Drive Technology \ Drive Automation \ System Integration \ Services

SEW EURODRIVE

Manual

MOVI-C® CONTROLLER

with EtherCAT®/SBusPLUS System Bus

Edition 12/2018

25929488/EN



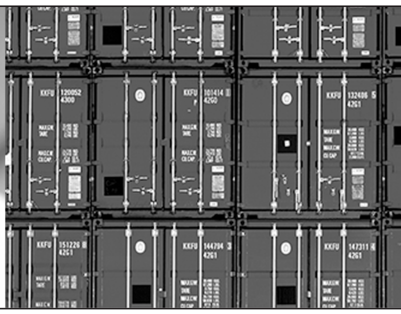



Table of contents

25929488/EN – 12/2018

25929488/EN – 12/2018

25929488/EN – 12/2018

1 General information

1.1 About this documentation

The documentation is part of the product and contains important information. The documentation is for everyone who works with this product.

The documentation must be accessible and legible. Make sure that persons responsible for the system and its operation as well as persons who work independently with the software and the connected units of SEW-EURODRIVE have read through the manual carefully and understood it. If you are unclear about any of the information in this documentation or if you require further information, please contact SEW-EURODRIVE.

1.2 Structure of the warning notes

1.2.1 Meaning of signal words

The following table shows the grading and meaning of the signal words for safety notes.

Signal word	Meaning	Consequences if disregarded
⚠ **DANGER**	Imminent hazard	Severe or fatal injuries
⚠ **WARNING**	Possible dangerous situation	Severe or fatal injuries
⚠ **CAUTION**	Possible dangerous situation	Minor injuries
NOTICE	Possible damage to property	Damage to the product or its environment
INFORMATION	Useful information or tip: Simplifies handling of the product.	

25929488/EN – 12/2018

1.2.2 Structure of section-related safety notes

Section-related safety notes do not apply to a specific action but to several actions pertaining to one subject. The hazard symbols used either indicate a general hazard or a specific hazard.

This is the formal structure of a safety note for a specific section:

SIGNAL WORD

Type and source of hazard.

Possible consequence(s) if disregarded.

- Measure(s) to prevent the hazard.

Meaning of the hazard symbols

The hazard symbols in the safety notes have the following meaning:

Hazard symbol	Meaning
	General hazard

1.2.3 Structure of embedded safety notes

Embedded safety notes are directly integrated into the instructions just before the description of the dangerous action.

This is the formal structure of an embedded safety note:

SIGNAL WORD Type and source of hazard. Possible consequence(s) if disregarded. Measure(s) to prevent the hazard.

25929488/EN – 12/2018

1.3 Right to claim under warranty

A requirement of fault-free operation and fulfillment of any rights to claim under limited warranty is that you adhere to the information in the documentation at hand. Therefore, read the documentation before you start working with the software and the connected devices from SEW-EURODRIVE.

Make sure that the documentation is available to persons responsible for the machinery and its operation as well as to persons who work independently on the units. Also ensure that the documentation is legible.

1.4 Content of the documentation

The descriptions in this documentation apply to the current software/firmware version at the time of publication. When new versions of software/firmware are installed, the descriptions may differ. In this case, contact SEW-EURODRIVE.

1.5 Other applicable documentation

Observe the following other applicable documentation:

- "MOVI-C® CONTROLLER power UHX85A and power eco UHX84A" manual
- "MOVIDRIVE® modular Application Inverters" operating instructions
- "MOVIDRIVE® system Application Inverters" operating instructions

Always use the latest edition of documentation and software.

The SEW-EURODRIVE website (www.sew-eurodrive.com) provides a wide selection of documents for download in various languages. If required, you can also order printed and bound copies of the documentation from SEW-EURODRIVE.

1.6 Product names and trademarks

The brands and product names in this documentation are trademarks or registered trademarks of their respective titleholders.

1.6.1 Trademark of Beckhoff Automation GmbH

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

1.7 Copyright notice

25929488/EN – 12/2018

2 Safety notes

2.1 Preliminary information

The following general safety notes serve the purpose of preventing injury to persons and damage to property. They primarily apply to the use of products described in this documentation. If you use additional components, also observe the relevant warning and safety notes.

2.2 Use

As the user, you must ensure that the basic safety notes are observed and complied with. Make sure that persons responsible for the machinery and its operation as well as persons who work on the device independently have read through the documentation carefully and understood it.

The following safety notes refer to the use of the software.

This document does not replace the detailed documentation for the connected devices. This documentation assumes that the user has access to and is familiar with the documentation for all connected products.

Do not perform installation or startup if the product is damaged.

Removing required covers without authorization, improper use or incorrect installation and operation may result in severe injury to persons, or damage to machinery.

All work in the areas of transportation, storage, operation and waste disposal must be carried out by persons who are trained appropriately.

2.3 Target group

Software specialist

Any work with the software may only be performed by a qualified specialist. Specialists in the context of this documentation are persons who have the following qualifications:

- Appropriate instruction
- Knowledge of this documentation and other applicable documentation
- SEW-EURODRIVE recommends additional training for products that are operated using this software.

The above-mentioned persons must have the authorization expressly issued by the company to operate, program, configure, label, and ground devices, systems, and circuits in accordance with the standards of safety technology.

2.4 Network security and access protection

A bus system makes it possible to adapt electronic drive technology components to the particulars of the machinery within wide limits. There is a risk that a change of parameters that cannot be detected externally may result in unexpected but not uncontrolled system behavior and may have a negative impact on operational safety, system availability, or data security.

Ensure that unauthorized access is prevented, especially with respect to Ethernet-based networked systems and engineering interfaces.

Use IT-specific safety standards to increase access protection to the ports. For a port overview, refer to the respective technical data of the device in use.

25929488/EN – 12/2018

3 Introduction

3.1 Short designation in the documentation

The following short designations are used in this documentation.

Type designation	Short designation
EtherCAT®/SBus^{PLUS} master	SBusPLUS master
MOVI-C® CONTROLLER power eco MOVI-C® CONTROLLER power MOVI-C® CONTROLLER advanced MOVI-C® CONTROLLER standard	MOVI-C® CONTROLLER
Higher-level controller	PLC
MOVISUITE® standard	MOVISUITE®

3.2 Content of this documentation

This document describes the communication between MOVI-C® CONTROLLER and all lower-level automation components using the following system bus system: EtherCAT®/SBus^{PLUS}.

25929488/EN – 12/2018

3.3 MOVI-C® CONTROLLER

The MOVI-C® CONTROLLER control family is characterized by a variety of performance classes and a suitable housing shape for every task. It ranges from a high quality cell controller to sophisticated automation control.

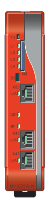
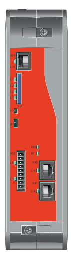
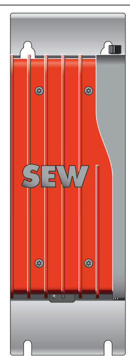
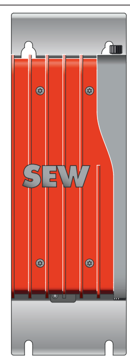
The MOVI-C® CONTROLLER comprises 4 performance classes in the following variants:

- MOVI-C® CONTROLLER standard is a compact profile rail controller and is optimized for coordinated two-axis applications (2 interpolating axes) paired with an additional 6 auxiliary axes.
- MOVI-C® CONTROLLER advanced is optimized for coordinated applications for up to 8 axes (interpolating axes) paired with an additional 8 auxiliary axes. It is installed with a stand-alone rear wall panel.

  This controller also optionally offers installation in a master module that can then be installed as a module within the MOVIDRIVE® modular series in an axis system. This can result in additional wiring advantages.
- MOVI-C® CONTROLLER power eco is designed in the performance class as an industrial PC. It is optimized for coordinated applications for up to 16 axes (interpolating axes) paired with an additional 16 auxiliary axes.
- MOVI-C® CONTROLLER power is also designed in the performance class as an industrial PC. It is optimized for coordinated applications for up to 32 axes (interpolating axes) paired with an additional 32 auxiliary axes. Furthermore, a second operating system, running separately from the real time PLC via modern Hypervisor technology, can be optionally activated on this controller, e.g. for integrated visualization. Workload spikes in the real-time PLC or the second Windows® Embedded Standard 7 operating system have no influence on the respective other side.

Difference between interpolating axes and auxiliary axes:

- With interpolating axes, the motion profile is calculated cyclically in the controller and is transmitted to the inverter via process data. The inverter then follows this cyclical setpoint by interpolation. The use of this operating mode ranges from coordinated positioning and electronic cams all the way to robotics.
- By contrast, auxiliary axes operate autonomously when it comes to the motion profile. Setpoint values, such as a target position, are transmitted to an auxiliary axis, which then approaches this target independently. This operating mode is suitable for single-axis applications, e.g. positioning, speed or torque control tasks.

