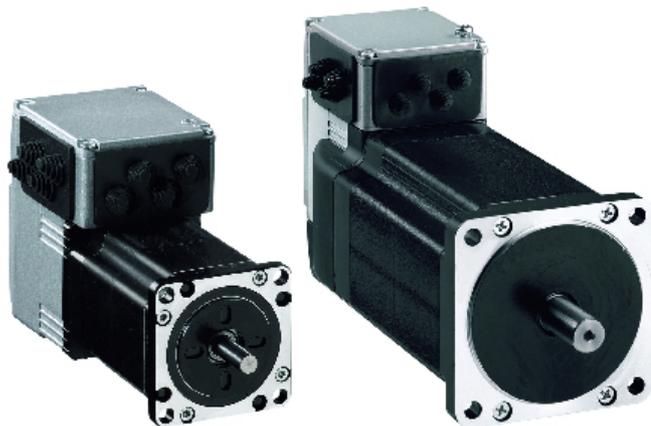


ILx2 EtherCAT

Lexium Integrated Drive

Manual for PLCopen Motion
Library for Beckhoff
controllers

V1.02, 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.

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

Table of contents

1	Introduction	4
1.1	Scope of supply	4
2	Integration of the library	5
2.1	Preparation	5
2.2	Creating the application.....	6
2.3	Integration of the drives into the TwinCAT System Manager	9
2.4	Linking the process data and the data of the application	14
2.4.1	Creating the process image in the System Manager	14
2.4.2	Linking the Tx variables from the drive to the master to the inputs of the controller.....	17
2.4.3	Linking the Rx variables from the master to the drive to the outputs of the controller.....	19
2.4.4	Linking the status variable.....	21
2.5	Creating mappings between application and System Manager	24
2.5.1	Activating the configuration and starting the controller	29
2.6	Troubleshooting.....	31
2.7	Addressing the drives in TwinCAT	33
2.8	Addressing the drives with Lexium CT	34
3	Function blocks	36
3.1	Function block names	36
3.2	Signal diagrams for inputs and outputs	36
3.2.1	Control inputs	36
3.2.2	Control outputs	37
3.2.3	Signal diagrams for function blocks with Execute control input.....	38
3.2.4	Signal diagram for function blocks with control input Enable	39
3.3	Description of the function blocks	40
3.4	Usage of the function blocks	41
3.4.1	Initialization.....	41
3.4.2	Jog.....	41
3.4.3	Homing	41
3.4.4	Profile Position operating mode	42
3.4.5	Profile Velocity operating mode	42
3.4.6	Stopping	42
3.4.7	Fast position capture.....	43
3.4.8	Reading parameters.....	43
3.4.9	Writing parameters.....	44
3.4.10	Inputs/outputs.....	44
3.4.11	Error handling.....	44
3.5	Detailed description of function blocks.....	45
3.5.1	MC_Power_ETC_ILX.....	45

3.5.2	MC_Jog_ETC_ILX.....	46
3.5.3	MC_Home_ETC_ILX.....	48
3.5.4	MC_SetPosition_ETC_ILX.....	51
3.5.5	MC_MoveAbsolute_ETC_ILX.....	52
3.5.6	MC_MoveRelative_ETC_ILX.....	54
3.5.7	MC_MoveAdditive_ETC_ILX.....	56
3.5.8	MC_MoveVelocity_ETC_ILX.....	57
3.5.9	MC_Stop_ETC_ILX.....	59
3.5.10	MC_TouchProbe_ETC_ILX.....	61
3.5.11	MC_AbortTrigger_ETC_ILX.....	63
3.5.12	MC_ReadActualPosition_ETC_ILX.....	65
3.5.13	MC_ReadActualVelocity_ETC_ILX.....	66
3.5.14	MC_ReadStatus_ETC_ILX.....	67
3.5.15	MC_ReadParameter_ETC_ILX.....	70
3.5.16	GetVersion_ETC_ILX.....	72
3.5.17	ReadActualPositionInc_ETC_ILX.....	73
3.5.18	ReadRefPosition_ETC_ILX.....	74
3.5.19	ReadRefPositionInc_ETC_ILX.....	75
3.5.20	ReadRefVelocity_ETC_ILX.....	77
3.5.21	MC_WriteParameter_ETC_ILX.....	78
3.5.22	SetDriveRamp_ETC_ILX.....	81
3.5.23	StoreParameters_ETC_ILX.....	82
3.5.24	ResetParameters_ETC_ILX.....	84
3.5.25	MC_ReadDigitalInput_ETC_ILX.....	85
3.5.26	MC_ReadDigitalOutput_ETC_ILX.....	87
3.5.27	MC_Reset_ETC_ILX.....	89
3.5.28	MC_ReadAxisError_ETC_ILX.....	90
4	Appendix.....	93
4.1	Error numbers.....	93

Information on this edition

Version 0.0/1

Manual created

Version 0.1/0

Addressing in TwinCAT and EtherCAT added

Version 1.0/0

Cover sheet updated, corrections included after review

Version 1.0/1

Translated Version

Version 1.0/2

Updated parameter information in FB SetDriveRamp

1 Introduction

Programs according to the IEC 61131-3 standard are created with the function blocks as per PLCopen Motion specification in the TwinCAT programming environment provided by Beckhoff. The function blocks control one or several Schneider Electric Lexium ILx2 drives via the EtherCAT fieldbus. The motion 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 Beckhoff TwinCAT programming system.

The version number is a part of the library name; `_Vxxxx_` stands for the corresponding version number, for example V1001.

- The library SE_Motion_ETC_ILX_Vxxxx.lib
- The library SE_CIA405_ETC_ILX_Vxxxx.lib
- The electronic datasheet file Schneider Electric.xml
- This manual
- A sample program

2 Integration of the library

The library was created for the Beckhoff TwinCAT programming software. The following steps are required to make the Lexium ILx2 drives known to the TwinCAT 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 the TwinCAT software.

- Copy the electronic datasheet file Schneider Electric.xml to the subfolder ...\\TwinCAT\\Io\\EtherCAT of the TwinCAT software as shown below:

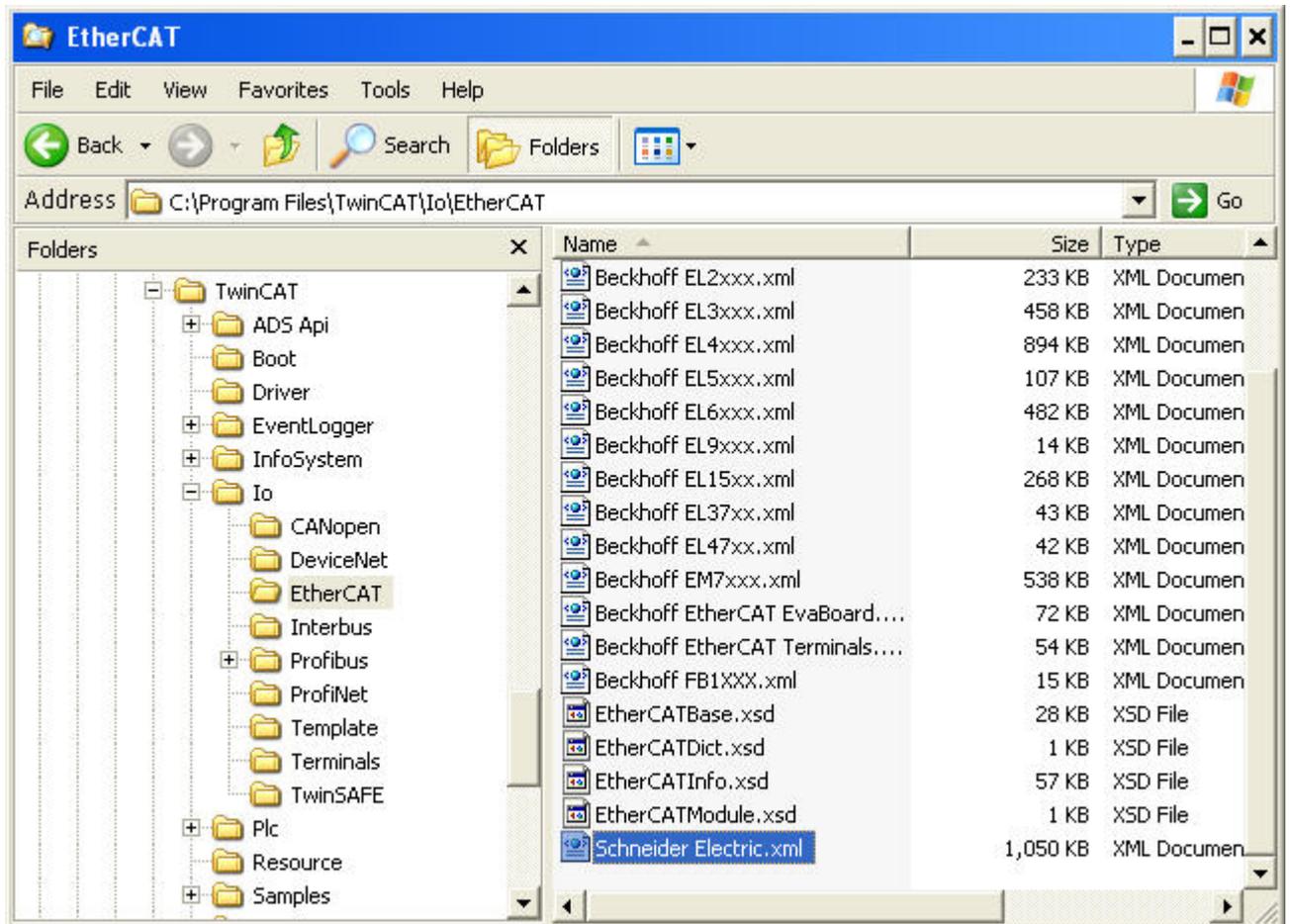


Bild 1) Folder for Schneider Electric.xml

- Copy the two libraries SE_Motion_ETC_ILX_Vxxxx.lib and SE_CIA405_ETC_ILX_Vxxxx.lib provided by Schneider Electric to the TwinCAT library folder ...\\TwinCAT\\Plc\\Lib as shown below:

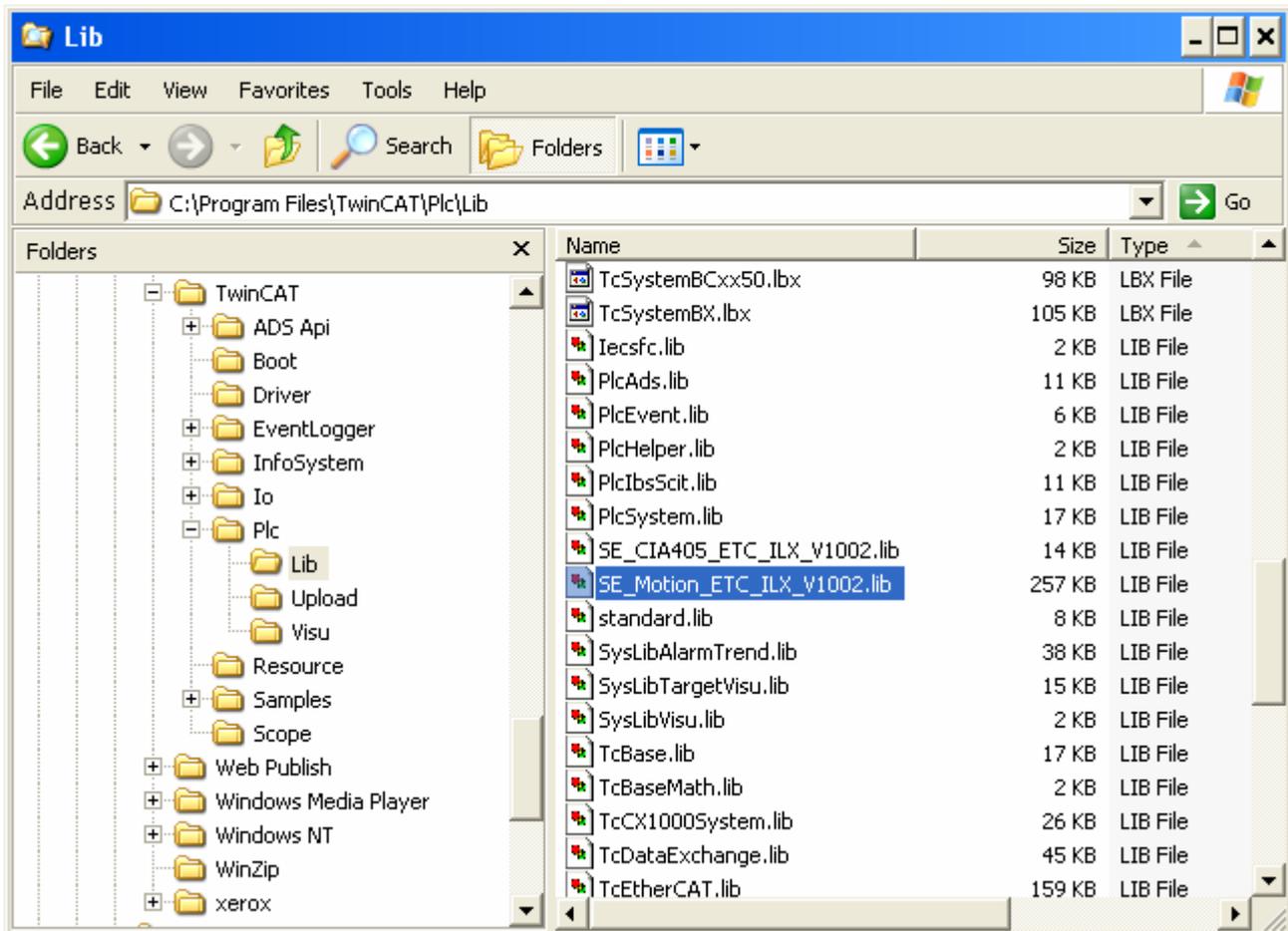


Bild 2) Folder for Schneider Electric libraries

2.2 Creating the application

Integration of the libraries into the application

- Start the TwinCAT PLC Control software
- Create a new project or integrate the Lexium ILx2 drives into an existing project.
- Display the Library Manager and integrate the standard libraries below. They are part of TwinCAT.
 - STANDARD.LIB
 - TcBase.lib
 - TcSystem.lib
 - TcUtilities.lib
 - TcEtherCAT.lib
- Now integrate the two Schneider Electric libraries.

- SE_Motion_ETC_ILX_Vxxxx.lib
- SE_CIA405_ETC_ILX_Vxxxx.lib
- Your Library Manager must now display at least the following libraries:

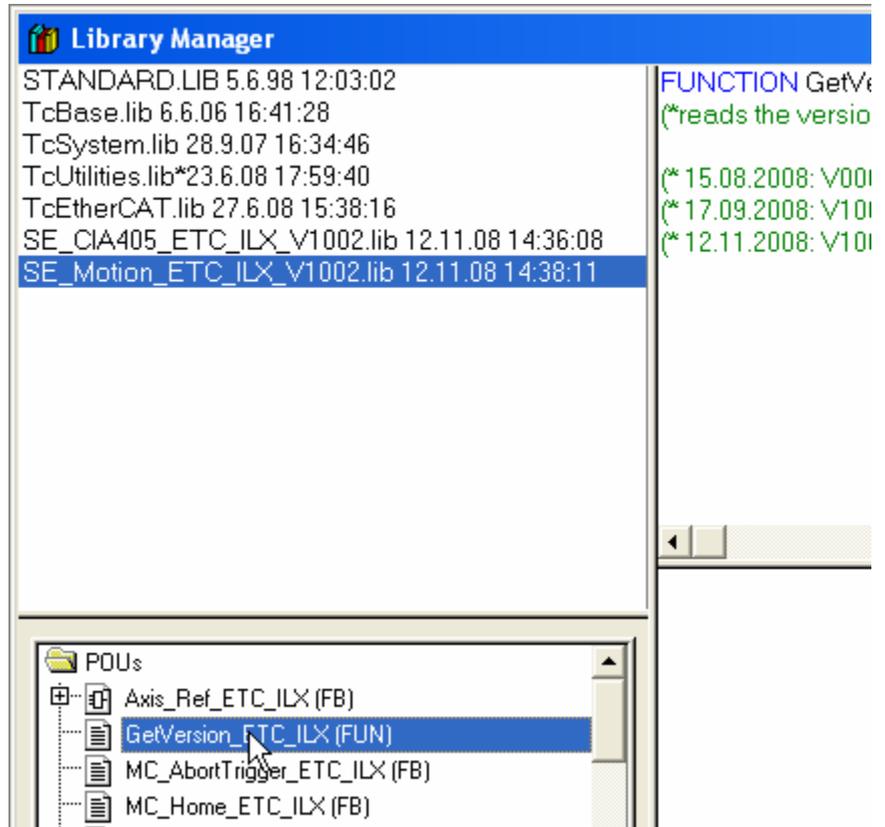


Bild 3) Required libraries in the TwinCAT Library Manager

Creating the global data structure per drive

- Now create a data structure in the global variables for each Lexium ILx2 drive you want to operate via EtherCAT. The data structure must be of the type Axis_Ref_ETC_ILX.



Bild 4) Data structure for each drive

The configuration in TwinCAT PLC Control is now complete and you

can start creating the application. It is recommended to perform the following configuration steps directly in the TwinCAT System Manager.

Preparing the project for integration into the System Manager

- Build your project by selecting the "Rebuild all" command from the "Project" menu.

