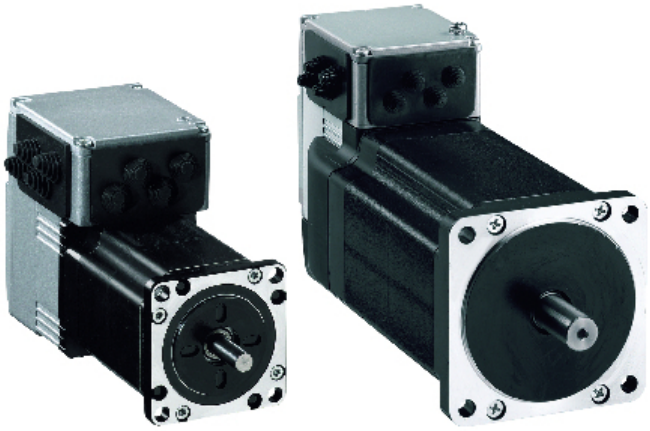
ILx2 Powerlink

Lexium Integrated Drive

Manual for PLCopen Motion Library for B&R controllers

V1.03, 12.2008



BERGER LAHR

www.schneider-electric.com

Schneider Electric

Important information

This manual is part of the library.

Carefully read this manual and observe all instructions.

Keep this manual for future reference.

Some products are not available in all countries.

Please consult the latest catalog for information on the availability of products.

Subject to technical modifications without notice.

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

Table of contents

Information on this edition

Version 0.0/1

Manual created

Version 0.1/0

Amendments added for Automation Studio.

Version 1.01

After review

Version 1.02

Change in chapter 2.5, reference to "ModuleOk"

Version 1.03

English version created

1 Introduction

Programs according to the IEC 61131-3 standard are created with the function blocks as per PLCopen Motion specification in the Automation Studio programming environment provided by B&R Automation. The function blocks control one or several Schneider Electric Lexium ILx2 drives via the Ethernet Powerlink fieldbus. The function blocks are universal so that the library is suitable for the following drives:

– ILS2E…

– ILE2E…

– ILA2E…

1.1 Scope of supply

Check the delivery for completeness to enable easy and complete integration into the B&R Automation Studio programming system.

– The library EPL_ILX as a folder containing several subfolders and files.

– The library CIA405_EPL as a folder containing several subfolders and files.

– The device description files. Two versions of Automation Studio (AS) are supported. The following files are provided for version 2.6 of Automation Studio:
  • AS2.6_HW_FBE.ILx2P.PLCOPEN.exe
  • AS2.6_HW_FBE.ILx2P.PLCOPEN.xml

– The following files are provided for version 3.0 of Automation Studio:
  • AS3.0_HW_FBE.ILx2P.PLCOPEN.exe
  • AS3.0_HW_FBE.ILx2P.PLCOPEN.xml

– This manual

– A sample program

2 Integration of the library

The library was created for the B&R Automation Studio software. The following steps are required to make the Lexium ILx2 drives known to the Automation Studio software and to link the process data of the drives and the data structures of the library. Perform the steps in the sequence listed.

2.1 Preparation

Perform the steps below before starting B&R Automation Studio.

Storage locations for the device description files

The device description files should be stored on the local hard disk. They can be stored in any folder; remember the name of the folder as you will need the files in chapter 2.3.

Select the following for Automation Studio version 2.6:

– AS2.6_HW_FBE.ILx2P.PLCOPEN.exe

– AS2.6_HW_FBE.ILx2P.PLCOPEN.xml

Select the following for Automation Studio version 3.0.

– AS3.0_HW_FBE.ILx2P.PLCOPEN.exe

– AS3.0_HW_FBE.ILx2P.PLCOPEN.xml

Storage locations for the libraries

If you store the libraries in the default folder of B&R Automation Studio, it is easier to integrate them into the project.

Select the two library folders and copy them along with all subfolders and files to the following folder of your Automation Studio:

…AS26000de\As\Library*Vxxx*

In the example, *…AS26000de* stands for the Automation Studio folder and *Vxxx* for the folder of the latest operating system in Automation Studio. The screenshot below provides an example.

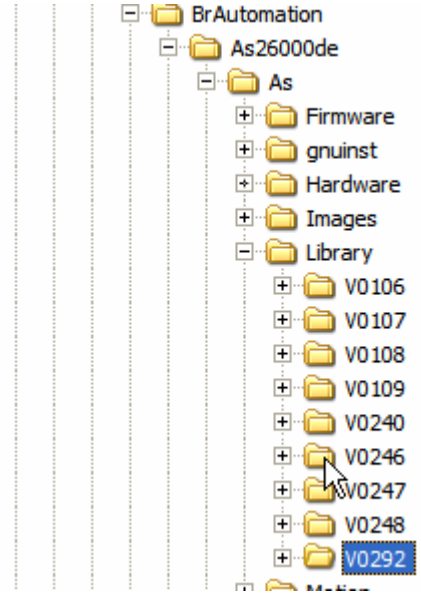


Bild 1) Folder for the libraries

2.2 Creating the application

The application contains the application program as well as the libraries used to control the Lexium ILx2P drives. B&R Automation Studio manages the application as a project. Follow the steps below to integrate the Lexium ILx2P drives into a project.

Integration of the B&R standard libraries into the project

When you create a new project in B&R Automation Studio, you must make basic settings and configurations. These initial steps are not described in this manual. The sections below describe the steps required to integrate the Lexium IL2xP drives into a project.

➢ Add the following libraries to the project:

– AsEPL
– AsString
– CONVERT
– Ncglobal
– OPERATOR
– Powerlnk
– Runtime
– Standard
– SYS_LIB
– TypeConv

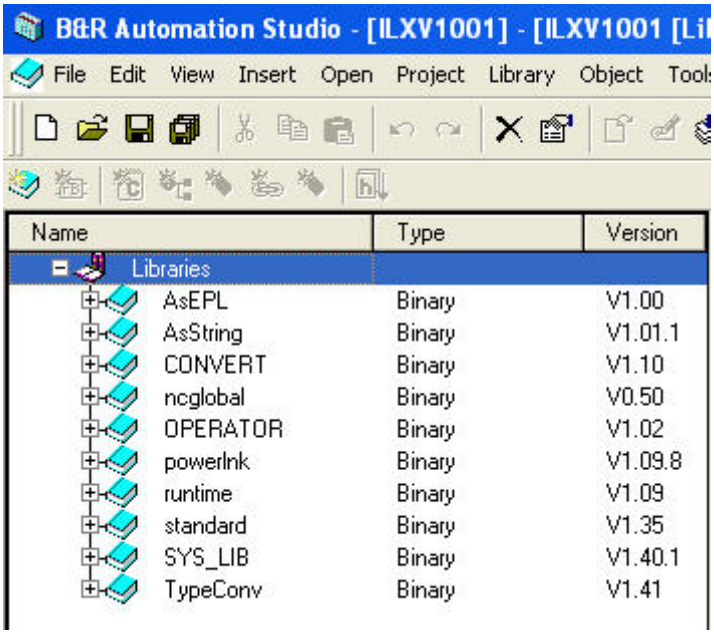


Bild 2) Automation Studio standard libraries

- To do so, open the Library Manager by choosing "Library Manager" from the "Open" menu.

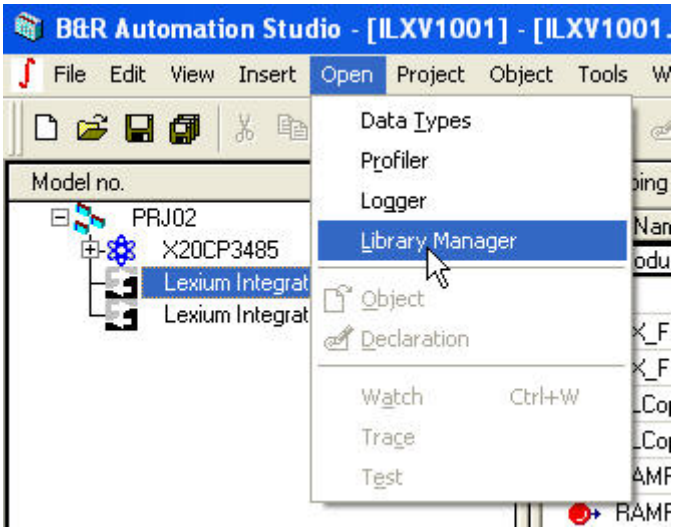


Bild 3) Opening the library manager

- Right-click the entry "Libraries" in the left pane.
- Select "Insert Library".

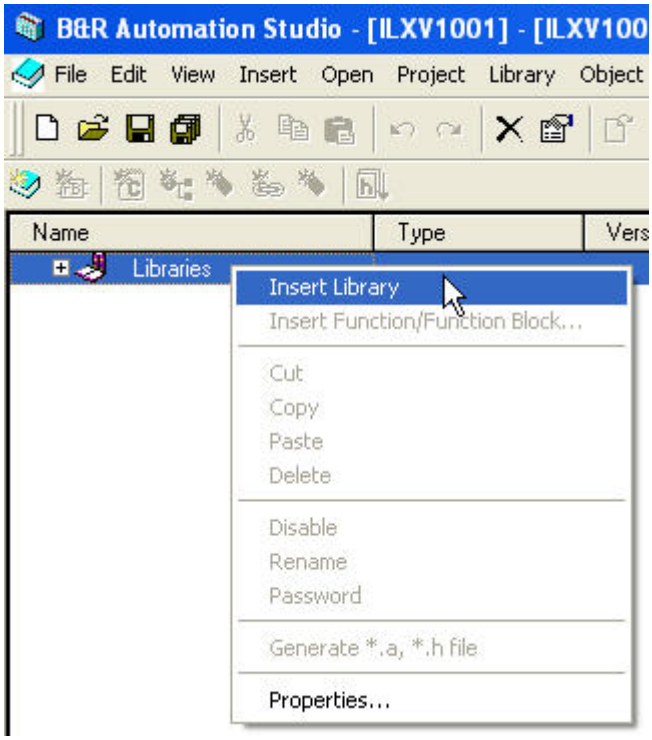


Bild 4) Inserting a library

➢ Activate the option "Add library" in the window displayed.

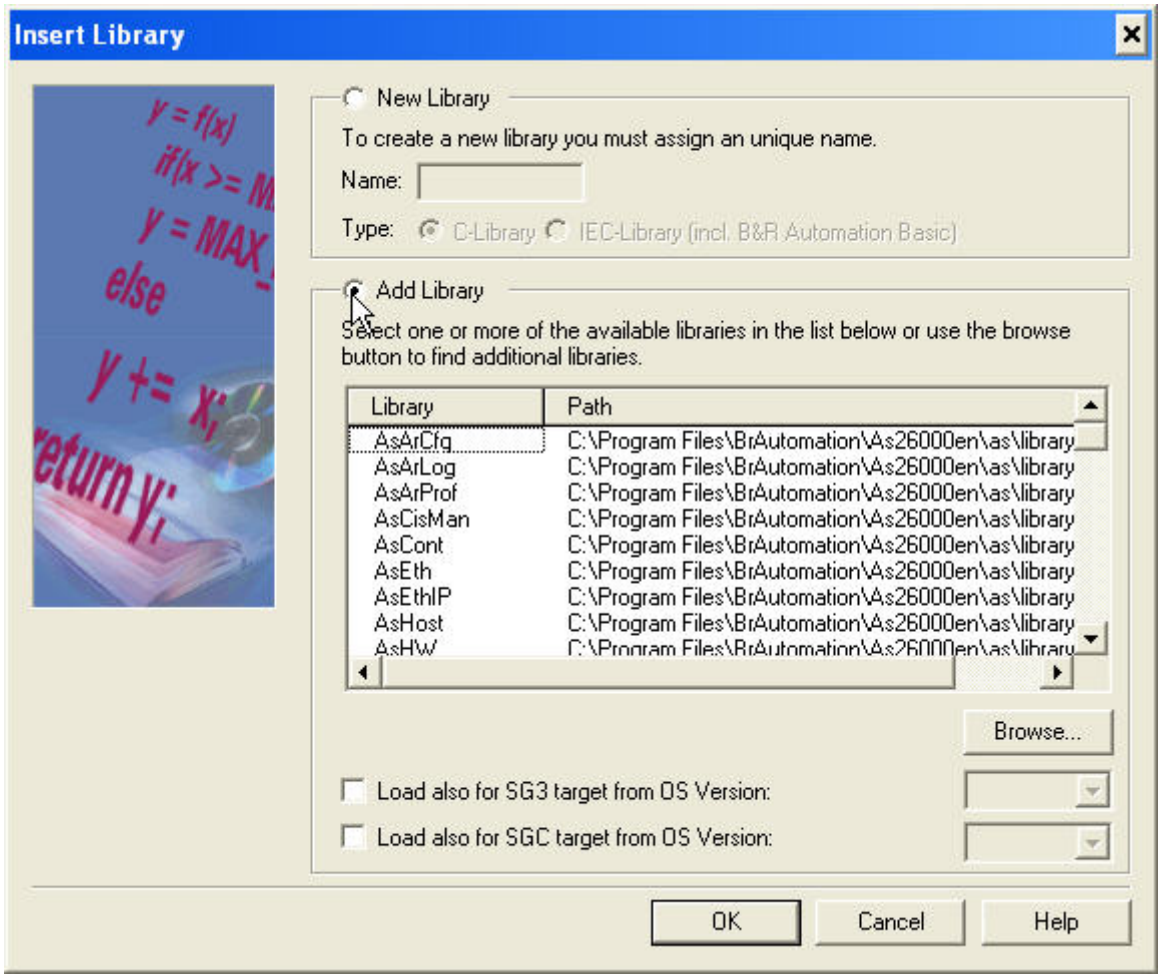


Bild 5) Adding a library

- Select the libraries mentioned above and click OK.
- Repeat the previous steps until you have added all libraries to the project.

Integration of the Schneider Electric libraries into the project

- Open the window for inserting libraries as described in the previous section.
- Select both libraries EPL_405 and EPL_ILX by holding down the SHIFT key and clicking the libraries.

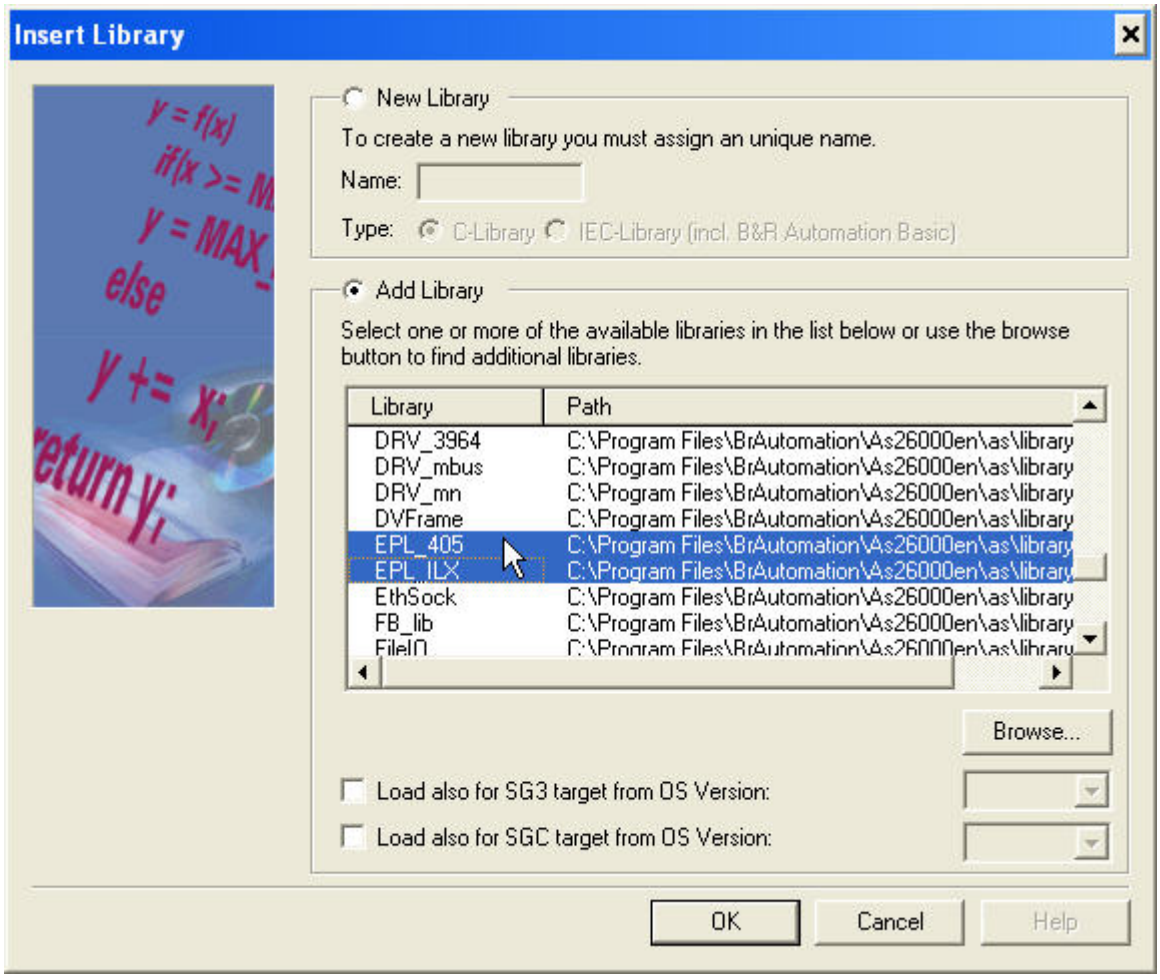


Bild 6) Selecting the Schneider Electric EPL libraries

➢ Click the OK button.

The libraries are now displayed in the left pane of the Library Manager.

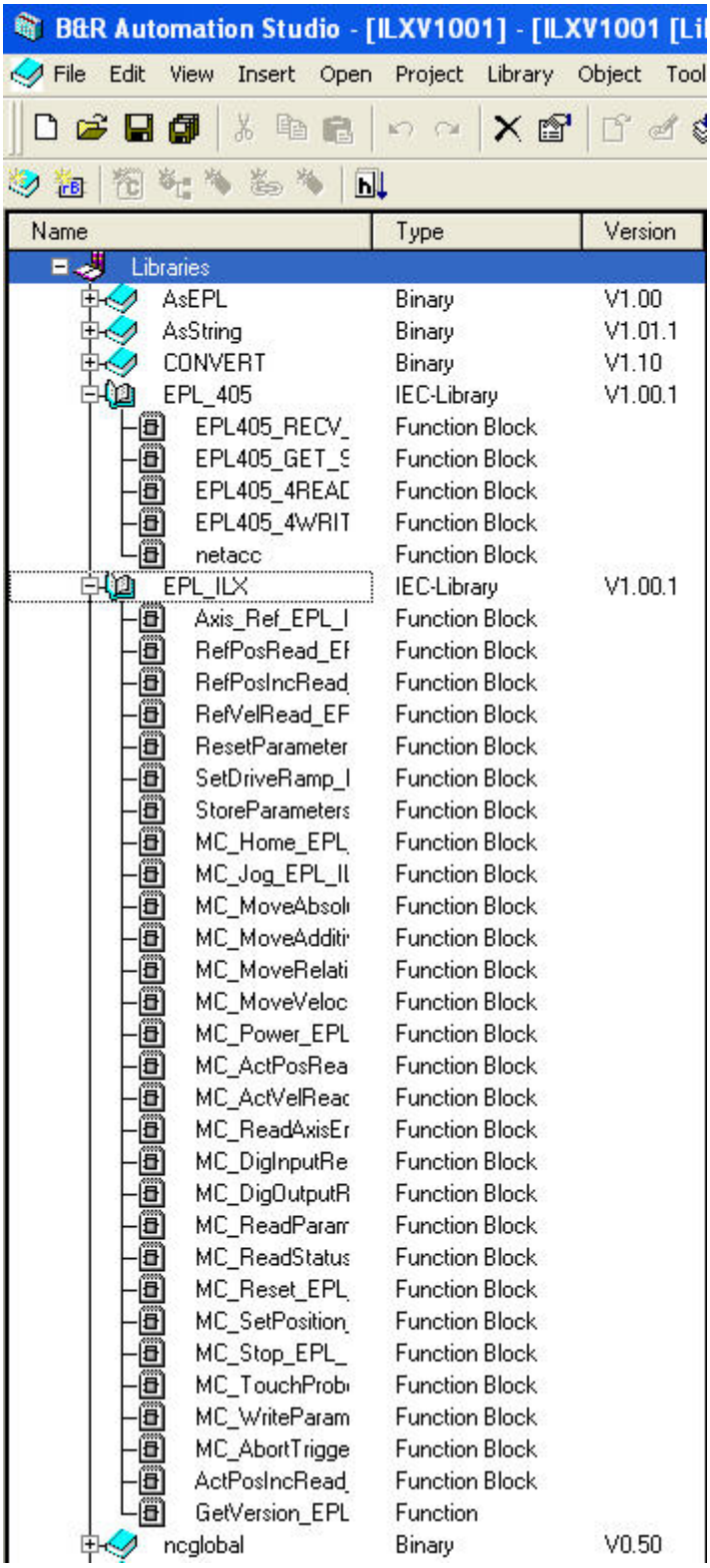


Bild 7) Inserted libraries for the Lexium ILx2P drives

- Close the Library Manager.

