BECKHOFF New Automation Technology

Documentation | EN

EP6601-0002

1-port communication interface, Ethernet switch port

2024-07-05 | Version: 1.0

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
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 satisfy all the 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 prior announcement.

No claims for the modification of 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 trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

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

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

Copyright

1.2 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

1.3 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations 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 engineering 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

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

1.4 Documentation issue status

Version	Comment
1.0	First release

Firmware and hardware versions

This documentation refers to the firmware and hardware version that was applicable at the time the documentation was written.

The module features are continuously improved and developed further. Modules having earlier production statuses cannot have the same properties as modules with the latest status. However, existing properties are retained and are not changed, so that older modules can always be replaced with new ones.

The firmware and hardware version (delivery state) can be found in the batch number (D-number) printed on the side of the EtherCAT Box.

Syntax of the batch number (D-number)

D: WW YY FF HH	Example with D no. 29 10 02 01:
WW - week of production (calendar week) YY - year of production FF - firmware version HH - hardware version	29 - week of production 29 10 - year of production 2010 02 - firmware version 02 01 - hardware version 01

Further information on this topic: Version identification of EtherCAT devices [▶ 60].

2 EtherCAT Box - Introduction

The EtherCAT system has been extended with EtherCAT Box modules with protection class IP67. Through the integrated EtherCAT interface the modules can be connected directly to an EtherCAT network without an additional Coupler Box. The high-performance of EtherCAT is thus maintained into each module.

The extremely low dimensions of only 126 x 30 x 26.5 mm (h x w x d) are identical to those of the Fieldbus Box extension modules. They are thus particularly suitable for use where space is at a premium. The small mass of the EtherCAT modules facilitates applications with mobile I/O interface (e.g. on a robot arm). The EtherCAT connection is established via screened M8 connectors.

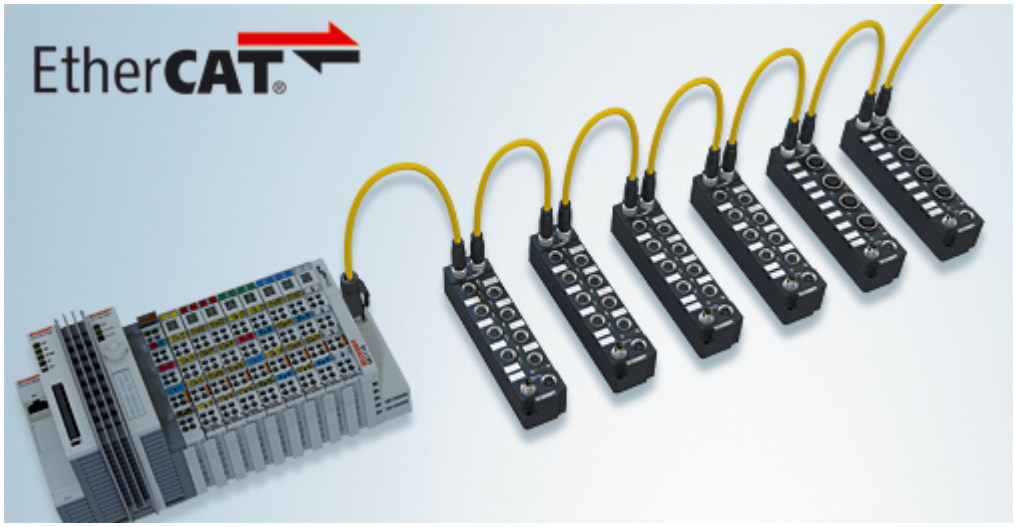


Fig. 1: EtherCAT Box Modules within an EtherCAT network

The robust design of the EtherCAT Box modules enables them to be used directly at the machine. Control cabinets and terminal boxes are now no longer required. The modules are fully sealed and therefore ideally prepared for wet, dirty or dusty conditions.

Pre-assembled cables significantly simplify EtherCAT and signal wiring. Very few wiring errors are made, so that commissioning is optimized. In addition to pre-assembled EtherCAT, power and sensor cables, field-configurable connectors and cables are available for maximum flexibility. Depending on the application, the sensors and actuators are connected through M8 or M12 connectors.

The EtherCAT modules cover the typical range of requirements for I/O signals with protection class IP67:

- digital inputs with different filters (3.0 ms or 10 µs)
- digital outputs with 0.5 or 2 A output current
- analog inputs and outputs with 16 bit resolution
- Thermocouple and RTD inputs
- Stepper motor modules

XFC (eXtreme Fast Control Technology) modules, including inputs with time stamp, are also available.

Fig. 2: EtherCAT Box with M8 connections for sensors/actuators

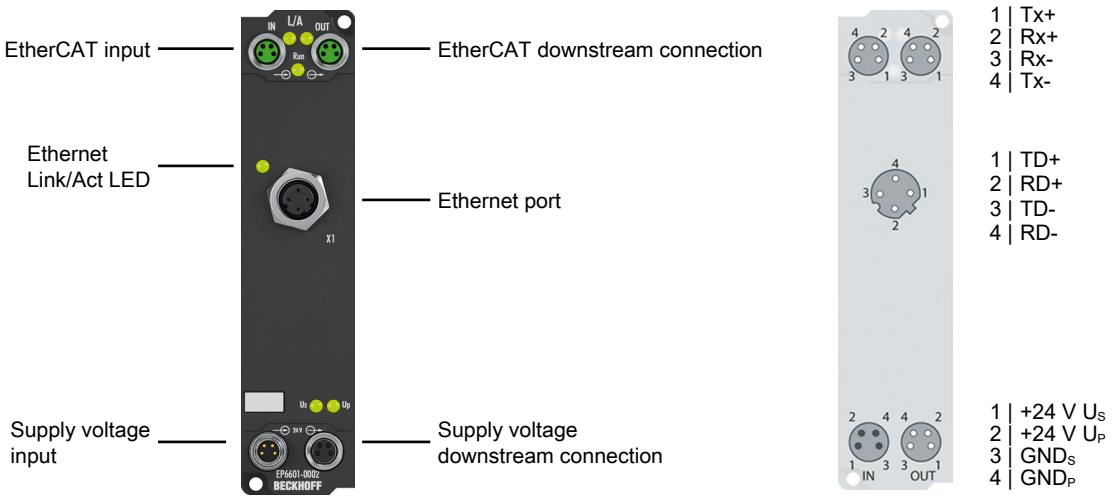
Fig. 3: EtherCAT Box with M12 connections for sensors/actuators

Basic EtherCAT documentation

You will find a detailed description of the EtherCAT system in the Basic System Documentation for EtherCAT, which is available for download from our website (www.beckhoff.com) under Downloads.

3 Product overview

3.1 Introduction



The EP6601-0002 Ethernet switch port module is used for decentralized connection of Ethernet devices to the EtherCAT terminal network. The Ethernet communication of the connected devices is forwarded by the EtherCAT system in a fully transparent and collision-free manner.

The switch port module can be used at any position within the EtherCAT segment. No configuration is required.

Further benefits underline the particular suitability for the application in industrial environments:

- compact design in IP67 housing
- 10/100 Mbit/s, half and full duplex, automatic detection of data transfer rate
- support for network variables

Quick links

Technical data [▶ 12]
Process image [▶ 14]
Ethernet connection [▶ 26]

3.2 Technical data

All values are typical values over the entire temperature range, unless stated otherwise.

EtherCAT	
Connection	2 x M8 socket, 4-pin, A-coded, shielded
Electrical isolation	500 V

Supply voltages	
Connection	Input: M8 connector, 4-pin, A-coded Downstream connection: M8 socket, 4-pin, A-coded
U_S nominal voltage	24 V_{DC} (-15 % / +20 %)
U_S sum current: $I_{S,sum}$	max. 4 A
Current consumption from U_S	120 mA
Rated voltage U_P	24 V_{DC} (-15 % / +20 %)
U_P sum current: $I_{P,sum}$	max. 4 A
Current consumption from U_P	None. U_P is only forwarded.

Ethernet port	
Bus system	All Ethernet (IEEE 802.3) based protocols, store-and-forward switching mode
Connection	M12 socket, 4-pin, D-coded 10BASE-T/100BASE-TX Ethernet
Data transfer rates	10/100 Mbit/s, IEEE 802.3u Auto negotiation, half or full duplex at 10 and 100 Mbit/s possible, automatic settings
Hardware diagnostics	Link/Act LED
Cable length	up to 100 m twisted pair
Electrical isolation	500 V
Configuration	not required
Special features	Support for RT-Ethernet, Publisher/Subscriber, DHCP/BootP server

Housing data	
Dimensions W x H x D	30 mm x 126 mm x 26.5 mm (without connectors)
Weight	approx. 165 g
Installation position	variable
Material	PA6 (polyamide)

Environmental conditions	
Ambient temperature during operation	-25 … +60 °C -25 … +55 °C according to cURus
Ambient temperature during storage	-40 … +85 °C
Vibration resistance, shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27 Additional tests [▶ 13]
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP65, IP66, IP67 (conforms to EN 60529)

Approvals/markings	
Approvals/markings *)	CE, cURus [▶ 27]

*) Real applicable approvals/markings see type plate on the side (product marking).

Additional tests

The devices have undergone the following additional tests:

Test	Explanation
Vibration	10 frequency sweeps in 3 axes
	5 Hz < f < 60 Hz displacement 0.35 mm, constant amplitude
	60.1 Hz < f < 500 Hz acceleration 5 g, constant amplitude
Shocks	1000 shocks in each direction, in 3 axes
	35 g, 11 ms

3.3 Scope of supply

Make sure that the following components are included in the scope of delivery:

- 1x EP6601-0002
- 2x protective cap for EtherCAT socket, M8, green (pre-assembled)
- 1x protective cap for supply voltage input, M8, transparent (pre-assembled)
- 1x protective cap for supply voltage output, M8, black (pre-assembled)
- 10x labels, blank (1 strip of 10)

Pre-assembled protective caps do not ensure IP67 protection

Protective caps are pre-assembled at the factory to protect connectors during transport. They may not be tight enough to ensure IP67 protection.

Ensure that the protective caps are correctly seated to ensure IP67 protection.

3.4 Process image

The process image depends on the network variable configuration.

If you use network variables, see chapter Process data of the Automation Protocol device [▶ 39].

3.5 Basic function principles

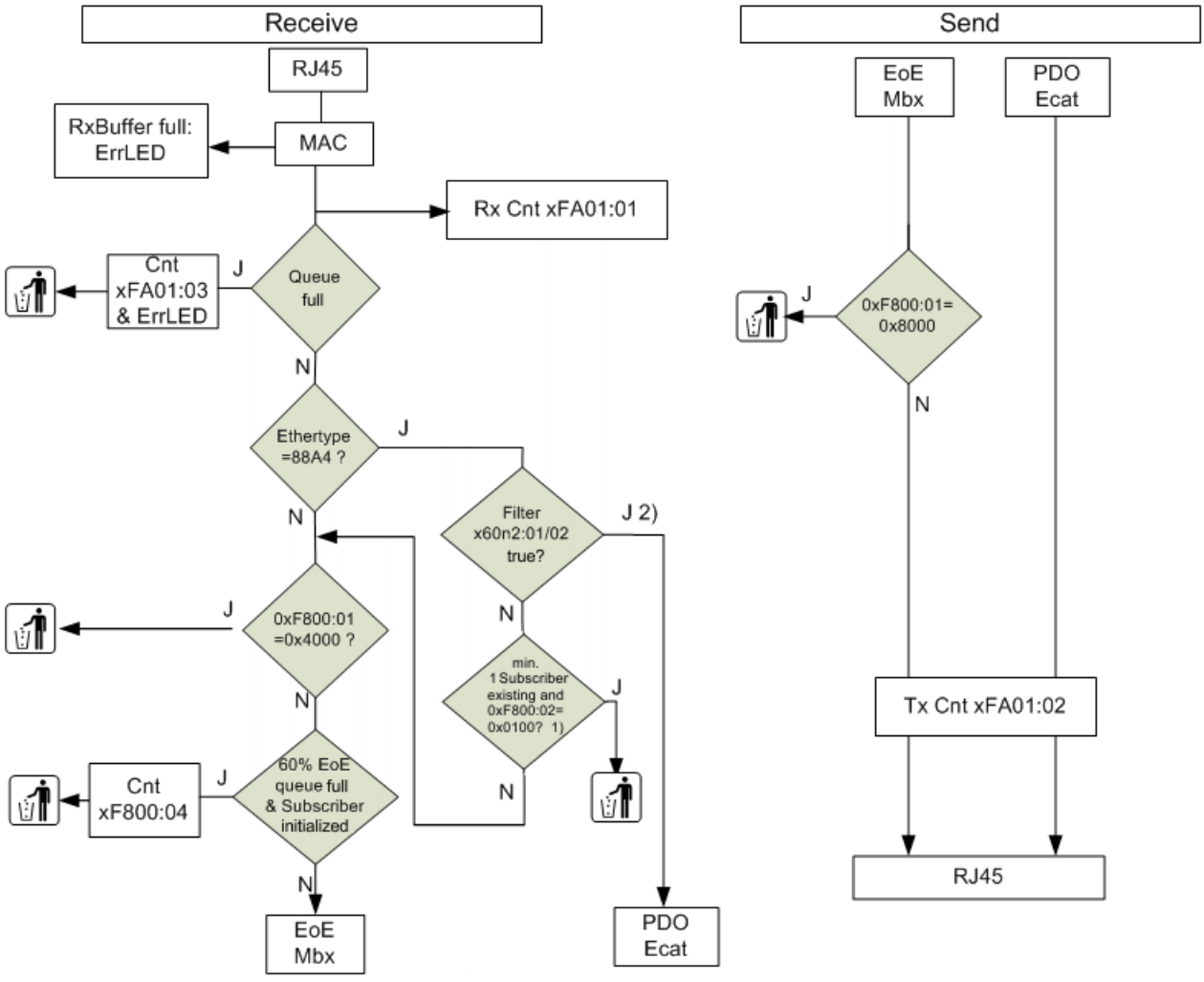
The EP6601-0002 supports two types of Ethernet traffic, which can also be processed in parallel:

- **Real-time traffic** (Publisher/Subscriber, Beckhoff network variables, EAP)
  - Sending network variables as Publisher. The box receives the network variables to be sent from the EtherCAT master via the cyclic EtherCAT traffic.
  - Receiving network variables as Subscriber. The box forwards the received network variables to the EtherCAT master via the cyclic EtherCAT traffic.
- **Non-real-time traffic**
  Non-real-time traffic is transmitted between the box and the EtherCAT master using acyclic EtherCAT mailbox traffic (EoE = "Ethernet over EtherCAT").
  This transmission is throughput-optimized and, if necessary, with automatic fragmentation - by default, all telegrams that are not transmitted in PDO context are transported in the acyclic channel via EoE.

The EP6601-0002 cannot transport EtherNet Industrial Protocol (EtherNet/IP).

Data flow

The following diagram shows the data flow in the EP6601-0002.



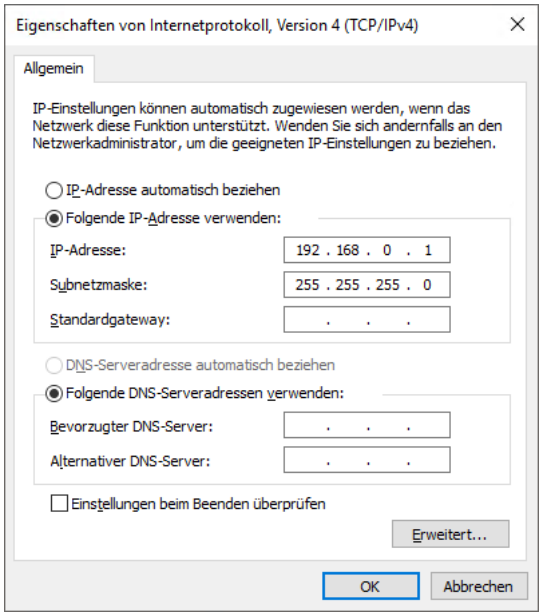
[1)] A maximum of as many bytes are copied into the PDO as are configured in the PDO. If the data volume is higher, incomplete data may arrive at the controller. In the CoE object 0x60xx (for each Subscriber), the individual components can be added to the SubscriberFilter for "IgnoreItem":

- 0x60n0: filter values as value
- 0x60n2: corresponding filter value activated. Default: "VariableID" is activated

Cable redundancy

If the EP6601-0002 is operated in a system with cable redundancy, please keep the following in mind:

- real-time operation with network variables is possible.
- in the event of non-real-time operation with IP transfer, the IP traffic is routed via the primary EtherCAT port. Therefore the Windows IP settings of this port are also used.



