

**Commissioning instructions
FTS with EtherCAT interface,
firmware 2.1.0
Force-torque sensor**

Translation of original commissioning
instructions

Imprint

Copyright:

This manual is protected by copyright. The author is SCHUNK SE & Co. KG.
All rights reserved.

Technical changes:

We reserve the right to make technical improvements.

Document number: 1634813-EC FW 2.1.0

Version: 04.00 | 1/15/2026 | en-US

Dear Customer

Dear Customer,

Thank you for putting your trust in our products and our family-owned company, the leading technology supplier of robots and production machines.

Our team is always available to answer any questions on this product and other solutions. We look forward to your challenging questions. We will find a solution!

Best regards,

Your SCHUNK team

Customer Management

Tel. +49-7133-103-2503

Fax +49-7133-103-2189

cmg@de.schunk.com



Please read the operating manual in full and keep it close to the product.

Table of contents

| | |
|---|-----------|
| 1 General | 4 |
| 1.1 About this document | 4 |
| 2 Communication | 5 |
| 2.1 Data exchanges | 5 |
| 2.1.1 Cyclical data exchange | 5 |
| 2.1.2 Acyclical data exchange | 9 |
| 3 Module functions | 10 |
| 3.1 Booting and restarting | 10 |
| 3.1.1 Booting and establishing operational readiness..... | 10 |
| 3.1.2 Restart..... | 10 |
| 3.2 Taring functions..... | 11 |
| 3.2.1 Tare | 11 |
| 3.2.2 Reset tare..... | 11 |
| 3.3 Switching tool settings | 12 |
| 3.4 Switching the noise reduction filter..... | 13 |
| 4 System parameters | 14 |
| 4.1 Value ranges | 14 |
| 4.2 Parameter list..... | 14 |
| 4.2.1 Sensor | 15 |
| 4.2.2 Interface box..... | 20 |
| 5 Start-up | 23 |
| 5.1 Safety | 23 |
| 5.2 System integration | 23 |
| 5.3 Commissioning with Beckhoff "TwinCAT 3 ®" software for EtherCAT | 24 |
| 5.4 Configuring Ethernet over EtherCAT with TwinCAT..... | 29 |
| 5.5 SCHUNK Control Center – App FTS..... | 34 |
| 6 Appendix | 36 |
| 6.1 Control word | 36 |
| 6.2 Status double word | 37 |
| 6.3 Brands..... | 38 |

1 General

1.1 About this document

This manual describes the commissioning as well as the operating and parameterization options for a Force-Torque Sensor with the following interfaces:

- EtherCAT (EC)

Validity

This version of the manual describes the functions for firmware versions 2.1.0 of the interface box and sensor.

The firmware version can be read. For information on the corresponding parameter, see ▶ 4.2.1 [15] and ▶ 4.2.2 [20].

Conventions

The following conventions apply to this guide:

- The Force-Torque sensor is referred to below as a "Module".
- Actions initiated by the user that the module is to perform are hereafter referred to as a "commands" and can be initiated in the control byte via control bits.
- Identification of parameters: <Parameter>
- Identification of events: WARNING
- Page number in references: [▶ 4]

NOTE: The illustrations in this manual are intended to provide a basic understanding and may deviate from the actual version.

Applicable documents

- General terms of business *
- Assembly and Operating Manual of the module *

The documents labeled with an asterisk (*) can be downloaded from schunk.com/downloads.

2 Communication

2.1 Data exchanges

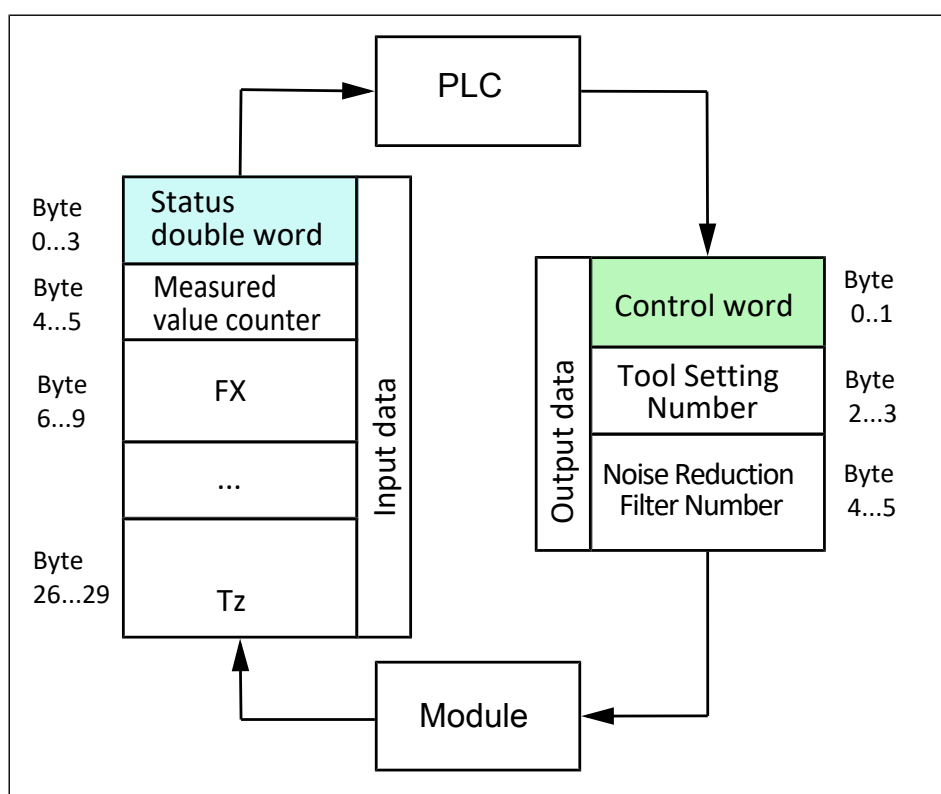
Integrated fieldbus interfaces can be used to exchange data cyclically and acyclically between the module and the controller.

Communication types The module supports the following communication types:

- Full EtherCAT Slave with 4 FMMUs (Fieldbus Memory Management Unit) and 4 Sync Managers
- CANopen over EtherCAT (CoE)
- Ethernet over EtherCAT (EoE)
- 10/100 Mbit full/half duplex Ethernet

2.1.1 Cyclical data exchange

For cyclical data exchange, a fixed data frame for input and output data is defined. The data frame is based on the use of data double words, control and status bytes. The output data frame is set to a data length of six bytes and the input data frame to a data length of 34 bytes.



For reasons of clarity, the two reserved double words (bytes 30 to 33) have been omitted from the input data in the diagram.

For further information on data transmission and interpretation, see the following sections.

2.1.1.1 Cyclical output data

The cyclical output data is transferred from the control unit to the module and thus commands are sent to the module.

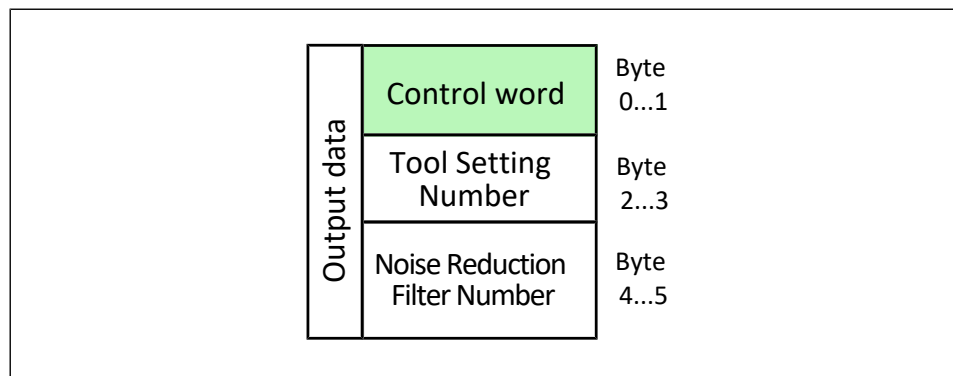
Implementation of the control commands

Commands sent to the module may or may not be permitted.

- Permitted commands are implemented by the module. This is indicated to the PLC by setting the "Command Processed Toggle" status bit.
- Illegal commands are not implemented. This is indicated to the PLC by setting the "Command Error" status bit.

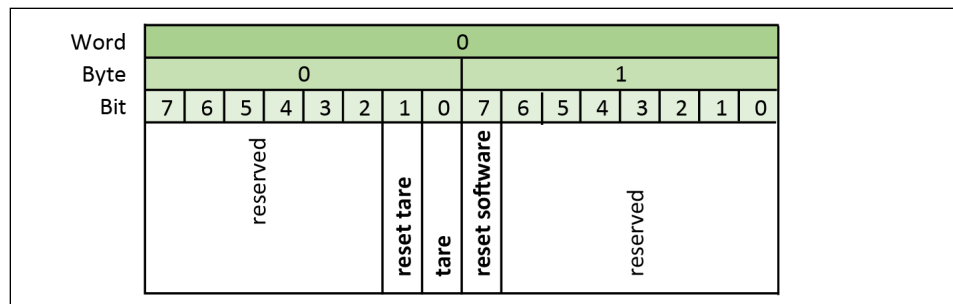
Data frame

The data frame of cyclical output data consists of the control word, the numbers for the tool settings and the noise suppression filter.



Control byte

The control byte is transferred in bytes 0 and 1 of the cyclical output data.



