BECKHOFF New Automation Technology

Manual | EN

CXxxxx-B110

EtherCAT Slave Optional Interface for CX9020, CX5xx0 and CX20xx

2023-12-13 | Version: 1.3

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

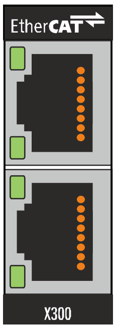
Information on handling the product

ℹ This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Documentation issue status

Version	Changes
1.0	First version
1.1	Chapter Diagnostic LEDs reworked
1.2	Chapter Technical data - EtherCAT reworked
1.3	Chapter "Technical data- EtherCAT" and "EtherCAT connection" adjusted.

2 System overview EtherCAT



The Beckhoff Embedded PCs can be ordered ex works with an optional interface, e.g. PROFIBUS, CANopen or RS232. Some of the optional interfaces can be delivered as master or slave.

The following Embedded PCs can be ordered with an optional interface:

- CX9020
- CX50x0
- CX51x0
- CX20x0

EtherCAT slave (B110)

The optional interface B110 is an EtherCAT slave and enables an Embedded PC to be used as subordinate decentral controller for configuring complex or modular systems.

The optional interface B110 offers the possibility to operate an Embedded PC as an EtherCAT slave device, which can then be integrated into an existing topology with further EtherCAT devices.

The EtherCAT slave receives external process data from the master and processes them or returns data from its own process periphery to the master after processing.

The optional EtherCAT slave interface is parameterized and configured in TwinCAT.

Further Information
EtherCAT system documentation https://infosys.beckhoff.com/content/1033/ethercatsystem/index.html?id=7792317249389602120
EtherCAT Technology Group https://www.ethercat.org/default.htm

Further information about EtherCAT and features such as cable redundancy, distributed clocks or diagnostics can be found in the EtherCAT system documentation or on the website of the EtherCAT Technology Group.

Functioning

EtherCAT is the real-time Ethernet technology from Beckhoff, which makes the familiar Ethernet telegram real-time capable. In contrast to Ethernet, the telegram is no longer initially received and interpreted in each connection, before the process data are copied. Instead, each EtherCAT device picks up the data intended for it while the telegram passes through the device. In the same way, input data are dynamically added to the data stream.

The slave detects the commands intended for it and executes them accordingly. The process is hardware-implemented in the slave controller and is therefore independent of the protocol stack software run times or the processor power. The last EtherCAT slave returns the processed telegram, so that it is sent from the first slave to the controller as a response telegram.

Protocol

The EtherCAT protocol is optimized for process data and is either transported directly in the Ethernet frame or packed into UDP/IP datagrams. The UDP version is used in situations where EtherCAT segments in other subnets are addressed via routers. Ethernet frames may contain several EtherCAT telegrams, with each telegram serving a particular memory area of the logical process image with a size of up to 4 GB. The data sequence is independent of the physical order of the EtherCAT terminals in the network; addressing can be in any order. Broadcast, Multicast and communication between slaves are possible. The protocol can also handle parameter communication, which typically is acyclical.

Performance

EtherCAT reaches new dimensions in network performance. The update time for the data from 1.000 distributed inputs/outputs is only 30 µs - including terminal cycle time. Up to 1.486 bytes of process data can be exchanged with a single Ethernet frame - this is equivalent to almost 12.000 digital inputs and outputs. The transfer of this data quantity only takes 300 µs.

Topology

Line, tree or star: EtherCAT supports almost any topology. The bus or line structure known from the fieldbuses thus also becomes available for Ethernet. Particularly useful for system wiring is the combination of line and junctions or stubs. The required interfaces exist on the couplers; no additional switches are required. Naturally, the classic switch-based Ethernet star topology can also be used.

2.1 EtherCAT Highlights

Performance	
• 256 digital-I/Os in 12 µs • 1.000 digital-I/Os in 30 µs	• 200 analog-I/Os (16 Bit) in 50 µs, corresponding to 20-kHz sampling rate • 100 servo axes every 100 µs • 12,000 digital I/Os in 350 µs

Topology	
• line, tree or star topology • up to 65,535 devices • network size: almost unlimited (> 500 km) • operation with or without switches • cost-effective cabling: Industrial Ethernet patch cable (CAT5)	• twisted pair physical layer: • Ethernet 100BASE-TX, up to 100 m between 2 devices • alternative: fibre-optic cable variants 50 to 2,000 m • hot connect/disconnect of bus segments

Address space	
• network-wide process image: 4 Gbyte • device process image: 1 bit to 64 kbyte	• address allocation: freely configurable • device address selection: automatically via software

Protokoll	
• optimised protocol directly within the Ethernet frame • fully hardware-implemented • for routing and socket interface: UDP datagram • processing while passing	• distributed clock for accurate synchronisation • time stamp data types for resolution in the nanosecond range • oversampling data types for high-resolution measurements

Diagnostics	
• breaking point detection • Topology View	• continuous “quality of line” measurement enables accurate localisation of transmission faults

Interfaces	
• switch port terminal for standard Ethernet devices • fieldbus terminals for fieldbus devices • decentralised serial interfaces	• communication gateways • gateway to other EtherCAT systems

Openness	
• fully Ethernet-compatible • operation with switches and routers possible, but not required • mixed operation with other protocols also possible • internet technologies (Web server, FTP, etc.)	• compatible with the existing Bus Terminal range • protocol is published completely • EtherCAT is IEC, ISO and SEMI standard.

2.2 Technical data - EtherCAT

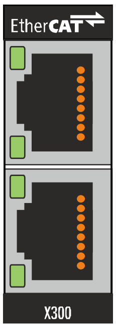
Optional interface B110

Technical data	B110
Fieldbus	EtherCAT (Slave)
Data transfer rate	100 Mbaud
Bus interface	EtherCAT IN and Out (2 x RJ45)
Bus devices	Up to 65,535 devices
max. process image	480 Byte in / 480 Byte out
Properties	Protocols (real-time): EtherCAT Protocols (non-real-time): CoE, AoE Distributed Clocks, Hot-Connect, Mailbox, ADS-Routing

3 Connection and cabling

3.1 EtherCAT connection

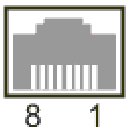
The latest generation of Embedded PCs can be ordered with the optional EtherCAT slave interface (B110). The optional interface B110 is identified as X300 on the devices and has as black border to identify it.



The incoming EtherCAT signal is connected to the upper port of the X300 interface. The lower port forwards the signal to further EtherCAT slave devices. Up to 480 bytes of input and output data or 256 variables can be connected via the EtherCAT interface.

It is not possible to create 480 bytes individually, for sample, since this would require 480 variables.

Assignment of the LAN ports (X300)



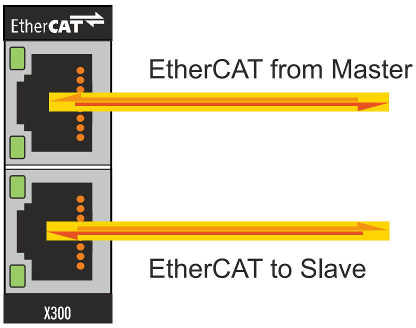
PIN	Signal	Description
1	TD +	Transmit +
2	TD -	Transmit -
3	RD +	Receive +
4	connected	reserved
5		
6	RD -	Receive -
7	connected	reserved
8		

3.2 Cabling

Use the following components to connect an Embedded PC with integrated optional interface B110 to an EtherCAT master or a further EtherCAT slave:

1. Ethernet/EtherCAT RJ45 connector.
2. RJ45 patch cable (CAT5).

Use PUR cables in robust industrial quality. Due to their mechanical properties and shielding, these cables are suitable for industrial environments. Do not use ordinary office cables. Suitable cables in robust industrial quality can be ordered from Beckhoff.



The incoming EtherCAT signal is connected to the upper port of the X300 interface. The lower port forwards the signal to further EtherCAT slave devices. The lower port can also be used for cable redundancy, in which case it is connected to the master.

Transmission standards

10Base5

The transmission medium for 10Base5 consists of a thick coaxial cable ("yellow cable") with a max. data transfer rate of 10 Mbaud arranged in a line topology with branches (drops) each of which is connected to one network device. Because all the devices are in this case connected to a common transmission medium, it is inevitable that collisions occur often in 10Base5.

10Base2

10Base2 (Cheaper net) is a further development of 10Base5, and has the advantage that the coaxial cable is cheaper and, being more flexible, is easier to lay. It is possible for several devices to be connected to one 10Base2 cable. It is frequent for branches from a 10Base5 backbone to be implemented in 10Base2.

10BaseT

Describes a twisted pair cable for 10 Mbaud. The network here is constructed as a star. It is no longer the case that every device is attached to the same medium. This means that a broken cable no longer results in failure of the entire network. The use of switches as star couplers enables collisions to be reduced. Using full-duplex connections they can even be entirely avoided.

100BaseT

Twisted pair cable for 100 Mbaud. It is necessary to use a higher cable quality and to employ appropriate hubs or switches in order to achieve the higher data rate.

10BaseF

The 10BaseF standard describes several optical fiber versions.

Short description of the 10BaseT and 100BaseT cable types

Twisted-pair copper cable for star topologies, where the distance between two devices may not exceed 100 meters.

UTP

Unshielded twisted-pair
This type of cable belongs to category 3, and is not recommended for use in an industrial environment.

S/UTP

Screened/unshielded twisted-pair (shielded with copper braid)
Has an overall shield of copper braid to reduce influence of external interference. This cable is recommended for use with Bus Couplers.

FTP

Foiled shielded twisted-pair (shielded with aluminum foil)
This cable has an outer shield of laminated aluminum and plastic foil.

S/FTP

Screened/foiled shielded twisted-pair (shielded with copper braid and aluminum foil)
Has a laminated aluminum shield with a copper braid on top. Such cables can provide up to 70 dB reduction in interference power.

STP

Shielded twisted-pair
Describes a cable with overall shielding without further specification of the type of shielding.

S/STP

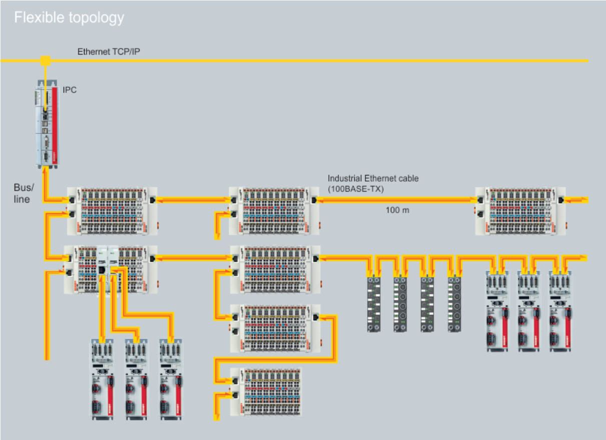
Screened/shielded twisted-pair (wires are individually shielded)
This identification refers to a cable with a shield for each of the two wires as well as an outer shield.

ITP

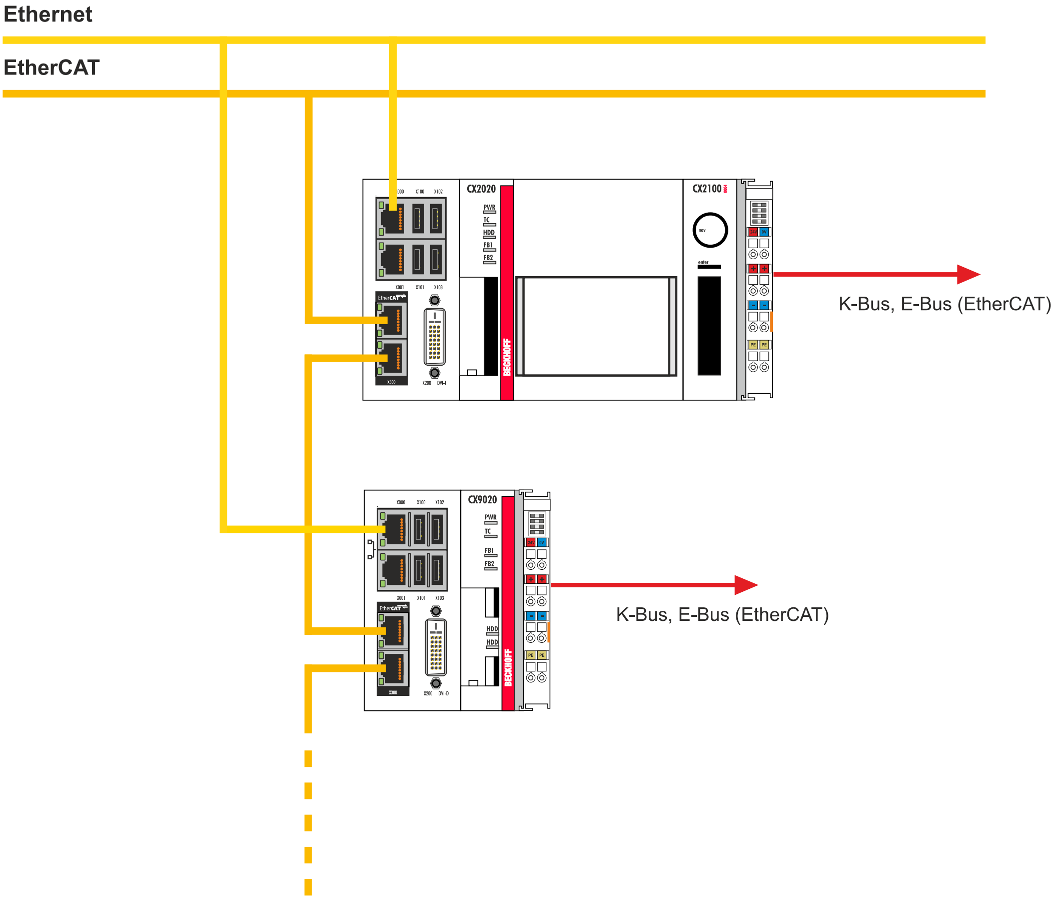
Industrial Twisted-Pair
The structure is similar to that of S/STP, but, in contrast to S/STP, it has only two pairs of conductors.

3.3 Topology

EtherCAT offers free choice of topology and maximum wiring flexibility. The EtherCAT devices can be wired with or without switches, lines or tree topology. EtherCAT supports almost any topology. Addresses are assigned automatically, and an IP address does not have to be set separately.



It is possible to operate up to 65.535 EtherCAT devices, with a maximum distance of 100 m between two devices. EtherCAT devices with optical fibers can reach a cable length of 20.000 m. Twisted-pair Ethernet 100BASE-TX is commonly used as physical layer. The Hot Connect function can be used to subdivide whole bus segments into groups for rapid replacement.