If the link to this port is lost, IP communication to this port is currently no longer possible from Windows under TwinCAT 2 or 3.
It must therefore be avoided that the Ethernet connection between the primary EtherCAT port and the first EtherCAT slave fails, as otherwise IP communication via the EP6601-0002 is no longer possible.

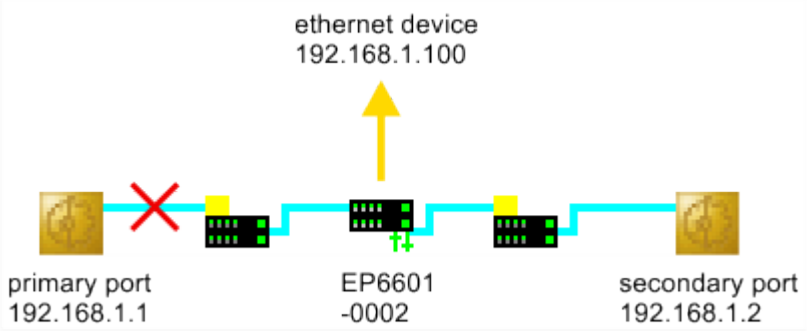


Fig. 4: Connection failure between the primary EtherCAT port and the first EtherCAT device

3.5.1 Beckhoff network variables

The EP6601-0002 supports the sending and receiving of network variables.

A maximum total of 32 Publisher and Subscriber per EP6601-0002 is permitted.

Network variables are specially configured Ethernet frames that enable Beckhoff devices to communicate with each other in real-time via Ethernet. Such a device can be a sender of messages (Publisher) or a recipient (Subscriber).
One Ethernet frame is sent per Publisher (Ethernet-based). A maximum of 1500 bytes of data can thus be sent per Publisher. Several variables - the Publisher or Subscriber variables - can be created within a Publisher/Subscriber.
Several Publisher/Subscriber can usually be configured for each sending/receiving device (e.g. IPC or EP6601-0002).

The hierarchy of a data sender, for example, consists of

- the sending device with at least one Ethernet interface: IPC, CX, FC9011, EP6601-0002, ...
  - FastEthernet/100 Mbit and 1 Gbit are supported
  - this Ethernet interface is configured in the local TwinCAT System Manager as a real-time Ethernet device
- 1..n configured Publisher - each Publisher is sent as a separate Ethernet frame and can therefore comprise a maximum of 1500 bytes
- 1..n Publisher variables contained therein for linking with the task/PLC
  - for each Publisher variable the user data and diagnostic data are transferred

The configuration is mirrored on the receiver side.

The EP6601-0002 can also process Publisher and Subscriber independently, the framework data are

- max. 32 Publisher and/or Subscriber
- per transmission direction (Publisher or Subscriber) are maximum permissible in total
  - all Publisher: 1024 bytes total data
  - all Subscriber: 1024 bytes total data

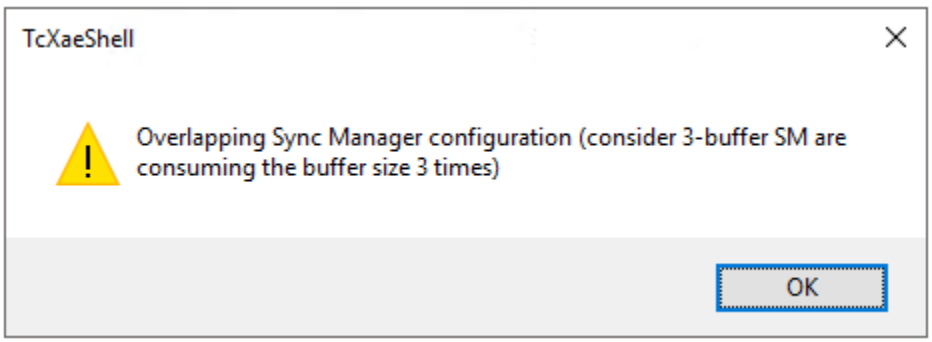
Note on the term total data

The EP6601-0002 can transmit a maximum of 1024 bytes of total data per data direction. The total data consists of the user data (e.g. a UDINT to be transmitted) and diagnostic data from EP6601-0002.

Formula for number of diagnostic data bytes

- Publisher direction: 2 + ((number of Publishers) * 2)
- Subscriber direction: 2 + ((number of Subscriber variables) * 4)

If the configured amount of data exceeds 1024 bytes, this is indicated by a message window when an activation attempt is made:



Note regarding the amount of data

The EP6601-0002 has 8 kbyte of usable memory, which is allocated as follows by default

Type	Usable extent	Operation mode	Used memory
Mailbox Out	1024 bytes		1024 bytes (fixed)
Mailbox In	1024 bytes		1024 bytes (fixed)
Publisher	1024 bytes	3-buffer mode	3072 bytes
Subscriber	1024 bytes	3-buffer mode	3072 bytes

If more Publisher or Subscriber data are required for an application, the SyncManagers can be modified accordingly. The mailbox cannot be modified.

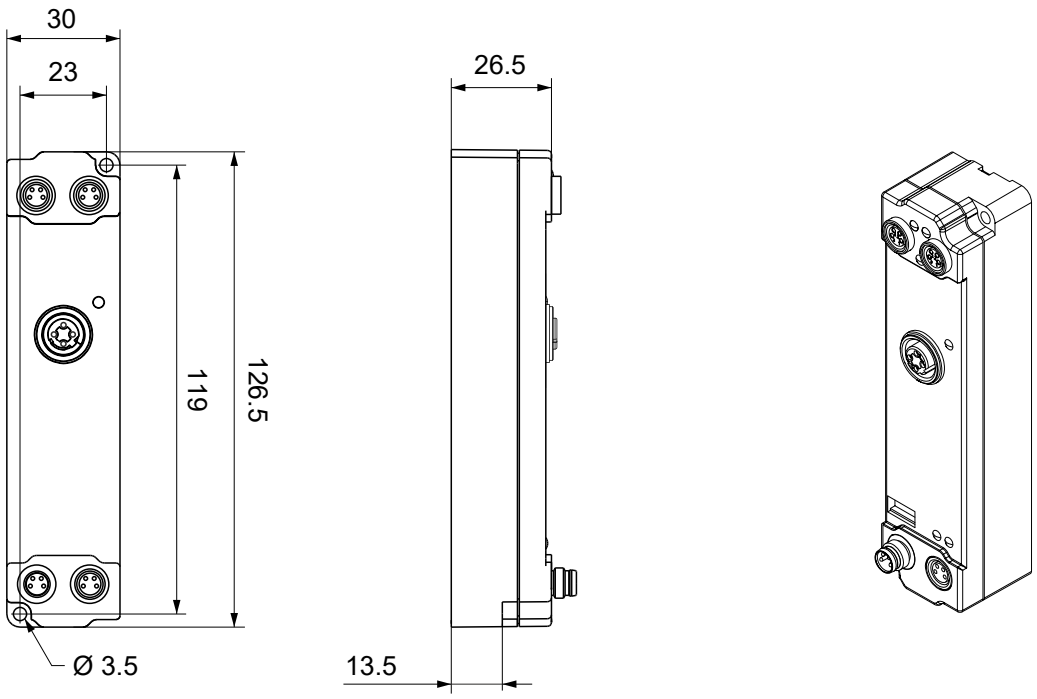
3.5.1.1 Notes

- The RT statistics displays are not supported under TwinCAT for an EP6601-0002-RT device.
  Solution: Alternatively, corresponding CoE parameters can be read for diagnostic purposes.
- The Publisher features "OnChangeOnly" and "DataExchange (Divider/Modulo)" are not supported in connection with EP6601-0002.
  Solution: The sending of the configured Publisher variables can be cyclically prevented by DevCtrl.
- If a Publisher is set up on an EP6601-0002, the CycleIndex [▶ 41] of the Publisher must be operated by the user. On a PC, however, they are incremented by TwinCAT.
- The following is recommended for diagnosing a network variable connection:
  1. Monitor LinkState in the "*DevState*" of the RT device (Device --> Inputs --> DevState). *DevState* = 0 is the target state.
  2. Monitor the *Quality* and *CycleIndex* in the Subscriber.
- The Link LED in the EP6601-0002 only indicates the state of the cable connection, not that of any network variable connection that may exist.
- If the EP6601-0002 is used exclusively for network variable traffic, the "Connect to TCP/IP Stack" button should be disabled. See chapter Virtual switch [▶ 34].
- A maximum of 32 Publishers and Subscribers are permitted for each EP6601-0002.

4 Mounting and connection

4.1 Installation

4.1.1 Dimensions



All dimensions are given in millimeters.
The drawing is not true to scale.

Housing features

Housing material	PA6 (polyamide)
Sealing compound	polyurethane
Mounting	two mounting holes Ø 3.5 mm for M3
Metal parts	brass, nickel-plated
Contacts	CuZn, gold-plated
Power feed through	max. 4 A
Installation position	variable
Protection class	IP65, IP66, IP67 (conforms to EN 60529) when screwed together
Dimensions (H x W x D)	approx. 126 x 30 x 26.5 mm (without connectors)

4.1.2 Fixing

NOTICE

Dirt during assembly

Dirty connectors can lead to malfunctions. Protection class IP67 can only be guaranteed if all cables and connectors are connected.

- Protect the plug connectors against dirt during the assembly.

Mount the module with two M3 screws on the mounting holes in the corners of the module. The mounting holes have no thread.

4.1.3 Tightening torques for plug connectors

Screw connectors tight with a torque wrench. (e.g. ZB8801 from Beckhoff)

Connector diameter	Tightening torque
M8	0.4 Nm
M12	0.6 Nm

4.2 Connections

4.2.1 EtherCAT

4.2.1.1 Connectors

NOTICE

Risk of confusion: supply voltages and EtherCAT

Defect possible through incorrect insertion.

- Observe the color coding of the connectors:
  black: Supply voltages
  green: EtherCAT

EtherCAT Box Modules have two green M8 sockets for the incoming and downstream EtherCAT connections.

Connection

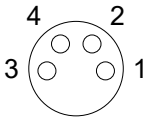


Fig. 5: M8 socket

EtherCAT	M8 socket	Core colors		
Signal	**Contact**	**ZB9010, ZB9020, ZB9030, ZB9032, ZK1090-6292, ZK1090-3xxx-xxxx**	**ZB9031 and old versions of ZB9030, ZB9032, ZK1090-3xxx-xxxx**	**TIA-568B**
Tx +	1	yellow[1)]	orange/white	white/orange
Tx -	4	orange[1)]	orange	orange
Rx +	2	white[1)]	blue/white	white/green
Rx -	3	blue[1)]	blue	green
Shield	Housing	Shield	Shield	Shield

[1)] Core colors according to EN 61918

Adaptation of core colors for cables ZB9030, ZB9032 and ZK1090-3xxxx-xxxx

For standardization, the core colors of the ZB9030, ZB9032 and ZK1090-3xxx-xxxx cables have been changed to the EN61918 core colors: yellow, orange, white, blue. So there are different color codes in circulation. The electrical properties of the cables have been retained when the core colors were changed.

4.2.1.2 Status LEDs

L/A (Link/Act)

A green LED labelled "L/A" is located next to each EtherCAT socket. The LED indicates the communication state of the respective socket:

LED	Meaning
off	no connection to the connected EtherCAT device
lit	LINK: connection to the connected EtherCAT device
flashes	ACT: communication with the connected EtherCAT device

Run

Each EtherCAT slave has a green LED labelled "Run". The LED signals the status of the slave in the EtherCAT network:

LED	Meaning
off	Slave is in "Init" state
flashes uniformly	Slave is in "Pre-Operational“ state
flashes sporadically	Slave is in "Safe-Operational" state
lit	Slave is in "Operational" state

Description of the EtherCAT slave states

4.2.1.3 Cables

For connecting EtherCAT devices only shielded Ethernet cables that meet the requirements of at least category 5 (CAT5) according to EN 50173 or ISO/IEC 11801 should be used.

EtherCAT uses four wires for signal transmission.
Thanks to automatic line detection ("Auto MDI-X"), both symmetrical (1:1) or cross-over cables can be used between Beckhoff EtherCAT.

Detailed recommendations for the cabling of EtherCAT devices

4.2.2 Supply voltages

⚠ WARNING

Power supply from SELV / PELV power supply unit!

SELV / PELV circuits (safety extra-low voltage / protective extra-low voltage) according to IEC 61010-2-201 must be used to supply this device.

Notes:

- SELV / PELV circuits may give rise to further requirements from standards such as IEC 60204-1 et al, for example with regard to cable spacing and insulation.
- A SELV supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV supply also requires a safe connection to the protective conductor.

⚠ CAUTION

Observe the UL requirements

- When operating under UL conditions, observe the warnings in the chapter UL Requirements [▶ 27].

The EtherCAT Box has one input for two supply voltages:

- **Control voltage U_S**
  The following sub-functions are supplied from the control voltage U_S:
  - the fieldbus
  - the processor logic
  - typically the inputs and the sensors if the EtherCAT Box has inputs.
- **Peripheral voltage U_P**
  For EtherCAT Box modules with digital outputs the digital outputs are typically supplied from the peripheral voltage U_P. U_P can be supplied separately. If U_P is switched off, the fieldbus function, the function of the inputs and the supply of the sensors are maintained.

The exact assignment of U_S and U_P can be found in the pin assignment of the I/O connections.

Redirection of the supply voltages

The power IN and OUT connections are bridged in the module. Hence, the supply voltages U_s and U_p can be passed from EtherCAT Box to EtherCAT Box in a simple manner.

NOTICE

Note the maximum current!

Ensure that the permitted current for the connectors is not exceeded when routing the supply voltages U_S and U_P:

M8 connector: max. 4 A
7/8" connector: max 16 A

NOTICE

Unintentional cancellation of the electrical isolation possible

In some types of EtherCAT Box modules the ground potentials GND_S and GND_P are connected.

- If several EtherCAT Box modules are supplied with the same electrically isolated voltages, check whether there is an EtherCAT Box among them in which the ground potentials are connected.

4.2.2.1 Connectors

NOTICE

Risk of confusion: supply voltages and EtherCAT

Defect possible through incorrect insertion.

