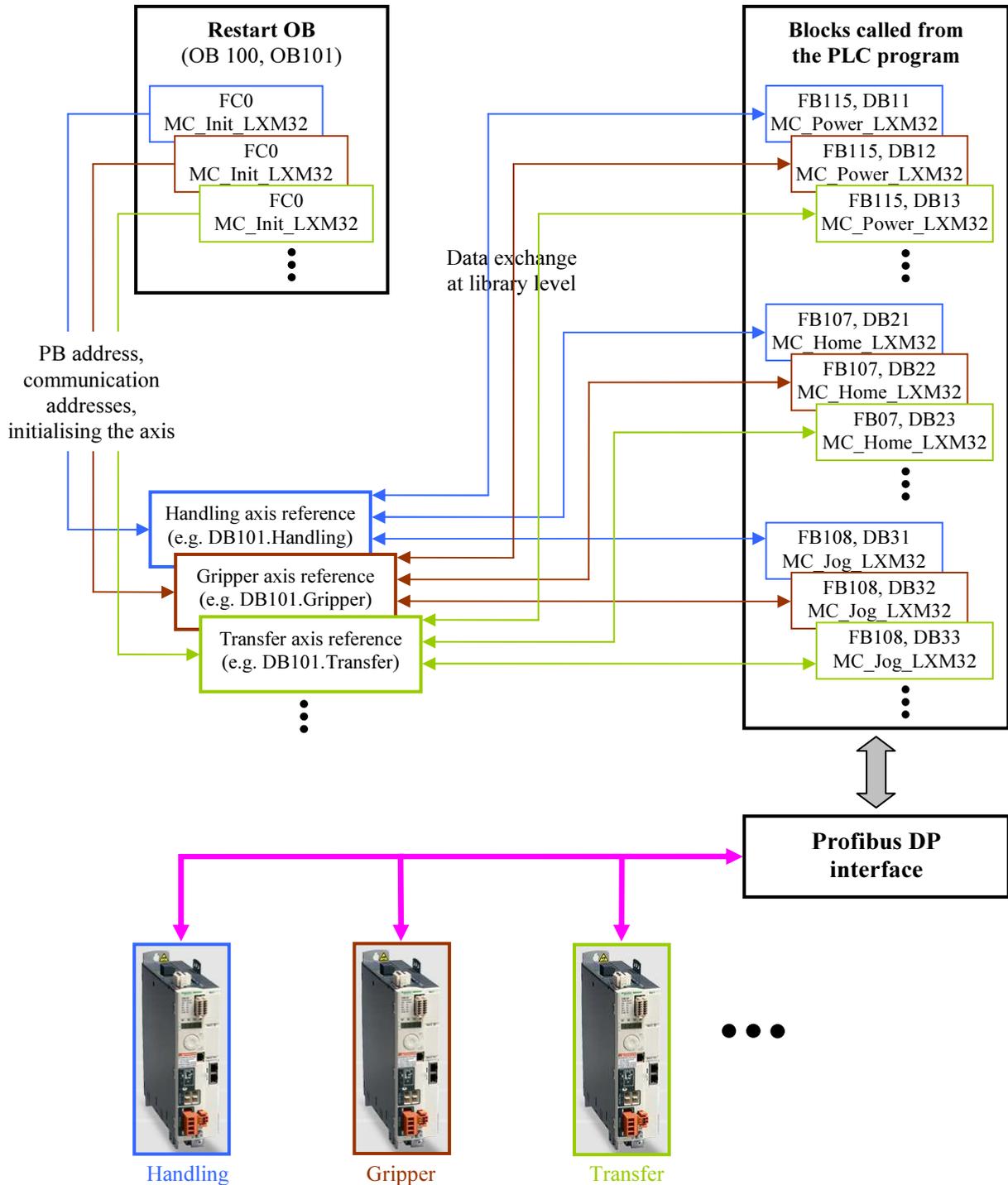
8.4 Creating a block-DB for “Dataset_Read and Dataset_Write”

“WriteDataSet” und “ReadDataSet” requires a Block DB of type UDT10.

UDT10 is the data type with the data for one dataset.

9 Library blocks

9.1 Basic calling procedures

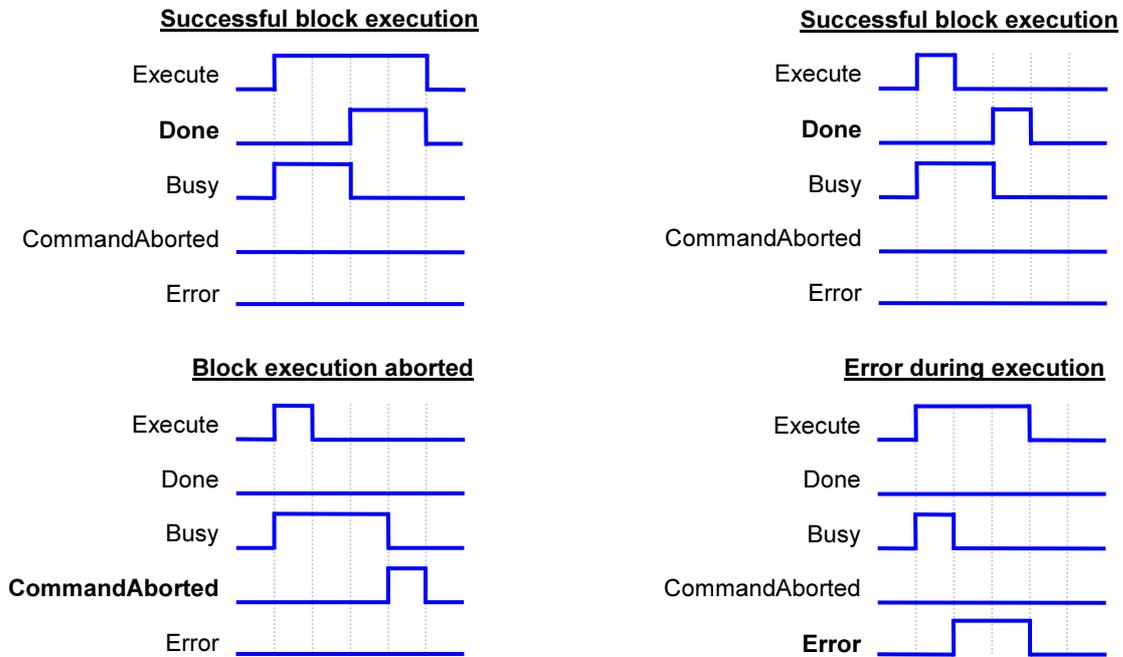


9.2 Explanation of common parameters

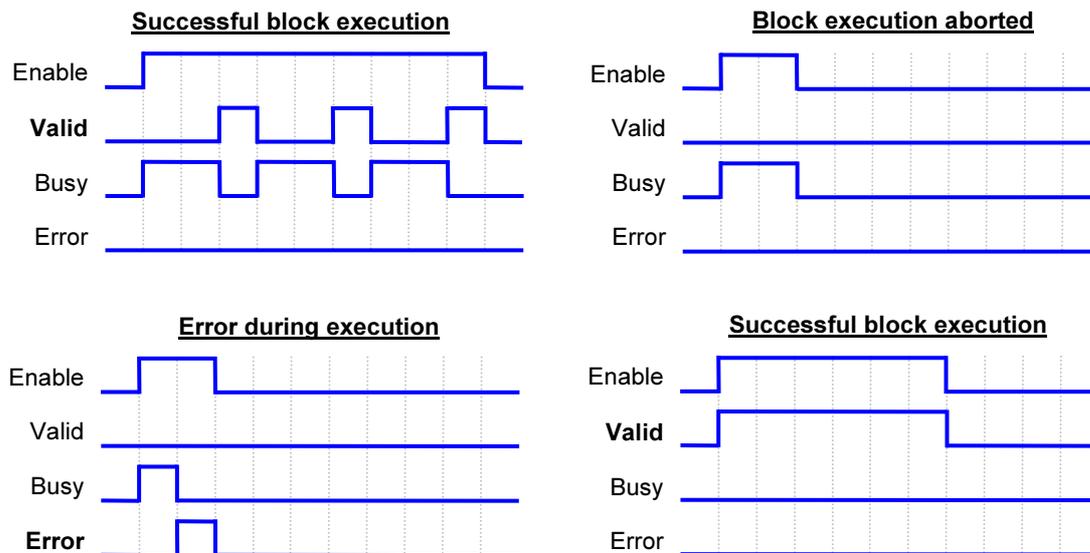
| Par. type | Parameter | Data type | Description |
|-----------|----------------|-----------|---|
| IN | Enable | BOOL | Starts (=TRUE) and stops (=FALSE) the block's execution. The block continues to be executed as long as TRUE is returned (level-sensitive). |
| | Execute | BOOL | <p>The block is executed 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 | 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. Example: Init := DBname.Axisname.Init.Ix with Ix = I0 .. I63. |

9.3 Phasing diagrams

Phasing diagrams with Execute



Phasing diagrams with Enable



9.4 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_LXM32 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 !