			
MOVI-C® CONTROLLER ...			
standard UHX25A	advanced UHX45A	power eco UHX84A	power UHX85A

• 2 interpolating axes • 6 auxiliary axes	• 8 interpolating axes • 8 auxiliary axes	• 16 interpolating axes • 16 auxiliary axes PC-based	• 32 interpolating axes • 32 auxiliary axes PC-based Optional 2nd WIN7 operating system embedded
Allocated to MOVIRUN® ...			
smart/flexible	smart/flexible	flexible	flexible

All 4 controller classes can be optimally embedded in the system from SEW-EURODRIVE and have the following in common:

- Modern, clock-synchronized EtherCAT®/SBus^{PLUS} system bus master for the interface to the inverters and peripherals from SEW-EURODRIVE.
- Connection to conventionally available higher-level controllers via standard fieldbus-slave interface connections such as PROFIBUS, PROFINET, EtherNet/IP™ and Modbus TCP.
- Ethernet engineering interface with routing function to subordinate devices from SEW-EURODRIVE that are connected to the system bus.
- Common software world for engineering, programming, diagnostics, the MOVIRUN® software platform and the MOVIKIT® software modules.
- Freely programmable sequence control in accordance with IEC 61131 for automating drive and logic tasks (MOVIRUN® flexible).
- Central data storage for all MOVI-C® inverters from SEW-EURODRIVE.
- Plug-and-play device replacement through automatic data restoration.
- Routing of conventional security protocols from higher-level controllers to the drive inverters. In the first step, PROFIsafe was implemented via PROFINET.

3.4 MOVISUITE® engineering software

The MOVISUITE® engineering software is the operating platform for all MOVI-C® hardware and software components.

The following engineering tasks can be conveniently performed with MOVISUITE®:

- Project planning
- Startup
- Parameter setting
- Programming
- Diagnostics

3.4.1 Advantages of MOVISUITE®

MOVISUITE® offers the following advantages:

- Easy and intuitive operation
- Project management for storing the devices and data sets
- Convenient forwarding of projects via automatically created e-mails
- Automatic project creation from scanned devices
- Startup preparation in the planning phase even without connected devices

25929488/EN – 12/2018

- Design of the drive train from the motor to the gear unit via the product catalog or automatic identification of the gearmotor data by reading the electronic nameplate stored in the motor encoder.
- Product catalog for selecting encoders, brakes, control modes, user units
- Provision of standard drive trains and drive functions summarized in FCBs (**f**unction **c**ontrol **b**locks)
- Convenient, graphical startup and parameterization of the drive train
- Function view with overview of the status of the connected devices
- Creation of an overall system by structuring the devices as needed in the function view
- Easily calling up different manual modes
- Detailed messages and remedial measures in case of malfunction
- Identifying and supporting device replacements
- Synchronization functions for device data
- Data management with clearly marked transfer directions during data transmission
- Diagnostics of the drives with the integrated oscilloscope with up to 10 recording channels
- Integration of the IEC Editor for programming MOVI-C® CONTROLLER
- Integration of MOVIKIT® software modules (e.g. MOVIKIT® MultiMotion or MOVIKIT® MultiMotion Camming)
- Long-term data acquisition on the engineering PC hard disk

25929488/EN – 12/2018

4 SBus^{PLUS} system bus

4.1 Basics

The SBus^{PLUS} is the EtherCAT®-based system bus in MOVI-C® automation.

The communication protocol is standardized in accordance with international IEC standards and meets all of the requirements necessary with respect to plant and machine automation. These include among others a topology-free network, synchronization of system components, and extensive service and diagnostics options.

The SBusPLUS master enables MOVI-C® CONTROLLERs to integrate system components such as the MOVIDRIVE® application inverter and the MOVI-PLC® I/O system C. For complete system solutions, the open EtherCAT® protocol provides the opportunity to integrate third-party components such as IP67 I/O systems, linear measurement systems, RFID systems or valve blocks.

SBusPLUS master meets the following requirements in the MOVI-C® system:

- Startup functions, parameterization and engineering of system components with the MOVISUITE® engineering software.
- Clock-synchronous process data exchange between the SEW-EURODRIVE controller and the EtherCAT® stations in the network.
- Automatic addressing of EtherCAT® components during the startup phase and system expansion.
- High-precision synchronization through distributed clocks for interpolated axes.
- Simplified and fast startup with a configuration state of the system components.
- Safe data transmission for safety protocols.
- Services for saving and restoring the data sets of system components.
- Services for parameterization of additional automation components.
- Advanced diagnostics functions of network components.
- Identification and localization of errors in the network.

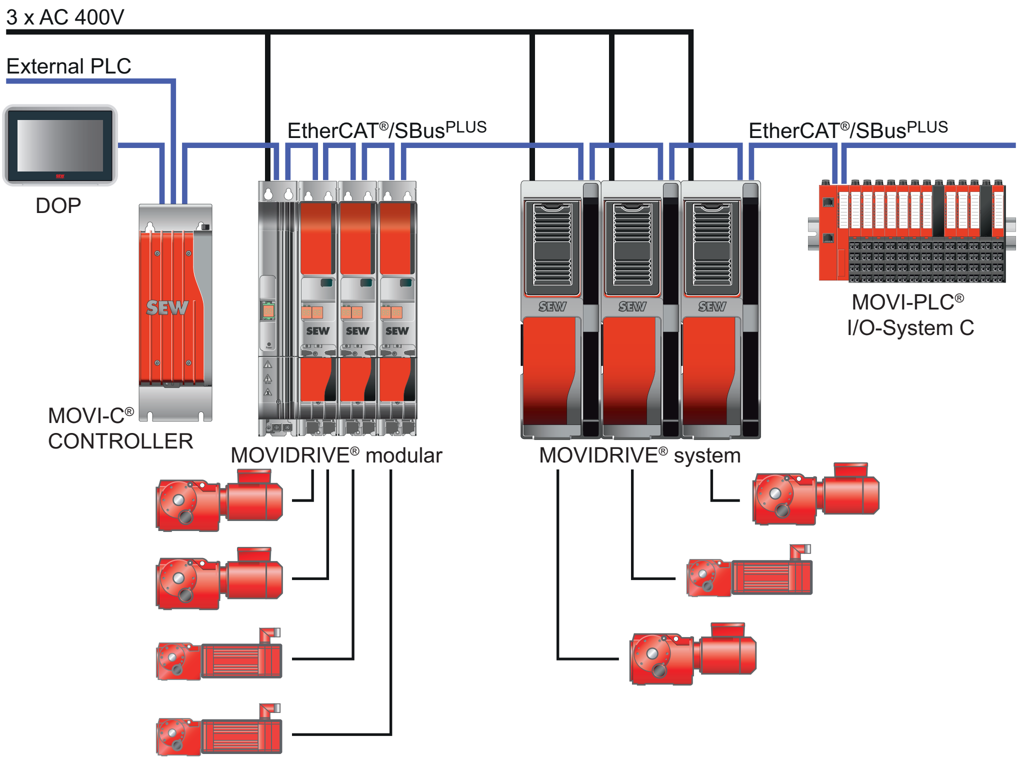
25929488/EN – 12/2018

4.2 Overview of the communication network

The entire clock-synchronous communication between MOVI-C® CONTROLLER and all subordinate automation components is implemented with the EtherCAT®/SBus^{PLUS} system bus.

Safe and non-safe process data, engineering and diagnostics data are transported via this bus.

Third-party components for importing the projection file that support EtherCAT® are also supported. A variety of addressing options for slave components enable simple data management in case of service incidents.



21908779275

25929488/EN – 12/2018

5 EtherCAT® basics

EtherCAT® is an Ethernet based fieldbus system. Process data between master and connected slaves is exchanged clock-synchronously in one cycle. The slaves retrieve their data while at the same time inserting input data into the telegram. This means the master receives the input data of the slaves within one cycle.

This chapter explains the most important EtherCAT® terms.

5.1 Description files

EtherCAT® Slave Information file (ESI file)

The ESI file is a device description in XML format. The user creates this file in the configuration tool separately for each EtherCAT® slave. The ESI file contains information about the objects and object properties of the slaves. The configuration tool accesses the ESI files to create the ENI file.

EtherCAT® Network Information file (ENI file)

The ENI file is created by the configuration tool. The ENI file provides the EtherCAT® master with the following information:

- Information on the network topology
- Information on the initialization commands for all network stations
- Commands that the master sends cyclically to the slaves

Slave Information Interface (SII)

SII is an EEPROM that contains information about the hardware configuration of the EtherCAT® slave controller (ESC). When booting, the slave applies the SII file to the register of the ESC.

25929488/EN – 12/2018

5.2 Service commands

EtherCAT® datagrams transport various service commands and are addressed to one or several slaves in the network.

The master sends, receives, and evaluates service commands. Each service command has a working counter (WC). The purpose of the working counter is to provide feedback to the master whether data has been exchanged successfully or not. This feedback is obtained in the same cycle.