- Observe the color coding of the connectors:
  black: Supply voltages
  green: EtherCAT

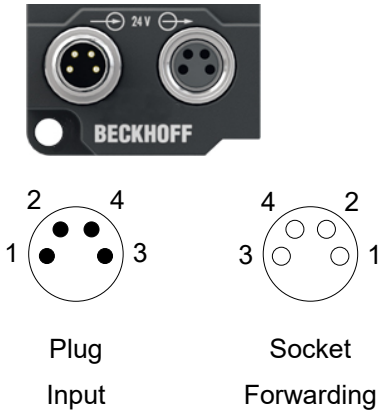


Fig. 6: M8 connector

Contact	Function	Description	Core color [1]
1	U_S	Control voltage	Brown
2	U_P	Peripheral voltage	White
3	GND_S	GND to U_S	Blue
4	GND_P	GND to U_P	Black

[1] The core colors apply to cables of the type: Beckhoff ZK2020-3xxx-xxxx

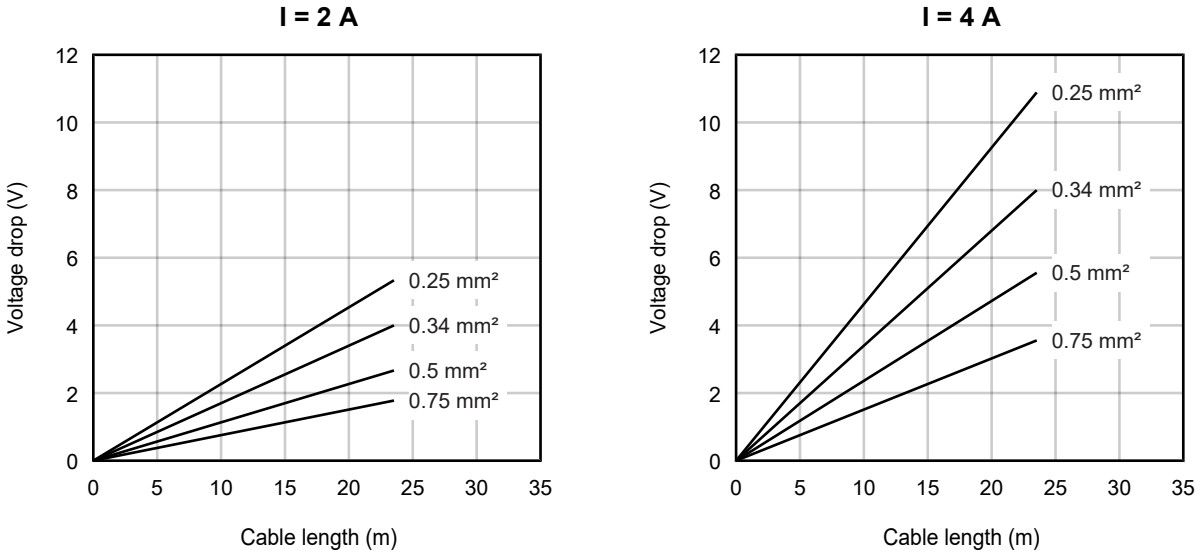
4.2.2.2 Status LEDs

LED	Display	Meaning
U_S (control voltage)	off	The supply voltage U_S is not available.
	green illuminated	The supply voltage U_S is available.
U_P (peripheral voltage)	off	The supply voltage U_P is not available.
	green illuminated	The supply voltage U_P is available.

4.2.2.3 Conductor losses

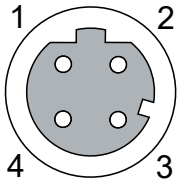
Take into account the voltage drop on the supply line when planning a system. Avoid the voltage drop being so high that the supply voltage at the box lies below the minimum nominal voltage.
Variations in the voltage of the power supply unit must also be taken into account.

Voltage drop on the supply line



4.2.3 Ethernet

The Ethernet connection is established via D-coded M12 socket.



Pin	Function
1	TD +
2	RD +
3	TD -
4	RD -
Housing	Shield

4.2.3.1 Status LED

L/A (Link/Act)

There is a green LED next to the Ethernet socket. The LED signals the Ethernet communication state:

LED	Meaning
off	no connection to the connected Ethernet device
lit	LINK: connection to the connected Ethernet device
flashes	ACT: communication with the connected Ethernet device

4.3 UL Requirements

The installation of the EtherCAT Box Modules certified by UL has to meet the following requirements.

Supply voltage

⚠ CAUTION

CAUTION!

This UL requirements are valid for all supply voltages of all marked EtherCAT Box Modules!
For the compliance of the UL requirements the EtherCAT Box Modules should only be supplied

- by a 24 V_{DC} supply voltage, supplied by an isolating source and protected by means of a fuse (in accordance with UL248), rated maximum 4 Amp, or
- by a 24 V_{DC} power source, that has to satisfy *NEC class 2*.
  A *NEC class 2* power supply shall not be connected in series or parallel with another (class 2) power source!

⚠ CAUTION

CAUTION!

To meet the UL requirements, the EtherCAT Box Modules must not be connected to unlimited power sources!

Networks

⚠ CAUTION

CAUTION!

To meet the UL requirements, EtherCAT Box Modules must not be connected to telecommunication networks!

Ambient temperature range

⚠ CAUTION

CAUTION!

To meet the UL requirements, EtherCAT Box Modules has to be operated only at an ambient temperature range of -25 °C to +55 °C!

Marking for UL

All EtherCAT Box Modules certified by UL (Underwriters Laboratories) are marked with the following label.

Fig. 7: UL label

4.4 Disposal

Products marked with a crossed-out wheeled bin shall not be discarded with the normal waste stream. The device is considered as waste electrical and electronic equipment. The national regulations for the disposal of waste electrical and electronic equipment must be observed.

5 Commissioning

5.1 Integrating into a TwinCAT project

The procedure for integration in a TwinCAT project is described in these Quick start guide.

5.2 Ethernet configuration

The EoE method (Ethernet over EtherCAT) is used for the EP6601-0002. Dedicated settings are available for this in the System Manager.

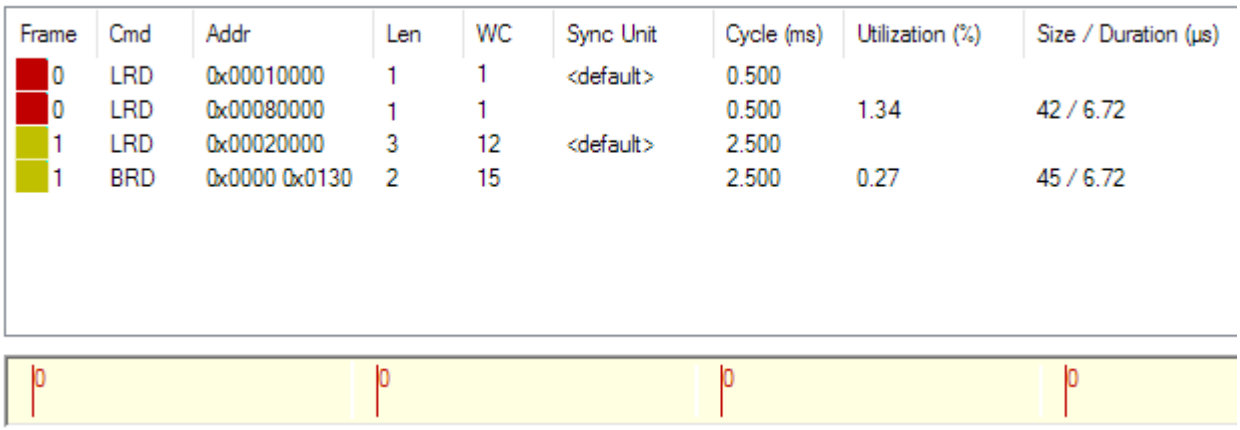
5.2.1 Data throughput

The following factors influence the Ethernet data throughput:

- The EtherCAT cycle time: the shorter the EtherCAT cycle used for the process data, the more acyclic mailbox queries can be carried out. If several different EtherCAT cycle times are used in an EtherCAT segment, the fastest cycle time is decisive.
- The time between process data frames that is available for mailbox communication: the longer the Ethernet line is free for acyclic mailbox communication, the higher the Ethernet data throughput of the EP6601-0002.
- The size of the EP6601-0002 mailbox: the larger the mailbox, the more Ethernet frames the EP6601-0002 can send to or receive from the EtherCAT master in the same time. See chapter Mailbox [▶ 33].
- The number of terminals used in the EtherCAT system that simultaneously use mailbox communication.
- The settings of the virtual switch. See chapter Virtual switch [▶ 34].

Tips for shortening the response times

To shorten the response times in your application (e.g. to ping requests), we recommend the following procedure: Reduce the EtherCAT cycle time currently used significantly or add a new task with a smaller cycle time, e.g.: 500 µs if you have previously used a 2.5 ms EtherCAT cycle. Important: This task must access real I/O process data of the EtherCAT slaves and must be recognizable under EtherCAT device -> EtherCAT tab, see Fig. *Real frame structure from the TwinCAT System Manager*



Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (µs)
0	LRD	0x00010000	1	1	<default>	0.500		
0	LRD	0x00080000	1	1		0.500	1.34	42 / 6.72
1	LRD	0x00020000	3	12	<default>	2.500		
1	BRD	0x0000 0x0130	2	15		2.500	0.27	45 / 6.72

Fig. 8: Example of a frame structure from TwinCAT

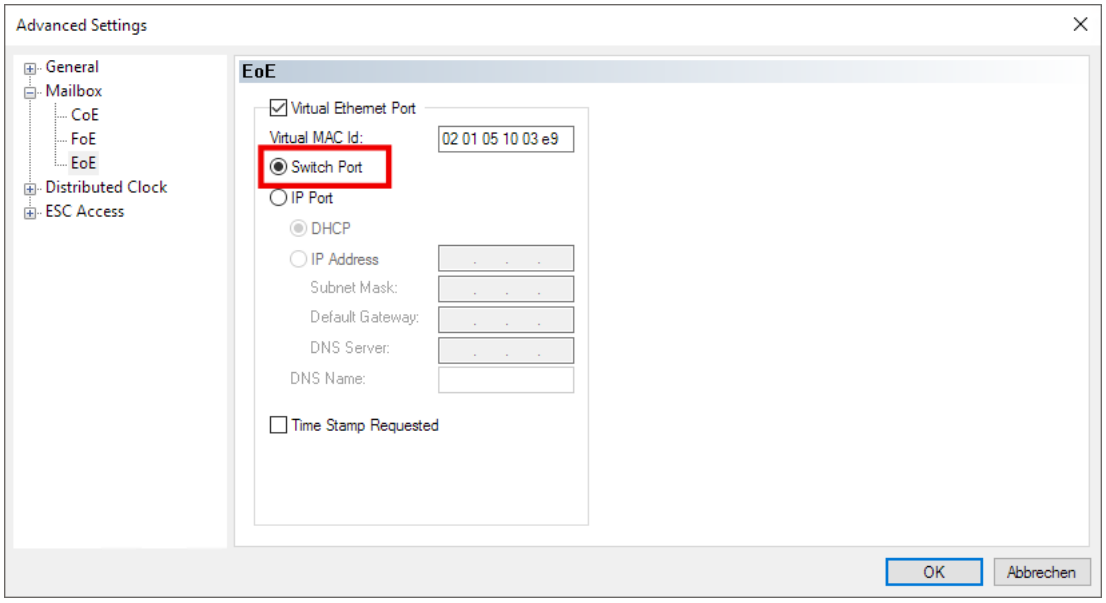
Note regarding the specified values

These values are typical values without warranty. Throughput rates may differ in different applications depending on boundary conditions.

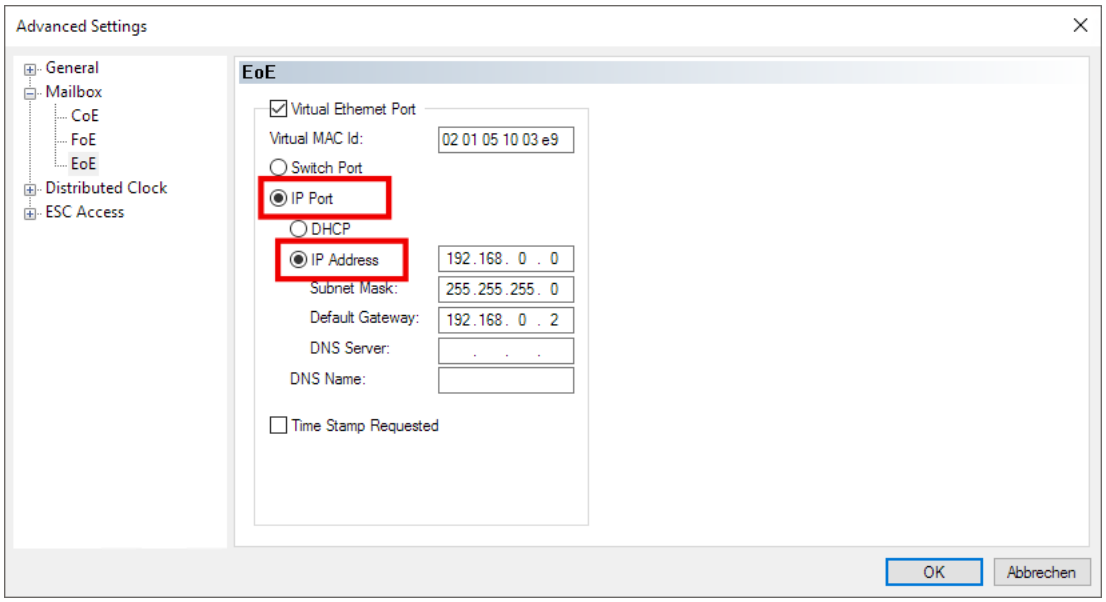
5.2.2 IP address assignment

The EP6601-0002 can assign IP addresses to connected devices and works as a DHCP or BootP server for *one* device. The settings for this must be made in the System Manager (EP6601-0002 --> Advanced Settings --> Mailbox --> EoE) as follows:

- Default setting "Switch Port".
  The EP6601-0002 works like a normal switch and transparently forwards Ethernet frames to TwinCAT/ Windows.



- Setting "IP Port".
  The EP6601-0002 works with address assignment to *one* connected Ethernet device. Enable a DHCP or BootP client in the Ethernet device (see Network adapter settings in the operating system). The EP6601-0002 responds to the DHCP or BootP request from the Ethernet device by assigning the specified IP address/subnet mask to the Ethernet device. In the DHCP procedure, this address is regularly requested again by the client and assigned by the DHCP/BootP server (EP6601-0002).



Please note:

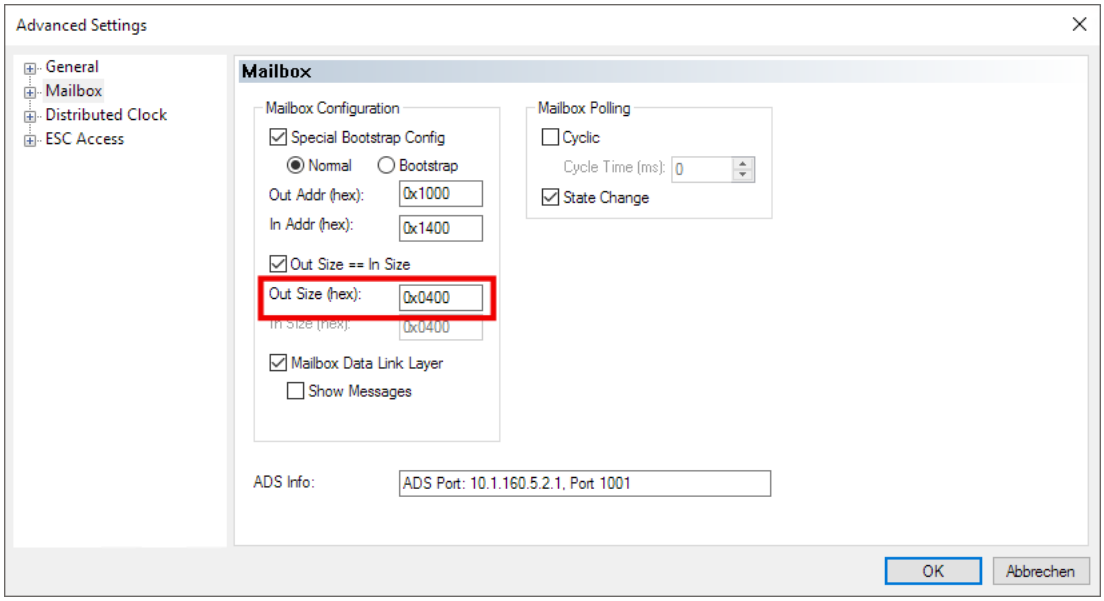
- do not use the "DHCP" checkbox - the checkbox "IP Address" enables the DHCP/BootP function in the EP6601-0002.
- the gateway, mask and server settings are also communicated to the client/device.
- only *one* address can be assigned, i.e. no switch with connected Subscribers may follow.

- the address range must match that of the EtherCAT adapter.
- DHCP server identifier: some DHCP servers require a ServerID in the answer telegram.
  Solution for EP6601-0002: The value 0x1000 must be entered in CoE 0xF800:01. If a default gateway is entered in the EP6601-0002, this is then used as the DHCP server identifier.