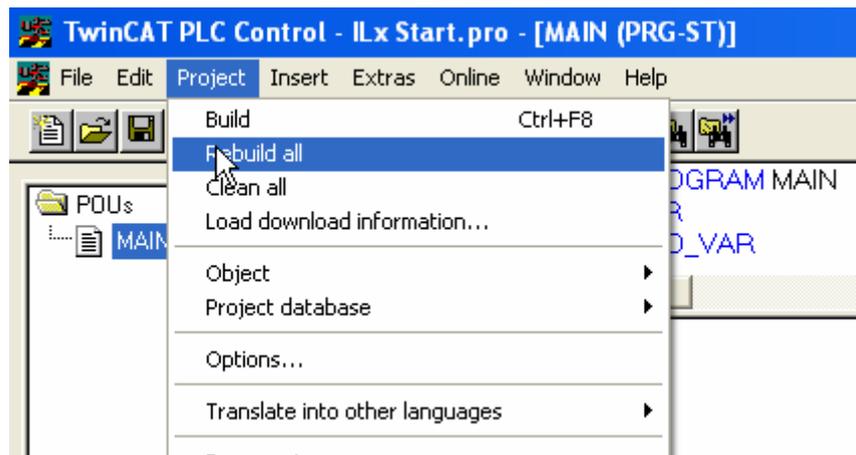


Bild 5) Project - Rebuild all

- The TwinCAT message window displays message relating to the data structured created. These messages are a result of the fact that you have not yet created the required links in the TwinCAT System manager. This is done in a later step so ignore the messages for the time being.

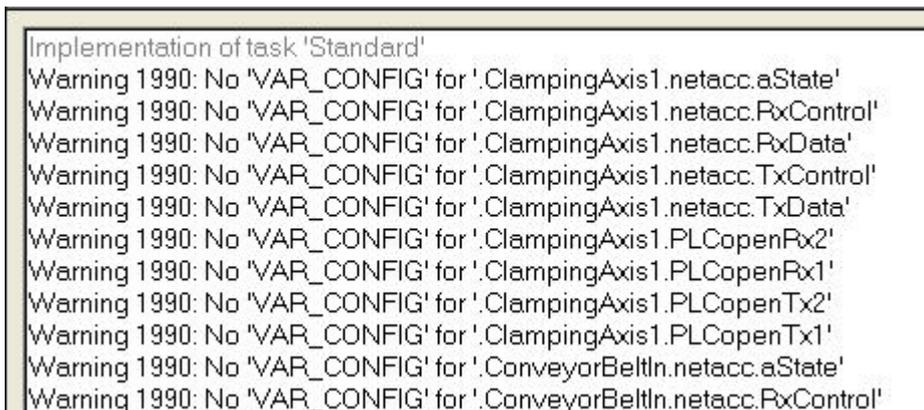


Bild 6) Warnings after building the project

- If the message window displays error, you must correct them so that the application can be built without errors.

During the build process, TwinCAT creates a file with the project name and the extension .tpy in the folder in which the project is stored. This .tpy file is required by the TwinCAT System Manager.

2.3 Integration of the drives into the TwinCAT System Manager

The I/O devices are managed in the TwinCAT System Manager. The Lexium ILx2 drives with EtherCAT interface are added in the System Manager.

I/O configuration

The I/O Devices node in the TwinCAT System Manager is used to manage the I/O devices. There are two ways of adding devices.

- Scanning the EtherCAT network
- Adding devices manually in the System Manager

Scanning the EtherCAT network

To scan the network, proceed as described below. Please note that the display of the I/O devices in the System Manager corresponds to their sequence on the EtherCAT network. The device closest to the controller in the network is displayed at the top of the list. Further addressing is not required as this results from the arrangement in the network.

For scanning, you must connect to the controller, also referred to as target system.

- Make sure the controller is in configuration mode. To do so, choose "Set/Reset TwinCAT to Config Mode..." from the "Actions" menu.

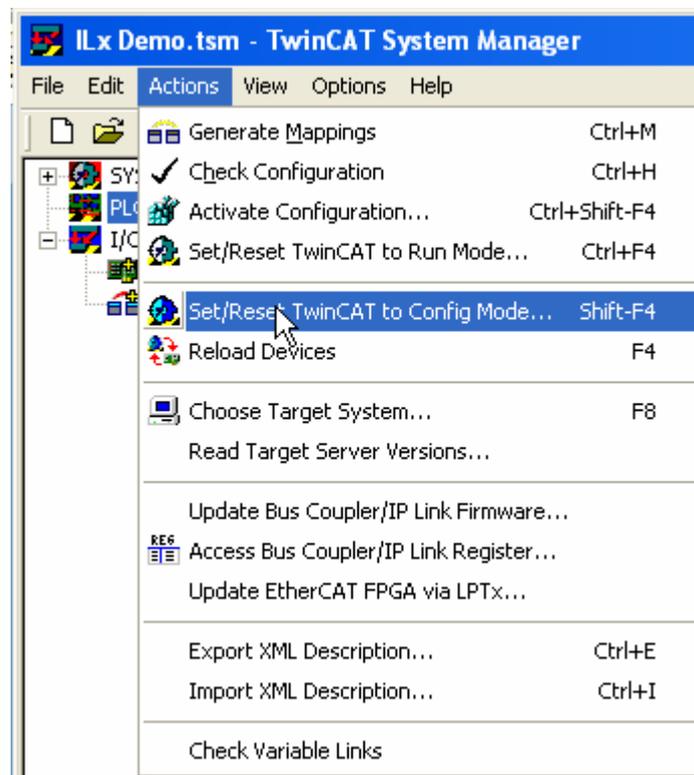


Bild 7) Starting TwinCAT in Config Mode

- In the left pane of the System Manager, open the "I/O Configuration" node and select the "I/O Devices" entry.

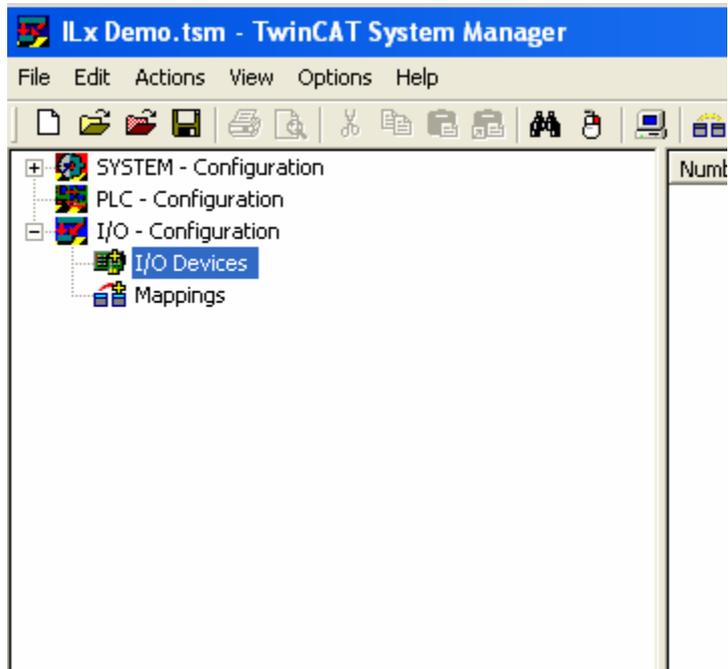


Bild 8) Selecting I/O devices in the System Manager

- Right-click the entry "I/O Devices" and choose "Scan Devices..." from the menu.

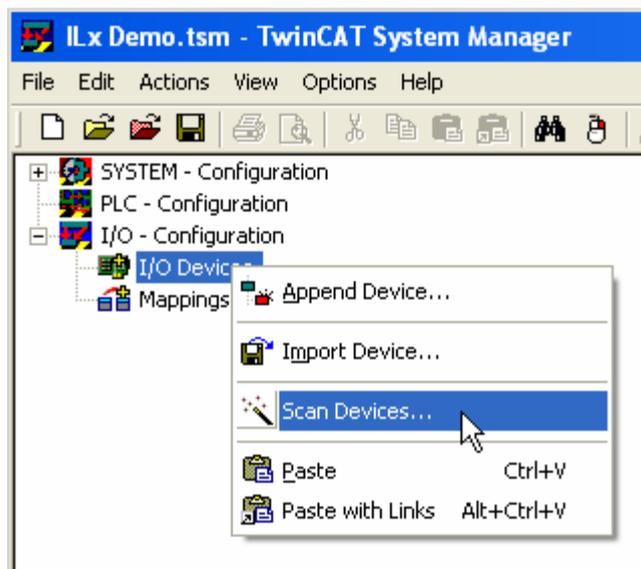


Bild 9) Scanning the network for devices

- If TwinCAT displays a message, confirm it by clicking "OK".

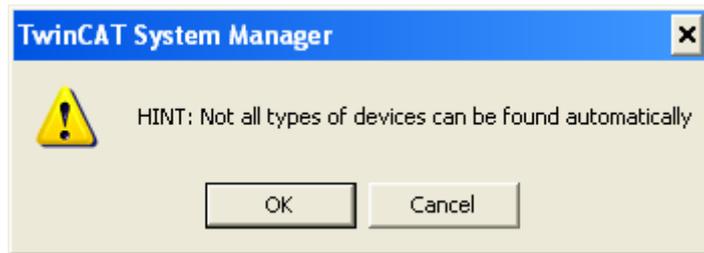


Bild 10) Confirming the message

- The scanning process starts. At the end of the process, the software displays an overview of all I/O devices found. If there are several devices for selection, select the desired EtherCAT device. Also if several Ethernet interfaces are displayed.
- Then close the selection with "OK".

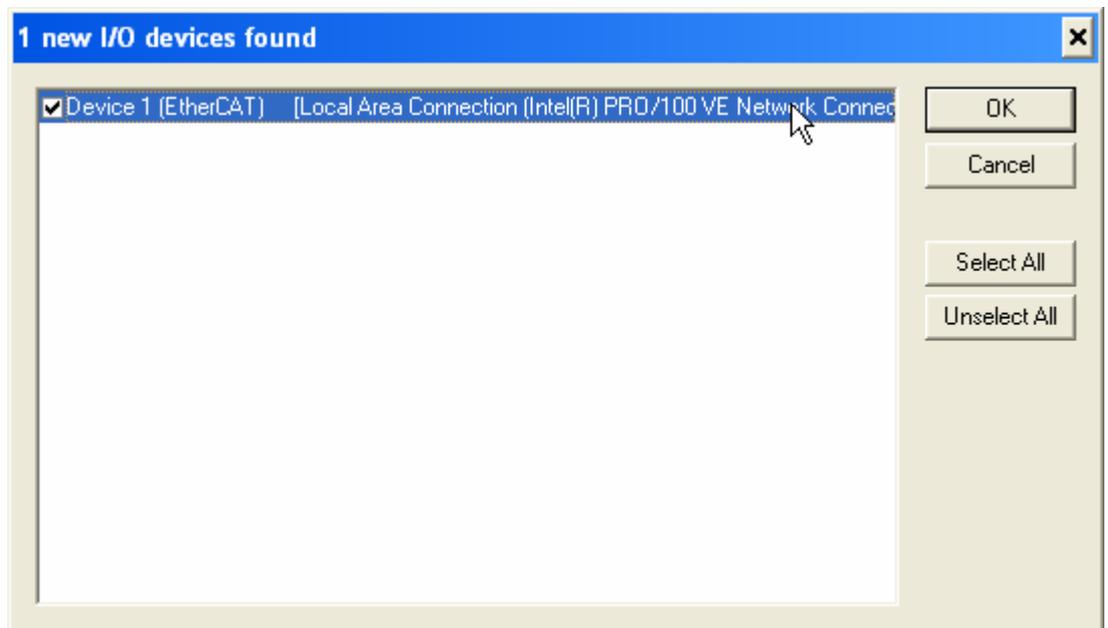


Bild 11) I/O devices found

- The system displays a dialog box asking you whether you want to scan for boxes. Click the "Yes" button.

The System Manager uses the term boxes for EtherCAT network devices.



Bild 12) Scanning for new boxes

- Depending on your configuration at the EtherCAT network, the system displays message windows on NC axes. If the questions relate to the Lexium ILx2 drives, click the "No" button.
- Click the "No" button if the system prompts you whether or not to perform a Free Run.



Bild 13) Activating Free Run

This completes the configuration which is now available in the system. The following screenshot shows an example.

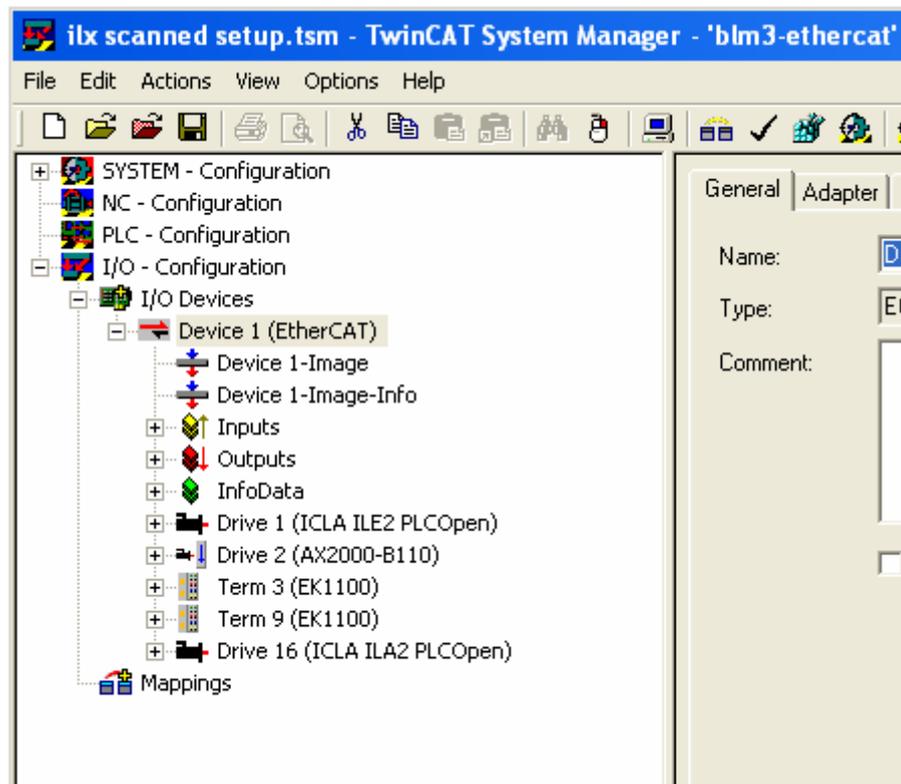


Bild 14) EtherCAT configuration

Adding products manually in the System Manager

- Add the Lexium ILx2 drives manually in the System Manager. To do so, select the EtherCAT master, right-click it and choose "Append Box..." from the menu.

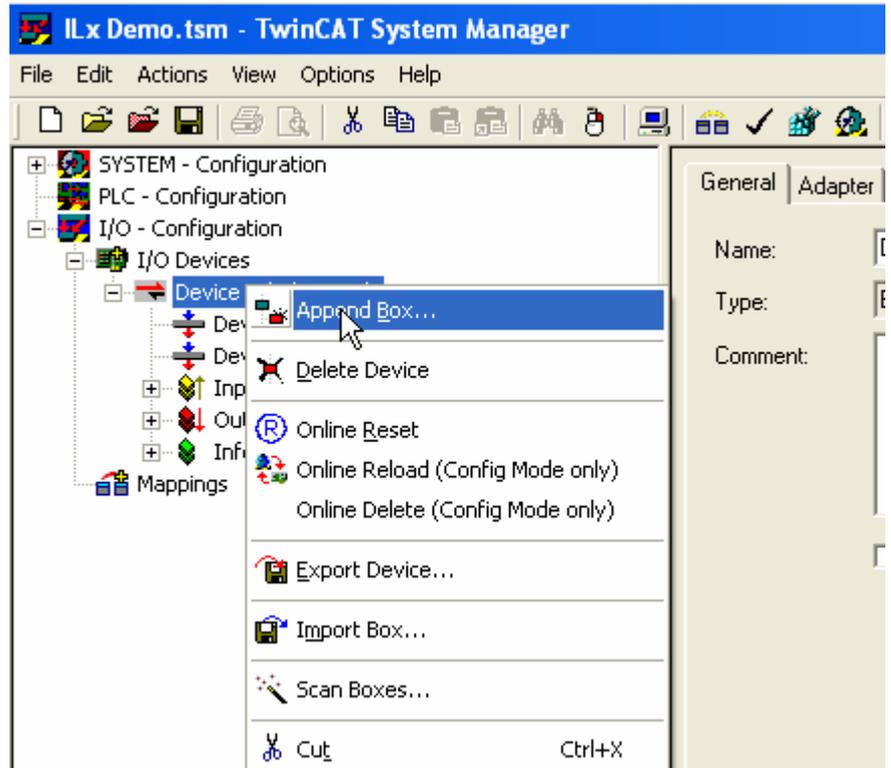


Bild 15) Adding a box to the EtherCAT master

- The system opens a window; select the appropriate drive type below Schneider Electric Motion. Choose and entry with the ending PLCopen. Confirm by clicking "OK"

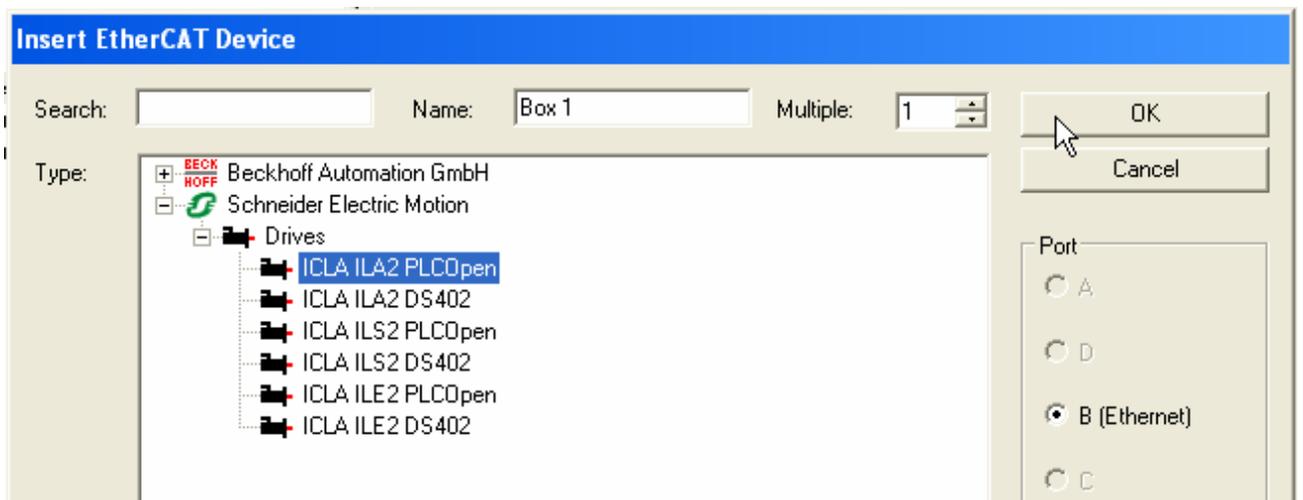


Bild 16) Adding an EtherCAT drive

- Please note that the sequence of the devices in the System Manager corresponds to the sequence of the devices in the EtherCAT network.
- Repeat the previous step until you have added all drives to the EtherCAT master.

