angewandte Microelectronic

Haller + Erne GmbH

MSTKN-HM-UM

MSTKN Socket tray user manual

HEI-24-0314

Revision R21

Revisions

Revision	History	Author	Date
R20	Updated from R19, added EtherCAT	he	2024-03-14

About

This document provides information about the MSTKN socket trays

Table of contents

1 Introduction

1.1 System Overview

The MSTKN family of modular socket selector trays provide a flexible solution for monitoring and controlling manual tool operations. Some of the main features are:

- Multiple mechanical and enclosure options for tool diameters in the 5…65mm range. Available with fixed or variable tool trays.
- Selection of multiple I/O interfaces: industry standard 24V-I/O-interface and various field bus systems (Profibus-DP, ProfiNet, EtherCAT Ethernet/IP, Indralogic-CAA/Modbus-UDP/TCP)
- Extensible architecture allows connecting multiple devices over an internal bus system to build a socket selector for up to 60 tools with a single I/O connection.

All devices of the MSTKN are built from a combination of up to two basic elements, which are combined in different ways and assembled into different mechanical housings. These elements are:

- **Gateway-element**: A gateway-element provides the connection between an I/O interface and the internal bus system. Gateway-elements are available for the industry standard 24V-I/O-interface and for various field bus systems (Profibus-DP, ProfiNet, Ethernet/IP, IndraLogic-CAA, …). A gateway-element connects the socket selector system with a process controller or the tightening system.
- **Sensor-element**: A sensor-element consists of sensors for detecting sockets in the socket trays and a connection to the internal bus system. Sensor-elements are available in different mechanical shapes (number of sensors, mechanical sizes – for small and big sockets) and technologies (simple socket presence detection or socket unique ID detection).

The following device types are available in the MSTKN socket selector system:

- **Main module (HM)**: A main module consists of a gateway-element and one or more sensor-elements assembled into a single enclosure. A main module device is a full featured self-contained socket selector, ready for use with a PLC or a tightening system. It is expandable by connecting additional extension modules over the internal bus connector to provide additional sockets.
- **Extension module (EM)**: An extension module consists of one or more sensor elements assembled into a single enclosure. An extension module does not work stand-alone, it must always be connected to either a MSTKN main module (HM) or a MSTKN gateway module (GM).
- **Gateway module (GM)**: A gateway module consists of a single gateway element assembled into an enclosure, normally for mounting inside a cabinet (DIN-rail mount). A gateway module allows connecting one or more extension modules (EM) to a 24V-I/O interface or some field bus system.

All devices of the MSTKN system are compatible with each other and can be connected over the internal bus system. To build a socket selector tray system, you simply need to select a set of devices from the MSTKN system consisting of exactly one gateway element and one or more sensor elements. You may combine these devices regardless of enclosure type and sensor element type to build a flexible and perfectly adapted solution for your requirements.

1.2 Ordering Information

The ordering codes for the members of the MSTKN socket selector family use the following scheme:

MSTKN – H – S14 – HM – DB25M – OP

Enclosure
H = Aluminium enclosure
P = Enclosure vor variable trays

Socket tray type
S = Single plugs Ø 38mm
T = Common block for standard bits
Q = Single plugs Ø 70mm
P = variable prismatic tray

Socket number
13 = single row, 3 sockets
14 = single row, 4 sockets
18 = single row, 8 sockets
24 = two rows, 4 sockets per row (total: 8 sockets)
25 = two rows, 5 sockets per row (total: 10 sockets)
(etc)

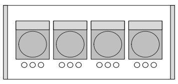
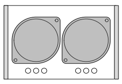
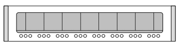
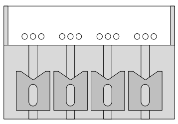
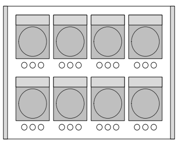
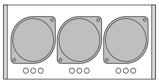
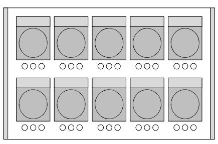
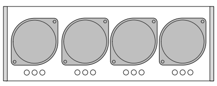
Options
(none) standard
OP = Open Protocol

Interface type
Only for Modules HM and GM:
DB25M = 24V I/O
PB = Profibus DP Slave M12
IOL = IO-Link M12
ETH=Ethernet/Modbus M12
PN/PN2P = Profinet IO-Device M12
EIP/EI2P = Ethernet/IP Device M12
ECT/EC2P = EtherCAT M12
WLAN=Wifi/Modbus M12

Module type
HM = Main module
EM = Extension module
GM = Gateway module

The MSTKN devices described here are available as Main modules and extension Modules in typically four different enclosure variants each. For all main modules, there is a standard type, an “OP” (open protocol) option for use with the “Open Protocol” implementation of the System 350 tightening controllers (and IM24 interface) and an “XML” version with different signals and color codings to comply with Audi standards.

1.2.1 Variant matrix

Tool **max. Ø 38mm**	**Tool** **max. Ø 70mm**	**Tool** **max. Ø 18mm**	**Prismatic** **max. Ø 48mm**
H-S14 	H-Q12 	H-T18 	P-P14 
H-S24 	H-Q13 		
H-S25 	H-Q14 		

1.3 Technical Data

The technical specifications for all types of MSTKN are located in their respective datasheets.

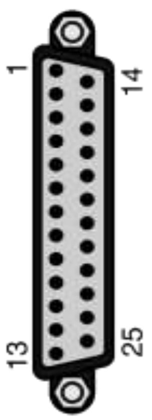
1.4 Interface Connections

1.4.1 DB25M 24V-I/O

This connector is only available on main modules and gateway modules. The interfaces provides industry standard compatible 24V I/O signals (11 digital input signals, 10 digital output signals and two directly connected LED's). The pinout is 1:1 compatible to the System 350 IM24V interface module (on the XDD1-interface), so an inexpensive DB25 extension cable can be used.

For a description on how the input and output signals are used, please refer to the functional description of the MSTKN main devices below.

DB25 male



MSTKN Input Signals

Pin	Signal	Beschreibung	IM24V	
19	I0	Input bit 0	A1	B0.0
7	I1	Input bit 1	A2	B0.1
20	I2	Input bit 2	A3	B0.2
8	I3	Input bit 3	A4	B0.3
21	I4	Input bit 4	A5	B0.4
23	I5	Input bit 5	A9 (!)	B5.0
11	I6	Input bit 6	A10 (!)	B5.1
12	I7	Input bit 7	A12 (!)	B5.3
9	I8	Input bit 8	A6	B0.5
22	I9	Input bit 9	A7	B0.6
10	I10	Input bit 10	A8	B0.7

Additional Signals and Power Supply

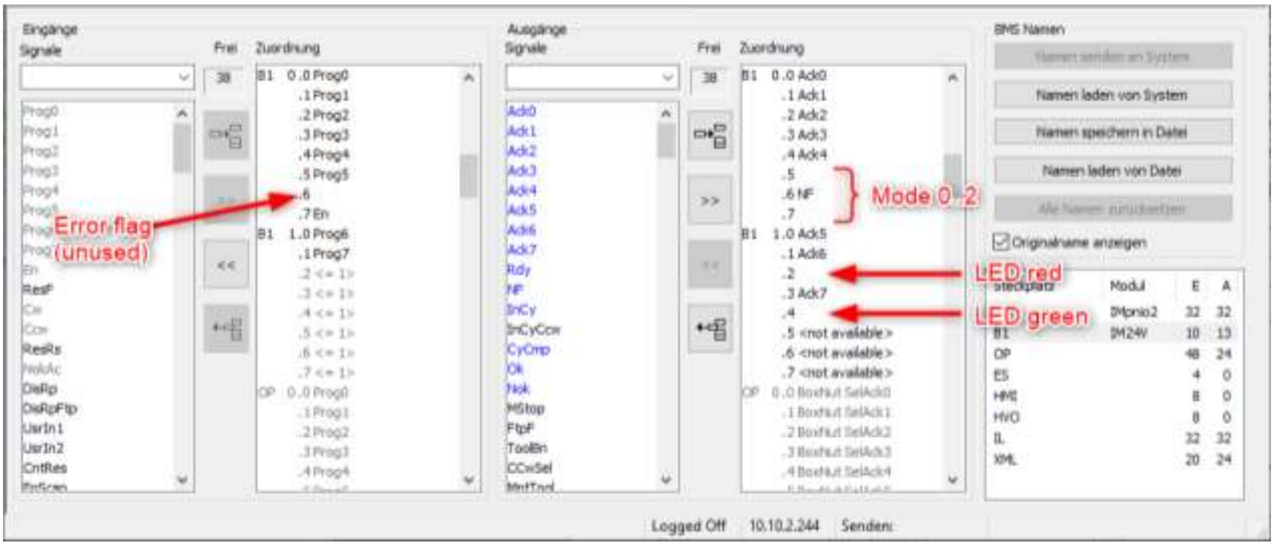
Pin	Signal	Beschreibung	IM24V	
1	24V	24V Power supply		
13	0V	Power ground 0V		
24	LEDR	LED red	A11(!)	B5.2
25	LEDG	LED green	A13	B5.4

MSTKN Output Signals

Pin	Signal	Beschreibung	IM24V	
14	Q0	Output bit 0	E1	B0.0
2	Q1	Output bit 1	E2	B0.1
15	Q2	Output bit 2	E3	B0.2
3	Q3	Output bit 3	E4	B0.3
16	Q4	Output bit 4	E5	B0.4
4	Q5	Output bit 5	E6	B0.5
18	Q6	Output bit 6	E9 (!)	B1.0
6	Q7	Output bit 7	E10 (!)	B1.1
17	Q8	Output bit 8	E7	B0.6
5	Q9	Output bit 9	E8	B0.7

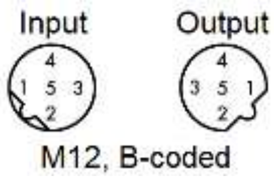
NOTE: To be compatible with the first hardware version of the MSTKN the assignment between IM24V and MSTKN is not 1:1 (please refer column IM24V!).

Typical configuration in BS350 (in direct operation mode with wire break detection):



1.4.2 M12 Profibus

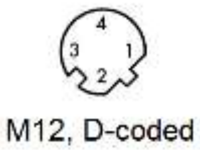
The Profibus Main module comes with two M12 fieldbus connectors (Bus IN, Bus OUT). The pinout of the 5pole socket (Out) and plug (In) are according to Profibus standards.



M12 B-kodiert	Pin	Signal	Description/Function
	1	+5V	Power supply (only Bus OUT, max. 100mA)
	2	A	TX-/RX- Bus Signal
	3	0V	Power supply ground (0V)
	4	B	TX+/RX+ Bus Signal
	5	-	Not connected

1.4.3 M12 Ethernet

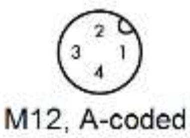
The ethernet-based main modules are connected by a 4-Pin, D-coded M12 connector with standard assignment (ProfiNet, EtherCAT and Ethernet/IP uses two ports).



M12 D-coded, female	Pin	Signal	Description/Function
	1	Tx+	Transmit +
	2	Rx+	Submit +
	3	Tx-	Transmit -
	4	Rx-	Submit -

1.4.4 M12 Power Supply

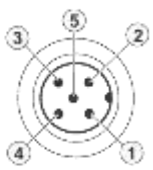
The power is supplied through a 4-Pin, A-coded M12 connector.



M12 A-coded, male	Pin	Signal	Description/Function
	1	+24V	Power supply (IN)
	2		Not used
	3	0V	Power supply ground (0V)
	4		Not used

1.4.5 M12 Internal Bus Connectors

Depending on the type of device there are one or two M12 (A-coded, 5 pins) connectors used for the internal bus connection between the MSTKN modules. For main and gateway modules only a bus out connector is available. For extension modules two connectors („bus-in“ and „bus-out“) are available. The bus out (M12-female) provides power for the extension devices.


male – bus in


female – bus out

M12 A-kodiert	Pin	Signal	Beschreibung/Funktion
	1	+5V	Power supply (bus in: input, bus out: output)
	2	A	TX-/RX- bus signal
	3	0V	Power supply reference
	4	B	TX+/RX+ bus signal
	5	-	Not connected

The internal bus connection can also be used to diagnose and parameterize the socket holder system. Further information: see chapter 4.

1.4.6 Configuration Interface

The interface for the internal bus connection (as in 1.4.5) can be used to configure and diagnose the socket tray system. This function requires the MSTKN Programming Cable and the accompanying software.

The configuration interface can be used for additional functions, like activating the tool grouping or tool recognition.

Further information on this can be found in chapter 4.

2 Functional Description

2.1 General Features

2.1.1 24V I/O

The 24V I/O master and gateway modules provide easy to use socket selector devices. The main features are:

- industry standard compatible 24V I/O signals
  - 13 Inputs (11 Digital input signals, 2 LEDs)
  - 10 Digital output signals
- Three LED's per socket tray for status indication and worker guidance
- Tightening system 350 pin compatible DSub25 I/O-connector
- Multiple operation modes for increased flexibility (guided and non-guided modes)
- Increased security due to special loopback mode and hardware cable connection verification
- Two variants for socket number encoding: one I/O-signal per socket for maximum flexibility (option – OP) or binary encoded socket number (standard option) for maximum number of addressable sockets

2.1.2 Profibus

The modules of the "Profibus" socket tray category offer the following functions:

- Profibus-Slave
  - Adress 1-99 can be set with two rotary switches
  - Dynamic addressing possible
- Three LED's per socket as status indicator and worker guidance
- Standard M12 Profibus-Port (Bus IN and Bus OUT)
- Multiple operating modes for flexible use (with/without worker guidance)
- Increased process safety through special operating modes („direct mode") and wire check
- Configuration by modules from the GSD File
  - MSTKN intelligent mode
  - MSTKN user mode

2.1.3 Ethernet

The modules of the „Ethernet" socket tray category (Profinet, OpenModbus, Ethernet/IP) offer the following functions:

- Setting the IP-Adress
- Static IP-Adress configured by rotary switches between 192.168.0.**3** – 192.168.0.**98** (then Netmask 255.255.255.0 and Gateway 192.168.0.1)
  - Configurable by a software tool (HICP)
  - Dynamic over DHCP
  - Dynamic over a fieldbus-specific mechanism
- Three LED's per socket as status indicator and worker guidance

- Standard M12 Profibus-Port (D-coded)
- Multiple operating modes for flexible use (with/without worker guidance)
- Increased process safety through special operating modes („direct mode“) and wire check
- Configuration over „MODE“-Bit in the data telegram
  - MSTKN intelligent mode
  - MSTKN user mode
- 66 Byte I/O data exchange

operating mode and IP-address can be configured by rotary switches on the back side of the MSTKN main module as shown below: (the picture shows the print on the back side of the main module):

Position	Mode	Description
00	Default	Standard-configuration. The previously saved configuration is used. This mode is usually used for assembly lines and it offers the highest amound of safety, as all optional configurations are inactive. The fieldbus-specific configuration will stay active nonetheless (e. g. address- and name preset over DCP with Profinet).
01	HICP	The previously saved configuration is used. In addition, the HICP-function is active. This allows the remote-configuration of the IP-Adress over the AnyBus-IP-Config software.
02	DHCP	The previously saved configuration is used. In addition, the socket tray will try to obtain an IP-configuration from a DHCP-Server. This configuration will also be saved.
03-98	Static	The IP address is statically determined by the position of the rotary switches in the range of 192.168.0.X (X being the switch position between 3 and 98). Netmask and Gateway address are static: Netmask = 255.255.255.0 Gateway = 192.168.0.1
99	Admin	Not for WizNet! Only the previously saved configuration is used. The previously saved configuration is used. In addition, admin-mode is active, meaning all Network services available in the system are activated and the system configuration can be changed over Network (e. g. Firmware-Update). active services are (depending on the actual ethernet-fieldbus type): - HICP (for IP-configuration look above) - DHCP (for IP-configuration look above) - ARP Gleaming (for IP-configuration) - Web/FTP/Telnet Server (if available for this type), possibly also for firmware-updates

2.2 Calibration

The purpose of the calibration modes is to maximize sensor sensitivity and recognize extension modules.

WARNING: It is required to remove all tools from the device before the calibration is started. If calibrated incorrectly, the socket tray will not work properly.

2.2.1 Standalone Use

The socket trays are calibrated before shipping, so normally it’s not necessary to calibrate them when using one socket tray in standalone mode.

2.2.2 Multiple socket trays (externsion bus)

When setting up a chain of socket trays, there are a few rules to follow to get a correct configuration. The chain can't be configured at once. One module must be added at a time, starting with the module that is supposed to get the socket numbers 1 ... 4 (or 1 ... 8 when using an 8-way module). Each subsequently added module will get the following socket numbers.
When chaining a gateway module (WiFi, Ethernet, ProfiNET, Profibus, 24V I/O) with one or more extension modules, the gateway module must be connected first. When only using extension modules in a chain, connect any of the devices first.

Please follow these steps to configure the system correctly:

1. Connect the first socket tray. (extension module or gateway)
2. Find the "Calibration" signal in the Input signal reference section for your module:
   - Chapter 2.4 (Detailed Description of the 24V I/O Operation Modes)
   - Chapter 2.5 (In/Out Profibus/Profinet)
   - Chapter 2.6 (In/Out Ethernet)
3. Send the calibration signal to the first socket tray. Its LEDs should start flashing yellow.
4. Remove the calibration signal. The socket tray will power down and restart automatically.
5. Add the next socket tray (extension module).
6. Send the calibration signal to the first socket tray. Its LEDs should start flashing yellow.
7. Remove the calibration signal. The socket tray will power down and restart automatically.
8. Add the next socket tray. (extension module).
9. ....
10. Repeat this procedure until all modules are connected to the chain and configured.

Important: For ProfiNet-Configurations in „User mode", ensure the correct sequence, as the calibration signal can only be sent with a "correct" configuration. To add a new extension module, do the following::

1. **Plug the extension module (without changing the PLC-configuration) and send the calibrartion signal**
2. **Change the PLC configuration (i.e. plug the addition module in the HW config)**

Changing the PLC configuration without running a calibration cycle before, the PLC will not be able to connect to the tray anymore (due to a hardware difference) – therefore you can't start a calibration cycle anymore!

Note: When a chain including a gateway module was configured incorrectly or a gateway module is to be used in a different setup (chain or standalone), please delete the old calibration by sending the signal "Reset Calibration" to the main module with no extension module connected. Please see the I/O signal reference sections for your module to find the "Reset Calibration" signal.

2.2.3 Unused places

If a place should be left unused, then there are two options for disabling it.

- Option 1: Stick the copper tape onto the plug of the unused place and mount it. The LED of this socket will then continuously light yellow and reports the socket as idle. If you want to use the place for a tool later, then simply remove the copper tape.
- Option 2: Do the same as in Option 1 (attach the copper tape). Then run a calibration cycle, while the plug with the copper tape is in the socket. At the end of the calibration cycle, the LED will get switched of – the calibration detects the plugged socket and disables the place.To reactivate the socket, unmount the plug with the copper tape and run a calibration cycle again.