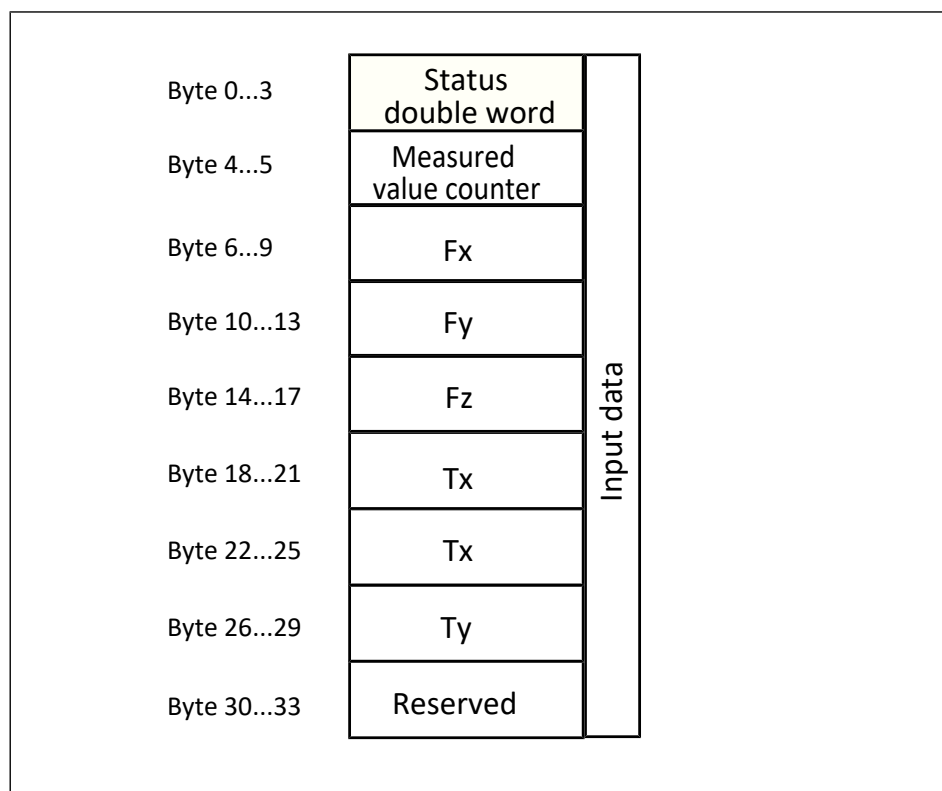
| Word | Byte | Bit | Cyclical output data |
|------|-------|-------|---|
| 0 | 0 | 0 | tare EN: tare DE: Tara |
| | | 1 | reset tare EN: reset tare DE: Tara Zurücksetzen |
| | | 2 - 7 | reserved |
| 1 | 0 - 6 | | reserved |
| | | 7 | reset software EN: reset software DE: Neustart |

- Tool Settings Number**
- The tool settings number is transmitted in bytes 2 and 3 of the cyclical output data in order to select the preconfigured memory bank of the tool settings.
 - The data format of the parameter is unsigned 16 bit. Valid values are 0 to 3.
- Noise Reduction Filter Number**
- The noise reduction filter is selected via a number in bytes 4 and 5 of the cyclical output data.
 - The data format of the parameter is unsigned 16 bits. Valid values are 0 to 4.

2.1.1.2 Cyclical input data

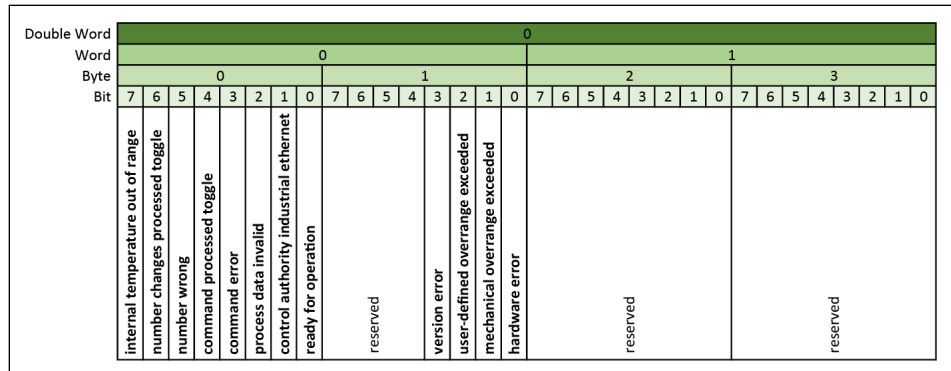
The cyclical input data is transferred from the module to the controller. As a result, the PLC receives feedback from the module on the commands and can react accordingly. The force/torque values are also transmitted.

Data frame The data frame of cyclical input data consists of the status double word, a measured value counter and the force/torque values.



Status double word

The status double word is transmitted in bytes 0 - 3 of the cyclical input data.



| Word | Byte | Bit | Cyclical input data |
|------|------|-----|--|
| 0 | 0 | 0 | ready for operation EN: ready for operation DE: Betriebsbereit |
| | | 1 | control authority fieldbus EN: control authority fieldbus DE: Steuerhoheit Feldbus |
| | | 2 | process data invalid * EN: cyclic process output data invalid DE: Zyklische Prozessausgangsdaten ungültig |
| | | 3 | command error EN: command error DE: Befehlsfehler |
| | | 4 | command processed toggle EN: command processed toggle DE: Befehl durchgeführt |
| | | 5 | number wrong EN: number wrong DE: Falsche Nummer |
| | | 6 | number changes processed toggle EN: number changes processed toggle DE: Änderung der Nummer erfolgreich |
| | | 7 | internal temperature out of range EN: internal temperature out of range DE: Interne Temperatur nicht im zulässigen Bereich |
| 0 | 1 | 8 | hardware error EN: hardware error DE: Hardware-Fehler |
| | | 9 | mechanical overrange exceeded EN: mechanical overrange exceeded DE: Mechanische Überlastgrenzen überschritten |

| Word | Byte | Bit | Cyclical input data |
|------|------|-----|--|
| | | 10 | user-defined overrange exceeded EN: user-defined overrange exceeded DE: Nutzerdefinierte Überlastgrenzen überschritten |
| | | 11 | firmware version error EN: firmware version error DE: Firmware-Versionsfehler |
| | | 12 | reserved |
| | | 13 | reserved |
| | | 14 | reserved |
| | | 15 | reserved |
| 1 | 2 | | reserved |
| 1 | 3 | | reserved |

*The "process data invalid" bit is set to "1" in the following cases:

- As long as the parameter <unlock_tool_settings> is set to enable changing the tool settings.
- If a hardware error has occurred in the sensor.
- If the mechanical overload limits have been exceeded.
- If the internal temperature is not within the permissible range.

Measured value counter

- A measured value counter is transmitted in bytes 4 and 5 of the cyclical input data.
- The data format of the parameter is unsigned 16 bits.
- The measurement value counter is typically increased by two with each new measurement value. Increases of one or three are system-related and do not indicate a malfunction or data loss. At 65535, the measured value counter jumps back to zero.

Force-torque values

- The six scaled force/torque values are transmitted in bytes 6 to 29 of the cyclical input data.
- The data format of each force or torque value is signed 32 bits.
- The <force_torque_scaling_factor> parameter can be used to scale the force/torque values and thus set the number of decimal places, for example. By default, the parameter is set to 1000, which corresponds to three decimal places.

Reserved

- No user data is currently transmitted in bytes 30 – 33 of the cyclical input data.

2.1.2 Acyclical data exchange

The "CANopen over EtherCAT" (CoE) protocol is used for acyclic data exchange.

The acyclic communication is implemented via SDOs (Service Data Object). For SDO communication, an index and a subindex must be specified.

3 Module functions

3.1 Booting and restarting

3.1.1 Booting and establishing operational readiness

| | |
|--|--|
| Short description | When booting, the internal hardware and the connected communication interfaces are checked after the electronics have booted up. It also checks whether a sensor is connected to the interface box. |
| Triggering | Booting can be triggered on the hardware side by applying the supply voltage, or triggered on the software side by a restart. |
| <hr/> | |
| NOTE | |
| To prevent any unexpected behavior of the module, all control bits equal to 0 should be cyclically transmitted to the module during booting. | |
| <hr/> | |
| Module feedback | <ul style="list-style-type: none"> • A change to the control byte is displayed by an immediate status change of the "command received toggle" status bit. This confirmation occurs regardless of whether the control command is subsequently completed successfully or if it can be processed at all. • If establishing operational readiness was successful, this is displayed by setting the "ready for operation" status bit. • If establishing operational readiness was not successful, the module remains in the error state. The "ready for operation" status bit is not set and the "process data invalid" status bit is set. |

3.1.2 Restart

| | |
|--------------------------|--|
| Short description | Booting is initiated when the module is restarted. |
| Trigger | The restart is triggered by a setting of the "reset software" control bit. |
| Module feedback | After the restart, the "command received toggle" status bit changes state and the "ready for operation" status bit is reset. |

3.2 Taring functions

3.2.1 Tare

| | |
|--------------------------|--|
| Short description | The tare function sets the current force/torque values to zero when the corresponding bit in the control word is set. An average value is formed over ten measured values and subtracted from future measurements. |
| Trigger | <ul style="list-style-type: none"> • The tare function is triggered by setting the "tare" control bit. • The bit must be reset after execution to enable reactivation. |
| Module feedback | <ul style="list-style-type: none"> • The end of the tare function is indicated by setting the "command processed toggle" status bit if no error has occurred. • Ending the tare function is indicated by setting the "command error" status bit if an error has occurred and the tare function could not be performed. |

3.2.2 Reset tare

| | |
|--------------------------|--|
| Short description | The reset tare function restores the original measurement when the corresponding bit in the control word is set. This means that the average value calculated in ▶ 3.2.1 [11] is no longer subtracted from future measured values. |
| Trigger | <ul style="list-style-type: none"> • The reset tare function is triggered by setting the "reset tare" control bit. • The bit must be reset after execution to enable reactivation. |
| Module feedback | <ul style="list-style-type: none"> • The end of the reset tare function is indicated by setting the "command processed toggle" status bit if no error has occurred. • The end of the reset tare function is indicated by setting the "command error" status bit if an error has occurred and the reset tare function could not be carried out. |

3.3 Switching tool settings

Short description

The tool settings switching function can be used to select between four pre-configured memory banks of tool settings. Each memory bank contains the following settings:

- Tool zero point – to perform a coordinate transformation from the sensor zero point (see operating manual) to the tool zero point. This is implemented by three translations and three rotations in the X, Y and Z directions. The translations are calculated first and then the rotations. The tool zero point can be reset to the sensor zero point if all translations and rotations are set to zero.

NOTICE

For example, if the tool zero point is set at the same point at which a force is applied, no torque acting on the sensor is displayed in the cyclical output data. This can lead to the sensor being overloaded.

The sensor zero point must therefore be used to monitor overload conditions. In addition, a status bit in the cyclical output data indicates when the sensor is mechanically overloaded.

- User-defined overload limits – to set user-defined limit values for limit value monitoring. If this is exceeded, the corresponding status bit is set in the cyclical output data.

Trigger

The tool settings function is triggered by changing the "tool settings number". The number must be transmitted cyclically to the module. Valid numbers are the values 0 – 3.

Command parameters Valid indices are the values 0 – 3.

Module feedback

- Exiting the tool settings function is indicated by setting the "number changes processed toggle" status bit if no error has occurred.
- Exiting the tool settings function is indicated by setting the "number wrong" status bit if an error has occurred and the function could not be executed. This occurs if an invalid value in <Tool Settings Number> was cyclically transferred to the module.