The System Manager now displays the process data of the drive. In

the next steps, this process data will be linked to the data structures of the application.

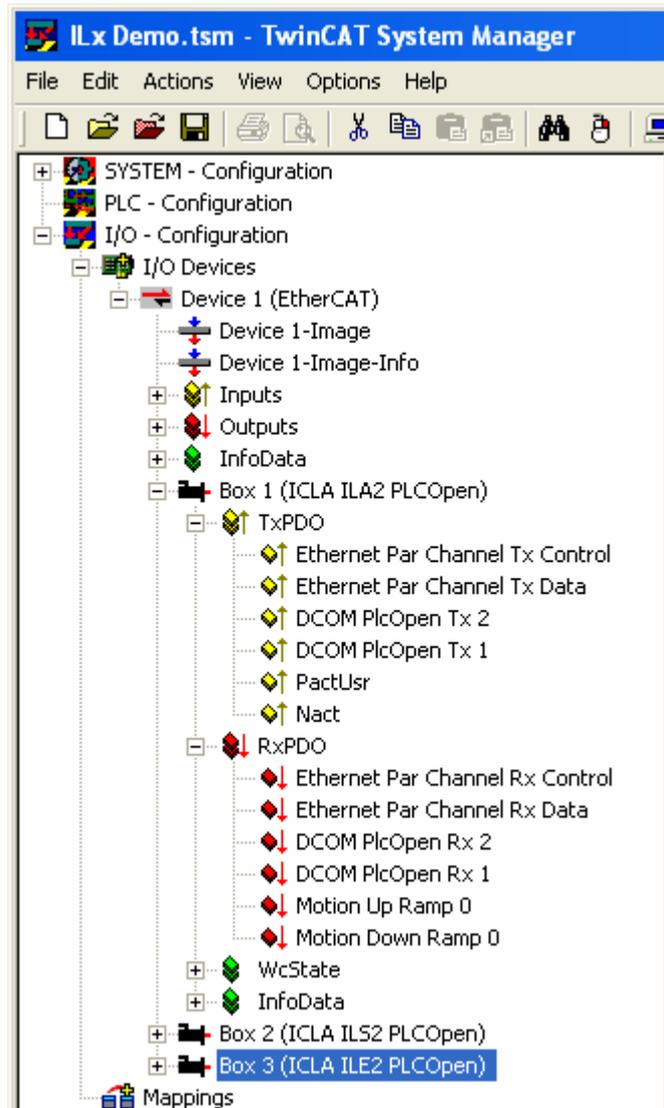


Bild 17) Drive with its process data

2.4 Linking the process data and the data of the application

The TwinCAT System Manager is used to link the data structures of the application and the process data of the Lexium ILx2 drives. If you have not yet performed the steps described in the previous chapters, do so now.

2.4.1 Creating the process image in the System Manager

- In the TwinCAT System Manager, select the PLC configuration and right-click it.
- Select the menu item "Append PLC project...".

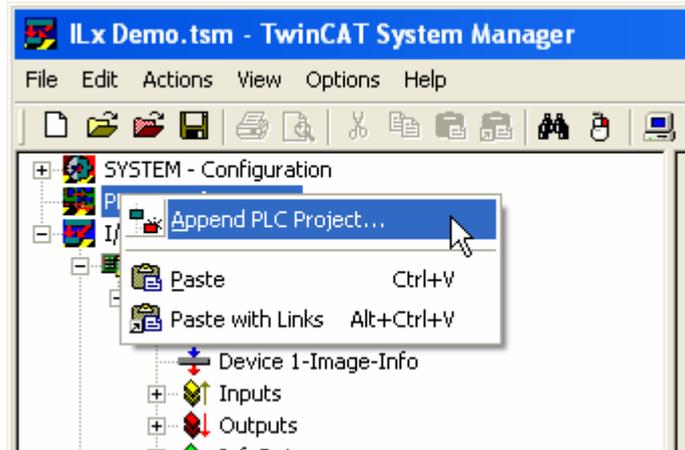


Bild 18) Adding a PLC project in the System Manager

- Open the folder in which the .tpy file was created and select this file.

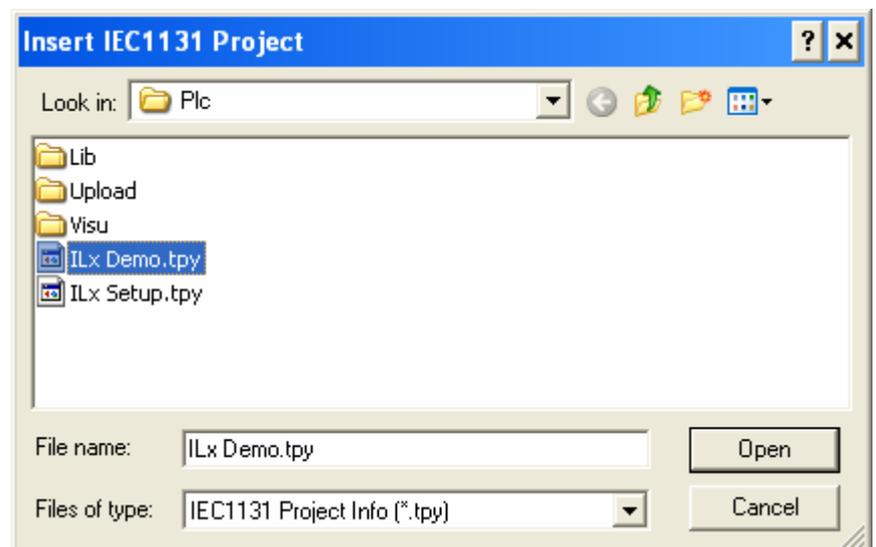


Bild 19) Selecting a project via the .tpy file

The System Manager now displays the project below the PLC. This project shows the input data and output data that need to be linked to the process image.

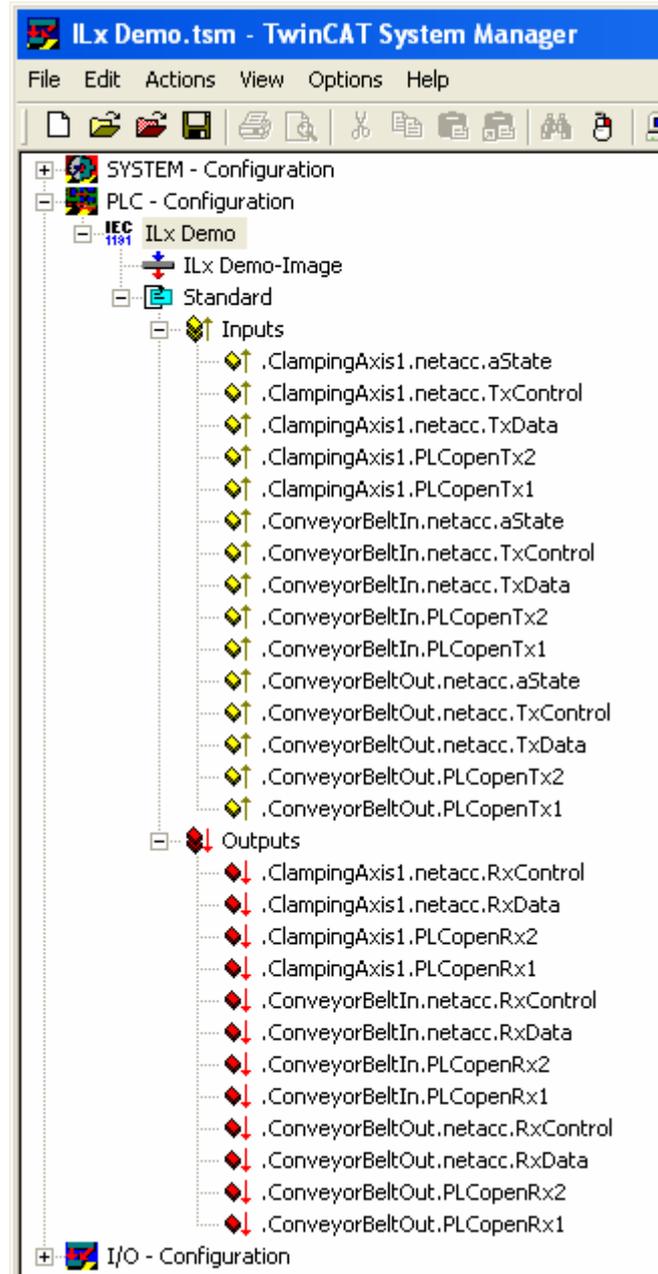


Bild 20) Inputs and outputs of the application in the System Manager

- Select the entry *ApplicationName-Image* (*ApplicationName* being the name of your application). The application name in the example is "ILX Demo". The right part of the window then displays all input and output variables of the application.

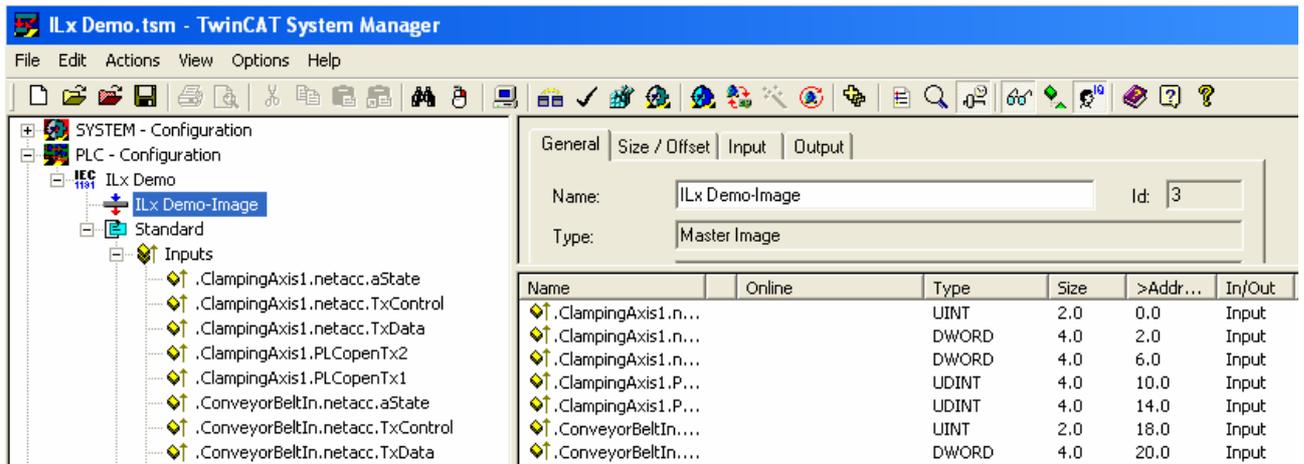


Bild 21) Selecting the process image in the System Manager

2.4.2 Linking the Tx variables from the drive to the master to the inputs of the controller

- Link the inputs to the Tx Variables. To do so, select the following entries:
 - *Name.netacc.TxControl*
 - *Name.netacc.TxData*
 - *Name.PLCopenTx1*
 - *Name.PLCopenTx2*

Name stands for the variable name of your application. In the example, *Name* is ClampingAxis1.

- Right-click the selected entries. Then choose the menu item "Change Multi Link...".

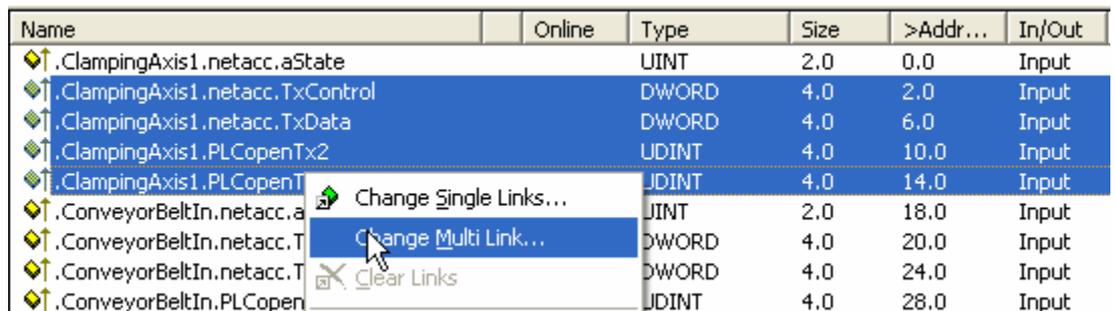


Bild 22) Changing the multi link for inputs

- The system displays a window; in the "Show Variables" box, select "Used and unused".
- In the "Show Variable Types" box, select "All Types".
- In the "Offsets" box, select "Continuous".

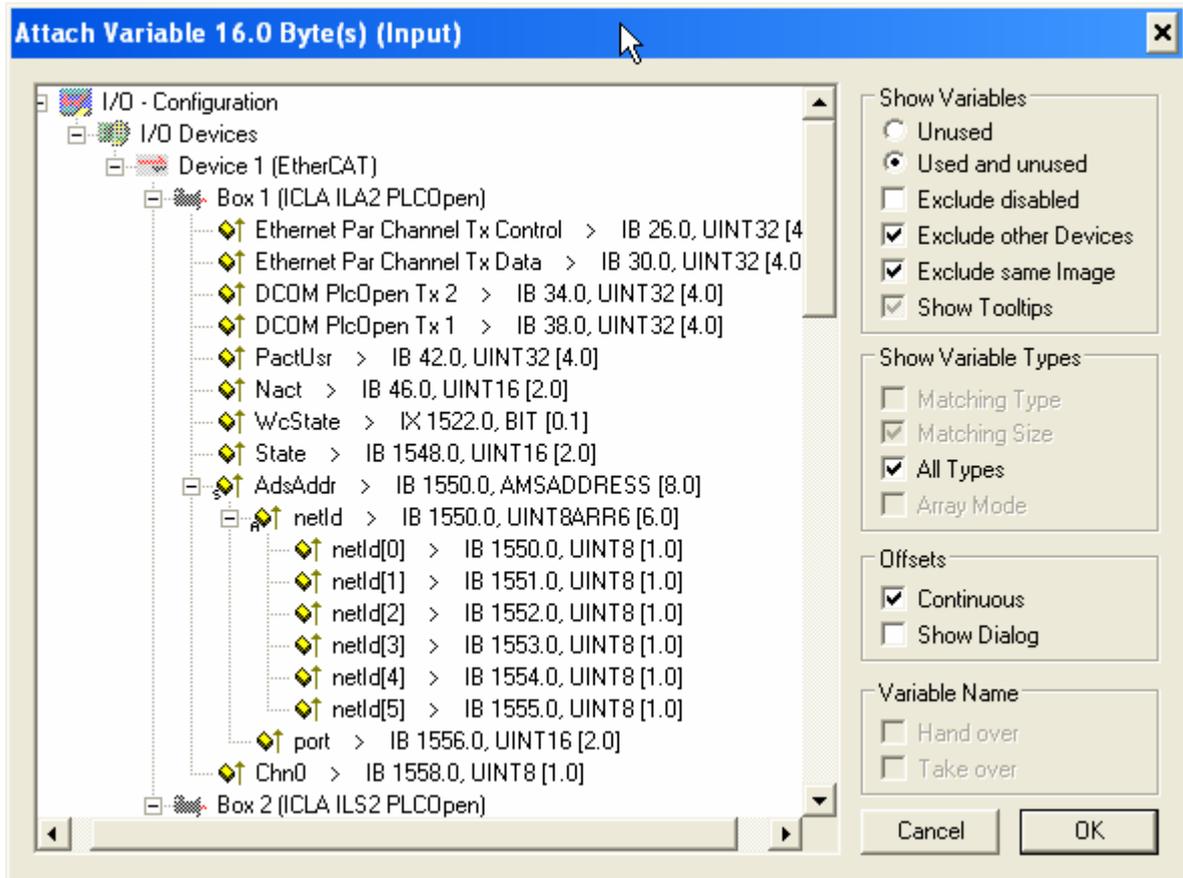


Bild 23) Settings for linking variables

- Select the first four entries for the appropriate drive and click "OK". The links must be made as follows:

Name.netacc.TxControl to Ethernet Par Channel Tx Control

Name.netacc.TxData to Ethernet Par Channel Tx Data

Name. PLCopenTx1 to DCOM PlcOpen Tx1

Name. PLCopenTx2 to DCOM PlcOpen Tx2

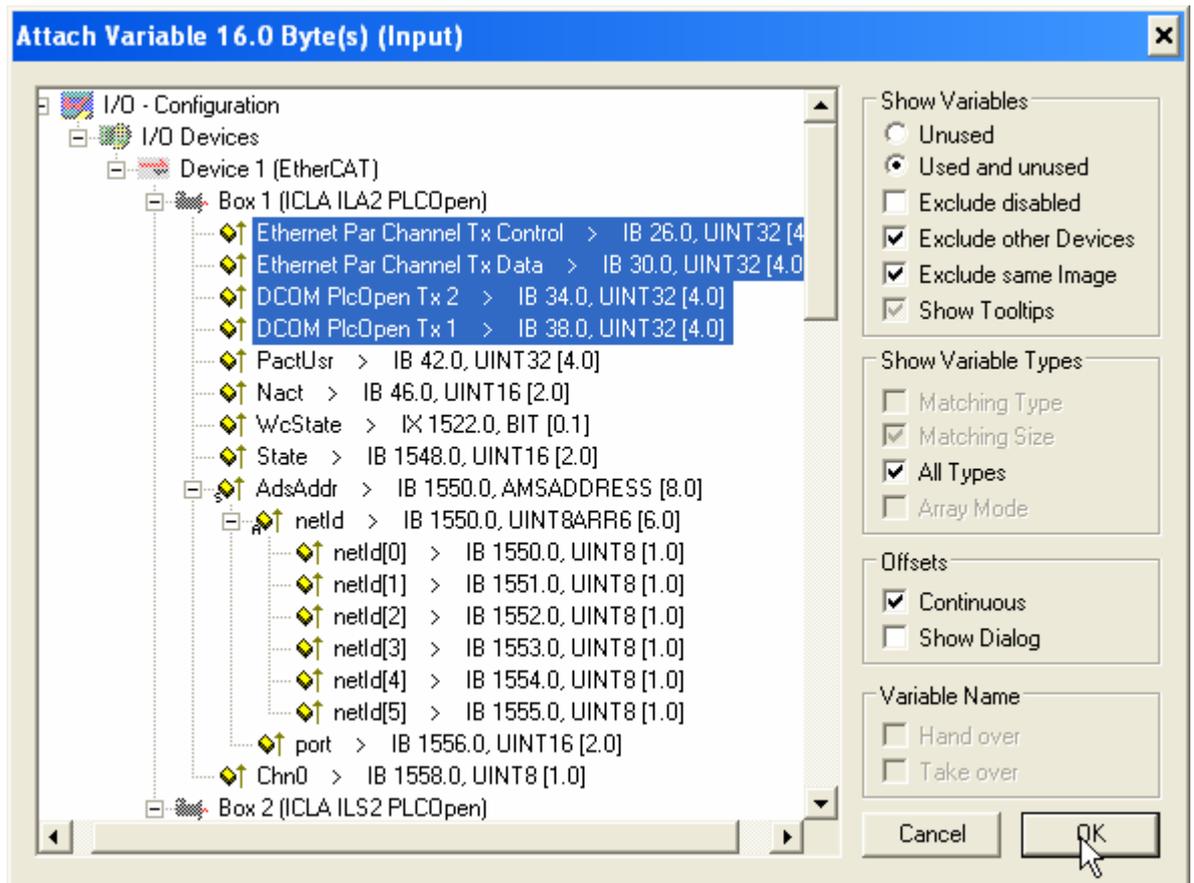


Bild 24) Selection of the input variables to be linked

- The links are now shown in the overview. Repeat the previous steps for all additional drives.