2.3 Operation Modes

The socket selector devices basically feature the following operation modes:

- **Free selection**: In this mode, the worker is allowed to freely take (remove) a single socket available in the trays. The number of the socket selected by the worker (removed from the tray) is then reported to the digital 24V-I/O signals. If more than one tool is removed, then an error is reported. The LED's on the socket trays only indicates if a socket is detected in the corresponding tray (or not) or if there is an error (more than one socket removed).
- **Guided operation**: In this mode the external controller (e. g. tightening controller or PLC) sets some of the 24V-I/O signals to define which socket the worker should use in the next operation step. The socket selector device then indicates this by blinking the LED on the corresponding tray, thus guiding the worker on what to do next. If the correct socket is removed by the worker, then this gets reported back to the controller and the LED state on the tray changes to acknowledge correct removal to the worker. If the wrong socket is removed (or more than one), this is also indicated on the LED's as well as on the 24V-I/O interface. Depending on the MSTKN device option, the controller can either allow the worker to choose exactly one socket (standard option) or to select from a set of allowed sockets (option –OP).
- **Direct operation**: This mode is similar to "Free selection" mode but provides additional monitoring of the digital signal lines to detect short or open circuits on the connection wires. In this operation mode, the socket selector device expects the controller (e. g. tightening controller or PLC) to "mirror" the selection of the worker back to the socket selector device (by setting the corresponding 24V input signals)
- **Diagnostic and configuration modes:** Depending on the actual device type, there are additional operation modes for diagnostics (e. g. wiring check) and configuration (e. g. sensor calibration). Please see the detailed description below.
- **Intelligent mode:** This mode is used by the fieldbus types (Profibus, Modbus UDP etc.). In this mode, the socket selector device controls both the LEDs, and the comparison of the states of each tray and the state of the preselected tray is done by the device itself.
- **User mode**: This mode is also only used by the fieldbus types. In this mode, the user takes full control over the operation and analysis of the socket selector device signals. This means the user can create any desired LED pattern and can allow the removal of multiple tools.
- **Grouping:** After Firmware-version 1.6, groups can be defined in „Intelligent Mode". These groups behave like a single tool on the I/O-Interface, but all tools of a group must be removed in order to get clearance. When preselecting a group, all the green LED's of that group are flashing until all tools have been removed – only then the clearance for tightening the bolt(s) will be given. The group-function is an extended function that can only be activated and configured with a programming cable (go to chapter 4.4 for further information on grouping).

NOTE: The free selection mode is the basic operation mode of the device and is similar (compatible) to that of simple socket selector boxes (e. g. single proximity sensors per tray). However, it is generally not recommended to use this operation mode, as there is no way of verifying if the 24V-signals read by the controller (tightening controller or PLC) are correct, i. e. show the correct selected socket (think of a broken signal wire!). For increased process security always use the **direct operation** mode instead!

2.4 In/Out EA24

2.4.1 Overview

Input and output signals (see chapter 1.4.1) are mapped as follows:

24V-I/O Input signals: Inputs I0…I10:

I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0		
			0…255 (0xFF)								Socket number preselection signals	8 signals for setting the preselection socket number (e. g. for worker guidance indication)
0…7											Control signals	3 signals used for controlling the socket selector operation and selecting the operation mode

24V-I/O output signals: Outputs Q0…Q9:

Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0		
		0…255 (0xFF)								Socket number selection signals	8 signals for indicating the currently selected socket (e. g. used for selecting the tightening program number)
0…3										Status signals	2 signals indicating the current operating status of the socket tray

2.4.2 Device options

The MSTKN 24V-E/A socket trays are available in different software variants, where the encoding of the I/O-signals are changed and/or the LED colors are different.

2.4.2.1 Standard

This option uses binary encoding for socket tray numbers. As eight signal lines (see above) are reserved for socket numbers, theoretically a maximum of 255 sockets could be addressed in the system (number zero is reserved for no socket selected). However, the currently supported maximum total number of sockets is 63. Please note, that only a single tool (or tool group) can be used in this mode.

Binary encoding works as follows:

7	6	5	4	3	2	1	0	**Signal number (Q0…Q7 or I0…I7)**
0	0	0	0	0	0	0	1	Number = $1*2^0 = 1$
0	0	0	0	0	0	1	0	Number = $1*2^1 = 2$
0	0	0	0	0	0	1	1	Number = $1*2^1 + 1*2^0 = 2 + 1 = 3$
0	0	0	1	1	1	1	1	Number = $1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 1*2^0 = 16 + 8 + 4 + 2 + 1 = 31$
…								…

2.4.2.2 Open Protocol (OP)

This option uses a single signal per socket tray. As eight signal lines (see above) are reserved for socket numbers, a maximum of 8 sockets is possible. Each of the 8 signal lines is directly mapped to one of the socket trays (I0/Q0 for socket #1, I1/Q1 for socket #2, …). Please note that with this option multiple pre-selections are supported, so in this case the worker could choose from a (sub-) set of allowed sockets, but it is only allowed to remove one tool. Removing more than one tool from the socket tray will result in an invalid return signal to the control interface and no release signal will be set.

Open Protocol encoding works as follows:

7	6	5	4	3	2	1	0	**Signal number (Q0…Q7 or I0…I7)**
0	0	0	0	0	0	0	1	Signal 0 → Socket 1 selected
0	0	0	0	0	0	1	0	Signal 1 → Socket 2 selected
0	0	0	0	0	0	1	1	Signal 0,1 → Socket 1 and socket 2 pre-selected (only valid on inputs I0…I7)
…								…

2.4.2.3 XML

The XML version is a modified “Open Protocol” version. It uses different color coding and return signals to meet international Audi production standards. It uses the same bit-coded signals as the “Open Protocol” version
(1 bit = 1 socket = 1 signal line, maximum sockets = 8).

For details on the bit-coded signals please see above in section 2.4.2.2 “Open Protocol”.

As opposed to the standard & OP version, return signals from the socket tray to the control interface include all sockets that were removed, even when falsely removed. Using the XML version, it is also possible to remove multiple preselected tools from the tray at the same time and get back a release signal from the control interface while OP MSTKN only allows one tool to be removed at a time.

The following chart shows the color coding that XML socket trays are using:

- Preselection bit set for a socket (this tool is to be used for this job)

State	Red LED	Yellow LED	Green LED
Tool not removed	ON	OFF	ON
Tool is removed (no tool from another socket is removed))	ON	OFF	OFF
Tool is removed(another tool is removed from its socket aswell)	ON	Flashing	OFF

- Preselection bit not set for a socket (this tool must not be used for this job)

State	Red LED	Yellow LED	Green LED
Tool not removed	OFF	OFF	ON
Tool (falsely) removed	OFF	Flashing	OFF

2.4.3 Operation Mode

The operation mode for the MSTKN 24V-I/O devices is selected by setting the control signals (I8…I10) to one of the following allowed settings:

24V-I/O Input signals: Inputs I0…I10:

Byte 1			Byte 0								Operation Mode
I 10	I 9	I 8	I 7	I 6	I 5	I 4	I 3	I 2	I 1	I 0	
0	0	0	0								Free selection
			Non-zero								Guided operation
0	0	1	0								No operation
0	1	0	Any value								Direct operation
1	0	0	0								Calibration
1	0	1	0								Reset Calibration
1	1	1	1	1	1	1	1	1	1	1	Wire check

All other combinations of the control signals (I8…I10) are reserved and should not be used.

2.4.3.1 Free Selection

The “free selection” operation mode is selected by setting the control signals to zero and the socket pre-selection signals also to zero.

24V-I/O Input signals: Inputs I0…I10:

Byte 1			Byte 0								Operation Mode
I 10	I 9	I 8	I 7	I 6	I 5	I 4	I 3	I 2	I 1	I 0	
0	0	0	0								Free selection

In this mode, the output signal assignment is as follows:

- Q8 = correct selection: this signal is set (activated), if a correct selection has been made by the worker (a single socket has been removed from the tray).
- error: this signal is set, if more than one socket has been removed (multiple selections) be the worker.

24V-I/O output signals: Outputs Q0…Q9:

Byte 1		Byte 0								State	LEDs	Comment
Q 9	Q 8	Q 7	Q 6	Q 5	Q 4	Q 3	Q 2	Q 1	Q 0			
0	0	0								Idle	All yellow	No socket removed from tray
1	0	> 0								Selected	selected green, rest yellow	A single socket is removed from tray, its LED is set to green and the Q0…Q7 signals are set according to the selected socket (see chapter 2.4.1 for more info)
0	1	0								Error	selected red, rest yellow	More than one socket is removed from trays – the LED’s on the removed sockets are set to red.

2.4.3.2 Guided Operation

The “guided operation” mode is selected by setting the control signals to zero and the socket pre-selection signals to a non-zero value. Depending on the MSTKN option (standard, OpenProtocol or XML), the controller should use a binary (standard) or a bitwise (OP and XML) coded pre-selection on the input signals I0…I7 (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** for more info):

24V-I/O Input signals: Inputs I0…I10:

Byte 1			Byte 0								Operation Mode
I 10	I 9	I 8	I 7	I 6	I 5	I 4	I 3	I 2	I 1	I 0	
0	0	0	socket pre-select (> 0)								Guided operation

In this mode, the output signal assignment is as follows:

- Q8 = correct selection: this signal is set (activated), if a correct selection has been made by the worker (a single socket has been removed from the tray).
- Q9 = error: this signal is set, if more than one socket has been removed (multiple selections) be the worker.

24V-I/O output signals: Outputs Q0…Q9:

Byte 1		Byte 0								State	LEDs	Comment
Q 9	Q 8	Q 7	Q 6	Q 5	Q 4	Q 3	Q 2	Q 1	Q 0			
0	0	0								Idle	Selected green blinking, rest yellow	No socket removed from tray. The LEDs of the preselected sockets are blinking green to indicate allowed selections.
1	0	> 0								Selected	selected green, rest yellow	Ein Werkzeug wurde entnommen, die Ausgangssignale entsprechen der entnommen Steckplatznummer (siehe Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.**)
0	1	0								Error	selected red, rest yellow	Mehr als ein Werkzeug entnommen. Die LEDs der entnommen Steckplätze leuchten rot.

2.4.3.3 Direct Operation

This operation mode has the same features as the “free selection” mode but adds a wire check mode. The tray sets its outputs as in the “free selection” mode but requires the outputs to be mirrored back within 150ms. Only if the mirrored signals are correctly read back, the enable output is set. Else the enable bit is not set and the LEDs on the tray show an error pattern.

24V-I/O Input signals: Inputs I0…I10:

Byte 1			Byte 0								Operation Mode
I 10	I 9	I 8	I 7	I 6	I 5	I 4	I 3	I 2	I 1	I 0	
0	1	0	Mirror value								Direct operation

In this mode, the output signal assignment is as follows:

- Q9 = correct selection: this signal is set (activated), if a correct selection has been made by the worker (a single socket has been removed from the tray) and the mirror value given is identical to the output value Q0…Q7.
- Q8 = error: this signal is set, if more than one socket has been removed (multiple selections) be the worker or the mirror value is incorrect (there is a small delay allowed for setting the acknowledge signals here).

MSTKN output signals: Outputs Q0…Q9:

Byte 1		Byte 0								State	LEDs	Comment
Q 9	Q 8	Q 7	Q 6	Q 5	Q 4	Q 3	Q 2	Q 1	Q 0			
0	0	0								Idle	All yellow	No socket removed from tray
1	0	> 0								Selected	Entnommen Steckplatz grün, Rest gelb	A single socket is removed from tray, its LED is set to green and the Q0…Q7 signals are set according to the selected socket. (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** for more info)
0	1	0								Error	selected red, rest yellow	More than one socket is removed from trays – the LED's on the removed sockets are set to red.

If the acknowledge is incorrect then an error pattern (as described in chapter 3) is shown.

2.4.3.4 No Operation

The "no operation" mode is selected by setting the control signal I8 and the socket pre-selection signals to zero. This mode can be used to check whether all sockets are in the tray or not and showing an error to the user, if one or more are removed:

MSTKN Input signals: Inputs I0…I10:

Byte 1			Byte 0								Operation Mode
I 10	I 9	I 8	I 7	I 6	I 5	I 4	I 3	I 2	I 1	I 0	
0	0	1	0								No operation

In this mode, the output signal assignment is as follows:

- Q8 = error: this signal is set, if not all sockets are in their trays.
- Q9 = 0 (always zero)

MSTKN output signals: Outputs Q0…Q9:

Byte 1		Byte 0								State	LEDs	Comment
Q 9	Q 8	Q 7	Q 6	Q 5	Q 4	Q 3	Q 2	Q 1	Q 0			
0	0	0								Idle	all yellow	No socket removed from tray.
0	1	0								Error	selected red, rest yellow	One or more sockets removed from their trays – the LED's on the removed sockets are set to red.

2.4.3.5 Kalibrierung

The "calibration" mode is used to maximize sensitivity of the sensors, "teach" socket characteristics or to add one or more Extension modules. It is normally not necessary to use calibration in the case of the MSTKN 24V-I/O devices, as they are factory calibrated. These devices do not support socket identification and socket "teaching", so only sensitivity maximization is supported. This type of calibration is activated by removing all sockets from their trays (also unused trays) and activating signal I10.

Please note: Make sure to remove all sockets from the trays before starting a calibration cycle!

24V-I/O Input signals: Inputs I0…I10:

Byte 1			Byte 0								Operation Mode	
I 10	I 9	I 8	I 7	I 6	I 5	I 4	I 3	I 2	I 1	I 0		
1	0	0	0								Calibration	Calibration to maximize sensor sensitivity
1	0	1	0								Reset Calibration	Delete existing calibration

In this mode, the output signal assignment is as follows:

24V-I/O output signals: Outputs Q0…Q9 while calibration is running:

Byte 1		Byte 0								State	LED's	Comment
Q 9	Q 8	Q 7	Q 6	Q 5	Q 4	Q 3	Q 2	Q 1	Q 0			
0	0	1	1	1	1	0				Running	LED's	Calibration is running (takes just a few milliseconds)

24V-I/O Outputs: Outputs Q0…Q9 after calibration is completed:

Byte 1		Byte 0								State	LEDs	Comment
Q 9	Q 8	Q 7	Q 6	Q 5	Q 4	Q 3	Q 2	Q 1	Q 0			
0	0	1	0	0	0	#sockets				Done	All yellow	Calibration is completed – number of sockets calibrated is reported in the lower 4 bits (binary coded)

Please note: Please note: The calibration input pattern (I10 = 1, I0…I9 = 0) must remain active until the "calibration done" acknowledge (Q7 = 1, Q4…Q6 = 0) from the socket selector device is received. Thereafter switching to another mode is allowed.

2.4.3.6 Wire Check

The "wire check mode" mode sets all output signals to "1" and can be used to verify cabling in case of I/O-connectivity problems (note: you may want to set this mode during system startup and possibly combine it with "no operation" mode to verify cabling from time to time). This mode is activated by setting all input signals I0…I10 to non-zero:

MSTKN Input signals: Inputs I0…I10:

Byte 1			Byte 0								Operation Mode
I 10	I 9	I 8	I 7	I 6	I 5	I 4	I 3	I 2	I 1	I 0	
1	1	1	1	1	1	1	1	1	1	1	Wire check

MSTKN output signals: Outputs Q0…Q9:

Byte 1		Byte 0								State	LED's	Comment
Q 9	Q 8	Q 7	Q 6	Q 5	Q 4	Q 3	Q 2	Q 1	Q 0			
1	1	1	1	1	1	1	1	1	1	Checking	All yellow	Wire check is running

2.5 In/Out Profibus/Profinet

The signals are different depending on the mode that the socket selector device is currently running in.

2.5.1 Intelligent Mode

Control byte 0 (PLC → MSTKN) – single socket preselection:

7	6	5	4	3	2	1	0		
0..255(0xFF)								Signals to preselect socket number	8 signals for preselecting the desired socket (guided operation mode)

Control byte 1 (PLC OUT) – mode selection:

7	6	5	4	3	2	1	0		
		(reserved)			0	0	0	mode	Normal mode
					0	0	1		No operation
					0	1	X		(reserved)
					1	0	0		Calibration
					1	0	1		Reset Calibration
					1	1	X		(reserved)
	X							Main module LEDs	Red LED (NOK Status)
X									Green LED (OK Status)

PLC OUT

Status byte 0 (**PLC** IN) – displaying the empty socket:

7	6	5	4	3	2	1	0		
0..255 (0xFF)								socket	Indicates the socket that is currently empty

Status byte 1 (**PLC** IN) – displaying the current status:

7	6	5	4	3	2	1	0		
	(reserved)				0	0	0	Status	No tool removed (all sockets filled)
					0	0	1		Error: Tool removed (unauthorized)
					0	1	0		At least one tool is removed
					0	1	1		Error: Sensor-error in this tray
					1	0	0		Calibration was started (not finished yet!)
					1	0	1		(reserved)
					1	1	0		Calibration done (OK)
					1	1	1		Calibration done (Error)
X								Configuration	Configuration Error*

PLC IN

2.5.2 User mode

Control byte (**PLC** OUT) – single socket preselection

7	6	5	4	3	2	1	0		
		(r es er v			0	0	0	Mode	Normal mode

			0	0	1		No operation
			0	1	X		(reserved)
			1	0	0		Calibration
			1	0	1		Reset Calibration
			1	1	X		(reserved)
	X					Main module LED's	Red LED (NOK Status)
X							Green LED (OK Status)

PLC OUT

LED-display byte (**PLC** OUT) – Care: 1 Byte per single socket

7	6	5	4	3	2	1	0		
(reserved)						0	0	LED Green	Off
						0	1		On
						1	X		Flashing
				0	0			LED Yellow	Off
				0	1				On
				1	X				Flashing
		0	0					LED Red	Off
		0	1						On
		1	X						Flashing

Status byte (**PLC** IN) – displaying the empty socket

7	6	5	4	3	2	1	0		
	(reserved)				0	0	0	Status	No tool removed (all sockets filled)
					0	0	1		Error: Tool removed (unauthorized)
					0	1	0		At least one tool is removed
					0	1	1		Error: Sensor-error in this tray
					1	0	0		Calibration was started (not finished yet!)
					1	0	1		(reserved)
					1	1	0		Calibration done (OK)
					1	1	1		Calibration done (Error)
X								Configuration	Configuration Error*

Socket status byte (**PLC** IN) – displaying current socket status – Careful! 1 byte per socket

7	6	5	4	3	2	1	0		
(reserved)						0	0	Status	Tool removed
						0	1		Tool available (not removed)
						1	0		Error: Value too high (teach with tool)
						1	1		Error: Value too low (tool recognition)

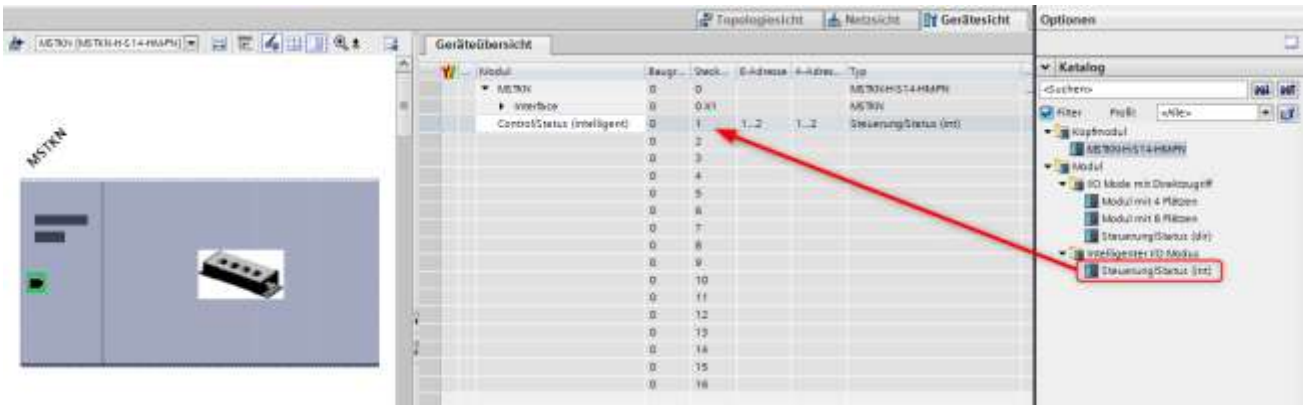
PLC IN

2.5.3 Sample Configuration Siemens PLC

2.5.3.1 Intelligent Mode

Intelligent mode uses 2 bytes I/O for the device. It uses a single Byte to preselect a socket and returns the used tool (empty socket) number. In general, the signals are meant to be compatible to the 24V I/O modules.