Creating the global data structure per drive

You must create a separate global data structure for the management of the drive in the library for each ILx2P drive that you want to operate via the Powerlink network.

- Select the CPU in the Hardware Configurator.
- Display the "Software" tab in the right pane.

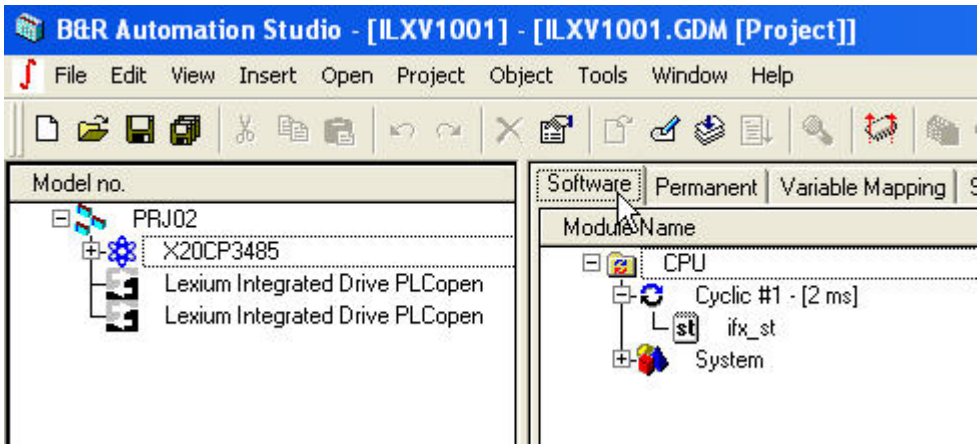


Bild 8) Displaying the "Software" tab

- Right-click the the entry "CPU" in the right pane and select "Declaration" from the menu.

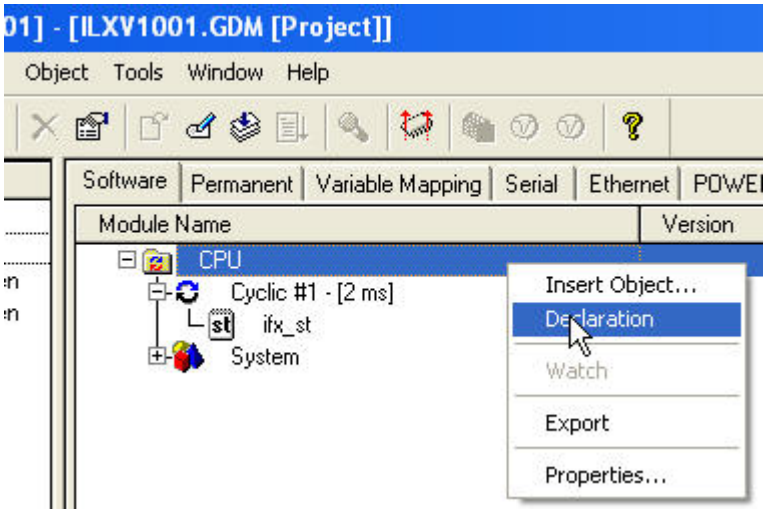


Bild 9) Declaring a global variable

- Right-click the top entry in the new window and select "New Variable" from the menu.

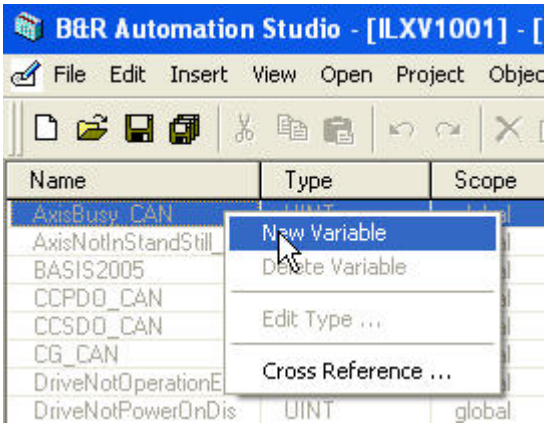


Bild 10) Adding a new variable

- Assign a name to the variable.
- Right-click the field "Type" and select "Edit Type ..." from the menu.

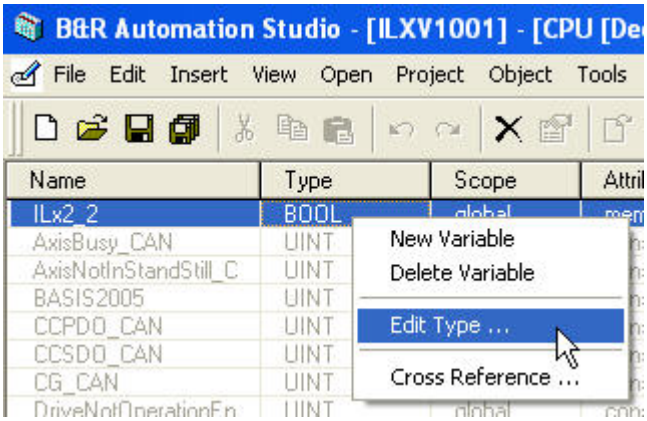


Bild 11) Editing the type

- In the new window, select "Function blocks" from the "Category" listbox.

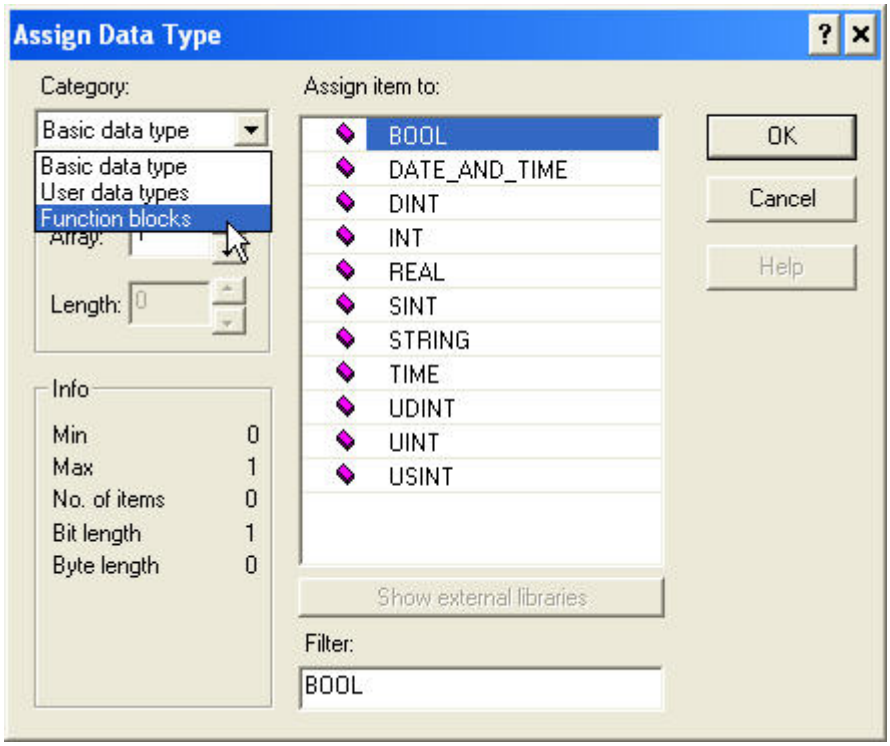


Bild 12) Selecting the category "Function blocks"

- In the "Assign item to:" list, select the entry "EPL_ILX" and then "Axis_Ref_EPL_ILX".
- Click the OK button.

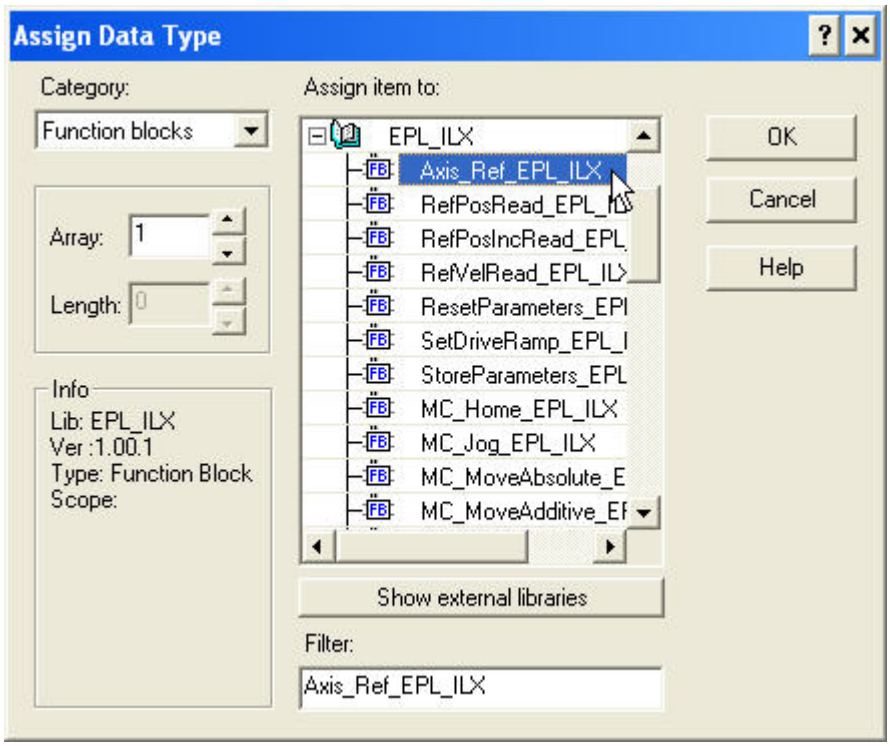


Bild 13) Selecting Axis_Ref_EPL_ILX

- Repeat the previous steps for each Lexium ILx2P drive you want to operate with the library.
- Then close the declaration window. You have now created the global variables as data structures for each drive.

2.3 Integration of the drives in Automation Studio

Integration of the device description

B&R Automation Studio supports a special file format for the description of devices. The files have the extension .hwc. You have already stored these files on your PC as described in chapter 2.1. To integrate these files in Automation Studio, perform the following steps.

- Start Automation Studio.
- From the "Tools" menu, choose "Upgrades".

Bild 14) Menu item "Upgrades"

- Click "Browse local storage..." in the new window. You can stop the running browse process by clicking this button.

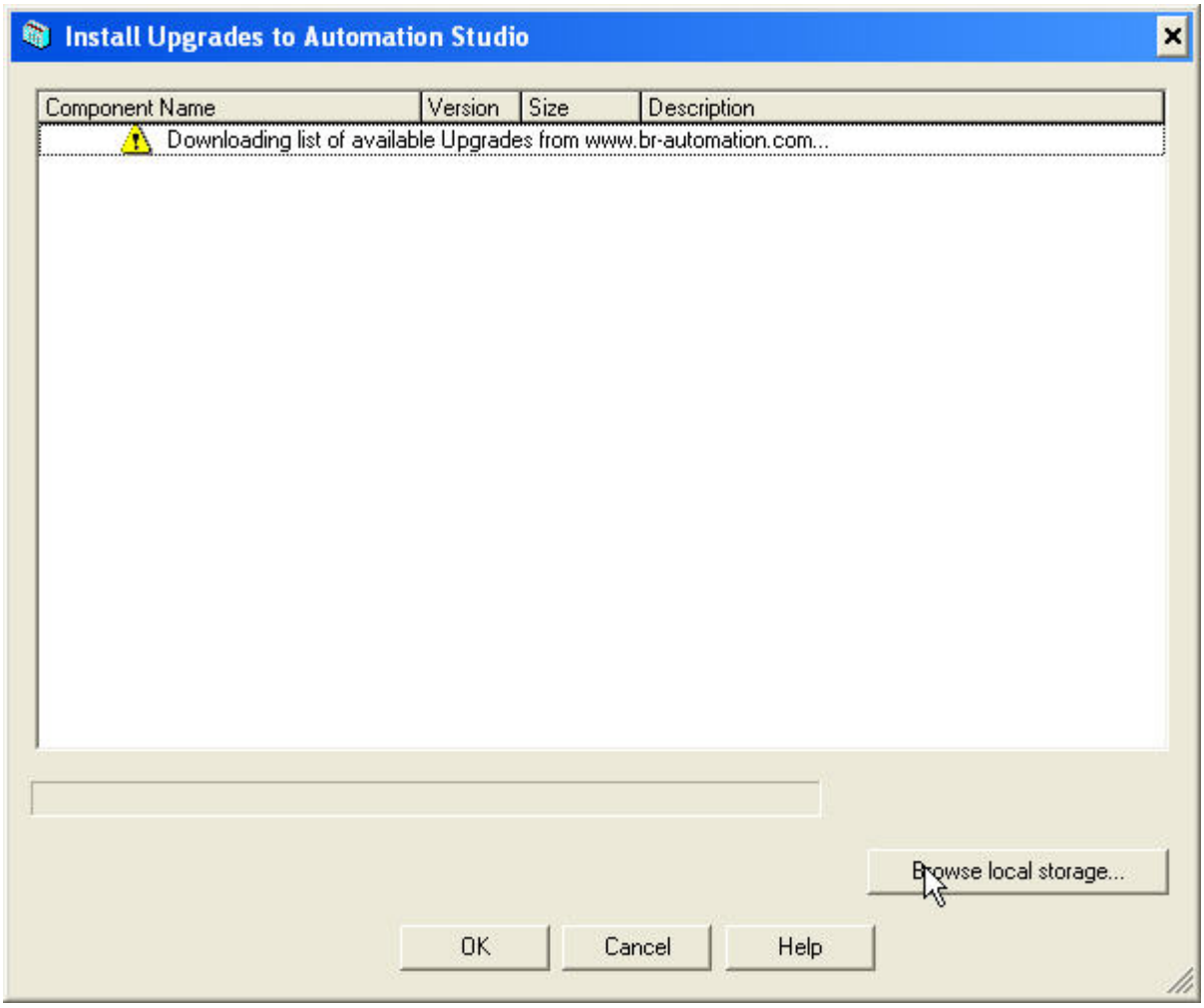


Bild 15) Browsing the local storage media

- Select the folder in which you have stored the files with the extensions .exe and .xml.
- Select the file that matches your version of Automation Studios.

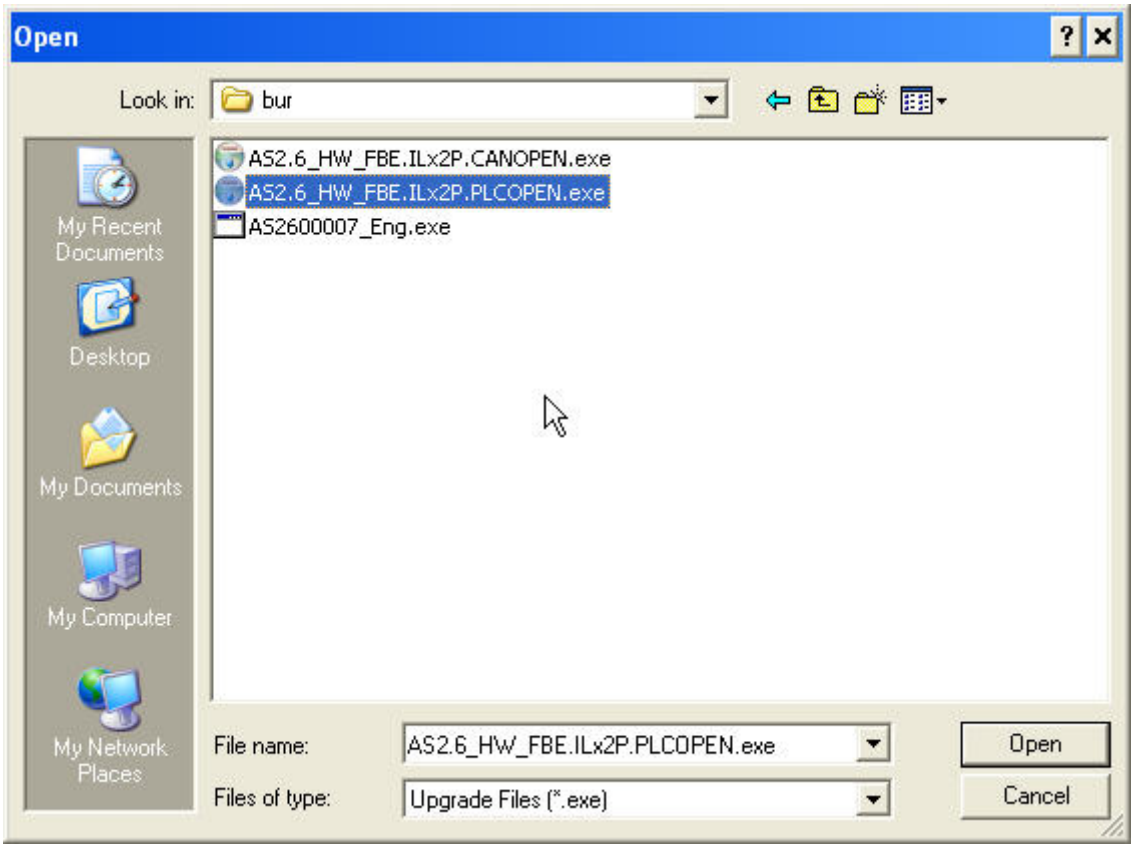


Bild 16) Selecting the file matching the Automation Studio version

- The system displays a new window prompting you for an update. Select the hardware module FBE.ILX2P.PLCOPEN.HWC.