The following service commands are defined:

Service command		Description
APRD	Auto Increment Read	Slave increment address. The slave writes read data into the EtherCAT® datagram if the received address is zero.
APWR	Auto Increment Write	Slave increment address. The slave writes data into memory location if the received address is zero.
APRW	Auto Increment Read Write	Slave increment address. The slave writes read data into the EtherCAT® datagram and writes the data to the same memory location if the received address is zero.
FPRD	Configured Address Read	The slave writes read data into the EtherCAT® diagram if the received address is identical with one of the configured addresses.
FPWR	Configured Address Write	The slave writes data into the memory location if the address is identical with one of its configured addresses.
FPRW	Configured Address Read Write	The slave writes read data into the EtherCAT® datagram and writes the data into the same memory location if the received address is identical with one of the configured addresses.
BRD	Broadcast Read	All slaves write the logical OR of the data of the memory location and of the EtherCAT® datagram into the EtherCAT® datagram. All slaves increment the position field.
BWR	Broadcast Write	All slaves write data to the memory location. All slaves increment the position field.
BRW	Broadcast Read Write	All slaves put logical OR of data of the memory area and data of the EtherCAT® datagram into the EtherCAT® datagram and write data into memory location. BRW is typically not used. All slaves increment to the position field.

25929488/EN – 12/2018

Service command		Description
LRD	Logical Memory Read	The slave writes read data into the EtherCAT® datagram if the received address is identical with one of the configured FMMU areas for reading.
LWR	Logical Memory Write	The slave writes data into the memory location if the received address is identical with one of the configured FMMU areas for writing.
LRW	Logical Memory Read Write	The slave writes read data into the EtherCAT® datagram if the received address is identical with one of the configured FMMU areas for reading. The slave writes data into the memory location if the received address is identical with one of the configured FMMU areas for writing.
APMW	Auto Increment Read Multiple Write	The slave increments the address. The slave writes read data into the EtherCAT® datagram if the received address is zero. Else the slave writes the data into the memory location.
FRMW	Configured Read Multiple Write	The slave writes read data into the EtherCAT® diagram if the address is identical with one of the configured addresses. Else the slave writes the data into the memory location.

INFORMATION

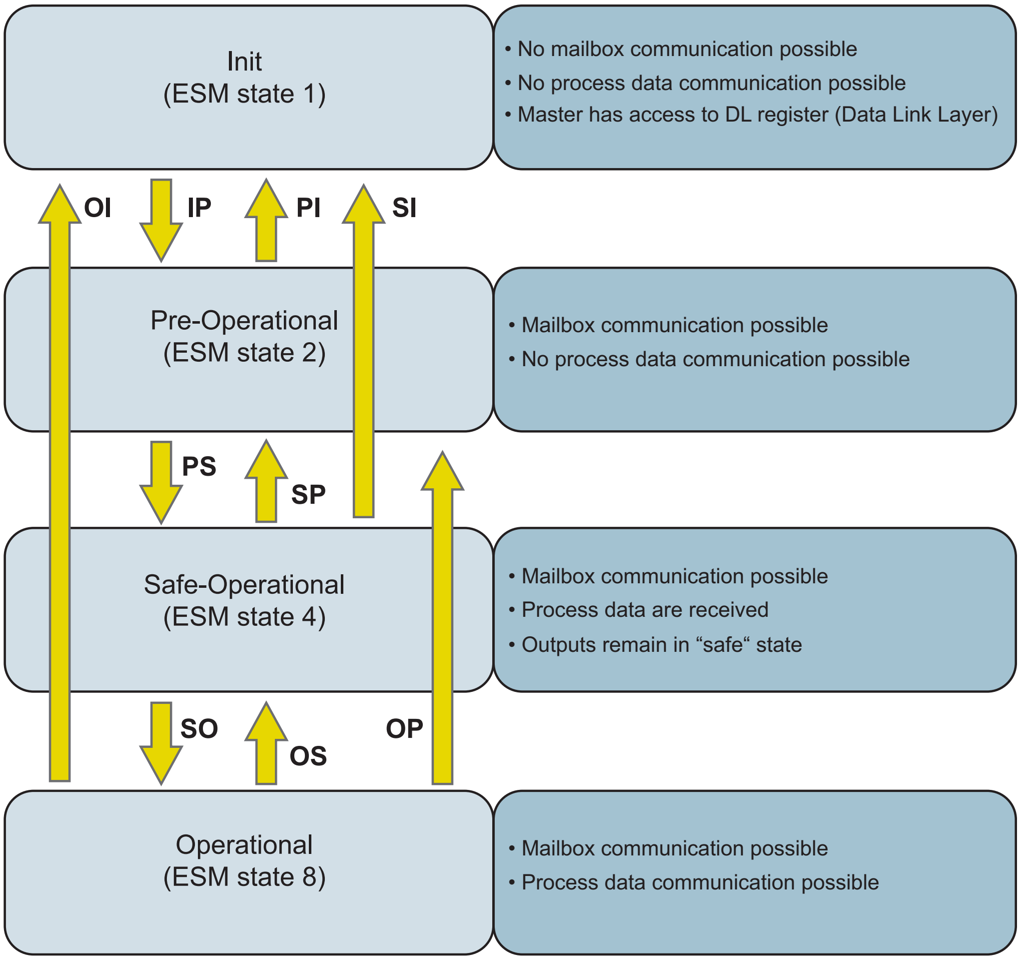
The process length is shorter if you use the LRW command instead of the LRD/LWR command. Not all slaves support LRW commands. Refer to the ESI file for information whether a slave supports these commands or not: UseLrdLwr = 0 or not specified means "Slave supports LRW and LRD/LWR". UseLrdLwr = 1 means "Slave only supports LRD/LWR".

5.3 EtherCAT® state machine (ESM)

The slaves pass a state machine, referred to as EtherCAT® state machine (ESM). The state machine allows the stations in the network different functions in the respective ESM states. The master sends requests to the slaves and the slaves have to confirm them in order to fulfill the transition conditions of the ESM states.

At the start, master and slaves are in "Init" ESM state. Communication in the network takes place in "operational" ESM state.

The figure below shows a simplified EtherCAT® state machine (without start routine) and the functions assigned to the various states.



22302514187

The master writes ESM requests, such as a change of state or error acknowledgement, into the "AL control register" of the slave controller (ESC). When the requests are acknowledged, the slave sets the AL status register to "acknowledge". If not, the slave sets the AL status register to "false" and enters an error code in the AL status code register. These registers are important for diagnosing network problems and for locating slave errors.

25929488/EN – 12/2018

5.4 Cyclic process data exchange

The cyclic process data, which are written and read synchronously and consistently in the network, are defined by the configuration tool.

A master can support several cycle tasks (sync units) that contain different cycle times and/or different frame lengths.

With EtherCAT®, any data of individual or various stations can be logically combined to so-called sync units. The data are grouped by means of the modularity of the control application. The grouping is not subjected to the rather arbitrary arrangement of I/O modules. Each sync unit contains a diagnostic input information (sync unit state) that is updated cyclically. The sync unit state indicates whether the data were exchanged successfully.

If a slave in a sync unit fails, the process data of the respective sync unit will become invalid. The process data of the other sync units remain valid.

Several sync units are useful in applications where parts of the machine have to run although other units are faulty, have been removed, switched off, or disabled. Several sync units allow for a modular system structure.

5.4.1 FailSafe over EtherCAT® (FSoE)

The EtherCAT® bus provides the option of transferring safety-relevant data on the same medium. The transport medium is considered as "black channel" and is embedded into the cyclical process data. The standard communication system thus remains a single-channel system and transfers standard information and safe process data in parallel.

5.5 Acyclic process data exchange

EtherCAT® stations exchange the following process data via acyclic communication:

- Information about configuration
- Information about parameterization
- Diagnostics information

The basis for this are mailbox channels.

5.5.1 CAN application protocol over EtherCAT® (CoE)

EtherCAT® provides the communication mechanisms known from CANopen:

- Object list
- Process data objects (PDO)
- Service data objects (SDO)

CoE is the most widespread acyclic EtherCAT® communication.

The object directory of a slave is accessed via SDOs. The master can access the parameters both at startup and during operation.

CoE offers configuration mechanisms of the PDOs (process data objects) for cyclic communication, such as process data length and process data content. This is useful when configuring slaves with variable PDOs, such as gateways and bus couplers.

25929488/EN – 12/2018

5.5.2 Vendor-specific protocol over EtherCAT® (VoE)

VoE allows for example to implement the protocol of SEW for setting parameters, engineering, and diagnostics of MOVI-C® system units using MOVISUITE®.

5.5.3 File access over EtherCAT® (FoE)

EtherCAT® provides an FoE protocol for easily transferring files or data. This protocol allows for downloading firmware without having to use the TCP/IP function.

5.5.4 Ethernet over EtherCAT® (EoE)

Other Ethernet-based protocols and services can also be used with EtherCAT® technology. Ethernet frames are tunneled via the EtherCAT® protocol. Doing so does not affect the real-time function.