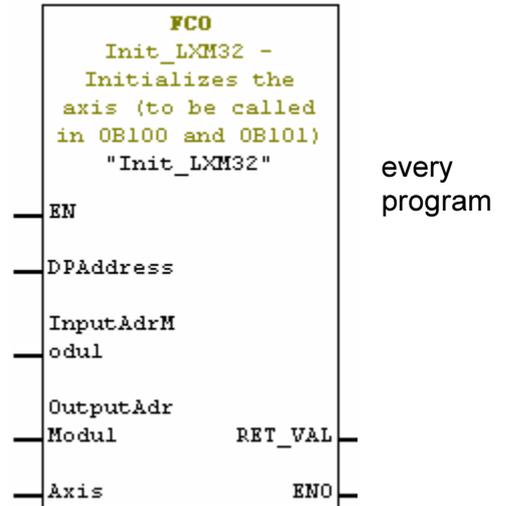
9.4.1 MC_Init_LXM32

Task:

Initialisation of an axis.

Calling:

Call the block MC_Init_LXM32 once for every axis after start of the CPU (OB100 and OB101). Depending on your 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 |
| | InputAdrModul | INT | Input address of the Parameter/Process - datachannel . |
| | OutputAdrModul | INT | Output address of the Parameter/Process - datachannel . |
| IN_OUT | Axis | STRUCT | Axis reference [AxisDB.AxisReference]. |
| OUT | Ret_Val | INT | Error number (value <>0 = error). |

Operating principle:

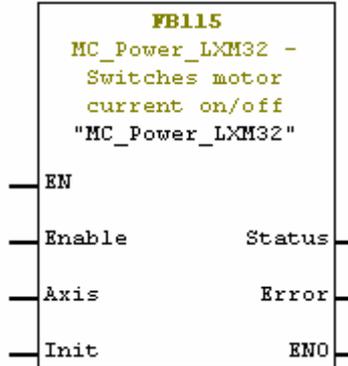
As far as possible, the transferred addresses are checked for plausibility, and entered into the axis structure of the parameter *Axis*. In addition, the initialisation bits are set in order to prepare for initialisation of the library blocks.

9.4.2 MC_Power_LXM32

Task:

Switching the motor current on/off.

Calling:



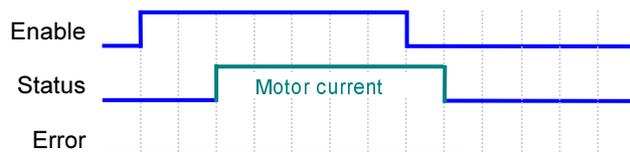
Parameter description:

| 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:



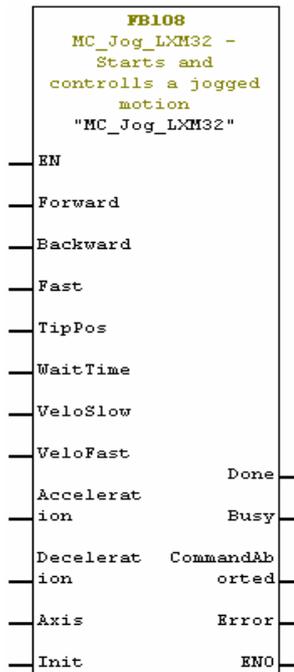
9.5 Jog

9.5.1 MC_Jog_LXM32

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 | Value range 1.. 2147483647. initial value: 20 To start continuous movement without step, parameter “JOG method” (10502) has to be set to 0 |

Step 7 Motion Library for Lexium 32

| | | | |
|--|----------|------|--|
| | 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 | DINT | Speed [rpm] for movement if Fast = FALSE. Value range: 1...2147483647; Initial value: 60. |

| | | | |
|--------|----------------|--------|---|
| IN | VeloFast | DINT | Speed [rpm] for movement if Fast = TRUE. Value range: 1...2147483647; Initial value: 180 |
| | Acceleration | DINT | Value for the acceleration ramp gradient [(10 rpm/s ²)] Value range: 1...2147483647; Initial value: 600. |
| | Deceleration | DINT | Value for the deceleration ramp gradient [(10 rpm/s ²)] Value range: 1...2147483647; Initial value: 600. |
| 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:

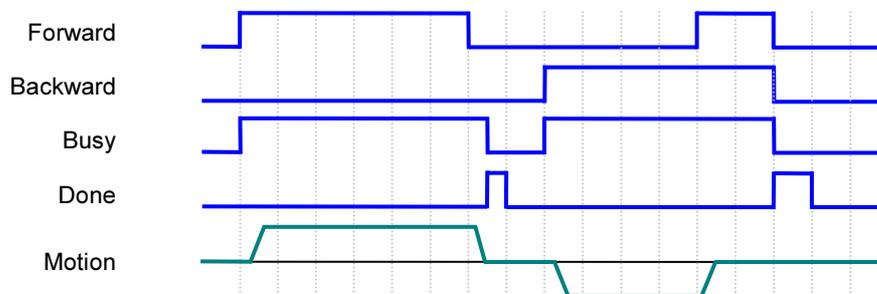
With TRUE at the *Forward* or *Backward* input, jog is started.

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:

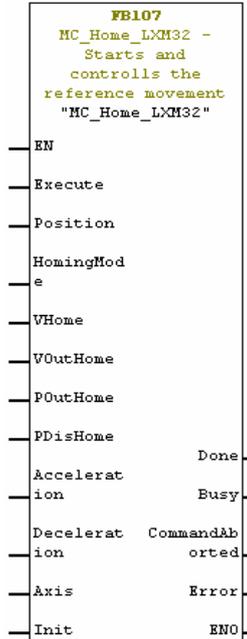


9.6.2 MC_Home_LXM32

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 |

Step 7 Motion Library for Lexium 32

| Par. type | Parameter | Data type | Description |
|-----------|-----------|-----------|--|
| | | | 29 = REF-, within REF, in direction of LIMP 30 = REF-, beyond REF, in direction of LIMP 33 = on index pulse, in direction of LIMN 34 = on index pulse, in direction of LIMP |
| | VHome | DINT | Speed for searching the limit or reference switch [rpm]. Drive stops when switching edge has been detected. Value range: 1...2147483647; Initial value: 60. |

| Par. type | Parameter | Data type | Description |
|-----------|----------------|-----------|--|
| IN | VOutHome | INT | Speed for clearance movement back to the switching edge [rpm]. The max. travel distance when searching for the switching edge can be restricted with the parameter <i>POutHome</i> . Value range: 1...2147483647; Initial value: 6. |
| | POutHome | DINT | 0: Clearing monitor switched off. >0: Run-off [usr], i.e. max. travel distance when searching for the switching edge. If the switching edge is not found in this distance, the reference movement is interrupted with an error. Value range: 0..2147483647, initial value: 0. |
| | PDisHome | DINT | Distance between the switching edge and the reference point [usr]. At end of movement, the drive moves back towards switching edge until the distance has been reached. Value range: 1..2147483647, initial value: 200. |
| | Acceleration | DINT | Value for the acceleration ramp gradient [(10 rpm/s ²)] Value range: 1...2147483647; Initial value: 600. |
| | Deceleration | DINT | Value for the acceleration ramp gradient [(10 rpm/s ²)] Value range: 1...2147483647; Initial value: 600. |
| 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*.