5.2.3 Mailbox

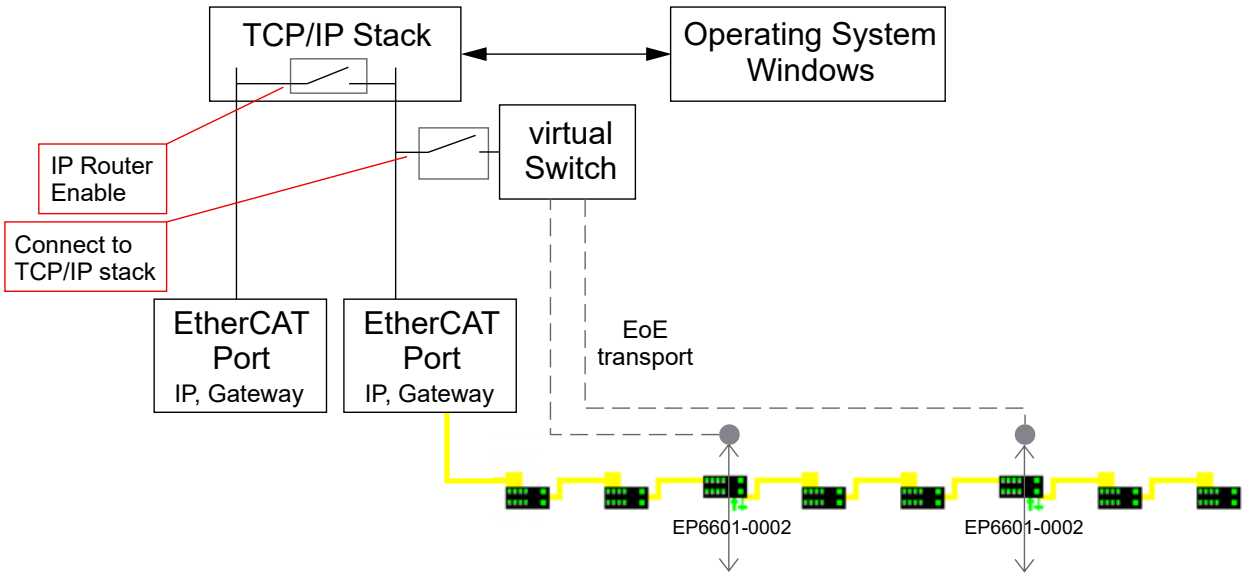
The mailbox size can be set in TwinCAT.

Under EP6601-0002 -> EtherCAT tab -> "*Advanced Settings...*" -> "*Mailbox*" the "Out Size" can be set to hexadecimal values between 42_{dec}/2 A_{hex} and 1024_{dec}/400_{hex} bytes. Ethernet frames that are larger than the mailbox of the EP6601-0002 are fragmented by the EP6601-0002 or the EtherCAT master and reassembled after passing through the EtherCAT system.

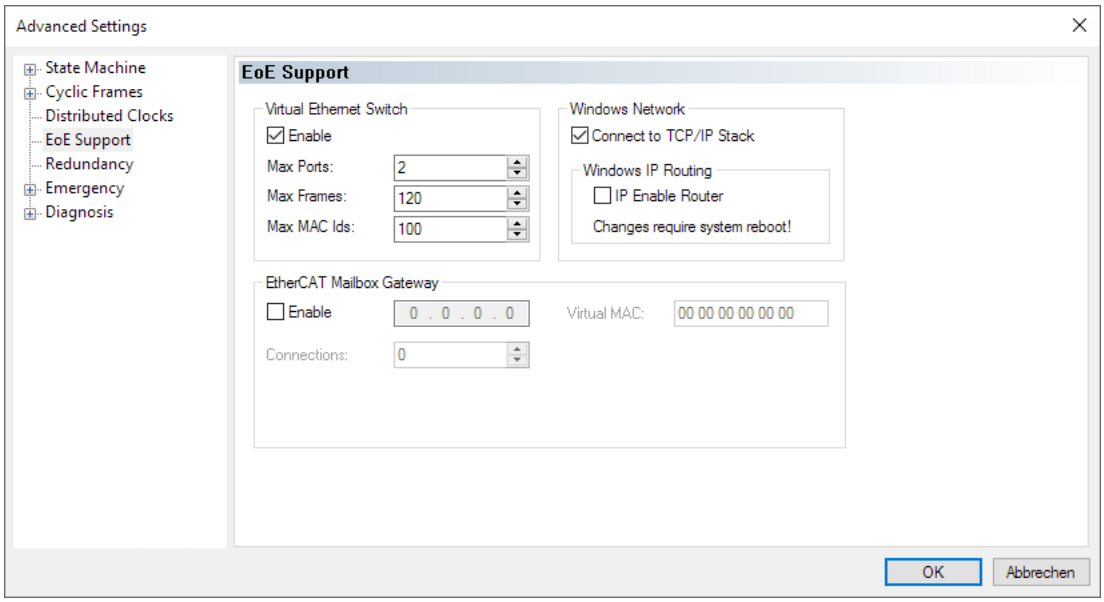


5.2.4 Virtual switch

The EP6601-0002 in the TwinCAT system act as a virtual switch, with the EtherCAT system as the "backbone".



The settings for this can be found under TwinCAT > EtherCAT device > "Advanced Settings".



Notes

- If there are a large number of EP6601-0002 or EL6601 in the EtherCAT segment, it may be useful to increase the specification "Max Frames".
- If the EP6601-0002 is used exclusively to transfer network variables, "Connect to TCP/IP Stack" should be disabled. See chapter Virtual switch [▶ 34].
- IP-routing is enabled by default. This can also be checked by entering "`ipconfig /all`" on the command line (Windows)

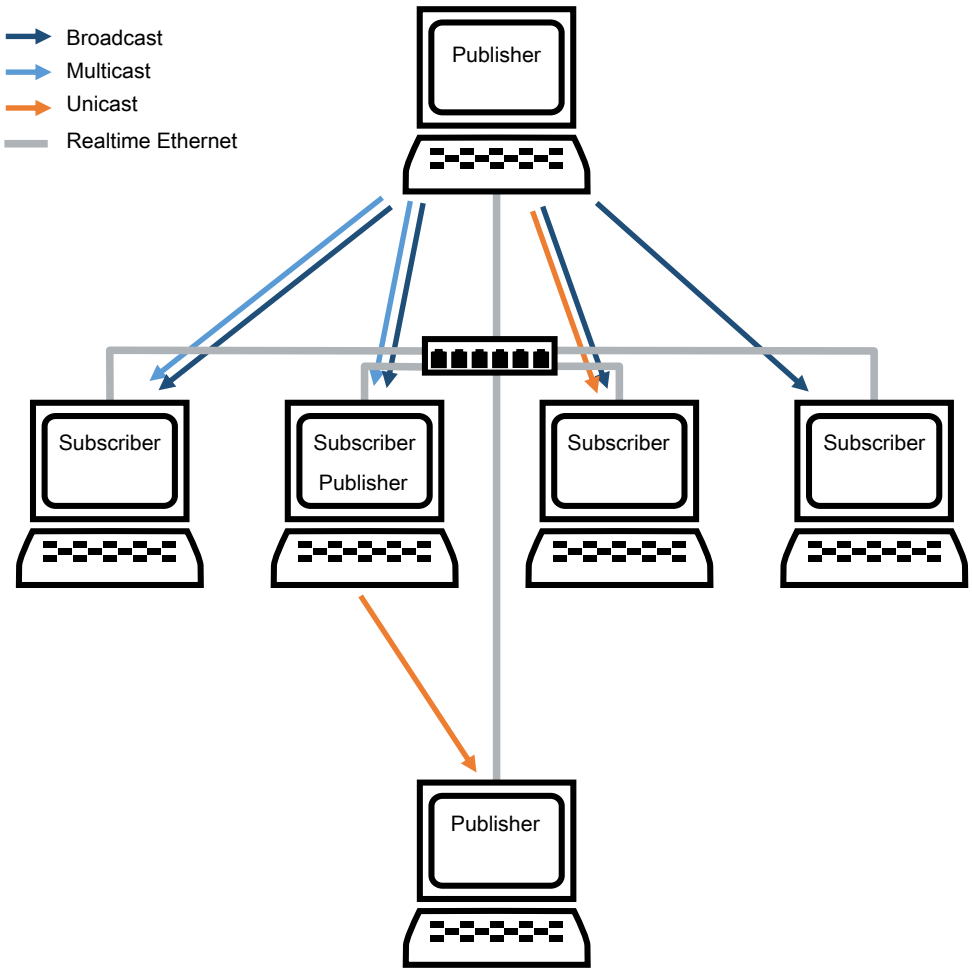
5.3 Beckhoff network variables

5.3.1 Introduction

Network variables are variables that can be exchanged cyclically between TwinCAT controllers via real-time Ethernet. Variables with any data types, including complex types, can be exchanged. The Publisher/Subscriber model is used. For highly deterministic communication, the real-time Ethernet driver for TwinCAT must be installed.

Publisher/Subscriber model

In the Publisher/Subscriber model, the Publisher provides variables. Subscribers can subscribe to a variable. The Publisher can make the variable available to one Subscriber, several Subscribers or all Subscribers. In Broadcast mode the variable is made available to all PCs, in Multicast mode to selected PCs and in Unicast mode only to one selected PC. A Subscriber can also be Publisher at the same time. This also allows a bidirectional data connection to be established.



Unicast

The Publisher makes the network variable available to a single selected PC.

Multicast

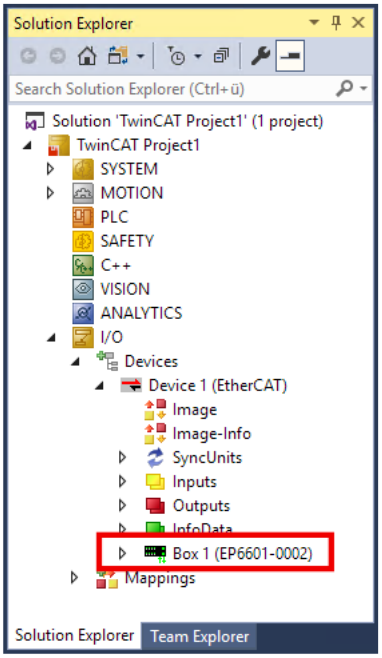
The Publisher makes the network variable available to selected PCs.

Broadcast

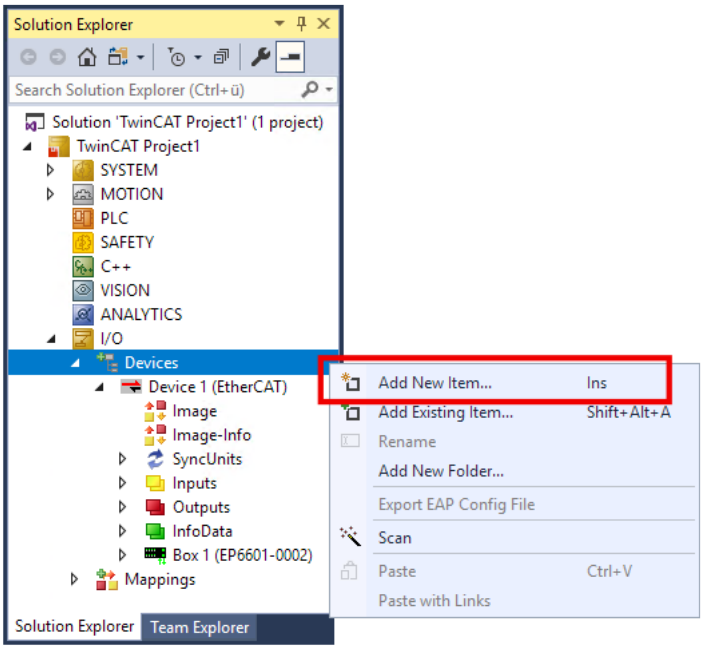
The Publisher makes the network variable available to all PCs.

5.3.2 Setup in TwinCAT 3

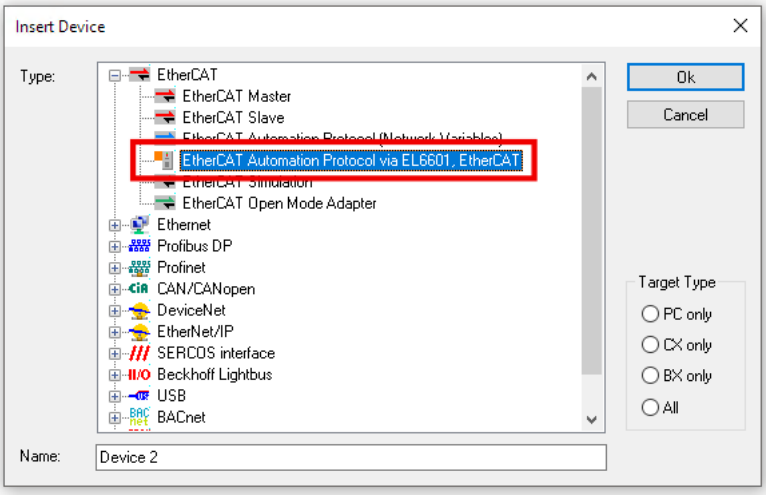
1. Add an EP6601-0002 in the Solution Explorer.



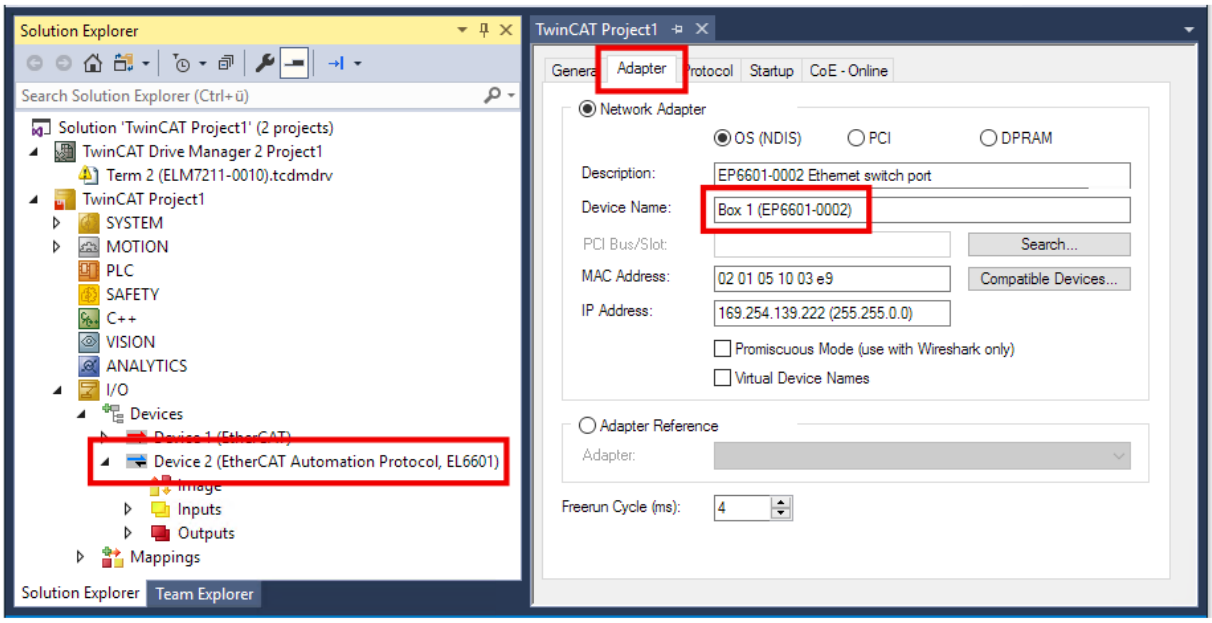
2. Right-click on "Devices" and select "Add New Item…".