3.4 Switching the noise reduction filter

| | |
|---------------------------|---|
| Short description | The noise reduction filter function allows you to choose between five noise reduction filters. The noise reduction is implemented using a moving average. The command parameter is used to set the window size. |
| Trigger | The noise reduction filter function is triggered by changing the "noise reduction filter number". The number must be transmitted cyclically to the module. Valid numbers are the values 0 - 4, which stand for a window size of 1, 2, 4, 8 or 16 values in ascending order. |
| Command parameters | Valid numbers are the values 0 - 4, which stand for a window size of 1, 2, 4, 8 or 16 values in ascending order. |
| Module feedback | <ul style="list-style-type: none">• Exiting the noise reduction filter function is indicated by setting the "number changes processed toggle" status bit if no error has occurred.• Exiting the noise reduction filter function is indicated by setting the "number wrong" status bit if an error has occurred and the function could not be performed. This occurs in particular if an invalid value in <Noise Reduction Filter Number> was transmitted cyclically to the module. |

4 System parameters

4.1 Value ranges

Value ranges

The following internal data types are used:

| Data type | Threshold | Numerical values |
|-----------|------------|------------------|
| BOOL | MIN_BOOL | 0 |
| | MAX_BOOL | 1 |
| UINT8 | MIN_UINT8 | 0 |
| | MAX_UINT8 | 255 |
| UINT16 | MIN_UINT16 | 0 |
| | MAX_UINT16 | 65535 |
| UINT32 | MIN_UINT32 | 0 |
| | MAX_UINT32 | 4294968295 |
| INT32 | MIN_INT32 | -2147483648 |
| | MAX_INT32 | 2147483647 |
| FLOAT | MIN_FLOAT | -3.402823E+38 |
| | MAX_FLOAT | 3.402823E+38 |
| CHAR | MIN_CHAR | 0 |
| | MAX_CHAR | 255 |
| ENUM | MIN_ENUM | 0 |
| | MAX_ENUM | 255 |

4.2 Parameter list

In the following, all system-relevant parameters are listed according to the diagram "HEX-Code/DEC-Code <Parametername>"

NOTE

The parameter list refers to parameters that can be read out or written acyclically.

Some of the parameters listed here as "read only" can be changed in principle, but the user does not have the right to change these parameters.

All parameters that do not appear in this list are internal or reserved parameters.

Parameter configuration

All system parameters whereby the user has write permissions can be configured via acyclical data exchange.

4.2.1 Sensor

HEX 0x2001/0
DEC 8193/0

<product_name>

Short description: The pending product name can be read out with this parameter.

Parameter name: Product name

Access rights: Read

Data type: CHAR[30]

Format: ASCII-String

HEX 0x2001/1
DEC 8193/1

<product_text>

Short description: The product text can be read out with this parameter.

Parameter name: Product text

Access rights: Read

Data type: CHAR[30]

Format: ASCII-String

HEX 0x2001/2
DEC 8193/2

<device_id>

Short description: The device ID can be read out with this parameter.

Parameter name: Device ID

Access rights: Read

Data type: UINT32

HEX 0x2001/3
DEC 8193/3

<product_id>

Short description: The product ID of the sensor can be read out via this parameter.

Parameter name: Product ID

Access rights: Read

Data type: UINT32

HEX 0x2002
DEC 8194

<serial_number>

Short description: The serial number of the sensor can be read out with this parameter.

Parameter name: Serial number

Access rights: Read

Data type: CHAR[8]

Format: ASCII-String

HEX 0x2003/0
DEC 8195/0

<hardware_version>

Short description: The current hardware version of the sensor can be read out with this parameter.
 Parameter name: Hardware version
 Access rights: Read
 Data type: CHAR[8]
 Format: ASCII-String

HEX 0x2003/1
DEC 8195/1

<firmware_version>

Short description: The firmware version of the sensor can be read out with this parameter.
 Parameter name: Firmware version
 Access rights: Read
 Data type: CHAR[8]
 Format: ASCII-String

HEX 0x2035
DEC 8245

<internal_temperature>

Short description: This parameter can be used to read out the internal temperature of the sensor.
 Parameter name: Internal temperature
 Access rights: Read
 Data type: FLOAT
 Unit: degrees Celsius

HEX 0x2060
DEC 8288

<unlock_tool_settings>

Short description: This parameter can be used to lock or unlock the tool settings.
 Parameter name: Unlocking tool settings
 Access rights: Read and write
 Data type: BOOL
 Values: TRUE = Tool settings unlocked
 FALSE = Tool settings locked
 Default: FALSE
 sequence: To change the tool settings, follow these steps:

- 1.** Set parameter to TRUE.
 - ⇒ The "process data invalid" bit is set in the status double word.
- 2.** Make changes to memory banks 0 to 3 of the tool settings.
- 3.** Set parameter to FALSE.
 - ⇒ The changes are saved permanently and the values are applied to the calculations.
 - ⇒ The "process data invalid" bit is reset in the status double word.

HEX 0x2061
DEC 8289

<0_tool_center_point>

Short description: Memory bank 0: This parameter can be used to move the tool zero point by means of a coordinate transformation starting from the sensor zero point.

Parameter name: 0: Tool zero point

Access rights: Read and write

Data type: FLOAT

Subindices: 0: Translation x
1: Translation y
2: Translation z
3: Rotation x
4: Rotation y
5: Rotation z

Unit Translation [m], Rotation [rad]

HEX 0x2062
DEC 8290

<0_user_defined_overrange>

Short description: Memory bank 0: This parameter allows the user to set their own limit values for limit value monitoring. If this is exceeded, the corresponding status bit is set in the cyclical data. For each axis (Fx, Fy, Fz, Tx, Ty, Tz) there is a sub-index for the positive and negative limit value.

Parameter name: 0: User-defined overload limits

Access rights: Read and write

Data type: FLOAT

Subindices: 0: Upper overload limit Fx
1: Lower overload limit Fx
2: Upper overload limit Fy
3: Lower overload limit Fy
4: Upper overload limit Fz
5: Lower overload limit Fz
6: Upper overload limit Tx
7: Lower overload limit Tx
8: Upper overload limit Ty
9: Lower overload limit Ty
10: Upper overload limit Tz
11: Lower overload limit Tz

Unit F [N], T [Nm]

**HEX 0x2063
DEC 8291**

<1_tool_center_point>

Short description: Memory bank 1: This parameter can be used to move the tool zero point by means of a coordinate transformation starting from the sensor zero point.

Parameter name: 1: Tool zero point

Access rights: Read and write

Data type: FLOAT

Subindices: 0 – 5: see memory bank 0

Unit Translation [m], Rotation [rad]

**HEX 0x2064
DEC 8292**

<1_user_defined_overrange>

Short description: Memory bank 1: This parameter allows the user to set their own limit values for limit value monitoring. If this is exceeded, the corresponding status bit is set in the cyclical data. For each axis (Fx, Fy, Fz, Tx, Ty, Tz) there is a sub-index for the positive and negative limit value.

Parameter name: 1: User-defined overload limits

Access rights: Read and write

Data type: FLOAT

Subindices: 0 – 11: see memory bank 0

Unit Fx+, Fx-, Fy+, Fy-, Fz+, Fz- [N], Tx+, Tx-, Ty+, Ty-, Tz+, Tz- [Nm]

**HEX 0x2065
DEC 8293**

<2_tool_center_point>

Short description: Memory bank 2: With this parameter, the tool zero point can be shifted by means of a coordinate transformation starting from the sensor zero point.

Parameter name: 2: Tool zero point

Access rights: Read and write

Data type: FLOAT

Subindices: 0 – 5: see memory bank 0

Unit Translation [m], Rotation [rad]

HEX 0x2066
DEC 8294

<2_user_defined_overrange>

Short description: Memory bank 2: This parameter allows the user to set their own limit values for limit value monitoring. If this is exceeded, the corresponding status bit is set in the cyclical data. For each axis (Fx, Fy, Fz, Tx, Ty, Tz) there is a sub-index for the positive and the negative limit value

Parameter name: 2: User-defined overload limits

Access rights: Read and write

Data type: FLOAT

Subindices: 0 - 11: see memory bank 0

Unit Fx+, Fx-, Fy+, Fy-, Fz+, Fz- [N], Tx+, Tx-, Ty+, Ty-, Tz+, Tz- [Nm]

HEX 0x2067
DEC 8295

<3_tool_center_point>

Short description: Memory bank 3: With this parameter, the tool zero point can be shifted by means of a coordinate transformation starting from the sensor zero point.

Parameter name: 3: Tool zero point

Access rights: Read and write

Data type: FLOAT

Subindices: 0 - 5: see memory bank 0

Unit Translation [m], Rotation [rad]

HEX 0x2068
DEC 8296

<3_user_defined_overrange>

Short description: Memory bank 3: This parameter allows the user to set their own limit values for limit value monitoring. If this is exceeded, the corresponding status bit is set in the cyclical data. For each axis (Fx, Fy, Fz, Tx, Ty, Tz) there is a sub-index for the positive and negative limit value.

Parameter name: 3: User-defined overload limits

Access rights: Read and write

Data type: FLOAT

Subindices: 0 - 11: see memory bank 0

Unit Fx+, Fx-, Fy+, Fy-, Fz+, Fz- [N], Tx+, Tx-, Ty+, Ty-, Tz+, Tz- [Nm]

4.2.2 Interface box

HEX 0x3000/0
DEC 12288/0

<vendor_name>

Short description: The manufacturer can be read out with this parameter.

Parameter name: Manufacturer name

Access rights: Read

Data type: CHAR[30]

Format: ASCII-String

HEX 0x3000/1
DEC 12288/1

<vendor_text>

Short description: The manufacturer text can be read out with this parameter.

Parameter name: Manufacturer text

Access rights: Read

Data type: CHAR[30]

Format: ASCII-String

HEX 0x3001/0
DEC 12289/0

<product_id>

Short description: The product ID of the interface box can be read out with this parameter.

Parameter name: Product ID

Access rights: Read

Data type: UINT32

HEX 0x1001/1
DEC 12289/1

<serial_number>

Short description: The serial number of the interface box can be read out with this parameter.

Parameter name: Serial number

Access rights: Read

Data type: CHAR[8]

Format: ASCII-String

HEX 0x3002/0
DEC 12290/0

<hardware_version>

Short description: The current hardware version of the interface box can be read out with this parameter.

Parameter name: Hardware version

Access rights: Read

Data type: CHAR[8]

Format: ASCII-String

HEX 0x3002/1
DEC 12290/1

<firmware_version>

Short description: The firmware version of the interface box can be read out with this parameter.

Parameter name: Firmware version

Access rights: Read

Data type: CHAR[8]

Format: ASCII-String

HEX 0x3003/0
DEC 12291/1

<function_tag>

Short description: The function tag can be read out with this parameter.