Here is a screenshot from the hardware config (TIA 15) – to configure, only one module needs to be plugged:



To map the I/O signals into the application, one can use a custom data type and map it to the IO addresses accordingly. Here are sample user data type definitions (UDTs) for inputs and outputs respectively:

UDT_MSTKN_Int_I

	Name	Datentyp	Defaultwert	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar
1	Selected	Byte	16#0	✓	✓	✓		Actually selected socket
2	state.0	Bool	false	✓	✓	✓		State bit 0
3	state.1	Bool	false	✓	✓	✓		State bit 1
4	state.2	Bool	false	✓	✓	✓		State bit 2
5	bit3	Bool	false	✓	✓	✓		
6	bit4	Bool	false	✓	✓	✓		
7	bit5	Bool	false	✓	✓	✓		
8	bit6	Bool	false	✓	✓	✓		
9	config_error	Bool	false	✓	✓	✓		Configuration error (mismatch)

UDT_MSTKN_Int_Q

	Name	Datentyp	Defaultwert	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar
1	Preselect	Byte	16#0	✓	✓	✓		Set socket number to blink to inform operator (0 = all...
2	mode.0	Bool	false	✓	✓	✓		Mode bit 0
3	mode.1	Bool	false	✓	✓	✓		Mode Bit 1
4	mode.2	Bool	false	✓	✓	✓		Mode Bit 2
5	bit3	Bool	false	✓	✓	✓		
6	bit4	Bool	false	✓	✓	✓		
7	bit5	Bool	false	✓	✓	✓		
8	led_rd	Bool	false	✓	✓	✓		LED red
9	led_gn	Bool	false	✓	✓	✓		LED green

Note that the State and Mode bits are actually a encoded number (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** for the details). To map these to the actual I/Os, set up a variable table as follows:

MSTKN

	Name	Datentyp	Adresse	Rema...	Erreic...	Schrei...	Sichtb...	Üb...	Kommentar
1	MSTKN_Q	"UDT_MSTKN_Int_Q"	%Q1.0		✓	✓	✓		MSTKN socket tray control signals (intelligent mode)
2	MSTKN_I	"UDT_MSTKN_Int_I"	%I1.0		✓	✓	✓		MSTKN socket tray status signals (intelligent mode)

To monitor the inputs and outputs, a watch table can be created, e.g. as follows:

	i	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert	⚡	...	Variablen-Kommentar
1		// Inputs							
2		"MSTKN_I".Selected	%IB1	Hex	16#03				Actually selected socket
3		"MSTKN_I"."state.0"	%I2.0	BOOL	FALSE				State bit 0
4		"MSTKN_I"."state.1"	%I2.1	BOOL	TRUE				State bit 1
5		"MSTKN_I"."state.2"	%I2.2	BOOL	FALSE				State bit 2
6		"MSTKN_I".config_error	%I2.7	BOOL	FALSE				Configuration error (mismatch)
7									
8		// Outputs							
9		"MSTKN_Q".Preselect	%QB1	Hex	16#00				Set socket number to blink to inform oper...
10		"MSTKN_Q"."mode.0"	%Q2.0	BOOL	FALSE				Mode Bit 0
11		"MSTKN_Q"."mode.1"	%Q2.1	BOOL	FALSE				Mode Bit 1
12		"MSTKN_Q"."mode.2"	%Q2.2	BOOL	FALSE				Mode Bit 2
13		"MSTKN_Q".led_gn	%Q2.7	BOOL	FALSE				LED green
14		"MSTKN_Q".led_rd	%Q2.6	BOOL	FALSE				LED red
15			<Hinzufügen>						

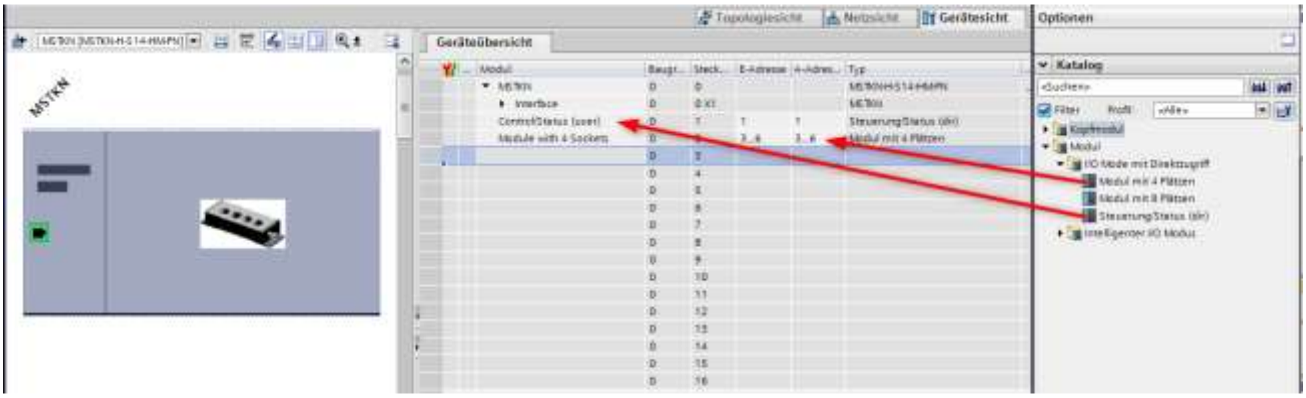
To e.g. run a calibration cycle, force the „mode“ value to 4 – i.e. mode bit 2 set, mode bits 0,1 reset (warning: make sure to empty all trays/remove all tools before doing so!):

	i	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert	⚡	...	Variablen-Kommentar
1	// Inputs								
2		"MSTKN_I".Selected	%IB1	Hex	16#04		☐		Actually selected socket
3		"MSTKN_I"."state.0"	%I2.0	BOOL	TRUE		☐		State bit 0
4		"MSTKN_I"."state.1"	%I2.1	BOOL	TRUE		☐		State bit 1
5		"MSTKN_I"."state.2"	%I2.2	BOOL	FALSE		☐		State bit 2
6		"MSTKN_I".config_error	%I2.7	BOOL	FALSE		☐		Configuration error (mismatch)
7							☐		
8	// Outputs								
9		"MSTKN_Q".Preselect	%QB1	Hex	16#00		☐		Set socket number to blink to inform oper...
10		"MSTKN_Q"."mode.0"	%Q2.0	BOOL	FALSE		☐		Mode bit 0
11		"MSTKN_Q"."mode.1"	%Q2.1	BOOL	FALSE		☐		Mode Bit 1
12		"MSTKN_Q"."mode.2"	%Q2.2	BOOL	TRUE	TRUE	☑	⚠	Mode Bit 2
13		"MSTKN_Q".led_gn	%Q2.7	BOOL	FALSE		☐		LED green
14		"MSTKN_Q".led_rd	%Q2.6	BOOL	FALSE		☐		LED red
15			<Hinzufügen>				☐		

This will then report the number of actually detected sockets in „MSTKN_I".Selected. To go back to normal operation, de-assert mode.2.

2.5.3.2 User mode

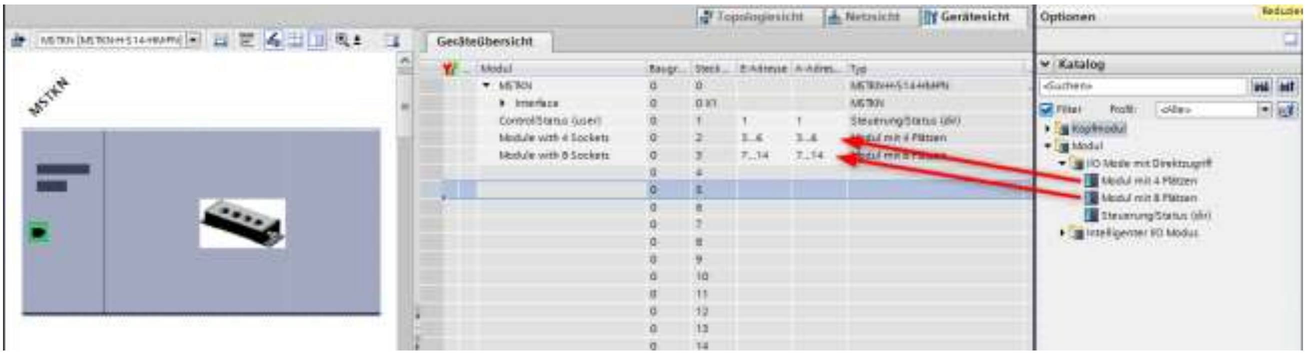
User mode uses 1 byte I/O for the overall device control/status plus a single Byte I/O for each socket. With this, the PLC has full control over the LED's and can read the sensor information from each socket individually. The tray only acts as a kind of remote I/O module, the PLC has to take care of the logic. This allows implementing custom behaviors or LED color schemes. Here is a screenshot from the hardware config (TIA 15) – to configure, the 1-Byte Status/Control module plus a module per device (in this case a 4-socket block with 4 Bytes I/O needs to be plugged):



The device I/O byte (status/control) works identically to the "intelligent mode" status/control and uses the same bit mapping (see above for a sample UDT definition). Compared to the "intelligent mode" it is however located at I/O address offset 0 (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** for detailed information about the signal layout).

The socket I/O byte is used to control the three LEDs for each socket and returns information about the sensor state. Please also see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** for details.

For setups using a main module and additional extension modules connected over the extension bus, additional modules must be plugged in the PLCs hardware configuration. The module plugged must follow the physical setup, e. g. one 4-socket main module combined with a 8-socket extension module would look like the following:



2.6 In/Out Ethernet/WLan

2.6.1 Modbus-Communication

2.6.1.1 Overview

Ethernet and Wifi socket trays use Modbus/UDP (preferred) or Modbus/TCP for their communication. The socket tray implements a Modbus device (TCP server) and replies to commands received from the controller.

Modbus generally works in a request/response fashion, i.e. the device (MSTKN socket tray) never spontaneously sends data, but only answers commands received from the controller. To check, if a "connection" with the controller is established and active, the MSTKN tray expects to cyclically receive commands from the controller and measures the time since the last received command. If the timeout (500ms) is exceeded, then the tray detects a disconnected controller and shows an error pattern ("lost connection").

The MSTKN tray uses the following TCP/UDP ports:

- 502/UDP and 502/TCP: Modbus standard protocol
- 5003/UDP: U-Blox legacy Modbus protocol (non-standard Modbus frame)

Notes:

- UDP-responses will be sent to the same UDP port as where the command originated. On the controller side therefore is only a single UDP socket needed.
- To use Modbus/UDP on Windows, the firewall must enable the application, or the client must use a fixed UDP port (firewall enabled)!
- For optimum performance, single-cycle I/O using FC23 should be used. In this case, all 33 I/O-words (66 bytes) will be send and read in a single read/write Modbus transaction. A fixed size transaction (33 word read/write FC23) can be used for intelligent and user operation modes (select mode in register 40001).

WARNING: Depending on the firmware version, accessing partial register ranges is not possible. For maximum compatibility, always read/write all 33 registers in the FC23 transaction (read 33 words starting at address #0 and write 33 registers starting at address #0)!

WARNING: Depending on the firmware version, Modbus/TCP might not be supported! For maximum compatibility, use Modbus/TCP!

WARNING: When using TCP then operating system timeouts and automatic retransmits on the network layer might make detecting the actual communication state difficult. Use of UDP is recommended to better control communication timings (especially when using Wi-Fi)!

WARNING: Multi-master connections are not allowed. A new connection is only accepted, after the existing connection is disconnected (e. g. disconnect timeout)!

2.6.1.2 Supported Modbus function codes

Currently, the socket trays only support FC23.

Function Code (FC)	Name	Max. Message Size
23 (0x17)	Read and Write Registers	246 Bytes

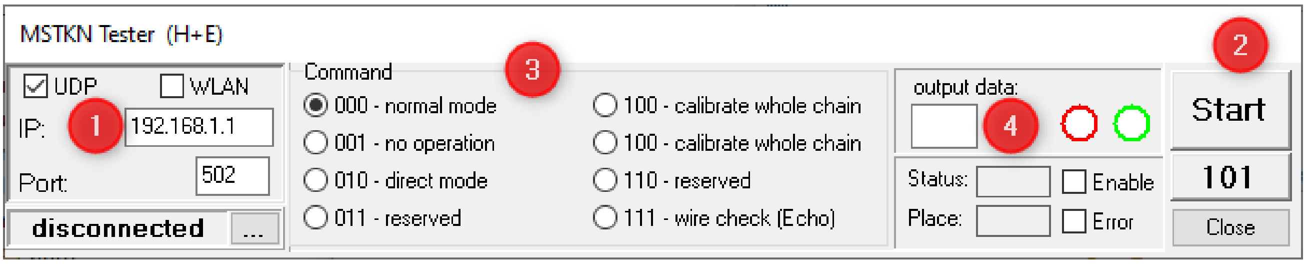
2.6.1.3 Modbus register mapping

The socket tray Modbus implementation only provides register access (registers 4xxxx) – controlled over FC23 (address >= 0 ⬄ address = register - 40001). The registers are mapped differently when reading and writing as follows:

Register	R/W	Name	Description
40001	R	Mode/Sockets	Mirrors actual mode, reports actual number of sockets
40002	R	Reserved	Always reads 0
40003	R	Select/Status	EA24 compatible status / selection (intelligent mode)
40004-40033	R	Sensor status	1 Byte per socket (user mode only)
40001	W	Mode/Sockets	Set mode (user/intelligent) and expected number of sockets
40002	W	Reserved	Always set to 0
40003	W	Preselect/Command	EA24 compatible commands / preselection (see below)
40004-40033	W	LED colors	1 Byte per socket LED colors (user mode only)

2.6.1.4 Test

The „MSTKN Tester“ (MSTKN-Modbus-Master.exe) application uses Modbus/UDP to control the socket trays and can be used to test communications:



The following functions are available:

- ❶ Setup communication parameters (UDP für Ethernet/WLan)
- ❷ Start cyclic communications
- ❸ Commands (for intelligent operation mode)
- ❹ Socket preselection and current selection state (response from socket tray)

2.6.2 Intelligent Mode

2.6.2.1 Modbus-Controller → MSTKN

<table>
<tr><th>Address</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td rowspan="2">Data</td><td rowspan="2">Mode (0 = „intelligent“)</td><td rowspan="2">Expected #Places</td><td rowspan="2">reserved (=0)</td><td rowspan="2">reserved (=0)</td><td colspan="2">Compatible to EA24</td></tr>
<tr><td>Preselection</td><td>Action</td></tr>
</table>

Byte 0: Operation mode selection („intelligent“ = 0)

Byte 1: Expected number of sockets. If non-zero, then the MSTKN tray compares the actual number of detected/configured sockets to the given number. If set to zero, no check is done.

Byte 4: Preselection

7	6	5	4	3	2	1	0		
0									No preselection
1..60								Socket	Number of preselected socket
>60									reserved

Byte 5: Control byte (action)

7	6	5	4	3	2	1	0		
		(reserved)			0	0	0	Operation mode	Normal operation
					0	0	1		No selection
					0	1	X		(reserved)
					1	0	0		Calibration
					1	0	1		Calibration (with reset)
					1	1	X		(reserved)
	X							Main module	Red LED (NOK status)
X								LEDs	Green LED (OK status)

2.6.2.2 MSTKN → Modbus-Controller

Address	0	1	2	3	4	5
Data	Mode (0 = „intelligent“)	Detected #Slots	reserved (=0)	reserved (=0)	Compatible to EA24	
					Selection	State

Byte 0: Mirrors mode byte (from input)

Byte 1: Detected slots

7	6	5	4	3	2	1	0		
		0..60						Number of slots	Number of configured sockets/slots
	0							Reserved	
X								Config-Mismatch error	= 0: Config ok or check not active = 1: Config mismatch error

Byte 4: Selection

7	6	5	4	3	2	1	0		
0	0	0..60						Socket	Currently selected socket

Byte 5: State

7	6	5	4	3	2	1	0		
	(reserved)				0	0	0	State	No selection
					0	0	1		Error: tool selected (not allowed)
					0	1	0		At least one tool selected
					0	1	1		Error: Sensor error
					1	0	0		Calibration running
					1	0	1		(reserved)
					1	1	0		Calibration done (OK)

		1	1	1		Calibration done (Error)

2.6.3 User mode

2.6.3.1 Modbus-Controller → MSTKN

Address	0	1	2	3	4	5
Data	Mode (1=user)	Expected #Slots	reserved (=0)	reserved (=0)	reserved (=0)	Action
Address	6	7	…	…	64	65
Data	Socket #2*	Socket #1*	…	…	Socket #60*	Socket #59*

***IMPORTANT**: Note the byte order!

Byte 0: Operation mode selection („user mode“ = 1)

Byte 1: Expected number of sockets. If non-zero, then the MSTKN tray compares the actual number of detected/configured sockets to the given number. If set to zero, no check is done.

Byte 5: Control byte (action)

7	6	5	4	3	2	1	0		
		(reserved)			0	0	0	Operation mode	Normal operation
					0	0	1		No selection
					0	1	X		(reserved)
					1	0	0		Calibration
					1	0	1		Calibration (reset)
					1	1	X		(reserved)
	X							Main module - LEDs	Red LED (NOK State)
X									Green LED (OK State)

Bytes 6 – 65: 1 Byte per Socket to control the LEDs:

7	6	5	4	3	2	1	0		
(reserved)						0	0	LED Green	Off
						0	1		On
						1	X		Blink
				0	0			LED Yellow	Off
				0	1				On
				1	X				Blink
		0	0					LED Red	Off
		0	1						On
		1	X						Blink

2.6.3.2 MSTKN → Modbus-Controller

Address	0	1	2	3	4	5
Data	Mode (1=user)	Detected #Slots	reserved (=0)	reserved (=0)	reserved (=0)	State
Address	6	7	…	…	64	65
Data	Socket #2*	Socket #1*	…	…	Socket #60*	Socket #59*

***IMPORTANT**: Note the byte order!

Byte 0: Mirrors mode byte (from input)

Byte 1: Detected slots

7	6	5	4	3	2	1	0		
		0..60						Number of slots	Number of configured sockets/slots
	0							Reserved	
X								Config-Mismatch error	= 0: Config ok or check not active = 1: Config mismatch error

Byte 5: State

7	6	5	4	3	2	1	0		
	(reserved)				0	0	0	State	No selection
					0	0	1		Error: tool selected (not allowed)
					0	1	0		At least one tool selected
					0	1	1		Error: Sensor error
					1	0	0		Calibration running
					1	0	1		(reserved)
					1	1	0		Calibration done (OK)
					1	1	1		Calibration done (Error)

Bytes 6 – 65: 1 Byte per socket with state:

7	6	5	4	3	2	1	0		
(reserved)						0	0	State	No selection
						0	1		Tool selected
						1	0		Error: Sensor measurement too low
						1	1		Error: Sensor measurement too high

2.6.4 Python sample code

The following code shows how to access the socket trays using Python (minimal code, error checks omitted).

```
## NOTE: 'use pip install pymodbus' to install the python modbus library
from pymodbus.client.sync import ModbusUdpClient
import binascii
import time

client = ModbusUdpClient('10.10.2.64', 502)
arguments = {
    'read_address': 0, 'read_count': 33,
    'write_address': 0, 'write_registers': [
        0x0104,         # hi = 1 (direct mode), lo = 4 (number of sockets)
        0,              # reserved
        0x0000,         # hi = 0 (socket preselect intelligent),
                        # lo = 0 (mode select (intelligent mode only))
        # Direct mode only; one byte per socket, two bits per LED,
        # binary 00RRYYGG (RR=red, YY=yelow, GG=green,
        # bits per LED: 00=off, 01=steady on, 10=blinking, 11=blinking)
        # Sample:
        # 01            Socket #1: LED off
        #   02          Socket #2: LED green (steady)
        #         03    Socket #3: LED green blinking
        #           04  Socket #4: LED yellow
        0x0001, 0x0204, 0x2010, 0x4030,
        0, 0, 0, 0,
        0, 0, 0, 0,
        0, 0, 0, 0,
        0, 0, 0, 0,
        0, 0, 0, 0,
        0, 0, 0, 0,
```

```
        0, 0
    ]
}
while (True):
    result = client.readwrite_registers(**arguments)
    print(hex(result.registers[3]), hex(result.registers[4]),
hex(result.registers[5]))
    time.sleep(1.0)

client.close()
```

2.7 In/Out EtherCAT

2.7.1 Overview

The structure and mapping of the I/O-signals for the EtherCAT-interface is identical to what is described in chapter 2.6 for Ethernet/OpenModbus.

Overall EtherCAT exchanges 66 bytes inputs and 66 bytes outputs cyclically regardless of the operation mode "intelligent" or "user". Switching between the modes is exactly as with Ethernet/OpenModbus by setting the first byte.