2.4.3 Linking the Rx variables from the master to the drive to the outputs of the controller

- Link the outputs to the Rx Variables. To do so, select the following entries:
 - *Name.netacc.RxControl*
 - *Name.netacc.RxData*
 - *Name. PLCopenRx1*
 - *Name. PLCopenRx2*

Name stands for the variable name of your application. In the example, *Name* is *ClampingAxis1*.
- Right-click the selected entries. Then choose the menu item "Change Multi Link...".

.ConveyorBeltOut.netacc.RxData	^	DWORD	4.0	72.0	Input	0	LC1E
.ConveyorBeltOut.PLCopenTx2	X	UDINT	4.0	46.0	Input	0	DCC
.ConveyorBeltOut.PLCopenTx1	X	UDINT	4.0	50.0	Input	0	DCC
.ClampingAxis1.netacc.RxControl		DWORD	4.0	0.0	Output	0	
.ClampingAxis1.netacc.RxData		DWORD	4.0	4.0	Output	0	
.ClampingAxis1.PLCopenRx2		UDINT	4.0	8.0	Output	0	
.ClampingAxis1.PLCopenRx1		UDINT	4.0	12.0	Output	0	
.ConveyorBeltIn.netacc.RxControl				16.0	Output	0	
.ConveyorBeltIn.netacc.RxData				20.0	Output	0	
.ConveyorBeltIn.PLCopenRx2				24.0	Output	0	
.ConveyorBeltIn.PLCopenRx1				28.0	Output	0	
.ConveyorBeltOut.netacc.RxControl				32.0	Output	0	
.ConveyorBeltOut.netacc.RxData				36.0	Output	0	
.ConveyorBeltOut.PLCopenRx2				40.0	Output	0	
.ConveyorBeltOut.PLCopenRx1				44.0	Output	0	

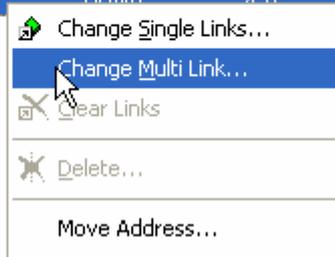


Bild 25) Changing the multi links for the outputs

- The system displays a window; in the "Show Variables" box, select "Used and unused".
- In the "Show Variable Types" box, select "All Types".
- In the "Offsets" box, select "Continuous".

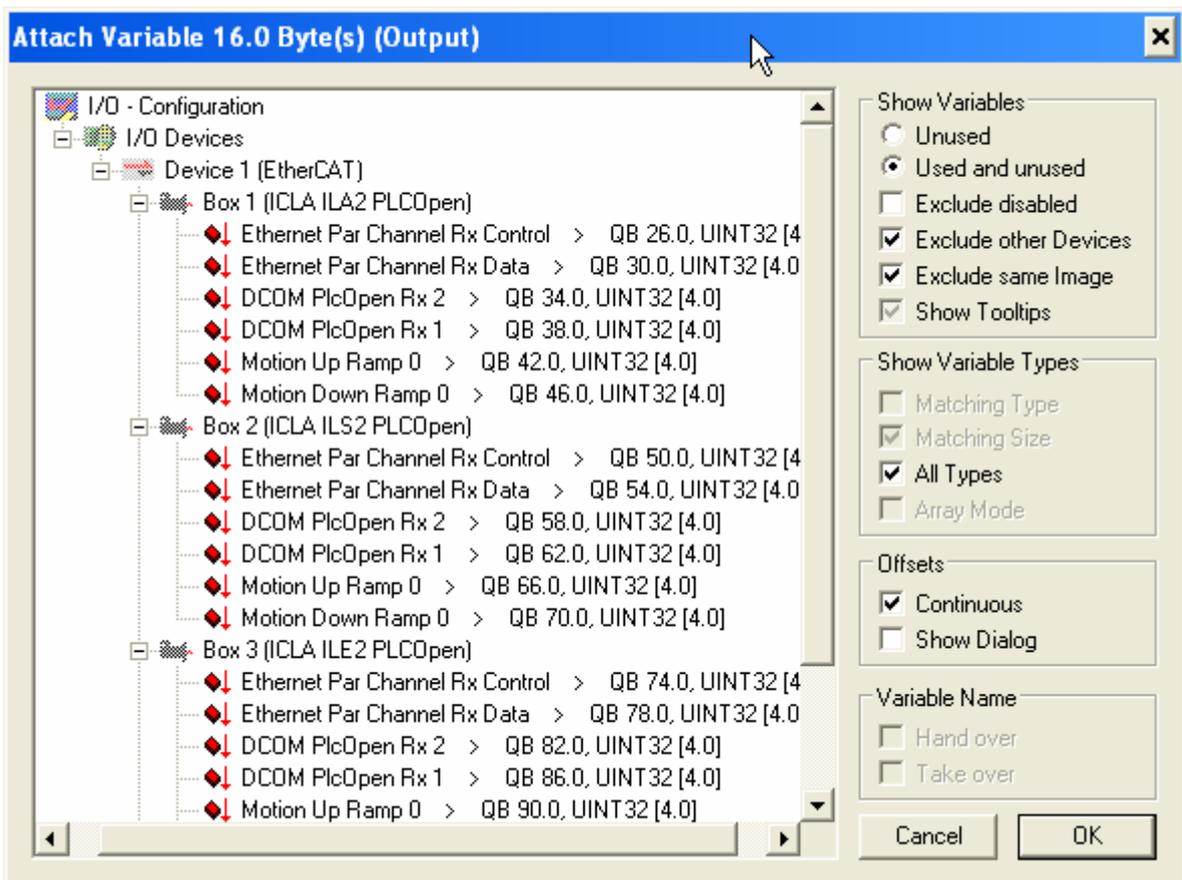


Bild 26) Settings for linking variables

- Select the first 4 entries for the appropriate drive and click "OK".

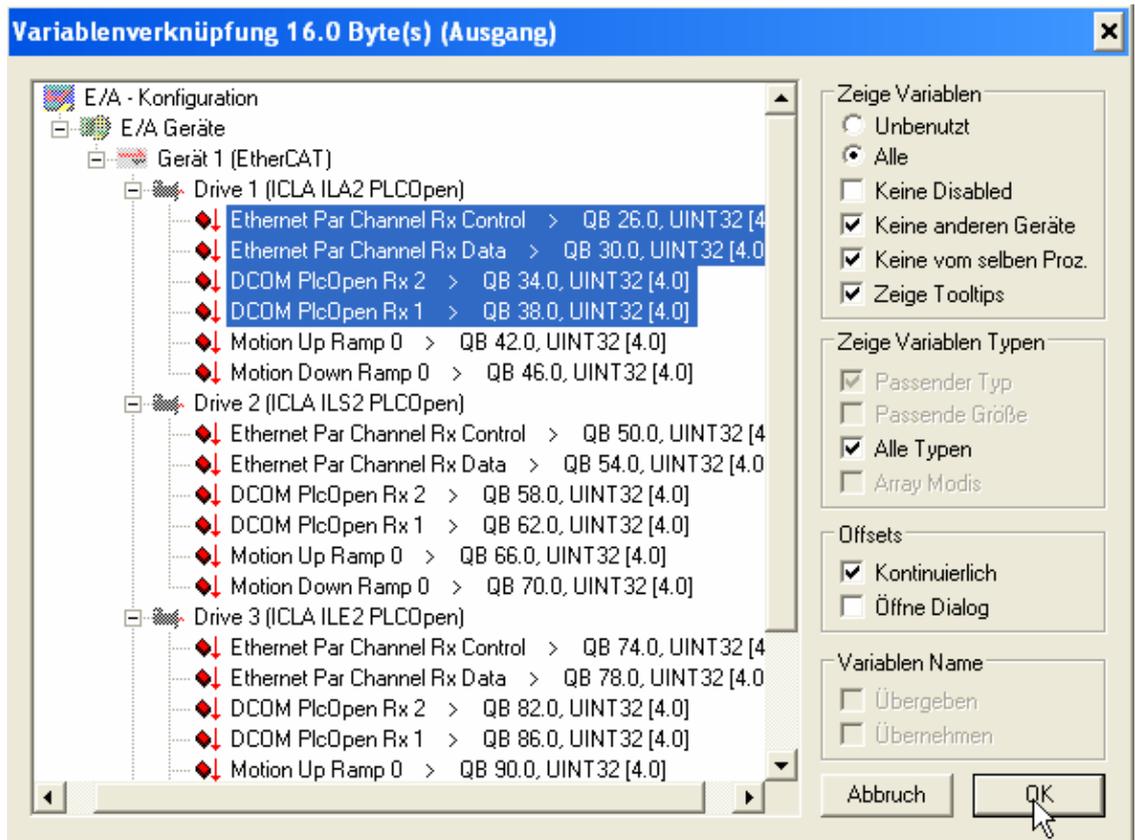


Bild 27) Selection of the output variables to be linked

- The links are now shown in the overview. The links must be made as follows:

Name.netacc.RxControl to Ethernet Par Channel Rx Control

Name.netacc.RxData to Ethernet Par Channel Rx Data

Name. PLCopenRx1 to DCOM PlcOpen Rx1

Name. PLCopenRx2 to DCOM PlcOpen Rx2

- Repeat the previous steps for all additional drives.

2.4.4 Linking the status variable

The status variable *IFx.netacc.aState* is used for connection monitoring. This service is similar to CANopen Nodeguarding or Heartbeat.

- In the System Manager, select the drive, the category "InfoData" and the variable "State".

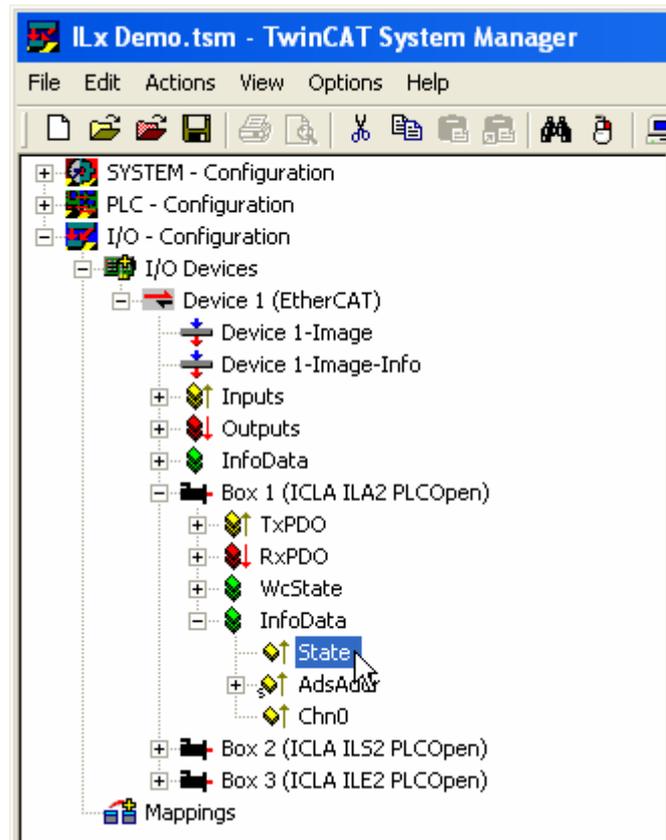


Bild 28) Selecting the variable "State"

- Right-click the entry and choose "Change Link..." from the menu.

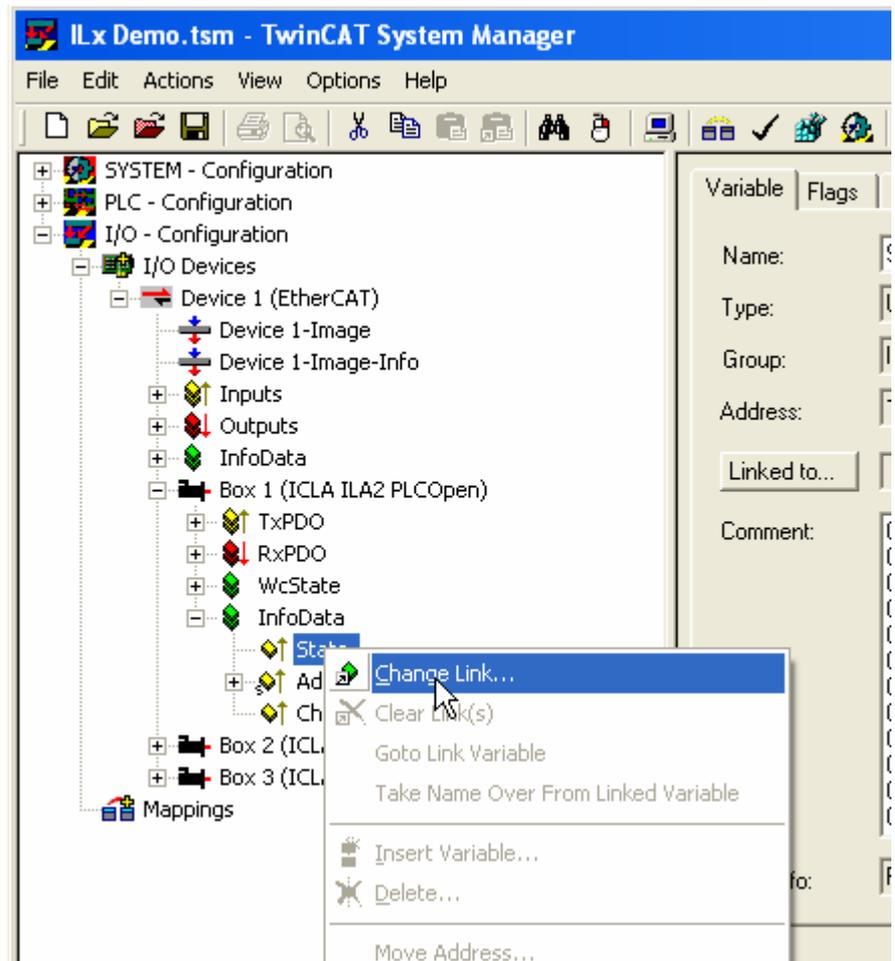


Bild 29) Changing the link of the "State" variable.

- Select the entry *Name.netacc.aState* (*Name* being the drive in your program). You can narrow down the selection by activating the option "Unused" in the "Show Variables" box.

