Commissioning instructions FTS with Ethernet Interface, Firmware 2.1.0

Force-torque sensor

Translation of original commissioning instructions

Hand in hand for tomorrow

Imprint

Technical changes:
We reserve the right to make technical improvements.

Document number: 1634813-EN 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

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:

- Ethernet (TCP/IP, UDP)

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 [🗋 20] and ▸ 4.2.2 [🗋 25].

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

The integrated interface can be used to exchange data between the module and the controller.

Communication types

The module supports the following communication types:

- TCP/IP
- UDP

Communication structure

Communication with the force/torque sensor can be established via a TCP socket. The TCP socket consists of the IP address of the sensor and the fixed port 82. Once the connection has been established, command packets can be sent to the force/torque sensor and response and process data packets can be received from the force/torque sensor.

Communication framework

The following table describes the basic communication framework for command, response and process data packets. All messages start with two bytes for synchronization, which signal the start of the new message and are permanently set to 0xFF. A packet counter follows with two bytes, which starts at zero and is incremented by one with each packet. At 65,535 there is an over run back to zero. The packet counter can be used to detect packet losses. The length value consisting of two bytes determines the number of bytes of user data subsequently transferred. A separate CRC check is not required as the transmission is already secured with a CRC by the Ethernet protocol.

NOTE

- Data is transmitted in "Little Endian" format, i.e. the least significant byte of a data word is transmitted first. This means that, for example, for an integer value 0x1234, which is represented by two consecutive bytes, the lower-value byte 0x34 is transmitted first and then the higher-value byte 0x12.
- Only the user data is described in the following sections. This means that the numbering of the bytes always refers to the start of the user data and starts at zero.

Byte	Content
0	Synchronization: Signals the start of the message with the fixed byte sequence 0xFFFF.
1	
2	Packet counter: For detecting packet loss. Takes values from 0 – 65,535 and then jumps back to zero.
3	
4	Length of user data N: Length of the user data contained in the package.
5	
6	User data
...	
N+6	

Response packets

Each command packet received by the force/torque sensor is answered with a packet that acknowledges the command received and returns any return values.

2.2 Process data

The process data is transferred by the Module to the control. The process data of the force/torque sensor is described in detail below.

TCP

The process data is sent via TCP at a fixed frequency of 20 Hz.

UDP

With pure Ethernet modules, the process data can also be sent with an adjustable frequency via UDP. See ▶ 4.2.2 [🗋 26], parameter *Output data rate UDP* to set the output data rate. The UDP process data is activated via a command. The UDP process data is sent from the source port 52964 to the destination port 54843 and the last IP address connected via TCP.

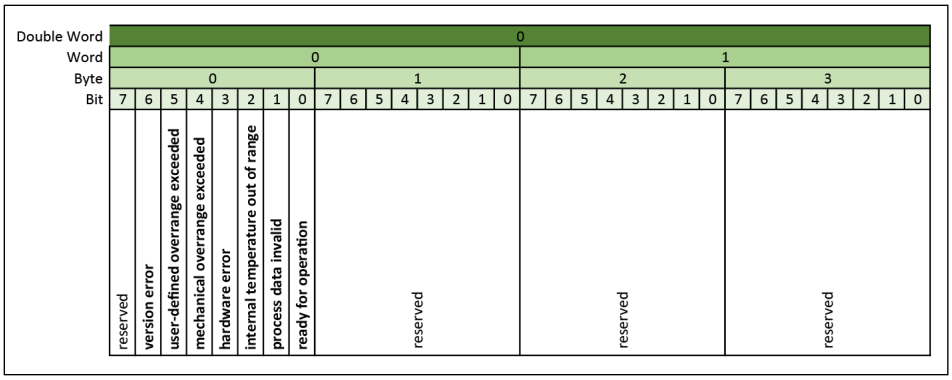
Data frame

The data frame of the process data consists of the packet ID, the status double word and the force/torque values. The packet ID for the process data is set to 0x01 for both TCP and UDP.