2.7.2 MSTKN → PLC

The 66 bytes of the input data are structured as follows (for details, see 2.6.2 and **Fehler! Verweisquelle konnte nicht gefunden werden.**):

Byte-Address	Name (ESI-file)	Description
0	OperationMode	Current socket tray operation mode (0=intelligent, 1=usermode)
1	Sockets_Act	Actual number of sockets as configured in the trays internal config
2-3	Reserved_01 Reserved_02	
4	Selected	Current selected socket (if any, else = 0, intelligent mode only)
5	Status	Bitmask as described in chapter 2.6.3.2
6..65	Socket_01 Socket_02 … Socket_60	Socket state of socket #1 (as described in chapter 2.6.3.2) Socket #2 … Socket #60

2.7.3 MSTKN → PLC

The 66 bytes of the output data are structured as follows (for details, see 2.6.2 and **Fehler! Verweisquelle konnte nicht gefunden werden.**):

Byte-Address	Name (ESI-file)	Description
0	OperationMode	Current socket tray operation mode (0=intelligent, 1=usermode)
1	Sockets_Exp	Expected number of sockets
2-3	Reserved_01 Reserved_02	
4	Preselect	To be selected socket (if any, else = 0, intelligent mode only)
5	Action	Bitmask as described in chapter 2.6.3.2

6..65	LEDs_01 LEDs_02 … LEDs_60	LED state for socket #1 (as described in 2.6.3.2, only in usermode) LED state for socket #2 … LED state for socket #60

2.7.4 Configuration with CtrlX

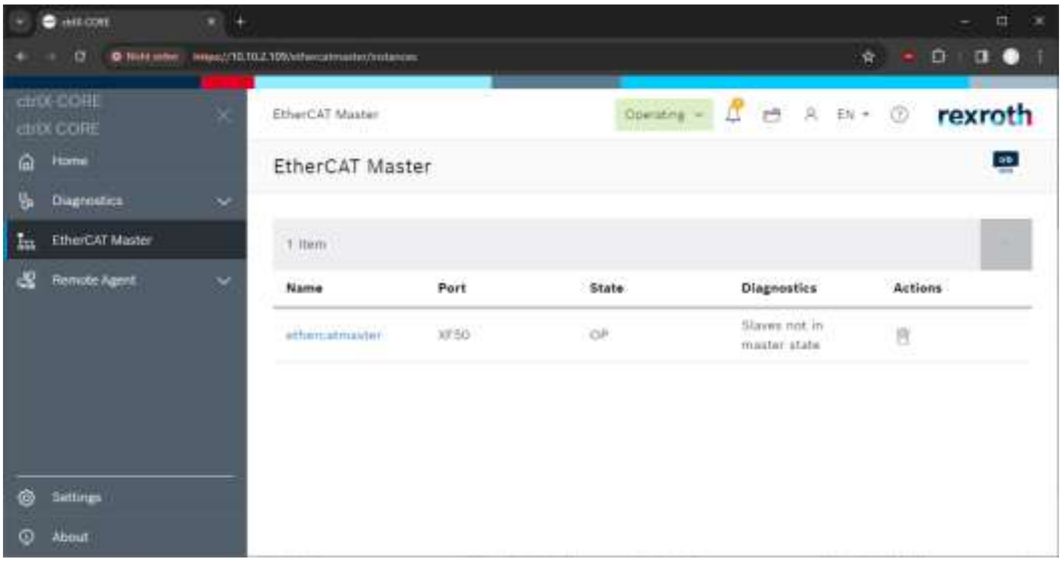
The following chapters shows the steps needed to configure the Ctrlx-Controller to connect to the socket tray over EtherCAT.

2.7.4.1 Run the required software

To configure the CtrlX-controllera matching version of the „ctrlX I/O Engineering“ software is required. The PC running the software must be connected to the interface XF10 of the CtrlX-device. The socket tray is connected to the XF50 interface of the CtrlX-device (EtherCat-Bus).

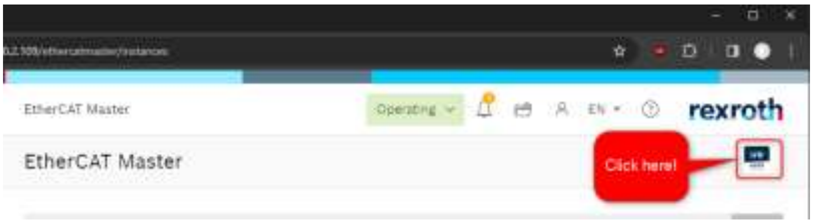
Enter the IP-Address of the XF10-interface into the PCs webbrowser to open the CtrlX device start page. (if you don't know the CtrlX IP address, then try the default 192.168.1.1 or install and run CtrlX WORKS to scan the network for CtrlX devices).

After logging on (e. g. https://192.168.1.1), switch to the „EtherCAT Master“ web application:

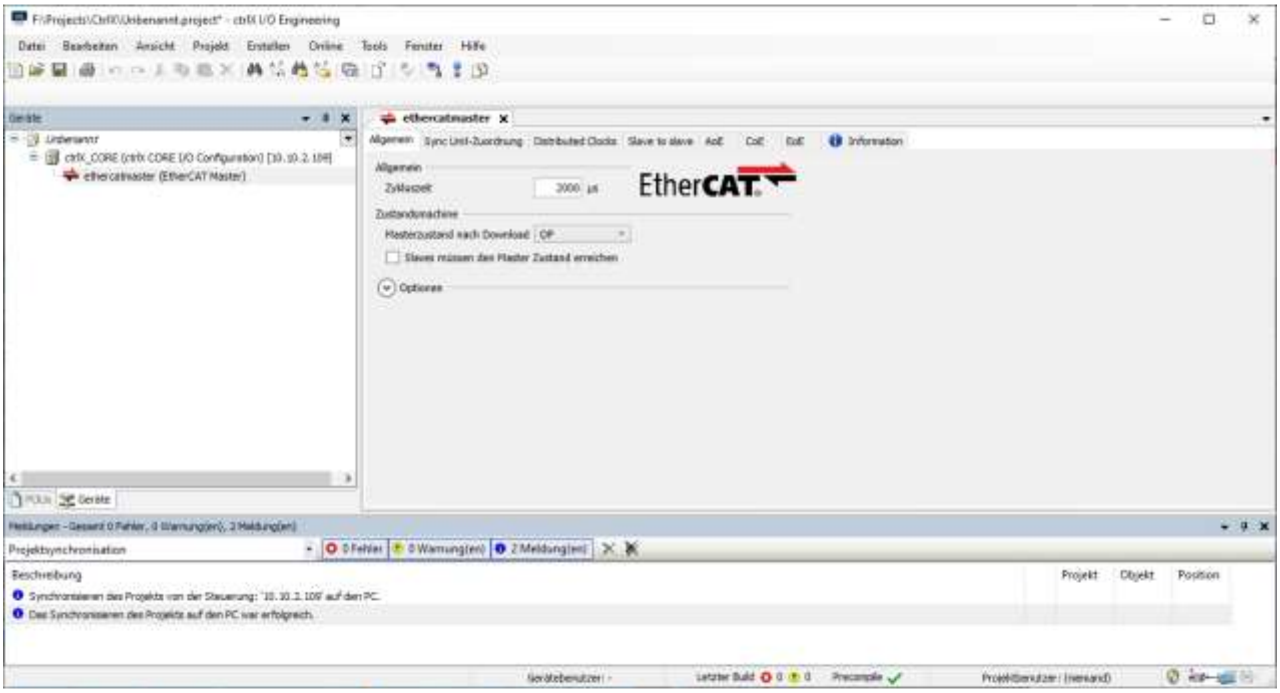


If you can't find this in the left-hand menu, then the EtherCAT-Master app is not installed. Check the CtrlX manual on how to connect to the CtrlX app store and how to install the EtherCAT app (**WARNING**: you will need a license to run the EtherCAT master app – registering an account in the CtrlX store will take a few days, also ordering through the store is not possible directly – you will be redirected to your local reseller, which will send you a quote, based on which you can then order – this will overall also take a few days, so prepare ahead accordingly)!

As you cannot configure the EtherCAT master through the apps web interface (this is for diagnostics only), you will have to start the I/O-Engineering. You can start the application by clicking the small button on the top right (showing the „I/O“ logo):

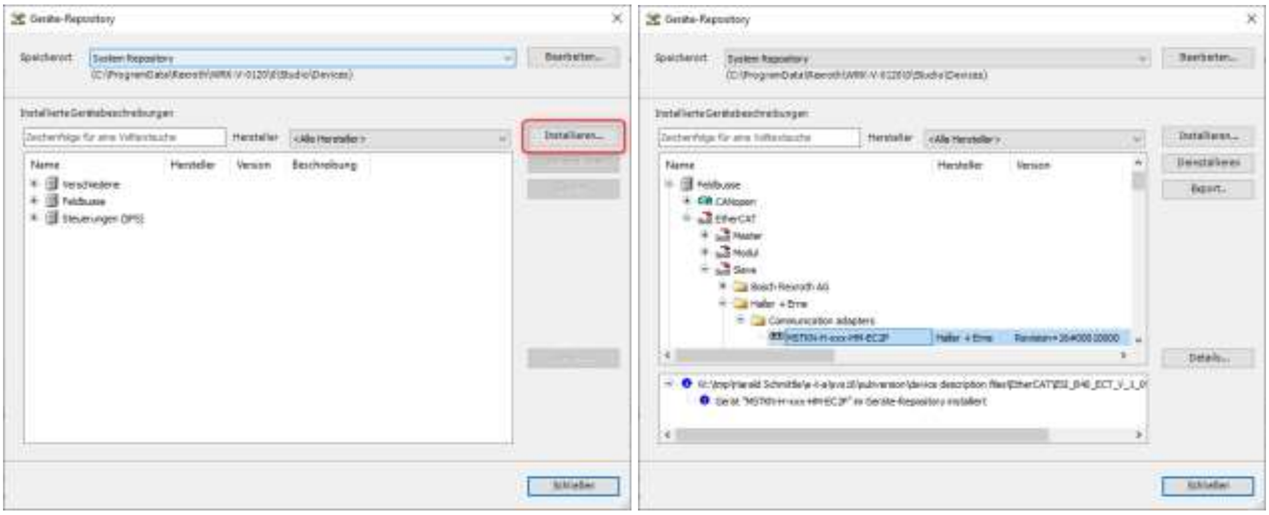


This will then open the Windows application „CtrlX I/O Engineering" (if already installed), usually with an empty project:



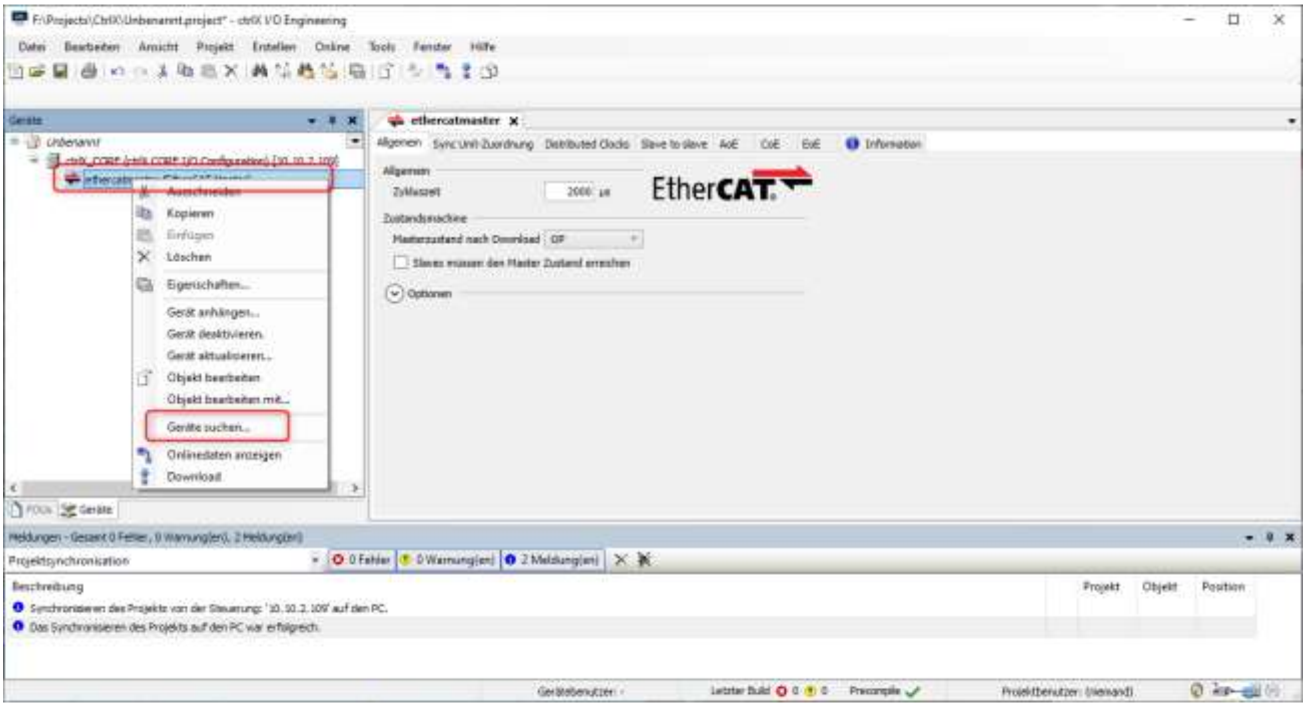
2.7.4.2 Installing the ESI-file

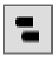
Before you can add the MSTKN EtherCAT device to the configuration, you will have to add the MSTKN device ESI file to the configuration database. The configuration database can be opened from the main menu from Tools → device repository. To add the ESI-file of the socket tray, click the Button "Install" and select the file ESI_B40_ECT_V_1_09-MSTKN-R0002.xml:

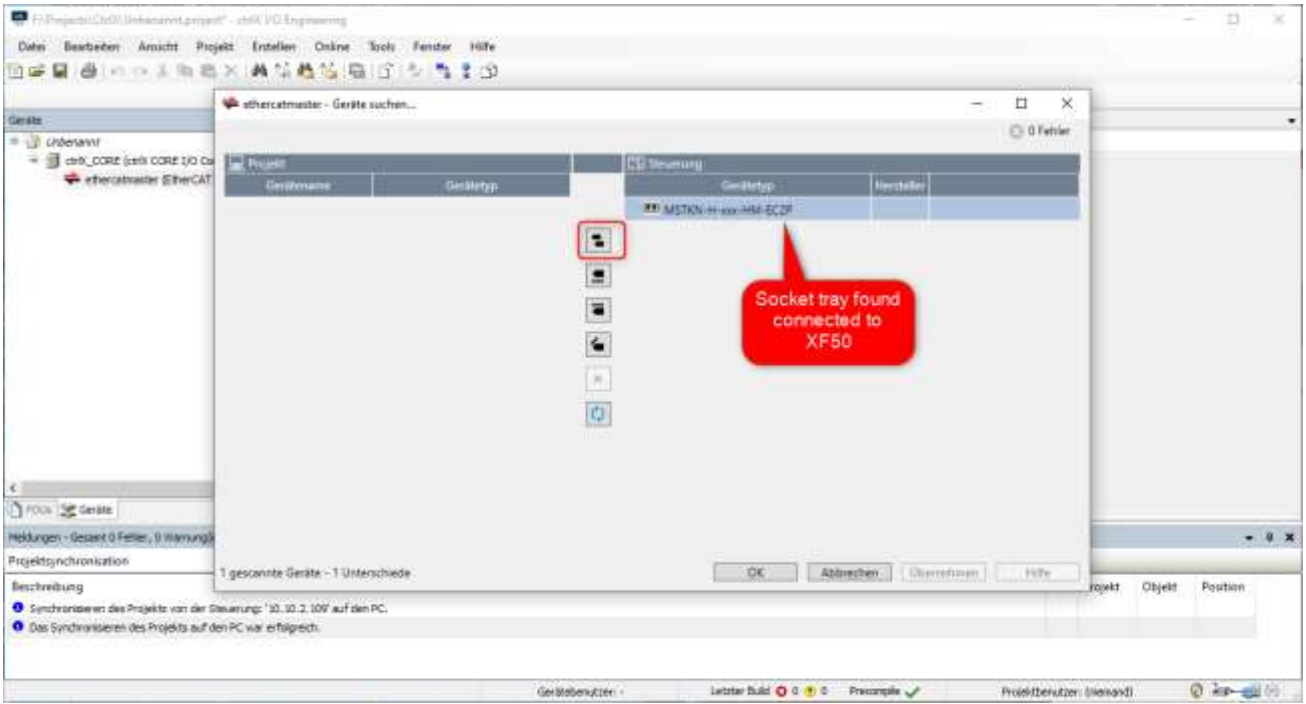


2.7.4.3 Add the socket tray to the I/O-configuration

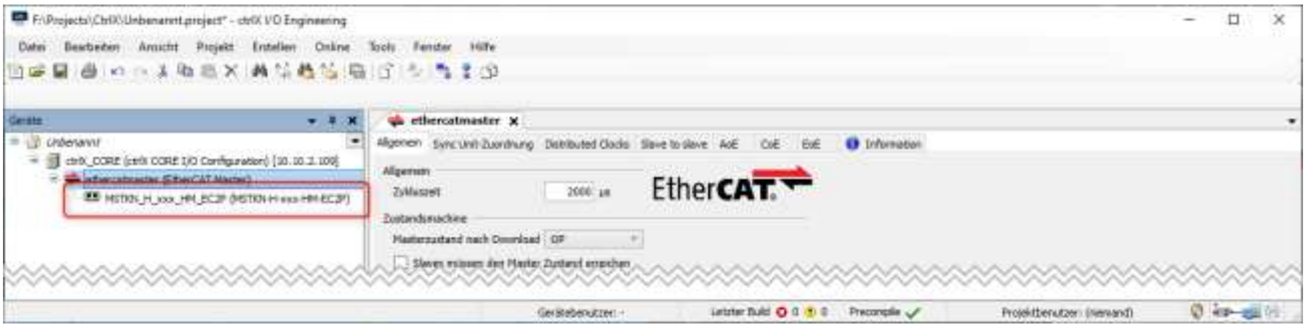
The quickest option to add the socket tray to the configuration is to search for the (at the XF50 connected) socket tray device. To do so, right-click the master node to open the context menu and select the Entry „Search for devices":



This will find the socket tray, which can then be added to the project by clicking the  button:

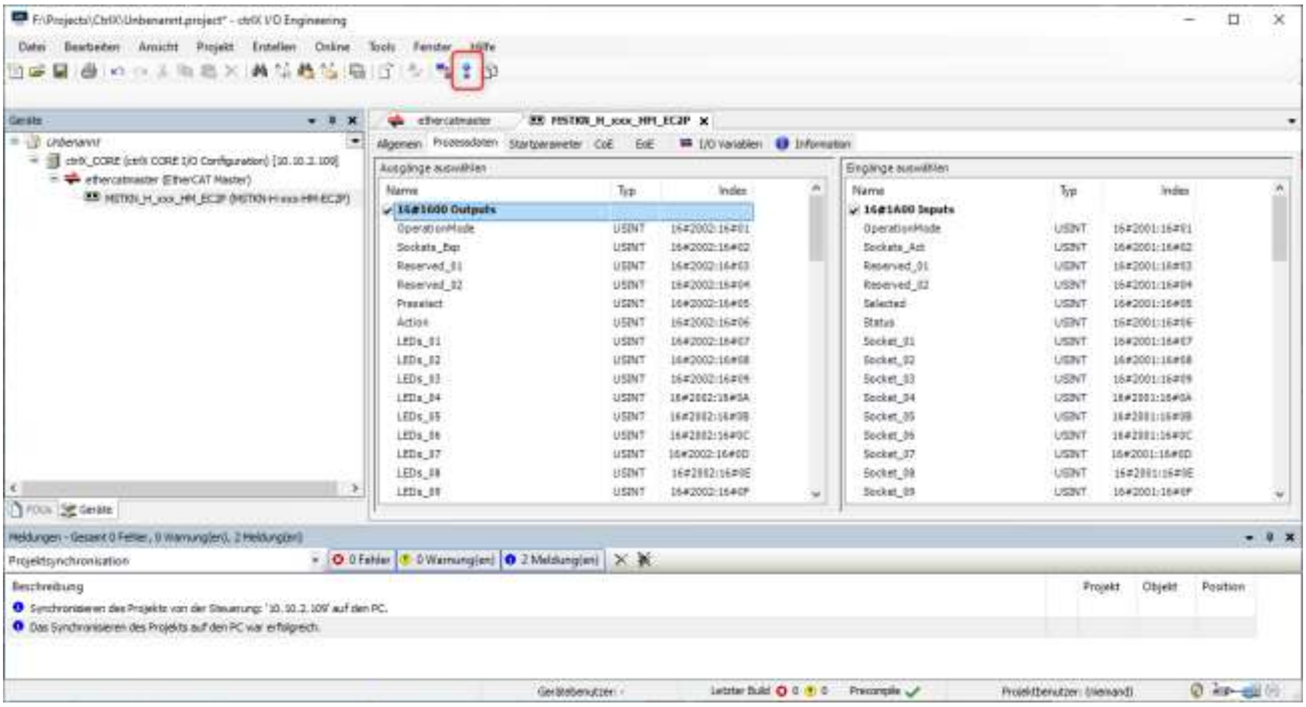


After clicking OK, the device is added to the project – the name can then be changed eventually:



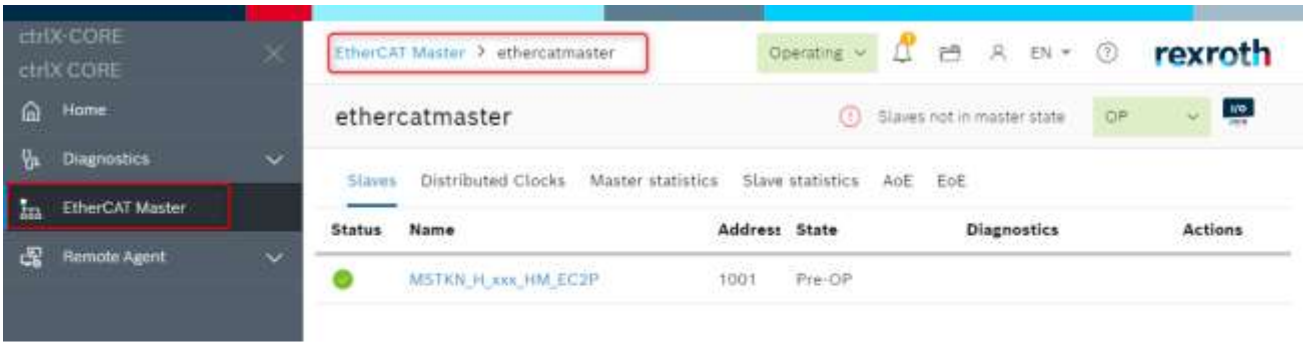
2.7.4.4 Load the I/O-configuration into CtrlX

After the configuration is set up, this must be uploaded into the CtrlX-device. To do so, click the “download”-button in the main toolbar:

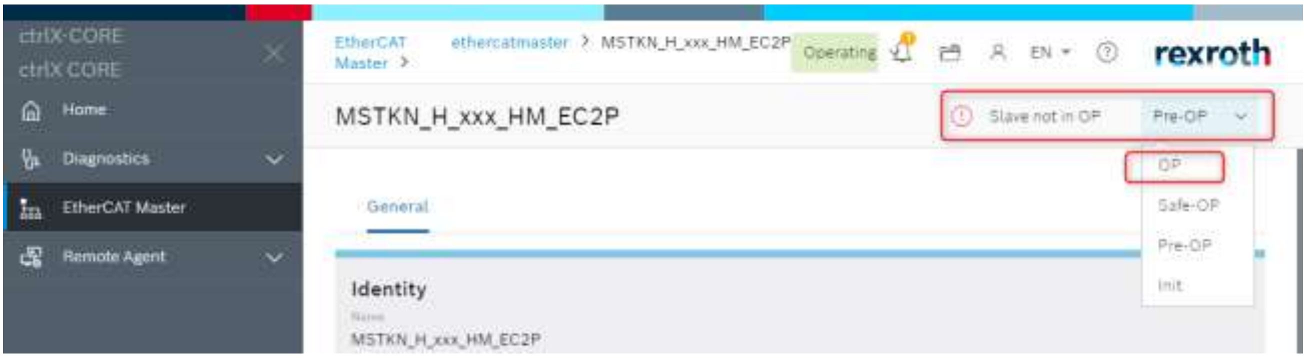


2.7.4.5 Start cyclic EtherCAT operation

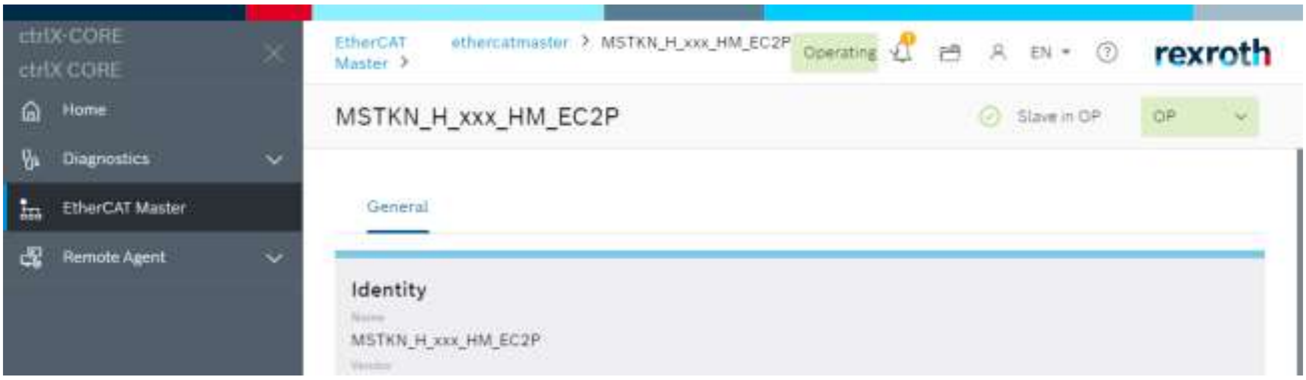
By default, the CtrlX device topps the EtherCAT master whenever the bus configuration is changed. To start the cyclic bus operation, either the EtherCAT app from the device website or the I/O-Engineering application can be used. In the device EtherCAT web application “EtherCAT Master“, click the interface name, which will show the configured bus devices and their state as follows:



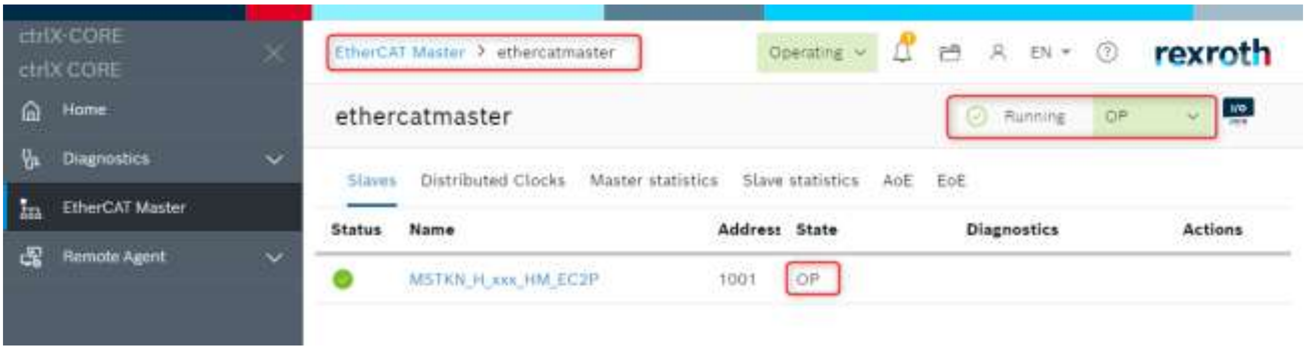
Clicking the device name will take you to the detail page of the device:



Now select "OP" from the dropdown to start the cyclic bus operation. The device state should then also change to "Slave in OP":



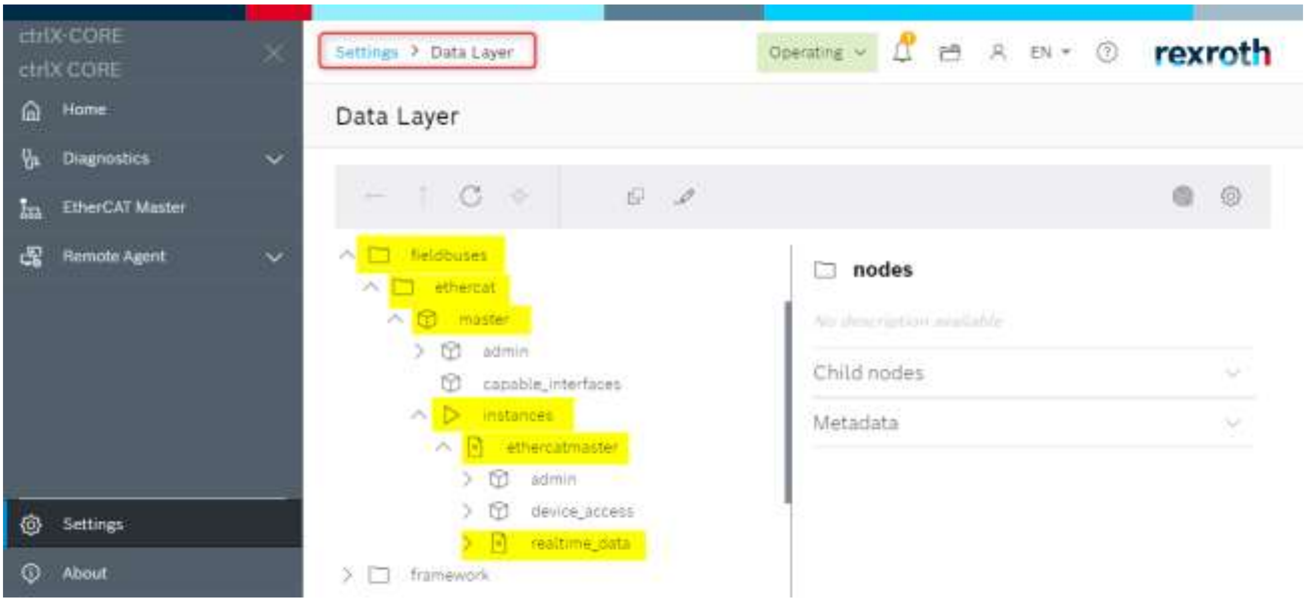
Going back one level will also show the masters state as "OP" and green (running):



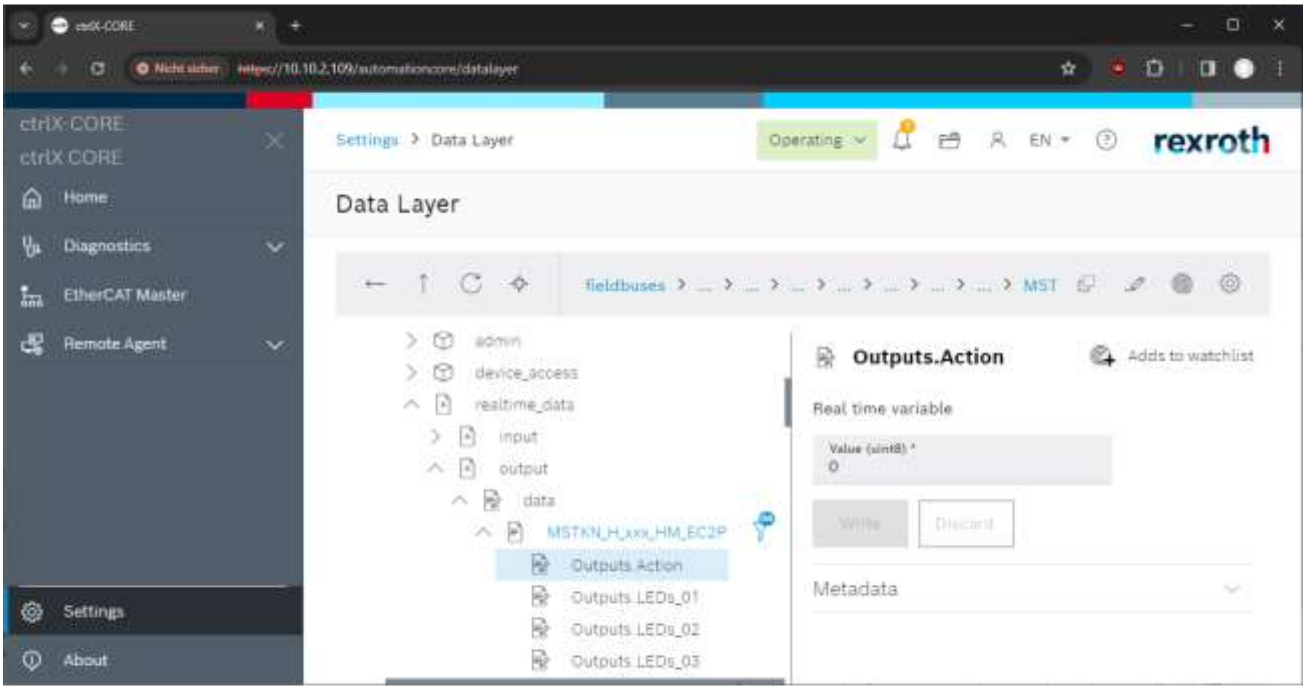
2.7.4.6 I/O-Diagnostics and Test (DataLayer)

Theoretically should the CtrlX operating system map all cyclic data into the “Realtime“-tree of the datalayer. This allows looking at the current state of the I/Os and also change these manually (“forcing” - for tests).

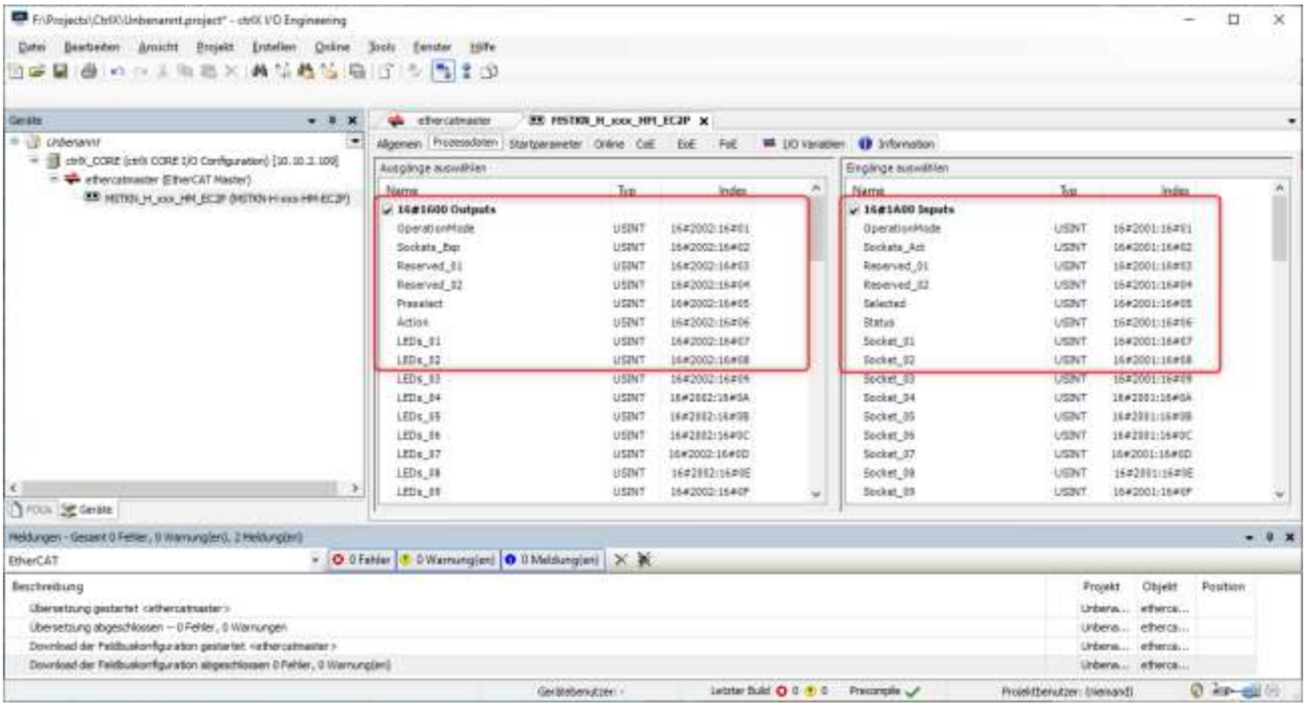
The web application for the datalayer access can be found through Settings → Datalayer, the actual data values are in the tree ethercat → master → instances → ethercatmaster → realtime:



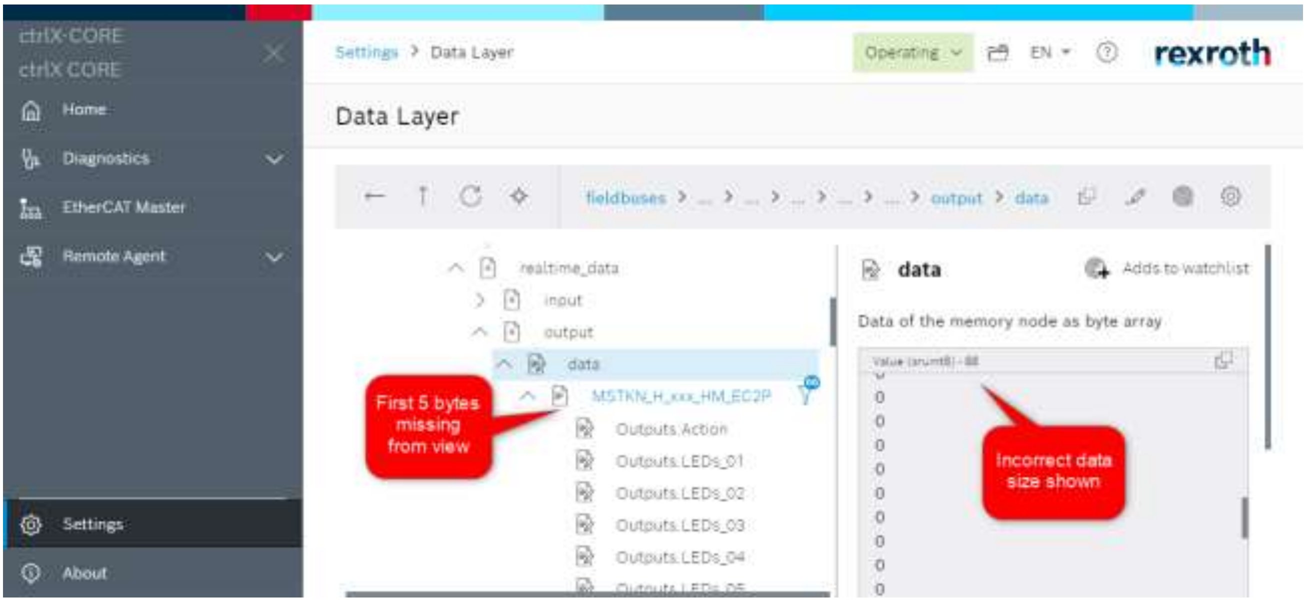
The data is listed in the nodes input/output → data → <device name>:



IMPORTANT: The current firmware version has some bugs with mapping EtherCAT data correctly to the datalayer. This leads to missing data items. For the MSTKN ESI file, this drops the first 5 variables of the „output“-Range of the tray – even though this is correctly available through the I/O-Engineering:



Even though the I/O-Engineering shows the data item correctly, the CtrlX web interface misses these – the view starts with „Action“ instead of „OperationMode“:



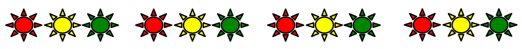
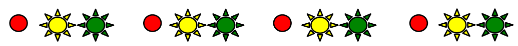
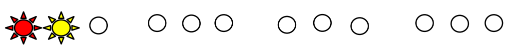
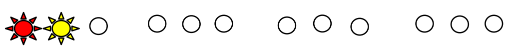
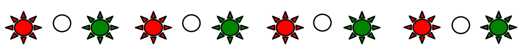
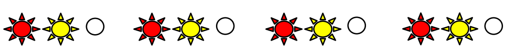
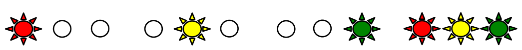
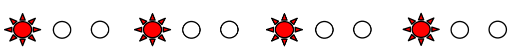
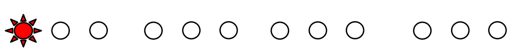
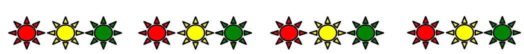
Also the size of the I/O-range is shown incorrectly and the last elements are also missing (Outputs.LEDs_50-60 are missing) – maybe the web-view is limited to 50 elements in this firmware version.