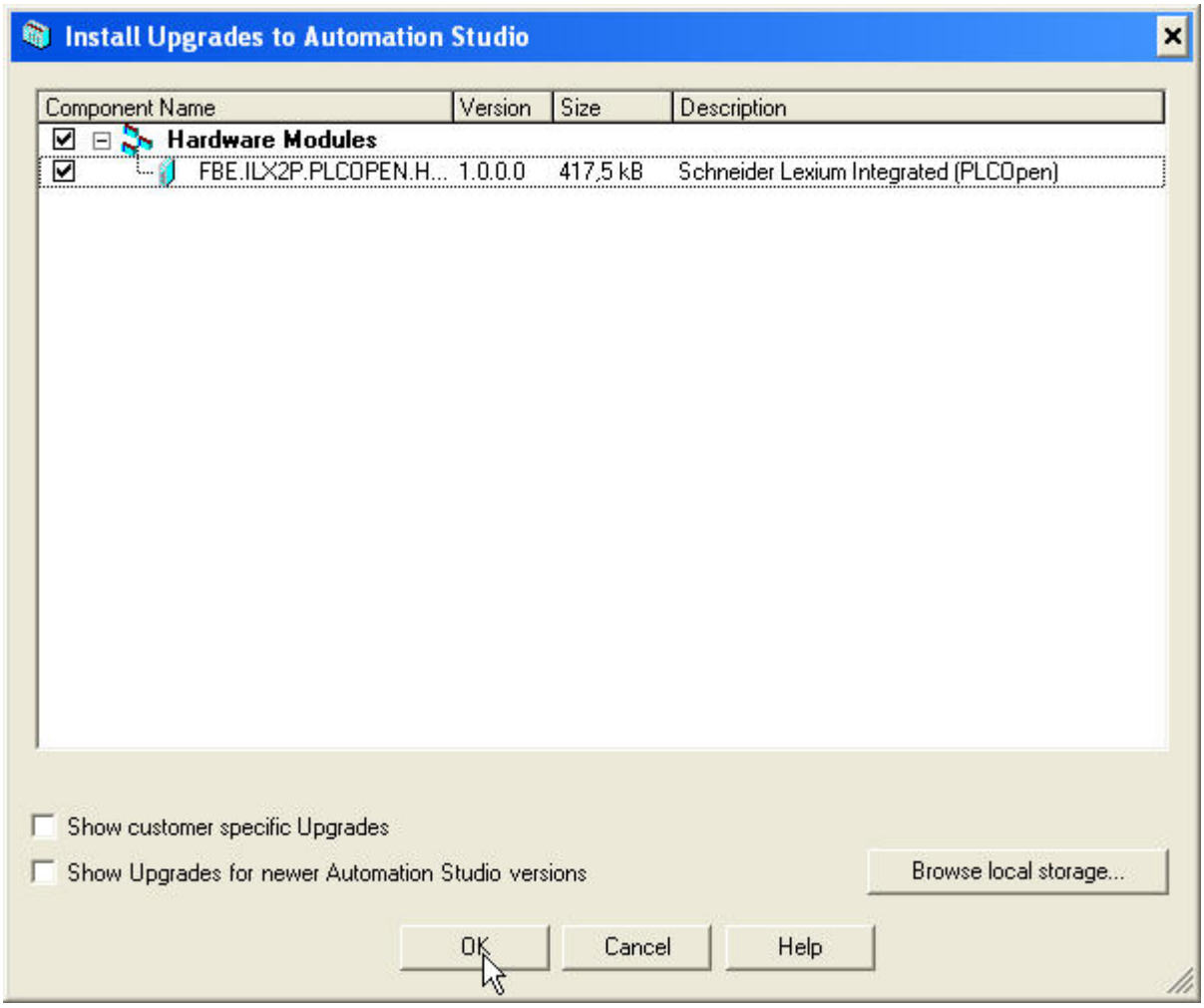


Bild 17) Selecting the upgrade

- Click OK to install the update. You can now add the Lexium ILx2P drives to the project.

Adding the drives manually

In the next steps, you will add the Lexium ILx2P drives to the project in B&R Automation Studio.

- In the left pane of the project window, select the hardware configuration of your project.
- Select the CPU.

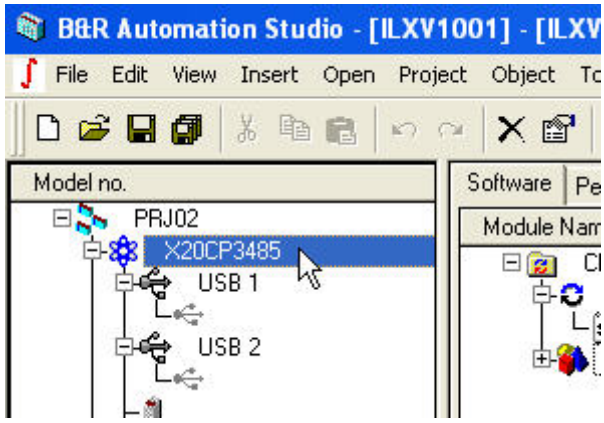


Bild 18) Selecting the CPU in the hardware configuration

- Now display the "Powerlink" tab in the right pane of the project window.

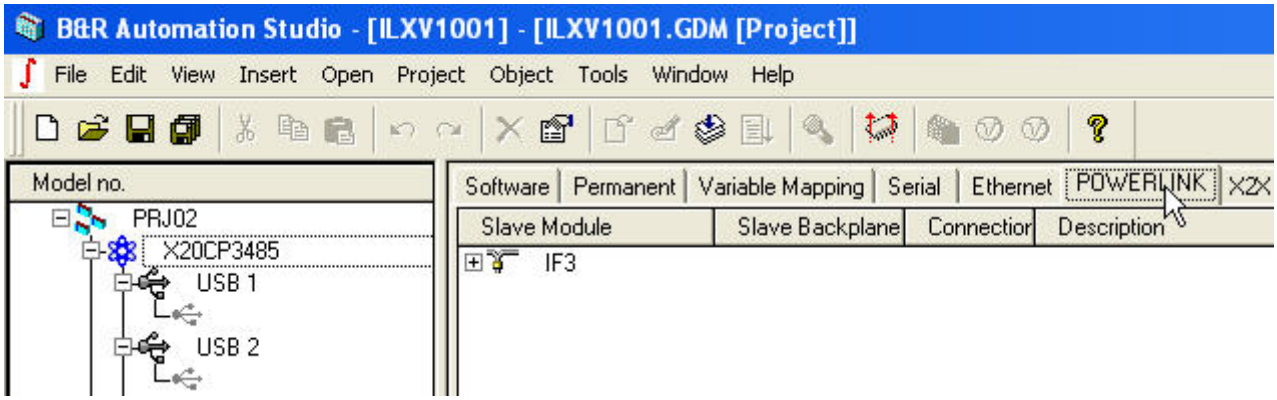


Bild 19) Displaying the "Powerlink" tab

- Right-click the entry of the slave module and select "Insert..." from the menu".

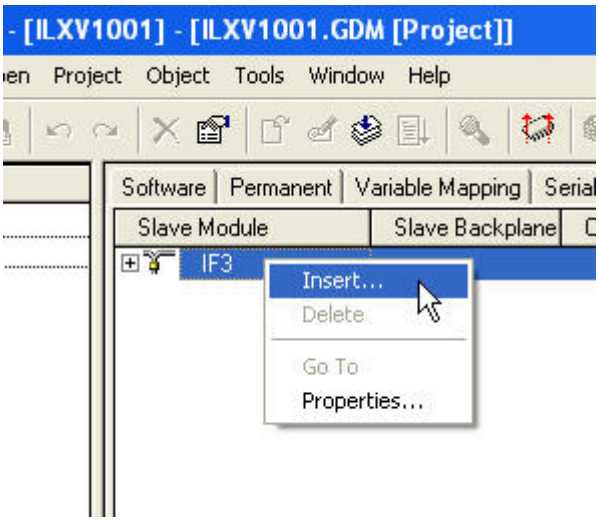


Bild 20) Inserting a Powerlink slave module

- Check the "Show customized products" checkbox.

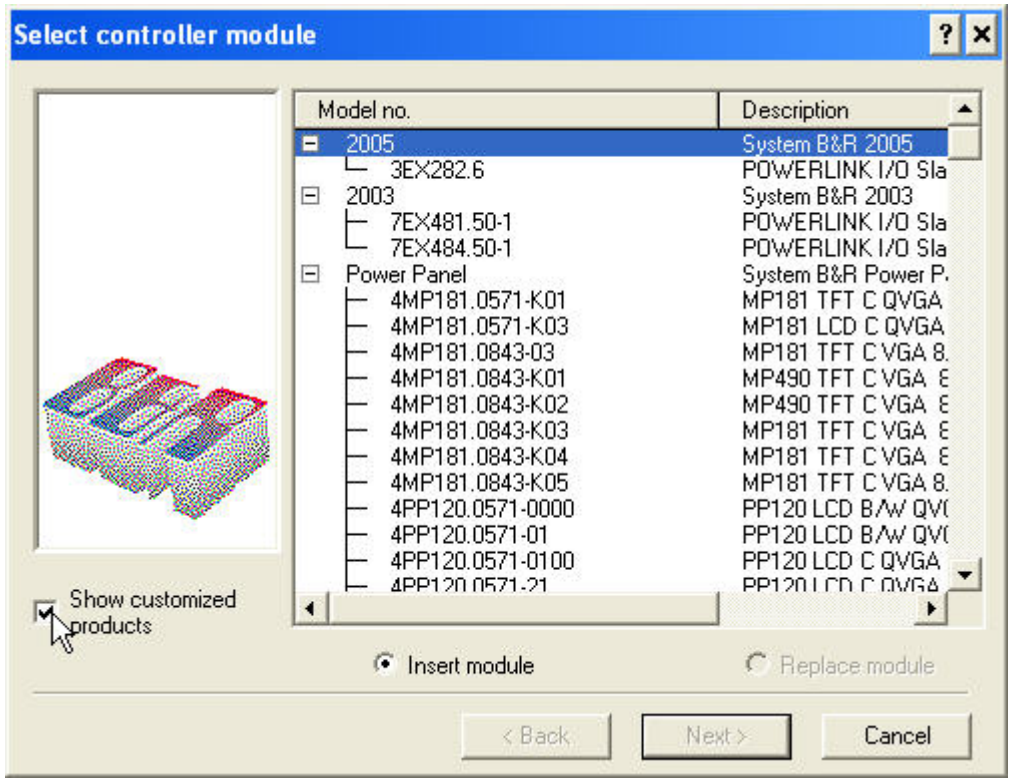


Bild 21) Activating the option "Show customized products"

- Open the "Powerlink Devices" node and select the entry "Lexium Integrated Drive PLCop". Then click "Next".

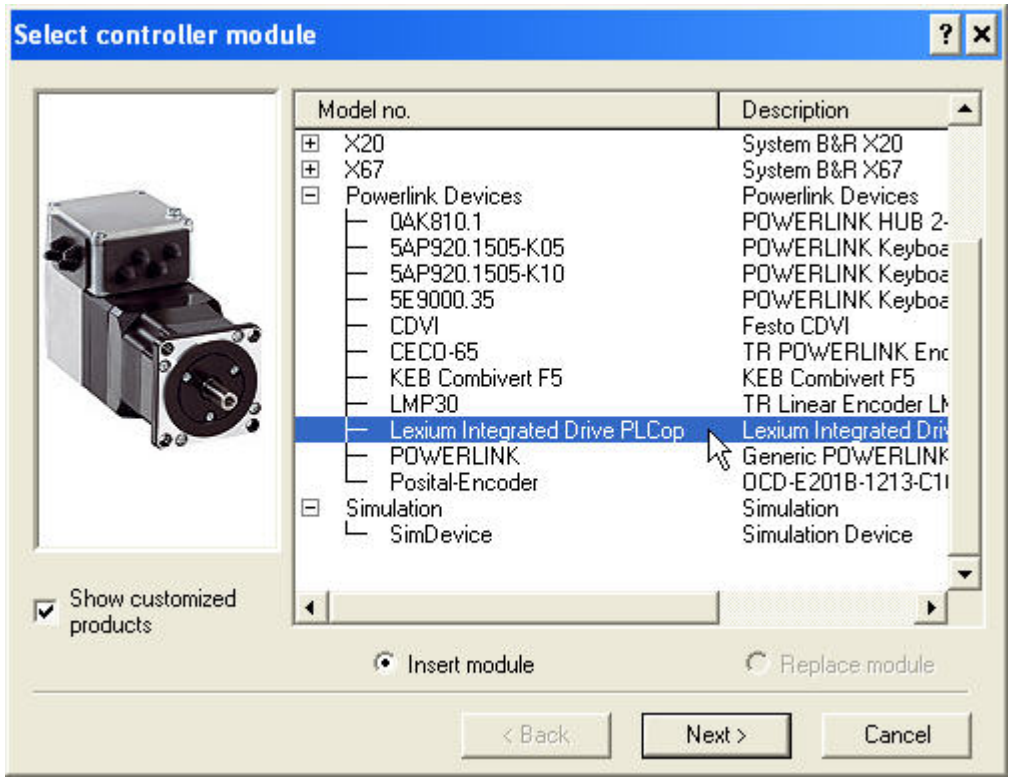


Bild 22) Selecting Lexium Integrated Drive

➢ In the new window, enter the node number as a module parameter.

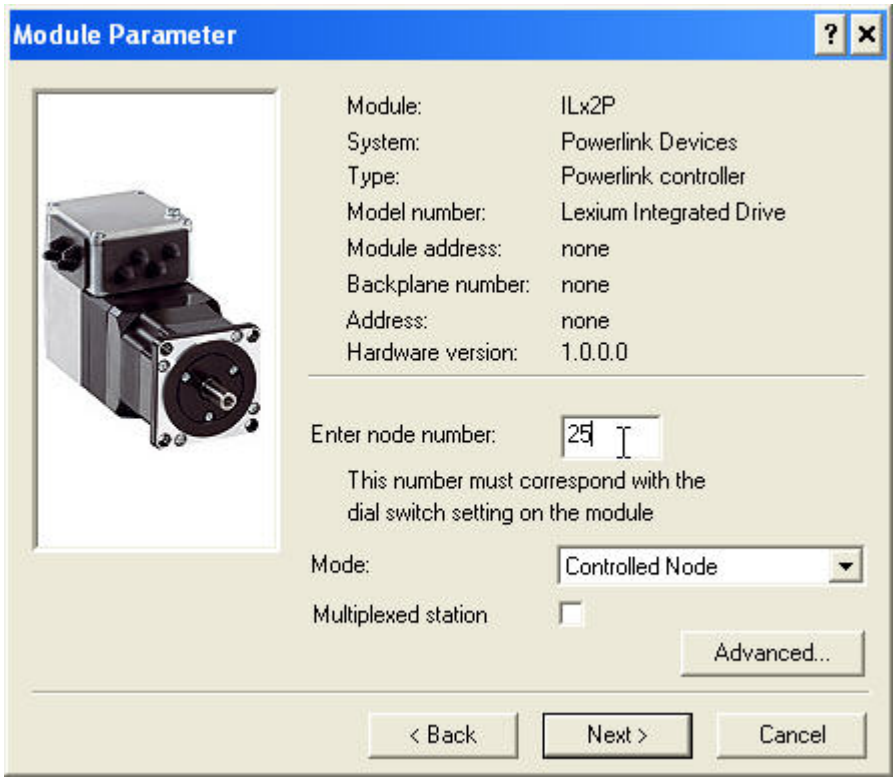


Bild 23) Setting the module parameters (node number in this case)

The Lexium ILx2P drive is now displayed below the controller in the left pane. You can repeat the previous steps in order to add additional ILx2P drives to the project.

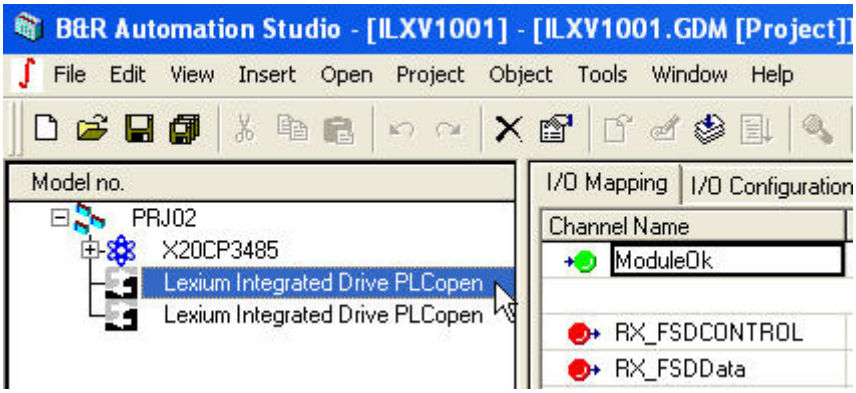


Bild 24) Lexium Integrated Drive in the Hardware Configuration

2.4 Linking the process data and the data of the application

The Lexium ILx2P drives added in chapter 2.3 must now be linked with the global data structures created in chapter 2.2. This is required so the library can access the process data of the drives.

➢ On the right pane of the hardware configuration, select the drive you want to link.

➢ Then display the "I/O Mapping" tab in the right pane.

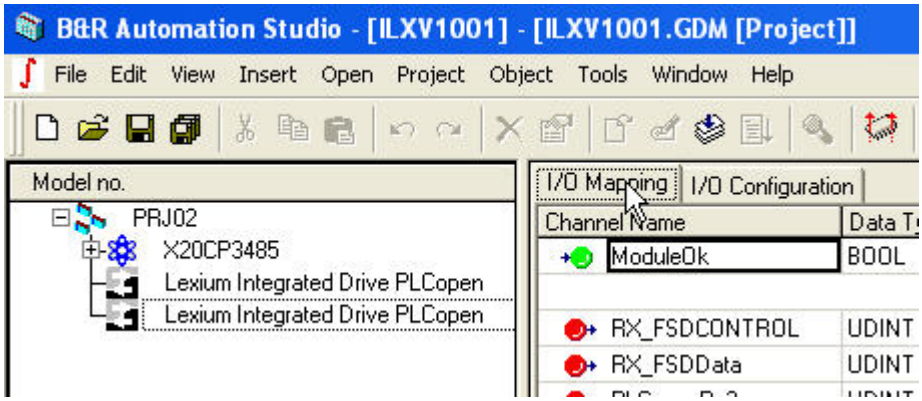


Bild 25) I/O mapping for Lexium ILx2P

Linking the status variable

➢ Double-click the name in the column "PV or Channel Name" of the "ModuleOK" channel entry.

➢ This activates the edit mode. Click the button with the three dots (...).

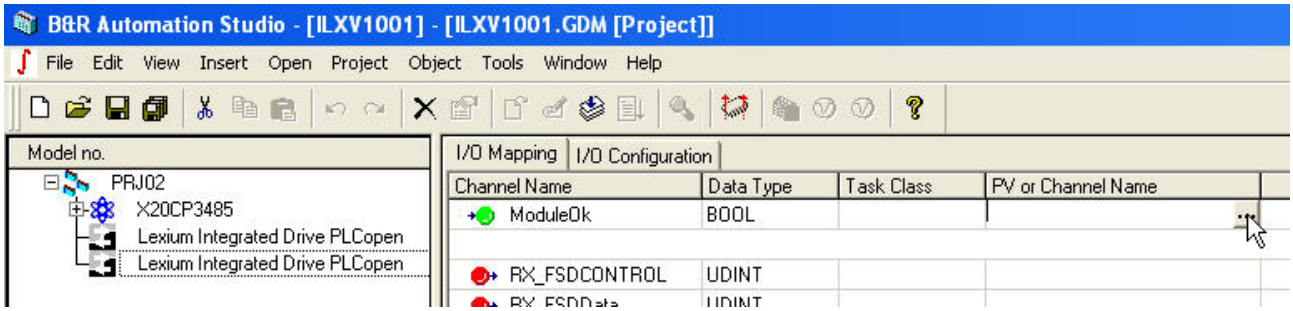


Bild 26) Editing "PV or Channel Name"

➢ In the new window, select the name of the global variable that you have created and that you want to use for this drive.

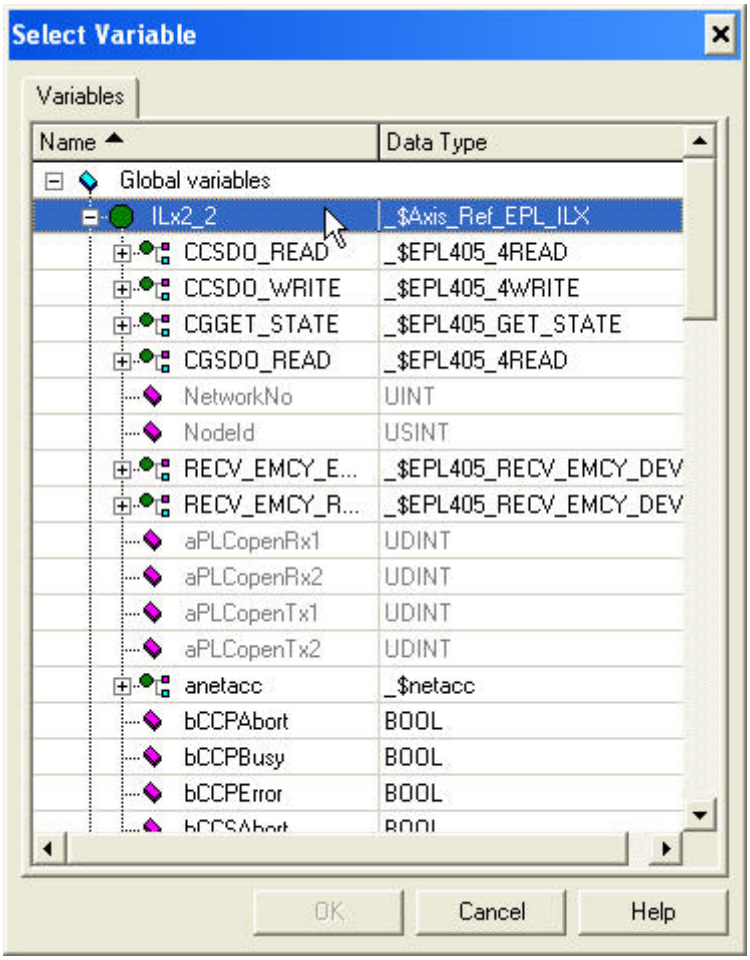


Bild 27) Selecting the variable

➢ Open the "anetacc" node and select the variable "bState".