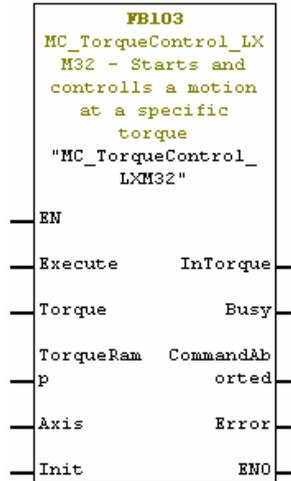
9.7 Torque control

9.7.1 MC_TorqueControl_LXM32

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. |
| | Torque | INT | Set current [0,1 % of rated torque]. Value range: -30000..+30000, initial value: 0. |
| | TorqueRamp | DINT | [0.1% of continous stall torque/s]. Value range: 1..3000000, 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 torque control operating mode, the reference value for the motor current is preset directly via the parameter *Torque*, 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:** With the device parameter CTRL_n_max (see drive manual) the maximum speed can be limited for protection. speed can be limited for protectio

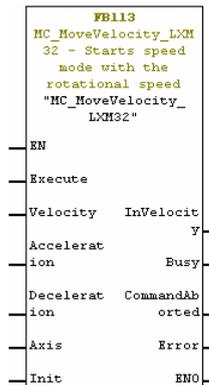
9.8 Velocity control

9.8.1 MC_Move_Velocity_LXM32

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: -2147483647...2147483647; Initial value: 0 |
| | Acceleration | DINT | Value for the acceleration ramp gradient [(10 rpm/s ²)] Value range: 1...2147483647; Initial value: 600. |
| | Deceleration | DINT | Value for the acceleration ramp gradient [(10 rpm/s ²)] Value range: 1...2147483647; Initial value: 600. |
| 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.

9.9 Profile position mode

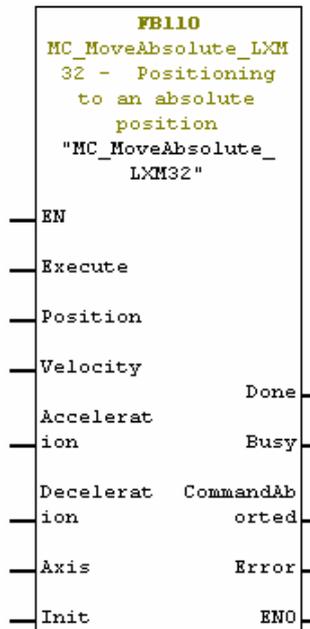
In the profile position operating mode, the motor is positioned from a point A to a point B by means of a positioning block.

9.9.1 MC_MoveAbsolute_LXM32

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 | DINT | Value for the set speed of the movement [rpm]. Value range: 1...2147483647; Initial value: 60. |
| | Acceleration | INT | Value for the acceleration ramp gradient [(10 rpm/s ²)]. Value range: 1...2147483647; Initial value: 600. |
| | Deceleration | INT | Value for the acceleration ramp gradient [(10 rpm/s ²)]. Value range: 1...2147483647; Initial value: 600. |

Step 7 Motion Library for Lexium 32

| | | | |
|--------|----------------|--------|---|
| 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. |

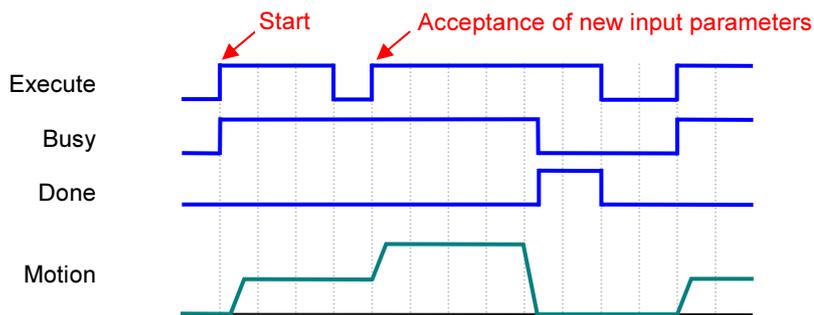
Operating principle:

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:

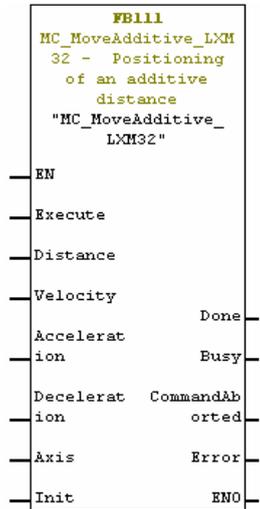


9.9.2 MC_MoveAdditive_LXM32

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 | DINT | Value for the set speed of the movement [rpm]. Value range: 1...2147483647; Initial value: 60. |
| | Acceleration | INT | Value for the acceleration ramp gradient [(10 rpm/s ²)]. Value range: 1...2147483647; Initial value: 600. |
| | Deceleration | INT | Value for the acceleration ramp gradient [(10 rpm/s ²)]. Value range: 1...2147483647; Initial value: 600. |
| 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:

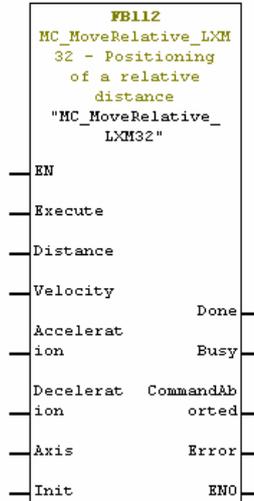
Positioning with a travel distance *Distance* referred to the current target position at speed *Velocity*. The motor moves as a function of these parameters until it reaches its target, a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC_Stop).

9.9.3 MC_MoveRelative_LXM32

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 | DINT | Value for the set speed of the movement [rpm]. Value range: 1...2147483647; Initial value: 60. |
| | Acceleration | INT | Value for the acceleration ramp gradient [(10 rpm/s ²)]. Value range: 1...2147483647; Initial value: 600. |
| | Deceleration | INT | Value for the acceleration ramp gradient [(10 rpm/s ²)]. Value range: 1...2147483647; Initial value: 600. |
| 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).

9.10 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 PTI- 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.

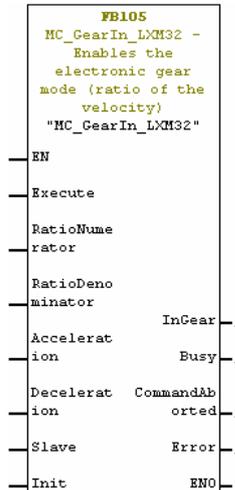
Direction enable: The direction enabling allows restriction of the movement to clockwise or counterclockwise rotation. The direction enable is activated with the device parameter `GEARdir_enabl` (see manual).

9.10.1 MC_GearIn_LXM32

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. |
| | RatioNumerator | DINT | Gear ratio numerator. Value range: -2147483648 .. 2147483647, initial value: 1. |
| | RatioDenominator | DINT | Gear ratio denominator. Value range: 1 .. 2147483647, initial value: 1. |
| IN_OUT | Slave | STRUCT | Axis reference [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:

Step 7 Motion Library for Lexium 32

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*).

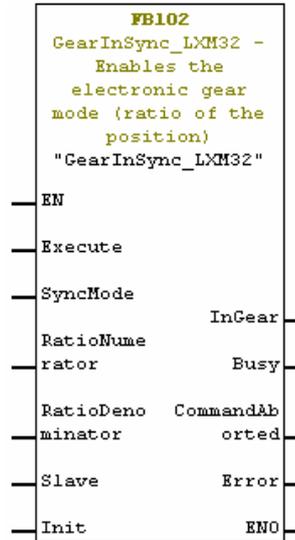
The motor moves as a function of this setpoint until it receives a new gear ratio. Operation is terminated by execution of the block MC_GearOut_LXM32 or by execution of another block (e.g. MC_Stop).

9.10.2 MC_GearInSync_LXM32

Task:

Starts an electronic gear mode with position synchronization.
 When the gear mode is active in the drive, *InGear* is set for one call of the function block.