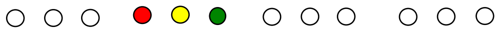
3 Optical Indicators (for all types)

There are two simple possibilities to diagnose and repair the socket selector devices. Each socket's LED's show the errors of that socket and there are also two LED's next to the plugs at the side of the device. The following chart shows all possible errors as well as a short instruction for troubleshooting.

3.1 Sensor Indicators

The following table shows all possible errors as well as corresponding information from the sensor displays.

Error message	Cause of the error	Troubleshooting
 All LEDs are flashing	Internal MSTKN communication disrupted	Send in the device
 The red LED is permanently on, yellow and green LED(s) flashes one for the other	There are missing sockets (single sensor PCBs), at single sensor sockets the rotary switch at the first sensor PCB is for number of single sensors in the socket	Please check the rotary switch at the single sensor PCBs – all single sensors in a socket must differ (the first sensor PCB has the number ‚1')
 At one or more sockets the red LED flashes permanently and the yellow LED flashes once	Single sensor receives no BROADCAST-Telegrams from the MASTER, at single sensors is the first sensor the MASTER	- At single sensors check the internal cables - Send in the device
 At one or more sockets the red LED flashes permanently and the yellow LED flashes twice	Socket doesn't receive no telegram from its neighbour	Please check the rotary switch at the single sensor PCBs, the numbering of the single sensors must be consecutively
 All red and green LEDs are flashing	Sensor element only receives messages which are not meant for its own Modbus ID	- Slave address is wrong - Calibrate the device
 All red and yellow LEDs are flashing	- Sensor element is receiving Modbus-Telegrams with wrong baud-rate - Sensor element receives telegrams with CRC Errors	- Check cable length - Check termination resistor - Send in the device
	Wrong answer in direct mode (24V I/O) Connection to fieldbus master lost (Profibus, Modbus UDP etc.)	- Check fieldbus master - Check cable connections - Check fieldbus address/IP - Mirror the Input from the controller to the Output (24V I/O) - deactivate direct mode (24V I/O)
 All red LEDs are flashing	- Sensor element not calibrated Master sends a message to the slave, but doesn't receive and answer	- Check if all extension modules are connected properly - Socket selector device calibrated
 One or several red LEDs are flashing	Sensor error, Value is higher than calibrated value	- Calibrate the device - Send in the device
 One or several red LEDs are flashing	Wire test active	- Deactivate wire test

Error message	Cause of the error	Troubleshooting
 Lauflicht (mit drei LEDs) von links nach rechts	Configuration defect	- Application has to be programmed new, with the Bootloader application on PC (with ‚clear config page‘ on) - If not possible, the Bootloader at the sensor PCB has to be programmed - send in the device
 All LEDs on one socket are permanently on	The internal master has an address conflict. There is a slave address which exists twice. The three LEDs of the slots show, which address it is.	- Check slave address - Designate slave addresses in ascending order and without gaps

3.2 Additional LED Signals

3.2.1 Ok/NOK-LEDs

The main modules of the MSTKN socket tray family come with two extra LEDs (green and red) which can be programmed as the user whishes (for example to show “tightening process ok” or “tightening process not ok”)

Signal for 24V I/O (DSUB25):

MSTKN input signals			
Pin	**Signal**	**Description**	**IM24V**
PIN 24	LEDR	red LED (24V=on)	A11
PIN 25	LEDG	green LED (24V=on)	A13

Ansteuerung bei Feldbussystemen:

MSTKN input signals									
Byte 1									
7	6	5	4	3	2	1	0		
	X							Main module LED’s	Red LED (NOK Status)
X									Green LED (OK Status)

4 PC-based Diagnose and Configuration

4.1 Overview

As described in chapter 1.4.6, the internal Bus-interface can also be used to configure and diagnose the socket tray system. An adapter cable from USB to M12 is used to connect the socket tray(s) and the PC. It is plugged in at the end of the Bus instead of a termination resistor.

Attention: A main module must be connected to the Bus to guarantee an error-free operation! Using a USB ⇔ M12 adapter cable with just an extension module is not permitted.

The diagnosing software then uses the bus to communicate with the different parts of the socket tray system and offers a variety of functions to set parameters and diagnose.

The USB ⇔ M12 adapter cable should be connected while the socket trays are volt-free. The power supply can be reconnected after the adapter cable was plugged in.

4.2 Software Installation

To use the parameterization/diagnosing software, a virtual COM-Port USB-Driver for FTDI-Serial adapters has to be installed. (see http://www.ftdichip.com/Drivers/VCP.htm).

The actual parameterization/diagnosing software does not require an installation, the files only have to be copied into a directory on a local hard drive and the application can be started.

4.3 Setting the Parameters for Tool Recognition

To activate and set the parameters for the tool recognition, the application STKN-SM-Config.exe is used.

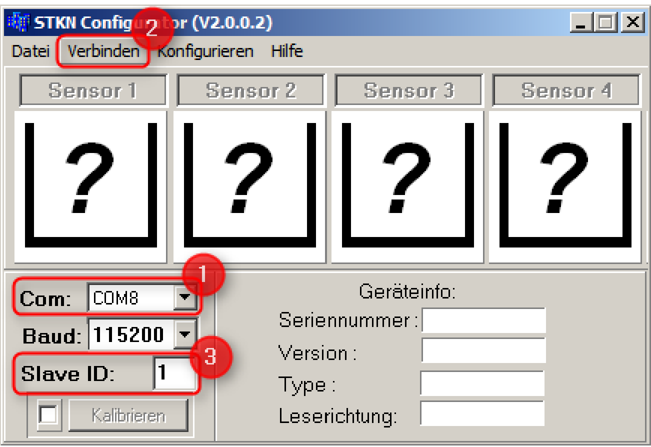
The basic steps are:

1. Establish the connection with the socket tray by using the USB ⇔ M12-cable (as described above)
2. Launch STKN-SM-Config.exe
3. Establish the connection with the socket tray system
4. Open the configuration screen for tool recognition
5. Teach the tool parameters
6. Send the changed parameters to the socket tray
7. Run a test (optional)

The following chapters show the procedure in detail:

4.3.1 Establishing a Connection with the Socket Tray System

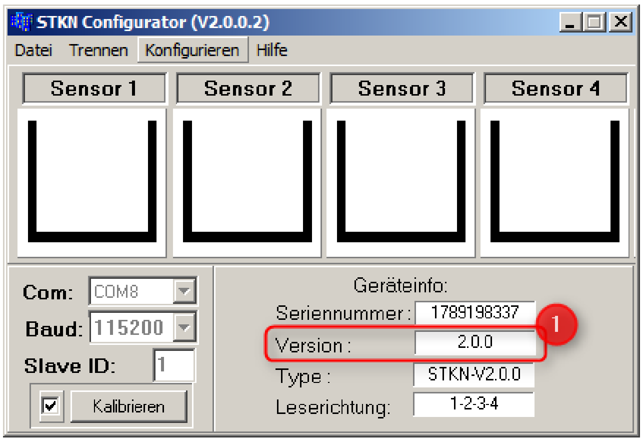
The following screenshot shows the main window of the STKN-SM-Config.exe-application after launching:

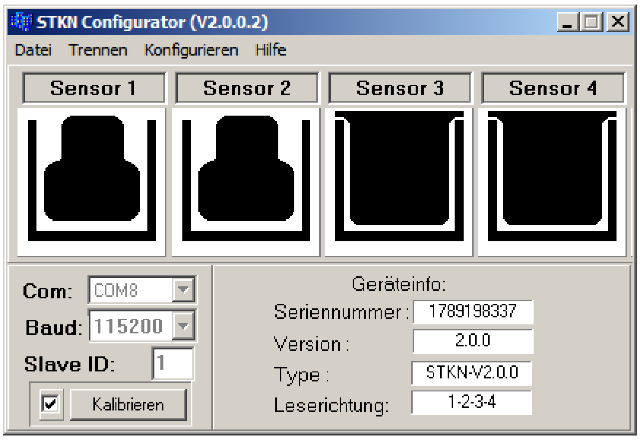


To establish a connection, please follow these steps:

1. Choose the serial port ❶. If you don't know the port number of the cable you are using, you can find that information by opening the Windows device manager (section "Ports").
2. Choose the Slave-ID (❸): The socket tray system assigns a unique address to each sensor module connected to the internal bus system. The smallest value is 1 and selects the first sensor module in the system. (most sensor modules have 4 sensors, only the MSTKN-**-T18-** have 8 sensors).
3. Choose the option „Verbinden (connect)" (❷) to establish the connection with the socket tray system.

When the connection has been established, the main window displays the information about the socket tray system as follows: (left picture: no tools, right picture: tools in slot 1/2, Slots 3/4 deactivated):

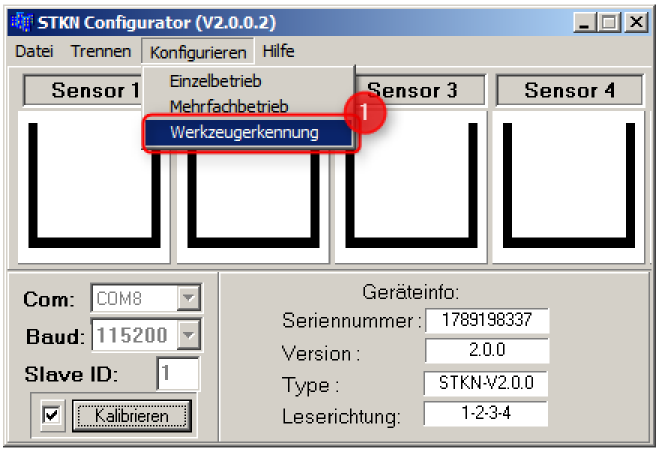




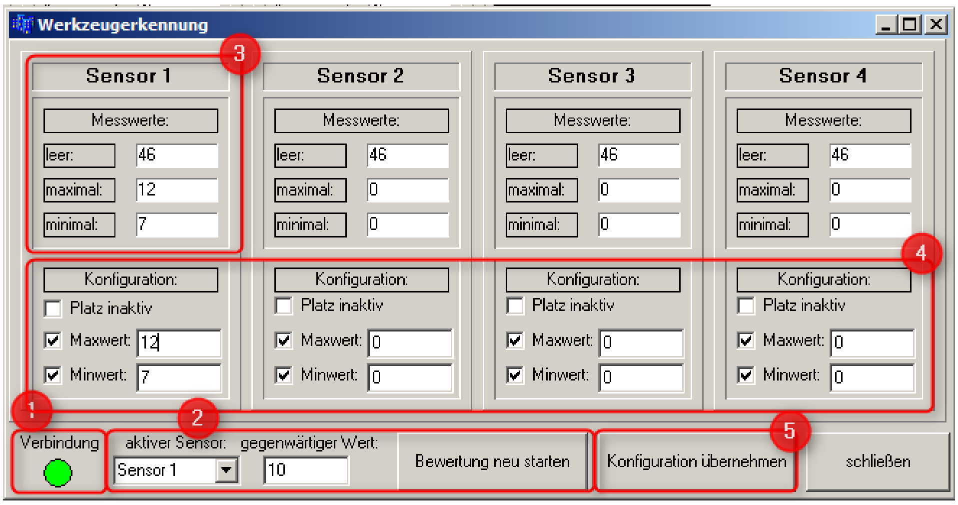
Note: Tool recognition and definition of tool parameters are only available for firmware version 2.0.0 and above! (see ❶ in the screenshot above)!

4.3.2 Overview of Tool Parameters

The tool recognition can now be accessed through the path Konfigurieren → Werkzeugerkennung (❶ in the screenshot below):



This opens the window for setting the tool parameters:



Das Fenster zeigt die folgenden Informationen:

This window shows the following information:

- State of the connection with the socket tray system (❶): green = connection OK, red = no connection (in this case, please close the window and try reconnecting as described above.)
- Active sensor (❷): the dropdown-menu allows you to quickly choose the active sensor from which the parameters shall be saved. The current value delivered by the sensor is shown in the box "gegenwärtiger Wert" while the maximum and minimum values for the sensor are shown in the box (❸) (here: sensor 1). The maximum and minimum value are only changed for the currently selected sensor – they can also be reset with the button „Bewertung neu starten".
- Input fields for tool recognition (❹): In this area, the threshold values for the tool recognition can be entered/changed. The maximum/minimum values from (❸) can be used here if the correct tool is in that spot. Basically, there are these input fields that can be configured:
  - Place inactive: If this checkbox is set, the tool slot is set inactive, rendering all LEDs of this spot inactive and the plug sealing this slot should be covered with a strip of copper tape.
  - Maxwert: If this checkbox is set, the value entered in the input field is used to check if the correct tool is placed in that spot. **Careful:** maximum value must not exceed the „empty" value (field „empty" at (❸)) and must not be the same value as the minimum value.

- Minwert: If this checkbox is set, the value entered in the input field is used to check if the correct tool is placed in that spot. **Careful:** minimum value must not exceed or be equal to the maximum value!
- If both checkboxes for minimum and maximum value are unchecked, the tool recognition is disabled and the socket tray behaves as it did when it was delivered.

- Send configuration parameters (❺): by pressing this button, all changes made to the configuration as described in this chapter are sent to the socket tray which saves them permanently (until you change them again).

Important: For tool recognition, Maxwert cannot be larger than the empty value (field „leer“ (❸)) and cannot have the same value as Minwert! Accordingly, Minwert is not allowed to be larger than Maxwert!

4.3.3 Step-by-Step Description

The procedure to configure the tool recognition requires the following steps:

1. Choose Sensor: with help of the dropdown “aktiver Sensor“ (❷), a sensor slot is activated and read.
2. Put the correct tool in the slot you activated.
3. Read the tool parameters:
    a. To initialize, press the button „Bewertung neu starten“ (❷) – this resets the min- and max values which are measured right now and shown in (❸).
    b. Then, move the tool around in the slot (if there is any space) to determine the minimum and maximum values. They will be shown in the field (❸).
4. Repeat these steps for alle other slots/sensors: Start over from step 1 (choose a new sensor) for all other slots.
5. Check the results: after all tool values have been taught to the socket tray, a test should be made if the tools are recognized without any mixups. This means that all minimum and maximum values for each tool must not overlap with each other. If that is not the case, you may have to repeat the process and also check the values the tools deliver when placed in the wrong spots (since the sensors in each spot may deliver slightly different values for the same tool. This is caused by the mechanical tolerances of each slot as well as the fact that each sensor delivers a slightly different value in general).

Careful: If it is not possible to recognize the tools without overlapping values, the tools cannot be distinguished reliably – in this case, the number of tools has to be reduced or a recognition without overlapping has to be guaranteed by mechanically altering the tool or the plug for the slots in question!

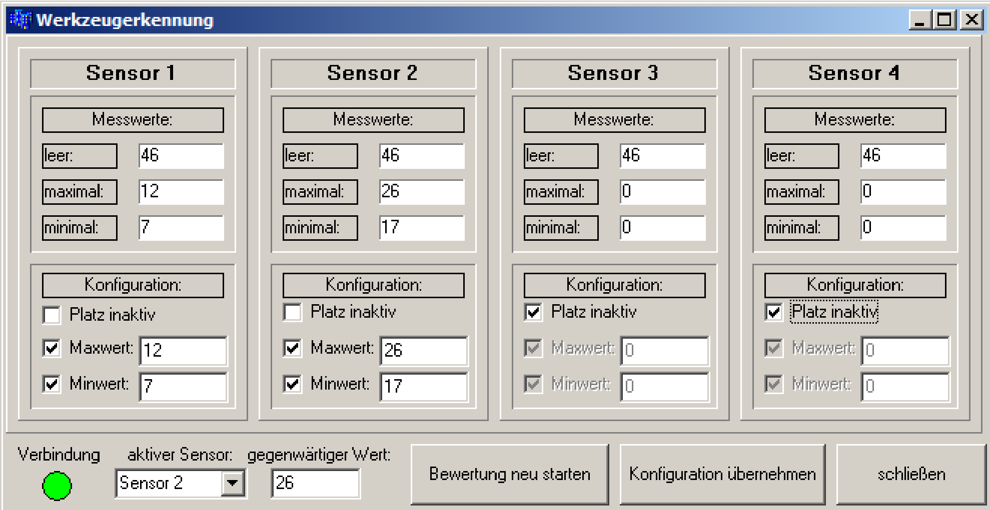
6. Inherit the parameters: after the overlap-free areas for the tool recognition have been found, they can be entered in the input fields for the tool recognition (❹). To guarantee an error-free operation, the min and max values entered should be a bit bigger than the values that were measured (for example: the minimum value entered is two points below the actual minimum value, the maximum value entered is two points above the actual maximum.)

Careful: To enter the values for the tool recognition in the appropriate input fields, make sure the the checkboxes before these fields are checked, otherwise it will not work!

If the tool recognition for a slot is to be deactivated, both checkboxes for maximum and minimum values have to be unchecked.

7. Send the newly configured values to the socket tray by pressing the button „Konfiguration übernehmen“

This example shows the tool recognition interface after successful configuration:



After the socket tray has been successfully configured, this window can be closed (button “schließen”).

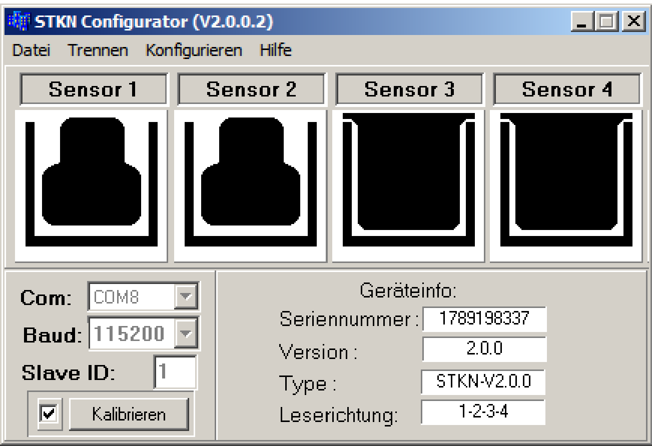
Careful: To check if the tool recognition is working properly, please perform a functional check in diagnostic mode (see Fehler! Verweisquelle konnte nicht gefunden werden.) or in normal mode. Please remember to also check if putting an incorrect tool into the slots is recognized as an error!

4.3.4 Testing the Tool Recognition

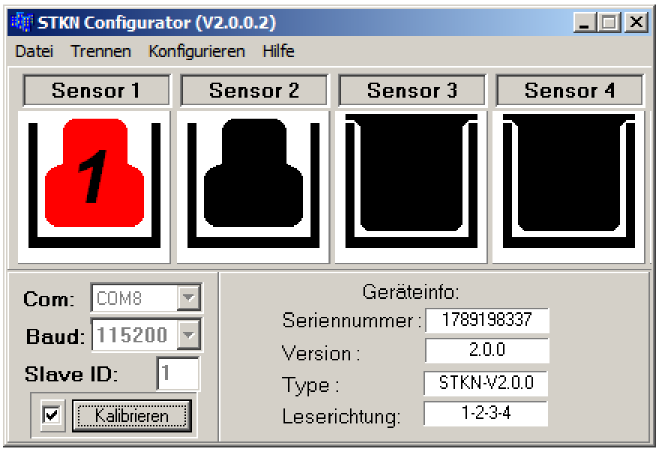
The proper function of the tool recognition should be tested in diagnostic mode (see **Fehler! Verweisquelle konnte nicht gefunden werden.**) or in normal mode. Testing if putting an incorrect tool into a slot prompts an error is also recommended!