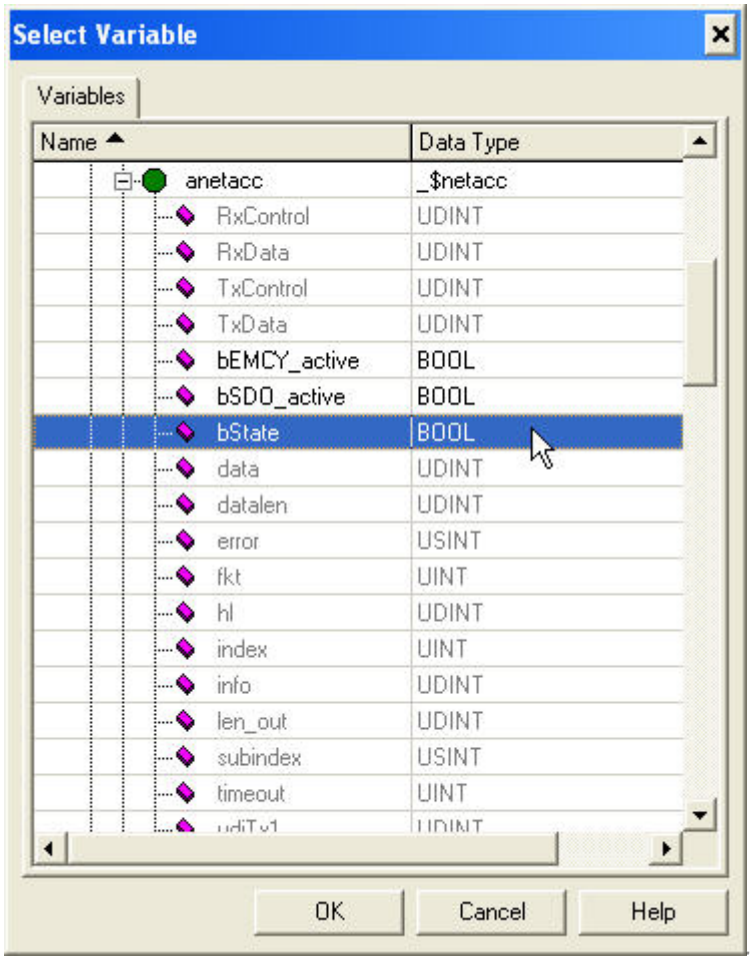


Bild 28) Selecting the variable "bState" below the "anetacc" node

➢ Click OK to establish the link.

Linking the inputs of the drive

➢ Double-click the field in the "PV or Channel Name" of the "RX_FSDCONTROL" channel entry.

➢ This activates the edit mode. Click the button with the three dots (...).

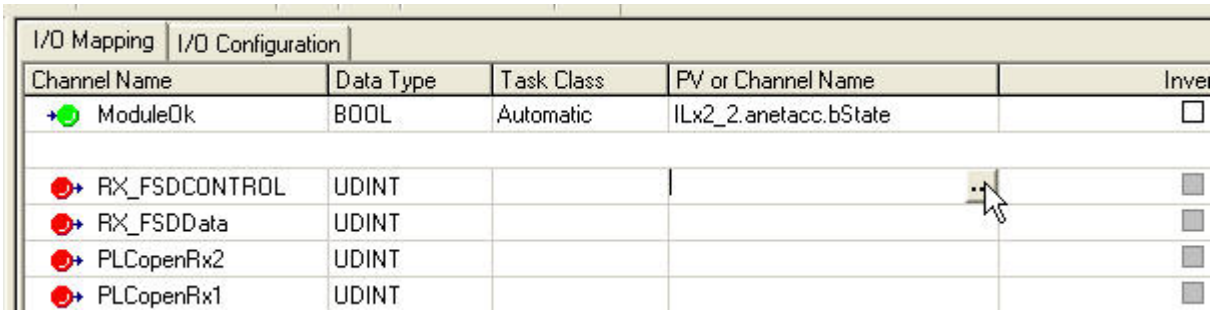


Bild 29) Linking RX_FSDCONTROL

- In the new window, select the name of the global variable that you have created and that you want to use for this drive.

- Open the "anetacc" node and select the variable "RxControl".

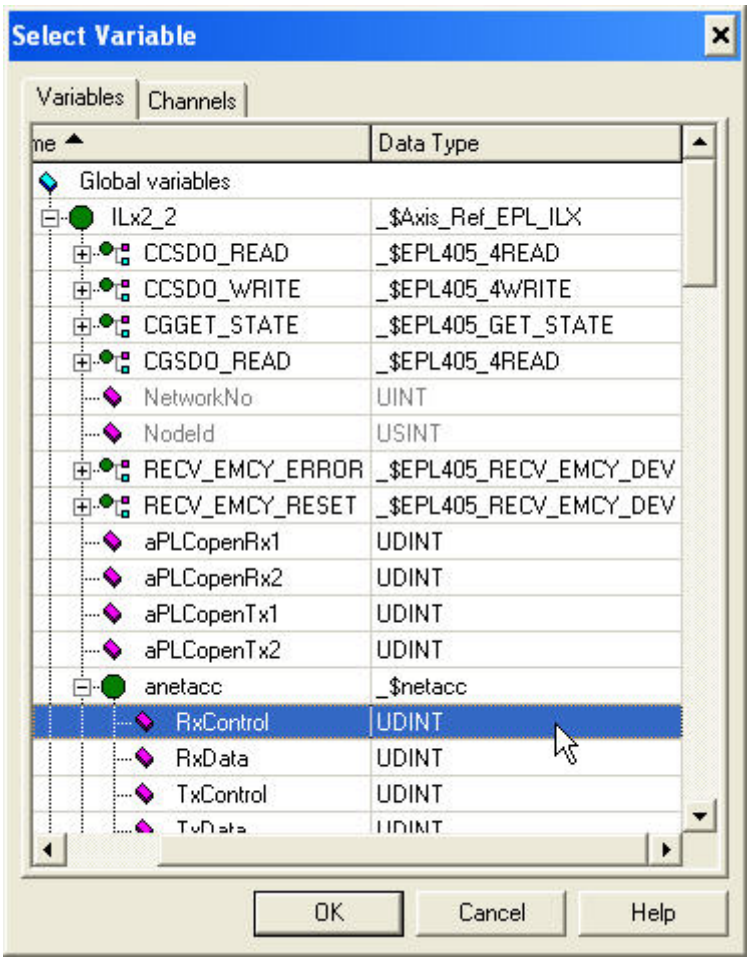


Bild 30) Selecting anetacc RxControl

- Click OK. This establishes the link.

- Repeat the previous steps until you have established the following links:

– RX_FSDCONTROL *Name*.anetacc.RxControl

– RX_FSDData *Name*.anetacc.RxData

– PLCopenRx2 *Name*.aPLCopenRx2

– PLCopenRx1 *Name*.aPLCopenRx1

Name represents ´the name of the global variable you have created for the drive.

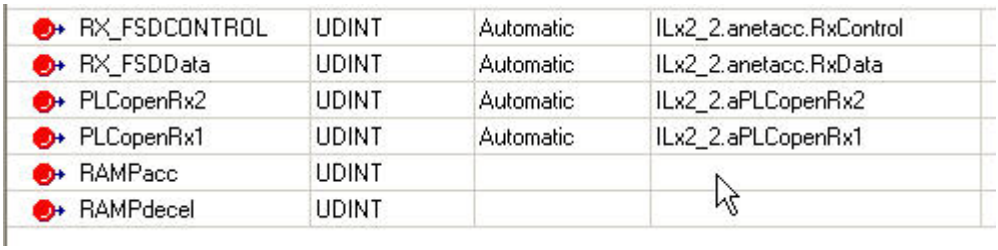


Bild 31) Linking the outputs

Linking the outputs of the drive

- Double-click the field in the "PV or Channel Name" column of the "TX_FSDCONTROL" channel entry.
- This activates the edit mode. Click the button with the three dots (...).

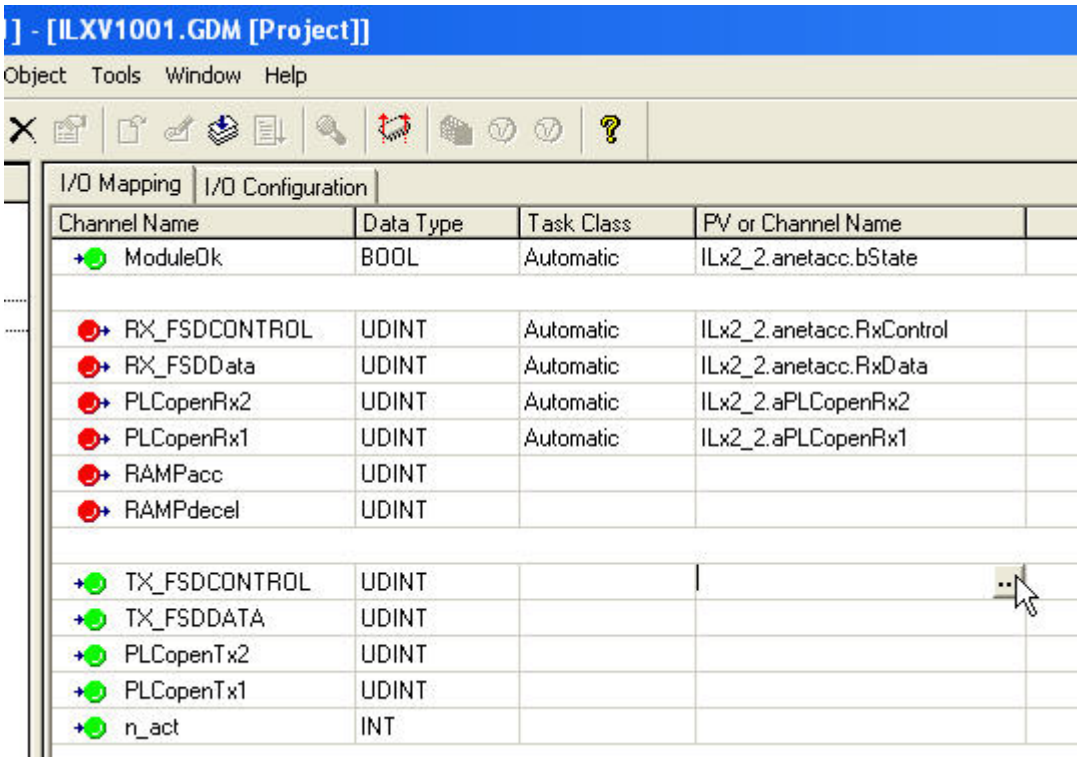


Bild 32) Linking TX_FSDCONTROL

- In the new window, select the name of the global variable that you have created and that you want to use for this drive.
- Open the "anetacc" node and select the variable "TxControl".

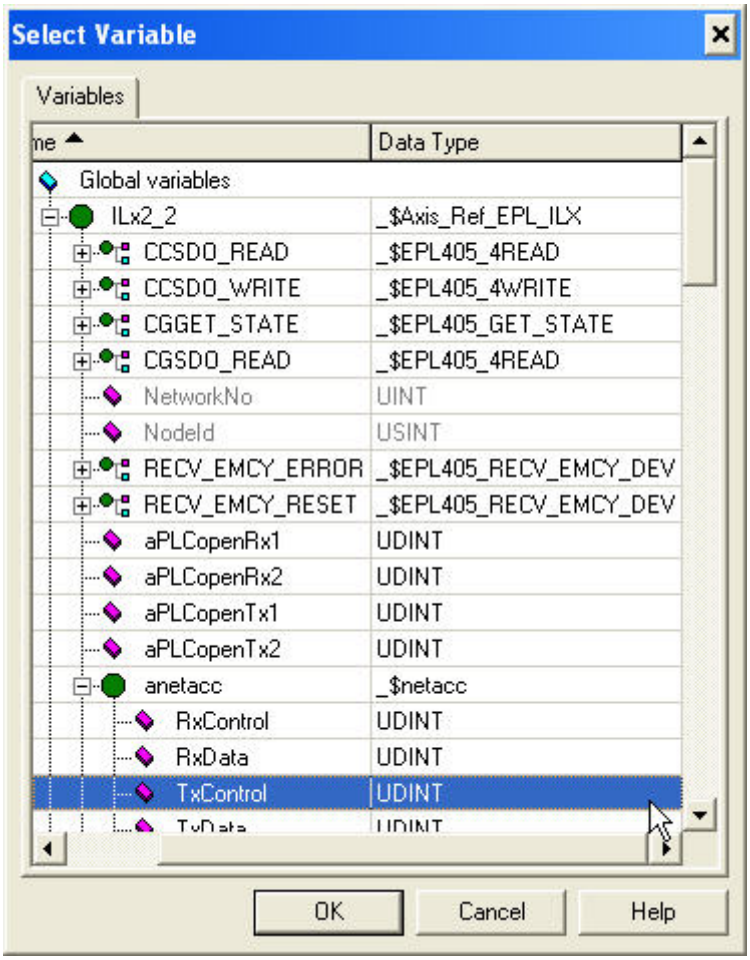


Bild 33) Selecting anetacc TxControl

- Click OK. This establishes the link.
- Repeat the previous steps until you have established the following links:

– TX_FSDCONTROL *Name*.anetacc.TxControl
– TX_FSDData *Name*.anetacc.TxData
– PLCopenTx2 *Name*.aPLCopenTx2
– PLCopenTx1 *Name*.aPLCopenTx1

Name represents ´the name of the global variable you have created for the drive.

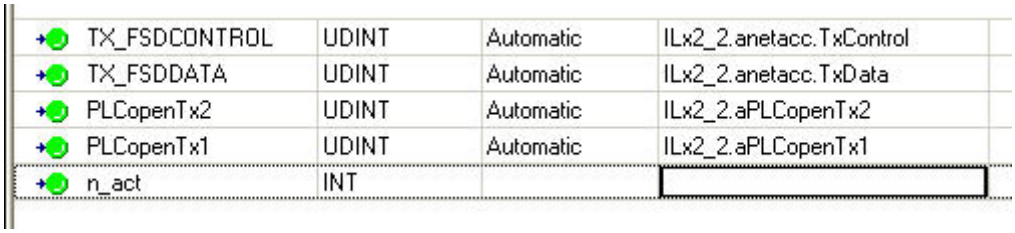


Bild 34) Linking the inputs

You have now set up all links for one drive.

➢ Repeat the steps described in this chapter for all additional Lexium ILx2P drives.

2.5 Notes on commissioning

The following notes are intended to support you in commissioning the drives in the Powerlink network at the B&R controller.

Deactivate module monitoring (supervision)

During commissioning, it may be possible that a device configured for the Powerlink network is not available. If you have activated module monitoring (supervision) for this device, the controller detects that the device is missing and switches to Service Mode. The application is not executed in this mode; the mode can only be exited by means of a soft reboot. Perform the following steps to deactivate module monitoring (supervision) for the Lexium ILx2P drive.

➢ Switch to the hardware configuration.

➢ In the left pane, select the Lexium ILx2P drive for which you want to deactivate module monitoring (supervision).

➢ Display the tab "I/O Configuration" on the right pane.

➢ The item "Module supervised" is located below the "General" node. Double-click the "Value" cell in the "Module supervised" row and select the value "off".

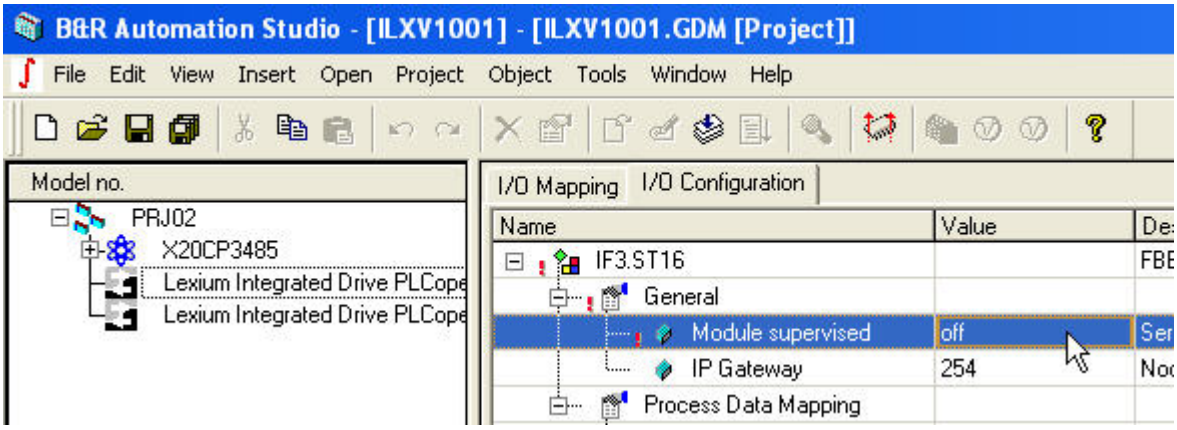


Bild 35) Deactivating module monitoring

➢ Repeat the previous steps for each Lexium ILx2P drive for which you want to deactivate module monitoring (supervision).

➢ Re-activate module monitoring (supervision) after you have commissioned the drive.

Use the Monitor view to monitor the connection

Automation Studio's Monitor view displays the process data transmitted to the Lexium ILx2P drive via the Powerlink network. You can use this view to verify that the connection between the controller and the drives operates as required. There must be a connection between Automation Studio and the controller for this view to be available.

- Select the ILx2P drive in the hardware configuration.
- Display the "I/O Mapping" tab on the right pane.
- From the "View" menu, choose the menu item "Monitor".
- The column "Physical Value" now displays the value "True" in the "ModuleOk" row if the connection to the drive is established.