The link is created as described below:

Name.netacc.aState to *InfoData.State*

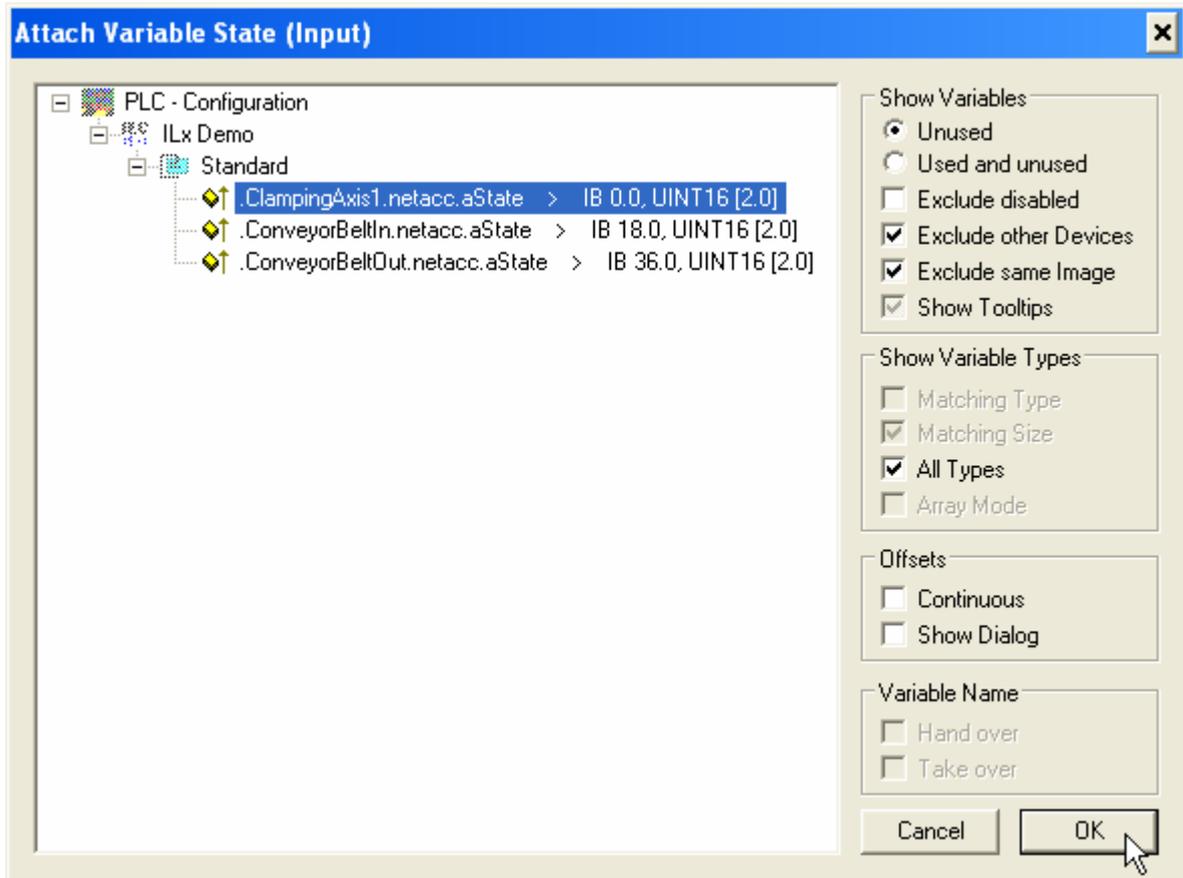


Bild 30) Selecting the link for the "State" variable

- Repeat the previous steps for all additional drives.

2.5 Creating mappings between application and System Manager

The configuration of the link has been described in the previous chapters. This link must now be tested and activated.

Generating a mapping

- In the "Actions" menu of the System Manager, select the menu item "Generate Mappings".

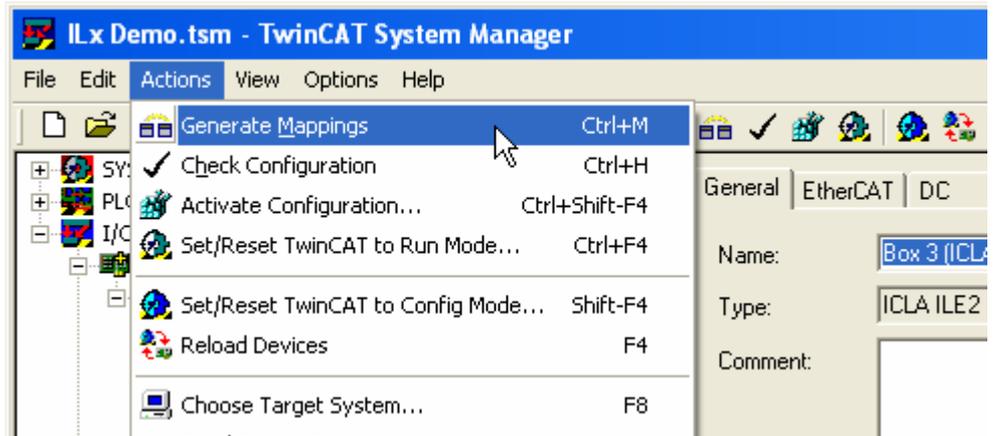


Bild 31) Generating a mapping

The mappings are shown in the left window of the System manager.

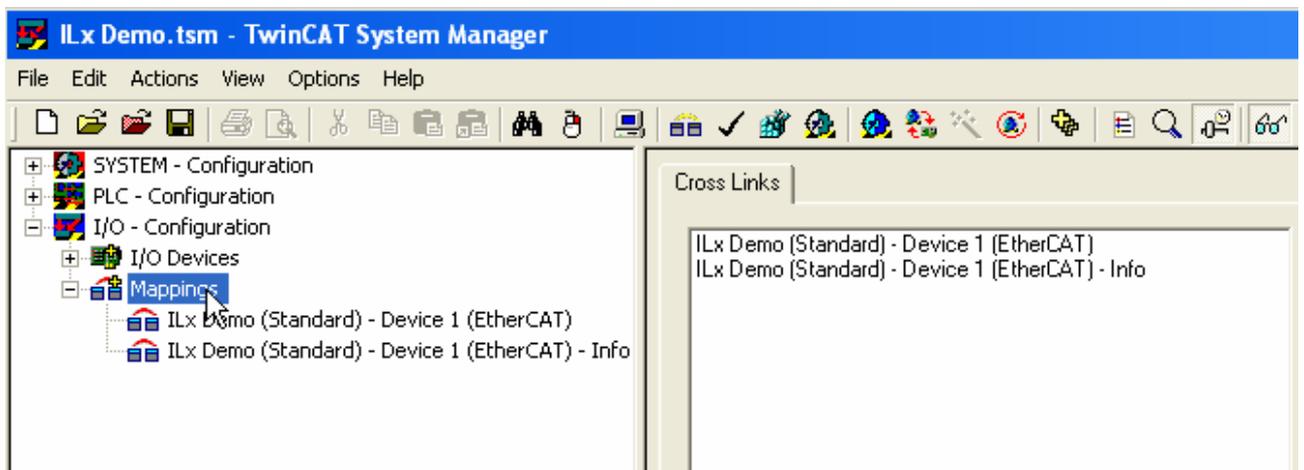


Bild 32) Mappings in the System manager

Checking the configuration

- Check the configuration. In the "Actions" menu of the System Manager, select the menu item "Check Configuration".

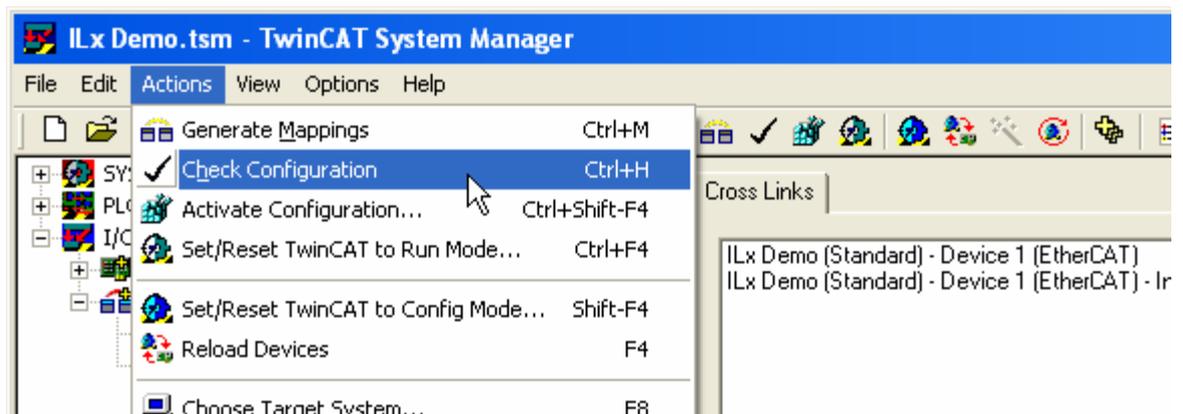


Bild 33) Checking the configuration

- If you get error messages as a result of the check, fix the errors and then repeat the steps described in this chapter.

Building an application with a new mapping

- Now switch from the TwinCAT System manager to the TwinCAT Control in which your application is loaded.
- From the "Project" menu, choose the menu item "Rebuild all".

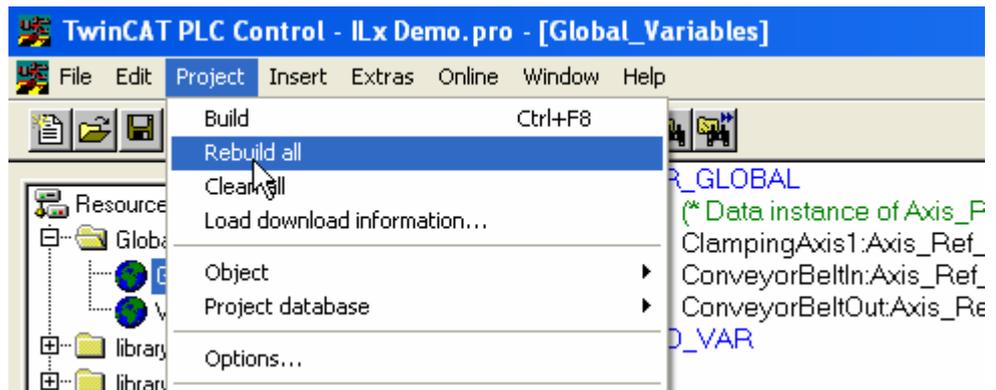


Bild 34) Rebuilding the application

If the links are correct, there are no warnings.

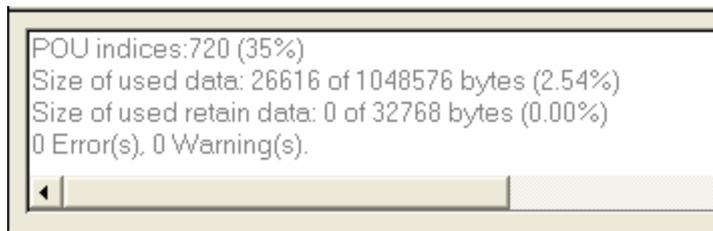


Bild 35) Build without warnings

At the same time, the configurations of the variables in the project are generated automatically. These can be found under "Global Variables", "TwinCAT Configuration".

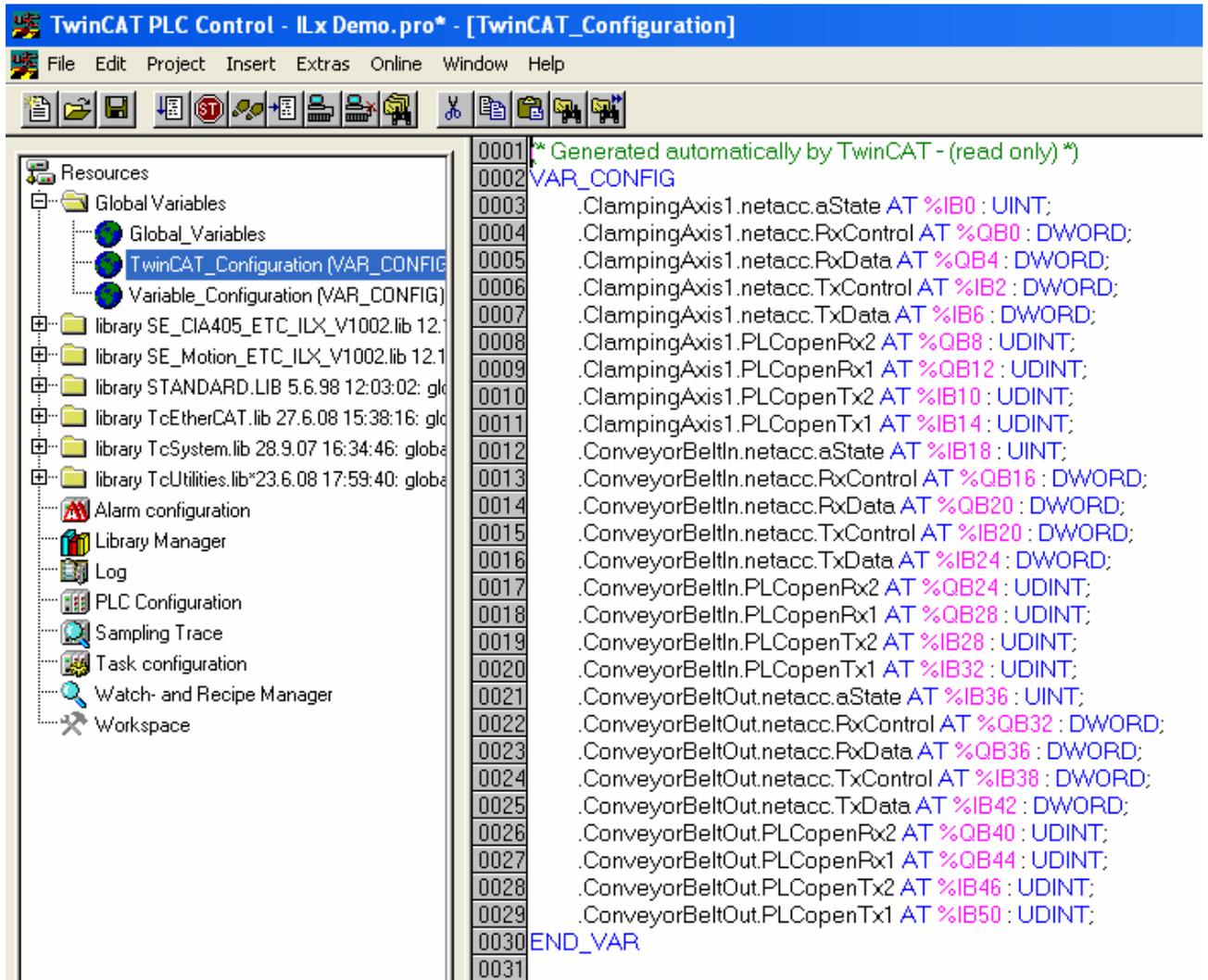


Bild 36) Configuration of the variables with mapping to process image

Updating the changed project in the System Manager

- Now switch back to the TwinCAT System Manager.
- From the "Options" menu, select the item "Check PLC Project Changes".

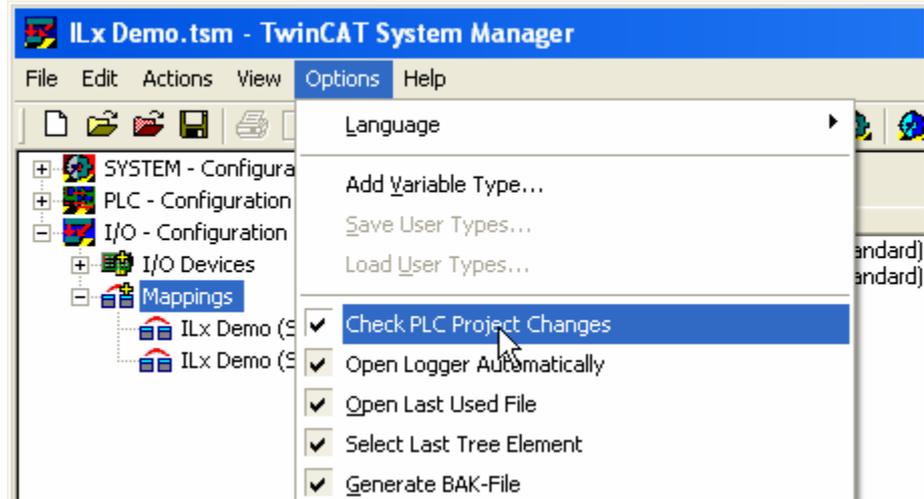


Bild 37) Option "Check PLC Project Changes"

- If the option "Check PLC Project Changes" is checked, the System Manager tries to re-read the IEC 61131 project.

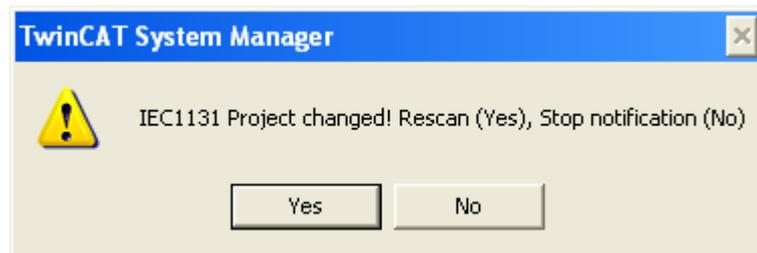


Bild 38) Rescanning the project in the System Manager

- Click "Yes" to confirm.
- If the System manager did not try to automatically rescan the project, select your application's entry in the left window.

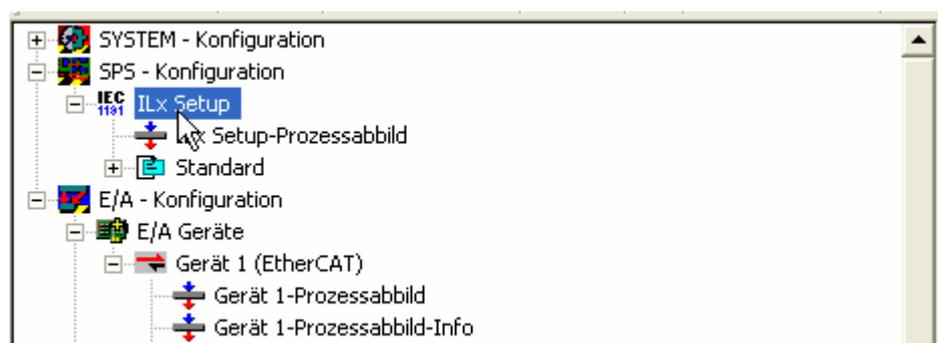


Bild 39) Selecting the application in the System Manager

- Then click the "Rescan" button on the right side pane.