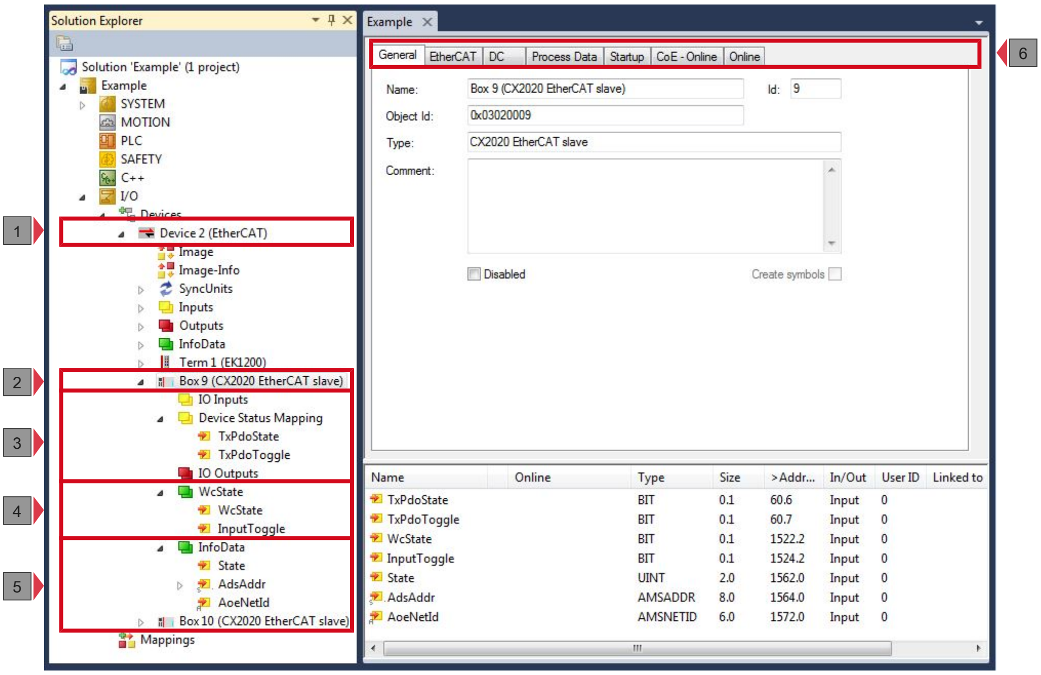
4 TwinCAT tabs

In TwinCAT, information and settings for the EtherCAT interface are added under tabs. The main TwinCAT tabs are described in this section. In addition, the section illustrates how the EtherCAT interface is displayed in the tree view under TwinCAT.

The tree view and the tabs for a EtherCAT interface are identical under TwinCAT2 and TwinCAT3.

4.1 Tree view

An EtherCAT master and an EtherCAT slave are displayed as follows in the tree view:



Name: Box 9 (CX2020 EtherCAT slave) Id: 9

Object Id: 0x03020009

Type: CX2020 EtherCAT slave

Comment:

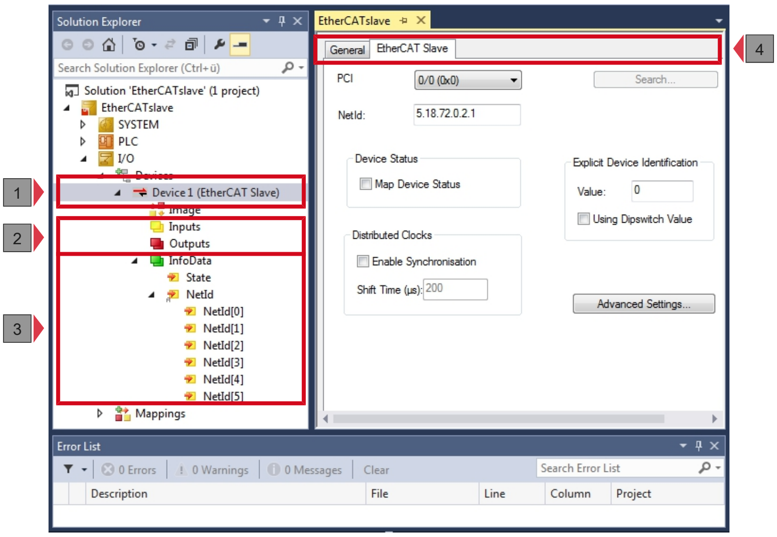
□ Disabled Create symbols □

Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to
TxPdoState		BIT	0.1	60.6	Input	0	
TxPdoToggle		BIT	0.1	60.7	Input	0	
WcState		BIT	0.1	1522.2	Input	0	
InputToggle		BIT	0.1	1524.2	Input	0	
State		UINT	2.0	1562.0	Input	0	
AdsAddr		AMSADDR	8.0	1564.0	Input	0	
AoeNetId		AMSNETID	6.0	1572.0	Input	0	

In this sample the slave was linked to the master. TwinCAT was then scanned for the master, and the master was added in TwinCAT together with the slave.

No.	Description
1	EtherCAT master with corresponding input and output variables
2	The EtherCAT slave is shown below the EtherCAT master. The slave is automatically added to the tree view, if the master and all connected devices are scanned.
3	Input and output variables of the EtherCAT slave. Process data can be created among the input and output variables, which can then be exchanged cyclically. The Device Status Mapping variables, which are used for slave diagnostics, are also displayed here.
4	The variables under WcState indicate whether the EtherCAT slave takes part in the cyclic process data traffic successfully and error-free. The variable InputToggle counts new valid incoming telegrams.
5	InfoData contains input variables, which provide information about the EtherCAT slave, such as status and NetId.
6	Further settings for the EtherCAT slave can be implemented under the tabs.

If only the EtherCAT slave is scanned and then added in the tree view, the EtherCAT is displayed differently. An EtherCAT slave alone is displayed in the tree view as follows:



PCI: 0/0 (0x0)

NetId: 5.18.72.0.2.1

Device Status

□ Map Device Status

Explicit Device Identification

Value: 0

□ Using Dipswitch Value

Distributed Clocks

□ Enable Synchronisation

Shift Time (µs): 200

If an EtherCAT slave is added in the tree view under I/O devices in TwinCAT, the EtherCAT slave is displayed with all list entries and the corresponding variables. These include variables for the inputs and outputs and variables, which can be used for diagnostic purposes or which provide additional information about the EtherCAT slave.

No.	Description
1	All input and output variables are arranged under the EtherCAT slave. The variables can be linked and processed with the PLC program.
2	Process variables can be created under the inputs and outputs of the EtherCAT slave, which can then be exchanged cyclically.
3	InfoData contains input variables, which provide information about the EtherCAT slave that only changes rarely.
4	Further settings for the EtherCAT slave can be implemented under the tabs.

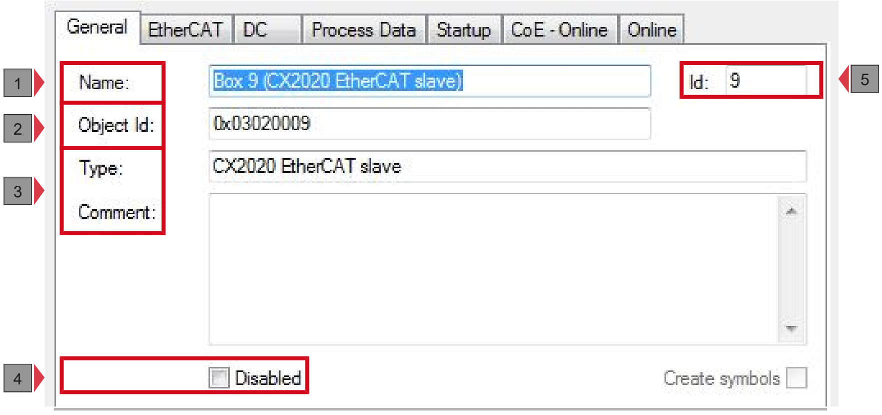
When the PLC process image is read, the inputs and outputs of the EtherCAT slave from the tree view can be linked with the variables of the PLC program. Double-click on a variable name in the tree view to open the link dialog. The linked signals are indicated by a small arrow on the signal symbol.

Further information about TwinCAT can be found in the TwinCAT documentation on the Beckhoff website: www.beckhoff.de

4.2 EtherCAT slave

4.2.1 General

The General tab contains general information for an EtherCAT slave, including name, type and ID.



Name: Box 9 (CX2020 EtherCAT slave)

Id: 9

Object Id: 0x03020009

Type: CX2020 EtherCAT slave

Comment:

□ Disabled

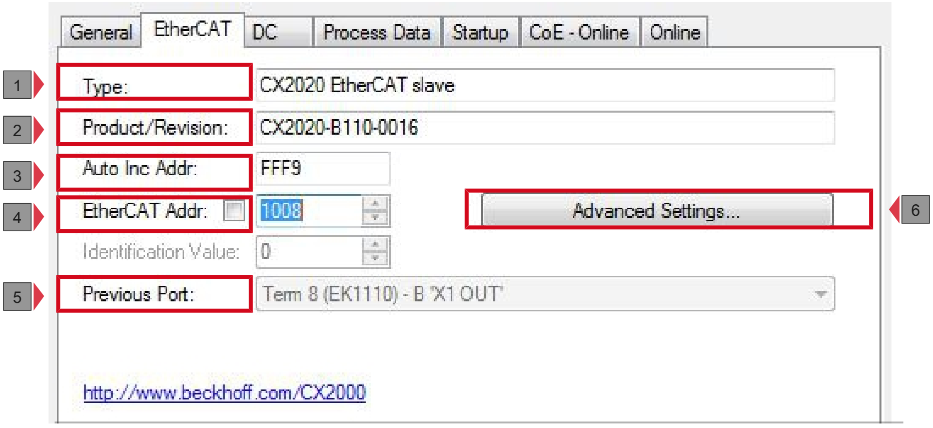
Create symbols □

No.	Description
1	Name of the EtherCAT device
2	EtherCAT device type
3	Here you can add a comment (e.g. notes relating to the system component)
4	Here you can deactivate the EtherCAT device
5	Number of the EtherCAT device

The EtherCAT slave can be switched off via this tab. A comment field is available for describing the EtherCAT slave.

4.2.2 EtherCAT

General information about the EtherCAT slave is displayed on the EtherCAT tab.



Type: CX2020 EtherCAT slave

Product/Revision: CX2020-B110-0016

Auto Inc Addr: FFF9

EtherCAT Addr: □ 1008

Identification Value: 0

Previous Port: Term 8 (EK1110) - B 'X1 OUT'

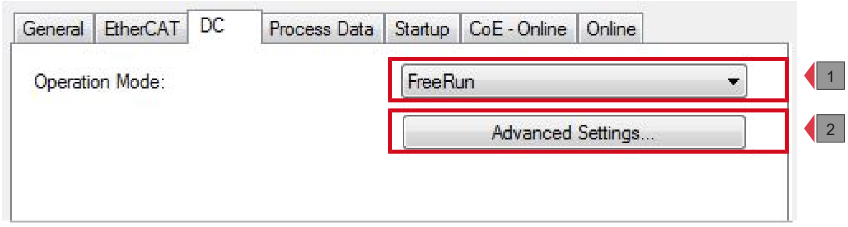
http://www.beckhoff.com/CX2000

No.	Description
1	Name and type of the EtherCAT device.
2	Product and revision number of the EtherCAT device.
3	Auto increment address of the EtherCAT device. The auto increment address can be used for addressing each EtherCAT device in the communication ring through its physical position. Auto increment addressing is used during the start-up phase when the EtherCAT master allocates addresses to the EtherCAT devices. With auto increment addressing the first EtherCAT slave in the ring has the address 0000hex. For each further slave the address is decremented by 1 (FFFFhex, FFFEhex etc.).
4	Fixed address of an EtherCAT slave. This address is allocated by the EtherCAT master during the start-up phase. Click the control box to the left of the input field in order to modify the default value.
5	Name and port of the EtherCAT device, to which the selected device is connected. This combobox is enabled if it is possible to connect this device to another device without changing the order of the EtherCAT devices in the communication ring. You can use it to select the EtherCAT device, to which the selected device is to be connected.
6	The **Advanced Settings** button opens further windows for additional settings.

A link at the bottom of the tab leads to the product page for the respective EtherCAT device.

4.2.3 DC

This tab can be used for the distributed clock settings. The EtherCAT slave can be operated in FreeRun mode or DC Synchronous mode.



Operation Mode: FreeRun

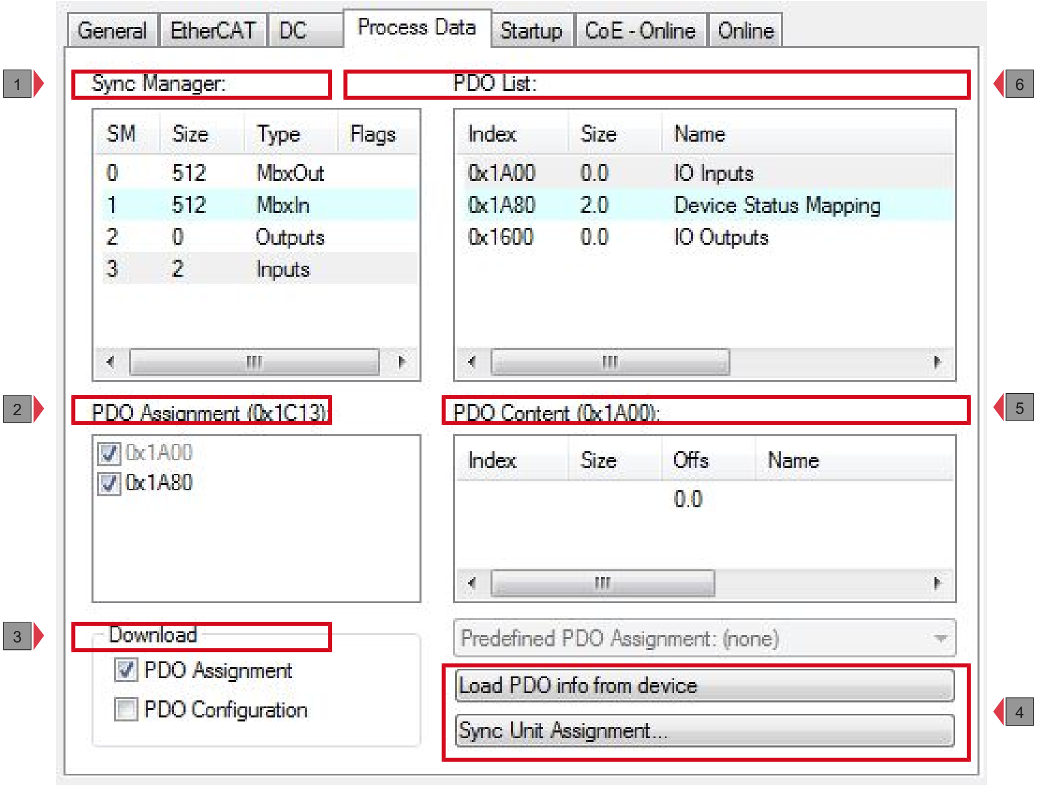
No.	Description
1	Here you can change the settings for the Operation mode. You can choose between FreeRun and DC Synchronous mode.
2	The **Advanced Settings** button opens a further window with additional settings for the distributed clocks function.