3. Select the device "EtherCAT Automation Protocol via EL6601, EtherCAT".

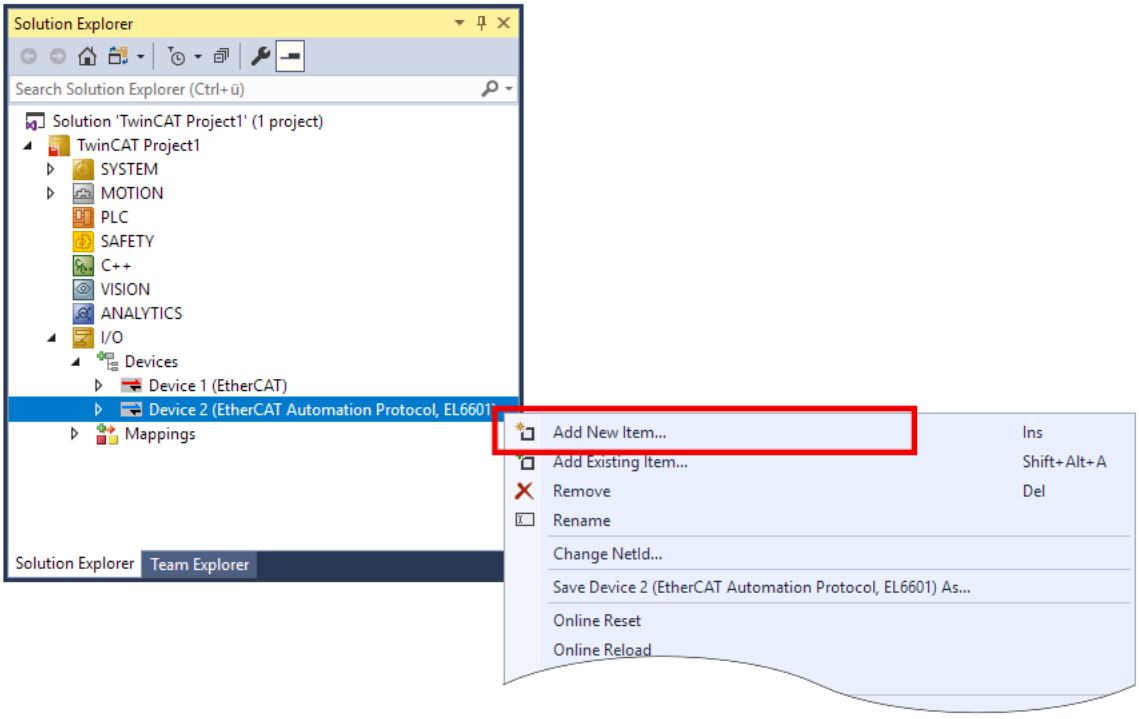


4. Select the device "EtherCAT Automation Protocol, EL6601" in Solution Explorer and check in the "Adapter" tab whether the EP6601-0002 was automatically entered under "Device Name".

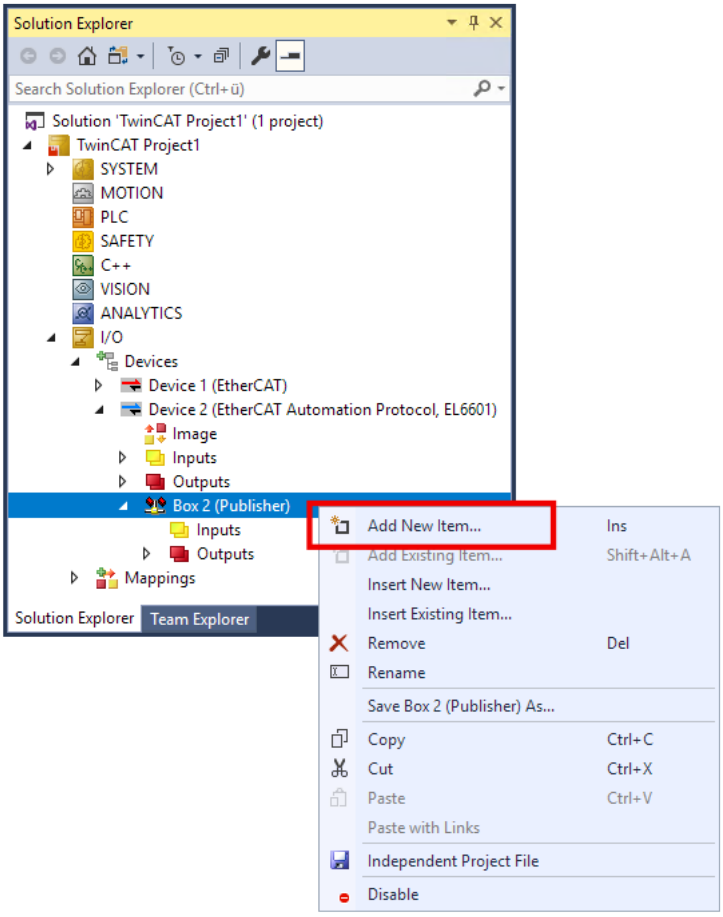


If not, select the box using the "Search" button.

5. Create Publisher and Subscriber by right-clicking on the device "EtherCAT Automation Protocol, EL6601" and selecting "Add New Item…".

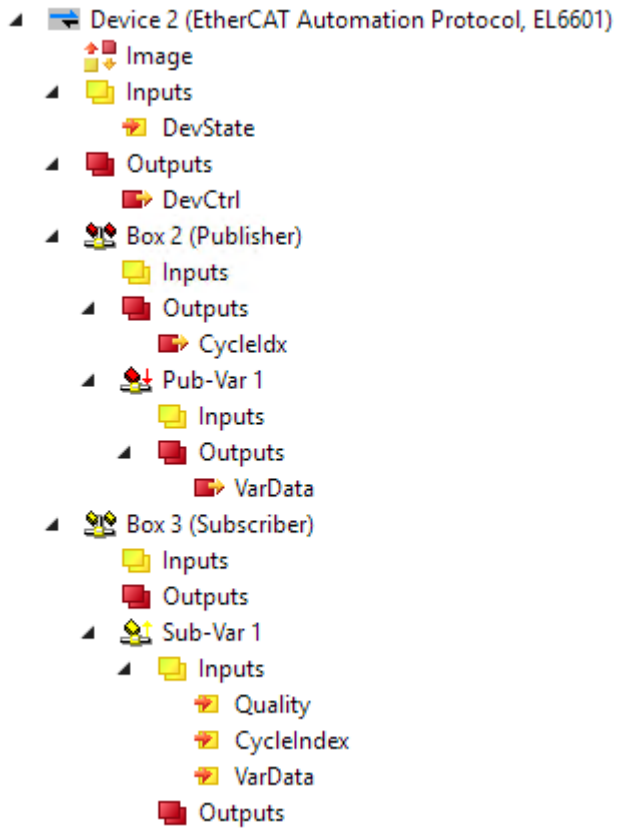


6. Create variables by right-clicking on a Publisher or Subscriber and selecting "Add New Item…".



5.3.3 Process data of the Automation Protocol device

The following figure shows an example of the process image with a Publisher variable and a Subscriber variable.



- "DevState":
  The Link state. Value 0 is the target state.
- "DevCtrl":
  Used, among other things, to suppress individual Publishers. See chapter Suppress publisher [▶ 48].

Publisher

- "CycleIdx":
  Must be incremented by the user for each transmission. See chapter Beckhoff network variables - Settings [▶ 41], section "Basic principles of Beckhoff network variables".
- "VarData":
  The data to be transmitted.

Subscriber

- "Quality":
  If a network variable arrives too late, "Quality" contains the measured delay. Unit: 100 µs.
  See also chapter Beckhoff network variables - Settings [▶ 41], section "Basic principles of Beckhoff network variables".
- "CycleIndex":
  Subscriber-side counterpart to CycleIdx.
- "VarData":
  Received data.

5.3.4 Beckhoff network variables - Settings

Beckhoff network variables (NWV) can be used for cyclic or acyclic sending of data between Windows-based PCs. In a device declared as a publisher (sender), such a network variable is received on the other side by a subscriber declared as the same type. As the name suggests, this data traffic is network-based, and the configuration is directly based on the protocols used.

A choice of two protocols is available:

- **MAC:** An ISO Layer 2 frame is sent with a sender and receiver MAC address, Ethertype 0x0806. An IP part with the destination IP address (e.g. 192.168.0.1) is not included. The telegram can therefore be further processed via a switch, but usually not via a router.
  MAC stands for media access control and in this case refers to the (unique) hardware address assigned to each Ethernet device during production. For sample, the Ethernet port of a Beckhoff PC might have the MAC ID 00:01:05:34:05.84, with "00:01:05" representing the Beckhoff ID and the rest assigned during production. The route of each Ethernet telegram between two Ethernet cable ends is determined by the source MAC and the destination MAC.
  The Ethernet telegram is identified as Beckhoff real-time Ethernet by the Ethertype 0x88A4. As a real-time Ethernet telegram (RT Ethernet) it bypasses the regular Windows TCP stack and is sent with higher priority, i.e. "immediately", via the specified Ethernet port of the PC.
  An option is available for configuring whether the sent telegram is received by all (broadcast), many (multicast) or a single subscriber (unicast).
- **UDP/IP**: The recipient is identified via an additional IP header in the Ethernet telegram. The UDP Ethernet frame can thus be further processed via a router.
  Once again, broadcast, multicast and unicast are available as options. The Ethernet telegram is identified as Beckhoff real-time Ethernet through the Ethertype 0x88A4 and treated as an RT protocol in the TwinCAT PC.
  In contrast to TCP, as a connection-less protocol UDP requires no acknowledgement of receipt for the message, i.e. the publisher does not know whether the subscriber has received the message. The ARP protocol [▶ 44] is therefore used for remote terminal monitoring in TwinCAT.

The telegram with the process data arrives at the recipient device (network port) via these addressing modes. In the Ethernet device/TwinCAT several transported process data are allocated via a variable ID

All network variables must be declared in the System Manager before they can be used.

The following intervention options are then available during operation:

- Sending of a configured network variable can be blocked dynamically
- The destination IP or destination MAC can be changed dynamically
- The variable ID "variable ID" can be changed dynamically
- The NWV content can be changed, but not the size (bit size)

Diagnostic variables on the publisher and subscriber side provide information about the connection quality.

If network variables are used, the temporal boundary conditions for the network topology used must be taken into account: in the case of IP addressing (routed) on a case-by-case basis several 100 ms communications cycle can be achieved, in the case of MAC addressing (switched) approximately 10 ms and less.

Diagnostic variable "quality"

If the processing tasks operate with different cycle times or the user changes the DataExchangeDivider, this must be taken into account in the analysis of the diagnostic variables. In conjunction with a fast Subscriber (e.g. 10 ms), a slow Publisher (e.g. 100 ms) leads to poor connection quality (as reported by the diagnostic variable "Quality").
Dynamic temporary blocking of sending a Publisher must also be taken into account. In this case the Subscriber registers poor quality.

Diagnostic variable "CycleIndex"

Please note the following information in order to decide whether you have to serve the variable CycleIndex.

Basic principles of Beckhoff network variables

- **Quality**:
  Time in [100 µs] by which arrival of the NWV at the Publisher was delayed.
  Relative arrival location:
  Input process image of the TwinCAT system
  Relative arrival time:
  Time at which the next cycle is loaded into the input image

  Note:
  The reason for determining the delay so precisely is that the NWVs are managed directly by the IO driver, independent of the cycle. Nevertheless, the data of an NWV that is delayed by a few percent of the cycle time will not be taken into account until the input process image is read during the next task cycle.

  Note EP6601-0002:
  Even with the EP6601-0002 the NWV arrival time is defined as the time when the data are available in the input process image of the RT device, not the time of arrival at the EP6601-0002 or in the input image of the EtherCAT device.

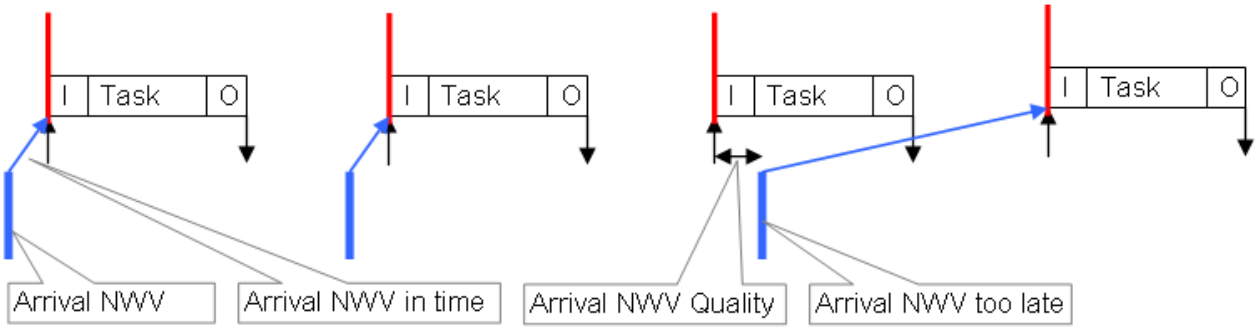


Fig. 9: Interrelationship between quality and delayed network variable

- **Variable ID**
  The variable ID (16 bit) is used for global identification of the individual process data. Therefore, an ID in the Publisher or Subscriber group may only be used once within a TwinCAT device, see Fig. *Sample for communication via network variables*: Publishers 1 and 2 on PC1 must have different IDs (10 and 8), although the same ID (8) may be used in Publisher 2 and Subscriber 1.

Selecting the variable ID

In order to achieve unambiguous allocation we recommend using different IDs for each data communication between connected PCs. Reason: In Fig. *Sample for communication via network variables,* PC2/Subscriber2 not only receives the designed ID=8-variable from PC1/Publisher2, but, since it is sent as a broadcast (!), it also receives the NWV from PC3/Publisher1. Differentiation is then no longer possible in PC2.

- Cycle Index
  The 16-bit cycle index is a counter sent by the Publisher together with the data. It is generally incremented with each transmission and can therefore be used as an indicator for flawless transfer. It can be read on the subscriber side as *CycleIndex*. Its appearance depends on the Publisher platform:
  - Publisher on a PC: The variable *CycleIndex* is not visible and is automatically and cyclically incremented by the System Manager
  - Publisher on an EP6601-0002: The variable *CycleIndex* is visible and must be incremented/served by the user such that it is not equal 0 on the subscriber side.

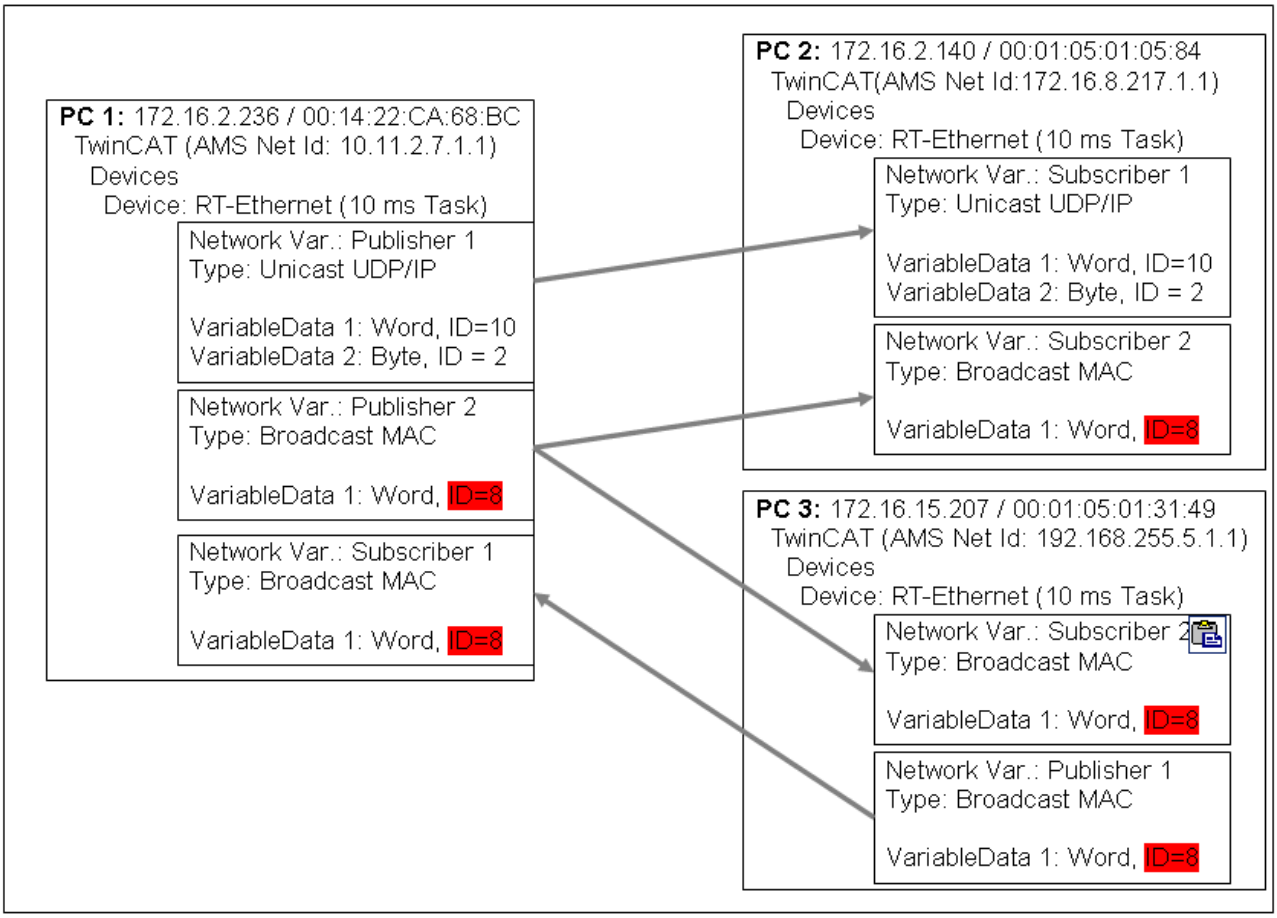


Fig. 10: Sample for communication via network variables

Data representation on different platforms

Please note that simple and complex data (WORD, ARRAYs, REAL, STRING, user-defined structures) are represented internally in a different manner on different platforms!

x86 platforms use byte-alignment, others (ARM) 2-byte or 4-byte alignment.
This means that if a complex structure is created in an x86/PC PLC project and an ARM PLC project, they can each have a different effective size and a different internal structure. (see Fig. *"Data representation e.g. x86 Systems vs. ARM Systems")*

In the sample, the structure in the CX (and hence the network variable to be created there) is larger than in the PC; also the word and real variables do not match each other because a variable can begin at any byte position in the PC, but only at every even-numbered one in the CX.