Byte	Content
0	Packet ID of the TCP and UDP process data: 0x01
1 – 4	Status double word
5 – 8	Fx as FLOAT 32 bit [N]
9 –12	Fy as FLOAT 32 Bit [N]
13 – 16	Fz as FLOAT 32 bit [N]
17 – 20	Tx as FLOAT 32 Bit [Nm]
21 – 24	Ty as FLOAT 32 Bit [Nm]
25 – 28	Tz as FLOAT 32 Bit [Nm]

Status double word

The status double word is transmitted in bytes 1 – 4 of the process data.



Word	Byte	Bit	Cyclical output data
0	0	0	ready for operation EN: ready for operation DE: Betriebsbereit
		1	process data invalid * EN: cyclic process output data invalid DE: Zyklische Prozessausgangsdaten ungültig
		2	internal temperature out of range EN: internal temperature out of range DE: Interne Temperatur nicht im zulässigen Bereich
		3	hardware error EN: hardware error DE: Hardware-Fehler
		4	mechanical overrange limits exceeded EN: mechanical overrange limits exceeded DE: Mechanische Überlastgrenzen überschritten
		5	user-defined overrange limits exceeded EN: user-defined overrange limits exceeded DE: Nutzerdefinierte Überlastgrenzen überschritten
		6	firmware version error EN: firmware version error DE: Firmware-Versionsfehler
		7	reserved
0	1		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.

Force-torque values

- The six force-torque values are transmitted in bytes 5 to 28 of the process data.
- The data format of each force or torque value is float 32 bit and represents a value in Newtons [N] or Newton meters [Nm].

2.3 Command set

The commands are transmitted from the control unit to the module. The module evaluates the commands and transmits a response back to the control unit. For correct assignment, the command ID of the command is always prefixed in the response. The individual commands of the force/torque sensor are described in detail below.

Execution of commands

Control commands sent to the module may be permissible or impermissible.

- Permissible commands are executed by the module. This is indicated to the control unit by returning the corresponding error code.
- Impermissible commands are not executed. This is indicated to the control unit by returning the corresponding error code.

2.3.1 Start process data output via TCP

The process data output must be started explicitly to avoid placing an unnecessary load on the network.

Structure of command package

Byte	Description
0	Command ID: 0x10

Structure of response package

Byte	Description
0	Command ID: 0x10
1	Error code

The command has no parameters and no return values apart from an error code.

2.3.2 Stop process data output via TCP

Stops the process data output via TCP.

Structure of command package

Byte	Description
0	Command ID: 0x11

Structure of response package

Byte	Description
0	Command ID: 0x11
1	Error code

The command has no parameters and no return values apart from an error code.

2.3.3 Tare

Tares the force-torque values.

Structure of command package

Byte	Description
0	Command ID: 0x12

Structure of response package

Byte	Description
0	Command ID: 0x12
1	Error code

The command has no parameters and no return values apart from an error code.

2.3.4 Reset tare value

The force-torque values are no longer tared.

Structure of command package

Byte	Description
0	Command ID: 0x13

Structure of response package

Byte	Description
0	Command ID: 0x13
1	Error code

The command has no parameters and no return values apart from an error code.

2.3.5 Restart

The module will restart.

Structure of command package

Byte	Description
0	Command ID: 0x20

Structure of response package

Byte	Description
0	Command ID: 0x20
1	Error code

The command has no parameters and no return values apart from an error code.

2.3.6 Switching tool settings

Selects the pre-configured memory bank of the tool settings

Structure of command package

Byte	Description
0	Command ID: 0x30
1	Tool Settings Index: Selection of the pre-configured memory bank of tool settings. Values from 0 – 3 are valid

Structure of response package

Byte	Description
0	Command ID: 0x30
1	Error code

The index of the tool settings is transferred in byte 1 of the command in order to select the preconfigured memory bank of the tool settings. The data format of the parameter is unsigned 8 bit. The command has no return values apart from an error code.

2.3.7 Switching the noise reduction filter

Selects the noise reduction filter.

Structure of command package

Byte	Description
0	Command ID: 0x31
1	Noise Reduction Filter Number: Selection of the noise reduction filter. Valid numbers are the values 0 – 4, which stand for a window size of 1, 2, 4, 8 or 16 values in ascending order.