EtherCAT slaves feature integrated local clocks, which are synchronized by the EtherCAT master via special datagrams. For sample, the local clock of an EtherCAT slave can be used as reference clock, based on which all other clocks are synchronized.

By default, TwinCAT sets the first EtherCAT slave that supports the distributed clocks function as reference clock. This setting can be changed, although we recommend retaining the default setting.

4.2.4 Process data

This tab indicates the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (PDO).



Sync Manager:

SM	Size	Type	Flags
0	512	MbxOut	
1	512	MbxIn	
2	0	Outputs	
3	2	Inputs	

PDO List:

Index	Size	Name
0x1A00	0.0	IO Inputs
0x1A80	2.0	Device Status Mapping
0x1600	0.0	IO Outputs

PDO Assignment (0x1C13):

☑ 0x1A00
☑ 0x1A80

PDO Content (0x1A00):

Index	Size	Offs	Name
		0.0	

Download

☑ PDO Assignment
☐ PDO Configuration

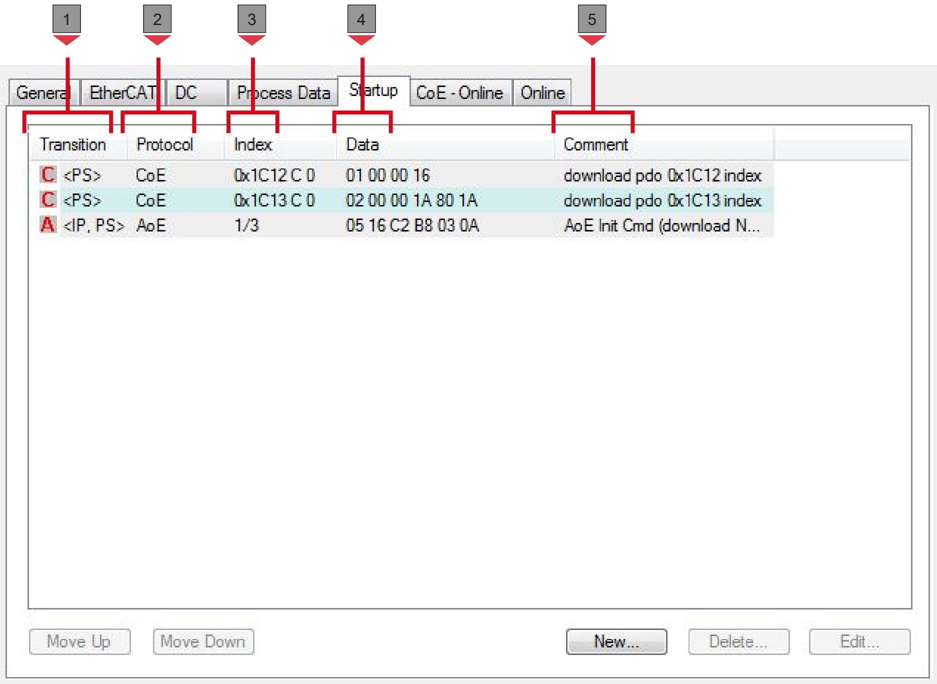
Predefined PDO Assignment: (none)

No.	Description
1	Lists the configuration of the sync manager (SM). If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn). SM2 is used for the output process data (outputs) and SM3 for the input process data (inputs). If an input is selected, the corresponding PDO assignment is displayed in the PDO Assignment list below.
2	PDO assignment of the selected Sync Manager. All PDOs defined for this Sync Manager type are listed here: • If the output Sync Manager (outputs) is selected in the Sync Manager list, all RxPDOs are displayed. • If the input Sync Manager (inputs) is selected in the Sync Manager list, all TxPDOs are displayed. The selected entries are the PDOs involved in the process data transfer. In the tree diagram of the System Manager these PDOs are displayed as variables of the EtherCAT device. The name of the variable is identical to the Name parameter of the PDO, as displayed in the PDO list. If an entry in the PDO assignment list is deactivated (not selected and greyed out), this indicates that the input is excluded from the PDO assignment. In order to be able to select a greyed out PDO, the currently selected PDO has to be deselected first.
3	If the device is intelligent and has a mailbox, the configuration of the PDO and the PDO assignments can be downloaded to the device. This is an optional feature that is not supported by all EtherCAT slaves. **PDO Assignment:** If this checkbox is selected, the assignment, which was configured in the PDO assignment list, can be loaded onto the device on startup. The commands required for this purpose, which have to be sent to the device, are listed on the Startup tab. **PDO configuration:** If this checkbox is selected, the configuration for the respective PDO is loaded onto the EtherCAT slave.
4	If the device is intelligent and features a mailbox, the PDO information can be downloaded from device. This is an optional feature that is not supported by all EtherCAT slaves. The Sync Unit Assignment function can be set here.
5	Indicates the content of the PDO. If flag F (fixed content) of the PDO is not set the content can be modified.
6	List of all PDOs supported by this EtherCAT device. The content of the selected PDOs is displayed below the PDO Content list. The PDO configuration can be modified by double-clicking on an entry. **Index:** PDO index. **Size:** Size of the PDO in bytes. **Name:** Name of the PDO. If this PDO is assigned to a Sync Manager, it appears as a variable of the slave with this parameter as the name. **Flags:** (F) Fixed content: The content of this PDO is fixed and cannot be changed by the System Manager. (M) Mandatory PDO. This PDO is mandatory and must therefore be assigned to a Sync Manager. Consequently, this PDO cannot be deleted from the PDO Assignment list **SM:** Sync Manager to which this PDO is assigned. If this entry is empty, this PDO does not take part in the process data traffic. **SU:** Sync unit to which this PDO is assigned.

4.2.5 Startup

The Startup tab is displayed if the EtherCAT slave has a mailbox and supports the CANopen over EtherCAT (CoE) or Servo drive over EtherCAT protocol.

This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the slave in the same order as they are shown in the list.

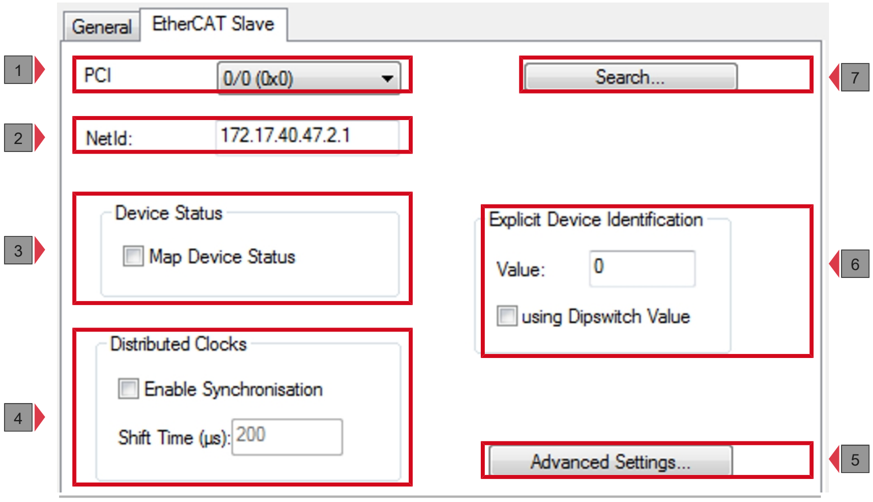


Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1C12 C 0	01 00 00 16	download pdo 0x1C12 index
C <PS>	CoE	0x1C13 C 0	02 00 00 1A 80 1A	download pdo 0x1C13 index
A <IP, PS>	AoE	1/3	05 16 C2 B8 03 0A	AoE Init Cmd (download N...

No.	Description
1	Transition to which the request is sent. This can either be • the transition from pre-operational to safe-operational (PS), or • the transition from safe-operational to operational (SO). If the transition is enclosed in "<>" (e.g. <PS>), the mailbox request is fixed and cannot be modified or deleted by the user.
2	Type of mailbox protocol.
3	Index of the object.
4	Data to be downloaded for this object.
5	Description of the request to be sent to the mailbox.

4.2.6 EtherCAT slave (slave only)

The EtherCAT Slave tab provides information such as the NetId of the EtherCAT slave. The functions Device State, Distributed Clocks and Explicit Device Identification can be activated on this tab. The Advanced Settings for the EtherCAT slave can also be accessed via this tab.



PCI: 0/0 (0x0)

NetId: 172.17.40.47.2.1

Device Status

□ Map Device Status

Distributed Clocks

□ Enable Synchronisation

Shift Time (µs): 200

Explicit Device Identification

Value: 0

□ using Dipswitch Value

No.	Description
1	Physical interface.
2	NetId of the EtherCAT device. The NetId is required for the ADS communication with the EtherCAT device.
3	If the **Map Device Status** checkbox is enabled, additional inputs and outputs are added in the tree view. This function can be used for EtherCAT slave diagnostics.
4	If the **Enable Synchronization** checkbox is enabled, the distributed clocks function is enabled.
5	The **Advanced Settings** button opens a further window with additional EtherCAT slave settings.
6	At this point an identification number for the EtherCAT slave can be allocated, so that it can be swapped with another EtherCAT slave during operation ("hot connect"). Ensures unambiguous identification of a device. In case of exchange the current address must be stored again in the replacement device.
7	This button can be used to find and link physical interfaces, if not already done automatically.

5 Parameterization and commissioning

The EtherCAT slave can be configured with TwinCAT 2 or TwinCAT 3. The following sections describe the EtherCAT state machine and how to insert an EtherCAT slave in TwinCAT, create variables and activate the device status and the device identification.

5.1 EtherCAT State Machine

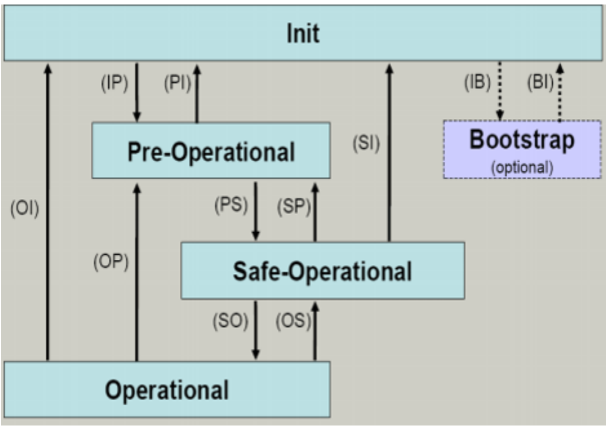
EtherCAT State Machine

The state of the EtherCAT slave is controlled via the EtherCAT State Machine (ESM). Depending upon the state, different functions are accessible or executable in the EtherCAT slave. Specific commands must be sent by the EtherCAT master to the device in each state, particularly during the bootup of the slave.

A distinction is made between the following states:

- Init
- Pre-Operational
- Safe-Operational and
- Operational
- Boot

The regular state of each EtherCAT slave after bootup is the OP state.



Init
After switch-on the EtherCAT slave in the Init state. No mailbox or process data communication is possible. The EtherCAT master initializes sync manager channels 0 and 1 for mailbox communication.

Pre-Operational (Pre-Op)
During the transition from Init to Pre-Op, the EtherCAT slave checks whether the mailbox has been correctly initialized. Mailbox communication is possible in the Pre-Op state, but no process data communication. The EtherCAT master initializes the sync manager channels for process data (from sync manager channel 2), the FMMU channels and, if the slave supports configurable mapping, PDO mapping or the sync manager PDO assignment. In this state the settings for the process data transfer and perhaps terminal-specific parameters that may differ from the default settings are also transferred.

Safe-Operational (Safe-Op)
During transition between Pre-Op and Safe-Op the EtherCAT slave checks whether the sync manager channels for process data communication and, if required, the distributed clocks settings are correct. Before it acknowledges the change of state, the EtherCAT slave copies current input data into the associated DP-RAM areas of the EtherCAT slave controller (ECSC).

Mailbox and process data communication are possible in the Safe-OP state; however, the slave maintains its outputs in the safe state and does not output them. The input data, however, are already cyclically updated.

Operational (Op)
Before the EtherCAT master switches the EtherCAT Slave from *Safe-Op* to *Op*, it must transmit already valid output data. In the *Op* state the slave copies the output data of the master to its outputs. Process data and mailbox communication is possible.

Boot
In the *Boot* state the slave firmware can be updated. The *Boot* state can only be reached via the *Init* state. In the *Boot* state mailbox communication is possible via the *File-Access over EtherCAT* (FoE) protocol, but no other mailbox communication and no process data communication is possible.

5.2 Parameterization with TwinCAT 2

This section illustrates how you can parameterize the EtherCAT slave option interface using TwinCAT 2.

First, the process of finding and selecting a target system in TwinCAT is illustrated. Next, an EtherCAT slave is added to TwinCAT. A small PLC project is then created and added to TwinCAT. The variables from the PLC project are then linked to the hardware and the finished configuration is loaded onto the device.

5.2.1 Searching for target systems

Before you can work with the devices, you must connect your local computer to the target device. Then you can search for the devices with the help of the IP address or the host name.

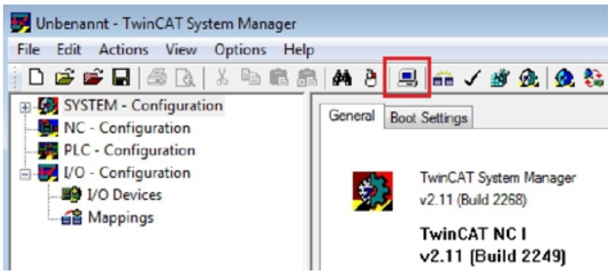
The local PC and the target devices must be connected to the same network or directly to each other via an Ethernet cable. In TwinCAT a search can be performed for all devices in this way and project planning subsequently carried out.

Prerequisites for this step:

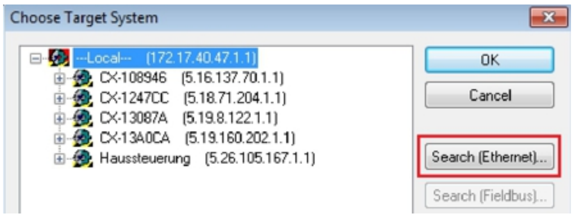
- TwinCAT 2 must be in Config mode.
- IP address or host name of the device. The host name is composed of CX- and the last 3 bytes of the MAC address. The MAC address is located on the side of the device.

Search for the devices as follows:

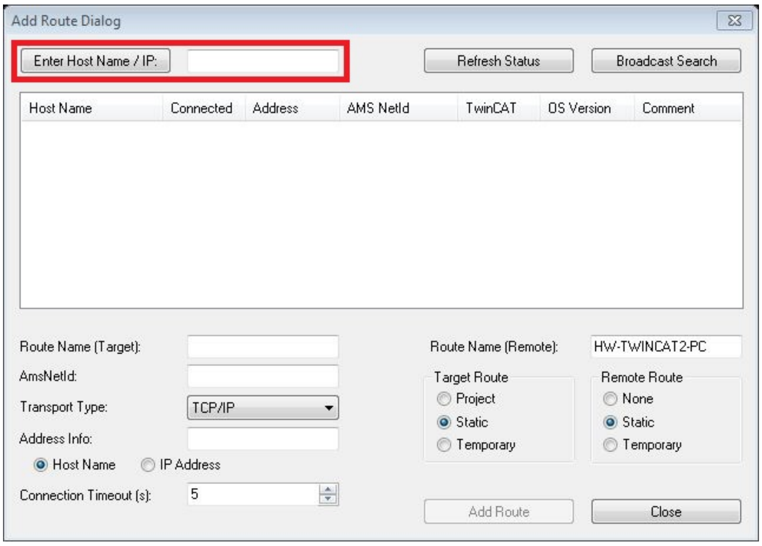
1. Click on **File > New** in the menu at the top.
2. Click on **Choose Target System** in the toolbar at the top.