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. |
| | RatioNumerator | DINT | Gear ratio numerator. Value range: -2147483648 .. 2147483647, initial value: 1. |
| | RatioDenominator | DINT | Gear ratio denominator. Value range: 1 .. 2147483647, initial value: 1. |
| IN_OUT | Slave | STRUCT | Axis reference [AxisDB.AxisReference]. |
| | Init | BOOL | Initialisation bit [Init.I0 .. Init.I63]. |
| OUT | InGear | BOOL | TRUE: The Gear ratio is 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:

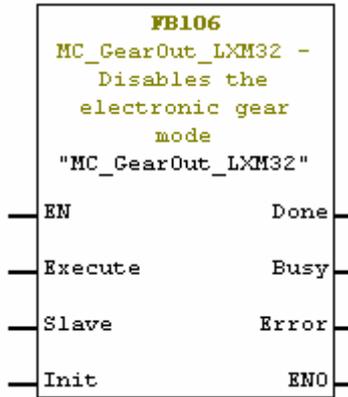
Starts an electronic gear mode with position synchronization. When the gear mode is active in the drive, *InGear* is set for one call of the function block.

9.10.3 MC_GearOut_LXM32

Task:

Switch-off of the electronic gear operating mode startet with: "MC_GearIn", or "MC_GearInSync".

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. |
| IN_OUT | Slave | 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 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_LXM32, which in turn signals CommandAborted = TRUE.

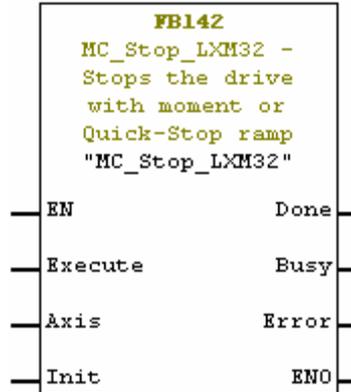
9.11 Stopping

9.11.1 MC_Stop_LXM32

Task:

Stopping the motor with Quick Stop 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. |
| 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:

This Function Block commands a controlled motion stop. It aborts any ongoing Function Block execution. After the axis has reached velocity zero, the *Done* output is set to TRUE. The axis remains in the state "Stopping" as long as input *Execute* is still TRUE or velocity zero is not yet reached. As soon as *Done* is SET and *Execute* is FALSE the axis goes to state "StandStill". Only then a new movement can be executed.

Note:

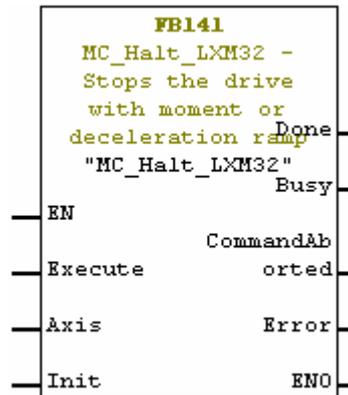
- **MC_Stop_xxx** is used to stop the axis under exceptional conditions.
- The motor is stopped with the Quick Stop ramp of the drive.
- The Quick Stop ramp of the drive can be configured with "MC_SetStopRamp".

9.11.2 MC_Halt_LXM32

Task:

This function block commands a controlled motion stop. It aborts any ongoing function block execution.

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. |
| 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: the execution was aborted by another block. FALSE: the execution was not (yet) aborted. |
| | Error | BOOL | TRUE: An error has occurred during execution. |

9.12 Fast position capture

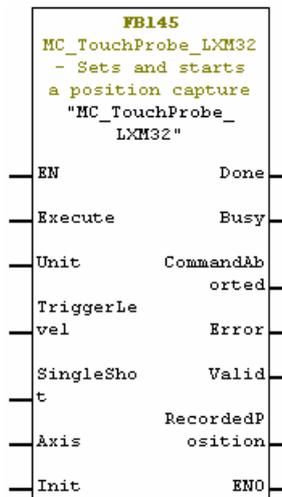
The motor position can be captured with a precision less than 10 µs delay by means of 2 parametrisable channels. Two signal inputs CAP1 and CPA2 are available for capturing the trigger signals.

9.12.1 MC_TouchProbe_LXM32

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. |
| | Unit | 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 (*Unit*).

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*).

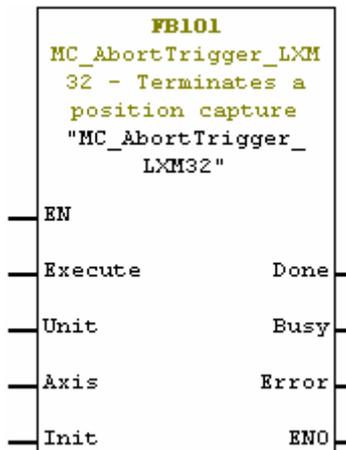
In case of continuous position capture, the block signals a *Valid* = TRUE and the detected position (*RecordedPosition*) with every occurring edge (*TriggerLevel*), whereby the previous position is overwritten. The block does not terminate itself – it can only be interrupted with MC_AbortTrigger_LXM32.

9.12.2 MC_AbortTrigger_LXM32

Task:

Terminating an active position capture.

Calling:



Parameter description:

| Par. type | Parameter | Data type | Description |
|-----------|-----------|-----------|-------------|
|-----------|-----------|-----------|-------------|

Step 7 Motion Library for Lexium 32

| | | | |
|--------|---------|--------|--|
| IN | Execute | BOOL | FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution. |
| | Unit | 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_LXM32 signals *CommandAborted* = TRUE.

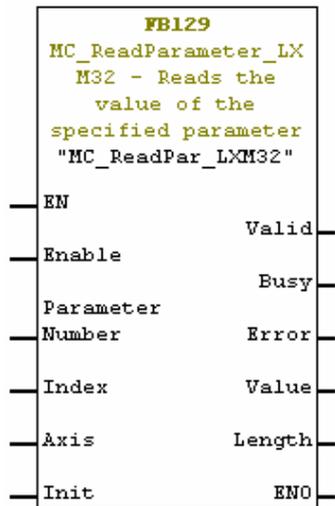
9.13 Read parameter

9.13.1 MC_ReadParameter_LXM32

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 | 1000: 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]. |

Step 7 Motion Library for Lexium 32

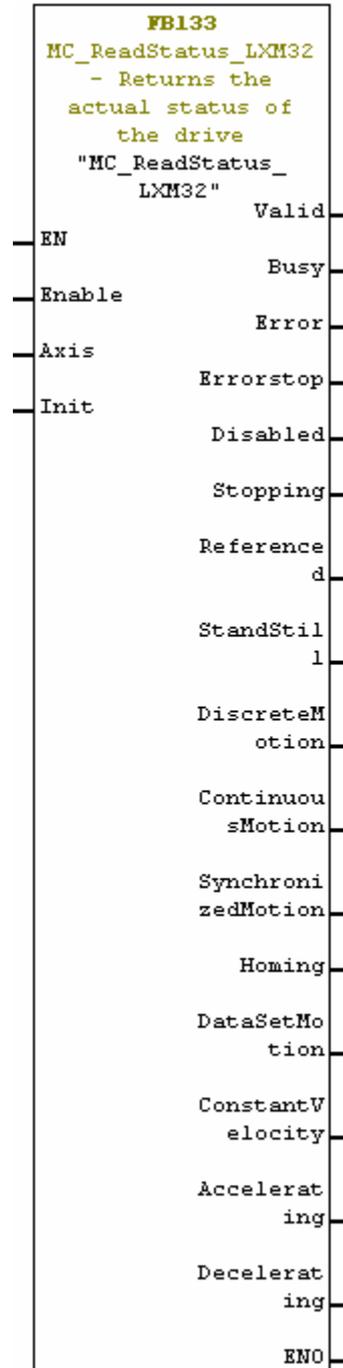
| | | | |
|-----|--------|------|---|
| OUT | Valid | BOOL | TRUE: The value to be read is available. |
| | Busy | BOOL | TRUE: Block is being executed. |
| | Error | BOOL | TRUE: An error has occurred during execution. |
| | Value | DINT | Value of the parameter to be read. Value range: -2147483648..2147483647, initial value: 0. |
| | Length | INT | Length of the parameter to be read [bytes]. Value range: 1..4, initial value: 0. |

9.13.2 MC_ReadStatus_LXM32

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_LXM32. |
| | 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. |
| OUT | ContinuousMotion | BOOL | TRUE: The drive is in an operating mode without a defined target position (e.g. MC_MoveVelocity_LXM32, MC_Jog_LXM32). |
| | 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.