Structure of response package

Byte	Description
0	Command ID: 0x31
1	Error code

The noise reduction filter is selected via a number in byte 1 of the command. The data format of the parameter is unsigned 8 bit. The command has no return values apart from an error code.

2.3.8 Start process data output via UDP (optional)

Starts the process data output via UDP. This command is only available for pure Ethernet modules and is only permitted if the process data output via TCP is stopped.

Structure of command package

Byte	Description
0	Command ID: 0x40

Structure of response package

Byte	Description
0	Command ID: 0x40
1	Error code

The command has no parameters and no return values apart from an error code.

2.3.9 Stop process data output via UDP (optional)

Stops the process data output via UDP. This command is only available for pure Ethernet modules.

Structure of command package

Byte	Description
0	Command ID: 0x41

Structure of response package

Byte	Description
0	Command ID: 0x41
1	Error code

The command has no parameters and no return values apart from an error code.

2.3.10 Reading parameters

Reads a parameter with the transmitted index and subindex. The available system parameters are described in Chapter ▸ 4 [🗋 19].

Structure of command package

Byte	Description
0	Command ID: 0xF0
1	Index of the parameter
2	
3	Subindex of the parameter

The index of the parameter is transferred in bytes 1 and 2 of the command. The data format of the index is unsigned 16 bit. The subindex of the parameter is transferred in byte 3 of the command. The data format of the subindex is an unsigned 8 bit.

Structure of response package

Byte	Description
0	Command ID: 0xF0
1	Error code
2	Index of the parameter
3	
4	Subindex of the parameter
5	Value of the parameter. Type, length and value depend on the parameter.
...	

The error code is transferred in byte 1 of the answer. The index of the requested parameter is transferred in bytes 2 and 3. The data format of the index is unsigned 16 bit. The subindex of the parameter is transferred in byte 4 of the command. The data format of the subindex is an unsigned 8 bit. The value of the parameter is transferred from byte 5. The length depends on the type of parameter. If an error has occurred, the value of the requested parameter is not transferred. The response then only contains the command ID, the error code and the index/ subindex.

2.3.11 Write parameters

Overwrites the value of the parameter with the transferred index and subindex. The available system parameters are described in chapter ▶ 4 [🗋 19].

Structure of command package

Byte	Description
0	Command ID: 0xF1
1	Index of the parameter
2	
3	Subindex of the parameter
4	Value of the parameter. Type, length and value depend on the parameter.

The index of the parameter is transferred in bytes 1 and 2 of the command. The data format of the index is unsigned 16 bit. The subindex of the parameter is transferred in byte 3 of the command. The data format of the subindex is an unsigned 8 bit. The value of the parameter is transferred from byte 4. The length depends on the type of parameter.

Structure of response package

Byte	Description
0	Command ID: 0xF1
1	Error code
2	Index of the parameter
3	
4	Subindex of the parameter

The error code is transferred in byte 1 of the response. The index of the requested parameter is transferred in bytes 2 and 3. The data format of the index is unsigned 16 bit. The subindex of the parameter is transferred in byte 4 of the command. The data format of the subindex is an unsigned 8 bit.

2.4 Error codes

An error code is transmitted with each response packet to inform the controller whether the command packet was valid and the command could be executed. All error codes and the corresponding descriptions are listed below.