Consequences

Recommendation for structures that are identical on both end devices
- firstly, all 4-byte variables (must lie at an address that is divisible by 4)

- then all 2-byte variables (must lie at an address that is divisible by 2)

- then all 1-byte variables

Further recommendations

- if STRING(x) is used, the "EndOfString" zero is also interpreted as a character, otherwise x+1 must be divisible by 4

- the above rules also apply to sub-structures.

See also data type `STRUCT` in the TwinCAT 3 PLC manual: Link.

Consequences

Use of Bus Terminal Controllers (BICxxxx, BXxxxx)

Since the representation of floating point numbers (REAL) on Bus Terminal Controllers (BCxxxx, BXxxxx) differs from that in the x86, these cannot be transmitted. "SINT", for sample, can be used for signed values.

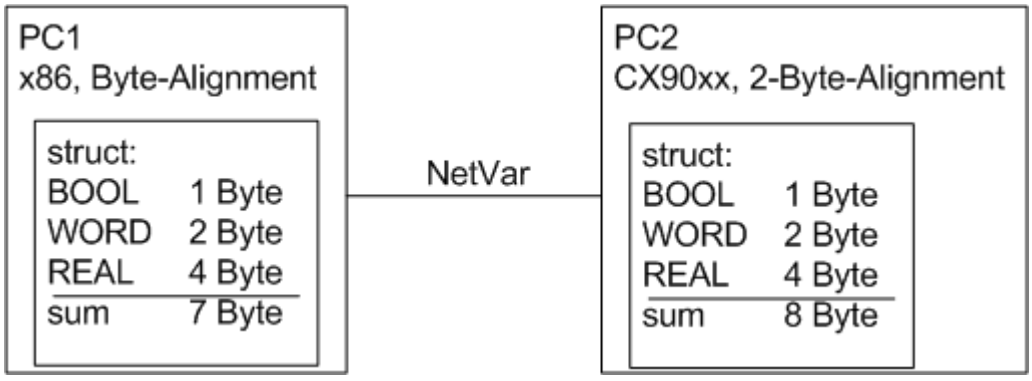


Fig. 11: Data representation e.g. x86 Systems vs. ARM Systems

Settings in the System Manager

Appearance of the variables

Depending on the platform used (PC or EP6601-0002), the publisher/subscriber will appear differently. A publisher/subscriber can be created

- On a PC network interface, see Fig. *Publisher settings - RT Ethernet*
- on an EP6601-0002, see Beckhoff network variables - Settings

The following settings options are available in TwinCAT:

Publisher, Box

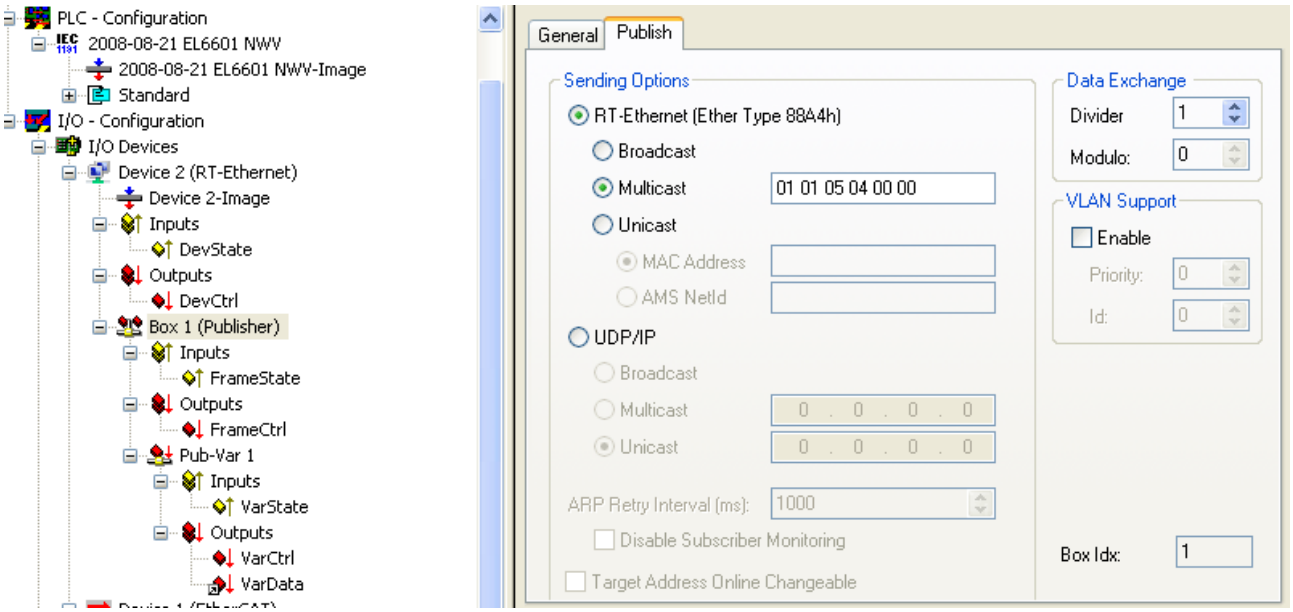


Fig. 12: Publisher RT Ethernet settings

RT Ethernet settings:

- **MAC-Broadcast:** Sent to all network devices, destination MAC FF:FF:FF:FF:FF:FF.
- **Multicast**: A destination MAC address becomes a multicast address if the first bit in the first byte of the MAC (the so-called group bit) is set. With the Beckhoff ID "00 01 05" the default target address "01 01 05 04 00 00" is formed, as shown in Fig. *Publisher RT Ethernet settings*.
  The MAC range 01:00:5E:00:00:00 to 01:00:5E:FF:FF:FF is intended for general multicast application, with the first 3 bytes specified by the IEEE and the last 3 bytes derived from the lower part of the IP address of the destination PC. The resulting destination MAC therefore never physically exists in the network. Instead, the destination network card detects Ethernet frames formed in this way as multicast frames sent to it, although the Ethernet port itself can have another, unique MAC address. Please refer to the relevant literature for further rules relating to the formation of multicast MAC/IP addresses.
- **Unicast**: Either direct entry of the destination MAC or via the AMS Net ID of the destination device, e.g. 123.456.123.456.1.1, in which case this route must be entered in the local AMS router (right-click on the TwinCAT icon in the taskbar --> Properties --> AMS router)

Use of broadcast and multicast

Network variables sent as broadcast or multicast at MAC or IP level can generate high network load (depending on the cycle time), since they are multiplied into the whole connected network. This may cause simple network devices such as printers to crash. With short cycle times all network traffic may become blocked. We strongly recommend using unicast addressing, taking into account variable identification, as described above.

Advanced Settings:

- Data exchange: The task cycle time * divider is the rhythm at which this network variable is sent. (not for EP6601-0002).
- VLAN support: In conjunction with manageable switches the Ethernet frame parameterized here can be assigned a fixed route via VLAN tagging (Virtual Local Area Network).

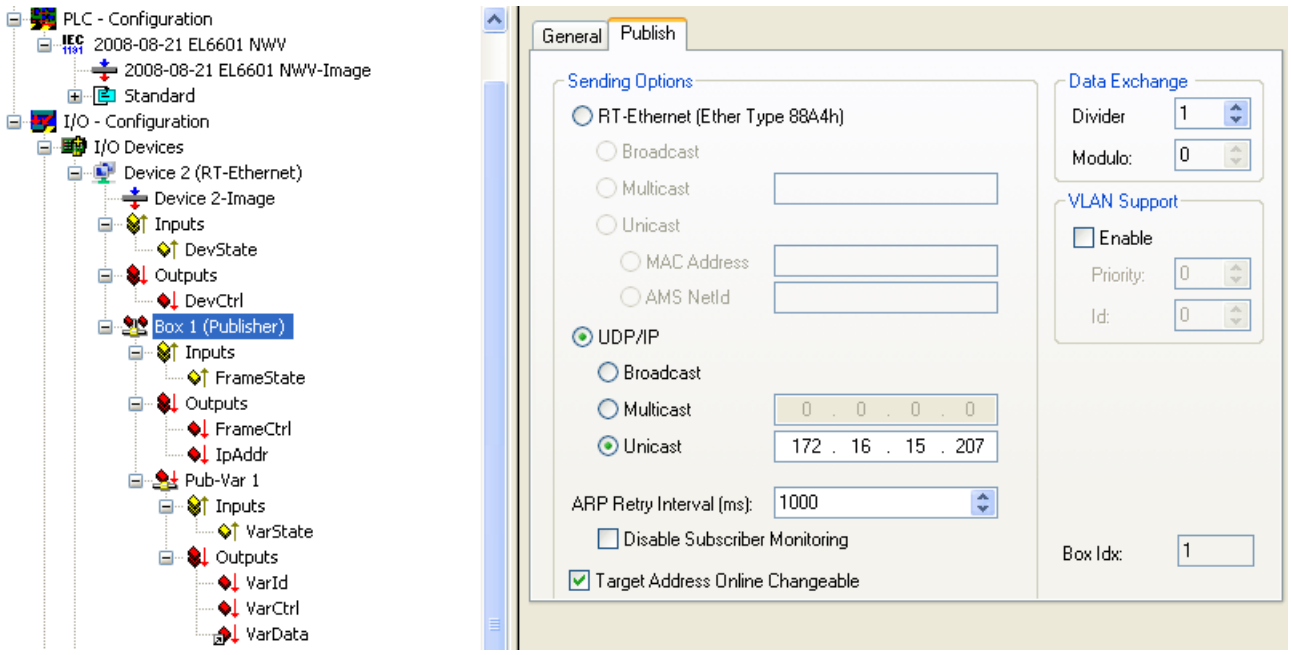


Fig. 13: Publisher settings - UDP/IP

UDP/IP settings - the addressing technique of the IP network layer with IP addresses is used. UDP is a connection-less protocol without feedback.

- Broadcast: Sent to all device with destination IP (v4) 255.255.255.255
- Multicast: The destination IP must be specified, see notes on MAC multicast
- Unicast: Specify the target device (e.g.: 192.168.0.1), making sure that it can be reached through the subnet mask

Use of broadcast and multicast

Network variables sent as broadcast or multicast at MAC or IP level can generate high network load (depending on the cycle time), since they are multiplied into the whole connected network. This may cause simple network devices such as printers to crash. With short cycle times all network traffic may become blocked. We strongly recommend using unicast addressing, taking into account variable identification, as described above.

Advanced Settings:

- "ARP Retry Interval": In order to ascertain the presence of the recipient, the publisher sends an ARP request to the target device at these intervals. If the network administration of the recipient is operational, it sends an ARP reply. This is only meaningful with unicast.
  In the event of an error bit 3 is set (0x0004) in the diagnostic FrameState variable.
  Note: ARP handling (ARP = Address Resolution Protocol: allocation of hardware/MAC addresses to network addresses [IP]) is managed by the operating system (Windows).
- "Disable Subscriber Monitoring": deactivates the procedure described above.
- "Target Address changeable": In this case the destination IP can be changed dynamically.

Publisher, Variable

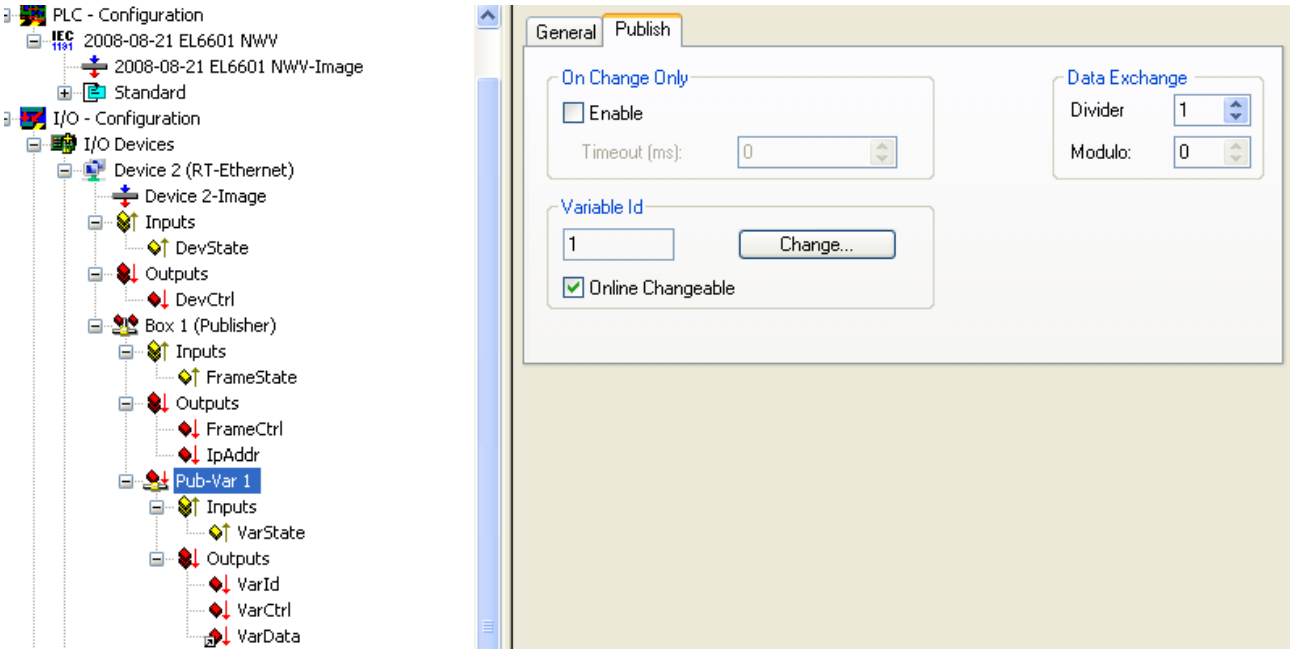


Fig. 14: Publisher variable settings

Settings:

- "Variable ID": Identification number with which the variable is sent. Can be changed online via PLC where appropriate.
- "Data exchange": see above (not for EP6601-0002).
- "On change only": NWV is only sent if the value changes (not for EP6601-0002).