3. Click on **Search (Ethernet)**.

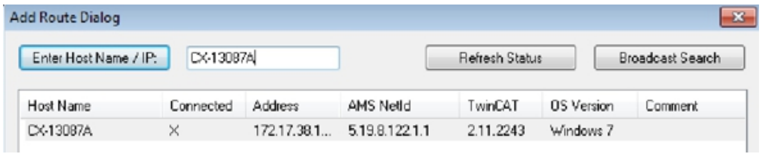

Choose Target System
--Local-- (172.17.40.47.1.1)
CX-108946 (5.16.137.70.1.1)
CX-1247CC (5.18.71.204.1.1)
CX-13087A (5.19.8.122.1.1)
CX-13A0CA (5.19.160.202.1.1)
Haussteuerung (5.26.105.167.1.1)
OK
Cancel
Search (Ethernet)...
Search (Fieldbus)...

4. Type the host name or the IP address of the device into the **Enter Host Name / IP** box and press **[Enter]**.



Route Name (Remote): HW-TWINCAT2-PC
Transport Type: TCP/IP
● Host Name ○ IP Address
Connection Timeout (s): 5
Target Route: ○ Project ● Static ○ Temporary
Remote Route: ○ None ● Static ○ Temporary

5. Mark the device found and click on **Add Route**.

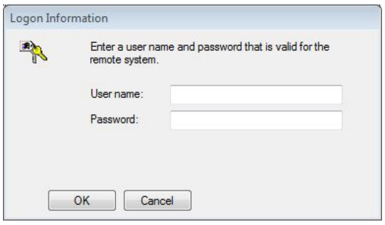


Enter Host Name / IP: CX-13087A

Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment
CX-13087A	X	172.17.38.1...	5.19.8.122.1.1	2.11.2243	Windows 7	

The Logon Information window appears.

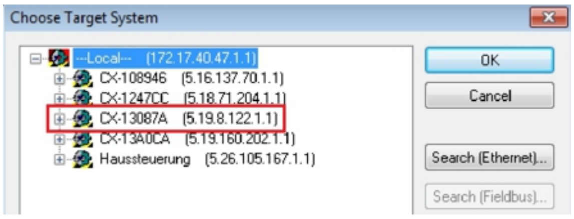
6. Enter the user name and password for the CX in the **User Name** and **Password** fields and click **OK**.



Enter a user name and password that is valid for the remote system.

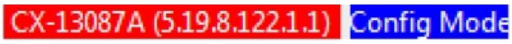
The following information is set as standard in CX devices:
User name: Administrator **Password:** 1

7. If you do not wish to search for any further devices, click on **Close** to close the Add Route Dialog. The new device is displayed in the Choose Target System window.
8. Mark the device that you wish to set as the target system and click on **OK**.



--Local-- (172.17.40.47.1.1)
CX-108946 (5.16.137.70.1.1)
CX-1247CC (5.18.71.204.1.1)
CX-13087A (5.19.8.122.1.1)
CX-13A0CA (5.19.160.202.1.1)
Haussteuerung (5.26.105.167.1.1)

⇨ You have successfully searched for a device in TwinCAT and inserted the device as the target system. The new target system is displayed in the bottom right-hand corner together with the host name and IP address (AMS Net ID).



CX-13087A (5.19.8.122.1.1) Config Mode

Using this procedure you can search for all available devices and also switch between the target systems at any time. Next, you can append the device to the tree view in TwinCAT.

5.2.2 Appending an EtherCAT device

Once you have selected an EtherCAT device as target system, you can add the EtherCAT device in the tree view in TwinCAT.

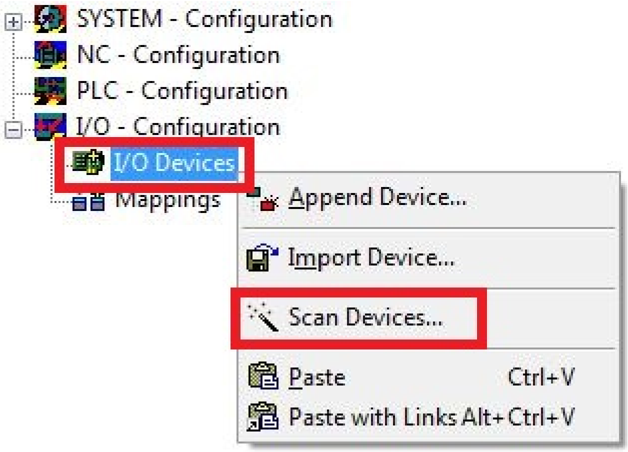
These operations can be used to add an EtherCAT master or EtherCAT slave in the tree view.

Prerequisites for this step:

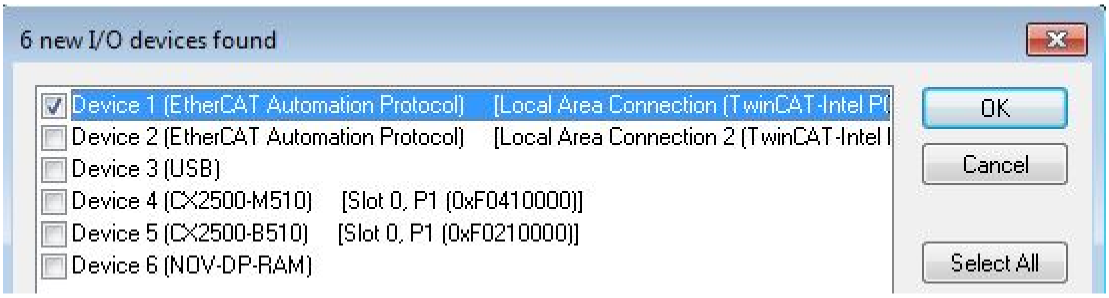
- A scanned and selected target device.

Add an EtherCAT device as follows:

1. In the tree view on the left, right-click on Devices.
2. In the context menu click on Scan.



3. Select the devices you want to use and confirm the selection with OK.


6 new I/O devices found

☑ Device 1 (EtherCAT Automation Protocol) [Local Area Connection (TwinCAT-Intel P(
□ Device 2 (EtherCAT Automation Protocol) [Local Area Connection 2 (TwinCAT-Intel I
□ Device 3 (USB)
□ Device 4 (CX2500-M510) [Slot 0, P1 (0xF0410000)]
□ Device 5 (CX2500-B510) [Slot 0, P1 (0xF0210000)]
□ Device 6 (NOV-DP-RAM)

4. Confirm the request with Yes, in order to look for boxes.

⇨ The EtherCAT device was added successfully in the tree view.

5.2.3 Creating variables

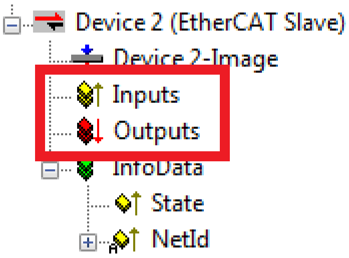
The EtherCAT slave has a simple structure and consists of inputs, outputs and InfoData. In TwinCAT you can add process data to the inputs or outputs, which can later be linked to variables from the PLC program. This section describes how to create such data.

Prerequisites for this step:

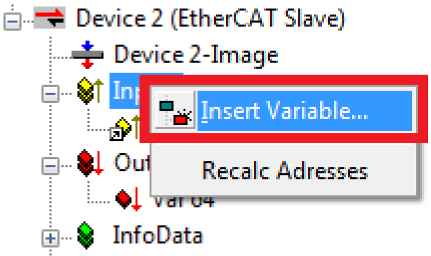
- A device with EtherCAT slave interface (B110) must already have been added as target system in TwinCAT.

Create the process data as follows:

1. In the tree view, right-click on the inputs or outputs.

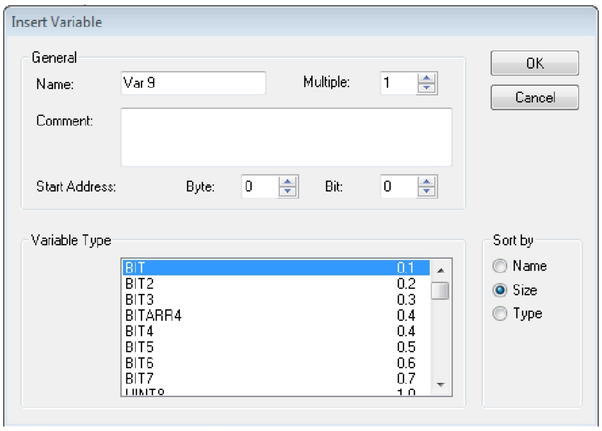


2. In the context menu click on **Insert Variable**.



The Insert Variable window appears.

3. Find the required variable type and click **OK**.



Name: Var 9

Multiple: 1

Comment:

Start Address: Byte: 0 Bit: 0

BIT	0.1
BIT2	0.2
BIT3	0.3
BITARR4	0.4
BIT4	0.4
BIT5	0.5
BIT6	0.6
BIT7	0.7
UINT8	1.0

○ Name
● Size
○ Type

⇨ You have successfully created process data. The new variable is shown in the tree view on the left. In this way you can add further process data or variables for the EtherCAT slave.

Process data with structures

In order to save a large number of links it is a good idea to use a data structure to save data that you wish to exchange. Note that data structures with different variables are processed differently on an x86 system and an ARM processor. An ARM processor always associates WORD (2 byte) variables with an even address and DWORD (4 byte) variables with an address that is divisible by 4.

Data structure sample
byTest: BYTE;
udTest: UDINT;

ARM address	ARM variable	Address x86	ARM variable
Byte Offset 0	Byte	Byte Offset 0	BYTE
Byte Offset 4	UDINT	Byte Offset 1	UDINT
Sum: 8 byte		Sum: 5 byte	

You can determine the length of a data structure on both systems using the command SIZEOF. If there is a difference here, this indicates that something is wrong with the data structure.

This problem can be solved by more skillful arrangement of the variables or by working with filler or dummy variables.

ARM address	ARM variable	Address x86	ARM variable
Byte Offset 0	Byte	Byte Offset 0	BYTE
		Byte Offset 1	BYTE (Dummy1)
		Byte Offset 2	BYTE (Dummy2)
		Byte Offset 3	BYTE (Dummy3)
Byte Offset 4	UDINT	Byte Offset 4	UDINT
Sum: 8 byte		Sum: 8 byte	

Filler or dummy variables can be used to adjust the length of the data structure, in order to avoid problems caused by potential differences.

5.2.4 Enable device status

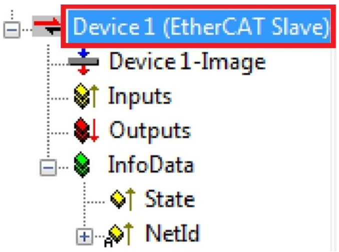
The device status function indicates whether the EtherCAT slave is in operation and exchanges data. The function can be used for diagnostic purposes and provides important information about the EtherCAT slave. The device status is disabled by default and has to be enabled in TwinCAT, before the slave status can be displayed.

Prerequisites for this step:

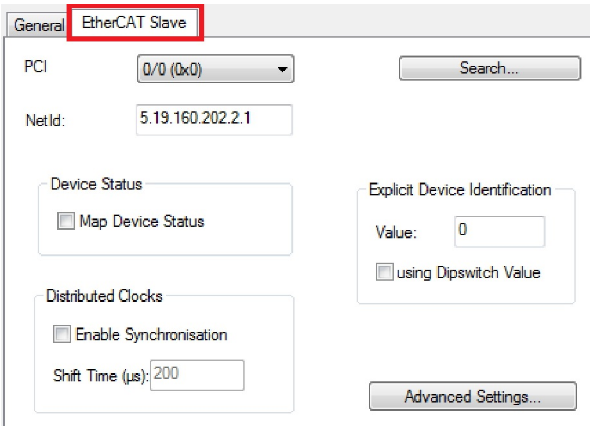
- A device with EtherCAT slave interface (B110) must already have been added as target system in TwinCAT.

Enable the device status as follows:

1. In the tree view click on the EtherCAT slave.



2. Click on the **EtherCAT Slave** tab.



PCI: 0/0 (0x0)

NetId: 5.19.160.202.2.1

Device Status

☐ Map Device Status

Explicit Device Identification

Value: 0

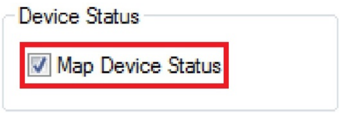
☐ using Dipswitch Value

Distributed Clocks

☐ Enable Synchronisation

Shift Time (µs): 200

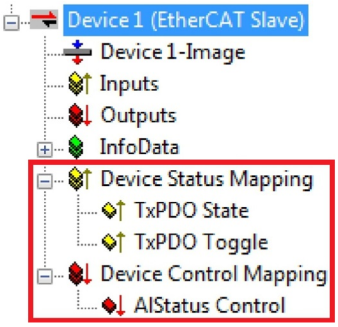
3. Under **Device Status** enable the function **Map Device Status**.



Device Status

☑ Map Device Status

⇨ You have successfully enabled the device status function. The **Device Status Mapping** inputs and the **Device Control Mapping** output are added to the tree view.



In the next step you can enable the device identification.

5.2.5 Enable device identification

The EtherCAT Hot Connect functionality enables preconfigured sections to be removed from or added to the data traffic before startup or during operation. This can be done by disconnecting/connecting the communication path, turning the device on/off or other measures.

This function is particularly advantageous for applications in which frequent topology changes are required.

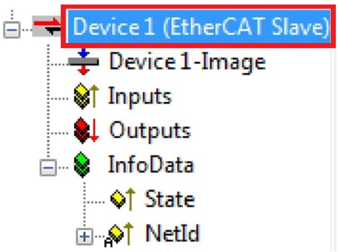
Use the Explicit Device Identification function to enable the Hot Connect functionality for the EtherCAT slave and allocate an identification number for the EtherCAT slave. In case of exchange the current address must be stored again in the replacement device. The identification number is set via TwinCAT.