Error code	Description
HEX 0x00 DEC 0	**None** The command was executed successfully.
HEX 0x01 DEC 1	**Unknown Command** The command ID was invalid and the command was not available.
HEX 0x02 DEC 2	**Invalid Command Length** The length of the command packet does not match the transmitted command.
HEX 0x03 DEC 3	**Invalid Command Value** The parameter transmitted with a command is invalid.
HEX 0x04 DEC 4	**Busy** The module is busy and cannot currently accept a command.
HEX 0x05 DEC 5	**Streaming Active** The command cannot be executed because a process data output has been started.
HEX 0x06 DEC 6	**Storage Error** An error has occurred while persistently saving parameters. If the error occurs repeatedly, SCHUNK Service must be contacted.
HEX 0x07 DEC 7	**Internal bus error** An error has occurred during internal communication between the interface box and the sensor. If the error occurs repeatedly, SCHUNK Service must be contacted.
HEX 0x08 DEC 8	**Timeout** The command was not executed because a timeout occurred.
HEX 0x10 DEC 16	**User Level Not Sufficient** The command cannot be executed because the user level is not sufficient.
HEX 0x11 DEC 17	**Is Read Only** The parameter with the transferred index and subindex can only be read.
HEX 0x12 DEC 18	**Is Write Only** The parameter with the transferred index and subindex can only be written.
HEX 0x13 DEC 19	**Index Does Not Exist** The parameter with the transferred index does not exist.
HEX 0x14 DEC 20	**Subindex Does Not Exist** The parameter with the transferred subindex does not exist.
HEX 0x15 DEC 21	**Parameter Value too Long** The length of the parameter value is too long.
HEX 0x16 DEC 22	**Parameter Value too Short** The length of the parameter value is too short.
HEX 0x17 DEC 23	**Invalid Parameter Value** The parameter value transmitted with the write command is invalid.
HEX 0x1A DEC 26	**Parameters Are Locked** The parameters of the memory banks of the tool settings are not yet unlocked and therefore cannot be overwritten. The memory banks must be enabled via the corresponding parameter.

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.

Module feedback

- 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 of the module is triggered by transmitting the restart command.

Module feedback

After the response to the restart command has been transmitted, the "ready for operation" status bit is reset and the restart is initiated.

3.2 Taring functions

3.2.1 Tare

Short description

The tare function sets the current force-torque values to zero when the corresponding command is transmitted.

An average value is formed over ten measured values and subtracted from future measurements.

Trigger

The tare function is triggered by the transmission of the tare command.

Module feedback

The end of the tare function is indicated by the response packet. If no error has occurred, the error code "None" is returned. If an error has occurred, the corresponding error code is returned.

3.2.2 Reset tare

Short description

The reset tare function restores the original measuring when the corresponding command is transmitted.

This means that the average value calculated in ▶ 3.2.1 [🗋 16] is no longer subtracted from future measured values.

Trigger

The reset tare function is triggered by the transmission of the "reset tare" command.

Module feedback

Ending the reset tare function is indicated by the response packet. If no error has occurred, the error code "None" is returned. If an error has occurred, the corresponding error code is returned.

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 switching function is triggered by transmitting the "Switch tool settings" command.

Command parameters

To switch between tool settings, the command parameter must be transferred to the module.

Valid indices are the values 0 – 3.

Module feedback

- The end of the function is indicated by the response packet. If no error has occurred, the error code "None" is returned.
- If an error has occurred, the corresponding error code is returned. This mainly occurs if an invalid value in the <Tool Settings Index> has been 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 switching function is triggered by transmitting the "Switch noise reduction filter" command.

Command parameters

To switch between noise reduction filters, the command parameter must be transmitted 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.

Module feedback

- The end of the function is indicated by the response packet. If no error has occurred, the error code "None" is returned.
- If an error has occurred, the corresponding error code is returned. This mainly occurs if an invalid value in <Noise Reduction Filter Number> has been transferred 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 0x0001/0
DEC 1/0

<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 0x0001/1
DEC 1/1

<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 0x0001/2
DEC 1/2

<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 0x0001/3
DEC 1/3

<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 0x0002/0
DEC 2/0

<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 0x0003/0
DEC 3/0

<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 0x0003/1
DEC 3/1

<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 0x0035/0
DEC 53/0

<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 0x0060/0
DEC 96/0

<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 0x0061
DEC 97

<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 0x0062
DEC 98

<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 0x0063
DEC 99

<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 0x0064
DEC 100

<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 0x0066
DEC 102

<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 0x0067
DEC 103

<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 0x0068
DEC 104

<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 0x1000/0
DEC 4096/0

<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 0x1000/1
DEC 4096/1

<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 0x1001/0
DEC 4097/0

<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 4097/1

<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 0x1002/0
DEC 4098/0

<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 0x1002/1
DEC 4098/1

<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 0x1003/0
DEC 4099/0

<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 0x1003/1
DEC 4099/1

<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 0x1020/0
DEC 4128/0

output_rate_udp_ethernet> (optional)