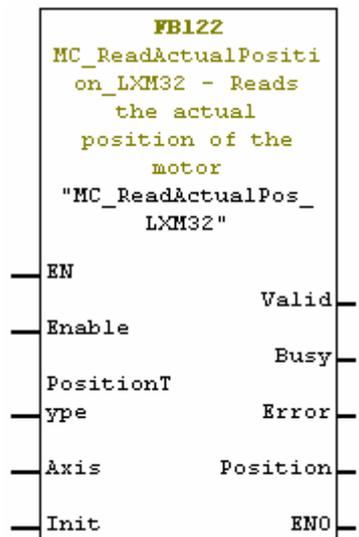
The same applies for the movement conditions *ConstantVelocity*, *Accelerating*, and *Decelerating*.

9.13.3 MC_ReadActualPosition_LXM32

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. |
| | PositionType | INT | Value range: 0...7, Initial value: 0 Selection of the position to be read from the drive: 0: actual motor position in user units (usr) Drive parameter: <code>_p_actusr</code> 1: actual motor position in internal units (Inc) Drive parameter: <code>_p_act</code> 2: reference position in user units [usr] Drive parameter: <code>_p_refusr</code> 3: reference position in internal units [inc] Drive parameter: <code>_p_ref</code> 4: actual position on external encoder interface in user units [usr] Drive parameter: <code>_p_actExtEncUsr</code> 5: actual position on external encoder interface in internal units [inc] Drive parameter: <code>_p_actExtEnc</code> 6: actual position of profile generator in user units [usr] Drive parameter: <code>_p_actRAMPusr</code> 7: actual position at position interface in internal units [inc] Drive parameter: <code>_p_PTI_act</code> |

Step 7 Motion Library for Lexium 32

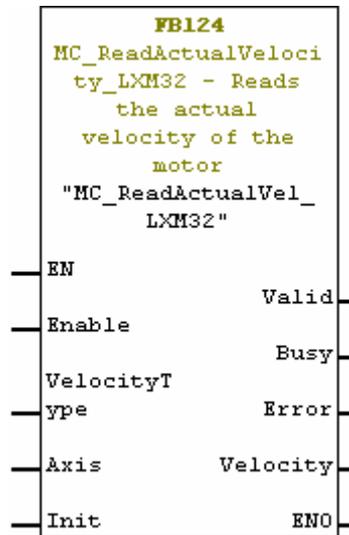
| | | | |
|-----|----------|------|---|
| | Init | BOOL | Initialisation bit [Init.I0 .. Init.I63]. |
| OUT | Valid | BOOL | TRUE: The value to be read is available. |
| | Busy | BOOL | TRUE: Block is being executed. |
| | Error | BOOL | TRUE: An error has occurred during execution. |
| | Position | DINT | Motor's current actual position In usr or Inc depending on the selected position to be read with <i>PositionType</i> input. |

9.13.4 MC_ReadActualVelocity_LXM32

Task:

Reading the motor's current speed in rpm or Inc/s.

Calling:



Parameter description:

| Par. type | Parameter | Data type | Description |
|-----------|--------------|-----------|---|
| IN | Enable | BOOL | FALSE: Terminates block execution. TRUE: Starts block execution. |
| | VelocityType | INT | Value range: 0...3, initial value 0. Selection of the velocity to be read from the drive: 0: actual motor speed in internal units [rpm] Drive parameter: <code>_n_act</code> 1: reference speed in internal units [rpm] Drive parameter: <code>_n_ref</code> 2: actual speed of profile generator in internal units [rpm] Drive parameter: <code>_RAMP_v_act</code> 3: actual velocity at PTI interface Inc/s Drive parameter: <code>_v_PTI_act</code> |
| 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. |
| | Velocity | DINT | In [rpm] or [Inc/s] depending on the selected velocity to be read |

Step 7 Motion Library for Lexium 32

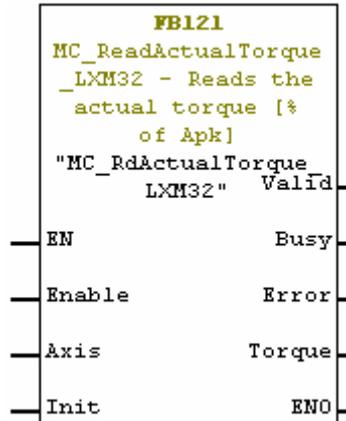
| | | | |
|--|--|--|--------------------------|
| | | | with VelocityType input. |
|--|--|--|--------------------------|

9.13.5 MC_ReadActualTorque_LXM32

Task:

Returns the value of the actual torque.

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 value to be read is available. |
| | Busy | BOOL | TRUE: Block is being executed. |
| | Error | BOOL | TRUE: An error has occurred during execution. |
| | Torque | INT | The value corresponds to 0.1% of the drive rated Torque. Example: if <i>Torque</i> = 300 then it means that the torque is equal to 30% of the rated torque of the drive. Drive rated Torque is in the drive with parameter: continuous stall torque of motor (<i>_M_M_0</i>). |

9.13.6 MC_UploadParameter_LXM32

Task:

Stores all the drive parameters in a DB. The DB requires 2200 word.

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. |
| | Parameters | DB | DB with a size of 2200 word |
| 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. |

Note:

With the two blocks MC_UploadParameter_LXM32 and MC_DownloadParameter_LXM032, a defective device can be exchanged without a special tool to parameterize the device.

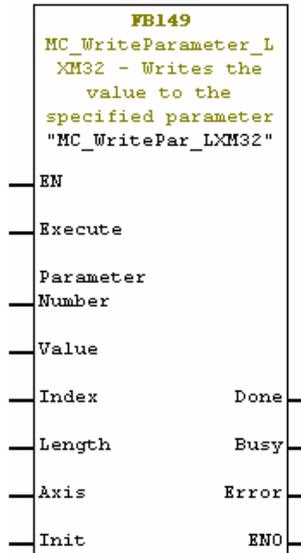
9.14 Write parameter

9.14.1 MC_WriteParameter_LXM32

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 | UINT | 1000: 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..65535, initial value: 0. |
| | Value | DINT | Value of the parameter to be written. Value range: -2147483648..2147483647, initial value: 0. |
| | Index | UINT | Index of the Object to be written – the Objects are listed in the manual with their index. Only valid with <i>ParameterNumber</i> = 1000. Value range: 0..65535, initial value: 0. |

Step 7 Motion Library for Lexium 32

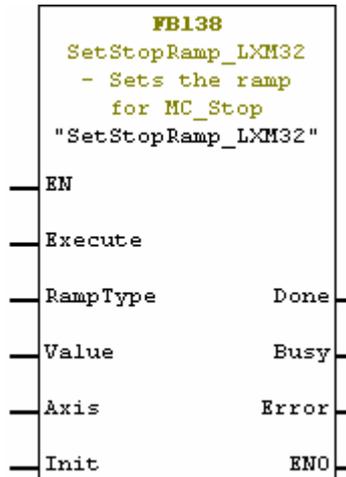
| | | | |
|--------|--------|--------|--|
| | Length | UINT | Length of the parameter to be written [bytes]. Value range: 1..4, initial value: 0. |
| IN_OUT | Axis | STRUCT | Axis reference [AxisDB.AxisReference]. |
| | Init | BOOL | Initialisation bit [Init.I0 .. Init.I63]. |
| OUT | Done | BOOL | TRUE: Block execution was completed successfully. |
| | Busy | BOOL | TRUE: Block is being executed. |
| | Error | BOOL | TRUE: An error has occurred during execution. |

9.14.2 MC_SetStop_Ramp

Task:

Configuration of the deceleration ramp used for MC_Stop_xxx.