Subscriber, Box

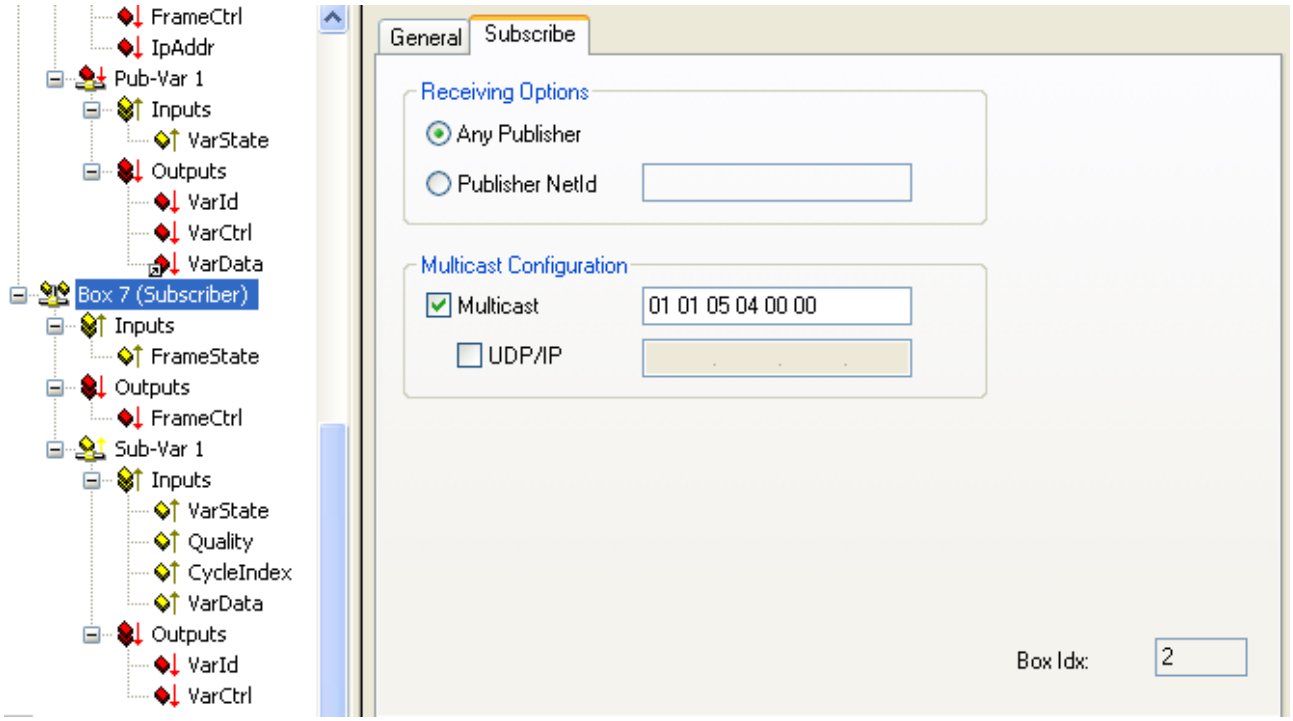


Fig. 15: Subscriber settings

Settings:

- "Receiving Options": Only permits NWVs from a certain publisher for this subscriber

- "Multicast Configuration": ditto.

Process data:

- "VarId": If activated, the variable ID can be modified online

Subscriber, variable

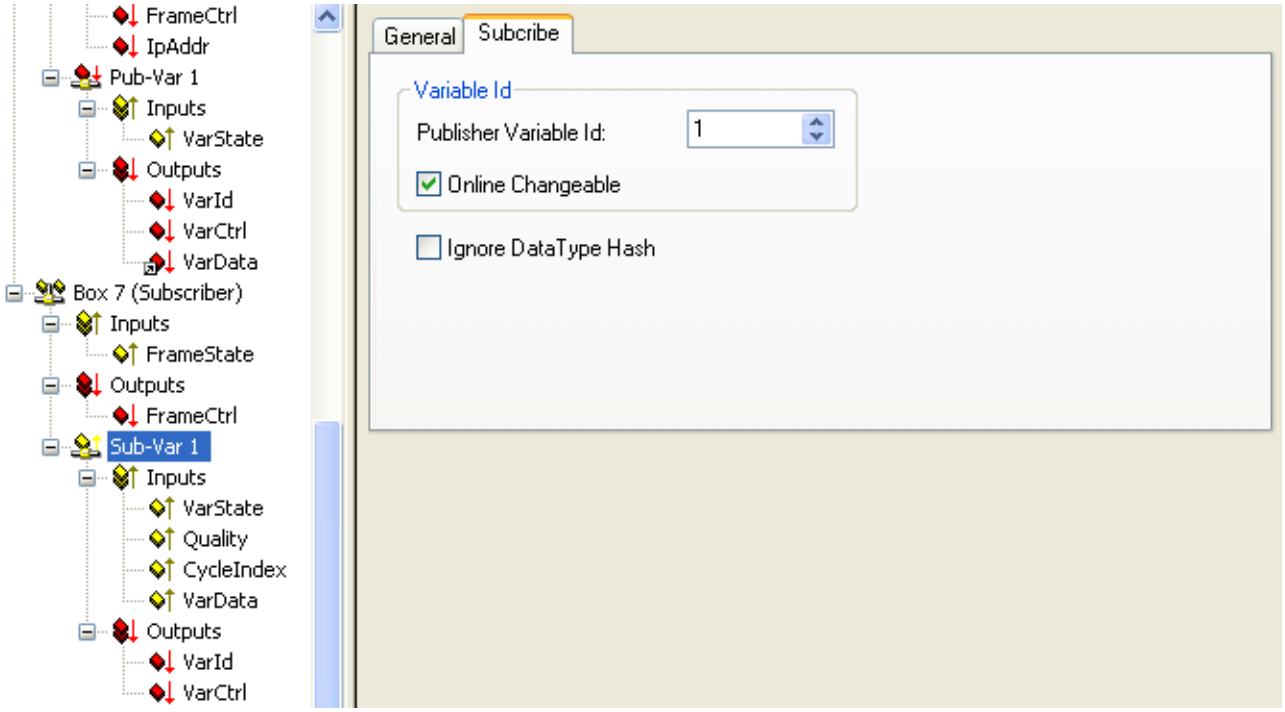


Fig. 16: Subscriber variable settings

Settings:

- "Variable ID": Only permits NWVs with a certain ID for this subscriber. Can be changed dynamically via PLC where appropriate.
- "Ignore Data Type Hash": Hash calculation is currently not supported

Process data:

- "Quality": See explanatory notes above.
- "CycleIndex": This index is incremented with each successful transfer, IF this is done by the opposite side, i.e. the publisher. If the publisher is an EP6601-0002, the user must increment *CycleIdx* there.
- "VarData": Transferred data.

5.3.5 Suppress publisher

If the EP6601-0002 is operated with a short cycle time and with configured Publishers, this can place a high loading on the connected network. The EP6601-0002 can therefore be configured so that the variable *DevCtrl* blocks the sending of individual Publishers. The object 0xF800:02 must be occupied in the CoE (CanOpenOverEtherCAT) for this purpose.

Groups of Publisher boxes can be blocked by setting appropriate bits (Publisher frames). The topmost 4 bits (the high nibble of high byte) from 0xF800:02 specify the granularity of the groups 1..15, i.e. how many Publisher frames are handled together as one group:

The upper 8 bits of *DevCtrl* (format: 16 bits) then block the transmission of the Publisher frames located in the corresponding group in the current cycle.

High byte of *DevCtrl*:

- 0 = no blocking
- n = each bit in *DevCtrl* means an n-group of Publishers, n from [1..31]

It This means that a maximum of 8 groups of Publishers can be blocked.

Example:

DevCtrl.10 = true and 0xF800:02 = 0x2000 signifies that the third group will be blocked in this PLC cycle. One group consists of 2 Publisher frames, which means that in this case all the Publisher variables that are located in Publisher frames 5 and 6 will not be transmitted.

NOTICE

Suppressing individual Publishers

The structure of a "Publisher" as a Publisher Box in the System Manager is
- an Ethernet frame, this contains
- n Publishers
The individual bits in DevCtrl each block a group of Publisher frames.

The success achieved in this way can be checked using a network monitor such as Wireshark, for example.

Changes in the CoE

The CoE contents (if writable) can be changed online using the TwinCAT System Manager. However, after the terminal or the EtherCAT system is restarted, this change will no longer be present; default values will apply. As a result, any permanent change must be stored in the terminal's CoE startup list.

Note: bit counting in this documentation starts at 0: value.0, value.1, ...

5.3.6 Filter subscribers

Depending on the configuration of the Ethernet network, the EP6601-0002 used in the network will receive a low or high number of Publisher telegrams. At startup, the EP6601-0002 is configured by the EtherCAT master to the Subscriber variables to be received by it: Source AMS Net ID and ID of the variables are loaded into the CoE for each Subscriber. The CoE objects 0x60n0:01 and 0x60n0:02 then contain the AmsNetId and VariableID to be checked. The EP6601-0002 can therefore filter on the incoming Publisher ID and compare it with its own Subscriber ID. For this purpose the Publisher variables contained in the Ethernet frames received are disassembled and checked individually.

If an incoming Subscriber

- corresponds to a configured AMS Net ID and Variables ID, then the contents are transferred to EtherCAT via PDO.
- does NOT correspond to the above, then the contents are transferred by default to the acyclic mailbox interface for transmission to the master.

This is the standard setting of the EP6601-0002.

The second way generates a high acyclic EtherCAT transport load, because Subscribers received by the EP6601-0002 are transported that should not be transported by this EP6601-0002. For this reason the Subscriber filter can be enabled by the **CoE entry 0xF800:02 = 0x0100 (bit 8 = TRUE)**. The Subscriber data that do not correspond to the AmsNetID/Variables ID filter are then discarded in the terminal and are not transferred to the mailbox.

Filter Subscribers

It is recommended to enable the Subscriber filter.
As the EP6601-0002 is reinitialized with every INIT-OP transition, the CoE entry mentioned must be set in the StartUp list.

Note: bit counting in this documentation starts at 0: value.0, value.1, ...

6 Application samples

NOTICE

See documentation on EL6601

The documentation for EL6601 contains sample programs and application examples that can be transferred to EP6601-0002.

Documentation on EL6601

7 Diagnostics

Online diagnostics

The following objects are available for initial diagnostic in the CoE directory:

- 0xFA01:01: "Rx Packets" (arriving at EP6601-0002).
- 0xFA01:02: "Tx Packets" (outgoing from EP6601-0002).

The values can be read from the controller using PLC function blocks (FB_EcCoeSdoRead in TcEtherCAT.lib).

Overrun

If the EP6601-0002 receives more Ethernet frames than can be transmitted to the controller via EtherCAT, the telegrams are discarded.

Overrun case

In the case of an overrun, the following measures can be taken to counteract this:

- Activating the Subscriber Filter [▶ 17] in the EP6601-0002 concerned
- Increasing/slowing down the cycle time of the Publisher
- Suppressing temporarily Publisher transmission or modulo in the System Manager
- Reducing/accelerating the EtherCAT cycle time of the Subscriber, so that more data are fetched by the EP6601-0002

8 CoE objects

Parameterization

You can parameterize the box via the "CoE - Online" tab in TwinCAT.

EtherCAT XML Device Description

The presentation matches that of the EtherCAT XML Device Description.
Recommendation: download the latest XML file from https://www.beckhoff.com and install it according to the installation instructions.

8.1 Objects for parameterization

F800 Device settings

Index (hex)	Name	Meaning	Data type	Flags	Default
F800:0	Device settings		USINT	-	0x04 (4_{dec})
F800:01	General	Possible values: • 0x0000: standard operation • 0x2000: transmission of EoE frames in the OP state is blocked. • 0x8000: transmission of EoE frames is blocked.	UINT	RW	0
F800:02	NetVars	This switch determines whether received subscriber data from frames with 0x88A4 in the header, that have not passed the subscriber filter, will be transported further via EoE/Mailbox to the EtherCAT Master. Possible values: • 0x0000: default, subscriber data are forwarded via EoE. • 0x0100: subscriber data are being discarded. See also Chapter Filter subscribers [▶ 49].	UINT	RW	0x0100 (256_{dec})
F800:03	Res.	no function	UINT	RW	0
F800:04	Status	no function	UINT	RW	0

8.2 Objects for operation

FA01 MAC info

Index (hex)	Name	Meaning	Data type	Flags	Default
FA01:0	MAC info		USINT	-	0x04 (4_{dec})
FA01:01	Rx Packets	Number of Ethernet telegrams received.	UDINT	RO	0
FA01:02	Tx Packets	Number of Ethernet telegrams sent.	UDINT	RO	0
FA01:03	Rx Frames Dropped	no function	UDINT	RO	0
FA01:04	Reserved	no function	UDINT	RO	0

FA02 AddrServer info

Index (hex)	Name	Meaning	Data type	Flags	Default
FA02:0	AddrServer info		USINT	-	0x01 (1_{dec})
FA02:01	Remote MaC-ID	MAC-ID of the Ethernet device that was assigned an IP address via the DHCP functionality of the EP6601-0002. See chapter IP address assignment [▶ 31].	ARRAY [0..5] OF BYTE	RO	00 00 00 00 00 00

8.3 Standard objects

1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: the Lo-Word contains the used CoE profile (5001). The Hi-Word contains the module profile according to the modular device profile.	UDINT	-	0x00001389 (5001_{dec})

1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave.	STRING	-	EP6601

1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave.	STRING	-	-

100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave.	STRING	-	-

100B Bootloader version

Index (hex)	Name	Meaning	Data type	Flags	Default
100B:0	Bootloader version	Bootloader version of the EtherCAT slave.	STRING	RO	-

1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore the delivery state.	USINT	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If you set this object to "0x64616F6C", all backup objects are reset to the delivery state.		RW	0

1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave.	USINT	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave.	UDINT	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave.	UDINT	RO	0x19C94052 (432619602_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description.	UDINT	RO	0
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0.	UDINT	RO	0

10E2 Manufacturer-specific Identification Code

Index (hex)	Name	Meaning	Data type	Flags	Default
10E2:0	Manufacturer-specific Identification Code	Vendor-specific identification code.	USINT	RO	0x01 (1_{dec})

10F0 Backup parameter handling

Index (hex)	Name	Meaning	Data type	Flags	Default
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries.	USINT	RO	0x01 (1_{dec})
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave.	UDINT	RO	0

1C00 Sync manager type

Index (hex)	Name	Meaning	Data type	Flags	Default
1C00:0	Sync manager type	Using the Sync Managers	UINT8	RO	0x04 (4_{dec})
1C00:01	SubIndex 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1_{dec})
1C00:02	SubIndex 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2_{dec})
1C00:03	SubIndex 003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3_{dec})
1C00:04	SubIndex 004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4_{dec})

1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: cycle time of the local timer • Synchron with SM 2 Event: cycle time of the master DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: Free Run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC-Mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 Event (only DC mode) Bit 14 = 1: dynamic times (measurement through writing of 1C32:08)	UINT16	RO	0x0003 (3_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0003D090 (250000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0
1C32:07	Minimum delay time	Delay due to signal propagation times in hardware, minimum value. Unit: ns	UINT32	RO	0
1C32:08	Get Cycle Time	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started Entries 1C32:03, 1C32:05, 1C32:06, 1C32:09, 1C33:03, 1C33:06, 1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only).	UINT	RO	0
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0

1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	USINT	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event 34: Synchron with SM 2 Event (outputs available)	UINT	RW	0
1C33:02	Cycle time	Cycle time (in ns): • Synchron with SM 2 Event: cycle time of the master DC-Mode: SYNC0/SYNC1 Cycle Time	UDINT	RW	0
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, DC mode only)	UDINT	RO	0
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: Free Run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC-Mode is supported • Bit 4-5 = 01: Input shift through local event (outputs available) • Bit 4-5 = 10: Input shift with SYNC1 event (no outputs available) Bit 14 = 1: dynamic times (measurement through writing of 1C33:08)	UINT	RO	0x0002 (2_{dec})
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UDINT	RO	0
1C33:06	Calc and copy time	Time between reading of the inputs and the inputs being available for the master (in ns, only DC mode)	UDINT	RO	0
1C33:07	Minimum delay time	Delay due to signal propagation times in hardware, minimum value. Unit: ns	UDINT	RO	0
1C33:08	Get Cycle Time	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 1C33:03, 1C33:06, 1C33:07, 1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT	RW	0
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, DC mode only)	UDINT	RO	0
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT	RO	0
1C33:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT	RO	0
1C33:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT	RO	0
1C33:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOL	RO	0

F000 Modular Device Profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular Device Profile	General information for the Modular Device Profile.	USINT	RO	0x02 (2_{dec})
F000:01	Index distance	Index distance of the objects of the individual channels.	UINT	RO	0x0010 (16_{dec})
F000:02	Maximum number of modules	Number of channels.	UINT	RO	0

F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0	Code word	reserved	UDINT	RW	0

F081 Download revision

Index (hex)	Name	Meaning	Data type	Flags	Default
F081:0	Download revision		USINT	RO	0x01 (1_{dec})
F081:01	Revision number	Revision number of the module. Relevant as a startup list entry for compatibility.	UDINT	RW	0x00120002 (1179650_{dec})

9 Appendix

9.1 General operating conditions

Protection rating according to IP code

The degrees of protection are defined and divided into different classes in the IEC 60529 standard (EN 60529). Degrees of protection are designated by the letters "IP" and two numerals: **IPxy**

- Numeral x: Dust protection and contact protection
- Numeral y: Protection against water

x	Meaning
0	Not protected
1	Protected against access to dangerous parts with the back of the hand. Protected against solid foreign objects of 50 mm Ø
2	Protected against access to dangerous parts with a finger. Protected against solid foreign objects of 12.5 mm Ø
3	Protected against access to dangerous parts with a tool. Protected against solid foreign objects of 2.5 mm Ø
4	Protected against access to dangerous parts with a wire. Protected against solid foreign objects of 1 mm Ø
5	Protection against access to dangerous parts with a wire. Dust-protected. Ingress of dust is not prevented completely, although the quantity of dust able to penetrate is limited to such an extent that the proper function of the device and safety are not impaired
6	Protection against access to dangerous parts with a wire. Dust-tight. No ingress of dust

y	Meaning
0	Not protected
1	Protection against vertically falling water drops
2	Protection against vertically falling water drops when enclosure tilted up to 15°
3	Protection against spraying water. Water sprayed at an angle of up to 60° on either side of the vertical shall have no harmful effects
4	Protection against splashing water. Water splashed against the enclosure from any direction shall have no harmful effects
5	Protection against water jets.
6	Protection against powerful water jets.
7	Protected against the effects of temporary immersion in water. Ingress of water in quantities causing harmful effects shall not be possible when the enclosure is immersed in water at a depth of 1 m for 30 minutes

Chemical resistance

The resistance refers to the housing of the IP67 modules and the metal parts used. In the table below you will find some typical resistances.