Parameter name: Function tag

Access rights: Read

Data type: CHAR[30]

Format: ASCII-String

HEX 0x3003/1
DEC 12291/1

<location_tag>

Short description: The location tag can be read out with this parameter.

Parameter name: Location tag

Access rights: Read

Data type: CHAR[30]

Format: ASCII-String

HEX 0x3021
DEC 12321

<force_torque_scaling_factor>

Short description: This parameter can be used to read and write the scaling factor for the force-torque values of the industrial bus.

Parameter name: Scaling factor for force-torque values of the industrial bus

Access rights: Read and write

Data type: UINT32

Default: 1000

Min. 1

Max. 1.000.000

HEX 0x3032
DEC 12338

<customer_interface_type>

Short description: The type of customer interface can be read out with this parameter.

Parameter name: Customer interface type

Access rights: Read

Data type: ENUM

Enumeration: 0 = Unknown
1 = EtherCat
2 = Profinet
3 = Ethernet/IP
4 = Plain Ethernet

5 Start-up

5.1 Safety

Commissioning of the module may only be carried out by qualified personnel with programming and interface knowledge!



⚠ CAUTION

Risk of injury due to electromagnetic interference!

Electromagnetic interference can cause malfunctions and lead to unexpected movements.

- Use electrical components, e.g. sensors, controllers, etc. according to EN 61000-5-7.

5.2 System integration

The communication protocol "SCHUNK Flexible Protocol" is available for operation within the plant.

For further information on communication, module functions and parameters, see the corresponding sections in this manual.

NOTE

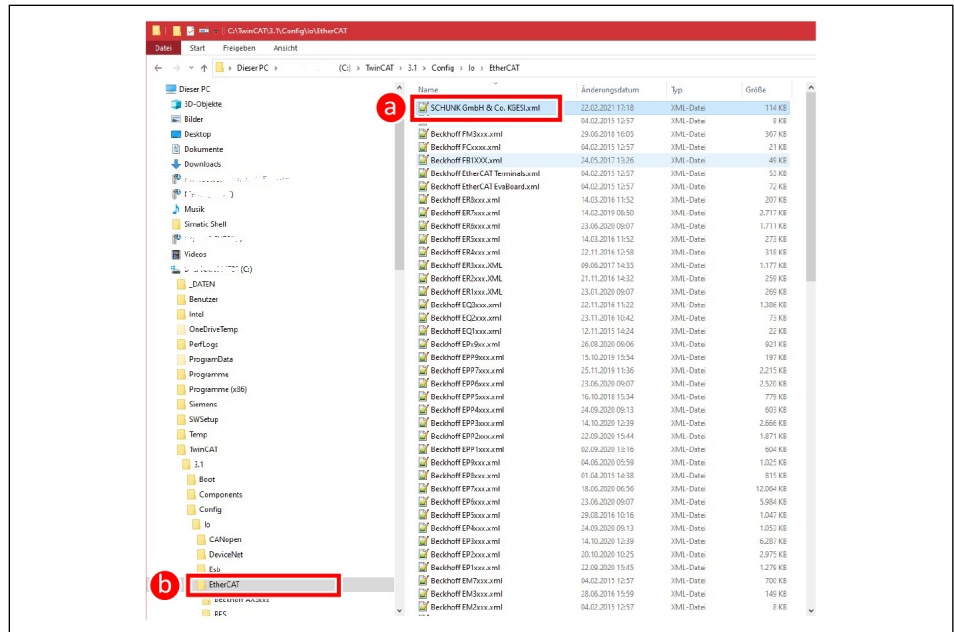
If the module is **not** operated at a Beckhoff control, the byte sequence must be checked and if necessary adapted at the control.

Overview

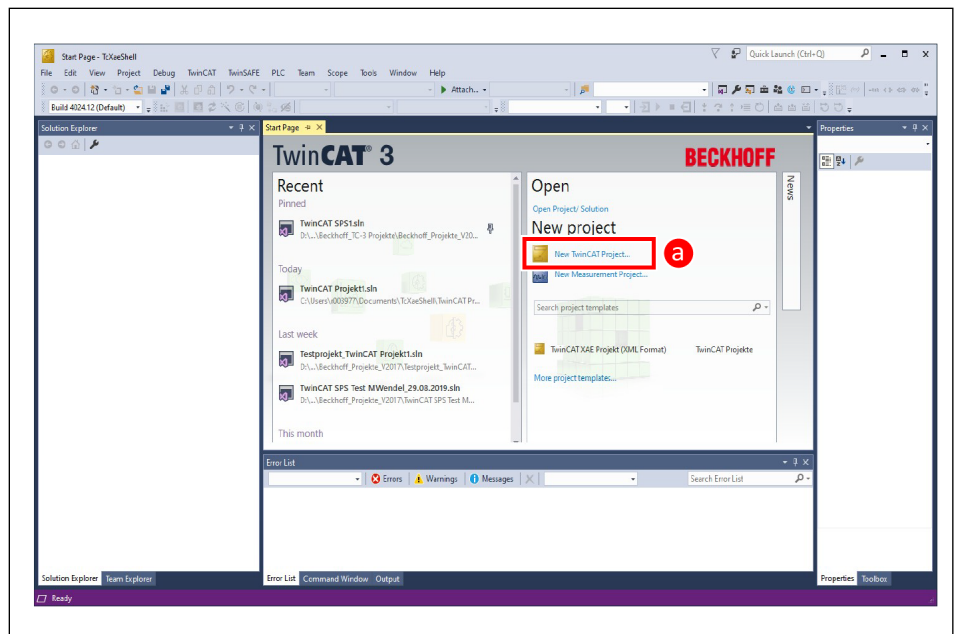
- The module is mounted and electrically connected. For more information, see Assembly and Operating Manual, ▶ 1.1 [4].
- 1. Activate logic and power supply.
 - ⇒ LED LOG and PWR light up green.
- 2. Connect the cables for communication.
 - ⇒ Communication is reported back by LED status indicator.
- 3. Configure controller and module, ▶ 5.3 [24].

5.3 Commissioning with Beckhoff "TwinCAT 3 ®" software for EtherCAT

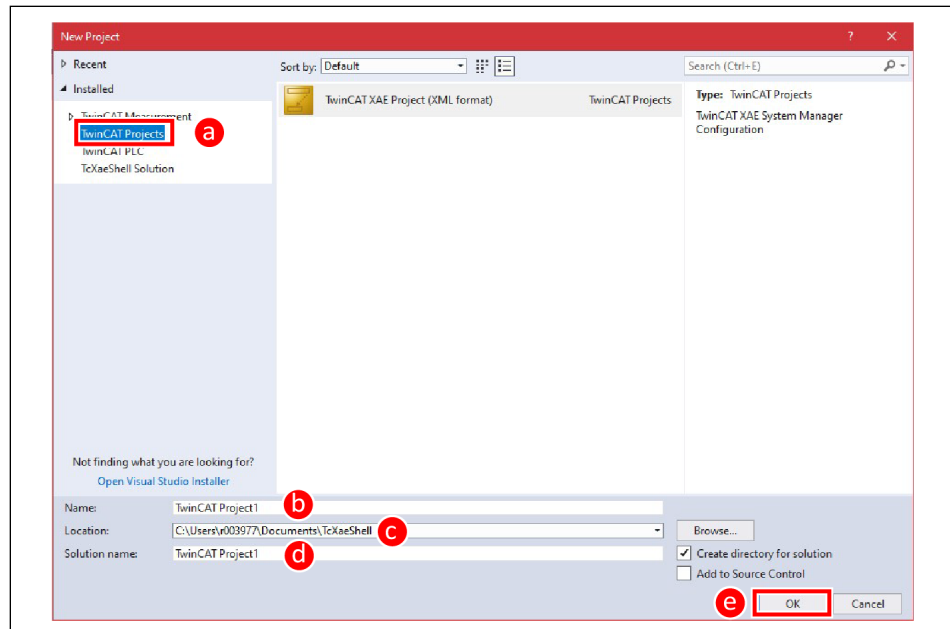
1. Copy the EtherCAT XML file (a) into the intended directory (b).



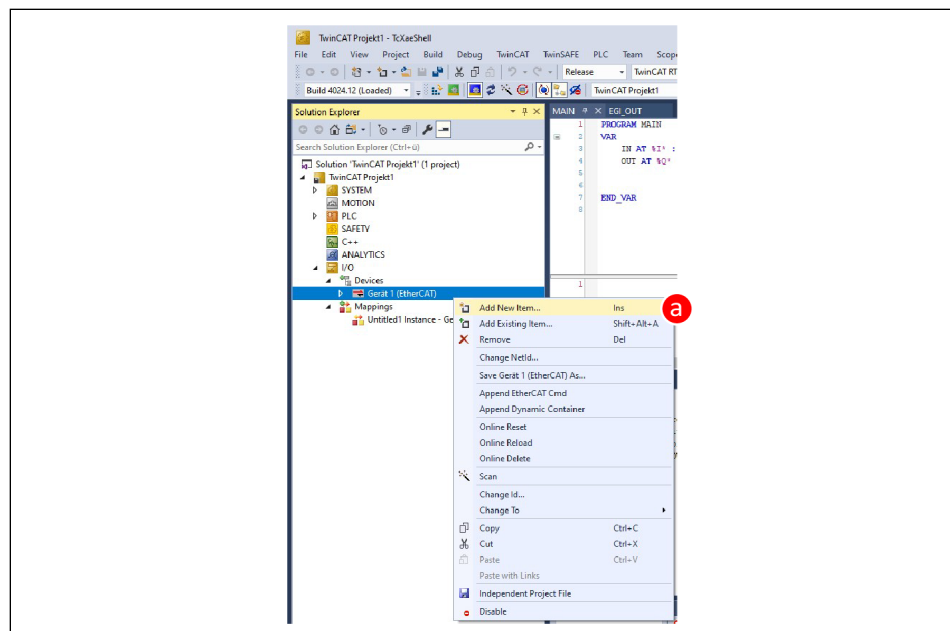
2. Start TwinCAT® 3 and click "New TwinCAT Project" (a) to open the window for creating a new project.