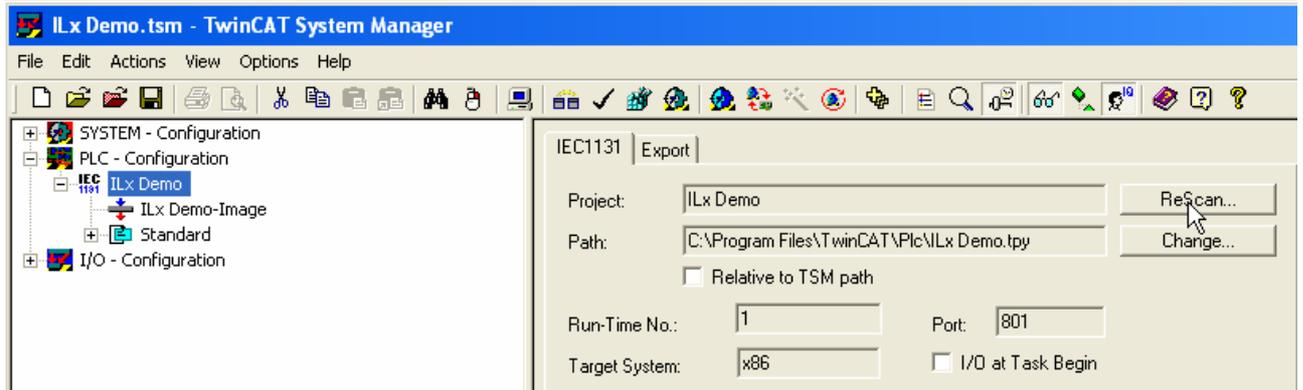


Bild 40) Rescanning the application

Updating the project again

- Now switch from the TwinCAT System manager to the TwinCAT Control in which your application is loaded.
- From the "Project" menu, choose the menu item "Rebuild all".

The automatically created entry TwinCAT_Configuration has been updated again. It now contains comments with the description of the links.

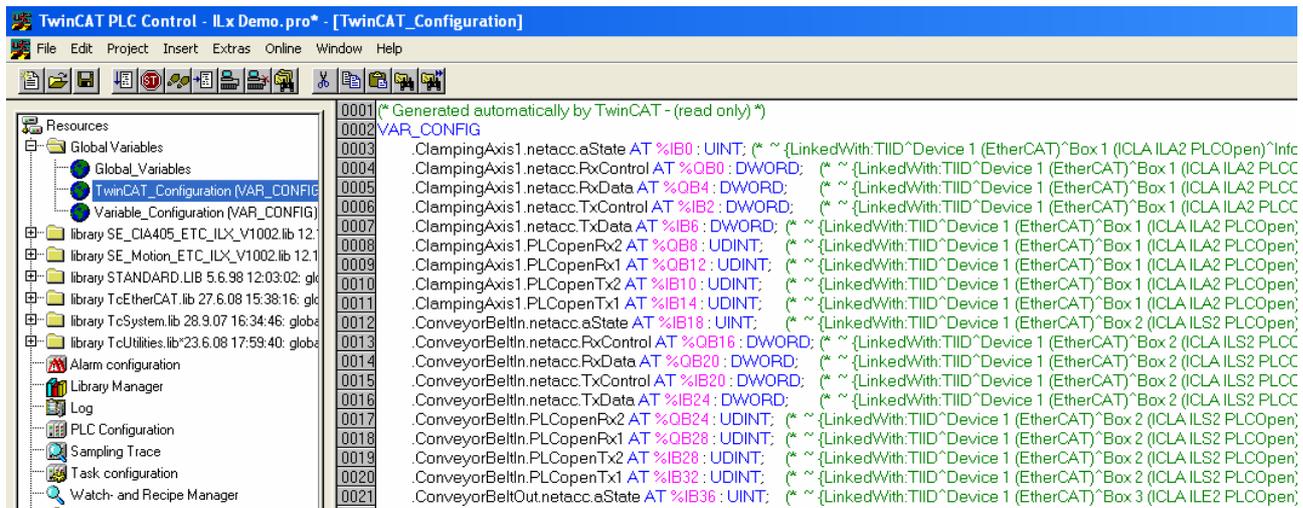


Bild 41) Updated configuration of the variables with mapping to process image

- Switch back to the System Manager. If the project is not updated automatically, update the project manually as described above.

2.5.1 Activating the configuration and starting the controller

The following steps transmit the configuration to the controller and start it.

DANGER

UNINTENDED CONSEQUENCES OF EQUIPMENT OPERATION

When the system is started, the drives are usually out of the operator's view and cannot be visually monitored.

- Only start the system if there are no persons in the hazardous area.

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

WARNING

UNEXPECTED MOVEMENT

Drives may perform unexpected movements because of incorrect wiring, incorrect settings, incorrect data or other errors. Interference (EMC) may cause unpredictable responses in the system.

- Carefully install the wiring in accordance with the EMC requirements.
- Switch off the voltage at the inputs STO_A (PWRR_A) and STO_B (PWRR_B) to avoid an unexpected restart of the motor before switching on and configuring the drive system.
- Do NOT operate the drive system with unknown settings or data.
- Perform a comprehensive commissioning test.

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

- Activate the configuration in the TwinCAT System Manager by opening the "Actions" menu.
- Choose the menu item "Activate Configuration..."

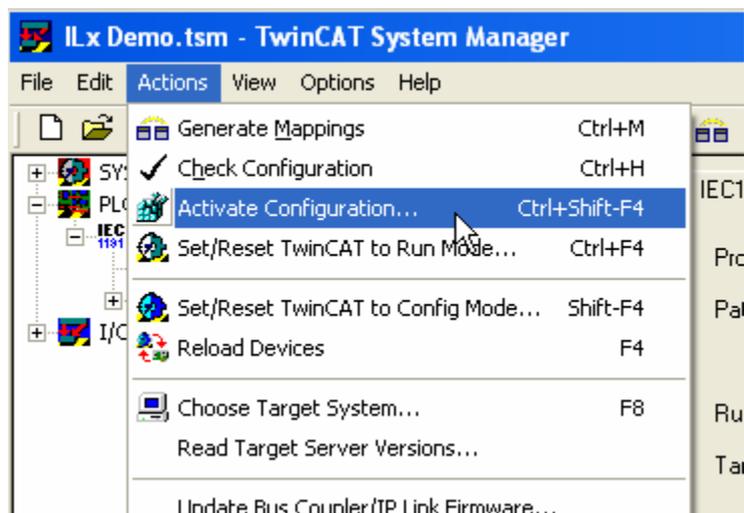


Bild 42) Activating the configuration

- In the dialog boxes displayed, click the "Yes" and "OK" buttons to load the configuration in the controller and to restart the controller.

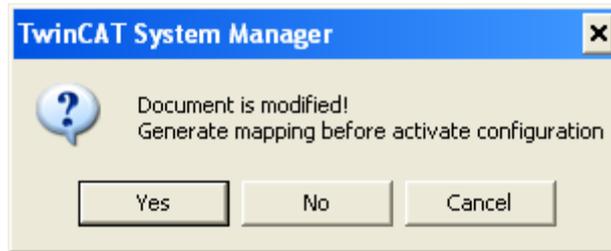


Bild 43) Message that the document has been modified



Bild 44) Message "Activate Configuration"



Bild 45) restart in Run Mode

- In order to start the controller, load the application with TwinCAT Control and start the application.

If there are errors, the appropriate messages are displayed in the message window. Analyze and fix these errors. If no errors occur, the controller is started and you can now check communication with the System manager.

2.6 Troubleshooting

TwinCAT System Manager

The Logger of the TwinCAT System Manager provides information on detected errors. To display the Logger, choose the menu item "Show Logger Output" from the "View" window.

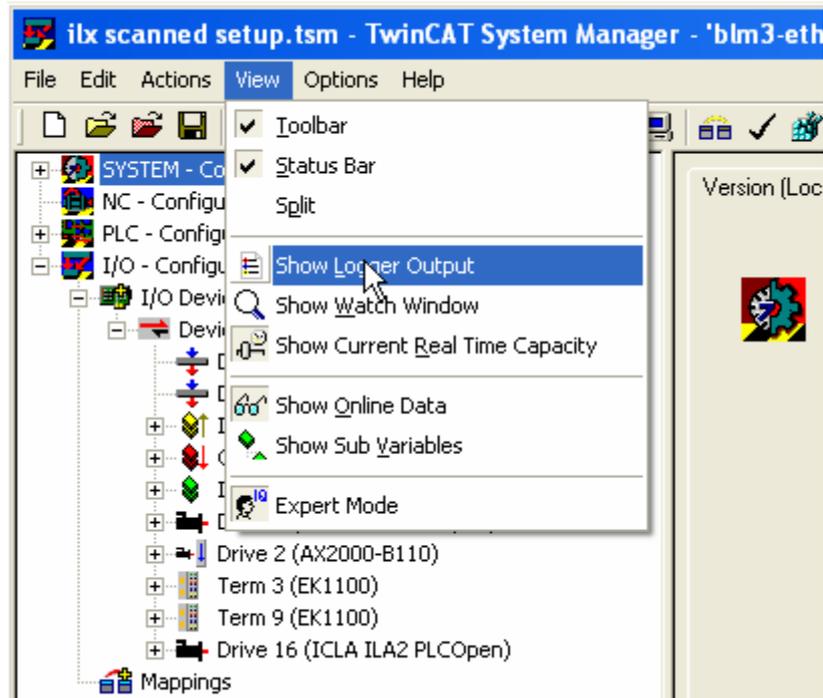


Bild 46) Displaying the Logger output

Communication status Lexium ILx2

The TwinCAT System Manager allows you to display and influence the communication state of the Lexium ILx2 drives. To do so, select the drive in the left window of the System manager. Then display the "Online" tab on the right pane. This tab displays the current state and the requested state. You can request the states with the corresponding buttons.

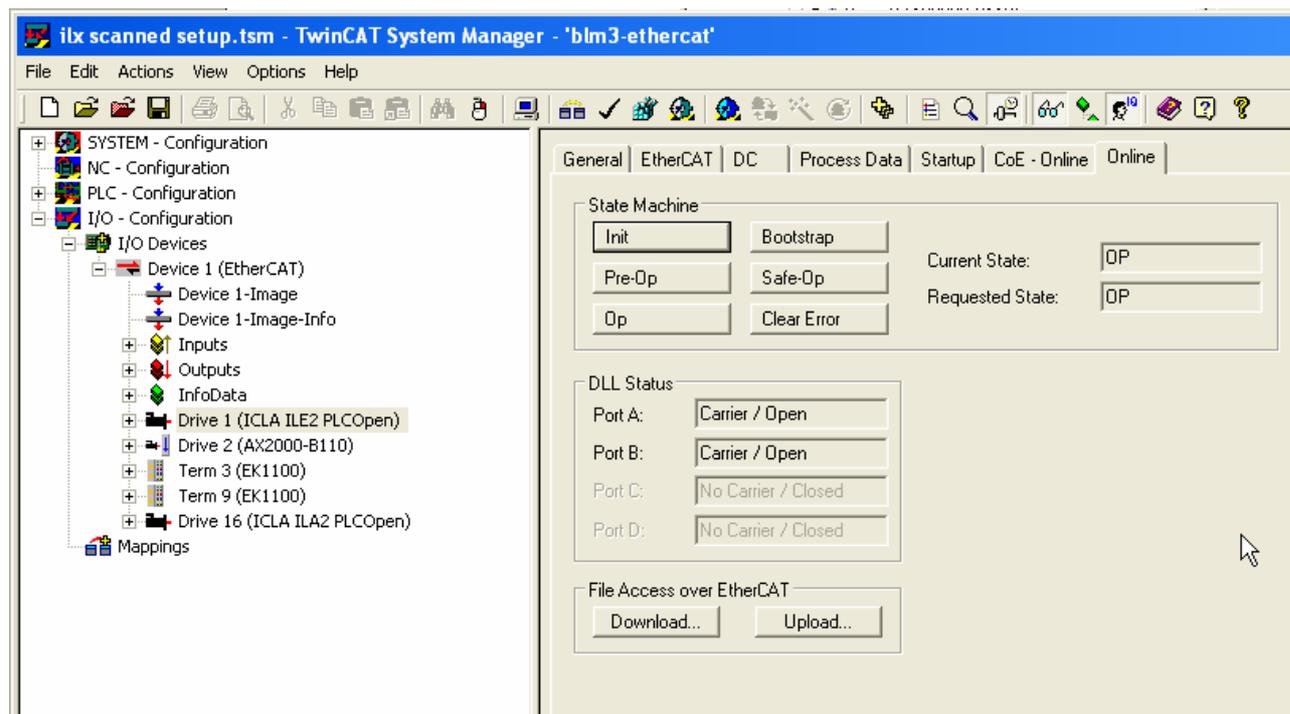


Bild 47) Communication states of the drives

Stopping the application

If the application in the Beckhoff controller is stopped, the drives automatically switch to the SAFEOP state. After a restart of the application, you must manually change the state of the drives to OPERATIONAL in the TwinCAT System Manager as described in the previous section.

2.7 Addressing the drives in TwinCAT

The previous chapters did not treat the subject of addressing the Lexium ILx2 drives. Addressing is not necessary when you work with TwinCAT since the System Manager addresses the EtherCAT network devices on the basis of their position in the network.

It is not necessary to set the addresses in the drives.

Notes on addressing

- The device closest to the controller is addressed first and displayed at the top of the list in the System Manager. If device and configuration do not match, the System Manager generates an error message.
- The System Manager checks the EtherCAT Vendor ID and the Product ID of the device and the configuration. If they match, the System manager does not generate an error message. If you use several devices with an identical Vendor ID and an identical Product ID, you must make sure the devices are properly arranged in the network. The System Manager does not detect if

two identical devices are interchanged.

2.8 Addressing the drives with Lexium CT

If you want to access the Lexium ILx2E drives via the Lexium CT commissioning software or a Web browser, the EtherCAT master must assign an IP address to the drives. No IP address is required for operation with EtherCAT.

- Select the drive in the TwinCAT System Manager.
- Right-click the "Advanced Settings" button on the "EtherCAT" tab.

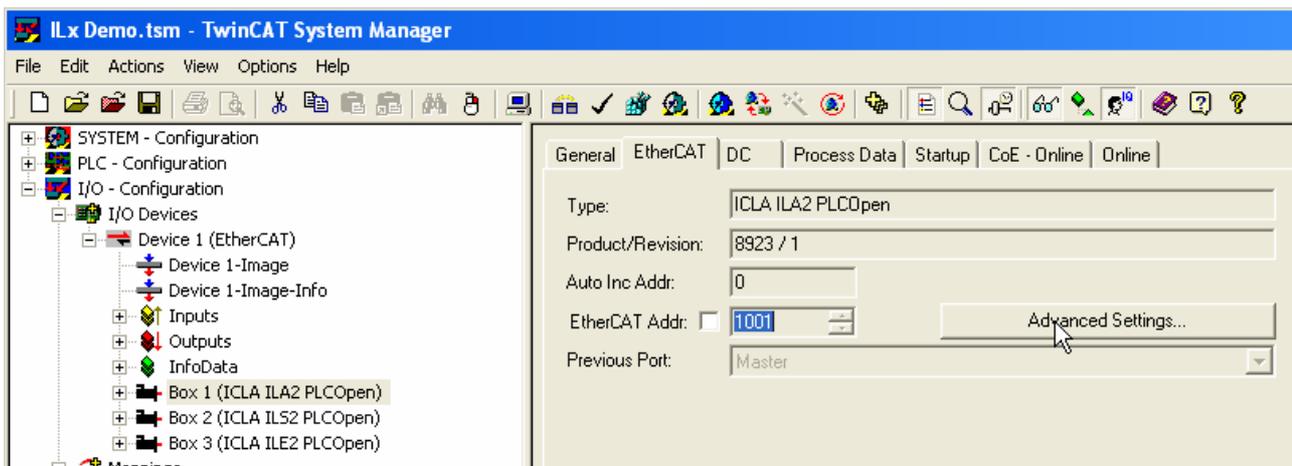


Bild 48) Selecting the drive in the System Manager

- In the "Advanced Settings" window, open the "Mailbox" node in the tree and select the "EoE" entry.
- Check the "Virtual Ethernet Port".
- Activate the configuration as described in chapter 2.5.1.

The IP addresses are assigned by the System Manager. Use this address to access the drive with the Lexium CT commissioning software after having activated the configuration. Automatically assigned addresses end with 0.

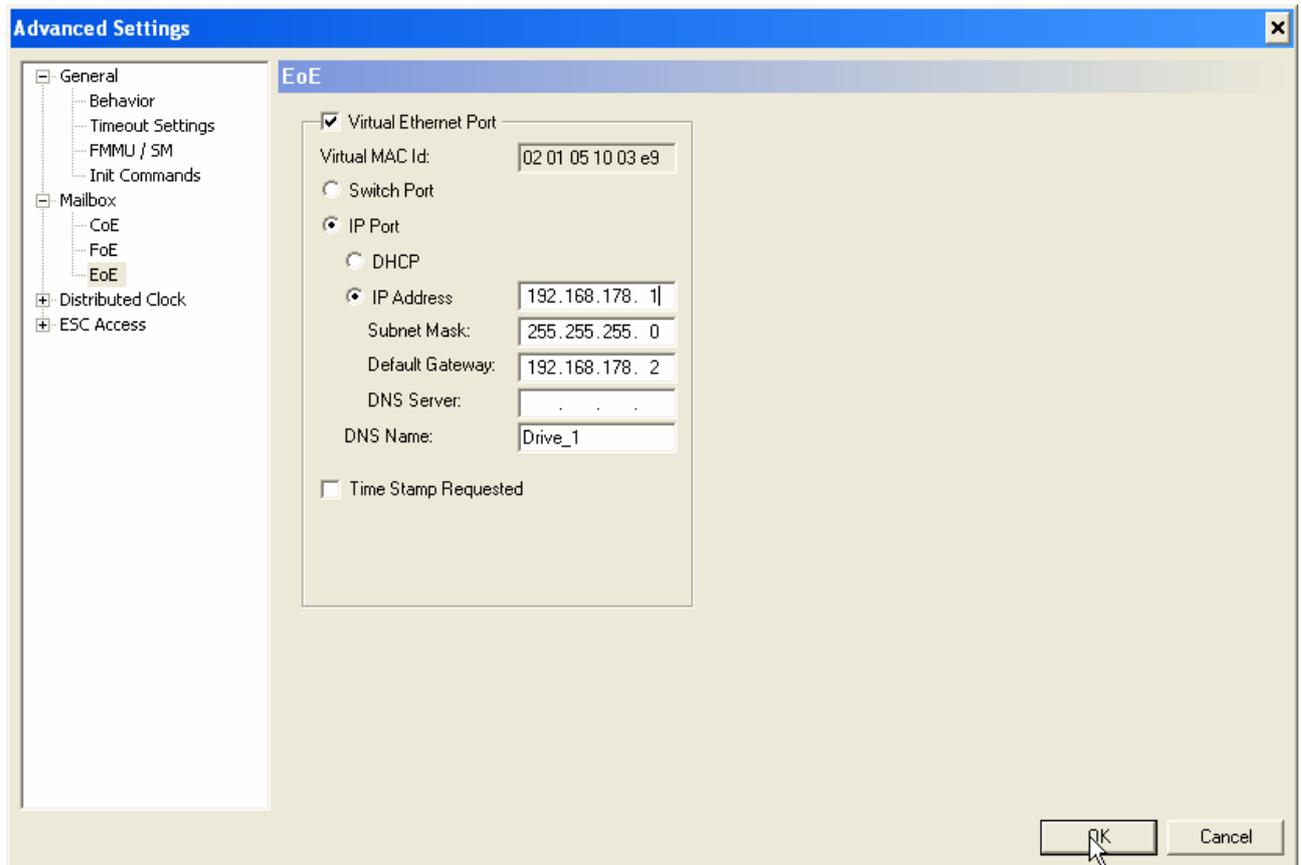


Bild 49) EoE – Ethernet over EtherCAT

- Enter an address that does not end with 0.