Type	Resistance
Water vapor	unstable at temperatures > 100 °C
Sodium hydroxide solution (ph value > 12)	stable at room temperature unstable > 40 °C
Acetic acid	unstable
Argon (technically pure)	stable

Key

- resistant: Lifetime several months
- non inherently resistant: Lifetime several weeks
- not resistant: Lifetime several hours resp. early decomposition

9.2 Accessories

Mounting

Ordering information	Description	Link
ZS5300-0011	Mounting rail	Website

Cables

A complete overview of pre-assembled cables can be found on the Beckhoff website: Link.

Ordering information	Description	Link
ZK1090-3xxx-xxxx	EtherCAT cable M8, green	Website
ZK1093-3xxx-xxxx	EtherCAT cable M8, yellow	Website
ZK1090-6xxx-xxxx	Ethernet cable M12, green	Website
ZK2020-3xxx-xxxx	Power cable M8, 4-pin	Website

Labeling material, protective caps

Ordering information	Description
ZS5000-0010	Protective cap for M8 sockets, IP67 (50 pieces)
ZS5000-0020	Protective cap for M12 sockets, IP67 (50 pcs.)
ZS5100-0000	Inscription labels, unprinted, 4 strips of 10
ZS5000-xxxx	Printed inscription labels on enquiry

Tools

Ordering information	Description
ZB8801-0000	Torque wrench for plugs, 0.4…1.0 Nm
ZB8801-0001	Torque cable key for M8 / wrench size 9 for ZB8801-0000
ZB8801-0002	Torque cable key for M12 / wrench size 13 for ZB8801-0000
ZB8801-0003	Torque cable key for M12 field assembly / wrench size 18 for ZB8801-0000

Further accessories

Further accessories can be found in the price list for fieldbus components from Beckhoff and online at https://www.beckhoff.com.

9.3 Version identification of EtherCAT devices

9.3.1 General notes on marking

Designation

A Beckhoff EtherCAT device has a 14-digit designation, made up of

- family key
- type
- version
- revision

Example	Family	Type	Version	Revision
EL3314-0000-0016	EL terminal 12 mm, non-pluggable connection level	3314 4-channel thermocouple terminal	0000 basic type	0016
ES3602-0010-0017	ES terminal 12 mm, pluggable connection level	3602 2-channel voltage measurement	0010 high-precision version	0017
CU2008-0000-0000	CU device	2008 8-port fast ethernet switch	0000 basic type	0000

Notes

- The elements mentioned above result in the **technical designation**. EL3314-0000-0016 is used in the example below.
- EL3314-0000 is the order identifier, in the case of “-0000” usually abbreviated to EL3314. “-0016” is the EtherCAT revision.
- The **order identifier** is made up of
  - family key (EL, EP, CU, ES, KL, CX, etc.)
  - type (3314)
  - version (-0000)
- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.
  From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. *“EL2872 with revision 0022 and serial number 01200815”*.
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

9.3.2 Version identification of IP67 modules

The serial number/ data code for Beckhoff IO devices is usually the 8-digit number printed on the device or on a sticker. The serial number indicates the configuration in delivery state and therefore refers to a whole production batch, without distinguishing the individual modules of a batch.

Structure of the serial number: **KK YY FF HH**	Example with serial number 12 06 3A 02:
KK - week of production (CW, calendar week)	12 - production week 12
YY - year of production	06 - production year 2006
FF - firmware version	3A - firmware version 3A
HH - hardware version	02 - hardware version 02

Exceptions can occur in the **IP67 area**, where the following syntax can be used (see respective device documentation):

Syntax: D ww yy x y z u

D - prefix designation
ww - calendar week
yy - year
x - firmware version of the bus PCB
y - hardware version of the bus PCB
z - firmware version of the I/O PCB
u - hardware version of the I/O PCB

Example: D.22081501 calendar week 22 of the year 2008 firmware version of bus PCB: 1 hardware version of bus PCB: 5 firmware version of I/O PCB: 0 (no firmware necessary for this PCB) hardware version of I/O PCB: 1

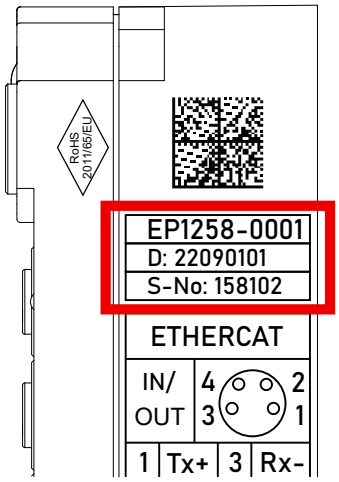


Fig. 17: EP1258-00001 IP67 EtherCAT Box with batch number/DateCode 22090101 and unique serial number 158102

9.3.3 Beckhoff Identification Code (BIC)

The Beckhoff Identification Code (BIC) is increasingly being applied to Beckhoff products to uniquely identify the product. The BIC is represented as a Data Matrix Code (DMC, code scheme ECC200), the content is based on the ANSI standard MH10.8.2-2016.

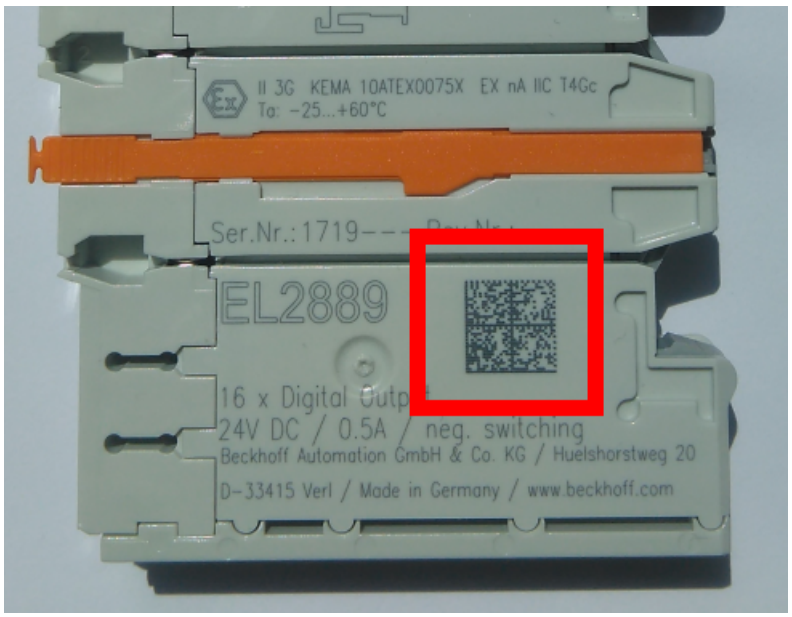


Fig. 18: BIC as data matrix code (DMC, code scheme ECC200)

The BIC will be introduced step by step across all product groups.

Depending on the product, it can be found in the following places:

- on the packaging unit
- directly on the product (if space suffices)
- on the packaging unit and the product

The BIC is machine-readable and contains information that can also be used by the customer for handling and product management.

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, spaces are added to it.

Following information is possible, positions 1 to 4 are always present, the other according to need of production:

Position	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	SBTN	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971, 2*K183
...					

Further types of information and data identifiers are used by Beckhoff and serve internal processes.

Structure of the BIC

Example of composite information from positions 1 to 4 and with the above given example value on position 6. The data identifiers are highlighted in bold font:

1P072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

Fig. 19: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, position 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTICE

This information has been carefully prepared. However, the procedure described is constantly being further developed. We reserve the right to revise and change procedures and documentation at any time and without prior notice. No claims for changes can be made from the information, illustrations and descriptions in this documentation.

9.3.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

The Beckhoff Identification Code (BIC) is applied to the outside of Beckhoff products in a visible place. If possible, it should also be electronically readable.

The interface that the product can be electronically addressed by is crucial for the electronic readout.

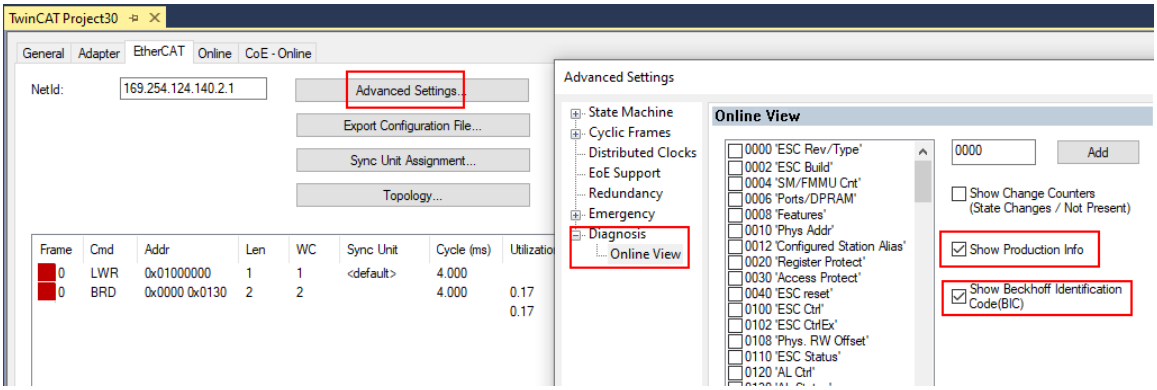
EtherCAT devices (IP20, IP67)

All Beckhoff EtherCAT devices have an ESI-EEPROM which contains the EtherCAT identity with the revision number. The EtherCAT slave information, also colloquially known as the ESI/XML configuration file for the EtherCAT master, is stored in it. See the corresponding chapter in the EtherCAT system manual (Link) for the relationships.

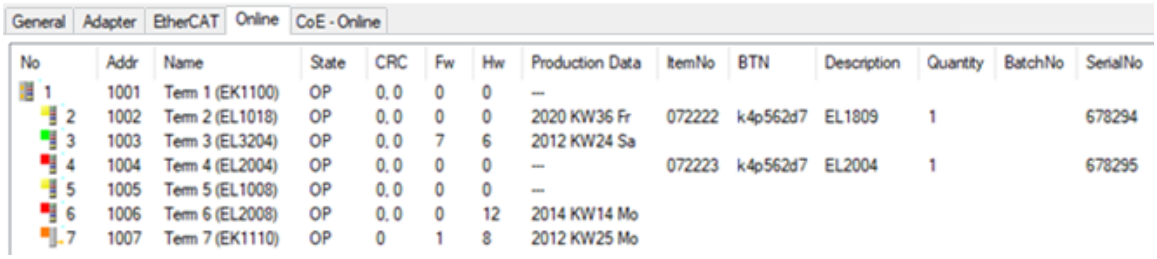
Beckhoff also stores the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

The user can electronically access the eBIC (if present) as follows:

- With all EtherCAT devices, the EtherCAT master (TwinCAT) can read the eBIC from the ESI-EEPROM
  - From TwinCAT 3.1 build 4024.11, the eBIC can be displayed in the online view.
  - To do this, check the "Show Beckhoff Identification Code (BIC)" checkbox under EtherCAT → Advanced Settings → Diagnostics:

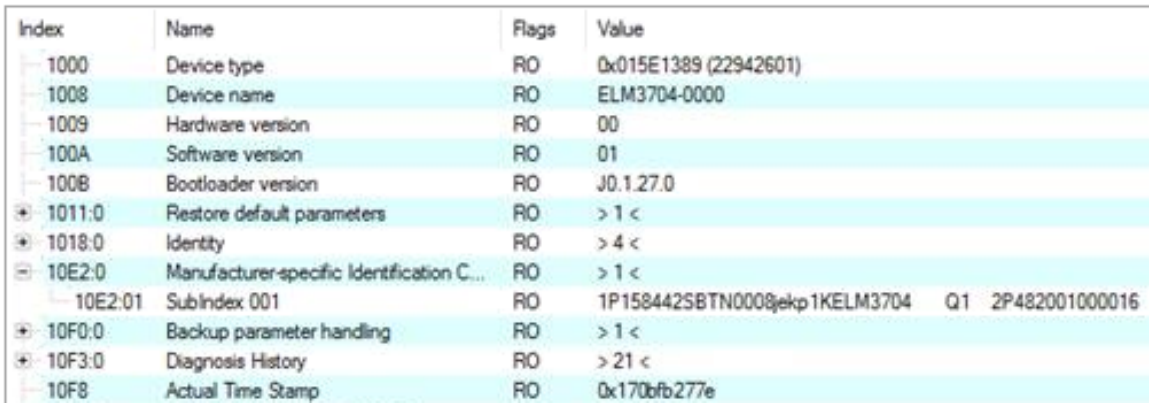


  - The BTN and its contents are then displayed:



  - Note: As shown in the figure, the production data HW version, FW version, and production date, which have been programmed since 2012, can also be displayed with "Show production info".
  - Access from the PLC: From TwinCAT 3.1. build 4024.24, the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0.
- EtherCAT devices with a CoE directory may also have the object 0x10E2:01 to display their own eBIC, which can also be easily accessed by the PLC:

- The device must be in PREOP/SAFEOP/OP for access:



Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	SubIndex 001	RO	1P158442SBTN0008jekp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

  - The object 0x10E2 will be preferentially introduced into stock products in the course of necessary firmware revision.
  - From TwinCAT 3.1. build 4024.24, the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0
- The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2_Utilities* as of TwinCAT 3.1 build 4024.24
  - F_SplitBIC: The function splits the Beckhoff Identification Code (BIC) sBICValue into its components using known identifiers and returns the recognized substrings in the ST_SplittedBIC structure as a return value
  - BIC_TO_BTN: The function extracts the BTN from the BIC and returns it as a return value
- Note: If there is further electronic processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
  The new BIC information is written as an additional category in the ESI-EEPROM during device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored using a category in accordance with the ETG.2010. ID 03 tells all EtherCAT masters that they may not overwrite these data in the event of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see here. The EEPROM therefore requires approx. 50..200 bytes of memory.
- Special cases
  - If multiple hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC information.
  - If multiple non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC information.
  - If the device consists of several sub-devices which each have their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

9.4 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.

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 her internet pages: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Support

The Beckhoff 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
web:	www.beckhoff.com/support

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
web:	www.beckhoff.com/service

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

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

More Information:
www.beckhoff.com/ep6601-0002

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