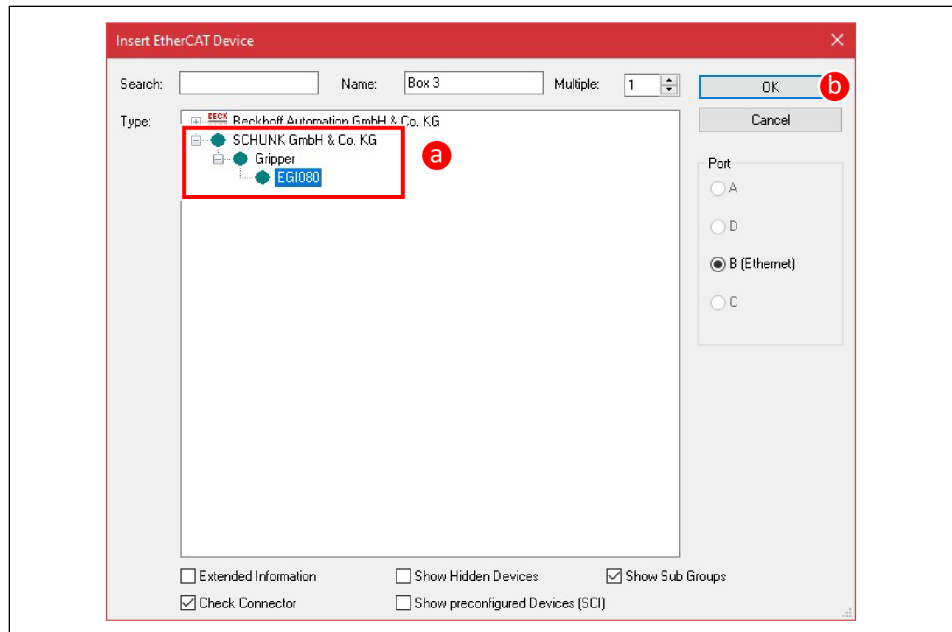
3. Select "TwinCAT Projects" (a).
4. Change name (b), storage location (c) and project folder name (d) as required.
5. Click the "OK" button (e) to create the project.



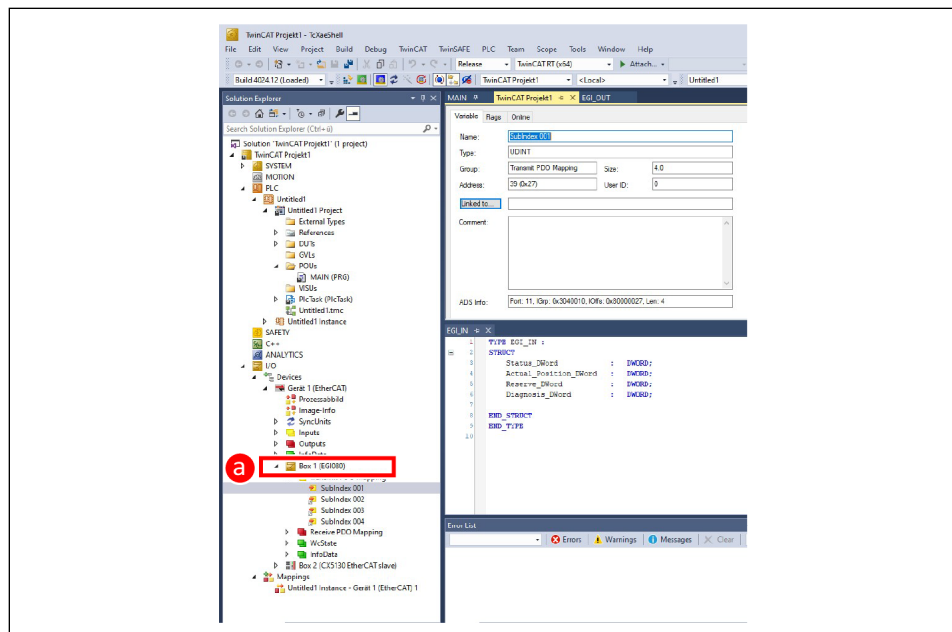
6. Right-click in the Solution Explorer (left side) in the subitem *I/O > Devices > Device 1 (EtherCAT)* to add a new device (a).



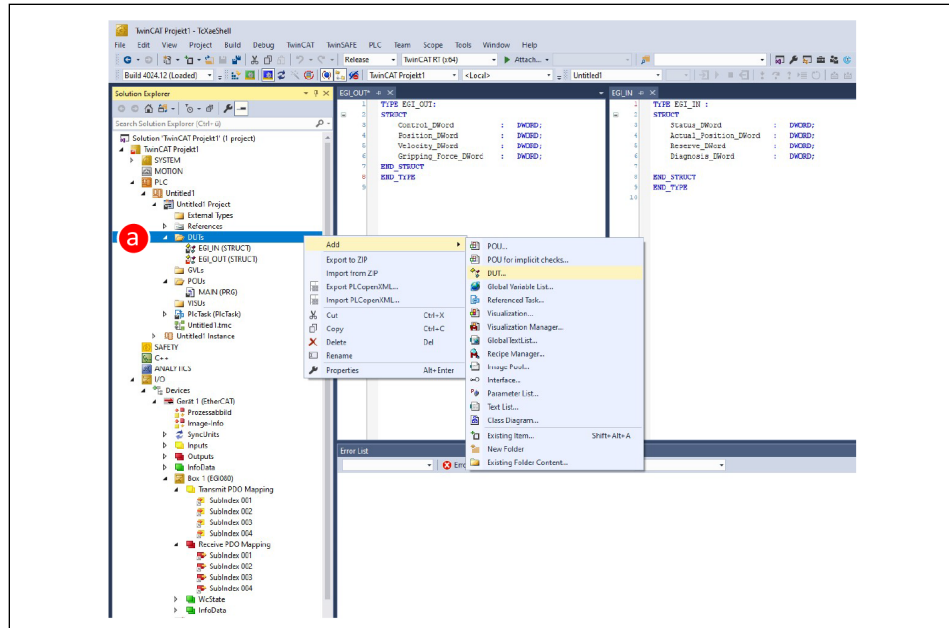
7. Select the appropriate device from the catalog under *Schunk GmbH & Co. KG* (a).
8. Click the "OK" button (b) to confirm the selection.



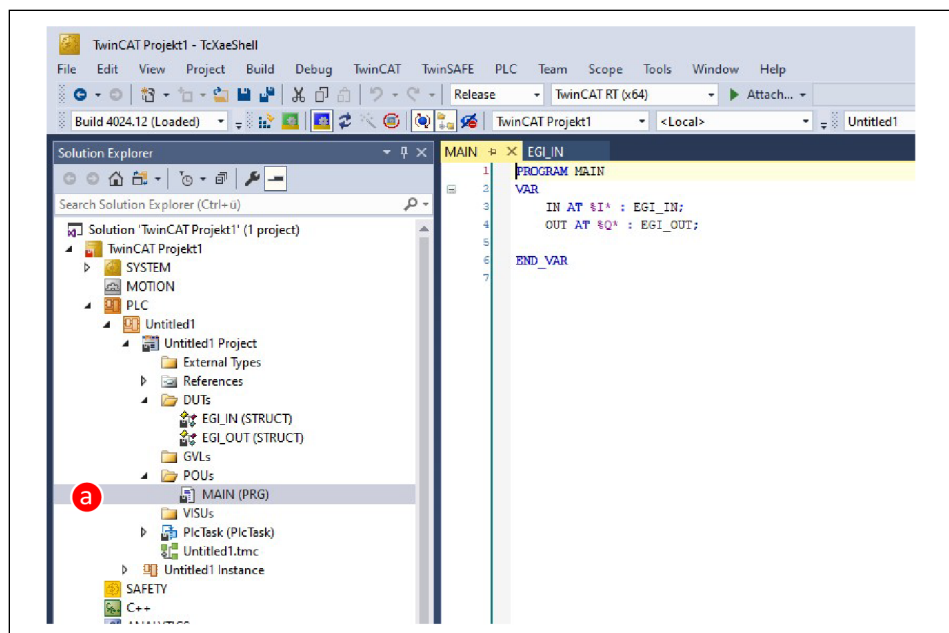
⇒ The added device is displayed in the Solution Explorer (left side) under *Device 1 (EtherCAT)* as "Box 1" (a).



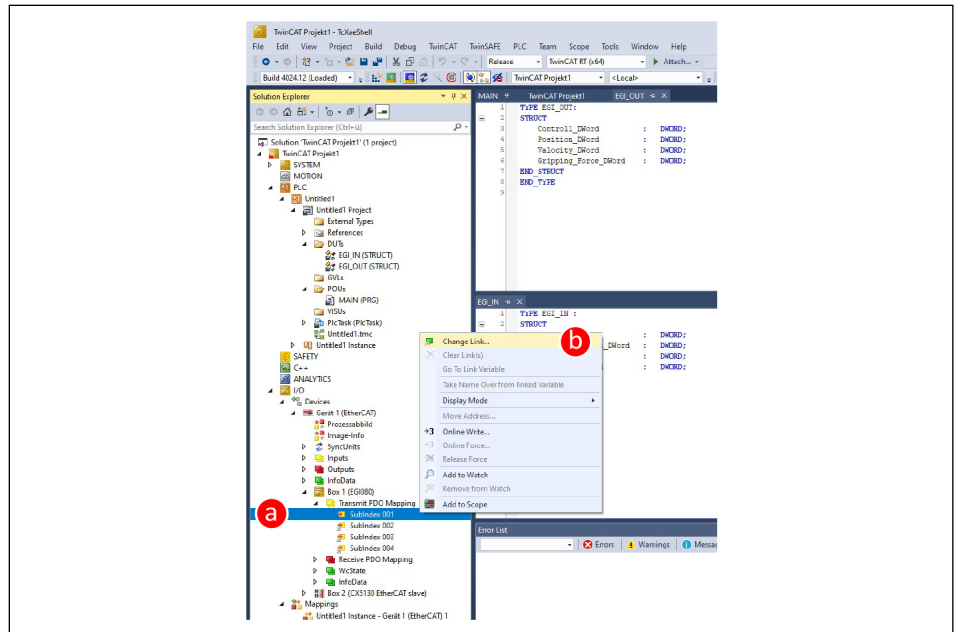
9. In the Solution Explorer, right-click the subitem *PLC > . . . > DUTs* (a) to open a menu.
10. Click "DUT" under "Add".
 - ⇒ A new window appears.
11. Create the necessary variable structures to connect the hardware with the software structures. When creating the variables, make sure that the same variable lengths are used. In the case of the SCHUNK protocol, these are "double words".