Note on access via Lexium CT

Please note that EtherCAT is a closed network; only devices with an EtherCAT interface may be directly connected to an EtherCAT network. To access the drive on the EtherCAT network from your PC, you must use a gateway. Devices with a gateway function include:

- The Beckhoff controller
- A Beckhoff Switchport for connection of an Ethernet device to EtherCAT
- See the manuals of the corresponding devices for details on the configuration.

3 Function blocks

The function blocks contained in the library 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.

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.

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 *_ETC* for the EtherCAT fieldbus and *_ILX* for the Lexium drives are used. This allows for unique identification.

Two typical examples of function block names:

- *MC_Power_ETC_ILX*
- *SetDriveRamp_ETC_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_ETC_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_ETC_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 can only be terminated by cancellation for Profile Velocity operating mode.

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

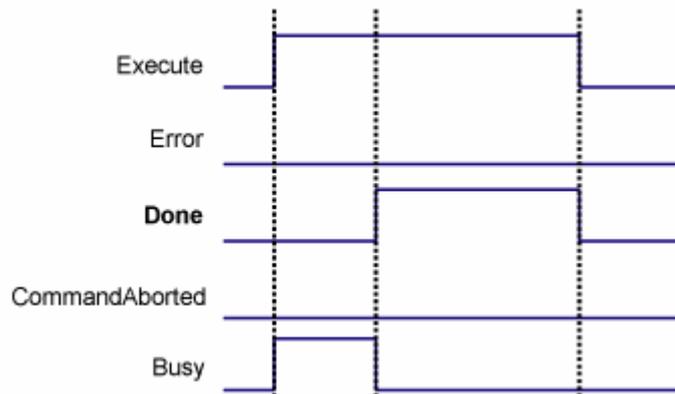


Bild 50) Execution completed without error

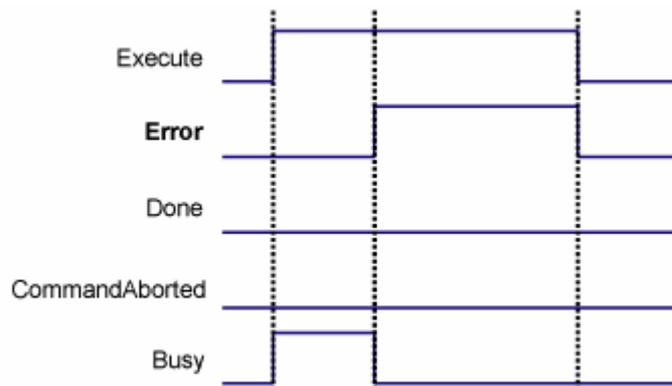


Bild 51) Execution terminated with error

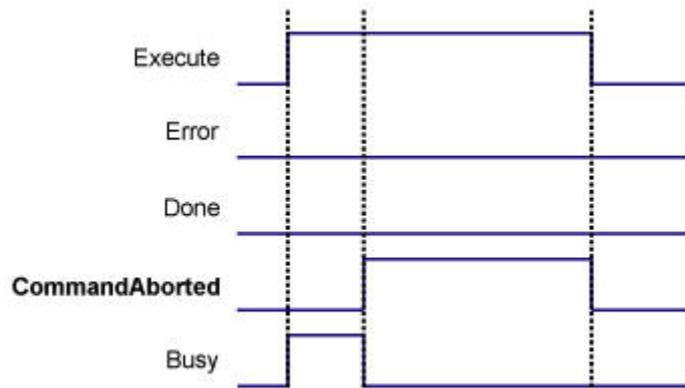


Bild 52) Cancellation of execution; another function block has taken over axis control

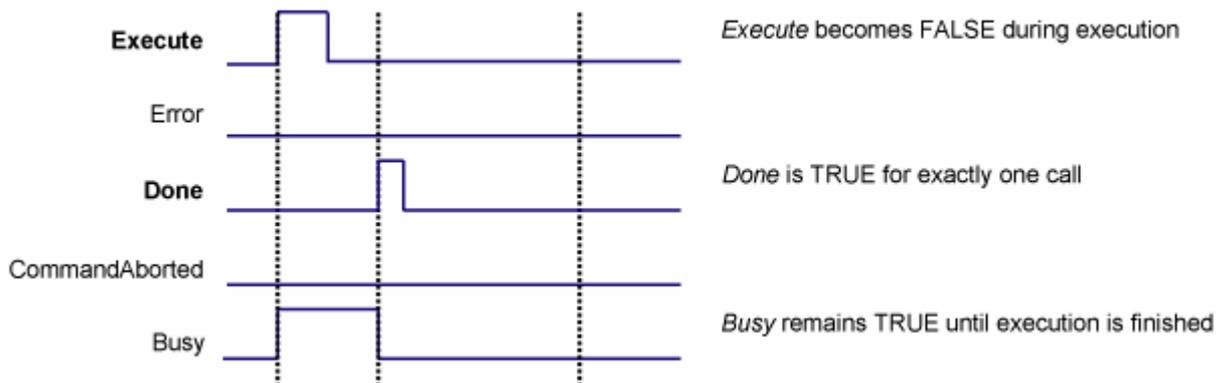


Bild 53) Execution completed without error after *Execute* was set to FALSE during execution

3.2.4 Signal diagram for function blocks with control input Enable

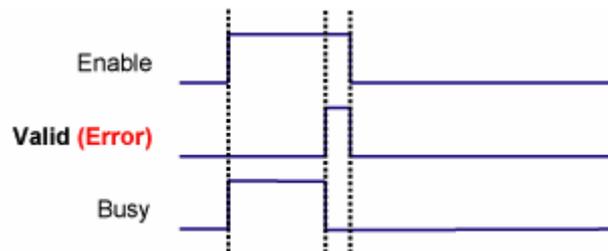


Bild 54) Single execution without (with) error [execution requires more than one call]



Bild 55) Single execution without (with) error [execution requires only one call]

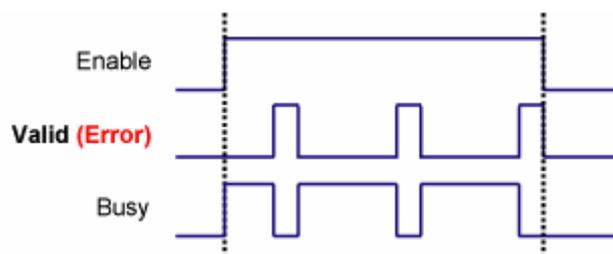


Bild 56) Multiple execution without (with) error [execution requires more than one call]

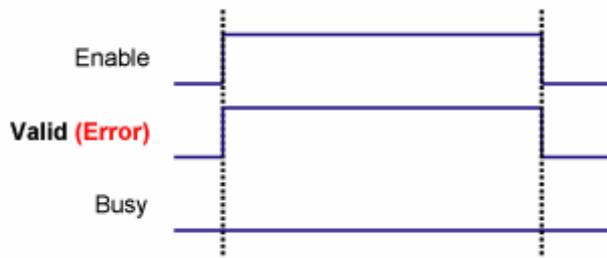


Bild 57) Multiple execution without (with) error [execution requires only one call]

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_ETC_ILX	Type DWORD (Value range 16#00000000..16#FFFFFFF) Contains the library version. In the case of hexadecimal representation, the numbers directly correspond to 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. It also enables and disables the power stage of the drive.

Function block	Task
MC_Power_ETC_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_ETC_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

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_ETC_ILX	Trigger reference movement
MC_SetPosition_ETC_ILX	Position setting

3.4.4 Profile Position operating mode

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_ETC_ILX	Absolute positioning
MC_MoveRelative_ETC_ILX	Relative positioning with reference to the current motor position
MC_MoveAdditive_ETC_ILX	Relative positioning with reference to the original position

3.4.5 Profile Velocity operating mode

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_ETC_ILX	Activate Profile Velocity operating mode

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_ETC_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_ETC_ILX	Adjust and start position capture
MC_AbortTrigger_ETC_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 ILx2E 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_ReadActualPosition_ETC_ILX	Read the the current position in user-defined units
MC_ReadActualVelocity_ETC_ILX	Reads the current speed in user-defined units
MC_ReadStatus_ETC_ILX	Reads the current status of the drive
MC_ReadParameter_ETC_ILX	Reads the specified parameters from the drive
GetVersion_ETC_ILX	Returns the library version
ReadActualPositionInc_ETC_ILX	Returns the current position in increments
ReadRefPosition_ETC_ILX	Reads the current reference position in user-defined units directly from the profile generator of the drive
ReadRefPositionInc_ETC_ILX	Reads the current reference position in increments directly from the profile generator of the drive
ReadRefVelocity_ETC_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 ILx2E drive. See the product manual for detailed descriptions of all parameters of the drive.

Function block	Task
MC_WriteParameter_ETC_ILX	Set parameter
SetDriveRamp_ETC_ILX	Set acceleration
StoreParameters_ETC_ILX	Permanently store all parameters
ResetParameters_ETC_ILX	Reset all parameters to the factory settings

3.4.10 Inputs/outputs

The digital inputs and digital outputs of the Lexium ILx2E can be read with the following parameters.

Function block	Task
MC_ReadDigitalInput_ETC_ILX	Read inputs
MC_ReadDigitalOutput_ETC_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_ETC_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_ETC_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.

The following error sources are possible:

- Function block errors
- Drive errors

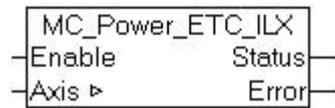
Drive errors can be the result of, for instance, invalid input values. Drive errors are usually caused by events such as reaching a limit switch. MC_Reset_ETC_ILX also resets the error in the drive.

Function block	Task
MC_ReadAxisError_ETC_ILX	Read error
MC_Reset_ETC_ILX	Delete error

3.5 Detailed description of function blocks

3.5.1 MC_Power_ETC_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_ETC_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 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* is set.

Note

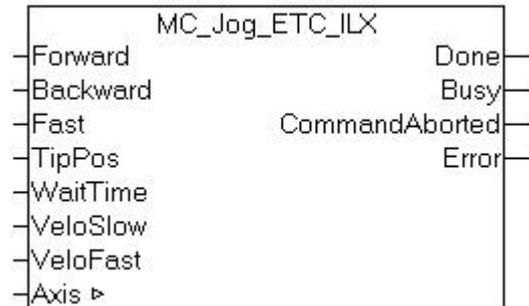
When the power supply to the drive is switched off, the motor current is also switched off. Since the drive can no longer signal this state transition, *Status* remains TRUE. When the power supply to the drive is switched on again, the fact that the motor is without current is signaled and the *Status* 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_ETC_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_ETC_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
	<p>movements from two values:</p> <p>FALSE: Speed VeloSlow is selected.</p> <p>TRUE: Speed VeloFast is selected.</p>
TipPos	<p>Type DINT</p> <p>(value range: 0..2147483647) initial value: 20</p> <p>0: The motor immediately starts a continuous movement.</p> <p>>0: Distance [usr] by which the motor moves after the start before it switches to continuous movement after the delay time (WaitTime) has elapsed.</p>
WaitTime	<p>Type UINT</p> <p>(value range: 1ms..32767ms) initial value: 500ms</p> <p>Delay time [ms], which starts after the motor has moved a defined distance (TipPos) and after which the motor switches to continuous movement.</p>
VeloSlow	<p>Type DINT</p> <p>(value range: 1..13200) initial value: 60</p> <p>Speed [rpm] for movement if Fast = FALSE.</p>
VeloFast	<p>Type: DINT</p> <p>(value range: 1..13200) initial value: 180</p> <p>Speed [rpm] for movement if Fast = TRUE.</p>
Axis	<p>Type: Axis_Ref_ETC_ILX</p> <p>(value range: <name of axis>) initial value: empty</p> <p>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.</p>

Output variables

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

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

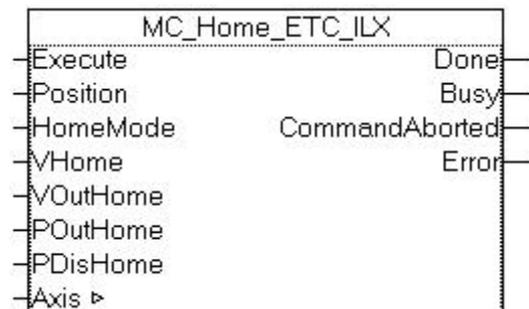
A jog movement is started with TRUE at *Forward* or *Backward*.

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_ETC_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

Variable	Possible values, meaning
	<p>output is FALSE .</p> <p>After termination of the execution, Execute determines the behavior of the outputs:</p> <p>FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
Position	<p>Type DINT</p> <p>(value range:-2147483648..2147483647) initial value: 0</p> <p>Position setting to position setting position (setting the absolute position) in user-defined units. Position is the current motor position after successful homing.</p>
HomeMode	<p>Type: UINT</p> <p>(value range 1..35) initial value: 1</p> <p>Please see the product manual for the modes supported by the drive.</p>
Vhome	<p>Type: UINT</p> <p>(value range: 1..13200) initial value: 60</p> <p>Speed for searching for the reference switch or a limit switch [rpm]. The drive stops when the switching edge has been detected.</p>
VoutHome	<p>Type: UINT</p> <p>(value range: 1..3000) initial value: 6</p> <p>Speed for moving away from the switch to the switching edge [rpm]. The maximum distance for searching for the switching edge can be limited with the parameter POutHome.</p>
PoutHome	<p>Type: DINT</p> <p>(value range: 0..2147483647) initial value: 0</p> <p>0: Monitoring of moving away from switch inactive</p> <p>> 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.</p>
PdisHome	<p>Type: DINT</p> <p>(value range: 1..2147483647)</p>

Variable	Possible values, meaning
	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_ETC_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

Configuration and start of the reference movement.

3.5.4 MC_SetPosition_ETC_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	<p>Type: BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>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 .</p> <p>After termination of the execution, Execute determines the behavior of the outputs:</p> <p>FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
Position	<p>Type: DINT</p> <p>(value range: -2147483648..2147483647) initial value: 0</p> <p>Position setting to position setting position in user-defined units</p>
Mode	<p>Type: BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>FALSE: Set current motor position to position.</p> <p>TRUE: Add position to current motor position.</p>
Axis	<p>Type: Axis_Ref_ETC_ILX</p> <p>(value range: <name of axis>) initial value: empty</p> <p>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.</p>

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_ETC_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
	<p>either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
Position	<p>Type: DINT</p> <p>(value range: -2147483648..2147483647) initial value: 0</p> <p>Value for the absolute target position in user-defined units.</p>
Velocity	<p>Type: INT</p> <p>(value range: 1..13200) initial value: 60)</p> <p>Value for the reference speed of the movement [rpm].</p>
Axis	<p>Type: Axis_Ref_ETC_ILX</p> <p>(value range: <name of axis>) initial value: empty</p> <p>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.</p>

Output variables

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

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

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_ETC_ILX allows you to read whether or not the absolute position reference is still available.

3.5.6 MC_MoveRelative_ETC_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.

Variable	Possible values, meaning
Distance	Type: DINT (value range: -2147483648..2147483647) initial value: 0 Value for the distance with reference to the 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 [rpm].
Axis	Type: Axis_Ref_ETC_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_ETC_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	<p>Type: BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>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 .</p> <p>After termination of the execution, Execute determines the behavior of the outputs:</p> <p>FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
Distance	<p>Type: DINT</p> <p>(value range: -2147483648..2147483647) initial value: 0</p> <p>Value for the original target position plus additional relative distance in user-defined units.</p>
Velocity	<p>Type: INT</p> <p>(value range: 1..13200) initial value: 60)</p> <p>Value for the reference speed of the movement [rpm].</p>
Axis	<p>Type: Axis_Ref_ETC_ILX</p> <p>(value range: <name of axis>) initial value: empty</p> <p>Name of the drive for which the function block is to be executed. A global data structure must</p>

Input variables

Variable	Possible values, meaning
Execute	<p>Type BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>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 .</p> <p>After termination of the execution, Execute determines the behavior of the outputs:</p> <p>FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
Velocity	<p>Type: INT</p> <p>(value range: -13200..13200) initial value: 0)</p> <p>Value for the reference speed of the movement [rpm].</p>
Axis	<p>Type: Axis_Ref_ETC_ILX</p> <p>(value range: <name of axis>) initial value: empty</p> <p>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.</p>

Output variables

Variable	Possible values, meaning
InVelocity	<p>Type: BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>FALSE: Target speed not (yet) reached. TRUE: Target speed reached.</p>
Busy	<p>Type BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.</p>
CommandAborted	<p>Type BOOL</p> <p>(value range: FALSE, TRUE)</p>

Variable	Possible values, meaning
	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 Profile Velocity operating mode with speed *Velocity* . When the target speed is reached, *InVelocity* is set.