5.6 Configuring and addressing the network

With EtherCAT®, the master assigns the slave addresses automatically. The IEC Editor configuration tool from SEW-EURODRIVE does not provide for setting the address manually.

The address of the first station begins with 1001 (fixed station address). All other addresses are incremented by 1 depending on the configuration of the project at the time of configuration.

The network is configured by scanning the master in the network and by reading the EEPROMs (SII) of the respective slaves. When booting, the master compares the existing network configuration with the created configuration in the project.

The master compares the following data when booting:

- Vendor ID
- Product code
- Revision no.

INFORMATION

The connection of devices of the same type (identical vendor ID, product code and revision code), can be swapped. The master will still start.

5.7 Explicit device identification

You can activate special identification mechanisms for EtherCAT® slaves in addition to the automated addressing of the network.

These identification mechanisms offer the following advantages:

- Comparing the ID in the loaded project with the ID in the slave prevents the slaves from being swapped.
- The sequence of slaves in the EtherCAT® branch is adhered to.

The identification mechanisms can be divided into the following categories:

- Station alias
- ID selection

25929488/EN – 12/2018

INFORMATION

The explicit device identification is an additional mechanism for identifying the slaves. The explicit device identification does not replace automatic addressing.

5.7.1 Station alias

When "station alias" is configured, the EEPROM of the slave can be written in the configuration tool using an online function. Doing so will assign the slave a unique ID in addition to the address.

INFORMATION

After replacing the device, the ID must be transferred to the device again.

5.7.2 ID selection

The ID assigned for the slave in the project must match the ID set using the DIP switch of the slave.

The advantage is that there is no need for an online function when replacing a device. This means modules can be replaced without project planning tool.

INFORMATION

Not all slaves support a DIP switch.

INFORMATION

Bear in mind that the ID can be set either in decimal or hexadecimal notation.

5.8 Optional slaves

The principle of optional slaves works as follows:

- All optional slaves must have been configured in the Configurator.
- But slaves marked as "optional" may be missing in the unit structure from perspective of the hardware. The master will still start.

This offers the advantage that you can create projects with devices that are available only partially in the plant. The project remains unchanged.

The following applies when using optional slaves:

- The master detects only optional slaves that are connected in terms of hardware when starting the master.
- Optional slaves that are added during ongoing operation will not be detected by the master until restart.
- Optional slaves need an explicit device identification either via ID selection with hardware DIP switch or via station alias.

5.9 Distributed clocks (DC)

The method of distributed clocks (DC) allows for precise synchronization between stations in the EtherCAT® network. The jitter is less than 1 µs.

Whether choosing the DC option is useful or not depends on the application.

In applications where servo drives travel along geometrical paths, for example, using DC is the basic requirement for simultaneous interrelated movements of the axes. This is the reason why the DC function is implemented in many slaves.

The first DC slave after the master contains the reference clock. This achieves the best possible synchronization.

The following applies to networks with activated DC:

- The master restarts the network as soon as one of the following criteria is met:
  - The master detects a slave in "Init" ESM state
  - The master detects a reference clock in "safe operational" ESM state
- The network is calibrated again by the restart. The slaves synchronize again.

5.10 EtherCAT® feature packs

Cable redundancy

When the "cable redundancy" feature is implemented, EtherCAT® communication is maintained despite communication errors (such as cable breakage).

The cable redundancy is implemented in the form of a ring topology. The ring is connected to the controller by a second network port. This increases the availability of the network.

Hot connect

When the "hot connect" feature is implemented, you can connect slaves to the bus without having to perform further actions.

External synchronization

When the feature "external synchronization" is implemented, the clocks of the slaves are synchronized beyond plants and sites based on a higher-level time base, such as the UTC universal time. This means a global time base is available that can be used, for example, for measurement data with timestamps.

6 Extended SBus^{PLUS} functions

This chapter describes basic information about simplified startup provided by the EtherCAT®/SBus^{PLUS} master.

Configuration state

The configuration state allows for parameterizing, configuring, and engineering the lower-level system components in MOVISUITE®. The network is scanned cyclically.

During runtime, you can add further slaves to the network without having to restart the master.

Automatic code generation lets you integrate devices, which are already available in the network, into an IEC project and check it for functioning.

Modular startup of the system is possible.

INFORMATION

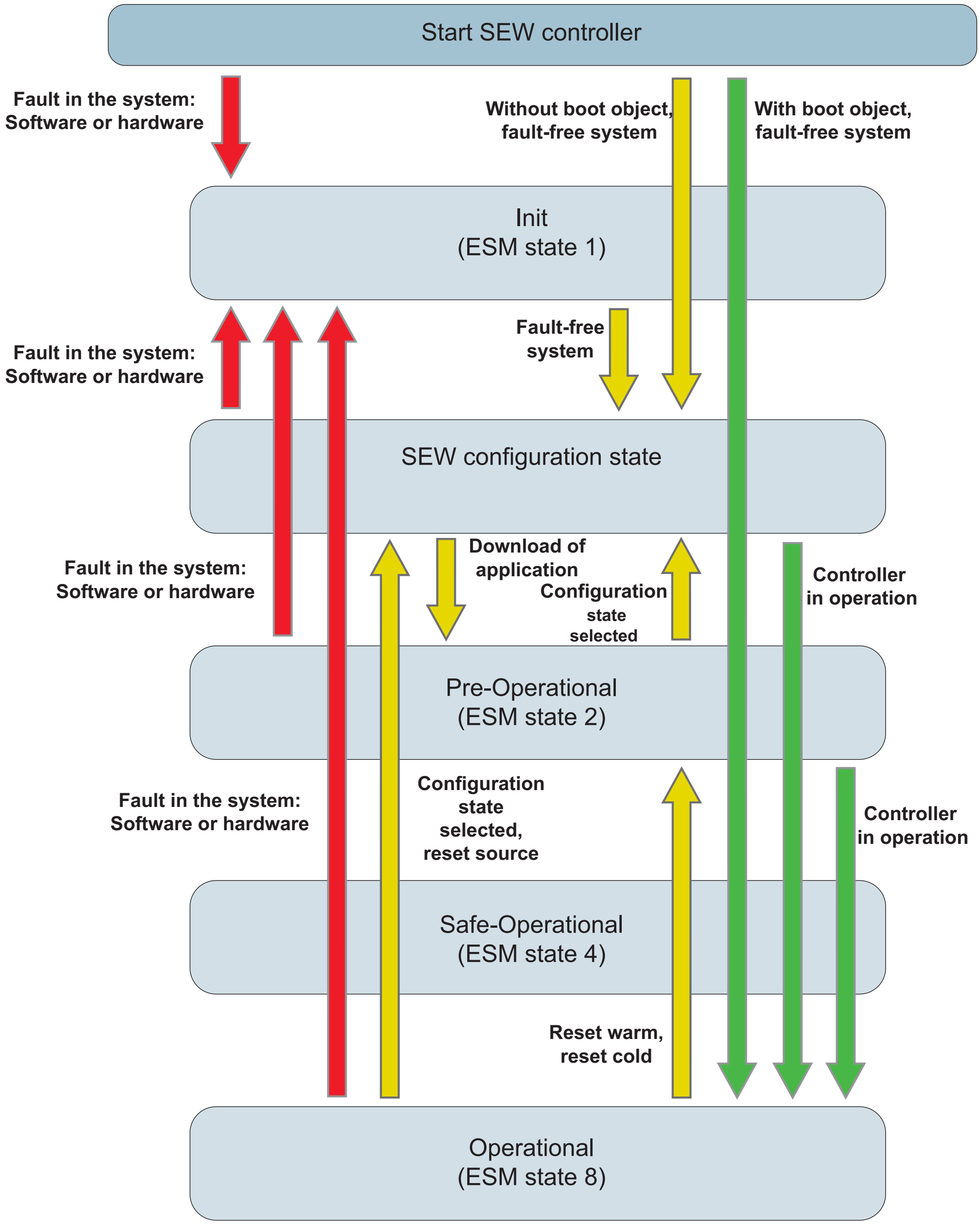
Unlike the states "pre-operational", "safe operational", and "operational", the master does not assign fixed station addresses in configuration state but assigns position addresses (AutoInc addresses).

The following table shows the ESM states of the advanced state machine with available functions:

	"Init"	Configuration state	"Pre-operational"	"Safe operational"	"Operational"
Network scan	Yes	Yes	With projected hardware configuration	With projected hardware configuration	With projected hardware configuration
Addressing of slaves	0 or more Increasing	0 or more Increasing	1001 or more Increasing	1001 or more Increasing	1001 or more Increasing
Use of engineering functions and status display in MOVISUITE®	No	Yes	Yes	Yes	Yes
Exchange of process data via SEW controller	No	No	No	Process inputs only	Yes

25929488/EN – 12/2018

The following figure shows the advanced state machine with the respective transition conditions between the ESM states.