12. Link DUTs as variables in the main program.



13. Double-click subindex (a) or select "Change link" (b) after right-clicking the subindex.

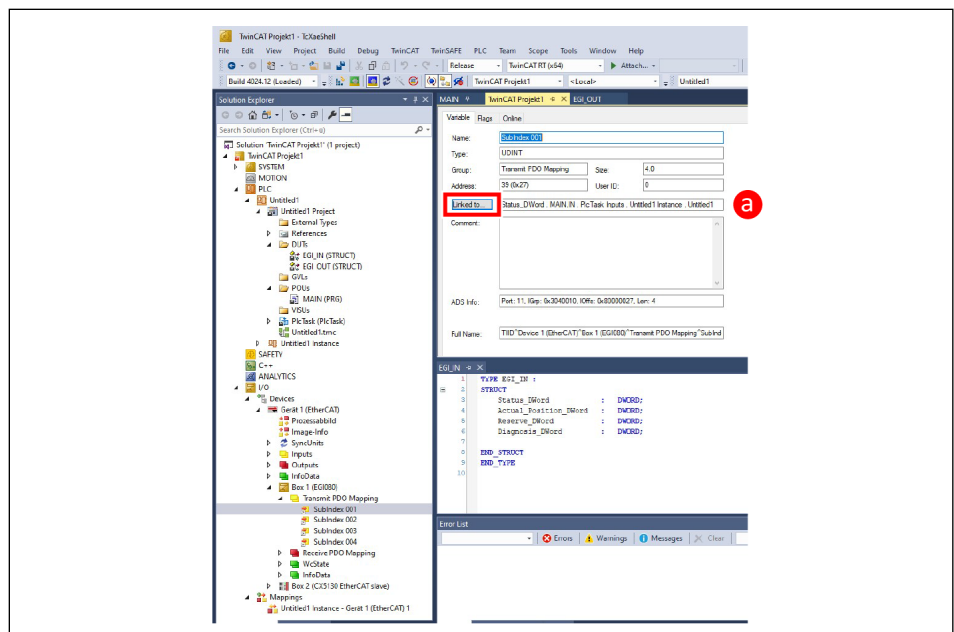


⇒ A window for the corresponding subindex opens.

14. Click "Linked m." (a).

⇒ A selection window appears.

15. Assign appropriate variables to the hardware.
16. Click the "OK" button.

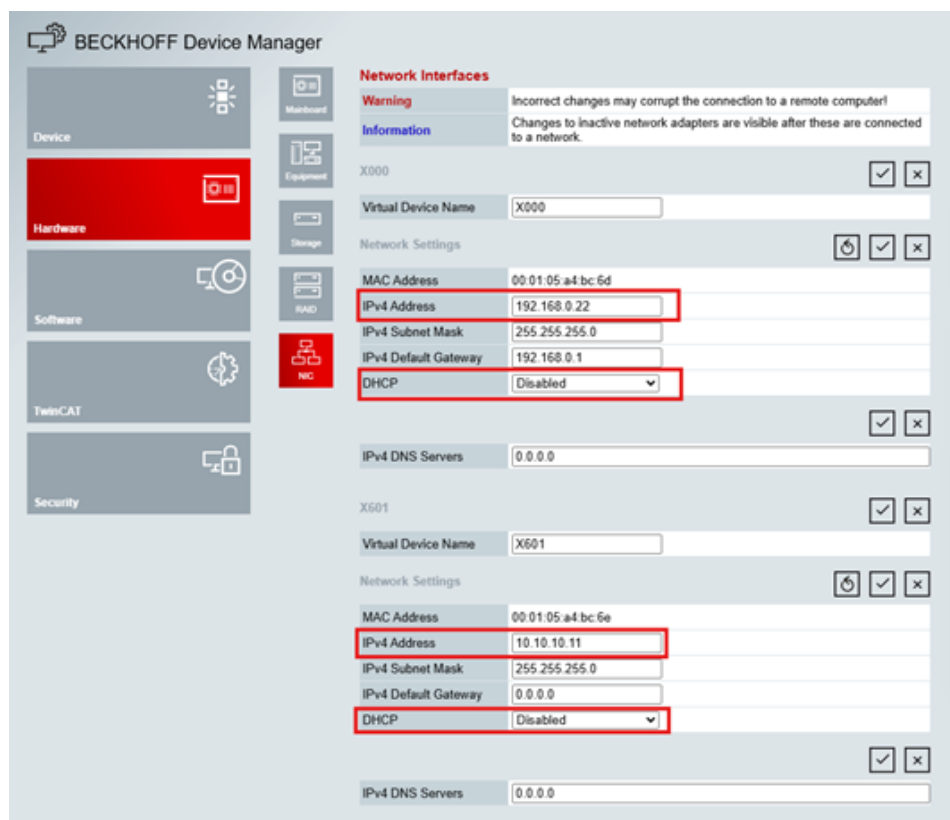


17. Transfer the program to the controller and start programming the device.

5.4 Configuring Ethernet over EtherCAT with TwinCAT

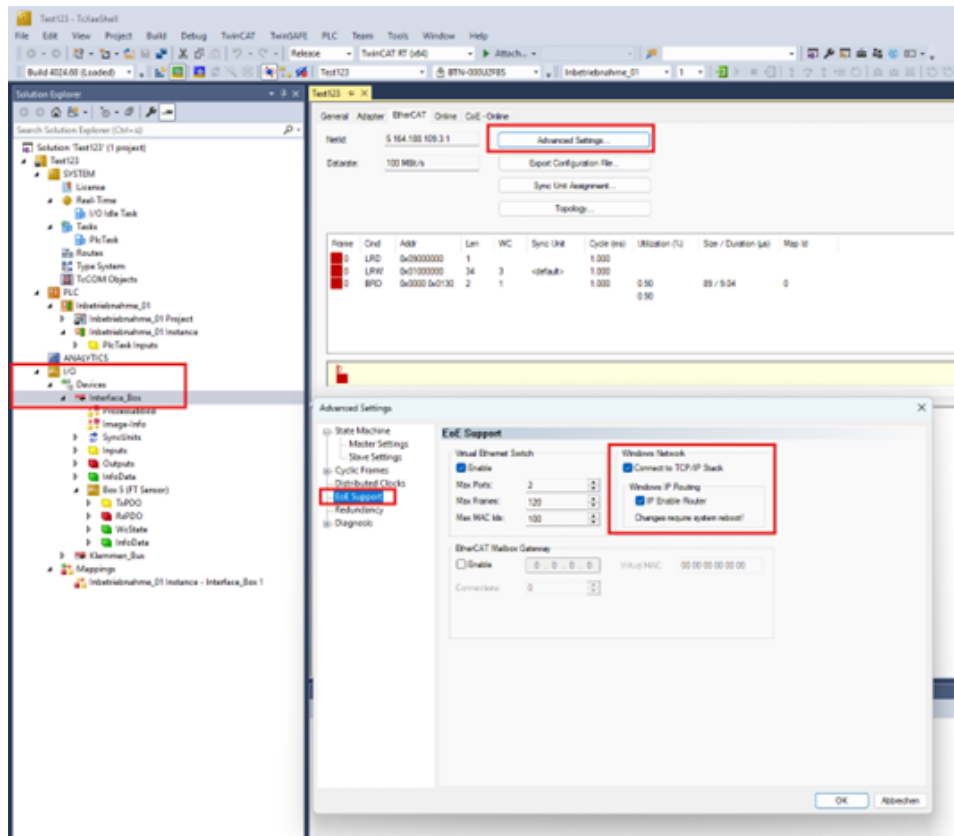
- The SCHUNK Control Center with EtherCAT must be used for configuration.
- The module must be connected to the EtherCAT master and commissioning must be complete ▶ 5.3 [24].
- A PC with TwinCAT and a PC with SCHUNK Control Center installed must be connected to each other. This can be a single PC with both programs or two separate PCs. TwinCAT is only required for the initial configuration.

Set up a fixed IP address on the Ethernet port and EtherCAT port of the EtherCAT master

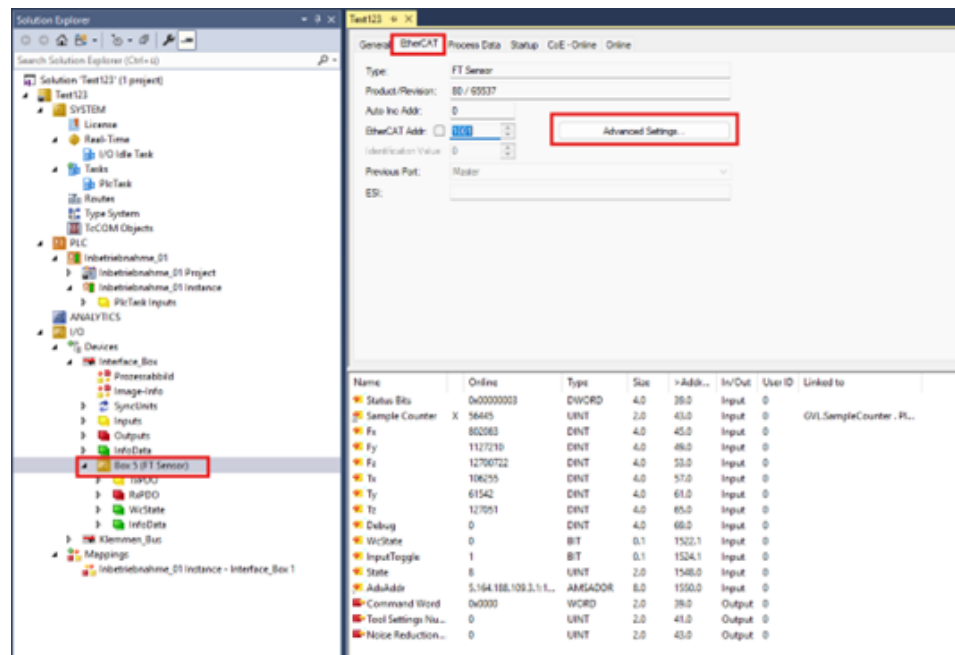


1. Start Beckhoff Device Manager.
2. Select the "Hardware" > "NIC" buttons.
3. Ethernet port
 - ⇒ IPv4 address: 192.168.0.22 (example)
 - ⇒ DHCP: Disabled
4. EtherCAT port of the EtherCAT master
 - ⇒ IPv4 address: 10.10.10.11 (example)
 - ⇒ DHCP: Disabled