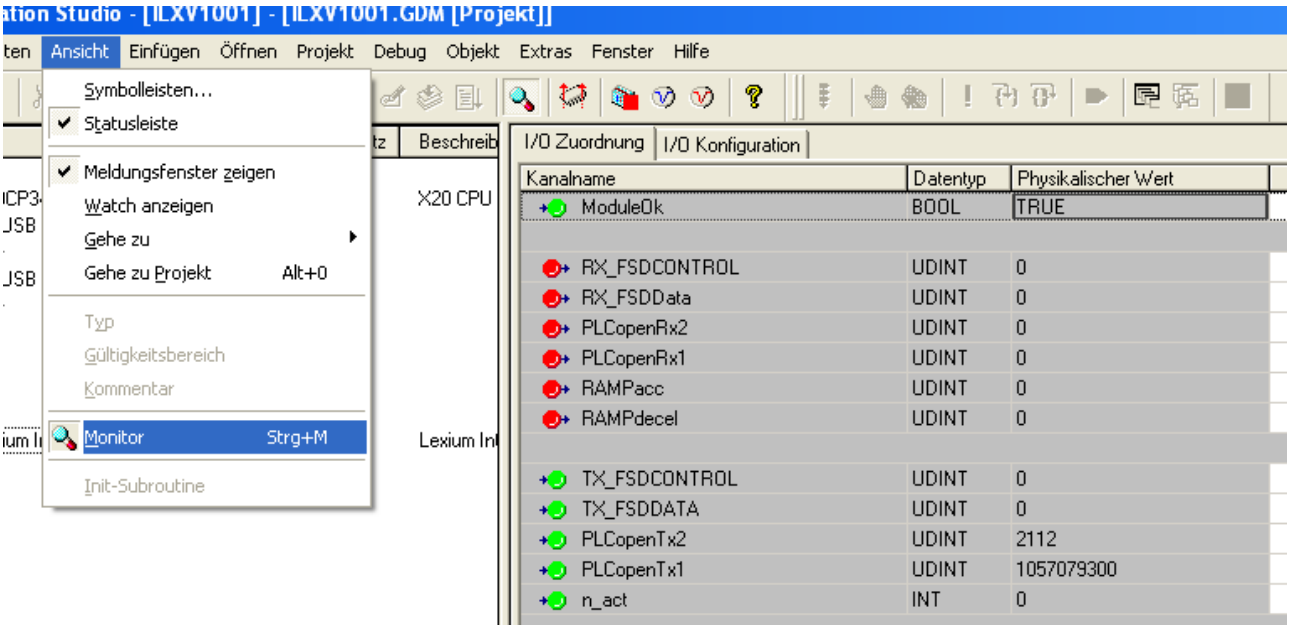


Bild 36) I/O Mapping with active Monitor view

2.6 Addressing the drives in Automation Studio

The Lexium IL2xP drives are addressed via an adjustable address. The drive has rotary switches used to set the node number. This node number must also be assigned to the drive in Automation Studio. For details, see Bild 23).

Addressing

Permissible addresses for the drives are in the range from 1 to 239.

2.7 Addressing the drives with Lexium CT

There are three ways of connecting the the Lexium ILx2P drives via the Lexium CT commissioning software or via a Web Browser.

– Direct point-to-point connection between the PC with Lexium CT and the Lexium ILx2P drive.

– Connection via the B&R controller; the Lexium ILx2P is installed in the Powerlink network. The PC with Lexium CT must be operated in the same subnet as the B&R controller.

– Connection via the B&R controller; the Lexium ILx2P is installed in the Powerlink network. The PC with Lexium CT is operated in a different network or a different subnet than the B&R controller.

Point-to-point connection via Lexium CT

After being switched on, the Lexium ILx2P drive waits for 5 seconds and then switches to Basic Ethernet Mode unless a Powerlink Managing Node or a Powerlink communication has connected to the drive within these 5 seconds. In Basic Ethernet Mode, the drive cannot only receive Powerlink messages, but also regular Ethernet TCP/IP packets.

The IP address of the drive in Basic Ethernet Mode is 192.168.100.*PowerlinkAddress*; PowerlinkAddress is the address between 1 and 239 set via the rotary switches of the drive.

➢ Verify that the IP address of your PC also begins with 192.168.100.x.

➢ You can now establish a connection with the drive via the Web browser or Lexium CT and configure the drive.

Connection via the B&R controller

The B&R can assume the function of a gateway between the Powerlink network and the company network with your PC. Verify that the B&R controller is properly connected. The controller must have separate connections for the company network (Ethernet) and the Powerlink network (EPL). Dedicated gateways are also available. The following sections assume that the controller and your PC are in the same subnet of the company network.

The Lexium ILx2P drives are addressed via the commissioning software and the IP address 192.168.100.*PowerlinkAddress*. The Powerlink address is set to a value between 1 and 239 using the rotary switches of the drive.

Before you can establish a connection, you must perform the following steps.

➢ In Automation Studio, set the controller or the gateway with the correct node address via the parameters of the Lexium ILx2P drives.

- Connect the drives to the Ethernet Powerlink (EPL) connection of the controller.
- Connect the company network to the Ethernet (ETH) connection of the controller.
- The network IP addresses of the company network and of the Powerlink network must be different. The gateway must be able to distinguish the two networks. Example: If you use the IP addresses 10.186.100.x for Powerlink, these IP addresses must not be used for the company network.

The next step requires you to have administrator rights on your PC. If you do not have administrator rights, check back with your network administrator.

Now add a manual routing entry for the Powerlink network. This entry is lost when you log off or shut down the PC. After a re-start, you must create the entry once again.

- Open a Command Prompt on your PC.
- Enter the command "route ADD 192.168.100.0 MASK 255.255.255.0 *192.168.1.2*". Replace *192.168.1.2* with the IP address of the B&R controller or the gateway.

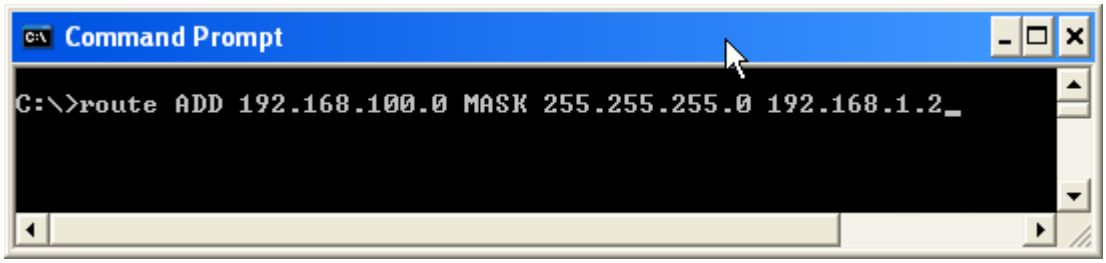


Bild 37) Adding the route to the Powerlink network

The connection is now ready to be used.

By adding the switch -p, you can make the routing entry permanent. With this switch, the route is still active the next time you switch on your PC.

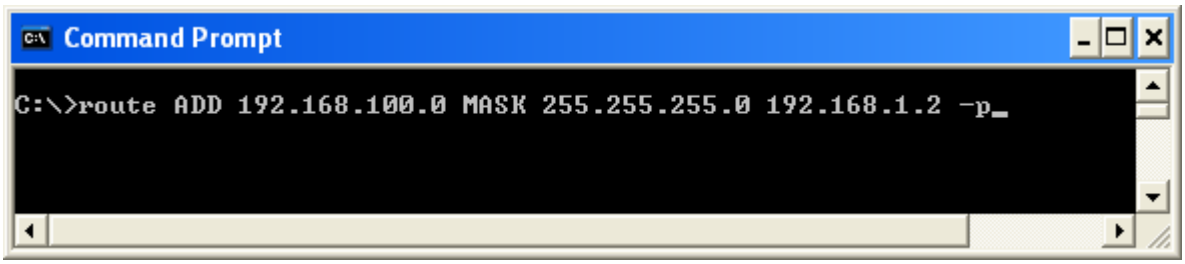


Bild 38) Creating a permanent route

Access from a different network or subnet

Check back with your network administrator for access to the Powerlink network from a PC in different subnet or in a network other than your company network.

3 Function blocks

The function blocks contained in the library EPL_ILX only control one ILx2 drive each. The library does not contain function blocks that control several drives simultaneously or one drive in dependence on another drive. However, it is possible to create several instances of a function block; each instance then controls another drive independently.

The function blocks of the CIA405_EPL library are used by the EPL_ILX library. You may not directly access these function blocks; only use the function blocks of the EPL_ILX library.

3.1 Function block names

Function blocks whose names begin with the prefix Prefix *MC_* are compliant with the specification developed by the PLCopen User Organization. They comply with a global standard for programming Motion Control applications.

Due to the Automation Studio naming conventions, the names of some function blocks with the prefix MC_ differ from the PLCopen standard.

Function blocks whose names do not contain the prefix Prefix *MC_* still comply with this standard; however, these function blocks are not yet covered by the specifications.

To identify the drives and fieldbuses for which the function blocks can be used, the postfixes _EPL for the Powerlink fieldbus and _ILX for the Lexium drives are used. This allows for unique identification.

Two typical examples of function block names:

– MC_Power_EPL_ILX
– SetDriveRamp_EPL_ILX

3.2 Signal diagrams for inputs and outputs

The signal behavior of the function blocks is uniform. There are two types of function blocks:

– Function blocks that are edge-controlled via the *Execute* input.
– Function blocks that are level-controlled via the *Enable* input.

3.2.1 Control inputs

Control input Execute

The control input Execute has two functions:

– A rising edge starts the execution of the function block. Input parameters such as position and speed are taken over and the operating mode is started.

– The control outputs are enabled or disabled: As long as Execute = TRUE, the signal outputs signal the current state of the function block. This means that as long Execute = TRUE, exactly one of the outputs Done, Busy, CommandAborted or Error is also TRUE.

On the other hand, if Execute = FALSE before the execution of the function block is completed, Busy = TRUE until the execution is completed. Upon completion of the execution, exactly one of the outputs Done, CommandAborted or Error for exactly one call becomes TRUE and then FALSE.

Control input Enable

The control input Enable starts and stops the execution of the function block (exception: MC_Power_EPL_ILX). With TRUE, the function block is repeated. With FALSE, the execution is immediately stopped and the control outputs Valid, Busy, CommandAborted and Error are immediately set to FALSE.

3.2.2 Control outputs

Control output Done

The control output Done signals the end of the execution if no errors occur. This is the case, for example, when the target position of a point-to-point movement is reached.

In a number of function blocks, the output *Done* has a different name, for example *InVelocity* in the case of the function block for Profile Velocity operating mode. *InVelocity* signals that the required velocity has been reached. The execution of the function block is not terminated with this; the movement continues to be monitored.

Control output Valid

The control output Valid signals that the outputs of the function block contain valid values. For example, in the case of the function block MC_ReadParameter_EPL_ILX, the outputs *Value* and *Length* are valid as soon as *Valid* becomes TRUE.

Control output Busy

The control output Busy signals that the function block is being executed. The execution of the function block is not completed until after *Busy* = FALSE; only after that, one of the outputs *Done*, *Error* or *CommandAborted* signals whether the function block was completed with or without an error or whether it was aborted.

Control output CommandAborted

The control output CommandAborted signals that another function block has taken over axis control. This terminates the execution of the function block that signals *CommandAborted*. For example, the function block MC_MoveVelocity can only be terminated by

cancellation.

Signal output Error

The signal output Error signals the end of the execution if an error has occurred.

3.2.3 Signal diagrams for function blocks with Execute control input

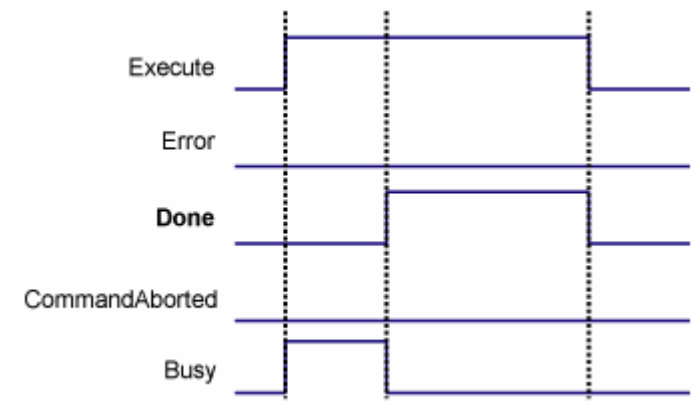


Bild 39) Execution completed without error

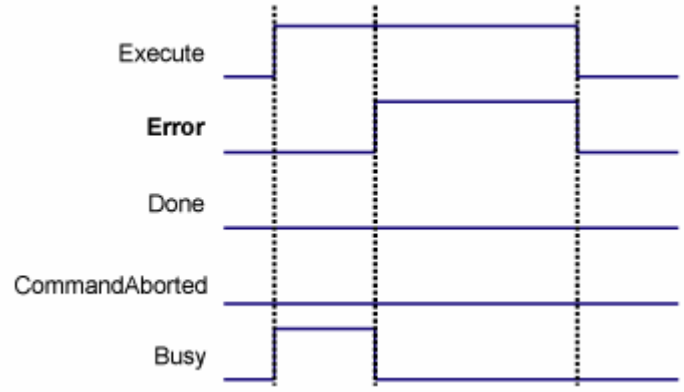


Bild 40) Execution terminated with error

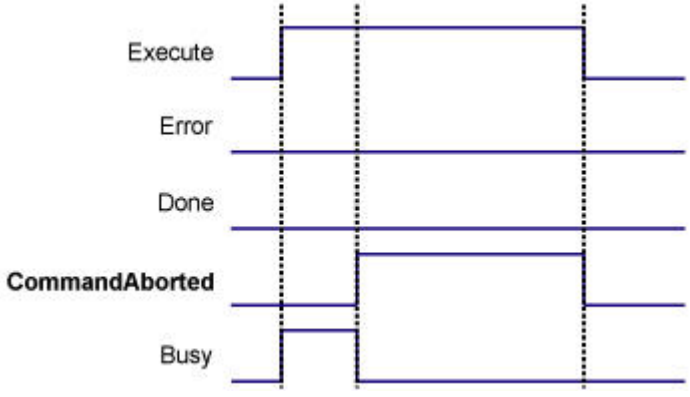


Bild 41) Cancellation of execution; another function block has taken over axis control

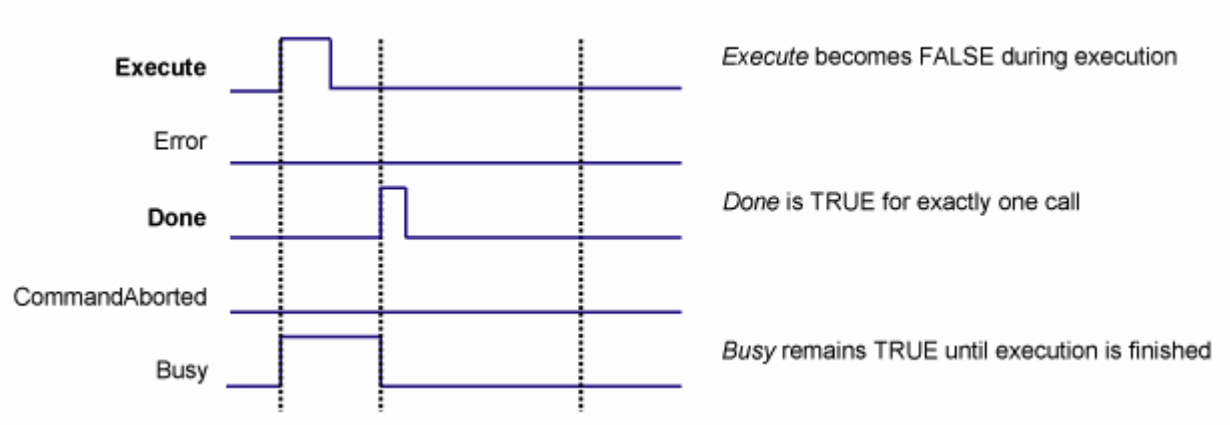


Bild 42) Execution completed without error after *Execute* was set to FALSE during execution

3.2.4 Signal diagram for function blocks with control input Enable

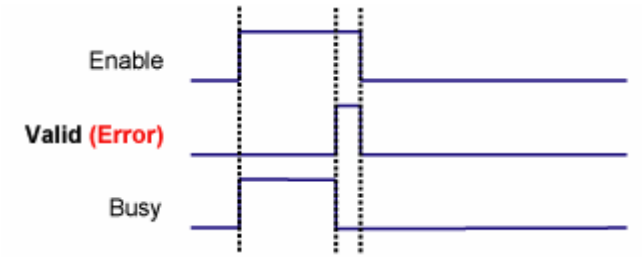


Bild 43) Single execution without (with) error [execution requires more than one call]

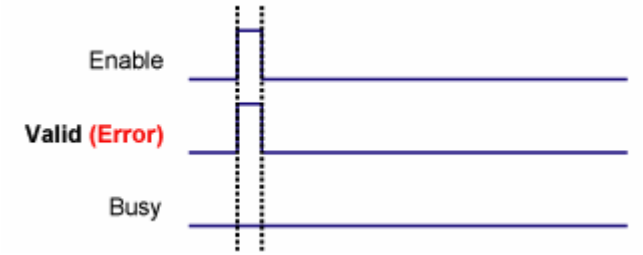


Bild 44) Single execution without (with) error [execution requires only one call]

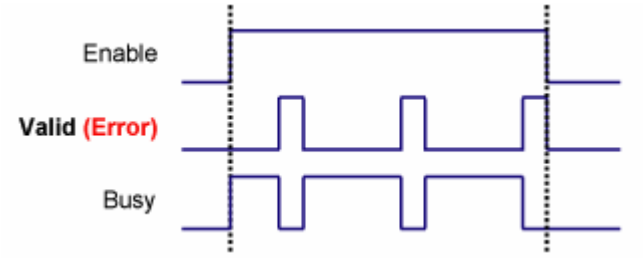


Bild 45) Multiple execution without (with) error [execution requires more than one call]

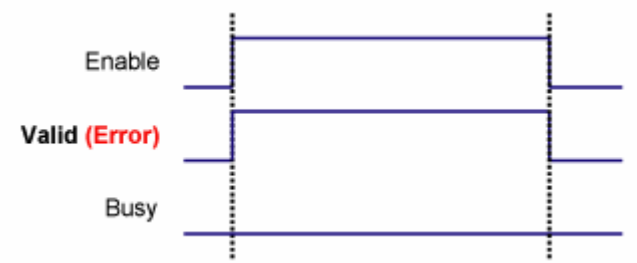


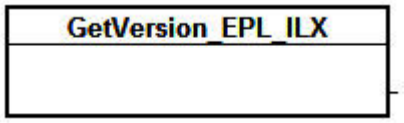
Bild 46) Multiple execution without (with) error [execution requires only one call]

3.3 Description of the function blocks

The following chapter provides descriptions of the function blocks, structured by the use of the function blocks. The chapter after that describes each function block in detail; the list is sorted alphabetically by function block name.

Structure of the detailed description of the function blocks:

Graphical representation



Inputs/outputs

Description of the inputs and outputs with specification of the type, the range of permissible values and the meaning of the values. Example

Output variables

Variable	Possible values, meaning
GetVersion_EPL_ILX	Type DWORD (Value range 16#00000000..16#FFFFFFFF) Contains the library version. In the case of hexadecimal representation, the numbers directly correspond top the version, for example, 16#00001001 stands for version 1.001

Task of the function block

The function block delivers the library version.

Note

Notes on using the function block, for example interdependencies with other function blocks

3.4 Usage of the function blocks

3.4.1 Initialization

The initialization function block switches the drive to a state in which other function blocks can be used. Enables and disables the power stage of the drive.

Function block	Task
MC_Power_EPL_ILX	Enables and disables the power stage of the drive.

3.4.2 Jog

The function block for jog movements moves the drive in Jog operating mode. A short signal at the inputs moves the drive by a short distance in positive or negative direction. In the case of a continuous signal, the drive starts a continuous movement.

Function block	Task
MC_Jog_EPL_ILX	Jog in positive or negative direction

3.4.3 Homing

The operating mode Homing establishes an absolute position reference between the motor position and a defined axis position. Homing is possible by:

– Reference movement or

– Position setting

The reference movement defines the position reference by means of a movement to a switch. During the reference movement, the drive moves to a defined point on the axis. The position is defined by a mechanical switch. Depending on the motor type you use, the follow types of switches are possible:

– Limit switches in positive and negative directions

– Reference switch

– Index switch of the motor encoder system

In addition, it is possible to combine the limit switch and the index pulse. Please refer to the product manual for information on the types of reference movement supported by your Lexium ILx2 drive.

When the position is reached, a position reference is automatically created. This way, the position becomes the absolute user position.

The search speed, the speed for moving away from the switch as well as a safety distance and the distance for moving away from the switch can be adjusted for the reference movement. A reference movement must be completed for the new reference point to be valid. If the reference movement is aborted, it must be started again.

Position setting defines a position reference with reference to the current motor position. Position setting lets you set an axis position as the reference point to which the subsequent position specifications

relate. The reference point for reference positions is moved to the new position setting position.

Position setting is only possible when the motor is at a standstill. Position setting can be used to carry out a continuous absolute positioning without exceeding the positioning limits.