In normal mode (socket tray is hooked up to the system) the function can be checked by commands from the superior system.

The diagnostic mode (see **Fehler! Verweisquelle konnte nicht gefunden werden.**) can be accessed by launching the STKN-SM-Config-application. The interface looks as shown below (example):



In the case shown above the tools in slot 1 and 2 have been placed correctly and the slots 3 and 4 are deactivated. If a tool is incorrectly placed (sensor values outside the predefined parameters) the slot is displayed in red. The screenshot below shows an interface displaying that there is a wrong tool placed in slot 1 (”Sensor 1”).



4.4 Configure grouping

It is also possible to create groups of multiple slots by using the application MSTKN-HM-Config.exe.

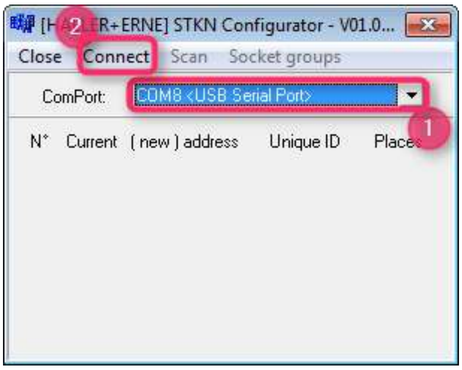
These are the basic steps:

1. Establish connection with the socket tray by USB ⇔ M12-cable
2. Launch MSTKN-HM-Config.exe
3. Establish a connection with the socket tray system
4. Open the interface for group configuration
5. Arrange groups
6. Send the parameters to the socket tray
7. Check function

The following chapters show the procedure in detail:

4.4.1 Establishing a Connection with the Socket Tray System

The following screenshot shows the MSTKN-HM-Config.exe’s main window:



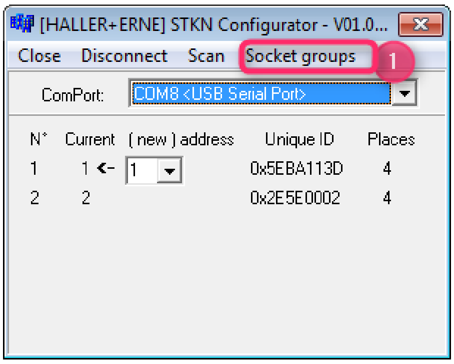
To establish a connection, follow these steps:

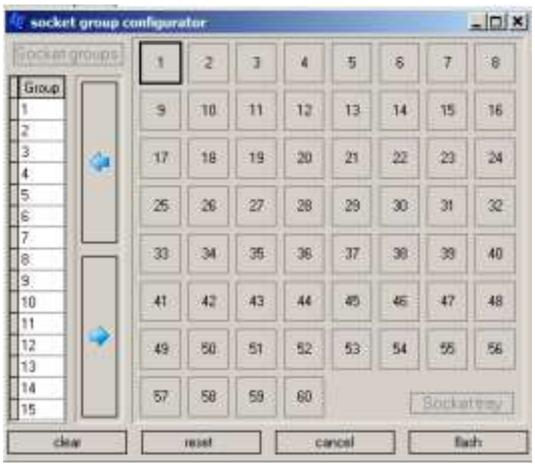
1. Choose serial path ❶. If you don’t know the number of the serial path your cable is using, you can find that information by opening the Windows device manager.
2. Choose the option „Connect“ (❷) to establish the connection with the socket tray system.

Careful: The grouping function is available only for firmware version 1.6.7 and above, please make sure to check your firmware version and update it incase it's older than V1.6.7!

4.4.2 Setting the Group Parameters

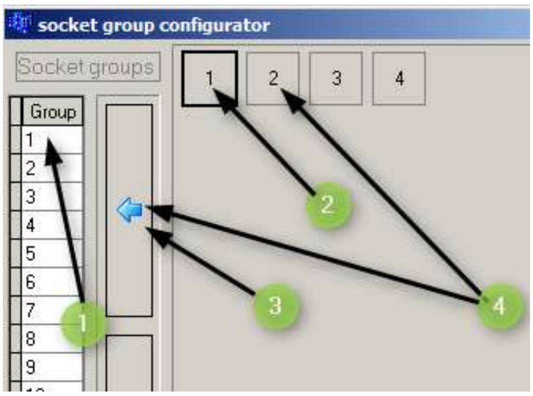
The group parameters can now be set through the menu „socket groups" (option ❶ in the screenshot below):





These steps are needed to delete all groups and return to standard configuration:

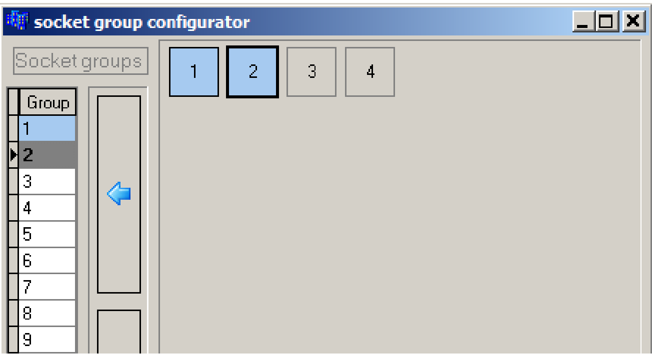
1. Press the "clear" button on the bottom left.
2. Press the "flash" button to send the configuration to the socket tray.



These steps are needed to configure a group:

1. Choose a group number (❶).
2. Choose a tool slot (❷).
3. Add the selected slot to the chosen group 1 (❸)
4. Repeat these steps for all other slots (❹).
5. Send the configuration you made to the socket tray by pressing the button „flash".

The screenshot below shows the interface after the first group containing slots 1 and 2 has already been configured and the second group has already been chosen to configure next (grey = currently selected group, blue = other groups).



4.4.3 Checking the Group Parameters

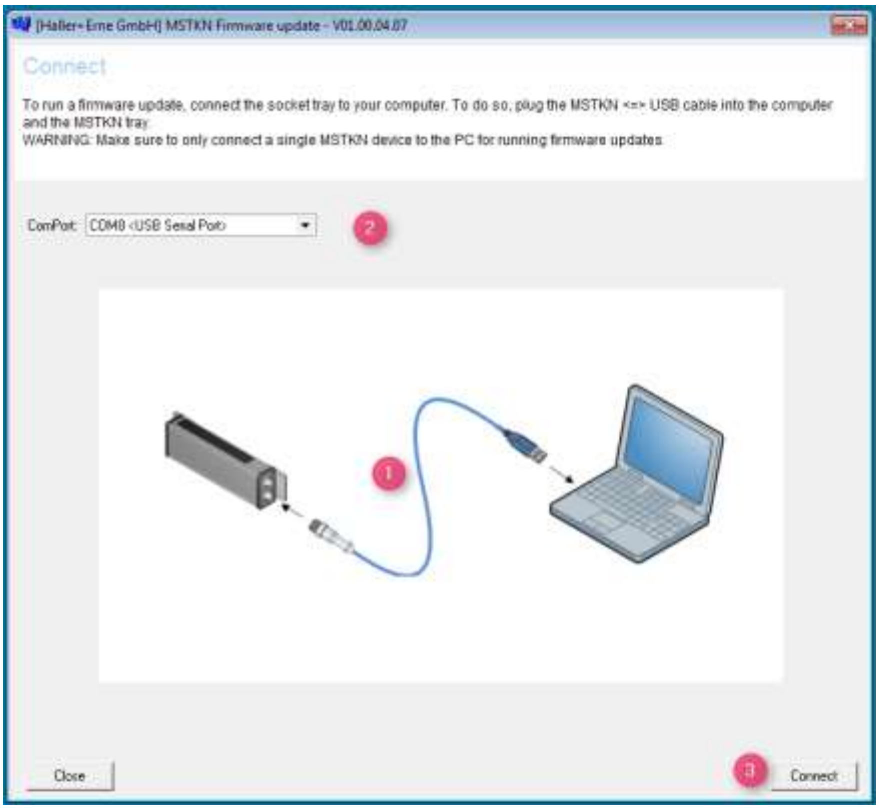
The group parameterization can be tested as described below:

- Connect the socket tray to the interface (e.g. PLC) and choose a „socket“ number. After configuring socket groups, choosing “socket 1” will result in choosing “group 1”.
- If the socket tray was configured as shown in the above example, the sockets 1 and 2 should be preselected now and their LEDs should start flashing. If you choose „2“, sockets 3 and 4 should be selected.

4.5 Firmware Update

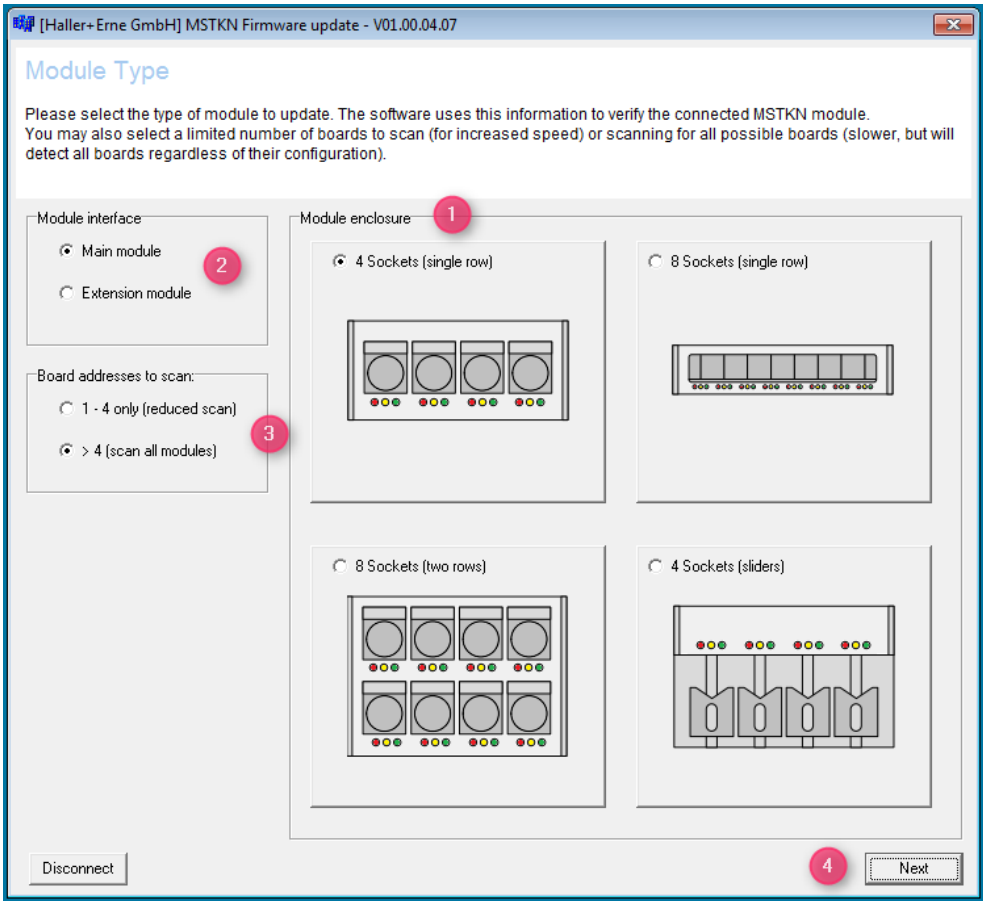
4.5.1 Establishing a Connection with Socket Tray System

Before you can start, please launch the MSTKN Firmware Updater program provided with your socket tray(s).



1. Connect the socket tray(s) via the USB → M12 cable.
2. Choose serial path. If you don’t know the number of the serial path your cable is using, you can find that information by opening the Windows device manager.
3. Click the „Connect“ button to establish a connection between the socket tray and the firmware updater.

4.5.2 Choosing the Type of Socket Tray



The image above shows the starting screen of the firmware updater after it was connected to the socket tray. Now please select the type of socket tray that you want to update.

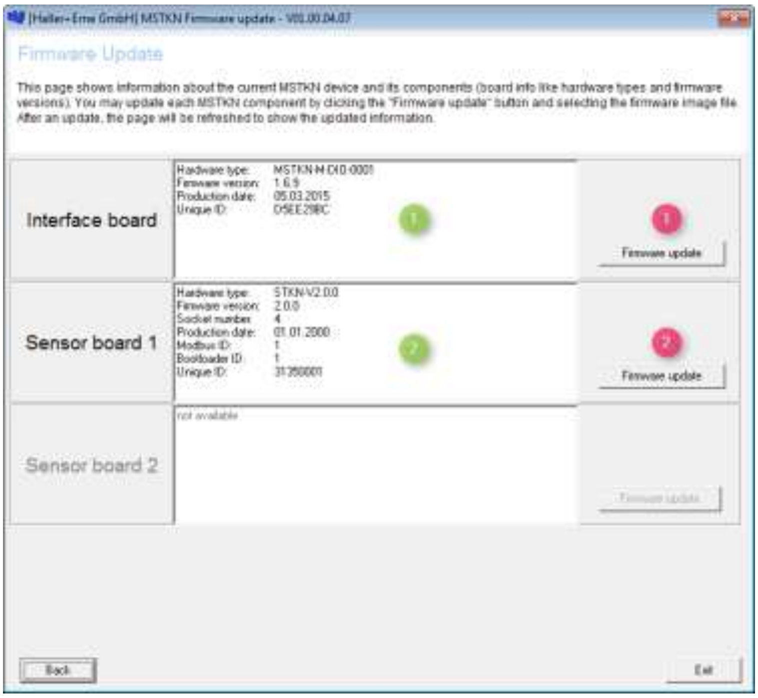
1. Select the type of enclosure.

4 Sockets (single row)	=	S14	(1 Sensor board)
8 Sockets (two rows)	=	S24	(2 Sensor board)
8 Sockets (single row)	=	T18	(1 Sensor board)
4 Sockets (sliders)	=	P14	(1 Sensor board)

2. Select if you want to update a “main module” or an “extension module”.
3. Select the number of sensor boards connected. Most of the times, it is recommended to use the Option „1 – 4 only (reduced scan)“, because a scan for all addresses takes very long. Even if you connect a main module with 2 sensor boards (8slots) and an extension module with 2 sensor boards (8 slots), the sensor boards will (if configured correctly) still have the addresses 1, 2, 3 and 4. So a „reduced scan“ is still enough to detect them all and it can save some time. The number of main boards is irrelevant for this option, they will be found anyway if “main module” was chosen. Only the number of sensor boards matters for this option.
4. If all options have been chosen correctly, click „Next“ to start scanning.

4.5.3 Choosing the correct File and performing the Firmware Update

The Firmware Updater has now scanned the connected socket tray(s) and listed all the boards and their respective, currently installed Firmware Versions.



The image above shows an example, in which a 24V I/O MSTKN with one sensor board (4 sockets) was connected.

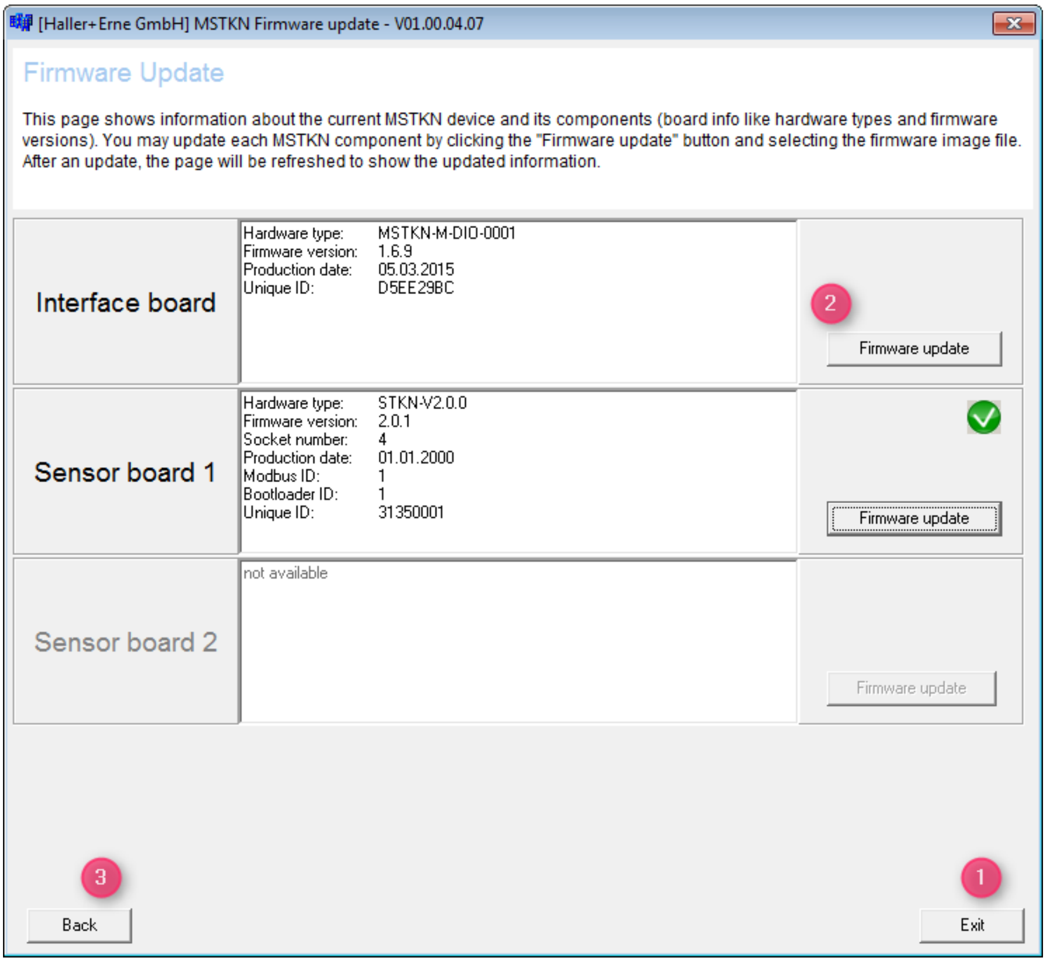
1. To update the gateway element („Interface board") click the button „Firmware update" ➊.
2. To update the sensor element („Sensor board 1") click the button „Firmware update" ➋. In this example, the sensor board will be updated.

Now the window shown to the right is opened. You can choose the file with which you want to update the board here.

1. Select the correct folder in the dropdown menu.
2. Select the correct .bin-file.
3. Now confirm your selection by pressing „Open"

The Firmware Updater is now beginning to update.

While the update is running, a progress bar will be shown. Once finished, the status window is shown again as follows:



5 Mechanical Data

5.1 Tray and Plugs

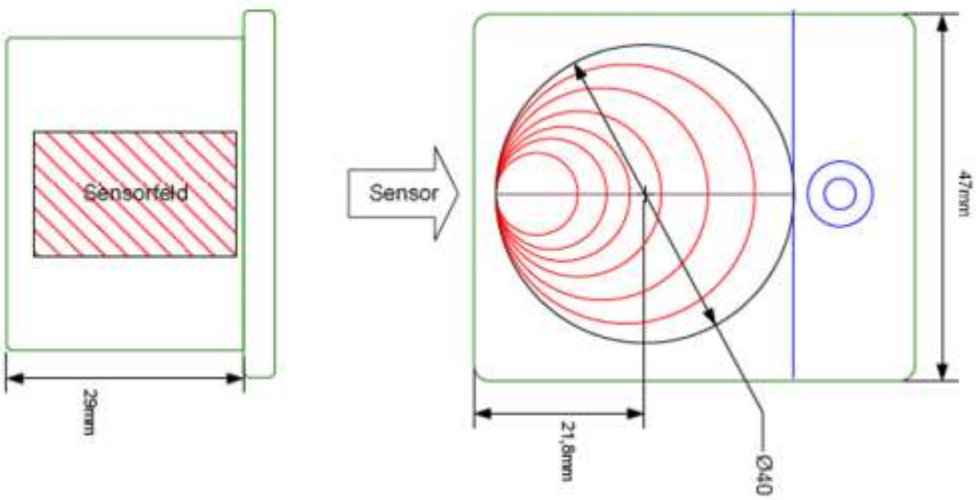
Careful: For maximum sensitivity, please make sure to drill/mill holes for the sockets according to the sensor field information in the dimensional drawings below. A milling on top of the plugs marks the center of sensor, as well as the optimum distance to the sensor.