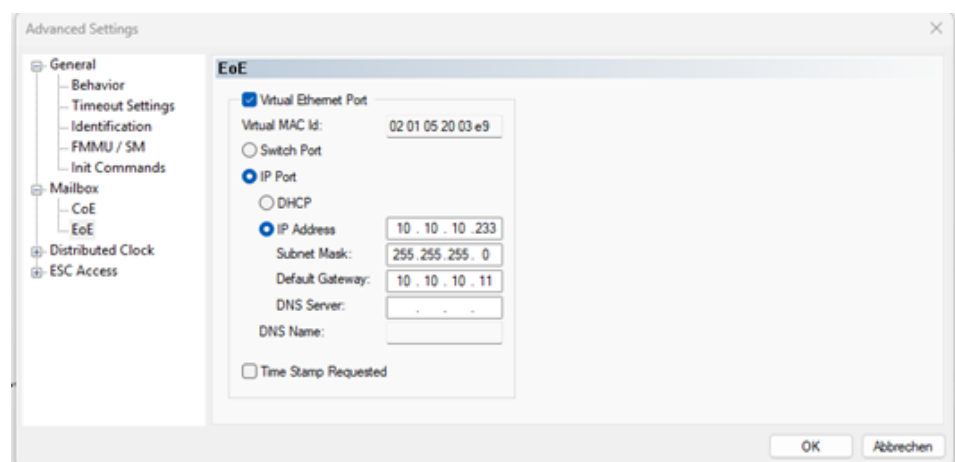
Enable IP routing on EtherCAT master



1. Start TwinCat.
2. Select EtherCAT I/O Master.
3. Select the "Advanced Settings" button.
4. Select the "EoE Support" section.
5. Check the boxes for "Connect to TCP/IP Stack" and "IP Enable Router."
6. Select the "OK" button.
 - ⇒ IP routing on the EtherCAT master is now enabled.
7. Alternatively, routing can be enabled on the IPC itself via the registry edit (HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters).
8. If problems arise subsequently, it may be necessary to deactivate the firewall (clarify any resulting risks beforehand). SCHUNK recommends continuing with the firewall activated for the time being.
9. Restart the EtherCAT master.



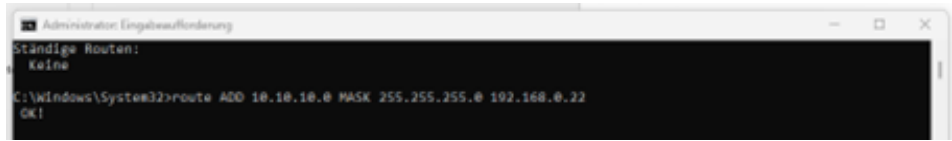
10. Select EtherCAT SubDevice (SCHUNK FTS).
11. Select the "EtherCAT" tab.
12. Select the "Advanced Settings" button.



13. Select "Mailbox" > "EoE."
14. Check the box next to "Virtual Ethernet Port."
15. Activate the "IP Post" radio button.
16. Activate the "IP Address" radio button.
17. Enter the gateway in "IP Address".
(Default: IP address of the EtherCAT master port, e.g., 10.10.10.11).
18. Enter the virtual IP address of the subdevice (SCHUNK FTS) in "Default Gateway" (e.g., 10.10.10.233).
19. Select the "OK" button.
20. Activate TwinCAT configuration and set to run mode.
21. Perform a power cycle on the subdevice (SCHUNK FTS).

Add routing to the commissioning PC (PC with SCHUNK Control Center).

1. Start the command prompt as administrator.

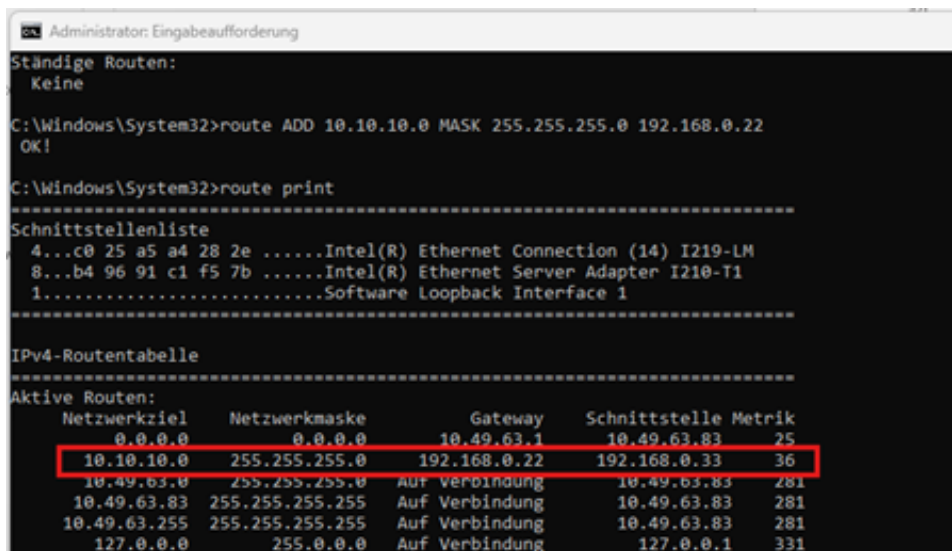


2. Enter "route ADD 10.10.10.0 MASK 255.255.255.0 192.168.0.22". (Route is not retained after restart)

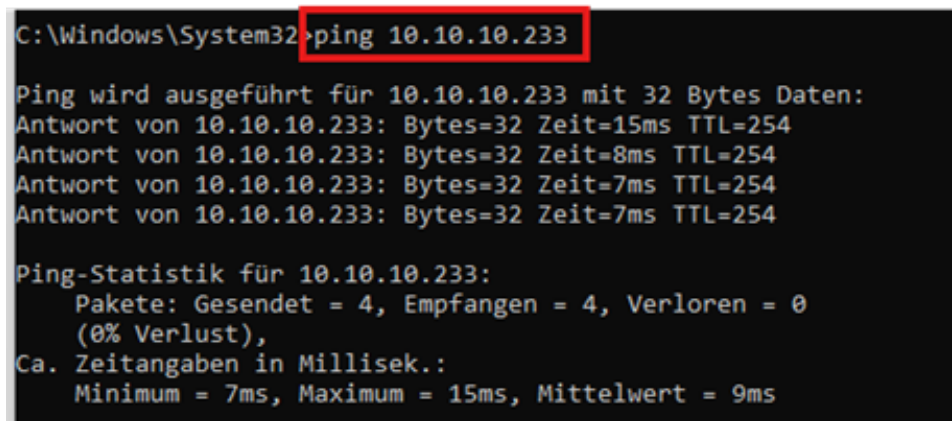
or

3. Enter "route -p ADD 10.10.10.0 MASK 255.255.255.0 192.168.0.22". (The route is set up permanently with the "-p" parameter)

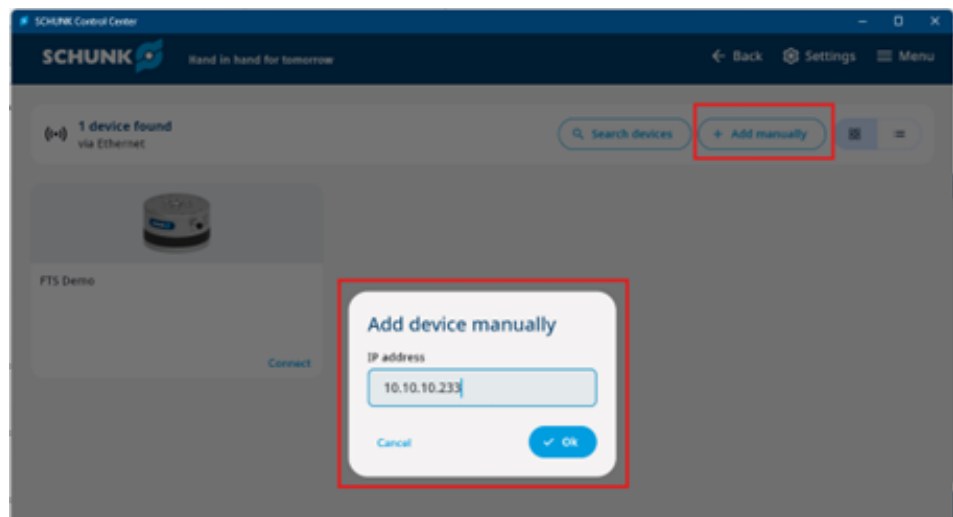
⇒ The subnet 10.10.10.0/24 can be reached via the router in the device with the IP address 192.168.0.22.



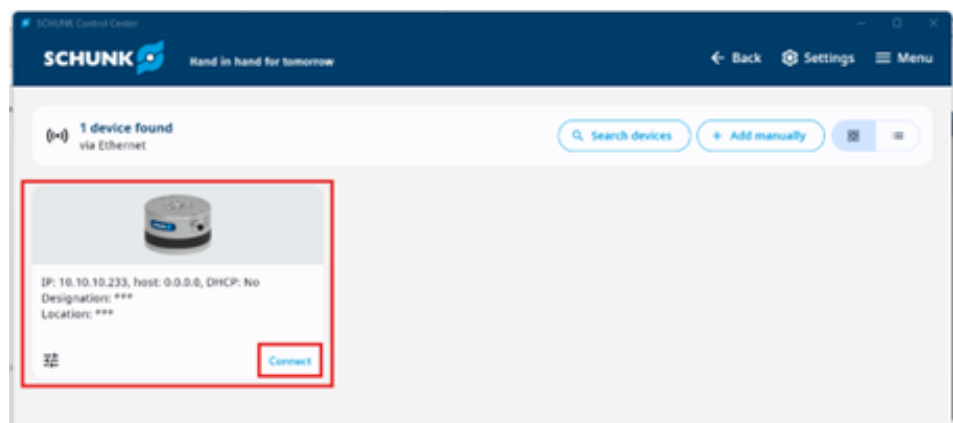
4. Enter "route print" to check the routes.



5. The subdevice (SCHUNK FTS) should now be pingable.
6. Start SCHUNK Control Center.
 - ⇒ The module should now be displayed.
 - ⇒ If the module is not displayed, it must be added manually using the IP address.



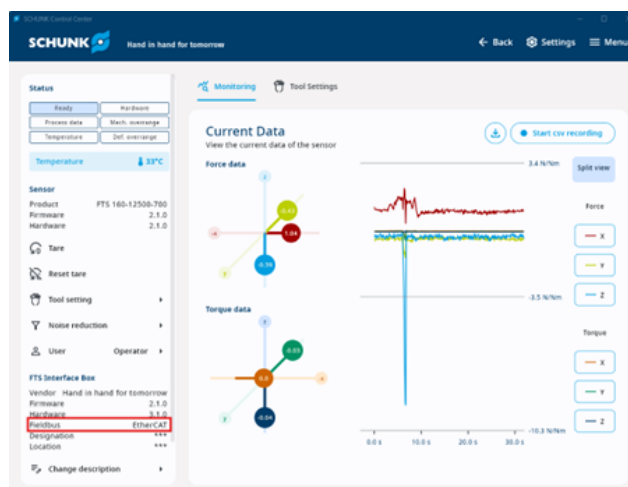
7. Select the "Add manually" button.
8. Enter the IP address.
9. Select the "OK" button.