Short description:	The process data rate of the UDP can be read out and written with this parameter. The parameter only has an effect on pure Ethernet modules.
Parameter name:	Output data rate UDP
Access rights:	Read and write
Data type:	ENUM
Enumeration:	0 = 1 kHz 1 = 500 Hz 2 = 250 Hz 3 = 100 Hz
Default:	0 = 1 kHz

HEX 0x1021/0
DEC 4129/0

<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 0x1030/0
DEC 4144/0

<use_static_ip_address>

Short description:	This parameter can be used to choose between a static IP address and DHCP. The setting is only effective for modules with Ethernet/IP industrial bus or pure Ethernet modules. A restart is required for the setting to take effect.
Parameter name:	Use static IP address
Access rights:	Read and write
Data type:	BOOL
Values:	TRUE = Use static IP address FALSE = Use DHCP
Default:	FALSE

HEX 0x1032/0
DEC 4146/0

<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.

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, Link Zubehör.

5.3 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**.

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

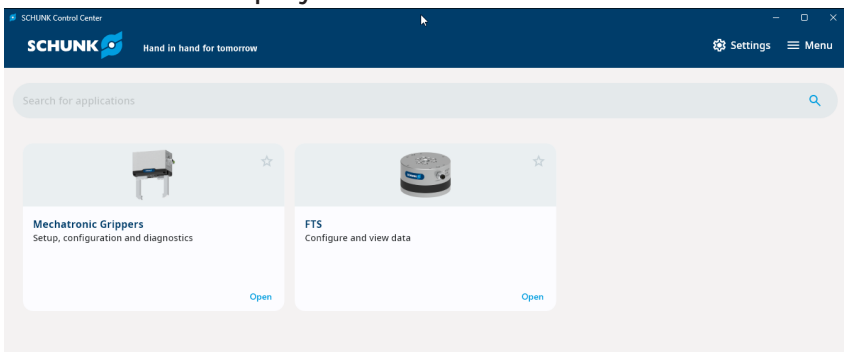
Start software

NOTE

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

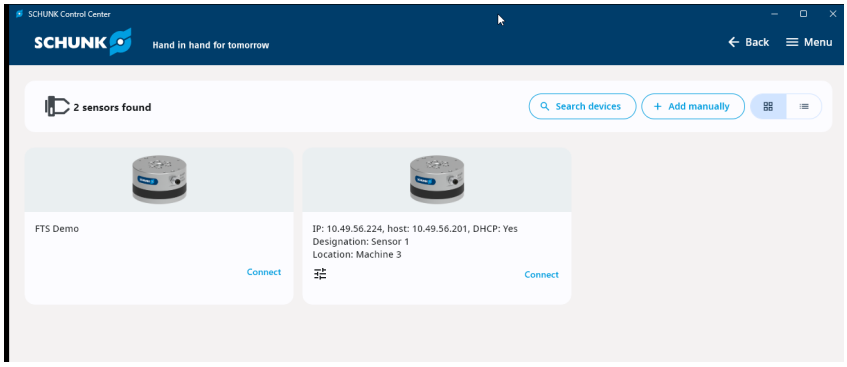
- Module is electrically connected to the power supply unit.
- SCHUNK Control Center is installed.

1. Connect the computer directly to the module via Ethernet.
   OR:
   Connect the computer to the network in which the module is integrated.
2. Open SCHUNK Control Center.
   ⇨ Start screen is displayed.



3. Select the *FTS* app.
   ⇨ The system automatically searches for sensors in the network.

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



4. Select the desired sensor.

⇨ The app connects to the sensor.

⇨ Access to the functions of the sensor is possible.

6 Appendix

6.1 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 Cyclical input data.

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 – process data invalid

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

Bit 2 – 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 3 – 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 4 – 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 5 – 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 6 – 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 7 to 31 – reserved

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

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*

01-2026

04.00 | FTS | Commissioning instructions | en-US | 1634813-EN FW 2.1.0