Prerequisites for this step:

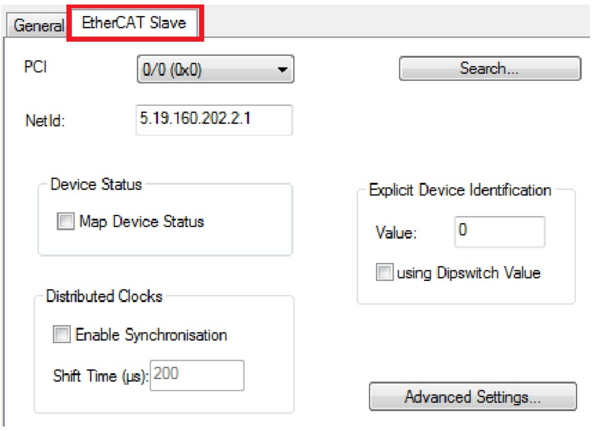
- An EtherCAT slave added in TwinCAT.

Enable the device identification as follows:

1. In the tree view click on the EtherCAT slave.



2. Click on the **EtherCAT Slave** tab



PCI: 0/0 (0x0)

NetId: 5.19.160.202.2.1

Device Status

□ Map Device Status

Distributed Clocks

□ Enable Synchronisation

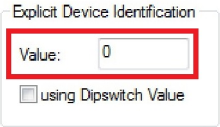
Shift Time (µs): 200

Explicit Device Identification

Value: 0

□ using Dipswitch Value

3. Enter an identification number for your EtherCAT slave under **Explicit Device Identification** in the **Value** field.



Explicit Device Identification

Value: 0

□ using Dipswitch Value

⇨ You have successfully enabled the device identification for an EtherCAT slave and allocated an identification number. The individual EtherCAT slave devices can now be distinguished from each other and swapped with devices with the same identification number during operation.

5.2.6 Creating a PLC project

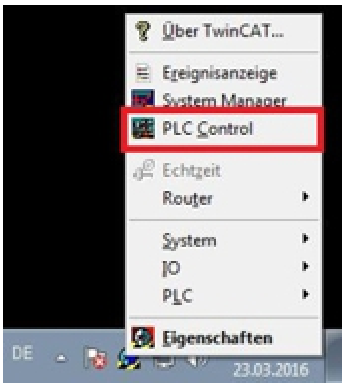
Use PLC Control to create a PLC project. The next steps describe how to create a PLC project in TwinCAT and add it in the tree view.

Prerequisites for this step:

- An Embedded PC, added in TwinCAT.

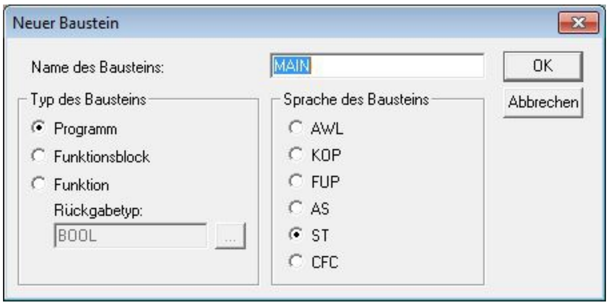
Create a PLC project as follows:

1. In the Start menu, right-click on the TwinCAT symbol.
2. In the context menu click on **PLC Control**.



The TwinCAT PLC Control window appears.

3. In the menu click on **File > New** and select the option **PC or CX (x86)**.
4. Under **Block type** select the option **Program**, and under block language select the option **ST (Structured Text)**.

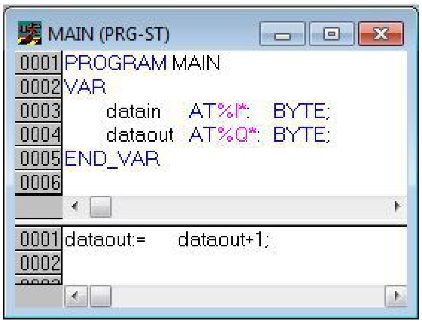


Name des Bausteins: MAIN

Typ des Bausteins
● Programm
○ Funktionsblock
○ Funktion
Rückgabetyp: BOOL

Sprache des Bausteins
○ AWL
○ KOP
○ FUP
○ AS
● ST
○ CFC

5. Write a small program.



```
PROGRAM MAIN
VAR
      datain   AT%I*:  BYTE;
      dataout  AT%Q*:  BYTE;
END_VAR

dataout:=     dataout+1;
```

6. Save the PLC project and click on **Project > Compile** in the menu.

⇨ Once the project has been compiled, a file with the extension .tpy is created in the same location as the project file. The file name of the new file is the same as the file name of the PLC project.

In the next step you can add the compiled PLC project in the TwinCAT System Manager.

Adding a PLC project

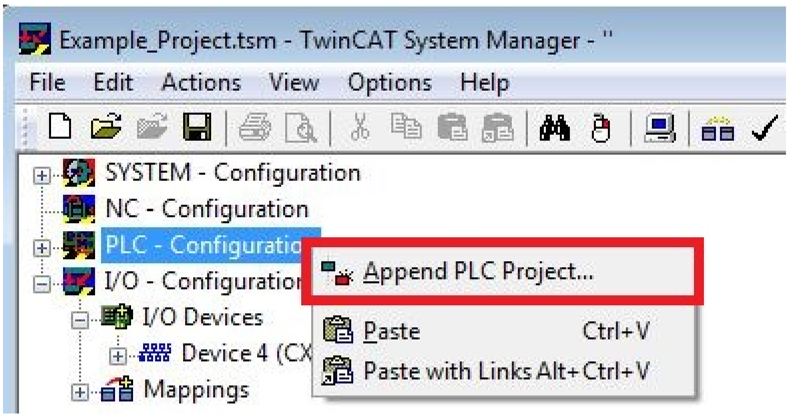
The PLC project can be added in the System Manager. The newly created variables from a PLC project are integrated in the System Manager and can be linked with the inputs and outputs of the hardware.

Prerequisites for this step:

- An Embedded PC, added in TwinCAT.
- A correctly compiled PLC project and a .tpy file.

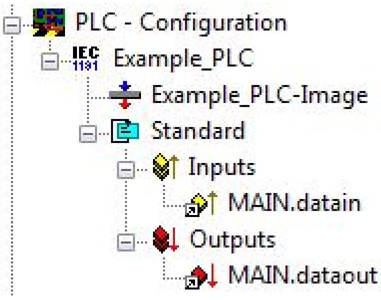
Proceed as follows:

1. Switch back to the System Manager window.
2. Right-click on **PLC – Configuration** in the tree view on the left.
3. In the context menu click on **Append PLC Project**.



4. Select a file with the extension .tpy in your system directory and confirm with **OK**.

The PLC project is added in the tree view under PLC – Configuration. The variables defined in the project are shown under the inputs and outputs.



In the next step you can link the variables with the hardware.

5.2.7 Linking variables

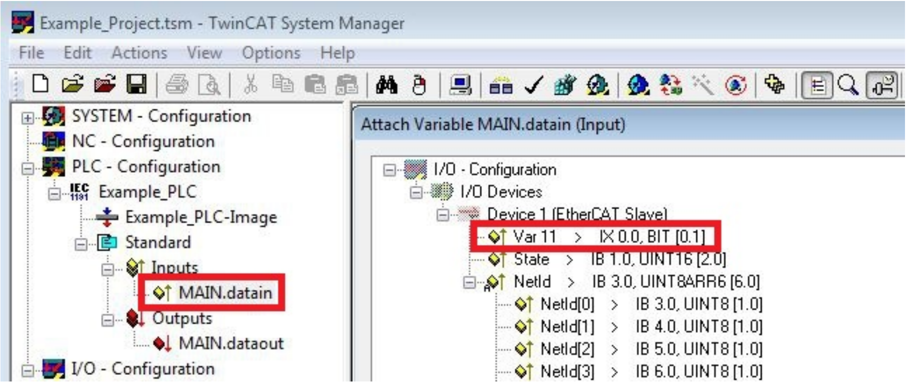
Once the PLC project was successfully added in the System Manager, you can link the newly created input and output variables from the PLC project with the inputs and outputs of your devices.

Prerequisites for this step:

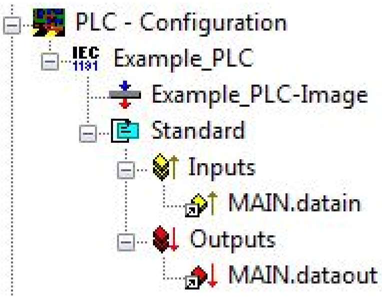
- An added PLC project in the System Manager.

Link the variables as follows:

1. Double-click on the input or output variables in the tree view under **PLC - Configuration**.
   The **Attach Variable** window appears and shows which inputs or outputs can be linked with variables.

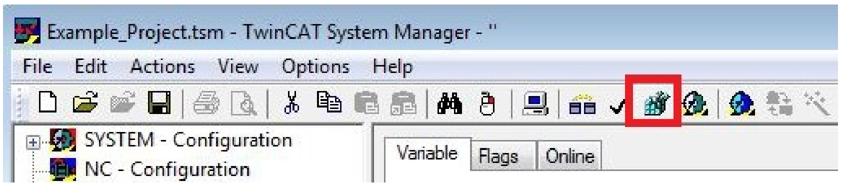

Example_Project.tsm - TwinCAT System Manager
File Edit Actions View Options Help
SYSTEM - Configuration
NC - Configuration
PLC - Configuration
Example_PLC
Example_PLC-Image
Standard
Inputs
MAIN.datain
Outputs
MAIN.dataout
I/O - Configuration
Attach Variable MAIN.datain (Input)
I/O - Configuration
I/O Devices
Device 1 (EtherCAT Slave)
Var 11 > IX 0.0, BIT [0.1]
State > IB 1.0, UINT16 [2.0]
NetId > IB 3.0, UINT8ARR6 [6.0]
NetId[0] > IB 3.0, UINT8 [1.0]
NetId[1] > IB 4.0, UINT8 [1.0]
NetId[2] > IB 5.0, UINT8 [1.0]
NetId[3] > IB 6.0, UINT8 [1.0]

2. Double-click on the inputs or outputs in the Attach Variable window.
   The input variables are linked with the inputs of your hardware, and the output variables with the outputs.



   Variables that are already linked are indicated with a small arrow icon in TwinCAT.

3. In the toolbar click on **Activate Configuration**.


Example_Project.tsm - TwinCAT System Manager - ''
File Edit Actions View Options Help
SYSTEM - Configuration
NC - Configuration
Variable
Flags
Online

4. Confirm the request whether TwinCAT is to start in Free Run mode with **Yes**.

⇨ You have successfully linked variables with the hardware. Use Activate Configuration to save and activate the current configuration.

The configuration can now be loaded on the CX, in order to automatically start TwinCAT in Run mode, followed by the PLC project.

5.2.8 Load configuration to CX

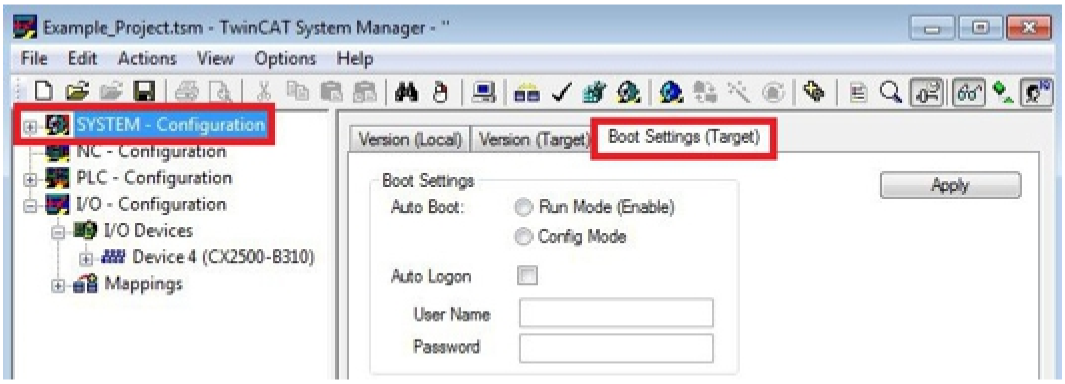
Once all variables are linked, the configuration can be saved and loaded on the CX. This has the advantage that the PLC project is loaded and started automatically when the CX is switched on. The start of the previously created PLC project can thus be automated.

Prerequisites for this step:

- A completed PLC project, added in the System Manager.
- Variables from the PLC project, linked with the hardware in the System Manager.
- A CX selected as target system.

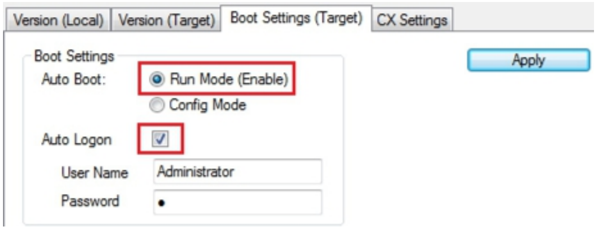
Load the configuration on the CX as follows:

1. In the tree view on the left click on **SYSTEM – Configuration**.
2. Click on the **Boot Settings (Target)** tab.



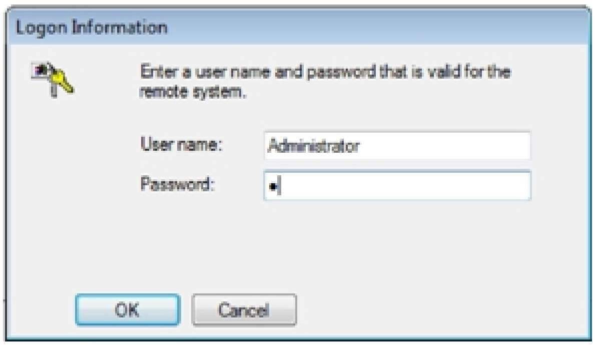
Boot Settings
Auto Boot: ○ Run Mode (Enable) ○ Config Mode
Auto Logon □
User Name
Password

3. Under Boot Settings select the option **Run Mode (Enable)** and tick the **Auto Logon** checkbox.



Boot Settings
Auto Boot: ● Run Mode (Enable) ○ Config Mode
Auto Logon ☑
User Name Administrator
Password •

4. Enter the user name and password for the CX in the **User Name** and **Password** fields.
5. Click on **Apply**.
The Logon Information window appears.

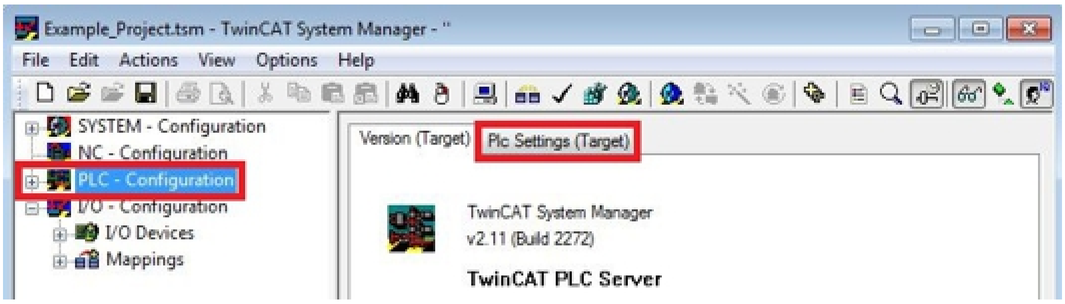


Enter a user name and password that is valid for the remote system.