Typical Dimensions:

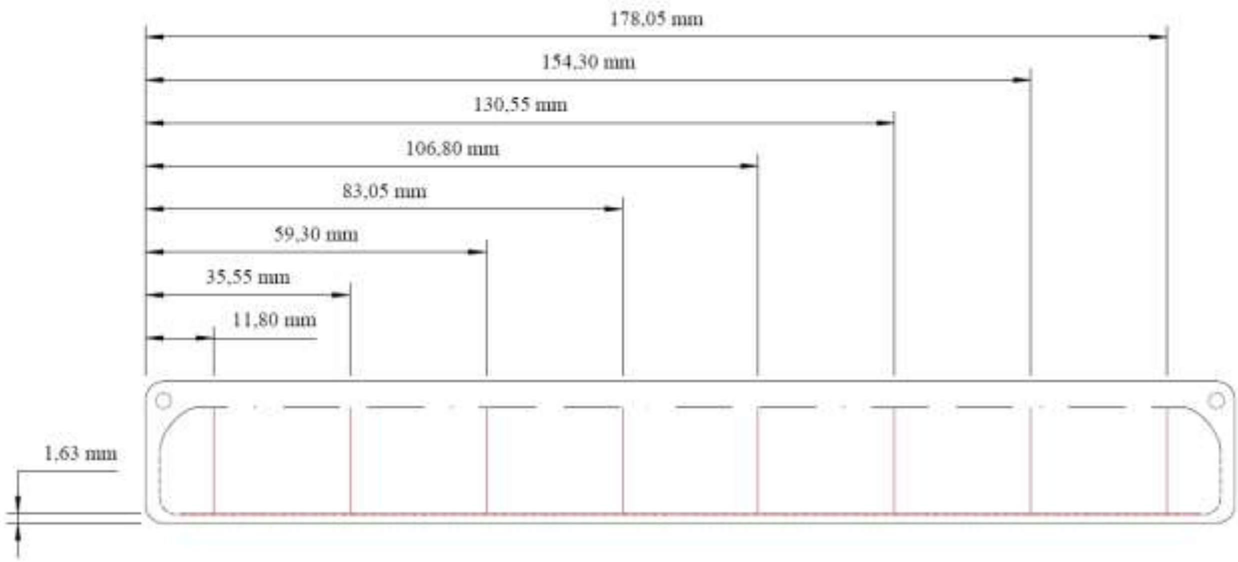
Drilling depth:	at least 12mm - max. 30mm (measured from top of the plug)
Distance to the sensor:	max. 2mm

If a problem occurs to detect a tool, this can be solved by reducing the distance to the sensor and or increase the drilling depth.

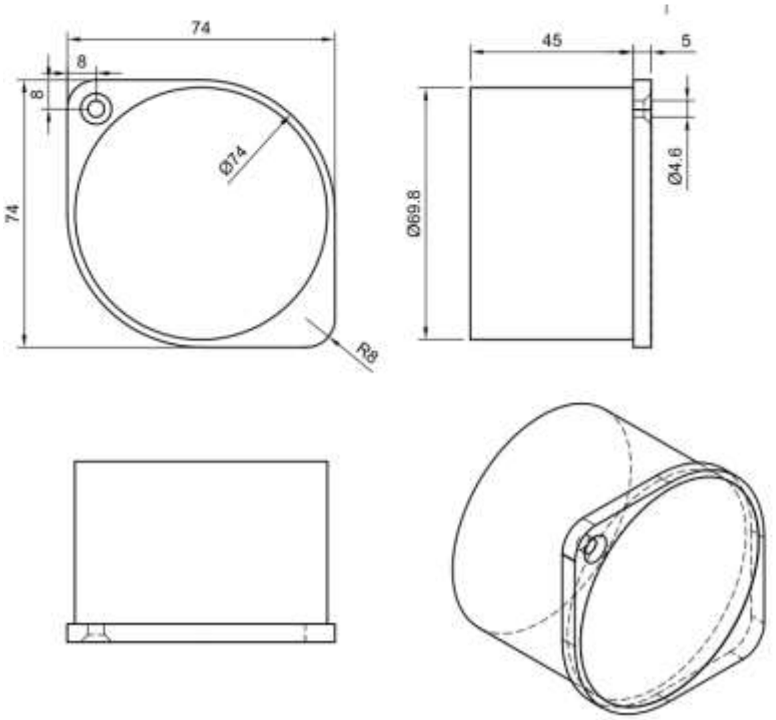
5.1.1 Dimensions for S14 and S24 Plugs



5.1.2 Dimensions for T18 Plugs

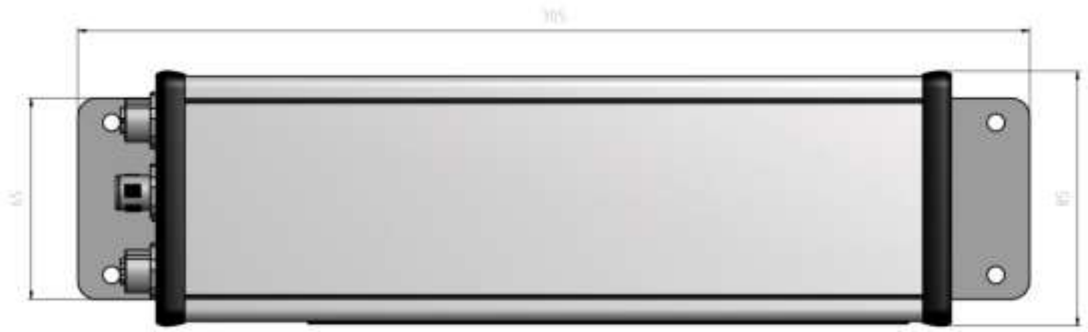


5.1.3 Dimensions for Plugs Qxx (70 mm)



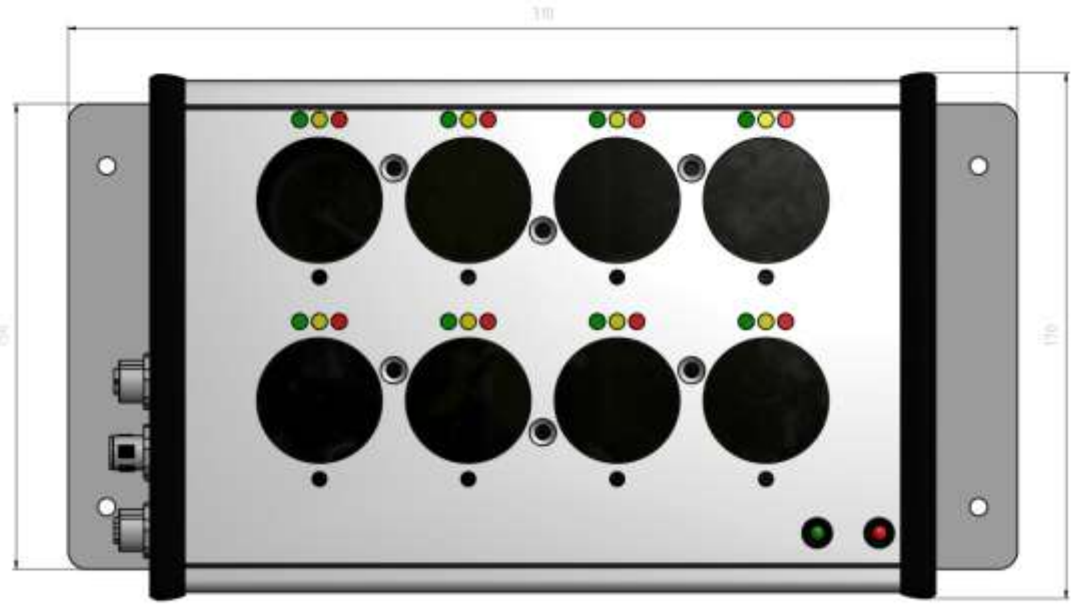
5.2 Enclosure Dimensions

H-T18 Enclosure:

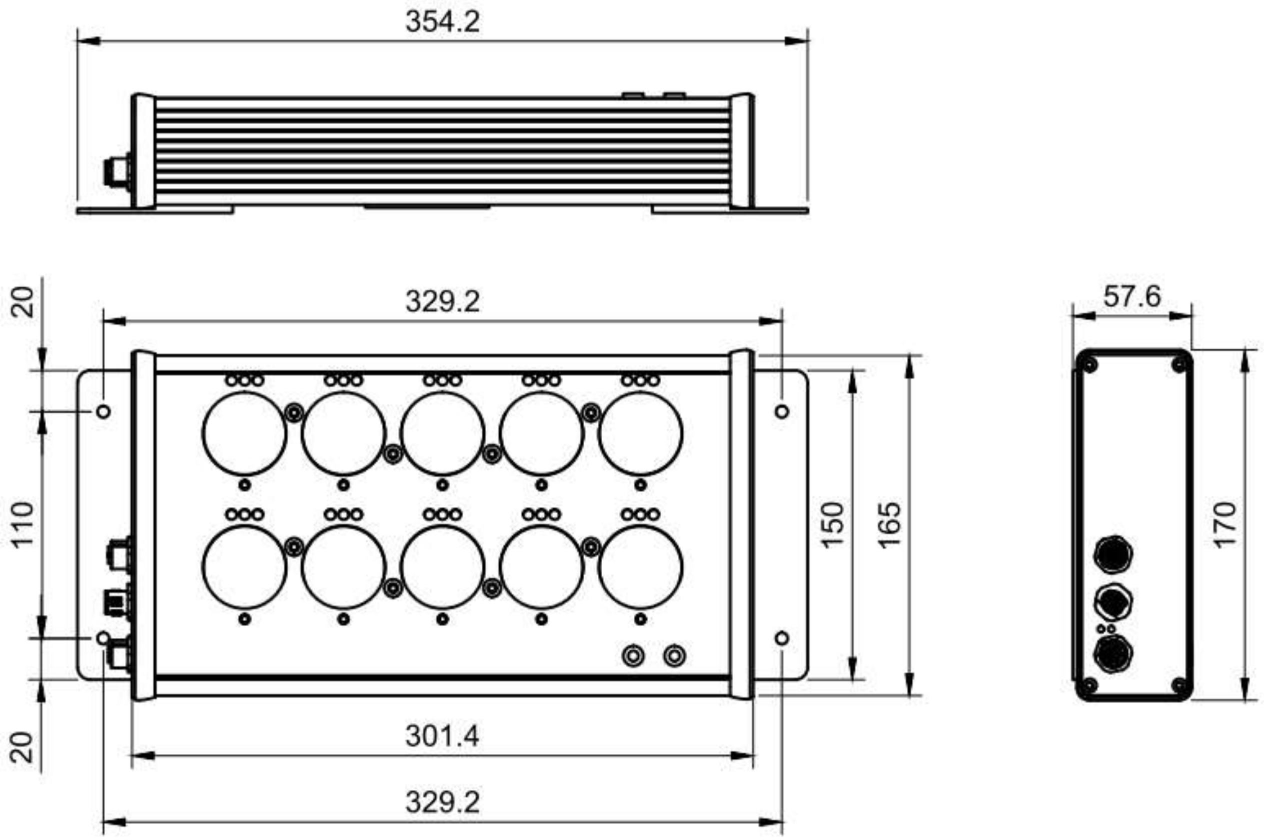


H-S14 Enclosure:

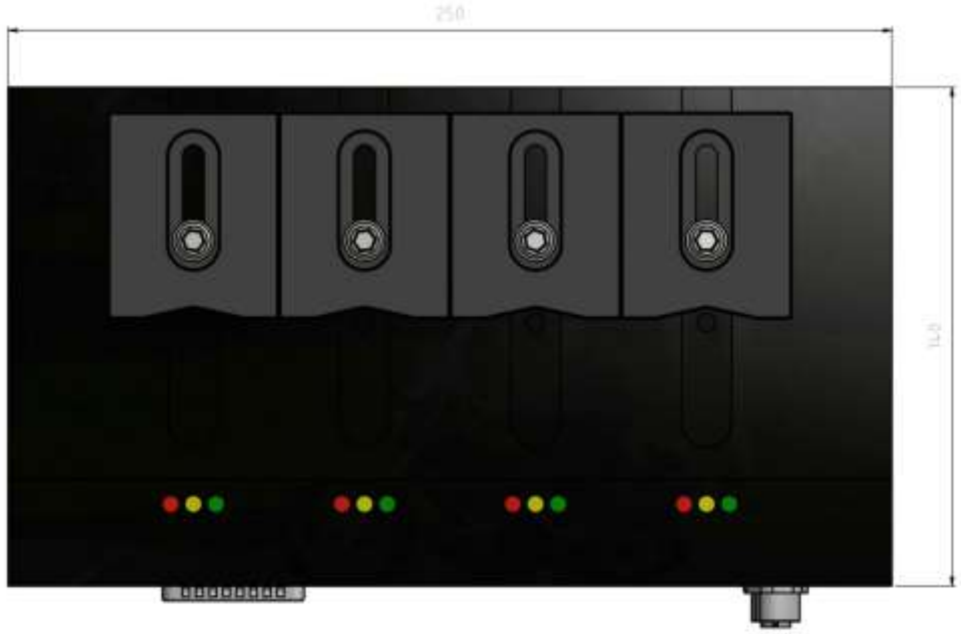
H-S24 Enclosure:



H-S25 Enclosure:



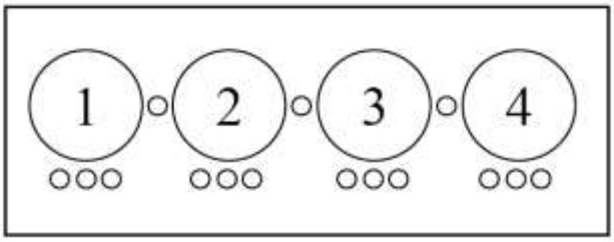
P-P14 Enclosure:



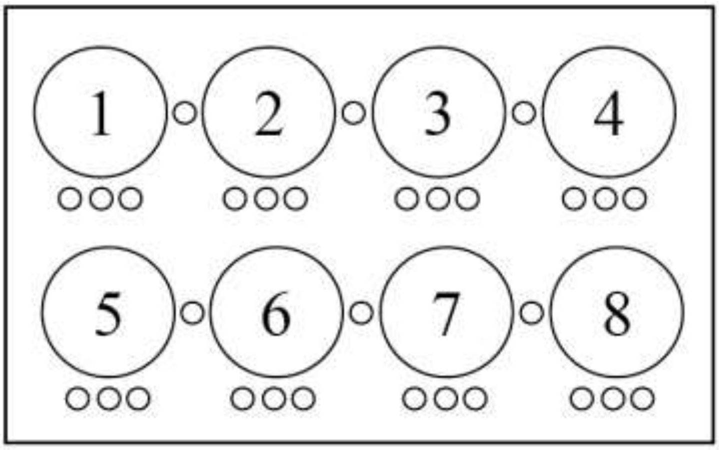
5.3 Tray numbering

The following drawings show the socket/tray numbering scheme – the numbering can be changed via software (see chapter 4.4):

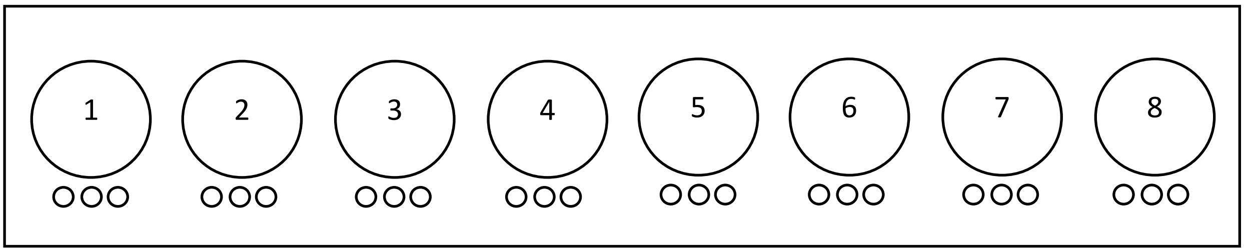
H-S14 / P-P14:



H-S24:



H-T18:



6 Declaration of Conformity

CE-Konformitätserklärung
CE-Conformity Declaration
Déclaration de conformité CE

Der Unterzeichner erklärt, dass die in der beiliegenden Artikelliste aufgeführten Stecknussköcher der EMV-Richtlinie 2014/30/EU und der Niederspannungsrichtlinie 2014/35/EU entsprechen

The undersigned declares, that the socket trays listed in the article list are conform to EMV guide-line 2014/30/EU and low-voltage-directive 2014/35/EU

Le signataire déclares que les boites à douilles énuméres sur le liste sont en la directive EMV 2014/30/EU et en la directive sur la basse tension 2014/35/EU

Produkttypen **Product types** **Types de produits**

MSTKN-H-S14-HM-DB25-M12	MSTKN-H-S24-HM-DB25-M12
MSTKN-H-S14-HM-DB25-M12-OP	MSTKN-H-S24-HM-DB25-M12-OP
MSTKN-H-S14-HM-DB25-M12-XML	MSTKN-H-S24-HM-DB25-M12-XML
MSTKN-H-S14-HM-PB-M12	MSTKN-H-S24-HM-PB-M12
MSTKN-H-S14-HM-PN-M12	MSTKN-H-S24-HM-PN-M12
MSTKN-H-S14-HM-PN2P-M12	MSTKN-H-S24-HM-PN2P-M12
MSTKN-H-S14-HM-ETH-M12	MSTKN-H-S24-HM-ETH-M12
MSTKN-H-S14-HM-WLAN-GBA	MSTKN-H-S24-HM-WLAN-GBA
MSTKN-H-S14-HM-WLAN-24V	MSTKN-H-S24-HM-WLAN-24V
MSTKN-H-S14-EM-M12	MSTKN-H-S24-EM-M12
MSTKN-H-S25-HM-DB25-M12	MSTKN-H-S25-HM-PN
MSTKN-H-S25-EM-M12	MSTKN-H-S25-HM-PN2P
MSTKN-H-Q13-ETH-M12	MSTKN-H-Q13-HM-WLAN-24V
MSTKN-H-Q12-ETH-M12	MSTKN-H-Q13-EM-M12
MSTKN-H-Q14-ETH-M12	MSTKN-H-Q12-EM-M12
MSTKN-H-Q15-ETH-M12	MSTKN-H-Q14-EM-M12
MSTKN-P-P14-HM-DB25-M12	MSTKN-H-Q15-EM-M12
MSTKN-P-P14-HM-PN-M12	MSTKN-P-P14-HM-DB25-M12-OP
MSTKN-P-P14-HM-PN2P-M12	MSTKN-P-P14-HM-ETH-M12
MSTKN-P-P14-EM-M12	MSTKN-H-T18-HM-DB25-M12
MSTKN-H-T18-HM-DB25-M12-OP	MSTKN-H-T18-HM-DB25-M12-XML
MSTKN-H-T18-HM-PB-M12	MSTKN-H-T18-HM-PN-M12
MSTKN-H-T18-HM-ETH-M12	MSTKN-H-T18-HM-PN2P-M12
	MSTKN-H-T18-EM-M12

Angewandte Normen: *Applied harmonized standards:* *Normes appliquées en particulier:*

EN 61000-6-4	EN 61000-6-2	EN 61000-4-3 2001-12
EN 61000-4-2: 1995 + A1:1998 + A2: 2001	EN 61000-4-3:2006 + A1: 2008	EN 60240-1

Nordheim, 01.01.2024

Holger Erne, Geschäftsführer

Haller + Erne GmbH
im Scheinmental 4
D 74226 Nordheim
Telefon 07133 961230
www.haller-erne.de