22302701835

7 Startup

7.1 Requirements

Before you can take an application into operation, the following requirements must have been met:

- MOVISUITE® is installed on your computer. MOVISUITE® has the IEC Editor program integrated for programming the MOVI-C® CONTROLLER.

  For information on MOVISUITE®, refer to the "MOVISUITE®" manual.
- The EtherCAT®/SBus^{PLUS} stations are connected to the MOVI-C® CONTROLLER. For more information, refer to the "MOVI-C® CONTROLLER" manual
- The MOVI-C® CONTROLLER is connected with the computer via Ethernet.
- MOVISUITE® is started.

INFORMATION

When delivered, the IP address of the engineering interface of the MOVI-C® CONTROLLER is 192.168.10.4

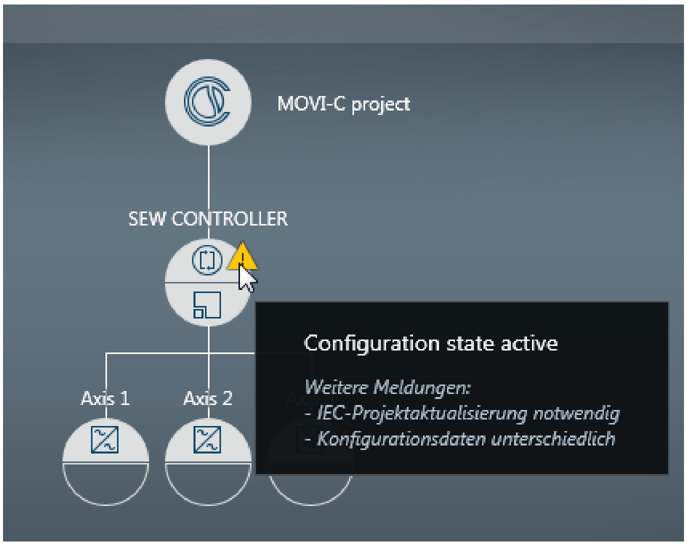
7.2 Activating the configuration state of the MOVI-C® CONTROLLER

The MOVI-C® CONTROLLER must be in configuration state to configure further SBus^{PLUS} stations.

The following applies to configuration state:

- You can add further SBus^{PLUS} stations to the network.
- Parameter setting, diagnostics and engineering functions are possible for the auxiliary axes.
- The exchange of process data with the MOVI-C® CONTROLLER is disabled.

A device notification is displayed for the module to indicate whether the MOVI-C® CONTROLLER is in configuration state.



22346563467

In the following cases, the MOVI-C® CONTROLLER is already in configuration state:

- When the network is used for the first time
- After a MOVI-C® CONTROLLER restart if a boot project has not yet been created

In all other cases the MOVI-C® CONTROLLER must first enter configuration state.

Proceed as follows:

✓ The system is free from faults.

✓ Master and slaves are in "operational" ESM state.

1. In MOVISUITE®, open the context menu of the MOVI-C® CONTROLLER with a right mouse click.
2. Select "Activate configuration state".
   ⇨ The safety prompt appears.
3. Confirm the safety prompt to inhibit the drives of the output stage and to activate configuration state.
   ⇨ The window for changing to configuration state opens.
   ⇨ The confirmation message is displayed.
4. Confirm the message.

⇨ The network now is in configuration state. You can add further SBus^{PLUS} stations to the network and set the parameters for auxiliary axes.

25929488/EN – 12/2018

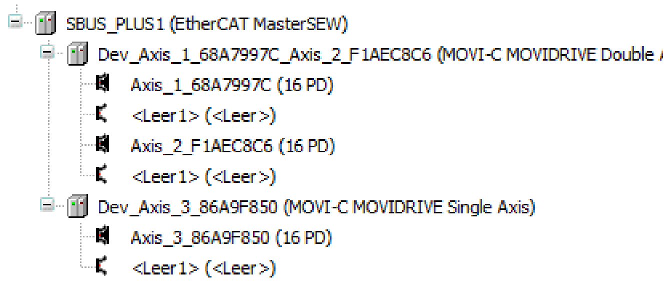
7.3 Creating a new IEC project or updating an existing IEC project

Do the following to create a new IEC project or to add items to an already existing IEC project:

✓ You have already created a project from a network scan in MOVISUITE®. The MOVI-C® CONTROLLER was detected during a network scan and has been applied to MOVISUITE®.

1. In MOVISUITE®, click the MOVI-C® CONTROLLER.
2. From the controller menu, select [IEC software platform].
   ⇨ The window opens for creating or extending an IEC project.
3. To create a new project, click the [Create new IEC project] button.
4. To add items to an already existing project, click the [Update IEC project] button.

⇨ The IEC Editor opens automatically once the IEC project has been created and the code has been generated. The SBus^PLUS string is displayed with the auxiliary axes.



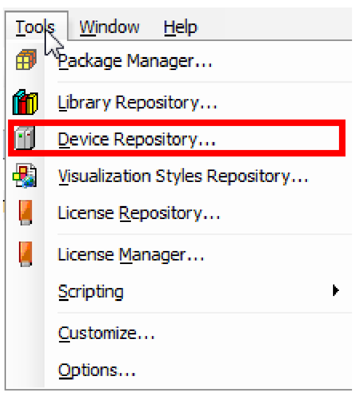
22346586123

7.4 Adding ESI files

To use further SBus^PLUS stations with the functions and setting options, you have to add the respective device description file (ESI file) for third-party components.

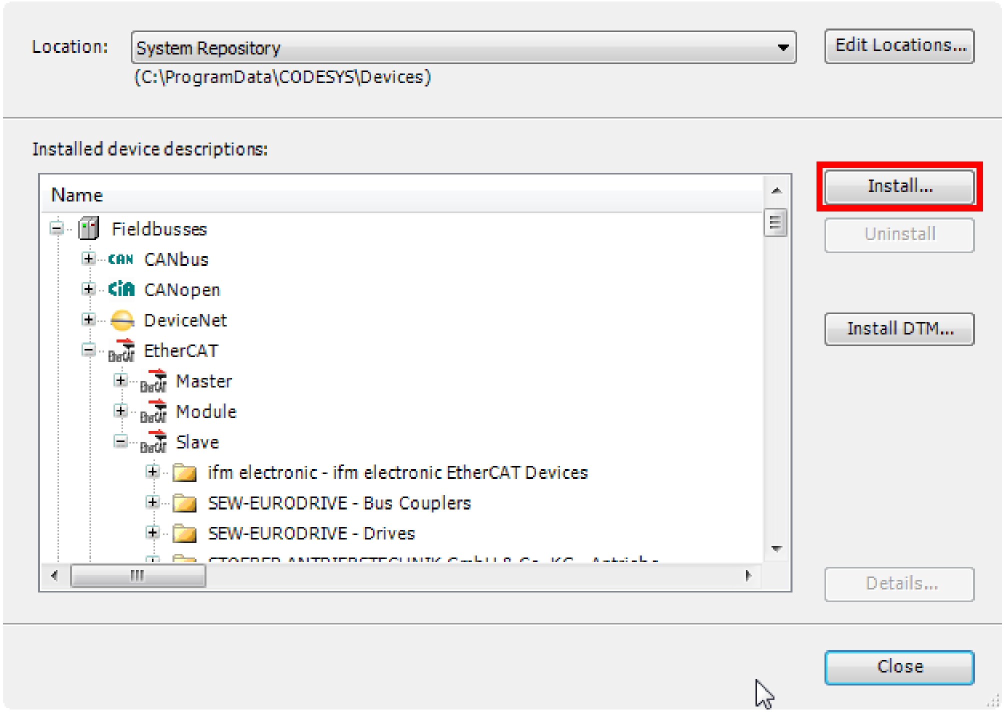
INFORMATION

Device description files for third-party components are available from the respective manufacturer. The latest device description files for SEW components are included in the installation package.



22354849675

1. In the IEC Editor, choose [Tools] > [Device Repository] from the menu.
   ⇨ The "Device Repository" window opens.



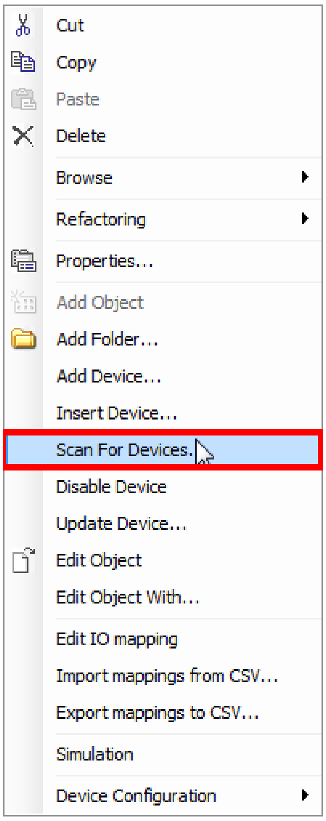
22346589963

2. To install the XML file, click [Install] and navigate to the required device description file.
3. Repeat the process for all further components.