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. | | | | | | | | | | | | | | | |
| | RampType | BOOL | FALSE: MC_Stop_xxx decelerate with a deceleration ramp. TRUE: MC_Stop_xxx decelerate with a current ramp. | | | | | | | | | | | | | | | |
| | Value | DINT | Value of the ramp used for MC_Stop_xxx block. Value depends on the type of ramp configured with <i>RampType</i> : If <i>RampType</i> = FALSE then <i>Value</i> is the deceleration ramp: <table border="1" data-bbox="671 1688 1390 1834"> <thead> <tr> <th>Drive Parameter</th> <th>Units</th> <th>Value range</th> <th>Initial value</th> </tr> </thead> <tbody> <tr> <td>RAMPquickstop</td> <td>(usr_a)</td> <td>200...2147483647</td> <td>6000</td> </tr> </tbody> </table> If <i>RampType</i> = TRUE then <i>Value</i> is a current ramp: <table border="1" data-bbox="671 1892 1347 2036"> <thead> <tr> <th>Drive Parameter</th> <th>Units</th> <th>Value range</th> <th>Initial value</th> </tr> </thead> <tbody> <tr> <td>LIM_I_maxQSTP</td> <td>[Arms]</td> <td>1...30000</td> <td>-</td> </tr> </tbody> </table> | Drive Parameter | Units | Value range | Initial value | RAMPquickstop | (usr_a) | 200...2147483647 | 6000 | Drive Parameter | Units | Value range | Initial value | LIM_I_maxQSTP | [Arms] | 1...30000 |
| Drive Parameter | Units | Value range | Initial value | | | | | | | | | | | | | | | |
| RAMPquickstop | (usr_a) | 200...2147483647 | 6000 | | | | | | | | | | | | | | | |
| Drive Parameter | Units | Value range | Initial value | | | | | | | | | | | | | | | |
| LIM_I_maxQSTP | [Arms] | 1...30000 | - | | | | | | | | | | | | | | | |

Step 7 Motion Library for Lexium 32

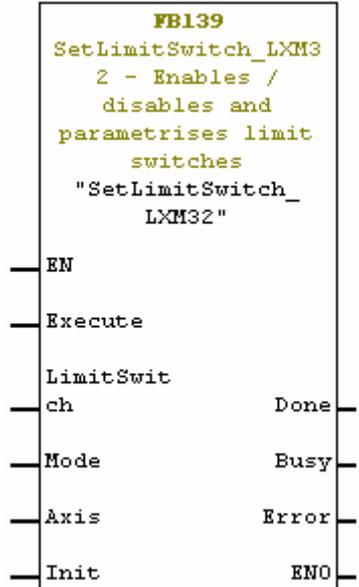
| | | | |
|--------|-------|--------|---|
| IN_OUT | Axis | STRUCT | Axis reference [AxisDB.AxisReference]. |
| | Init | BOOL | Initialisation bit [Init.I0 .. Init.I63]. |
| OUT | Done | BOOL | TRUE: Block execution was completed successfully. |
| | Busy | BOOL | TRUE: Block is being executed. |
| | Error | BOOL | TRUE: An error has occurred during execution. |

9.14.3 MC_SetLimitSwitch_LXM32

Task:

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

Calling:



Parameter description:

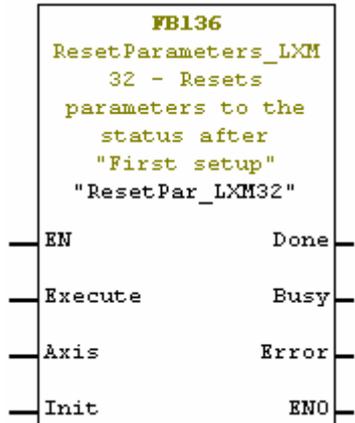
| Par. type | Parameter | Data type | Description |
|-----------|-------------|-----------|--|
| IN | Execute | BOOL | FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution. |
| | LimitSwitch | INT | 1: Positive limit switch LIMP 2: Negative limit 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. |

9.14.4 MC_ResetParameters_LXM32

Task:

Set back the persistent parameters to the default values.

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. |
| 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:

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:

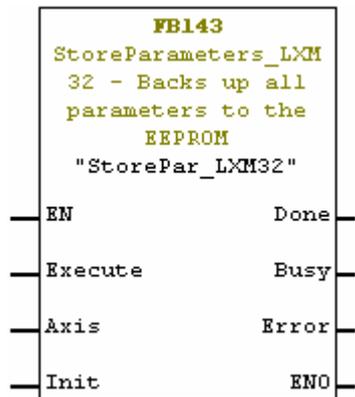
All the user set parameters will be lost if no backup has been made onto the data carrier with the commissioning software.

9.14.5 MC_StoreParameters_LXM32

Task:

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

Calling:



Parameter description:

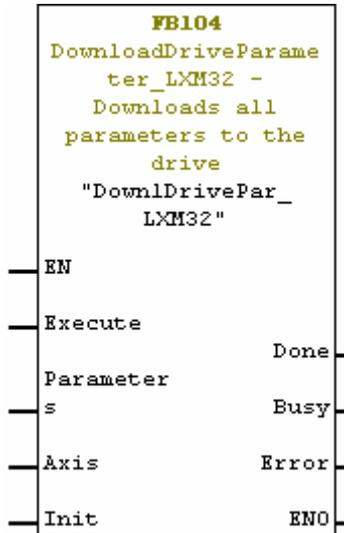
| Par. type | Parameter | Data type | Description |
|-----------|-----------|-----------|--|
| IN | Execute | BOOL | FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution. |
| IN_OUT | Axis | STRUCT | Axis reference [AxisDB.AxisReference]. |
| | Init | 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. |

9.14.6 MC_DownloadParameter_LXM32

Task:

Writing all variable parameters into 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. |
| | Parameters | DB | DB with a size of 2200 word |
| 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. |

Note:

It is recommended to make an upload of all parameters with the function block MC_UploadParameters_LXM032 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.

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

9.15 Inputs/outputs

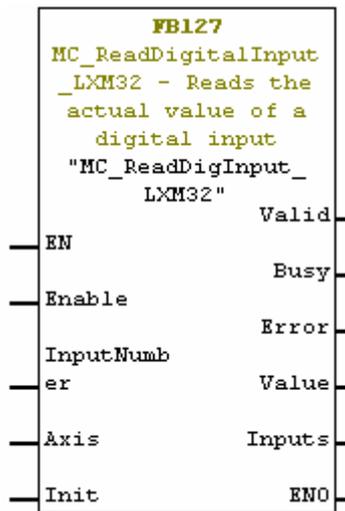
Apart from the process image, in which the digital inputs and outputs of the target system are displayed, other blocks are available, which provide access to the digital inputs/outputs of every drive in the system. The 24V signal interface of the drive provides 6 inputs and 3 outputs, which are assigned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

9.15.1 MC_ReadDigitalInput_LXM32

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 1: I1 2: I2 3: I3 4: I4 5: I5 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. |

Step 7 Motion Library for Lexium 32

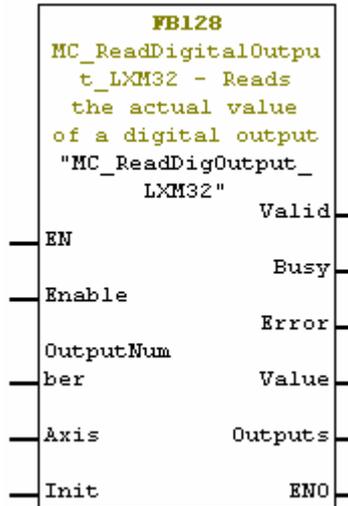
| | | | |
|--|--------|------|---|
| | Busy | BOOL | TRUE: Block is being executed. |
| | Error | BOOL | TRUE: An error has occurred during execution. |
| | Value | BOOL | TRUE: The read input (InputNumber) has a 24V signal level. FALSE: The read input (InputNumber) has a 0V signal level. |
| | Inputs | 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. |

9.15.2 MC_ReadDigitalOutput_LXM32

Task:

Reading the drive's current output status.