Note

Changing speed *Velocity* to "0"

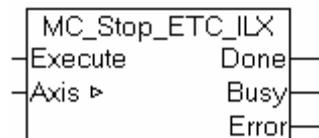
This allows you to stop the drive with the normal deceleration at any time. If the speed is set to "0", the drive is only stopped temporarily! This means that as soon as the speed is set to a value not equal to "0", the drive will immediately resume movement.

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_ETC_ILX allows you to read whether or not the absolute position reference is still available.

3.5.9 MC_Stop_ETC_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

Variable	Possible values, meaning
	<p>active. Execution is terminated once the Busy output is FALSE .</p> <p>After termination of the execution, Execute determines the behavior of the outputs:</p> <p>FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
Axis	<p>Type: Axis_Ref_ETC_ILX</p> <p>(value range: <name of axis>) initial value: empty</p> <p>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.</p>

Output variables

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

Task of the function block

Stops the drive with a torque ramp.

Note

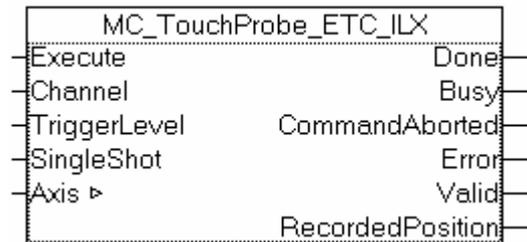
This function cannot be aborted by other function blocks. As long as Execute = TRUE, no other function block can be started. Even after

standstill, the drive remains blocked.

The function block decelerates the motor with a torque ramp. The parameter LIM_I_maxHalt (see drive 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_ETC_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 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

Variable	Possible values, meaning
	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_ETC_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.
Valid	Type: BOOL (value range: FALSE, TRUE)

Variable	Possible values, meaning
	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

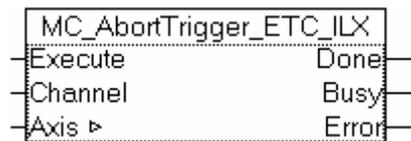
Adjust and start position capture.

Note

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

3.5.11 MC_AbortTrigger_ETC_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.
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_ETC_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

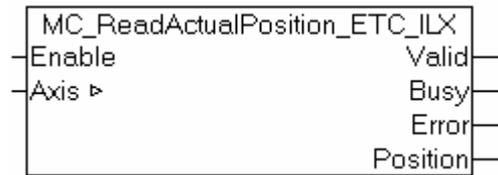
Abort position capture

Note

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

3.5.12 MC_ReadActualPosition_ETC_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_ETC_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

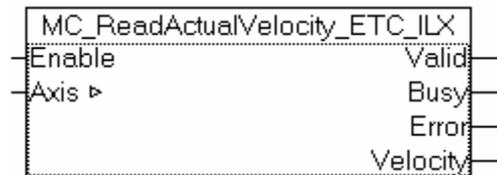
Variable	Possible values, meaning
	TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Position	Type: DINT (Value range:) Actual motor position in user-defined units

Task of the function block

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

3.5.13 MC_ReadActualVelocity_ETC_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_ETC_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

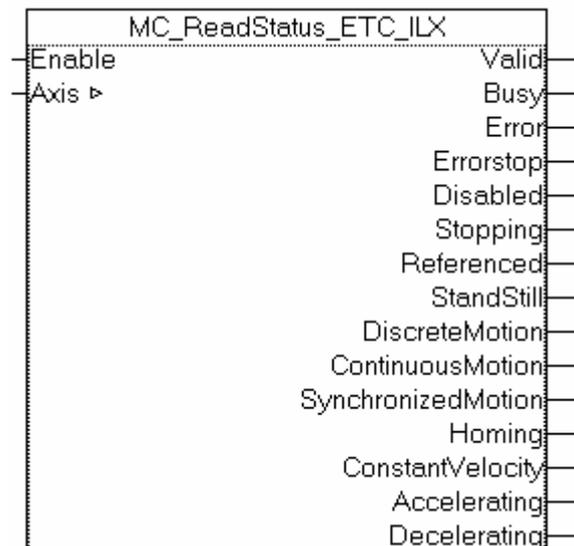
Variable	Possible values, meaning
	TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
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.
Velocity	Type: INT (value range: -13200..+13200) Current speed of rotation of motor [rpm]

Task of the function block

Read the current speed of rotation of the motor [rpm]

3.5.14 MC_ReadStatus_ETC_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
----------	--------------------------

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_ETC_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 TRUE: Motor current is switched off.

Variable	Possible values, meaning
	FALSE: Motor current is switched on.
Stopping	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Function block MC_Stop_ETC_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.
Decelerating	Type: BOOL

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

Task of the function block

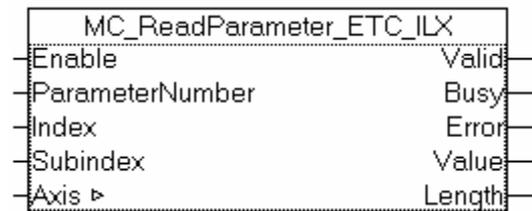
The current drive state is output.

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_ETC_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
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]

Variable	Possible values, meaning
	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 [rpm] 11: Current reference speed [rpm] Other numbers are not supported.
Index	Type: UINT (value range: 0..65535) Index of the object to be read; the objects are listed in the manual with their indexes and subindexes. 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 <i>ParameterNumber</i> = 0.
Axis	Type: Axis_Ref_ETC_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.

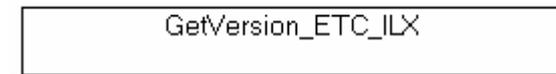
Variable	Possible values, meaning
	FALSE: No error has (yet) occurred during execution.
Value	Type: DINT (value range: -2147483648..2147483647) initial value: 0 Value of device parameter
Length	Type: UINT (value range: 0..65535) initial value: 0 Length in bytes of read device parameter

Task of the function block

Read an object from the device parameter list.

3.5.16 GetVersion_ETC_ILX

Graphical representation



Output variables

Variable	Possible values, meaning
GetVersion_ETC_ILX	Type DWORD (Value range 16#00000000..16#FFFFFFFF) Contains the library version. In the case of hexadecimal representation, the numbers directly correspond to 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 ReadActualPositionInc_ETC_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_ETC_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 <i>Position</i> is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output <i>Position</i> 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.

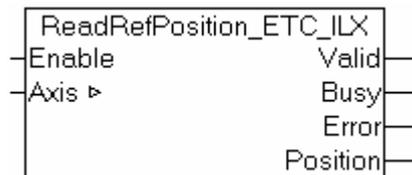
Variable	Possible values, meaning
Error	Type: BOOL (value range:TRUE; FALSE) initial value: FALSE TRUE: Execution was terminated with an error. FALSE: No error has (yet) occurred during execution.
Position	Type: DINT (value range: -2147483648..2147483647)) Actual motor position in increments

Task of the function block

Read actual motor position in increments.

3.5.18 ReadRefPosition_ETC_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_ETC_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 <i>Position</i> is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output <i>Position</i> 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 in user-defined units

Note

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

3.5.19 ReadRefPositionInc_ETC_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_ETC_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 <i>Position</i> is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output <i>Position</i> 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

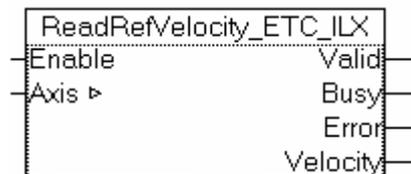
Read the current reference position in the profile generator in motor increments.

Note

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

3.5.20 ReadRefVelocity_ETC_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_ETC_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 [rpm].

Task of the function block

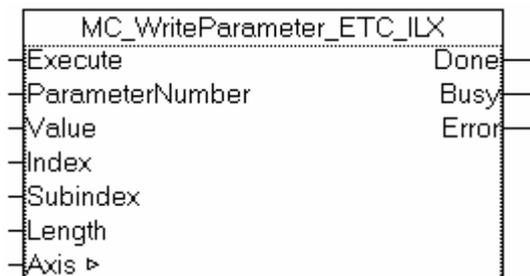
Reads the current speed of rotation in the profile generator.

Note

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

3.5.21 MC_WriteParameter_ETC_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	<p>Type BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>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 .</p> <p>After termination of the execution, Execute determines the behavior of the outputs:</p> <p>FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
ParameterNumber	<p>Type: INT</p> <p>(value range: 0..65535)</p> <p>0: Parameter is selected with index and subindex.</p> <p>>0: Number of the parameter to be written:</p> <p>2: Position of positive software limit switch [usr]</p> <p>3: Position of negative software limit switch [usr]</p> <p>4: Enable (bit0=1) or disable (bit0=0) positive software limit switch</p> <p>5: Enable (bit0=1) or disable (bit0=0) negative software limit switch</p> <p>Other numbers are not supported.</p>
Value	<p>Type: DINT</p> <p>(value range: -2147483648..2147483647) initial value: 0</p> <p>Value of device parameter</p>
Index	<p>Type: UINT</p> <p>(value range: 0..65535)</p> <p>Index of the object to be written; the objects are listed in the manual with their indexes and subindexes.</p> <p>Only valid if ParameterNumber = 0.</p>
Subindex	<p>Type: UINT</p> <p>(value range: 0..255)</p> <p>Subindex of the object to be written; the objects are listed in the manual with their indexes and subindexes.</p> <p>Only valid if ParameterNumber = 0.</p>

Variable	Possible values, meaning
Length	Type: UINT (value range: 0..65535) initial value: 0 Length in bytes of device parameter to be written
Axis	Type: Axis_Ref_ETC_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

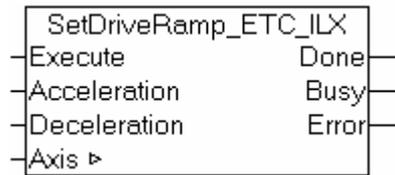
Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed.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

Write an object in the parameter list.

3.5.22 SetDriveRamp_ETC_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	<p>Type: BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>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 .</p> <p>After termination of the execution, Execute determines the behavior of the outputs:</p> <p>FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
Acceleration	<p>Type: UINT</p> <p>(value range: 1..3000000) initial value: 600</p> <p>Value for steepness of acceleration ramp [1 rpm/s]</p>
Deceleration	<p>Type: UINT</p> <p>(value range: 200..3000000) initial value: 750</p> <p>Value for steepness of deceleration ramp [1 rpm/s]</p>
Axis	<p>Type: Axis_Ref_ETC_ILX</p> <p>(value range: <name of axis>) initial value: empty</p> <p>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.</p>

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

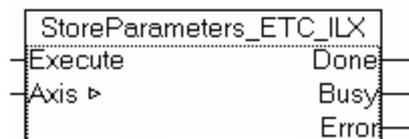
Defines the acceleration and deceleration ramps. The device controls the acceleration and deceleration behavior of the motor with ramp functions. The ramp steepness determines the change in speed of the motor. *Acceleration* defines the acceleration, *Deceleration* the deceleration.

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_ETC_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	<p>Type BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>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 .</p> <p>After termination of the execution, Execute determines the behavior of the outputs:</p> <p>FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
Axis	<p>Type: Axis_Ref_ETC_ILX</p> <p>(value range: <name of axis>) initial value: empty</p> <p>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.</p>

Output variables

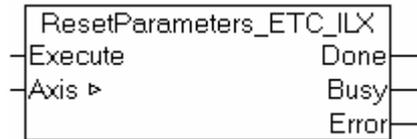
Variable	Possible values, meaning
Done	<p>Type: BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>TRUE: Execution terminated without errors.</p> <p>FALSE: Execution not (yet) terminated without errors.</p>
Busy	<p>Type BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> 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

Task of the function block

Save all parameters to the EEPROM of the Lexium ILx2E drive.

3.5.24 ResetParameters_ETC_ILX

Graphical representation



Input variables

Variable	Possible values, meaning
Execute	<p>Type BOOL</p> <p>(value range: FALSE, TRUE) initial value: FALSE</p> <p>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 .</p> <p>After termination of the execution, Execute determines the behavior of the outputs:</p> <p>FALSE: At the same time as Busy = FALSE, either Done, Error or CommandAborted becomes TRUE for exactly one call.</p> <p>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.</p>
Axis	<p>Type: Axis_Ref_ETC_ILX</p> <p>(value range: <name of axis>) initial value: empty</p> <p>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.</p>

Output variables

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

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.

Task of the function block

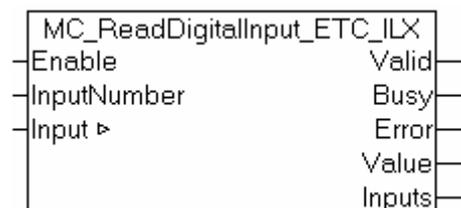
Reset all parameters to their state after the "First Setup" (see manual). All parameter values are reset to the default values with the exception of the communication parameters, the control mode and the logic type ("source" or "Sink" of the inputs/outputs).

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_ReadDigitalInput_ETC_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
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_ETC_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 <i>Value</i> is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output <i>Value</i> 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.															
Inputs	Type: WORD (value range: 16#00..16#3F) initial value: 16#00 Input assignment of the drive:															
	<table border="1"> <thead> <tr> <th>Input</th> <th>Bit</th> <th>Signal</th> </tr> </thead> <tbody> <tr> <td>LIO0</td> <td>0</td> <td>/LIMP</td> </tr> <tr> <td>LIO1</td> <td>1</td> <td>/LIMN</td> </tr> <tr> <td>LIO2</td> <td>2</td> <td>Available as required</td> </tr> <tr> <td>LIO3</td> <td>3</td> <td>/REF</td> </tr> </tbody> </table>	Input	Bit	Signal	LIO0	0	/LIMP	LIO1	1	/LIMN	LIO2	2	Available as required	LIO3	3	/REF
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

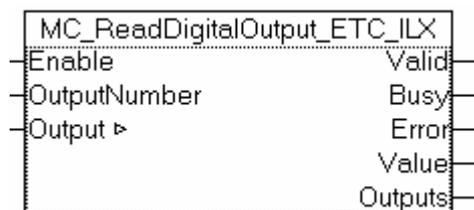
The meaning of the signal is described in the product manual.

The input I0 (/REF) is only used by the drive for homing to the reference signal (see MC_Home_ETC_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_ReadDigitalOutput_ETC_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_ETC_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 <i>Value</i> is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output <i>Value</i> 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.

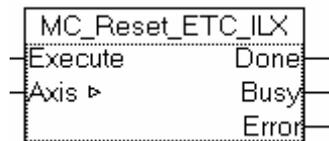
Variable	Possible values, meaning										
	FALSE: Read output has 0 level.										
Outputs	Type: WORD (value range: 16#00..16#03) initial value: 16#00 Output assignment of the drive: <table border="1"> <thead> <tr> <th>Output</th> <th>Bit</th> </tr> </thead> <tbody> <tr> <td>LIO0</td> <td>0</td> </tr> <tr> <td>LIO1</td> <td>1</td> </tr> <tr> <td>LIO2</td> <td>2</td> </tr> <tr> <td>LIO3</td> <td>3</td> </tr> </tbody> </table>	Output	Bit	LIO0	0	LIO1	1	LIO2	2	LIO3	3
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_ETC_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_ETC_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

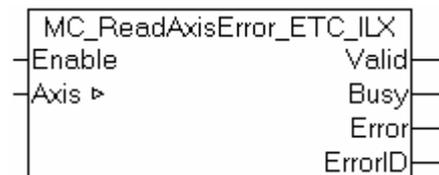
Variable	Possible values, meaning
Done	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: Execution terminated without errors. FALSE: Execution not (yet) terminated without errors.
Busy	Type: BOOL (value range: FALSE, TRUE) initial value: FALSE TRUE: The function block is being executed. FALSE: Execution terminated; the function block is not active.
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_ETC_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_ETC_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 <i>ErrorID</i> is valid. FALSE: Execution not (yet) terminated without errors. The read value at the parameter output <i>ErrorID</i> 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_ETC_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	Braking resistor overcurrent

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 ² t)
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 ² t)
16#4402	17410	0	Warning overload (I ² t) in braking resistor
16#5200	20992	3	No connection to the 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 the motor encoder
16#5204	20996	3	Connection to motor encoder lost
16#5430	21552	0	System error: EEPROM read error
16#5431	21553	0	System error: EEPROM write error
16#5435	21557	0	System error: EEPROM not formatted
16#5437	21559	0	System error: EEPROM checksum error 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 encoder signals
16#5602	22018	4	Interruption or incorrect 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 to motor encoder occupied
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	CANopen over EtherCAT: CAN overflow (message lost)
16#8130	33072	2	CANopen over EtherCAT: Heartbeat or Life Guard error
16#8201	33281	0	CANopen over EtherCAT: RxPDO1 could not be processed
16#8202	33282	0	CANopen over EtherCAT: RxPDO2 could not be processed
16#8203	33283	0	CANopen over EtherCAT: RxPDO3 could not be processed
16#8204	33284	0	CANopen over EtherCAT: RxPDO4 could not be processed
16#8205	33285	0	CANopen over EtherCAT: TxPDO could not be processed
16#8206	33286	0	CANopen over EtherCAT: 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		EtherCAT: Initialization error
16#B601	46593		EtherCAT: Realtime data error
16#B602	46594		EtherCAT: Realtime data warning
16#B603	46595		EtherCAT: Protocol-specific error
16#B604	46596		EtherCAT: Protocol-specific warning
16#B605	46597		EtherCAT: Unknown error
16#B606	46598		EtherCAT: Delayed parameter access to module
16#B607	46599		EtherCAT: Is currently processing another request
16#B608	46600		EtherCAT: Realtime Hot-Reset
16#B609	46601		EtherCAT: Realtime Hot-Stop
16#B60A	46602		EtherCAT: Timeout in internal communication
16#B60B	46603		EtherCAT: 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