7.5 Adding third-party components

To add further third-party components in addition to the MOVI-C® system components, do the following:

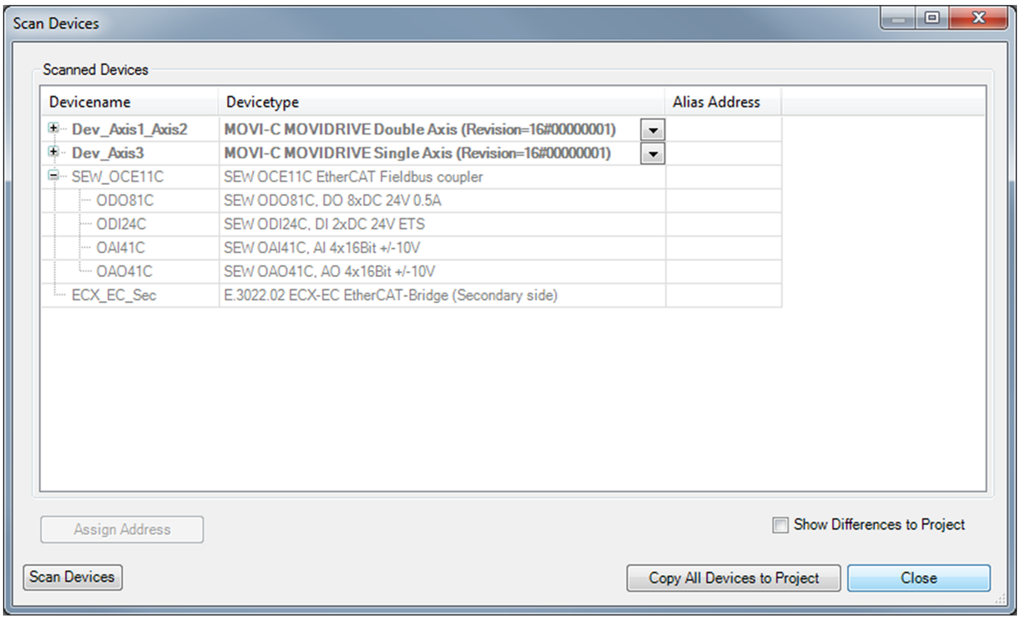
1. In the IEC Editor, right-click to open the context menu of the SBus^PLUS master.



22346591883

2. Choose [Scan For Devices] from the menu.

⇨ A window with the scan result opens.



22347276299

⇨ In the example, the MOVI-PLC® I/O system C and an ESD® bridge are the lower-level devices.

There are 2 ways to copy the scanned components into the IEC project:

INFORMATION

Make sure that you do not copy components that have already been inserted in the project. Else, all project settings you have made will be reset.

7.5.1 Copy and paste

1. Select all third-party components and copy them to the IEC project.
2. Once you have copied third-party components between already entered system components or above them, choose [Update IEC Project] from the MOVISUITE® menu.

7.5.2 Inserting manually

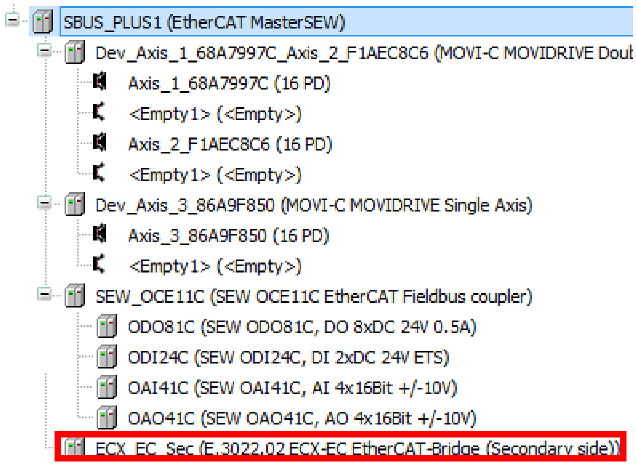
1. In the IEC Editor, right-click to open the context menu of the SBus^{PLUS} master.
2. Choose [Append Device] from the menu.

⇨ A window with available components opens.

22346606603

3. Select the required components.

⇨ Once you have added third-party components, the respective structure nodes are displayed in the IEC Editor under the SBus^{PLUS} master:



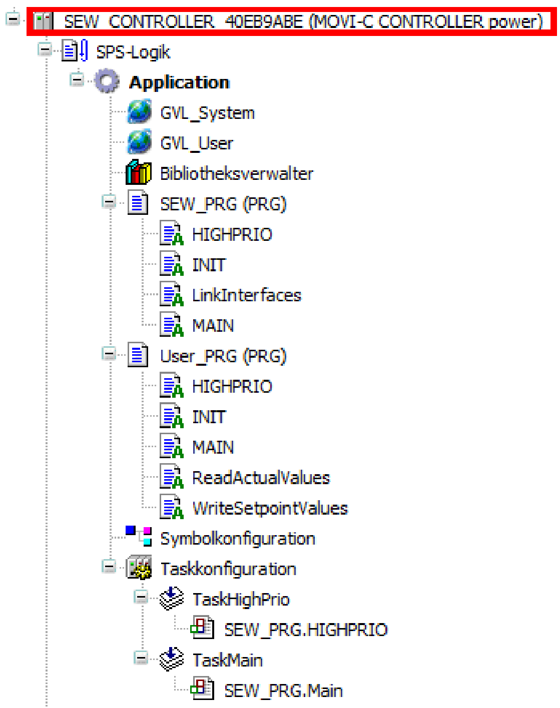
22346608523

INFORMATION

If you have not yet installed the ESI file, a warning will be issued in the IEC Editor during the scan.

7.6 Configuring MOVI-C® CONTROLLER

Do the following to configure the MOVI-C® CONTROLLER:



22346610443

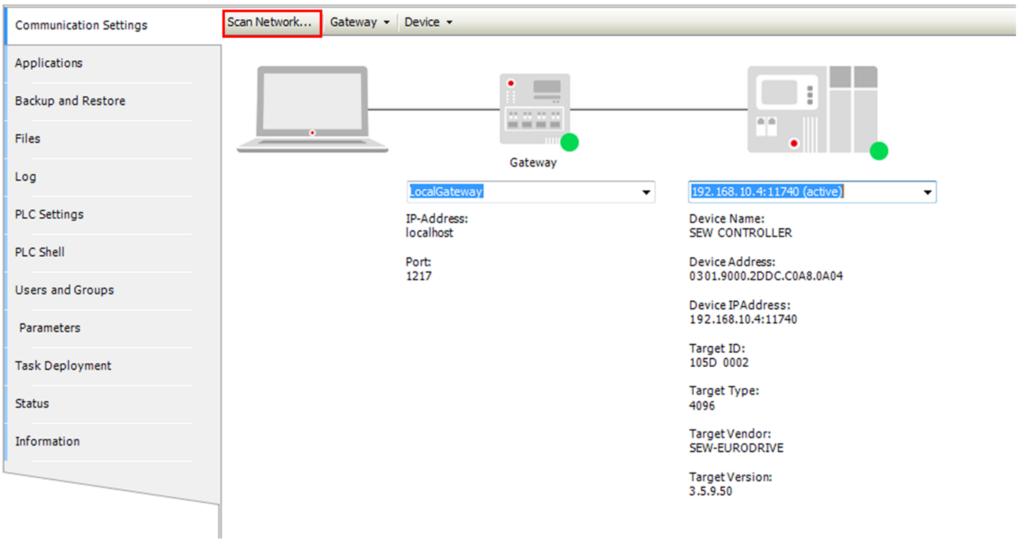
1. In the IEC Editor, double-click the device node of the MOVI-C® CONTROLLER.

   ⇨ The window for configuring the MOVI-C® CONTROLLER opens.

25929488/EN – 12/2018

7.6.1 Communication settings

Under [Communication], you can establish a connection between the MOVI-C® CONTROLLER and the target system.