User name: Administrator
Password: •

6. Re-enter the user name and the password and click **OK**.

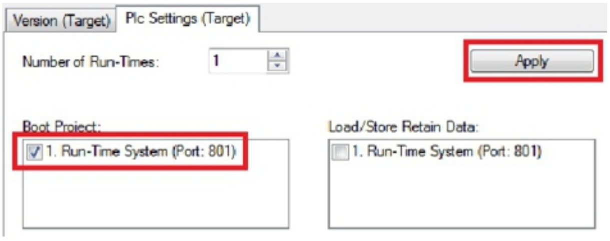
7. In the tree view on the left click on **PLC – Configuration**, then on the **PLC Settings (Target)** tab.



TwinCAT System Manager
v2.11 (Build 2272)

TwinCAT PLC Server

8. Select the Start PLC under Boot Project and click on **Apply**.

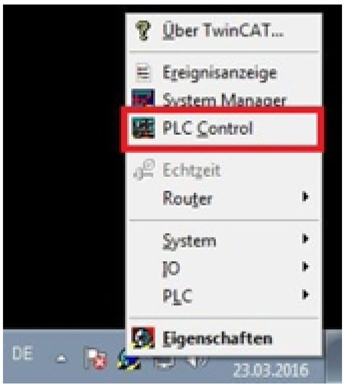


Number of Run-Times: 1

Boot Project:
☑ 1. Run-Time System (Port: 801)

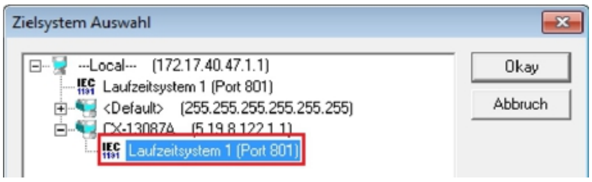
Load/Store Retain Data:
□ 1. Run-Time System (Port: 801)

9. Start PLC Control and open the PLC project.



10. In the menu bar at the top click on **Online**, and then on **Choose Runtime System**.

11. Select the runtime system from the CX and click on **OK**.



--Local-- (172.17.40.47.1.1)
Laufzeitsystem 1 (Port 801)
<Default> (255.255.255.255.255.255)
CX-13087A (5.19.8.122.1.1)
Laufzeitsystem 1 (Port 801)

12. In the menu bar at the top click on **Online**, then **Login**.
    The PLC project is logged in.

13. In the menu bar at the top click on **Online**, then **Create Boot Project**.

⇨ You have successfully loaded the CX configuration. From now on, TwinCAT will start in Run mode and the PLC project will start automatically.

Next, the master can be added in a new project in the System Manager and can then be used to find slaves that have already been set up.

5.3 Parameterization with TwinCAT 3

This section illustrates how you can parameterize the EtherCAT slave option interface using TwinCAT 3.

First, the process of finding and selecting a target system in TwinCAT is illustrated. Next, an EtherCAT slave is added to TwinCAT. A small PLC project is then created and added to TwinCAT. The variables from the PLC project are then linked to the hardware and the finished configuration is loaded onto the device.

5.3.1 Searching for target systems

Before you can work with the devices, you must connect your local computer to the target device. Then you can search for devices with the help of the IP address or the host name.

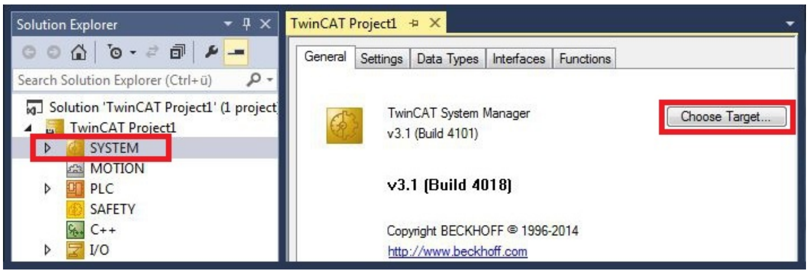
The local PC and the target devices must be connected to the same network or directly to each other via an Ethernet cable. In TwinCAT a search can be performed for all devices in this way and project planning subsequently carried out.

Prerequisites for this step:

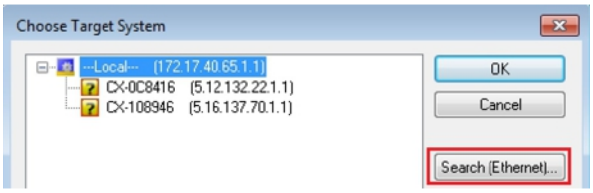
- TwinCAT 3 must be in Config mode.
- IP address or host name of the device.

Search for the devices as follows:

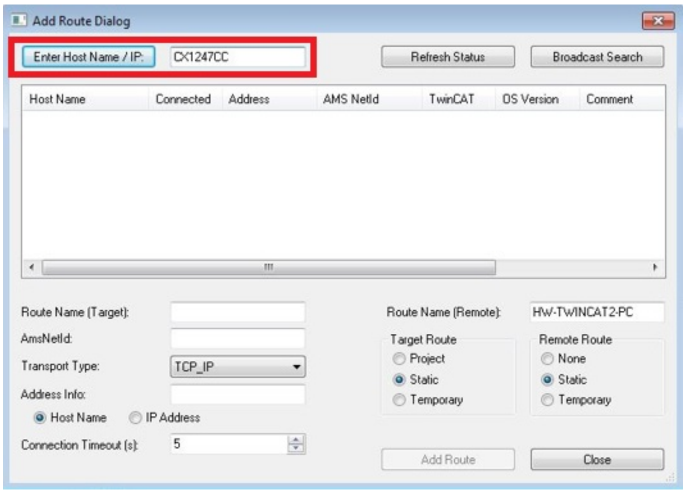
1. In the menu at the top click on **File > New > Project** and create a new TwinCAT XAE project.
2. In the tree view on the left click on **SYSTEM**, and then **Choose Target**.



3. Click on **Search (Ethernet)**.


Choose Target System
--Local-- (172.17.40.65.1.1)
CX-0C8416 (5.12.132.22.1.1)
CX-108946 (5.16.137.70.1.1)
OK
Cancel
Search (Ethernet)...

4. Type the host name or the IP address of the device into the **Enter Host Name / IP** box and press **[Enter]**.



Enter Host Name / IP: CX1247CC

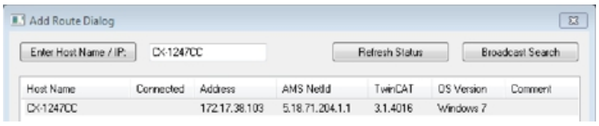
Route Name (Target):
AmsNetId:
Transport Type: TCP_IP
Address Info:
● Host Name ○ IP Address
Connection Timeout (s): 5

Route Name (Remote): HW-TWINCAT2-PC

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

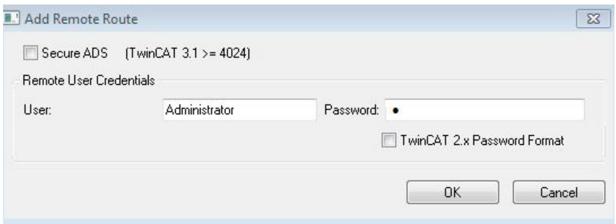
5. Mark the device found and click on **Add Route**.



Enter Host Name / IP: CX-1247CC

Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment
CX-1247CC		172.17.38.103	5.18.71.204.1.1	3.1.4016	Windows 7	

The Logon Information window appears.
Enter the user name and password for the CX in the **User Name** and **Password** fields and click **OK**.



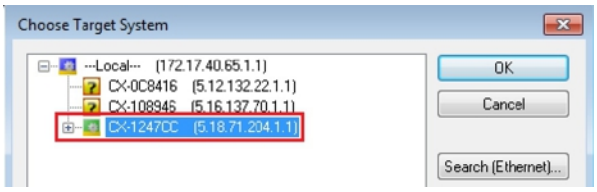
□ Secure ADS (TwinCAT 3.1 >= 4024)

User: Administrator
Password: •
□ TwinCAT 2.x Password Format

The following information is set as standard in CX devices:
User name: Administrator **Password:** 1

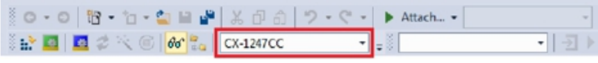
6. If you do not wish to search for any further devices, click on **Close** to close the Add Route Dialog.
The new device is displayed in the Choose Target System window.

7. Select the device you want to specify as target system and click **OK**.



---Local--- (172.17.40.65.1.1)
CX-0C8416 (5.12.132.22.1.1)
CX-108946 (5.16.137.70.1.1)
CX-1247CC (5.18.71.204.1.1)

⇨ You have successfully searched for a device in TwinCAT and inserted the device as the target system.
The new target system and the host name are displayed in the menu bar.



Using this procedure you can search for all available devices and also switch between the target systems at any time. Next, you can append the device to the tree view in TwinCAT.

5.3.2 Appending an EtherCAT device

Once you have selected an EtherCAT device as target system, you can add the EtherCAT device in the tree view in TwinCAT.

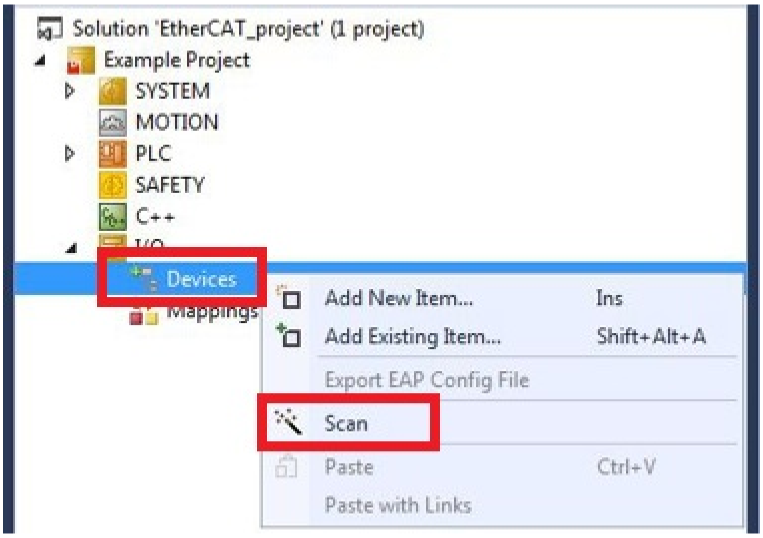
These operations can be used to add an EtherCAT master or EtherCAT slave in the tree view.

Prerequisites for this step:

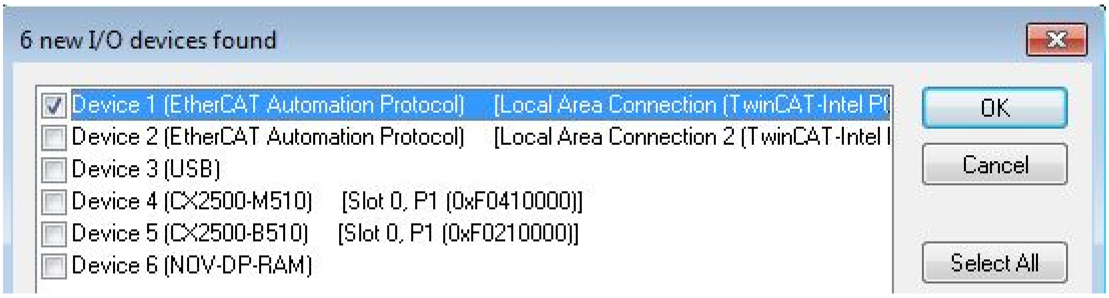
- A scanned and selected target device.

Add an EtherCAT device as follows:

1. In the tree view on the left, right-click on Devices.
2. In the context menu click on Scan.



3. Select the devices you want to use and confirm the selection with OK.


6 new I/O devices found

☑ Device 1 (EtherCAT Automation Protocol) [Local Area Connection (TwinCAT-Intel PC
□ Device 2 (EtherCAT Automation Protocol) [Local Area Connection 2 (TwinCAT-Intel I
□ Device 3 (USB)
□ Device 4 (CX2500-M510) [Slot 0, P1 (0xF0410000)]
□ Device 5 (CX2500-B510) [Slot 0, P1 (0xF0210000)]
□ Device 6 (NOV-DP-RAM)

4. Confirm the request with Yes, in order to look for boxes.

⇨ The EtherCAT device was added successfully in the tree view.

5.3.3 Creating variables

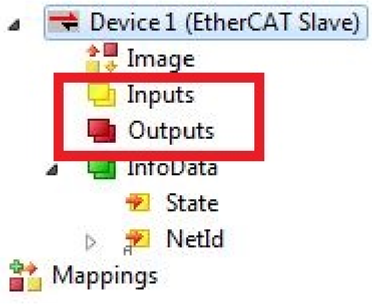
The EtherCAT slave has a simple structure and consists of inputs, outputs and InfoData. In TwinCAT you can add process data to the inputs or outputs, which can later be linked to variables from the PLC program. This section describes how to create such data.

Prerequisites for this step:

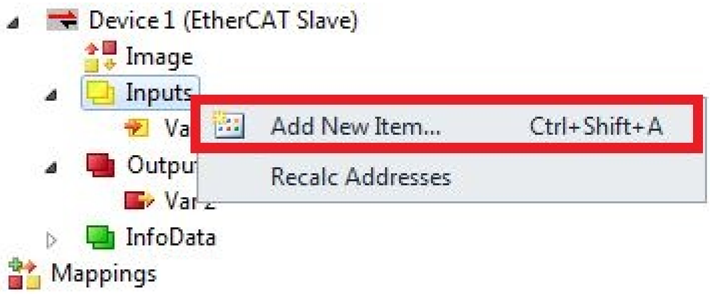
- A device with EtherCAT slave interface (B110) must already have been added as target system in TwinCAT.

Create the process data as follows:

1. In the tree view, right-click on the inputs or outputs.

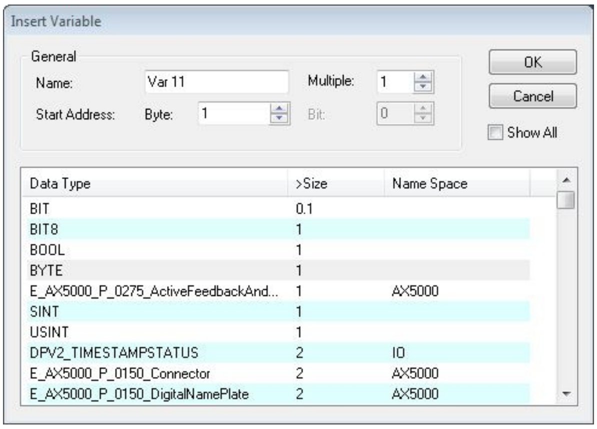


2. Click on **Add New Item** in the context menu.



The Insert Variable window appears.