⇒ The module will now be displayed.

10. Select the "Connect" button.

⇒ Details about the module are displayed.



NOTE

The EtherCAT subdevice must be in operator state for EoE to work.

5.5 SCHUNK Control Center – App FTS

The *FTS (Force Torque Sensor)* application can be started via the SCHUNK Control Center. This app enables fast commissioning and parameterization of the module.

The software can be downloaded from [schunk.com/downloads-software](https://www.schunk.com/downloads-software).

Range of functions of the *FTS* app

- Configuration and commissioning:
 - Display status information
 - Record force and torque values
 - Changing the IP address
 - Display diagnostic and error messages
 - Execute firmware updates
- Automatic and manual search for modules in the network
- Visual display of the connected module
- Configuration of connected tools
- Setting filters for the measurement signal

Start software

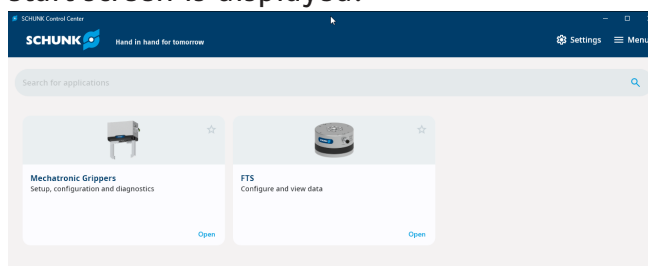
NOTE

In order for the *FTS* app to communicate with the module, the "Ethernet over EtherCAT" (EoE) function must be activated in the controller. It must also be ensured that communication is not blocked by a firewall or other network technology ▶ 5.4 [29].

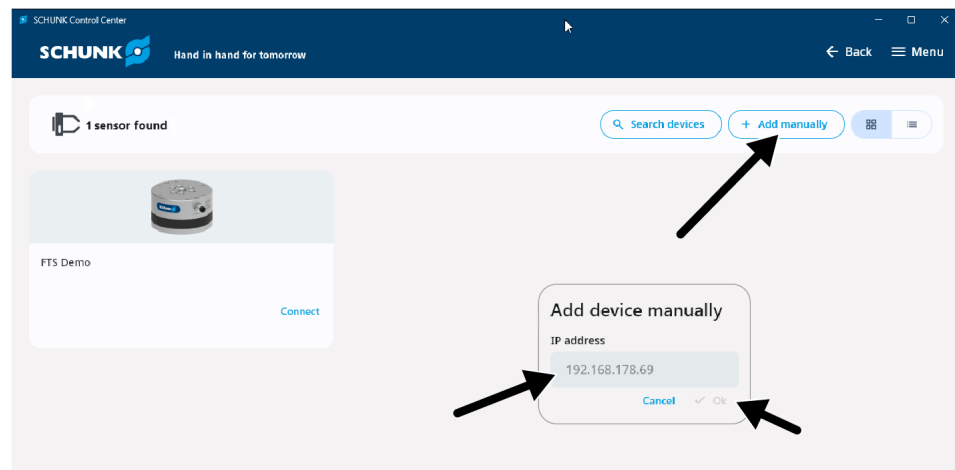
NOTE

In order for the *FTS* app to communicate with the module over an Ethernet network, it must be ensured that the communication is not prevented by a firewall or other network technology.

- Module is electrically connected to the power supply unit.
 - SCHUNK Control Center is installed.
1. Connect the computer to the network in which the module is integrated. For EtherCAT sensors, this network corresponds to the EoE network.
 2. Open SCHUNK Control Center.
 - ⇒ Start screen is displayed.



3. Select the *FTS* app.

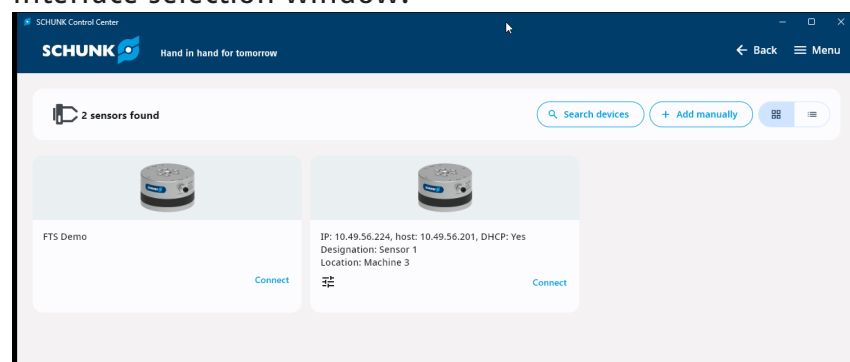


4. Select the "+ Add manually" button.

⇒ The "Add device manually" window opens,

5. Enter the IP address of the sensor to be connected and select the "OK" button.

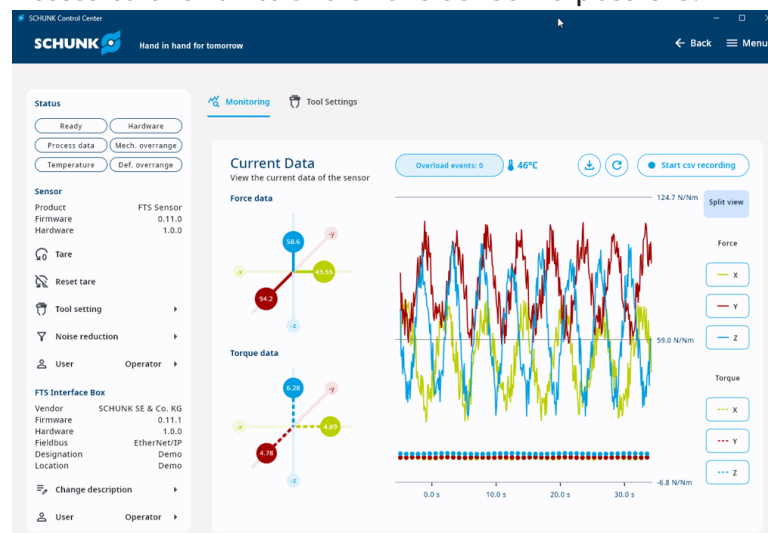
⇒ Sensors found are displayed in the communication interface selection window.



6. Select the desired sensor.

⇒ The app connects to the sensor.

⇒ Access to the functions of the sensor is possible.



6 Appendix

6.1 Control word

The controls of the control word are described in detail below. For a clear illustration of the control word, see chapter ▶ 2.1.1.1 [6].

Never set more than one control bit. If more than one control bit is set at any one time, the "command error" status bit is set.

Bit 0 – tare

| Edge change | Module reaction |
|-------------|---|
| 0 → 1 | Start tare process to calculate tare values |
| 1 → 0 | no reaction |

Bit 1 – reset tare

| Edge change | Module reaction |
|-------------|-------------------|
| 0 → 1 | Reset tare values |
| 1 → 0 | no reaction |

Bit 2 to 14– reserved

| Edge change | Module reaction |
|-------------|-----------------|
| 0 → 1 | no reaction |
| 1 → 0 | no reaction |

Bit 15 – reset software

| Edge change | Module reaction |
|-------------|--|
| 0 → 1 | The module is restarted on the software side |
| 1 → 0 | no reaction |

6.2 Status double word

The status bits of the status double word are described in detail below. For a clear representation of the status double word, see chapter ▶ 2.1.1.2 [7].

Bit 0 – ready for operation

| Status | Module feedback |
|--------|--|
| 0 | The module is not ready for operation. |
| 1 | The module is ready for operation. |

Bit 1 – control authority fieldbus

| Status | Module feedback |
|--------|---|
| 0 | The fieldbus does not have a control logic. |
| 1 | The fieldbus has a control logic. |

Bit 2 – process data invalid

| Status | Module feedback |
|--------|--|
| 0 | The cyclical process output data is valid. |
| 1 | The cyclical process output data is invalid. |

Bit 3 – command error

| Status | Module feedback |
|--------|---|
| 0 | No information is reported. |
| 1 | The command sent to the module is not feasible. |

Bit 4 – command processed toggle

| Status | Module feedback |
|--------|---|
| 0 → 1 | The command sent to the module was successfully executed. |
| 1 → 0 | The command sent to the module was successfully executed. |

Bit 5 – Number wrong

| Status | Module feedback |
|--------|---|
| 0 | No information is reported. |
| 1 | The number sent to the module cannot be executed. |

Bit 6 – number changes processed toggle

| Status | Module feedback |
|--------|--|
| 0 → 1 | The number sent to the module has been successfully changed. |
| 1 → 0 | The number sent to the module has been successfully changed. |

Bit 7 – internal temperature out of range

| Status | Module feedback |
|--------|---|
| 0 | The internal temperature of the sensor is within the permissible range. |
| 1 | The internal temperature of the sensor is outside the permitted range of 0 to XX degrees. |

Bit 8 – hardware error

| Status | Module feedback |
|--------|--|
| 0 | The hardware is working properly. |
| 1 | There is a hardware error, e.g. in the internal communication or the memory. |

Bit 9 – mechanical overrange limits exceeded

| Status | Module feedback |
|--------|--|
| 0 | No limit value exceeded. |
| 1 | The mechanical overload limits of the sensor have been exceeded. The sensor is possibly damaged. |

Bit 10 – user-defined overrange limits exceeded

| Status | Module feedback |
|--------|---|
| 0 | No limit value exceeded. |
| 1 | One or more of the user-defined overload limit values has been exceeded. See parameter 0x0061 ff. of the tool settings. |

Bit 11 – firmware version error

| Status | Module feedback |
|--------|---|
| 0 | Sensor and interface box have the same firmware version. |
| 1 | The sensor and interface box have different firmware versions and require an update or downgrade. |

Bit 12 to 31 – reserved

| Status | Module feedback |
|--------|---|
| 0 | No information is provided in feedback. |
| 1 | No information is provided in feedback. |

6.3 Brands

- EtherCAT is a registered trademark and patented technology licensed by Beckhoff Automation GmbH, Germany.
- TwinCat is a registered trademark of Beckhoff Automation GmbH.





SCHUNK SE & Co. KG
Spanntechnik | Greiftechnik | Automatisierungstechnik

Bahnhofstr. 106 - 134
D-74348 Lauffen/Neckar
Tel. +49-7133-103-0
info@de.schunk.com
schunk.com

Folgen Sie uns | *Follow us*



Wir drucken nachhaltig | *We print sustainable*