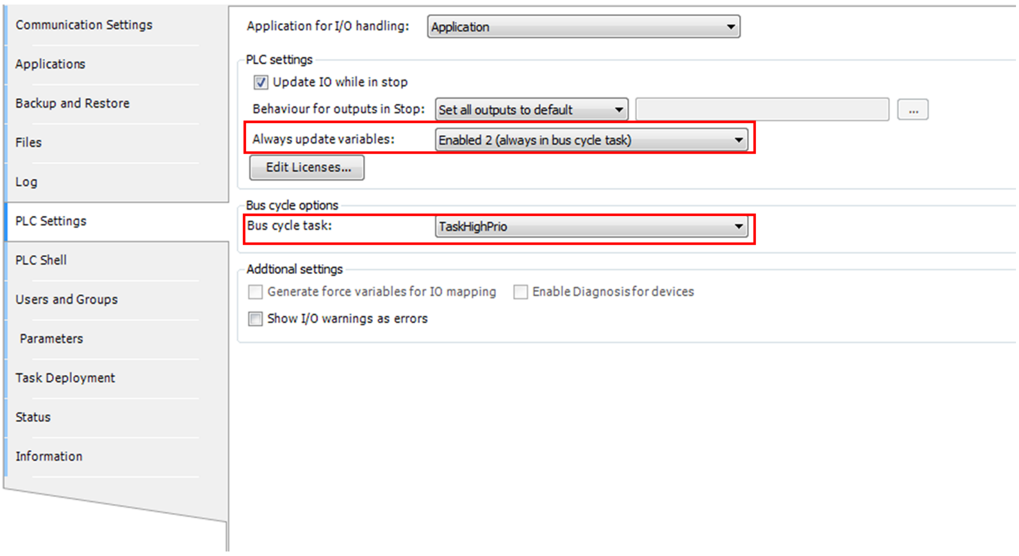
22346612363

1. To establish the connection, click [Scan Network].

Instead, you can enter the IP address of the target device manually.

7.6.2 Settings of the MOVI-C® CONTROLLER

Under [PLC Settings], you can set the behavior of the PLC during runtime and in the event of a fault.



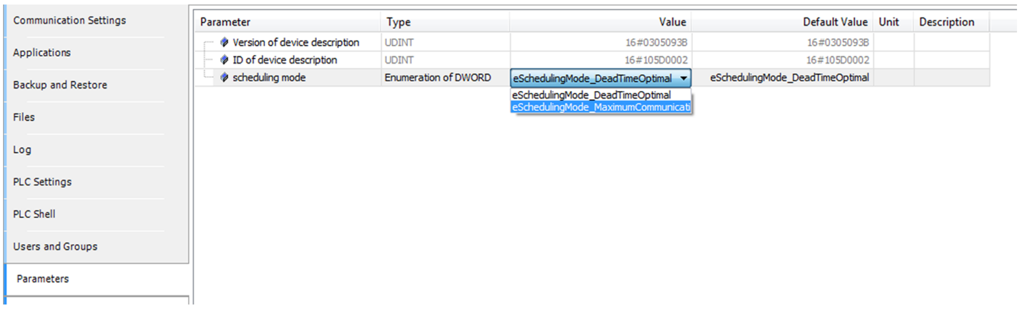
22346614283

INFORMATION

The variables must always be updated in the "Bus cycle task".

7.6.3 Setting scheduling mode

You can set scheduling mode under [Parameters].

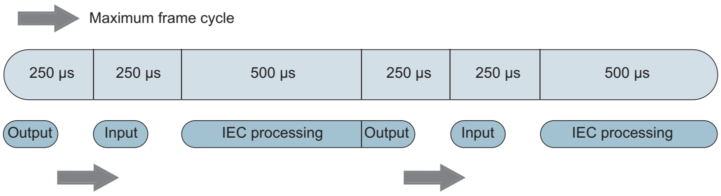


22354853259

The following settings can be made:

- **Dead-time optimized mode**

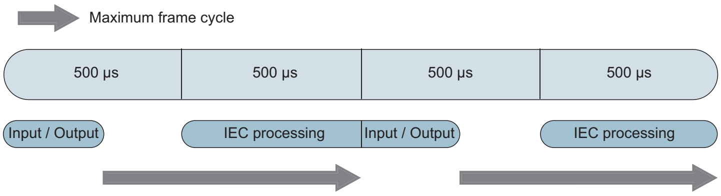
  Once the outputs of the process image have been written to the EtherCAT® frame, the inputs from the EtherCAT® frames are copied to the next PLC slot. The current data of theEtherCAT® frame are used for this purpose. This mode is suited for highly dynamic processes and is set by default.



22302525707

- **Station-optimized mode**

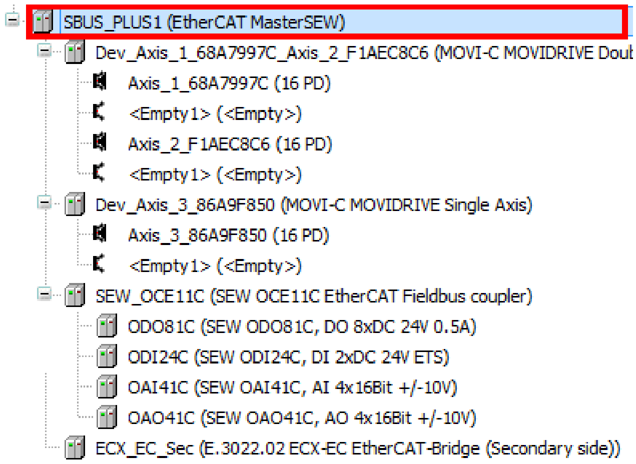
  Systems with a great number of stations result in a great amount of process data and consequently long EtherCAT® frames. Each frame needs time to receive and send a frame. In systems with many stations, this time has a significant impact. This is the reason why station-optimized mode is required for such systems.



22302523275

7.7 Configuring the master

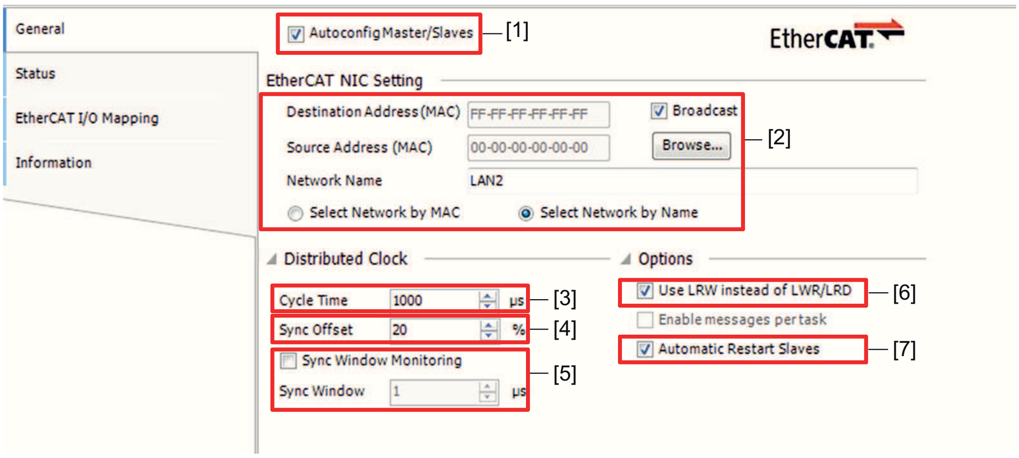
Do the following to configure the SBus^PLUS master:



22346616203

1. In the IEC Editor, double-click the device node of the SBus^PLUS master.
   ⇨ The window for configuring the master opens.

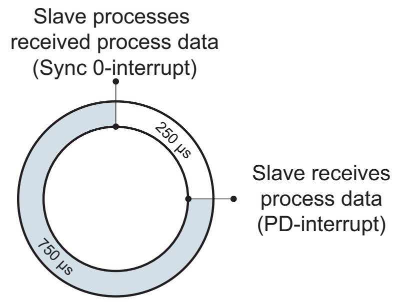
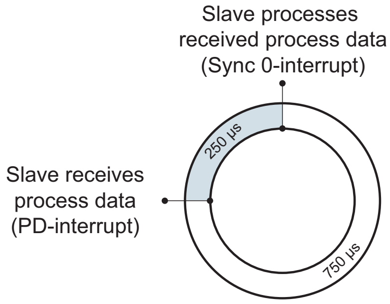
To make general settings for the master, choose [General] from the menu.



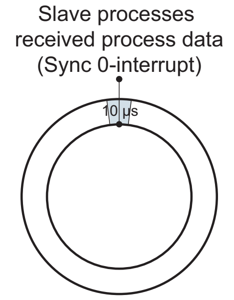
22346618123

No.	Description
[1]	The check box must always be enabled. Slave addresses cannot be set manually.
[2]	This section contains preconfigured settings for the EtherCAT® network. Do not make any changes to these settings.
[3]	In this drop-down list, you set the cycle time of the EtherCAT® frame. The cycle time is a whole-numbered multiple of 1 ms and has to be equal to the associated IEC bus cycle task.