3. Find the required variable type and click **OK**.



Name: Var 11
Multiple: 1
Start Address: Byte: 1
Bit: 0

Data Type	>Size	Name Space
BIT	0.1	
BIT8	1	
BOOL	1	
BYTE	1	
E_AX5000_P_0275_ActiveFeedbackAnd...	1	AX5000
SINT	1	
USINT	1	
DPV2_TIMESTAMPSTATUS	2	IO
E_AX5000_P_0150_Connector	2	AX5000
E_AX5000_P_0150_DigitalNamePlate	2	AX5000

⇨ You have successfully created process data. The new variable is shown in the tree view on the left. In this way you can add further process data or variables for the EtherCAT slave.

Process data with structures

In order to save a large number of links it is a good idea to use a data structure to save data that you wish to exchange. Note that data structures with different variables are processed differently on an x86 system and an ARM processor. An ARM processor always associates WORD (2 byte) variables with an even address and DWORD (4 byte) variables with an address that is divisible by 4.

Data structure sample
byTest: BYTE;
udTest: UDINT;

ARM address	ARM variable	Address x86	ARM variable
Byte Offset 0	Byte	Byte Offset 0	BYTE
Byte Offset 4	UDINT	Byte Offset 1	UDINT
Sum: 8 byte		Sum: 5 byte	

You can determine the length of a data structure on both systems using the command SIZEOF. If there is a difference here, this indicates that something is wrong with the data structure.

This problem can be solved by more skillful arrangement of the variables or by working with filler or dummy variables.

ARM address	ARM variable	Address x86	ARM variable
Byte Offset 0	Byte	Byte Offset 0	BYTE
		Byte Offset 1	BYTE (Dummy1)
		Byte Offset 2	BYTE (Dummy2)
		Byte Offset 3	BYTE (Dummy3)
Byte Offset 4	UDINT	Byte Offset 4	UDINT
Sum: 8 byte		Sum: 8 byte	

Filler or dummy variables can be used to adjust the length of the data structure, in order to avoid problems caused by potential differences.

5.3.4 Enable device status

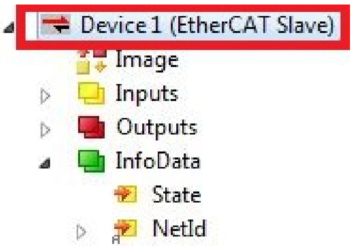
The device status function indicates whether the EtherCAT slave is in operation and exchanges data. The function can be used for diagnostic purposes and provides important information about the EtherCAT slave. The device status is disabled by default and has to be enabled in TwinCAT, before the slave status can be displayed.

Prerequisites for this step:

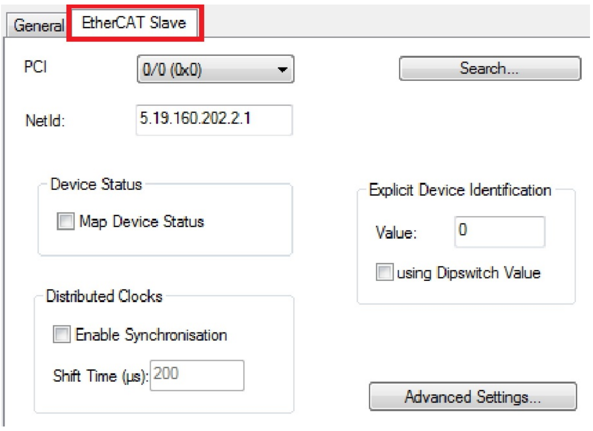
- A device with EtherCAT slave interface (B110) must already have been added as target system in TwinCAT.

Enable the device status as follows:

1. In the tree view click on the EtherCAT slave.



2. Click on the **EtherCAT Slave** tab.



PCI: 0/0 (0x0)

NetId: 5.19.160.202.2.1

□ Map Device Status

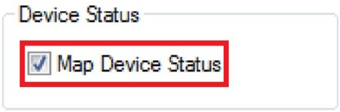
Value: 0

□ using Dipswitch Value

□ Enable Synchronisation

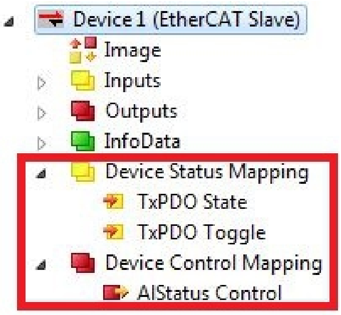
Shift Time (µs): 200

3. Under **Device Status** enable the function **Map Device Status**.



☑ Map Device Status

⇨ You have successfully enabled the device status function. The **Device Status Mapping** inputs and the **Device Control Mapping** output are added to the tree view.



In the next step you can enable the device identification.

5.3.5 Enable device identification

The EtherCAT Hot Connect functionality enables preconfigured sections to be removed from or added to the data traffic before startup or during operation. This can be done by disconnecting/connecting the communication path, turning the device on/off or other measures.

This function is particularly advantageous for applications in which frequent topology changes are required.

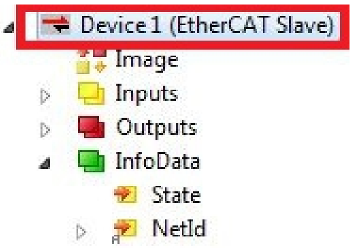
Use the Explicit Device Identification function to enable the Hot Connect functionality for the EtherCAT slave and allocate an identification number for the EtherCAT slave. In case of exchange the current address must be stored again in the replacement device. The identification number is set via TwinCAT.

Prerequisites for this step:

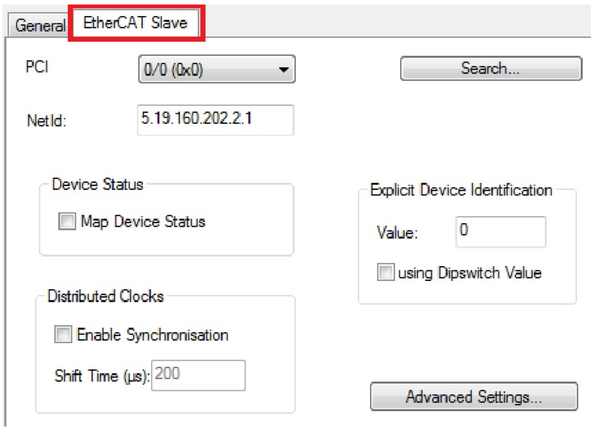
- An EtherCAT slave added in TwinCAT.

Enable the device identification as follows:

1. In the tree view click on the EtherCAT slave.



2. Click on the **EtherCAT Slave** tab



PCI: 0/0 (0x0)

NetId: 5.19.160.202.2.1

Device Status

□ Map Device Status

Distributed Clocks

□ Enable Synchronisation

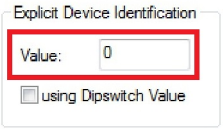
Shift Time (µs): 200

Explicit Device Identification

Value: 0

□ using Dipswitch Value

3. Enter an identification number for your EtherCAT slave under **Explicit Device Identification** in the **Value** field.



Explicit Device Identification

Value: 0

□ using Dipswitch Value

⇨ You have successfully enabled the device identification for an EtherCAT slave and allocated an identification number. The individual EtherCAT slave devices can now be distinguished from each other and swapped with devices with the same identification number during operation.

5.3.6 Creating a PLC project

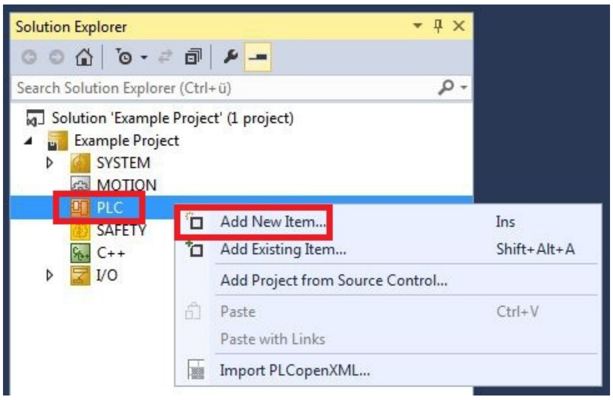
The next steps describe how to create a PLC project in TwinCAT and add it in the tree view.

Prerequisites for this step:

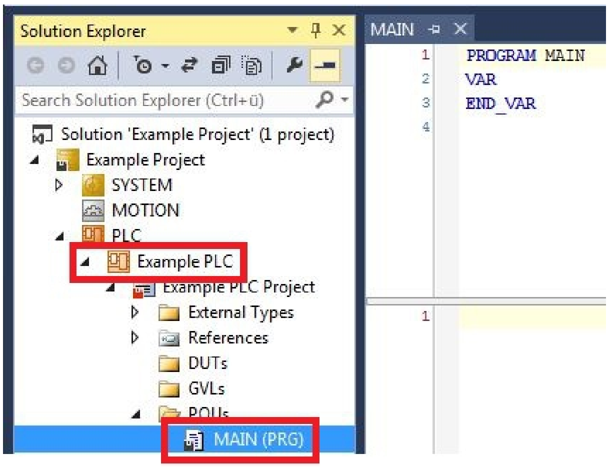
- A newly created TwinCAT XAE project.

Create a PLC project as follows:

1. Right-click on **PLC** in the tree view.
2. In the context menu click on **Add New Item** and select the **Standard PLC Project**.

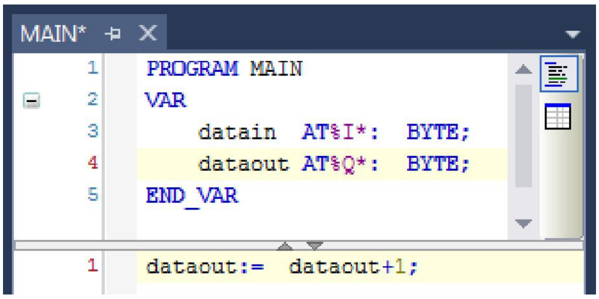

Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution 'Example Project' (1 project)
Example Project
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Add New Item... Ins
Add Existing Item... Shift+Alt+A
Add Project from Source Control...
Paste Ctrl+V
Paste with Links
Import PLCopenXML...

3. In the tree view click on the newly created PLC project, then double-click on **MAIN (PRG)** under **POUs**.



```
PROGRAM MAIN
VAR
END_VAR
```

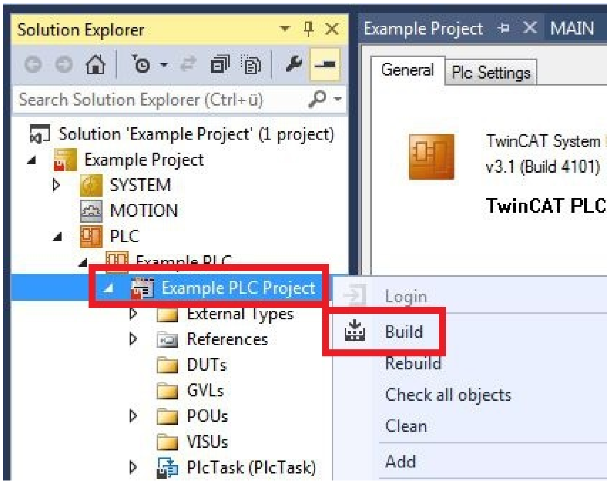
4. Write a small program, as shown in the diagram below.



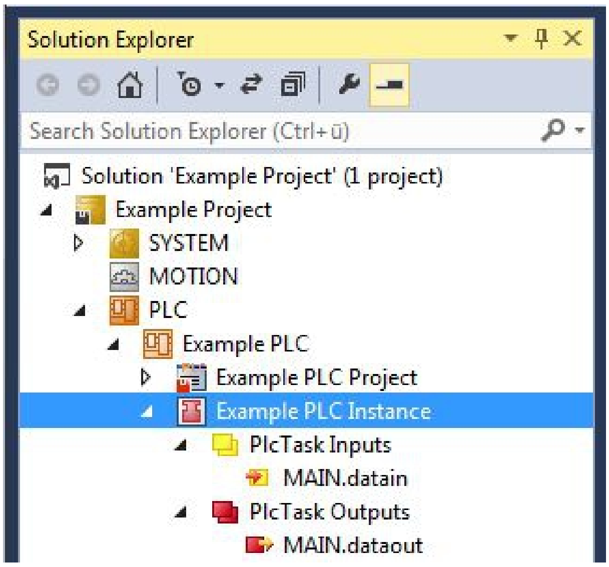
```
PROGRAM MAIN
VAR
    datain  AT%I*:  BYTE;
    dataout AT%Q*:  BYTE;
END_VAR
```

```
dataout:=  dataout+1;
```

5. In the tree view right-click on the PLC project, then click on **Build** in the context menu.



⇨ You have successfully created a PLC project and added the project in TwinCAT. A PLC instance with the variables for the inputs and outputs is created from the PLC project.



In the next step you can link the variables with the hardware.

5.3.7 Linking variables

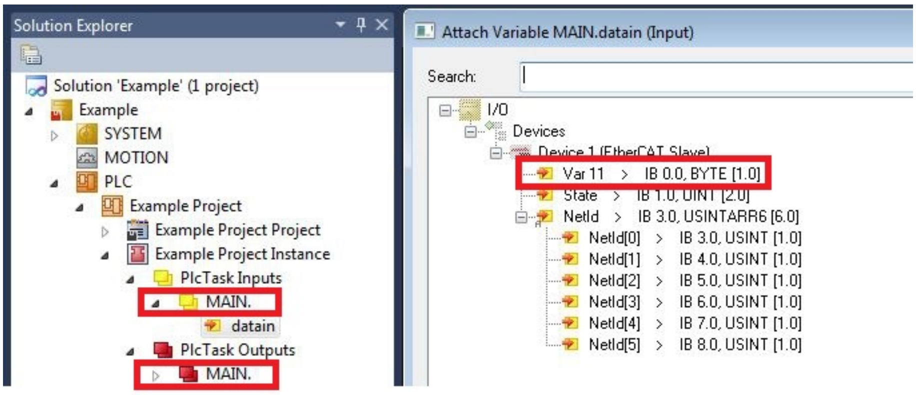
Once the PLC project was successfully added in the System Manager, you can link the newly created input and output variables from the PLC project with the inputs and outputs of your hardware.

Prerequisites for this step:

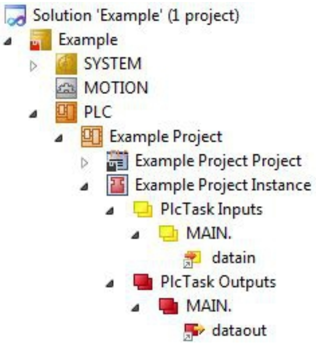
- A PLC program attached in TwinCAT.

Link the variables as follows:

1. Double-click on the input or output variables in the tree view under **PLC**.
   The **Attach Variable** window appears and shows which inputs or outputs can be linked with the variables from the PLC project.