Function block	Task
MC_Home_EPL_ILX	Trigger reference movement
MC_SetPosition_EPL_ILX	Position setting

3.4.4 Operating mode Profile Position

In the Profile Position operating mode, the motor is positioned from a point A to a point B by means of a function block. The positioning distance is specified in with reference to the zero point of the axis (absolute) or with reference to the original target position or the current motor position (relative). Prior to positioning, the reference point must be defined by homing.

Function block	Task
MC_MoveAbsolute_EPL_ILX	Absolute positioning
MC_MoveRelative_EPL_ILX	Relative positioning with reference to the current motor position
MC_MoveAdditive_EPL_ILX	Relative positioning with reference to the original position

3.4.5 Operating mode Profile velocity

In Profile Velocity operating mode, a reference speed for the motor is set and a movement without a target position is started. The motor moves at this speed until a different speed is set or the operating mode is terminated by execution of another function block.

Transitions between two speeds of rotation are defined via the profile generator in the drive. The profile generator calculates the transition to the new speed on the basis of the parameters for acceleration, deceleration, reference speed and actual speed.

Function block	Task
MC_MoveVelocity_EPL_ILX	Start operating mode Profile Velocity

3.4.6 Stopping

Every operating mode can be canceled by stopping the drive. This does not generate an error. The canceled function block terminates the execution by setting the output *CommandAborted* = TRUE. The drive switches to the state "Stopping" and, after the standstill and the reset of the input *Execute* in, to the state "Standstill". New movements are not possible before this is completed.

Function block	Task
MC_Stop_EPL_ILX	Stop drive

3.4.7 Fast position capture

The motor position can be captured with an accuracy depending on the motor type via 2 parameterizable channels. See the appropriate section in the product manual for details on fast position capture. Only the integrated drives ILA2E and ILS2E feature the fast position capture function.

Possible triggering events:

– Rising edge or falling edge at signal input CAP1

– Rising edge or falling edge at signal input CAP2

Function block	Task
MC_TouchProbe_EPL_ILX	Adjust and start position capture
MC_AbortTrigger_EPL_ILX	Cancel position capture

3.4.8 Reading parameters

The following function blocks allow for easy reading of specific parameters. In addition, there is a universal function block which enables a read access to all parameters of the Lexium ILx2P drive. See the product manual for detailed descriptions of all parameters of the drive.

The function blocks for reading the reference position and the reference speed directly from the profile generator only deliver different values than the other function blocks for reading position and speed in the case of the Lexium ILA2E drive. This is due to the fact that the current positions and speeds are determined via the motor encoder in the case of servo motors..

Function block	Task
MC_ActPosRead_EPL_ILX	Read the the current position in user-defined units
MC_ActVelRead_EPL_ILX	Reads the current speed in user-defined units
MC_ReadStatus_EPL_ILX	Reads the current status of the drive
MC_ReadParameter_EPL_ILX	Reads the specified parameters from the drive
GetVersion_EPL_ILX	Returns the library version
ActPosIncRead_EPL_ILX	Returns the current position in increments
RefPosRead_EPL_ILX	Reads the current reference position in user-defined units directly from the profile generator of the drive
RefPosIncRead_EPL_ILX	Reads the current reference position in increments directly from the profile generator of the drive
RefVelRead_EPL_ILX	Reads the current reference speed in user-defined units directly from the profile generator of the drive

3.4.9 Writing parameters

The following function blocks allow for easy writing of specific parameters. In addition, the parameters can be reset to the factory settings or stored permanently in the drive. In addition, there is a universal function block which enables a write access to all parameters of the Lexium ILx2P drive. See the product manual for detailed descriptions of all parameters of the drive.

Function block	Task
MC_WriteParameter_EPL_ILX	Set parameter
SetDriveRamp_EPL_ILX	Set acceleration
StoreParameters_EPL_ILX	Permanently store all parameters
ResetParameters_EPL_ILX	Reset all parameters to the factory settings

3.4.10 Inputs/outputs

The digital inputs and digital outputs of the Lexium ILx2P can be read with the following parameters.

Function block	Task
MC_DigInputRead_EPL_ILX	Read inputs
MC_DigOutputRead_EPL_ILX	Read outputs

3.4.11 Error handling

For error handling purposes, each function block has an *Error* output which is set when an error occurs. The function block MC_ReadAxisError_EPL_ILX is called for detailed analysis of the cause of the error. The error cell is deleted and available for future error messages with MC_Reset_EPL_ILX.

The error cell contains the error code and the error class of an error . A new error is entered providedthe error cell is free. If the error cell not free, the previous error message is not overwritten; instead, the new error message is ignored.

Possible error sources comprise:

– Function block errors

– Drive errors

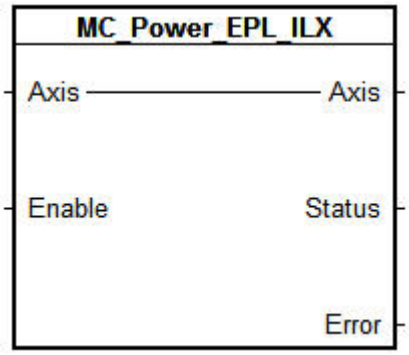
Drive errors can be the result of, for instance, invalid input values. Drive errors are usually caused by events such as reaching a limit switch. MC_Reset_EPL_ILX also resets the error in the drive.

Function block	Task
MC_ReadAxisError_EPL_ILX	Read error
MC_Reset_EPL_ILX	Delete error

3.5 Detailed description of function blocks

3.5.1 MC_Power_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type BOOL (value range TRUE, FALSE) initial value: FALSE TRUE: Switch on motor current. FALSE: Switch off motor current.
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Status	Type: BOOL (value range TRUE, FALSE) initial value: FALSE Indicates the status of the motor current. The status is signaled by the drive after each change. FALSE: Motor current is off. TRUE: Motor current is on.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE

Variable	Possible values, meaning
	TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Switching the motor current on/off . TRUE at the *Enable* input switches the motor current on. As soon as the motor current is switched on, the *Status* output is set. FALSE at the *Enable* input switches the motor current off . As soon as the motor is without current, the *Status* output is reset. If errors occur during execution, the *Error* output is set.

Note

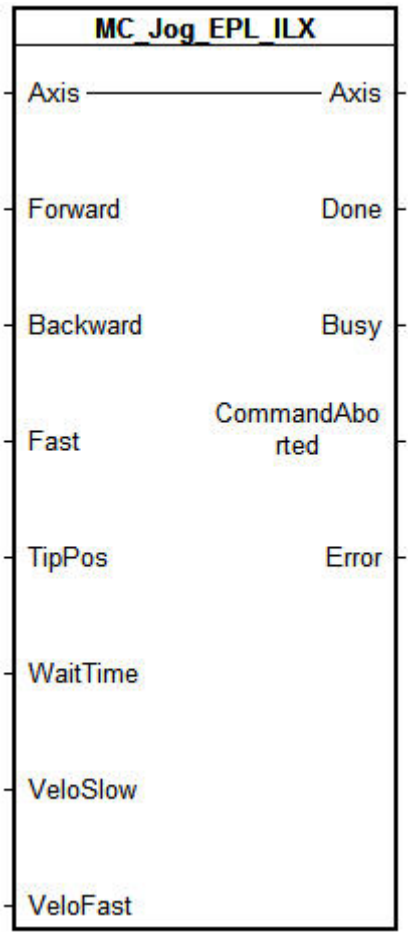
When the power supply to the drive is switched off, the motor current is also switched off. Since the drive can no longer signal this state transition, *Status* remains TRUE. When the power supply to the drive is switched on again, the fact that the motor is without current is signaled and the *Status* output changes to FALSE.

Switching on the motor current causes a transition to one of the states *Standstill* or *ErrorStop*, depending on whether there is an error or not. The drive state is read with MC_ReadStatus_EPL_ILX.

The motor current can be switched off in any state. Any function block that is being executed at this point is aborted.

3.5.2 MC_Jog_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Forward	Type BOOL (value range TRUE, FALSE) initial value: FALSE FALSE: No clockwise movement TRUE: Start clockwise movement
Backward	Type BOOL (value range TRUE, FALSE) initial value: FALSE FALSE: No counterclockwise movement TRUE: Start counterclockwise movement
Fast	Type BOOL (value range TRUE, FALSE) initial value: FALSE The speed can also be selected during

Variable	Possible values, meaning
	movements from two values: FALSE: Speed VeloSlow is selected. TRUE: Speed VeloFast is selected.
TipPos	Type DINT (value range: 0..2147483647) initial value: 20 0: The motor immediately starts a continuous movement. >0: Distance [usr] by which the motor moves after the start before it switches to continuous movement after the delay time (WaitTime) has elapsed.
WaitTime	Type UINT (value range: 1ms..32767ms) initial value: 500ms Delay time [ms], which starts after the motor has moved a defined distance (TipPos) and after which the motor switches to continuous movement.
VeloSlow	Type DINT (value range: 1..13200) initial value: 60 Speed [min-1] for movement if Fast = FALSE.
VeloFast	Type: DINT (value range: 1..13200) initial value: 180 Speed [min-1] for movement if Fast = TRUE.
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE

Variable	Possible values, meaning
	TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
CommandAborted	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution was canceled by another function block. FALSE: Execution not (yet) canceled.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

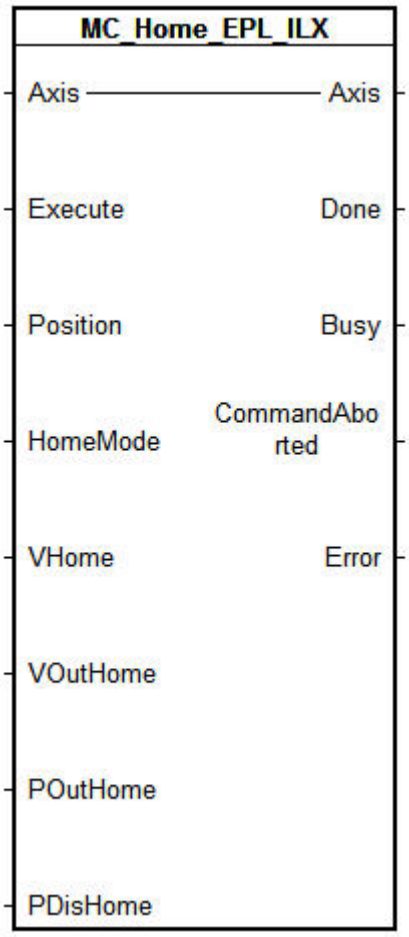
A jog movement is started with TRUE at *Forward* or *Backward*.

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

If *Forward* and *Backward* are TRUE, the operating mode remains active, the jog movement is stopped and *Busy* remains set.

3.5.3 MC_Home_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the

Variable	Possible values, meaning
	function block is called with Execute = FALSE.
Position	Type DINT (value range:-2147483648..2147483647) initial value: 0 Position setting to position setting position (setting the absolute position) in user-defined units. Position is the current motor position after successful homing.
HomeMode	Type: UINT (value range 1..35) initial value: 1 Please see the product manual for the homing methods supported by the drive.
Vhome	Type: UINT (value range: 1..13200) initial value: 60 Speed for searching for the reference switch or a limit switch [min-1]. The drive stops when the switching edge has been detected.
VoutHome	Type: UINT (value range: 1..3000) initial value: 6 Speed for moving away from the switch to the switching edge [min-1]. The maximum distance for searching for the switching edge can be limited with the parameter POutHome.
PoutHome	Type: DINT (value range: 0..2147483647) initial value: 0 0: Monitoring of moving away from switch inactive > 0: Distance for search for switching edge during movement away from switch [usr] , i.e. maximum movement for searching the switching edge. If the switching edge is not found in this distance, the reference movement is aborted with an error.
PdisHome	Type: DINT (value range: 1..2147483647) initial value: 200 Distance from switching edge to reference point in user-defined units When the switching edge is reached, the drive continues to move until the distance is covered.
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block

Variable	Possible values, meaning
	is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

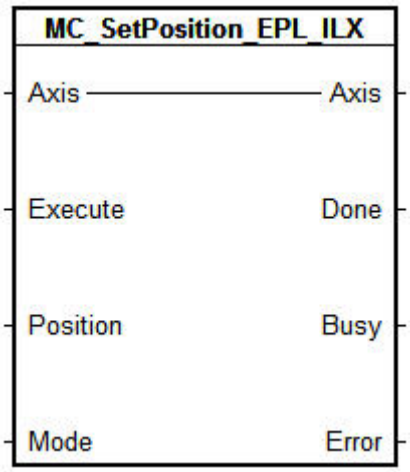
Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
CommandAborted	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution was canceled by another function block. FALSE: Execution not (yet) canceled.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Configuration and start of the reference movement.

3.5.4 MC_SetPosition_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Position	Type: DINT (value range: -2147483648..2147483647) initial value: 0 Position setting to position setting position in user-defined units
Mode	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE FALSE: Set current motor position to position. TRUE: Add position to current motor position.
Axis	Type: Axis_Ref_EPL_ILX

Variable	Possible values, meaning
	(value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

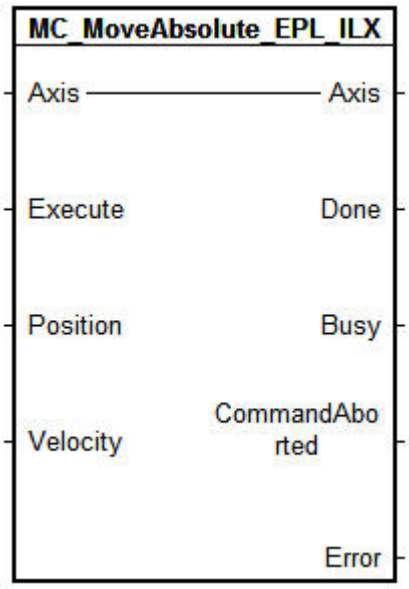
Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Position setting

3.5.5 MC_MoveAbsolute_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Position	Type: DINT (value range: -2147483648..2147483647) initial value: 0 Value for the absolute target position in user-defined units.
Velocity	Type: INT (value range: 1..13200) initial value: 60)

Variable	Possible values, meaning
	Value for the reference speed of the movement [min-1].
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
CommandAborted	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution was canceled by another function block. FALSE: Execution not (yet) canceled.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Positioning to absolute target position *Position* at speed *Velocity*

Note

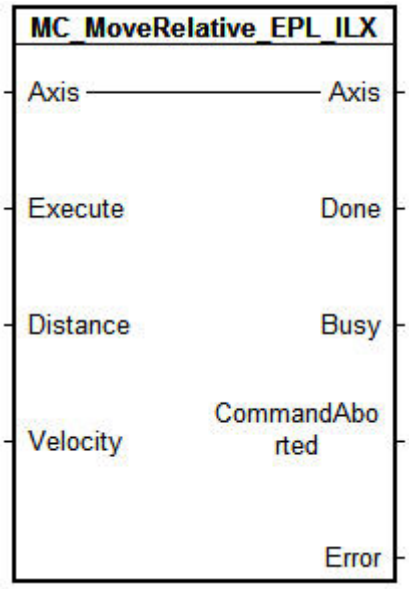
Position overrun

Absolute positioning cannot be started after a position overrun,

because the absolute position reference is lost during the position overrun. The *Referenced* output of the function block MC_ReadStatus_EPL_ILX allows you to read whether or not the absolute position reference is still available.

3.5.6 MC_MoveRelative_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Distance	Type: DINT (value range: -2147483648..2147483647) initial value: 0 Value for the distance with reference to the

Variable	Possible values, meaning
	current motor position in user-defined units.
Velocity	Type: INT (value range: 1..13200) initial value: 60) Value for the reference speed of the movement [min-1].
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

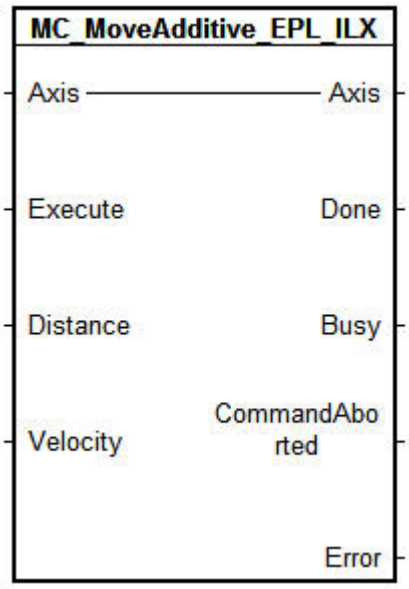
Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
CommandAborted	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution was canceled by another function block. FALSE: Execution not (yet) canceled.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Positioning by distance *Distance* at speed *Velocity*

3.5.7 MC_MoveAdditive_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Distance	Type: DINT (value range: -2147483648..2147483647) initial value: 0 Value for the original target position plus additional relative distance in user-defined units.
Velocity	Type: INT (value range: 1..13200) initial value: 60)

Variable	Possible values, meaning
	Value for the reference speed of the movement [min-1].
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

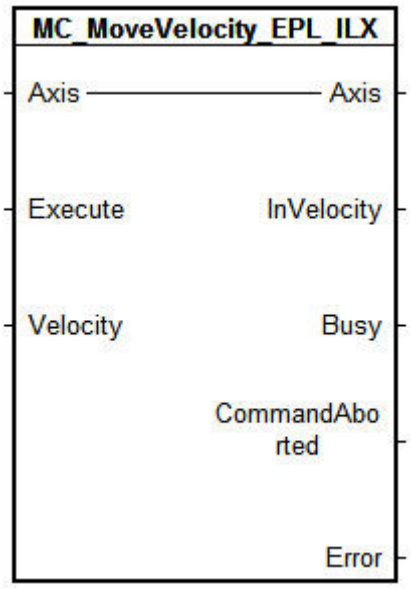
Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
CommandAborted	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution was canceled by another function block. FALSE: Execution not (yet) canceled.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Positioning by distance *Distance* in addition to the original target position as speed *Velocity.*

3.5.8 MC_MoveVelocity_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Velocity	Type: INT (value range: -13200..13200) initial value: 0) Value for the reference speed of the movement [min-1].
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>)

Variable	Possible values, meaning
	initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
InVelocity	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE FALSE: Target speed not (yet) reached. TRUE: Target speed reached.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
CommandAborted	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution was canceled by another function block. FALSE: Execution not (yet) canceled.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Start operating mode Profile Velocity with speed *Velocity*. When the target speed is reached, *InVelocity* is set.

Note

Changing speed *Velocity* to "0"

This allows you to stop the drive with the normal deceleration at any time. If the speed is set to "0“, the drive is only stopped temporarily! This means that as soon as the speed is set to a value not equal to "0", the drive will immediately resume movement.

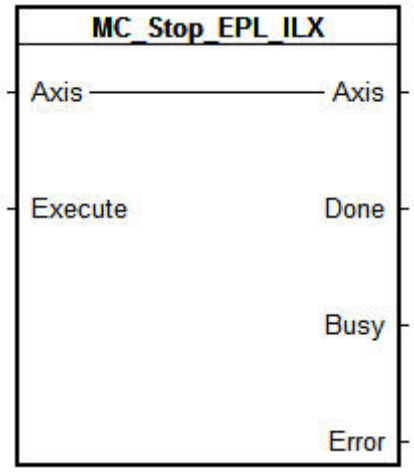
Position overrun

In the Profile Velocity operating mode, the drive may exceed the

position range. This is not an error for the drive, the operating mode continues to run. The *Referenced* output of the function block MC_ReadStatus_EPL_ILX allows you to read whether or not the absolute position reference is still available.

3.5.9 MC_Stop_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Stops the drive with a torque ramp.

Note

This function cannot be aborted by other function blocks. As long as Execute = TRUE, no other function block can be started. Even after standstill, the drive remains blocked.

The function block decelerates the motor with a torque ramp. The parameter LIM_I_maxHalt (see product manual) specifies the current for the torque ramp. After the drive has come to a standstill, the position is determined internally, position control is activated and the motor is held with the power stage enabled.