Calling:



Parameter description:

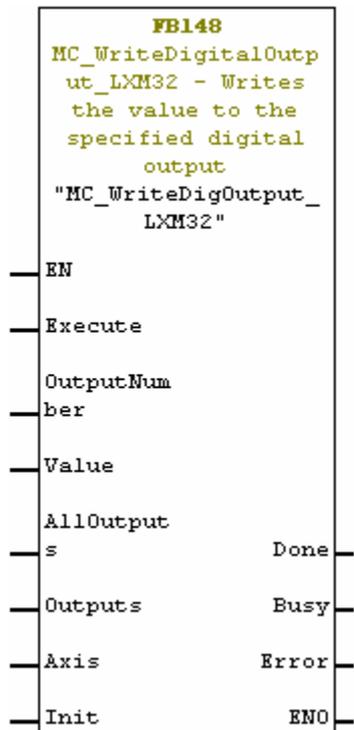
| Par. type | Parameter | Data type | Description |
|-----------|--------------|-----------|--|
| IN | Enable | BOOL | FALSE: Terminates block execution. TRUE: Starts block execution. |
| | OutputNumber | INT | Number of the output that is to be read: 0: Output 0 1: Output 1 2: Output 2 Value range: 0..2, 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. |

9.15.3 MC_WriteDigitalOutput_LXM32

Task:

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

Calling:



Parameter description:

| Par. type | Parameter | Data type | Description | | | |
|-----------|----------------|-----------|---|------|----------------|------|
| IN | Execute | BOOL | FALSE: Terminates block execution. TRUE: Starts block execution. | | | |
| | AllOutputs | BOOL | FALSE: output referenced by <i>OutputNumber</i> is written to <i>Value</i> . TRUE: all outputs are written following <i>Outputs</i> value. | | | |
| | Outputs | WORD | Value of the outputs to be written when <i>AllOutputs</i> is true. <table border="1" style="margin-left: 20px;"> <tr> <td>Bit0</td> <td>= value of DQ1</td> </tr> <tr> <td>Bit1</td> <td>= value of DQ2</td> </tr> </table> | Bit0 | = value of DQ1 | Bit1 |
| Bit0 | = value of DQ1 | | | | | |
| Bit1 | = value of DQ2 | | | | | |

Step 7 Motion Library for Lexium 32

| | | | |
|--------|--------------|--------|--|
| | | | Bit2 = value of DQ3 |
| | OutputNumber | INT | Number of the output that is to be read: 0: Output 0 1: Output 1 2: Output 2 Value range: 0..2, initial value: 0. |
| IN_OUT | Axis | STRUCT | Axis reference [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. |

9.16 Error handling

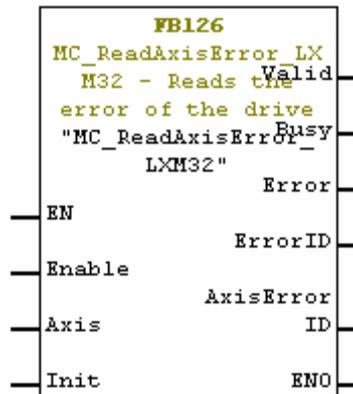
For the purpose of error handling, every block as an output *Error*, which is set if a [synchronous](#) or [asynchronous](#) error occurs. For a more detailed analysis of the error's cause, the block MC_ReadAxisError_LXM32 is called. By means of MC_Reset_LXM32, the error cell is cleared to make it available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

9.16.1 MC_ReadAxisError_LXM32

Task:

Reading the last [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. |
| 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. |
| | ErrorID | WORD | 0: No error message in the error cell >0: Error number (see list of error numbers). Value range: 0..65535, initial value: 0. |
| | AxisErrorID | WORD | Bit0...Bit7: indicates the ID of the function block which caused the error (signed value). Bit8...Bit15: indicates the input number (first input = 1, second = |

Step 7 Motion Library for Lexium 32

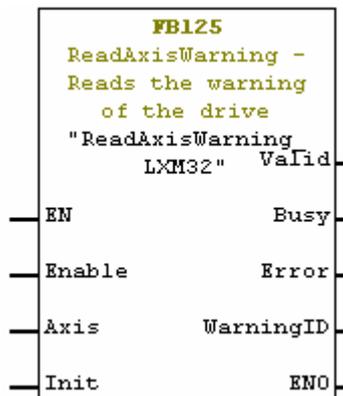
| | | | |
|--|--|--|--|
| | | | 2, ...) which caused the error. Control inputs are not considered (<i>Axis, Input, Execute, Enable, ...</i>). |
|--|--|--|--|

9.16.2 MC_ReadAxisWarning_LXM32

Task:

Returns the last active warning of the drive.

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 value to be read is available. |
| | Busy | BOOL | TRUE: Block is being executed. |
| | Error | BOOL | TRUE: An error has occurred during execution. |
| | WarningID | WORD | Last drive active warning ID. |

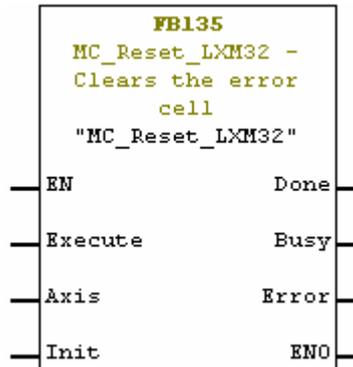
9.16.3 MC_Reset_LXM32

Task:

Error acknowledgement.

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

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. |
| 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 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

Step 7 Motion Library for Lexium 32

not overwritten (also not if the cause of the error has already been rectified), so that no new errors are entered.

10 Glossary

Asynchronous errors

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

[Errorhandling](#)

Movement profile generator

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

Error class

The device response depends on the severity of an error:

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

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

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

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

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

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

Error cell

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

Device parameter list or Object list

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

MC_WriteParameter_LXM032 MC_ReadParameter_LXM32

Inc, Inc/s

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

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

Resolution of the power amplifier: 131072 increments per revolution

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

Scaling

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

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

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

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

Also see the device manual.

Synchronous errors

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

[Errorhandling](#)

usr

stands for “user-defined unit”.

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

11 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#1613 | 5651 | 1 | Autotuning: Max. permissible positioning range exceeded |
| 16#1614 | 5652 | 0 | Autotuning: already active |
| 16#1617 | 5655 | 1 | Autotuning: Friction or load torque too high |
| 16#1618 | 5656 | 1 | Autotuning: Optimisation aborted |
| 16#1A01 | 6657 | 3 | Motor has been changed |
| 16#1B02 | 6914 | 3 | User parameter for max. current, holding current or Quick Stop current too high |
| 16#2300 | 8960 | 3 | Power amplifier overcurrent |
| 16#2301 | 8961 | 3 | Overcurrent in ballast resistor |
| 16#3100 | 12544 | par. | Phase error in mains supply |
| 16#3200 | 12800 | 3 | DC busovervoltage |
| 16#3201 | 12801 | 3 | DC busundervoltage (switch-off threshold) |
| 16#3202 | 12802 | 2 | DC busundervoltage (Quick Stop threshold) |
| 16#3203 | 12803 | 4 | Motor encoder supply voltage |
| 16#3206 | 12806 | 0 | DC busundervoltage (warning) |
| 16#4100 | 16640 | 3 | Power amplifier overtemperature |
| 16#4101 | 16641 | 0 | Warning power amplifier overtemperature |
| 16#4102 | 16642 | 0 | Power amplifier overload (I ² t) warning |
| 16#4200 | 16896 | 3 | Device overtemperature |
| 16#4300 | 17152 | 3 | Motor overtemperature |
| 16#4301 | 17153 | 0 | Warning motor overtemperature |
| 16#4302 | 17154 | 0 | Motor overload (I ² t) warning |
| 16#4402 | 17410 | 0 | Warning of overload (I ² t) in ballast resistor |