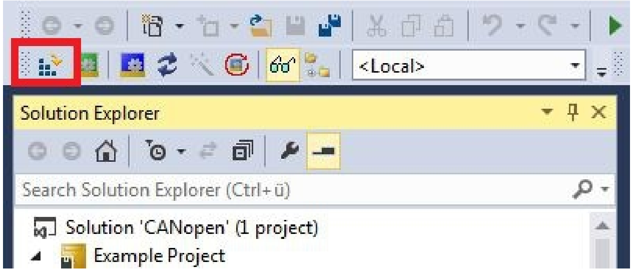

Solution Explorer
Solution 'Example' (1 project)
Example
SYSTEM
MOTION
PLC
Example Project
Example Project Project
Example Project Instance
PlcTask Inputs
MAIN.
datain
PlcTask Outputs
MAIN.
Attach Variable MAIN.datain (Input)
Search:
I/O
Devices
Device 1 (EtherCAT Slave)
Var 11 > IB 0.0, BYTE [1.0]
State > IB 1.0, UINT [2.0]
NetId > IB 3.0, USINTARR6 [6.0]
NetId[0] > IB 3.0, USINT [1.0]
NetId[1] > IB 4.0, USINT [1.0]
NetId[2] > IB 5.0, USINT [1.0]
NetId[3] > IB 6.0, USINT [1.0]
NetId[4] > IB 7.0, USINT [1.0]
NetId[5] > IB 8.0, USINT [1.0]

2. Double-click on the inputs or outputs of the hardware in the **Attach Variable** window.
   Link the input variables with the inputs and the output variables with the outputs of the hardware.



   Variables that are already linked are indicated with a small arrow icon in TwinCAT.

3. In the toolbar click on **Activate Configuration**.


<Local>
Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution 'CANopen' (1 project)
Example Project

4. Confirm the request whether TwinCAT is to start in Free Run mode with **Yes**.

⇨ You have successfully linked variables with the hardware. Use Activate Configuration to save and activate the current configuration.

The configuration can now be loaded on the CX, in order to automatically start TwinCAT in Run mode, followed by the PLC project.

5.3.8 Load configuration to CX

Once variables are linked, the configuration can be saved and loaded on the CX. This has the advantage that the PLC project is loaded and started automatically when the CX is switched on. The start of the previously created PLC project can thus be automated.

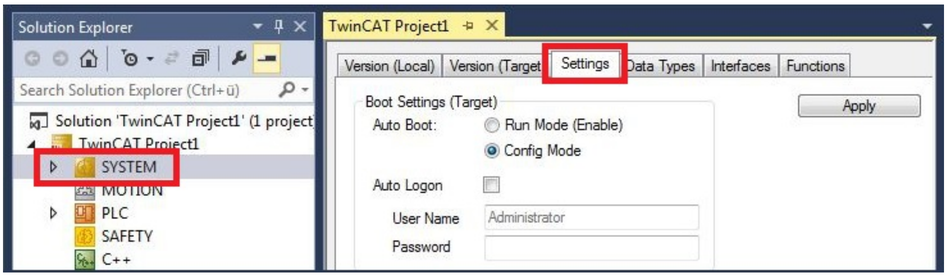
Prerequisites for this step:

- A completed PLC project, added in the System Manager.
- Variables from the PLC project, linked with the hardware in the System Manager.
- A CX selected as target system.

Load the configuration from the System Manager to the CX as follows:

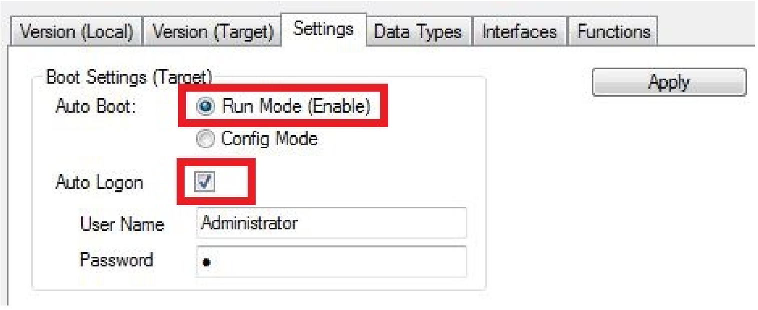
1. In the tree view on the left click on **SYSTEM**.

2. Click on the **Settings** tab.



Boot Settings (Target)
Auto Boot: ○ Run Mode (Enable) ● Config Mode
Auto Logon □
User Name Administrator
Password

3. Under Boot Settings select the option **Run Mode (Enable)** and tick the **Auto Logon** checkbox.



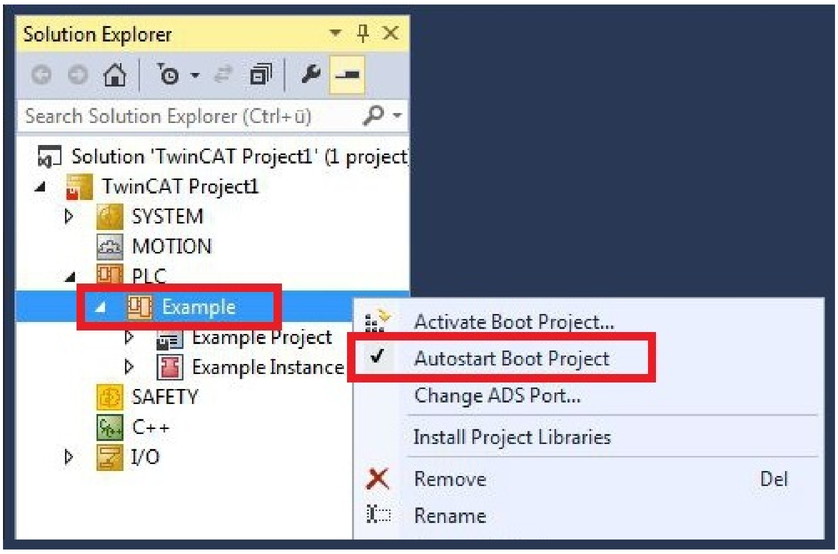
Boot Settings (Target)
Auto Boot: ● Run Mode (Enable) ○ Config Mode
Auto Logon ☑
User Name Administrator
Password •

4. Enter the user name and password for the CX in the **User Name** and **Password** fields.

5. Click on **Apply**.

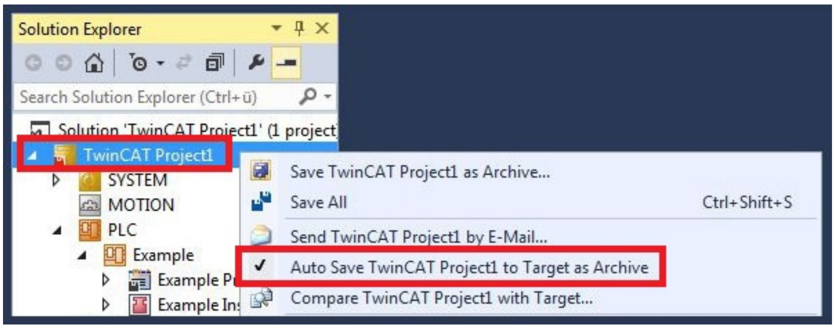
6. In the tree view on the left right-click on the PLC project under **PLC**.

7. In the context menu click on **Autostart Boot Project**.
   The setting is selected



8. Right-click on the project folder in the tree view.

9. In the context menu click on **Auto Save to Target as Archive**.
   The setting is selected.


Solution Explorer
Search Solution Explorer (Ctrl+ü)
Solution 'TwinCAT Project1' (1 project
TwinCAT Project1
SYSTEM
MOTION
PLC
Example
Example P
Example In
Save TwinCAT Project1 as Archive...
Save All Ctrl+Shift+S
Send TwinCAT Project1 by E-Mail...
✓ Auto Save TwinCAT Project1 to Target as Archive
Compare TwinCAT Project1 with Target...

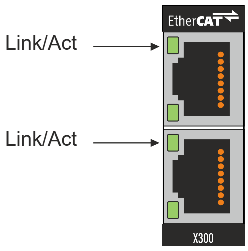
⇨ You have successfully loaded the CX configuration. From now on, TwinCAT will start in Run mode and the PLC project will start automatically.

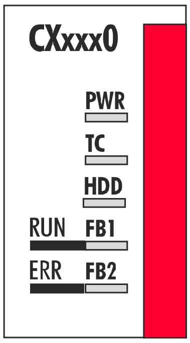
Next, the master can be added in a new project in the System Manager and can then be used to find slaves that have already been set up.

6 Error handling and diagnostics

6.1 Diagnostic LEDs

These LEDs are used for fieldbus diagnostics.

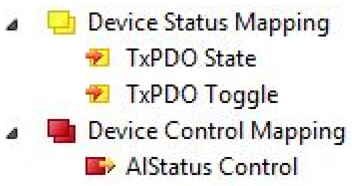
Display	LED	Display	State	Description
	Link/Act green	off	-	No connection on the EtherCAT strand
		on	linked	EtherCAT device connected
		flashing	active	Communication with EtherCAT device

Display	LED	Color	Meaning
	RUN	green	OP mode
		flashing (green) 200 ms on/off	PRE-OP mode
		flashing (green) 200 ms on/ 1000 ms off	SAFE-OP mode
		off	INIT mode
	ERR	off	OK
		red	not configured

6.2 Device Status

The device status function indicates whether the EtherCAT slave is in operation and exchanges data. The information can be used for EtherCAT slave diagnostics.

The device status function must be enabled in TwinCAT in order to display the corresponding variables in the tree view. If the function is enabled in TwinCAT, the following input and output variables are shown in the tree view:



Device Status Mapping
TxPDO State
TxPDO Toggle
Device Control Mapping
AlStatus Control

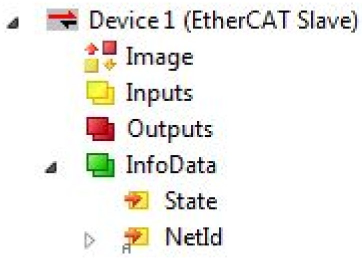
The following tables show which values the variables can assume:

Inputs	Meaning
TxPDO State	The variable provides information on whether or not the EtherCAT slave is in operational mode (OP). • “1“ the EtherCAT slave is in operational mode. • “0“ the EtherCAT slave is not in operational mode.
TxPDO Toggle	The variable provides information on whether or not the EtherCAT slave is in data exchange. • “1“ the EtherCAT slave exchanges data. • “0“ the EtherCAT slave does not exchange data.

Outputs	Meaning
AlStatus Control	The variable provides information on the operating mode of the EtherCAT slave. • 0x0001 INIT mode • 0x0002 PREOP mode • 0x0004 SAFEOP mode • 0x0008 OP mode All other bits are reserved.

6.3 State

In the tree view, input variables are consolidated under the **Info Data** menu item, which provide information about an EtherCAT slave.



The State variable indicates the EtherCAT status of the EtherCAT slave. In addition, the variable provides general information about the slave. For sample, it shows whether the slave has an invalid product code has or is disabled.

The following table shows which values the State variable can assume:

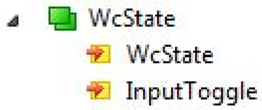
Value	Description
0x___1	Slave in ‘INIT’ state
0x___2	Slave in ‘PREOP’ state
0x___3	Slave in ‘BOOT’ state
0x___4	Slave in ‘SAFEOP’ state
0x___8	Slave in ‘OP’ state
0x001_	Slave signals error
0x002_	Invalid vendorId, productCode...read
0x004_	Initialization error occurred
0x008_	Slave disabled
0x010_	Slave not present
0x020_	Slave signals link error
0x040_	Slave signals missing link
0x080_	Slave signals unexpected link
0x100_	Communication port A
0x200_	Communication port B
0x400_	Communication port C
0x800_	Communication port D

6.4 WcState

The variables under WcState (working counter state) use a working counter to indicate whether the EtherCAT slave takes part in the cyclic process data traffic successfully and error-free.

The variable WcState of an EtherCAT slaves should therefore always be evaluated.

The variables are only displayed, if the EtherCAT slave is added via an EtherCAT master in TwinCAT.



The following table shows which values the WcState variable can assume:

Value	Description
0	Valid real-time communication in the last cycle
1	Invalid real-time communication

Invalid real-time communication may affect the process data of the other EtherCAT slaves in the same SyncUnit:

- In the event of an invalid WcState, outputs can still be switched by the PLC, in order to ensure a safe state of the system.
- Inputs freezes and show the last valid value. This means that old values are displayed, if the WcState assumed the value 1.

The variable InputToggle indicates whether a new valid telegram was received. The value is incremented by one after each successful cycle.

7 Appendix

7.1 Accessories

Connector

Item number	Description
ZS1090-0003	Ethernet/EtherCAT RJ-45 plug connector, 4-pin, IP 20, field-configurable
ZS1090-0005	Ethernet/EtherCAT RJ-45 plug connector, 8-pin, IP 20, field-configurable

Cable

Item number	Description
ZB9010	Industrial Ethernet/EtherCAT cable, fixed installation, CAT 5e, 4-wires
ZB9020	Industrial Ethernet/EtherCAT cable, suitable for drag chains, CAT 5e, 4-core
ZB903x	Industrial Ethernet/EtherCAT cable, fixed installation, AWG26, CAT 5e, 4-wires

Item number	Description
ZK1090-9191-0001	Industrial Ethernet/EtherCAT patch cable, 0.17 m
ZK1090-9191-0002	Industrial Ethernet/EtherCAT patch cable, 0.26 m
ZK1090-9191-0005	Industrial Ethernet/EtherCAT patch cable, 0.5 m
ZK1090-9191-0010	Industrial Ethernet/EtherCAT patch cable, 1.0 m
ZK1090-9191-0020	Industrial Ethernet/EtherCAT patch cable, 2.0 m
ZK1090-9191-0030	Industrial Ethernet/EtherCAT patch cable, 3.0 m
ZK1090-9191-0050	Industrial Ethernet/EtherCAT patch cable, 5.0 m
ZK1090-9191-0100	Industrial Ethernet/EtherCAT patch cable, 10.0 m
ZK1090-9191-0150	Industrial Ethernet/EtherCAT patch cable, 15.00 m
ZK1090-9191-0200	Industrial Ethernet/EtherCAT patch cable, 20.00 m
ZK1090-9191-0250	Industrial Ethernet/EtherCAT patch cable, 25.00 m
ZK1090-9191-0300	Industrial Ethernet/EtherCAT patch cable, 30.00 m
ZK1090-9191-0350	Industrial Ethernet/EtherCAT patch cable, 35.00 m
ZK1090-9191-0400	Industrial Ethernet/EtherCAT patch cable, 40.00 m
ZK1090-9191-0450	Industrial Ethernet/EtherCAT patch cable, 45.00 m
ZK1090-9191-0500	Industrial Ethernet/EtherCAT patch cable, 50.00 m

Further lengths and cable options available on request.

7.2 Certifications

FCC Approvals for the United States of America

FCC: Federal Communications Commission Radio Frequency Interference Statement

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

FCC Approval for Canada

FCC: Canadian Notice

This equipment does not exceed the Class A limits for radiated emissions as described in the Radio Interference Regulations of the Canadian Department of Communications.

7.3 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com