3.5.10 MC_TouchProbe_EPL_ILX

Graphical representation

Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Channel	Type: UINT (value range: 1..2)

Variable	Possible values, meaning
	initial value 1 Channel number: Selection of the channel to which the other parameters relate.
TriggerLevel	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Triggering signal edge: FALSE: Falling edge TRUE: Rising edge
SingleShot	Type: BOOL (value range: FALSE, TRUE) initial value: TRUE FALSE: If the triggering event occurs again, the captured position is overwritten with the most new position. TRUE: Position capture is switched off after the triggering event so that the captured position cannot be overwritten..
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
CommandAborted	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution was canceled by another function block. FALSE: Execution not (yet) canceled.

Variable	Possible values, meaning
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. A read value at the parameter output RecordedPosition is valid. FALSE: Execution not (yet) terminated without errors. A value at the parameter output RecordedPosition is not (yet) valid.
RecordedPosition	Type DINT (value range: -2147483648..2147483647) initial value: 0 Captured motor position when the triggering event occurs

Task of the function block

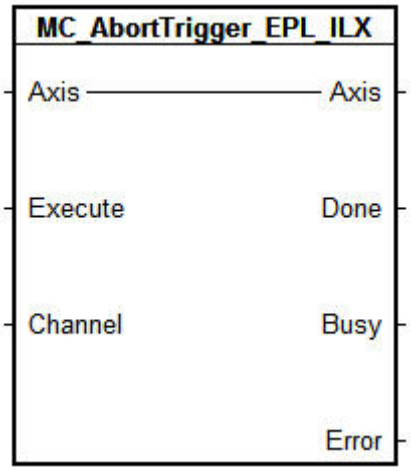
Adjust and start position capture.

Note

Only the drives ILA2E and ILS2E feature the fast position capture function.

3.5.11 MC_AbortTrigger_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Channel	Type: UINT (value range: 1..2) initial value 1 1: Abort position capture via channel 1 (CAP1). 2: Abort position capture via channel 2 (CAP2).
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE)

Variable	Possible values, meaning
	initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

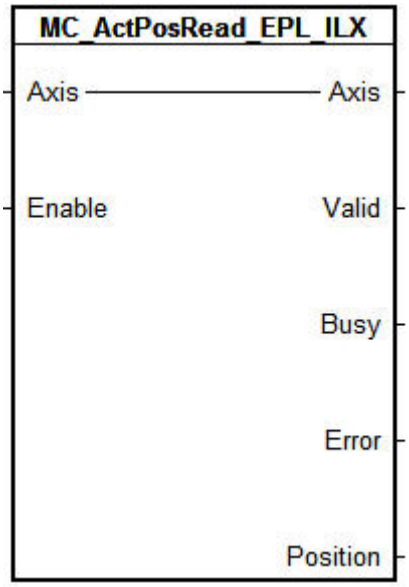
Abort position capture

Note

Only the drives ILA2E and ILS2E feature the fast position capture function.

3.5.12 MC_ActPosRead_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately

Variable	Possible values, meaning
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

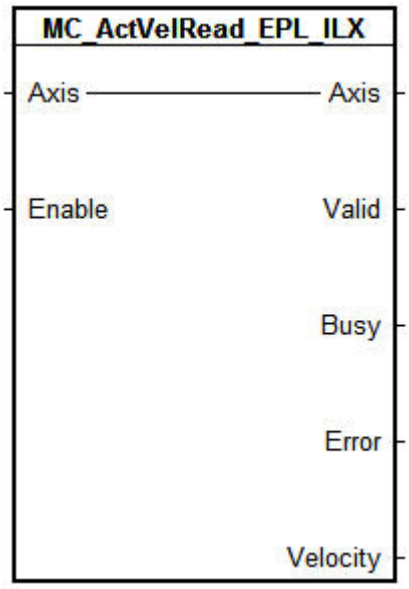
Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Position	Type: DINT (Value range:) Actual motor position in user-defined units

Task of the function block

Read the actual position of the motor in user-defined units

3.5.13 MC_ActVelRead_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without

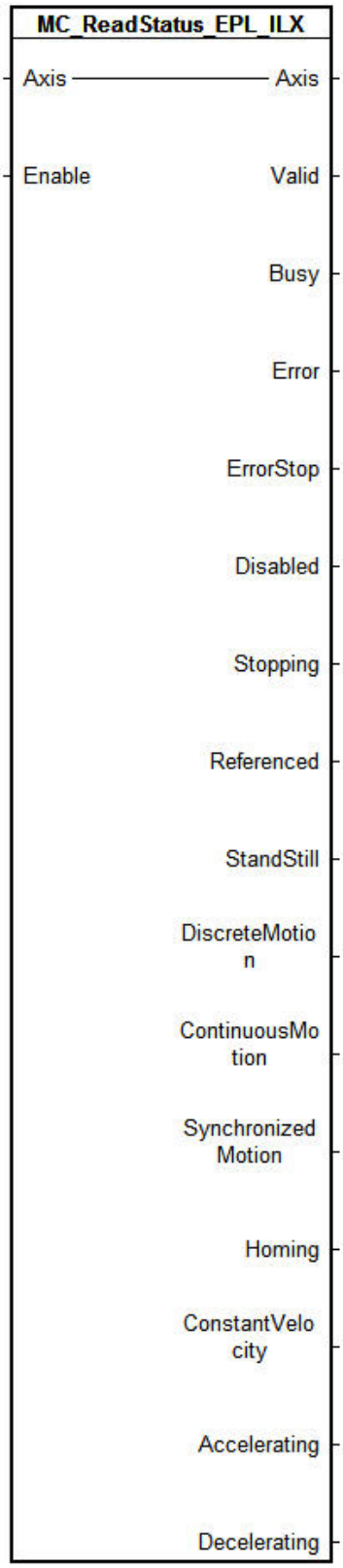
Variable	Possible values, meaning
	errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Velocity	Type: INT (value range: -13200..+13200) Current speed of rotation of motor [min-1]

Task of the function block

Read the current speed of rotation of the motor [min-1]

3.5.14 MC_ReadStatus_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Errorstop	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Movement was stopped due to an error
Disabled	Type: BOOL (value range: FALSE, TRUE) Initial value: FALSE

Variable	Possible values, meaning
	TRUE: Motor current is switched off. FALSE: Motor current is switched on.
Stopping	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Function block MC_Stop_EPL_ILX is executed; movement is being stopped.
Referenced	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Drive is homed; position reference with relation to the mechanical system is known.
Standstill	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Drive is at a standstill.
DiscreteMotion	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Drive is in Profile Position operating mode.
ContinuousMotion	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The drive is in an operating mode with continuous motion, the Profile Velocity operating mode.
SynchronizedMotion	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The drive is in an operating mode with synchronized motion, such as Electronic Gear operating mode.
Homing	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE) TRUE: The drive is in operating mode Homing.
ConstantVelocity	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The motor rotates at constant speed.
Accelerating	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The drive accelerates.

Variable	Possible values, meaning
Decelerating	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The drive decelerates.

Task of the function block

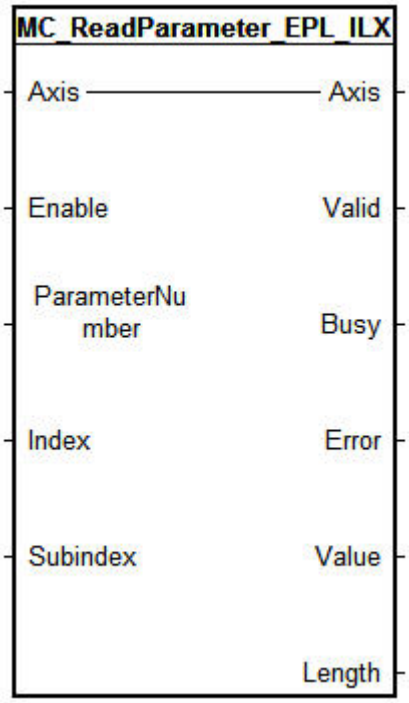
The current drive state is output.

Note

The drive is exactly in one of the states Zustände *StandStill*, *Homing*, *DiscreteMotion*, *ContinuousMotion*, *SynchronizedMotion*, *Stopping*, *Disabled* or *Errorstop*. The output with the corresponding name of the function block is then TRUE.

3.5.15 MC_ReadParameter_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the

Variable	Possible values, meaning
	function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately
Parameter Number	Type: INT (value range: 0..65535) 0: Parameter is selected with index and subindex. >0: Number of the parameter to be read: 1: Current reference position of profile generator [usr] 2: Position of positive software limit switch [usr] 3: Position of negative software limit switch [usr] 4: Enable (bit0=1) or disable (bit0=0) positive software limit switch 5: Enable (bit0=1) or disable (bit0=0) negative software limit switch 10: Actual speed [min-1] 11: Current reference speed [min-1] Other numbers are not supported.
Index	Type: UINT (value range: 8192..65535) Index of the object to be read; the objects are listed in the manual with their indexes and subindexes. Only indexes greater than 8191 can be read with this function. An error is passed in the case of access to a smaller index. Only B&R system functions can read indexes < 8192. Only valid if ParameterNumber = 0.
Subindex	Type: UINT (value range: 0..255) Sub-index of the Object to be read – the Objects are listed in the manual with their index and sub-index. Only valid if *ParameterNumber* = 0.
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

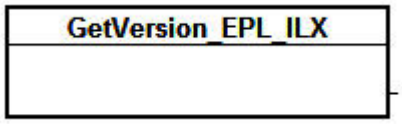
Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Value	Type: DINT (value range: -2147483648..2147483647) initial value: 0 Value of device parameter
Length	Type: UINT (value range: 0..65535) initial value: 0 Length in bytes of read device parameter

Task of the function block

Read an object from the device parameter list.

3.5.16 GetVersion_EPL_ILX

Graphical representation



Output variables

Variable	Possible values, meaning
GetVersion_EPL_ILX	Type DWORD (Value range 16#00000000..16#FFFFFFFF)

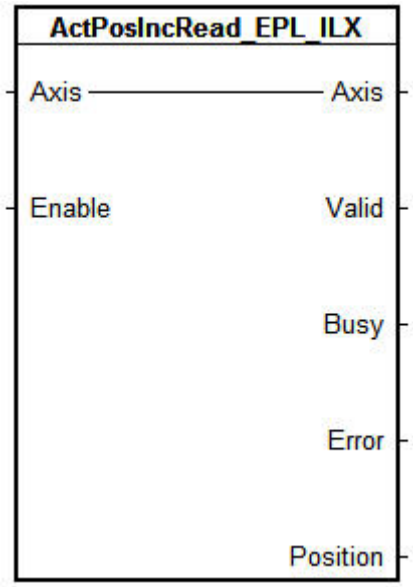
Variable	Possible values, meaning
	Contains the library version. In the case of hexadecimal representation, the numbers directly correspond top the version, for example, 16#00001001 stands for version 1.001

Task of the function block

Returns the version number of this function block library.

3.5.17 ActPosIncRead_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

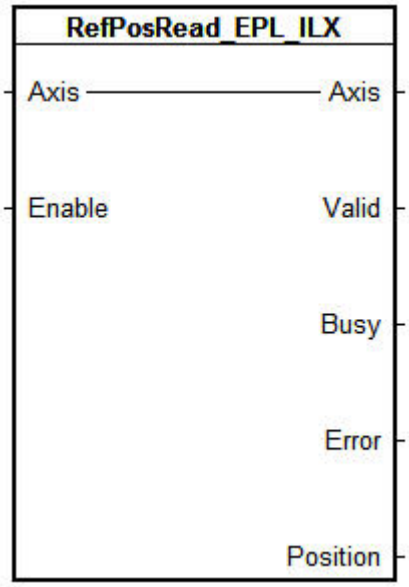
Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. The read value at the parameter output *Position* is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output *Position* is not (yet) valid.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Position	Type: DINT (value range: -2147483648..2147483647)) Actual motor position in increments

Task of the function block

Read actual motor position in increments.

3.5.18 RefPosRead_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning

Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. The read value at the parameter output *Position* is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output *Position* is not (yet) valid.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Position	Type: DINT (value range: -2147483648..2147483647) Current reference position in the profile generator in user-defined units

Task of the function block

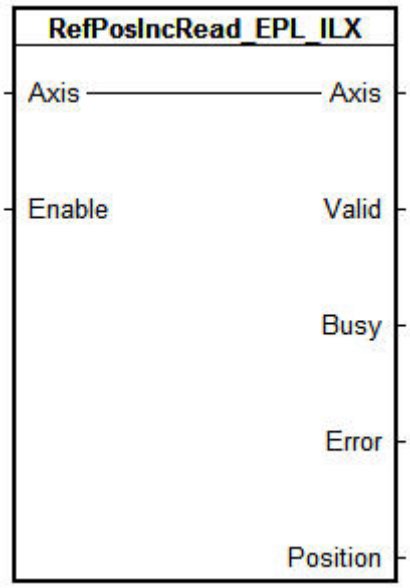
Read the current reference position in the profile generator.

Note

This function block for reading the reference position directly from the profile generator only delivers values different from the function block MC_ActPosRead_EPL_ILX in the case of the Lexium ILA2E.

3.5.19 RefPosIncRead_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors.

Variable	Possible values, meaning
	The read value at the parameter output *Position* is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output *Position* is not (yet) valid.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Position	Type: DINT (value range: -2147483648..2147483647) Current reference position in the profile generator in motor increments.

Task of the function block

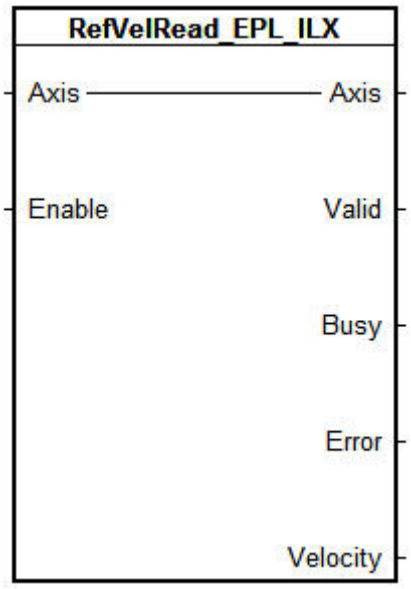
Read the current reference position in the profile generator in motor increments.

Note

This function block for reading the reference position directly from the profile generator only delivers values different from the function block MC_ActPosRead_EPL_ILX in the case of the Lexium ILA2E.

3.5.20 RefVelRead_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. The read value at the parameter output

Variable	Possible values, meaning
	Velocity is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output Velocity is not (yet) valid.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Velocity	Type: INT (value range: -13200..+13200) Current speed of rotation in profile generator [min-1].

Task of the function block

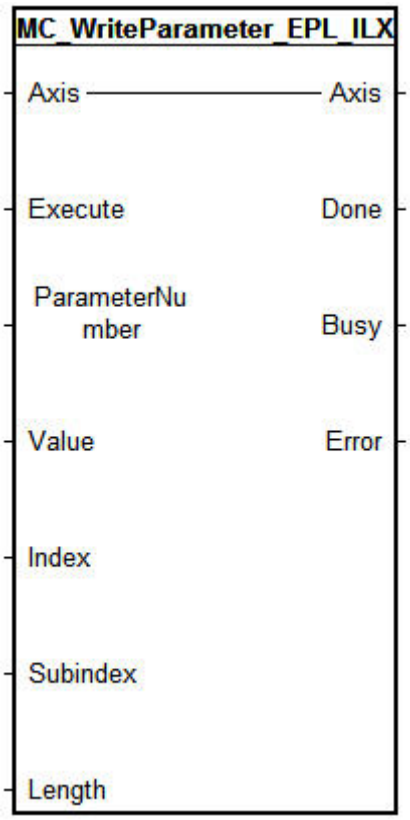
Reads the current speed of rotation in the profile generator.

Note

This function block for reading the reference speed directly from the profile generator only delivers values different from the function block MC_ActVelRead_EPL_ILX in the case of the Lexium ILA2E.

3.5.21 MC_WriteParameter_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
ParameterNumber	Type: INT (value range: 0..65535)

Variable	Possible values, meaning
	0: Parameter is selected with index and subindex. >0: Number of the parameter to be written: 2: Position of positive software limit switch [usr] 3: Position of negative software limit switch [usr] 4: Enable (bit0=1) or disable (bit0=0) positive software limit switch 5: Enable (bit0=1) or disable (bit0=0) negative software limit switch Other numbers are not supported.
Value	Type: DINT (value range: -2147483648..2147483647) initial value: 0 Value of device parameter
Index	Type: UINT (value range: 8192..65535) Index of the object to be written; the objects are listed in the manual with their indexes and subindexes. Only indexes greater than 8191 can be written with this function. An error is passed in the case of access to a smaller index. Only B&R system functions can write indexes < 8192. Only valid if ParameterNumber = 0.
Subindex	Type: UINT (value range: 0..255) Subindex of the object to be written; the objects are listed in the manual with their indexes and subindexes. Only valid if ParameterNumber = 0.
Length	Type: UINT (value range: 0..65535) initial value: 0 Length in bytes of device parameter to be written
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning

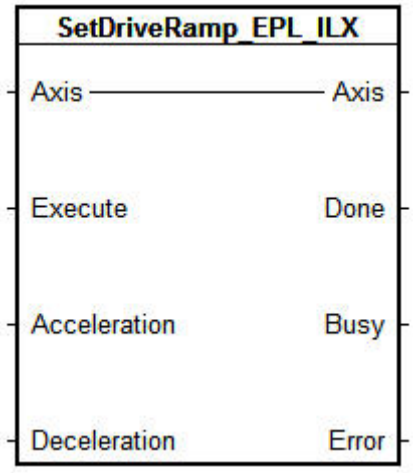
Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Write an object in the parameter list.

3.5.22 SetDriveRamp_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE

Variable	Possible values, meaning
	Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Acceleration	Type: UDINT (value range: 1..3 000 000) initial value: 600 Value for steepness of acceleration ramp [1min-1/s]
Deceleration	Type: UDINT (value range: 200..3 000 000) initial value: 750 Value for steepness of deceleration ramp [1min-1/s]
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL

Variable	Possible values, meaning
	(value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

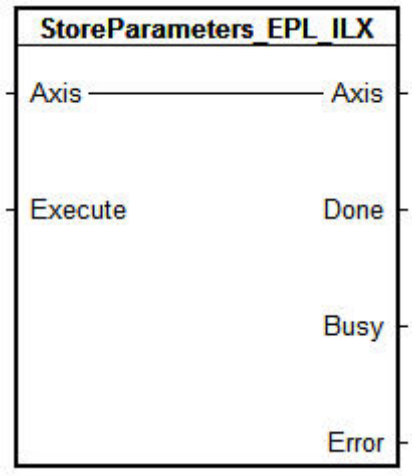
Defines the acceleration and deceleration ramps. The device controls the acceleration and deceleration behavior of the motor with ramp functions. The ramp steepness determines the change in speed of the motor. *Acceleration* defines the acceleration, *Deceleration* the deceleration.

Note

The drive absorbs excess braking energy during deceleration. If the DC bus voltage exceeds a permissible limit value, the drive switches off the power stage and signals an "Overvoltage" error. The motor then coasts down without any braking force. The steepness of the acceleration and the deceleration ramps should be set so the motor decelerates as quickly as possible without the power stage being switched off because of overvoltage.

3.5.23 StoreParameters_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the

Variable	Possible values, meaning
	execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

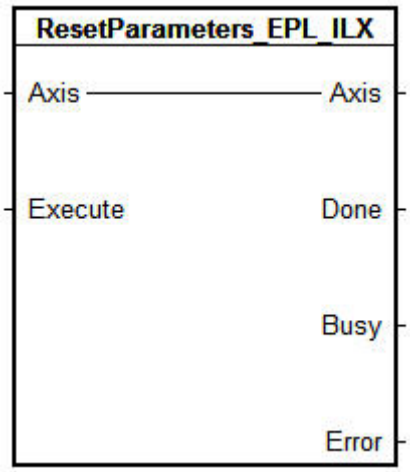
Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Save all parameters to the EEPROM of the Lexium ILx2P drive.