Step 7 Motion Library for Lexium 32

| | | | |
|---------|-------|------|---|
| 16#5200 | 20992 | 3 | No connection to the motor encoder |
| 16#5201 | 20993 | 4 | Errors in motor encoder communication |
| 16#5202 | 20994 | 4 | Motor encoder is not supported |
| 16#5203 | 20995 | 4 | No connection to the motor encoder |
| 16#5204 | 20996 | 3 | Connection to motor encoder lost |
| 16#5600 | 22016 | 3 | Motor connection phase fault |
| 16#5601 | 22017 | 4 | Interruption or faulty motor encoder signals |
| 16#5602 | 22018 | 4 | Interruption or faulty motor encoder signals |
| 16#5603 | 22019 | 4 | Commutation error |
| 16#6107 | 24839 | 0 | Parameters outside value range (calculation error) |
| 16#6108 | 24840 | 0 | Function not available |
| 16#610D | 24845 | 0 | Error in selection parameter |
| 16#610F | 24847 | 4 | Internal timebase fault (timer 0) |
| 16#7120 | 28960 | 4 | Invalid motor data |
| 16#7121 | 28961 | 2 | System error: Errors in motor encoder communication |
| 16#7123 | 28963 | 4 | Motor current offset outside permissible range |
| 16#7124 | 28964 | 4 | System error: Motor encoder faulty |
| 16#7329 | 29481 | 0 | Motor encoder sends: Warning |
| 16#7335 | 29487 | 0 | Communication to motor encoder occupied |
| 16#7400 | 29696 | 0 | System error: Illegal interrupt (XINT2) |
| 16#7500 | 29952 | 0 | RS 485/Modbus: Overrun error |
| 16#7501 | 29953 | 0 | RS 485/Modbus: Framing error |
| 16#7502 | 29954 | 0 | RS 485/Modbus: Parity error |
| 16#7503 | 29955 | 0 | RS 485/Modbus: receive error |
| 16#A060 | 41056 | 2 | Calculation error with electronic gearbox |
| 16#A061 | 41057 | 2 | Change in reference value with electronic gearbox too large |
| 16#A300 | 41728 | 0 | Torque ramp with HALT current active |
| 16#A301 | 41729 | 0 | Drive in status 'QuickStopActive' |
| 16#A302 | 41730 | 1 | Interruption via LIMP |
| 16#A303 | 41731 | 1 | Interruption via LIMN |
| 16#A304 | 41732 | 1 | Interruption via REF |
| 16#A306 | 41734 | 1 | Interruption by user-initiated software stop |
| 16#A307 | 41735 | 0 | Interruption by internal software stop |
| 16#A308 | 41736 | 0 | Drive in 'FAULT' status |
| 16#A309 | 41737 | 0 | Drive not in 'OperationEnable' status |
| 16#A310 | 41744 | 0 | Power amplifier not active |
| 16#A312 | 41746 | 0 | Profile generating interrupted |
| 16#A313 | 41747 | 0 | Position overrun present (pos_over=1), therefore reference point no longer defined (ref_ok=0) |
| 16#A314 | 41748 | 0 | No reference position |
| 16#A315 | 41749 | 0 | Homing active |
| 16#A316 | 41750 | 0 | Overflow on acceleration calculation |
| 16#A317 | 41751 | 0 | Drive not at standstill |
| 16#A318 | 41752 | 0 | Operating mode active (x_end = 0) |
| 16#A319 | 41753 | 1 | Manual/Autotuning: Distance range overflow |
| 16#A31A | 41754 | 0 | Manual/Autotuning: Amplitude/ offset set too high |
| 16#A31B | 41755 | 0 | HALT requested |
| 16#A31C | 41756 | 0 | Illegal position setting with software limit switch |
| 16#A31D | 41757 | 0 | Speed range overflow (CTRL_n_max) |
| 16#A31E | 41758 | 1 | Interruption via positive software limit switch |
| 16#A31F | 41759 | 1 | Interruption via negativesoftware limit switch |
| 16#A320 | 41760 | par. | Position tracking error |
| 16#A321 | 41761 | 0 | RS 422 position interface not defined as input |

Step 7 Motion Library for Lexium 32

| | | | |
|-------------------------------|-------|------|---|
| 16#A324 | 41764 | 1 | Error when homing (additional info = detailed error number) |
| 16#A325 | 41765 | 1 | Approach limit switch not enabled |
| 16#A326 | 41766 | 1 | REF switch not found between LIMP and LIMN |
| 16#A327 | 41767 | 1 | Reference movement to REF without direction reversal, impermissible limit switch LIM activated |
| 16#A328 | 41768 | 1 | Reference movement to REF without direction reversal, overrun of LIM or REF not permissible |
| 16#A329 | 41769 | 1 | More than one signal LIMP, LIMN, REF active |
| 16#A32A | 41770 | 1 | Ext. monitoring signal LIMP with counterclockwise rotation |
| 16#A32B | 41771 | 1 | Ext. monitoring signal LIMN with clockwise rotation |
| 16#A32C | 41772 | 1 | Reference movement error at REF (e.g. by impact) |
| 16#A32D | 41773 | 1 | Reference movement error at LIMP (e.g. by impact) |
| 16#A32E | 41774 | 1 | Reference movement error at LIMN (e.g. by impact) |
| 16#A32F | 41775 | 1 | Index pulse not found |
| 16#A330 | 41776 | 0 | Reproducibility of the index pulse movement uncertain, index pulse motion too close to the switch |
| 16#A331 | 41777 | 3 | No run-up operating mode with local control selected |
| 16#A332 | 41778 | 1 | Error with jog (additional info = detailed error number) |
| 16#A334 | 41780 | 2 | Timeout at Standstill window monitor |
| 16#A335 | 41781 | 1 | Processing only possible in fieldbus mode |
| 16#B100 | 45312 | 0 | RS 485/Modbus: Unknown service |
| 16#B200 | 45568 | 0 | RS 485/Modbus: Protocol error |
| 16#B201 | 45569 | 2 | RS 485/Modbus: Nodeguarderror |
| 16#B202 | 45570 | 0 | RS 485/Modbus: Nodeguardwarning |
| 16#B203 | 45571 | 0 | RS 485/Modbus: Number of monitor objects incorrect |
| 16#B204 | 45572 | 0 | RS 485/Modbus: service too long |
| 16#B300 | 45824 | 4 | Profibus: Initialising failed |
| 16#B301 | 45825 | 4 | Profibus: Initialising failed |
| 16#B302 | 45826 | 0 | Profibus: Write access denied (incorrect job identification) |
| 16#B303 | 45827 | par. | Profibus: Faulty processing of process data channel |
| 16#B304 | 45828 | par. | Profibus: Faulty processing of process data channel |
| 16#B305 | 45829 | par. | Profibus: Parameter cannot be mapped to the output data frame |
| 16#B306 | 45830 | par. | Profibus: Faulty processing of process data channel |
| 16#B307 | 45831 | par. | Profibus: Faulty processing of process data channel |
| 16#B308 | 45832 | par. | Profibus: Parameter cannot be read |
| 16#B309 | 45833 | 0 | Profibus: Sub-index not equal to zero |
| 16#B30A | 45834 | 0 | Profibus: Parameter does not exist |
| 16#B30B | 45835 | 1 | Profibus: Watchdog |
| 16#B30C | 45836 | 1 | Profibus: Motor stop via clear command of master |
| 16#B30D | 45837 | 0 | Profibus: Parameter cannot be mapped |
| Library error messages | | | |
| 16#FF00 | 65280 | | Wrong_AxisRef |
| 16#FF01 | 65281 | | Initialization_Failed |
| 16#FF02 | 65282 | | Wrong_Data_Length |
| 16#FF03 | 65283 | | Timeout |
| 16#FF04 | 65284 | | Axis_Busy |
| 16#FF05 | 65285 | | Invalid_Parameter_Number |
| 16#FF06 | 65286 | | Unknown_State |
| 16#FF07 | 65287 | | Capture_Busy |
| 16#FF08 | 65288 | | Trigger_Event_Lost |
| 16#FF09 | 65289 | | Axis_Not_In_Standstill |
| 16#FF0A | 65290 | | Unknown_Device_Type |

Step 7 Motion Library for Lexium 32

| | | | |
|---------|-------|--|-------------------|
| 16#FF0B | 65291 | | Wrong_Data_Struct |
|---------|-------|--|-------------------|