3.5.24 ResetParameters_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning

Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

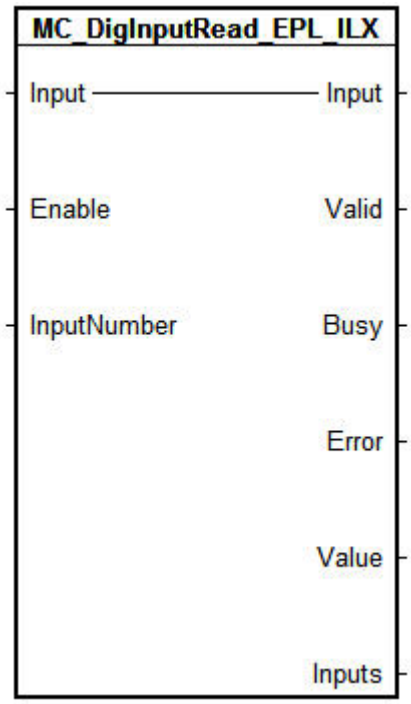
Reset all parameters to their state after the "First Setup" (see manual). All parameter values are reset to the default values with the exception of the communication parameters, the control mode and the logic type ("source" or "Sink" of the inputs/outputs).

Note

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

3.5.25 MC_DigInputRead_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately
Input Number	Type: INT (value range: 0..5) initial value: 0 Number of the input to be read: 0: LIO0 1: LIO1 2: LIO2 3: LIO3
Input	Type: Input_Ref_EPL_ILX (value range: <name of axis>) initial value: empty

Variable	Possible values, meaning
	Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. The read value at the parameter output *Value* is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output *Value* is not (yet) valid.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Value	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Read output has 24V level. FALSE: Read output has 0 level.
Inputs	Type: WORD (value range: 16#00..16#3F) initial value: 16#00 Input assignment of the drive: Input Bit Signal LIO0 0 /LIMP LIO1 1 /LIMN LIO2 2 Available as required LIO3 3 /REF

Task of the function block

Reads the current input assignment of the drive.

Note

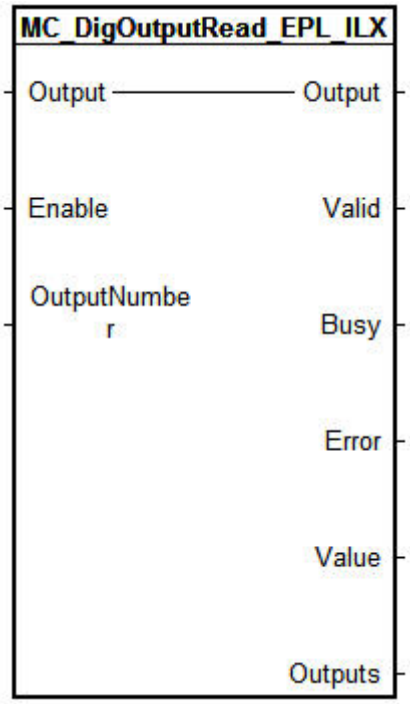
The meaning of the signal is described in the product manual.

The input I0 (/REF) is only used by the drive for homing to the reference signal (see MC_Home_EPL_ILX). If this function is not used, the input can be used for other purposes as required.

The limit switch function of the inputs I1 (/LIMN) and I2 (/LIMP) can be disabled. If the limit switch function is not used, the inputs can be used for other purposes as required.

3.5.26 MC_DigOutputRead_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error

Variable	Possible values, meaning
	become FALSE immediately
Output Number	Type: INT (value range: 0..2) initial value: 0 Number of the output to be read: 0: O0 NO_FAULT_OUT 1: O1 BRAKE_OUT 2: O2 ACTIVE2_OUT
Output	Type: Input_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. The read value at the parameter output *Value* is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output *Value* is not (yet) valid.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Value	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Read output has 24V level.

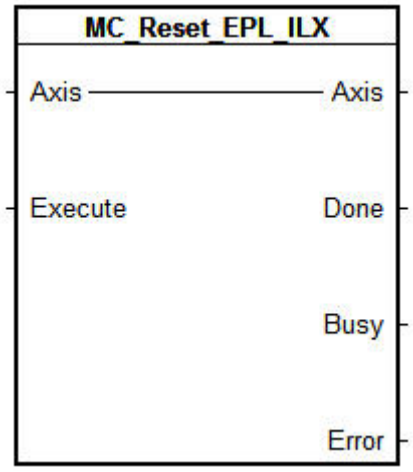
Variable	Possible values, meaning
	FALSE: Read output has 0 level.
Outputs	Type: WORD (value range: 16#00..16#03) initial value: 16#00 Output assignment of the drive: Output Bit LIO0 0 LIO1 1 LIO2 2 LIO3 3

Task of the function block

Reads the current output assignment of the drive.

3.5.27 MC_Reset_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	Type BOOL (value range: FALSE, TRUE) initial value: FALSE Edge-sensitive: FALSE->TRUE starts the execution. A new rising edge continues the execution with the input parameter that is then active. Execution is terminated once the Busy output is FALSE . After termination of the execution, Execute determines the behavior of the outputs: FALSE: At the same time as Busy = FALSE,

Variable	Possible values, meaning
	either Done, Error or CommandAborted becomes TRUE for exactly one call. TRUE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE and remains TRUE until the function block is called with Execute = FALSE.
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

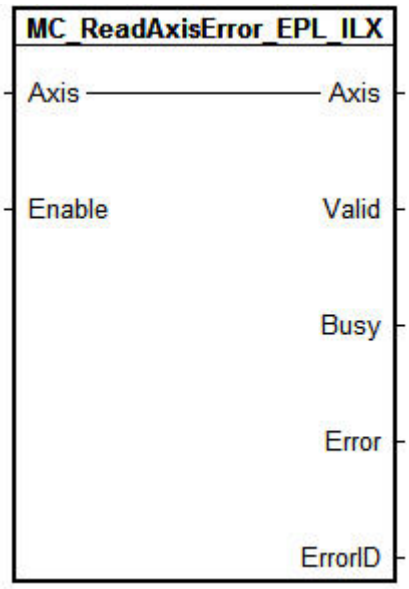
Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.

Task of the function block

Error acknowledgement. The error cell is cleared so that it is available for future error messages. 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.

3.5.28 MC_ReadAxisError_EPL_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Enable	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE Level-sensitive; starts or stops execution of the function block. TRUE: Function block is executed repeatedly. FALSE: Execution is terminated immediately; the control outputs Valid, Busy and Error become FALSE immediately
Axis	Type: Axis_Ref_EPL_ILX (value range: <name of axis>) initial value: empty Name of the drive for which the function block is to be executed. A global data structure must be created for each drive. It is passed here as a parameter.

Output variables

Variable	Possible values, meaning
Valid	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors.

Variable	Possible values, meaning
	The read value at the parameter output *ErrorID* is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output *ErrorID* is not (yet) valid.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
ErrorID	Type: WORD (value range: 16#0000...16#FFFF) initial value: 16#0000 0: No error message in the error cell > 0: Error number (see list of error numbers in the appendix).

Task of the function block

Reads device error.

4 Appendix

4.1 Error numbers

The error numbers are the return values of the function block MC_ReadAxisError_EPL_ILX.

ErrorID hex.	ErrorID dec.	Error class	Description
Up to 16#00FF	Up to 255		See CiA405 error messages
			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 (subindex)
16#1103	4355	0	Writing of parameter not permissible (read only)
16#1104	4356	0	Write access denied (no access authorization)
16#1106	4358	0	Command not allowed while power stage is active
16#1107	4359	0	Access via other interface blocked
16#1108	4360	0	Parameter cannot be read (Block Upload)
16#1109	4360	0	Power fail data invalid
16#110A	4362	0	No bootloader present
16#110B	4363	3	Initialization error
16#1300	4864	3	Safety function Safe Torque OFF triggered (STO_A and STO_B)
16#1301	4865	4	Inputs of the STO_A and STO_B safety function have different levels
16#1310	4880	3	Reference signal frequency too high
16#1603	5635	0	Capture memory used by other function
16#1606	5638	0	Capture still active
16#1607	5639	0	No trigger parameter defined for capture
16#1608	5640	0	Trigger option not permissible for trigger parameter
16#1609	5641	0	No capture channel defined
16#160A	5642	0	No capture data available
16#160B	5643	0	Parameter cannot be logged
16#160C	5644	1	Autotuning: Moment of inertia outside of permissible range
16#160E	5646	1	Autotuning:Ttest movement could not be started
16#160F	5647	1	Autotuning: Power stage cannot be enabled
16#1610	6548	1	Autotuning: Processing aborted
16#1611	5649	1	System error: Autotuning internal write access
16#1612	5650	1	System error: Autotuning internal read access
16#1613	5651	1	Autotuning: Max. permissible positioning range exceeded
16#1614	5652	0	Autotuning: Already active
16#1617	5655	1	Autotuning Friction torque or load torque too great
16#1618	5656	1	Autotuning: Optimization failed
16#1A00	6656	0	System error: FIFO memory overflow
16#1A01	6657	3	Motor has been changed
16#1A02	6658	3	Motor has been changed
16#1B00	6912	4	System error: Incorrect parameters for motor and power stage
16#1B01	6913	3	User parameter max. speed of rotation too high
16#1B02	6914	3	User parameter max. current, holding current or Quick Stop current too high
16#2300	8960	3	Power stage overcurrent
16#2301	8961	3	Overcurrent braking resistor

ErrorID hex.	ErrorID dec.	Error class	Description
16#3100	12544	par.	Phase error mains supply
16#3200	12800	3	DC bus overvoltage
16#3201	12801	3	DC bus undervoltage (switch-off threshold)
16#3202	12802	2	DC bus undervoltage (Quick Stop threshold)
16#3203	12803	4	Motor encoder supply voltage
16#3206	12806	0	DC bus undervoltage (warning)
16#4100	16640	3	Power stage overtemperature
16#4101	16641	0	Warning power stage overtemperature
16#4102	16642	0	Warning power stage overload (I^2t)
16#4200	16896	3	Device overtemperature
16#4300	17152	3	Motor overtemperature
16#4301	17153	0	Warning motor overtemperature
16#4302	17154	0	Warning motor overload (I^2t)
16#4402	17410	0	Warning braking resistor overload (I^2t)
16#5200	20992	3	No connection to motor encoder
16#5201	20993	4	Error in motor encoder communication
16#5202	20994	4	Motor encoder is not supported
16#5203	20995	4	No connection to motor encoder
16#5204	20996	3	Connection to motor encoder lost
16#5430	21552	0	System error: EEPROM read error
16#5431	21553	0	System error: EEPROM write error
16#5435	21557	0	System error: EEPROM not formatted
16#5437	21559	0	System error: EEPROM checksum error manufacturer data
16#5438	21560	0	System error: EEPROM checksum error user parameters
16#5439	21561	0	System error: EEPROM checksum error CAN parameters
16#543A	21562	0	System error: EEPROM HardwareInfo invalid
16#543B	21563	0	System error: EEPROM manufacturer data invalid
16#543C	21564	0	System error: EEPROM CAN data invalid
16#543D	21565	0	System error: EEPROM user parameters invalid
16#5600	22016	3	Motor connection phase error
16#5601	22017	4	Interruption or incorrect motor encoder signals
16#5602	22018	4	Interruption or incorrect motor encoder signals
16#5603	22019	4	Commutation error
16#6107	24839	0	Parameter outside of value range (calculation error)
16#6108	24840	0	Function not available
16#610D	24845	0	Error in selection parameter
16#610F	24847	4	Internal time base failed (timer 0)
16#7120	28960	4	Invalid motor data
16#7121	28961	2	System error: Error in motor encoder communication
16#7122	28962	4	Invalid motor data
16#7123	28963	4	Motor current offset outside of permissible range
16#7124	28964	4	System error: motor encoder defective
16#7200	29184	4	System error: calibration analog/digital converter
16#7201	29185	4	System error: Motor encoder initialization (quadrant evaluation)
16#7327	29479	4	System error: Position sensor not ready
16#7328	29480	4	Motor encoder signals: Incorrect position capture
16#7329	29481	0	Motor encoder signals: Warning
16#7330	29482	4	System error: Motor encoder (Hiperface)
16#7331	29483	4	System error: Motor encoder initialization
16#7333	29485	4	System error: Deviation in calibration of analog/digital converter
16#7334	29486	3	System error: Analog/digital converter offset too great
16#7335	29487	0	Communication with motor encoder busy
16#7336	29488	3	Offset during SinCos drift compensation too high

ErrorID hex.	ErrorID dec.	Error class	Description
16#7337	29489	1	Writing of offset not be successful
16#7400	29696	0	System error: Invalid interrupt (XINT2)
16#7500	29952	0	Modbus: Overrun error
16#7501	29953	0	Modbus: Framing error
16#7502	29954	0	Modbus: Parity error
16#7503	29955	0	Modbus:Rreceive error
16#8110	33040	0	Powerlink: Overflow (message lost)
16#8130	33072	2	Powerlink: Heartbeat or Life Guard error
16#8201	33281	0	Powerlink: RxPO1 could not be processed
16#8202	33282	0	Powerlink: RxPO2 could not be processed
16#8203	33283	0	Powerlink: RxPO3 could not be processed
16#8204	33284	0	Powerlink: RxPO1 could not be processed
16#8205	33285	0	Powerlink: TxPO could not be processed
16#8206	33286	0	Powerlink: Overflow internal queue message lost
16#A060	41056	2	Calculation error electronic gear
16#A061	41057	2	Change in reference value for electronic gear too great
16#A300	41728	0	Torque ramp with HALT current active
16#A301	41729	0	Drive in state 'QuickStopActive'
16#A302	41730	1	Interruption by LIMP
16#A303	41731	1	Interruption by LIMN
16#A304	41732	1	Interruption by REF
16#A305	41733	0	Power stage cannot be enable in current operating state of state machine
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 state 'Fault'
16#A309	41737	0	Drive not in state 'OperationEnable'
16#A310	41744	0	Power stage not enabled
16#A312	41746	0	Profile generation interrupted
16#A313	41747	0	Position overrun (pos_over=1), therefore, reference point is no longer defined (ref_ok=0)
16#A314	41748	0	No reference position
16#A315	41749	0	Homing active
16#A316	41750	0	Overrun calculation of acceleration
16#A317	41751	0	Drive is not at a standstill
16#A318	41752	0	Operating mode active (x_end=0)
16#A319	41753	1	Manual tuning/autotuning: Distance range exceeded
16#A31A	41754	0	Manual tuning/Autotuning: Amplitude/offset too high
16#A31B	41755	0	HALT requested
16#A31C	41756	0	Invalid position setting with software limit switch
16#A31D	41757	0	Speed range exceeded (CTRL_n_max)
16#A31E	41758	1	Interruption by positive software limit switch
16#A31F	41759	1	Interruption by negative software limit switch
16#A320	41760	par.	Position tracking error
16#A321	41761	0	RS422 position interface is not defined as input
16#A324	41764	1	Error during homing (additional info = detailed error number)
16#A325	41765	1	Limit switch to be approached not enabled
16#A326	41766	1	REF switch not found between LIMP and LIMN
16#A327	41767	1	Reference movement to REF without reversal of direction, invalid activation of limit switch LIM
16#A328	41768	1	Reference movement to REF without reversal of direction, 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 negative direction

ErrorID hex.	ErrorID dec.	Error class	Description
16#A32B	41771	1	Ext. monitoring signal LIMN with positive direction
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	Unreliable reproducibility of the index pulse movement, index pulse too close to the switch
16#A331	41777	3	No start-up operating mode selected for local control mode
16#A332	41778	1	Jog error (additional info = detailed error number)
16#A334	41780	2	Timeout standstill window monitoring
16#A335	41781	1	Processing only possible in fieldbus operation
16#B100	45312	0	Modbus: Unknown service
16#B200	45568	0	Modbus: Protocol error
16#B201	45569	2	Modbus: Nodeguard error
16#B202	45570	0	Modbus: Nodeguard warning
16#B203	45571	0	Modbus: Incorrect number of monitor objects
16#B204	45572	0	Modbus: Service too long
16#B600	46592		Powerlink: Initialization error
16#B601	46593		Powerlink: Realtime data error
16#B602	46594		Powerlink: Realtime data warning
16#B603	46595		Powerlink: Protocol-specific error
16#B604	46596		Powerlink: Protocol-specific warning
16#B605	46597		Powerlink: Unknown error
16#B606	46598		Powerlink: Delayed parameter access to module
16#B607	46599		Powerlink: Is currently processing another request
16#B608	46600		Powerlink: Realtime Hot-Reset
16#B609	46601		Powerlink: Realtime Hot-Stop
16#B60A	46602		Powerlink: Timeout in internal communication
16#B60B	46603		Powerlink: Error in internal communication
			Library error messages
16#FF00	65280	0	Toggle bit not toggled
16#FF01	65281	0	Time-out during SDO transfer
16#FF02	65282	0	Command specifier CS incorrect or unknown
16#FF03	65283	0	Invalid block size (only in Block Mode)
16#FF04	65284	0	Invalid sequence number (only in Block Mode)
16#FF05	65285	0	CRC error (only in Block Mode)
16#FF06	65286	0	No memory available
16#FF07	65287	0	Access to object impossible
16#FF08	65288	0	No read access, because write-only object (wo)
16#FF09	65289	0	No write access, because read object (ro)
16#FF0A	65290	0	Object does not exist in object dictionary
16#FF0B	65291	0	Object does not support PDO mapping
16#FF0C	65292	0	Number or length of objects exceed the byte length of the PDO
16#FF0D	65293	0	Parameters are incompatible
16#FF0E	65294	0	Device detects internal incompatibility
16#FF0F	65295	0	Hardware error, access denied
16#FF10	65296	0	Data type and parameter length do not match
16#FF11	65297	0	Data type does not match, parameter too long
16#FF12	65298	0	Data type does not match, parameter too short
16#FF13	65299	0	Subindex not supported
16#FF14	65300	0	Value range of parameter too large (relevant only for write access)
16#FF15	65301	0	Parameter values too great
16#FF16	65302	0	Parameter values too small

ErrorID hex.	ErrorID dec.	Error class	Description
16#FF17	65303	0	Upper value is less than lower value
16#FF18	65304	0	General error
16#FF19	65305	0	Data cannot be transmitted to the application and cannot be saved.
16#FF1A	65306	0	Device control is local, data cannot be transmitted or saved.
16#FF1B	65307	0	Device status keeps data from being transmitted and saved.
16#FF1C	65308	0	Object dictionary does not exist or cannot be generated (for example, if data error occurs during generation from file)
16#FF1D	65309	0	Reserved
16#FF1E	65310	0	Reserved
16#FF1F	65311	0	Reserved
16#FF20	65312	0	Unknown status
16#FF21	65313	0	Input variable was changed before response was received (read/ write parameter)
16#FF22	65314	0	An attempt was made to interrupt a non-interruptible function block (MC Power, MC Stop, MC Home, MC SetPosition)
16#FF23	65315	0	Trigger function already active
16#FF24	65316	0	PDO Timeout
16#FF25	65317	0	Gear processing is not active (BL_GearOffset)
16#FF26	65318	0	Reserved
16#FF27	65319	0	Drive is not in state StandStill
16#FF28	65320	0	Error connection monitoring
16#FF29	65321	0	Data set processing is not active (BL_DataSetChange)
16#FF2A	65322	0	Trigger event lost
16#FF2B	65323	0	Reserved
16#FF2C	65324	0	Synchronous mode is active, function cannot be executed
16#FF2D	65325	0	Variable not initialized
16#FF2E	65326	0	Variable already initialized
16#FF2F	65327	0	Node does not send Heartbeat
16#FF30	65328	0	Index outside of valid value range
16#FF31	65329	0	Subindex outside of valid value range
16#FF32	65330	0	Data set error in HMI
16#FF34	65332	0	Power stage does not switch to state Enabled
16#FF35	65333	0	Incorrect device program
16#FF36	65334	0	Operating mode not supported by drive
16#FF37	65335	0	Power stage is not in state Enabled
16#FF38	65336	0	Parameter list has not yet been read by the device
16#FF39	65337	0	Parameter list and device do not match
16#FF3A	65338	0	NMT state remains Pre-Operational (16#7F) (switch the device off and on again)
16#FF3B	65339	0	Power stage is not in state Disabled