25929488/EN – 12/2018

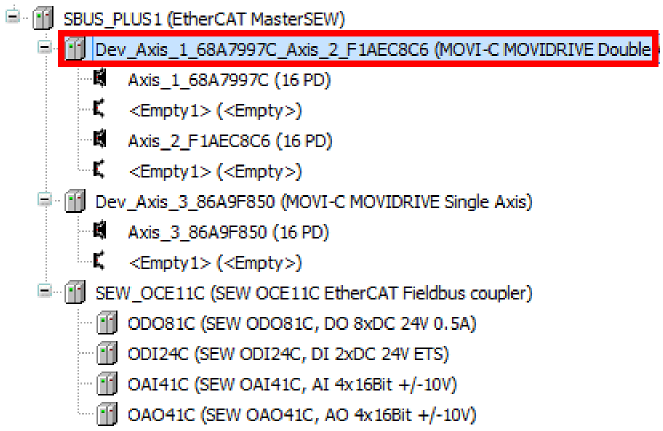
No.	Description
[4]	In the "Sync Offset" choice box, you can set the time difference between the receipt of process data of the EtherCAT® in the slave and the processing of process data. The "Sync Offset" compensates the time differences when the frame receives the process data and results in slight clock variations. The sync offset is indicated in percent and refers to the set cycle time. Examples: • When the sync offset is set to 25% and the cycle time is 1000 µs, the process data are processed after 750 µs after receipt in the slave.  *22302696971* • When the sync offset is set to -25% and the cycle time is 1000 µs, the process data are processed after 250 µs after receipt in the slave.  *22302699403* **INFORMATION!** A negative sync offset near the "Sync0-interrupt" will cause an error in process data processing. Bear in mind the runtime of EtherCAT® frames for your calculations.

25929488/EN – 12/2018

No.	Description
[5]	When the check box is enabled, the clock variations in the slaves are monitored when the process data of the EtherCAT® frames are processed. "Sync Window" refers to the set cycle time of the master. The default value for "Sync Window" is 10 µs based on a cycle time of 1 ms. Example:  *22302694539* **INFORMATION!** Booting of the network takes longer when setting "Sync Window" to smaller values.
[6]	When the check box is enabled, the LRW command will be used. This setting reduces the assigned process data length. Some slaves do not support LRW commands but only LWR/LRD commands. Disable the check box in this case. Refer to the ESI file for information whether a slave supports the LRW command or not.
[7]	When the check box is enabled, the master attempts to switch failed slaves in the system to "operational" ESM state. When the check box is disabled, the slaves remain in the following ESM state after failure: • "Safe operational" (after communication failure) • "Init" (after power failure)

7.8 Configuring slaves

Do the following to configure slaves:

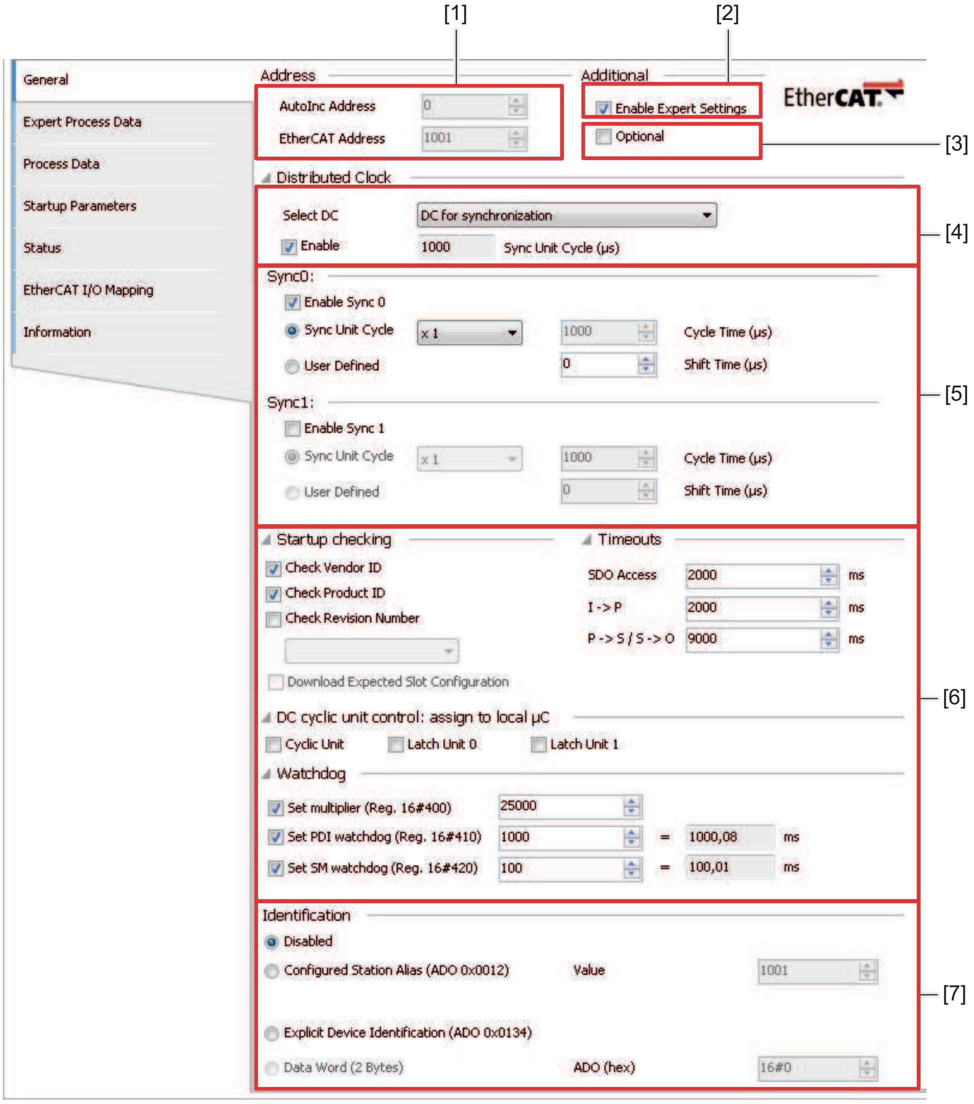


22346620555

1. In the IEC Editor, double-click the device node of the required slave.
   ⇨ The window for configuring the slave opens.

25929488/EN – 12/2018

To make general settings for the slave, choose [General] from the menu.



9007221601363467

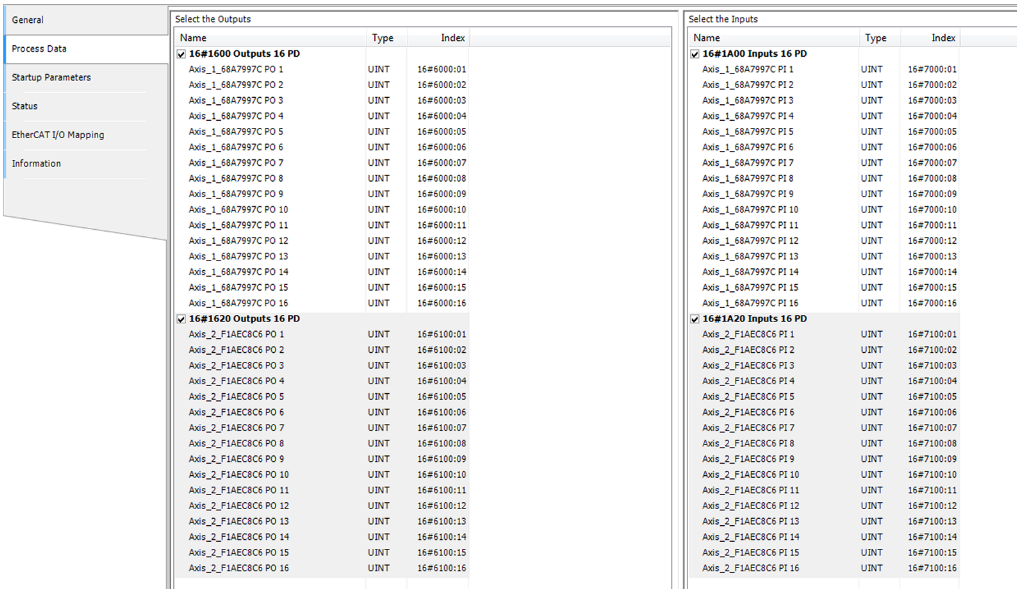
No.	Description
[1]	These two fields display the slave addresses in configuration state and in operational state. They cannot be edited.
[2]	Enabling the "Enable Expert Settings" check box displays additional configuration options. This includes the selection of the device identification, for example.
[3]	Enabling the "Optional" check box displays additional configuration options for using an optional slave. Optional slaves can only be used if also a device identification is selected.
[4]	The master provides the DC function as soon as DC is active for at least one slave. If the slave supports DC, enable the "Enable" check box. Set the "Sync Unit Cycle" to the cycle time of the master.
[5]	In this section, you can set the factor between the cycle time of the frames (sync unit) and the DC cycle. These settings have been adjusted to match the master.

No.	Description
[6]	The expert settings have been adjusted to match the master and may only be changed upon consultation with SEW-EURODRIVE.
[7]	If the slave supports the "Station Alias" or "Explicit Device Identification" function, select one of these functions for enabling device identification and additionally the optional slaves. To do so, select one of the two functions and enter the respective value under "Value".

7.8.1 Configuring process data

The [Process Data] menu item provides an overview of the configuration and assignment of process data objects (PDO) of the slaves. The master sends this information to the slave via CoE service when starting.

For many slaves, the configuration of process data is already pre-assigned by the device description file. This means the process data objects need not be configured or assigned.



22346624907

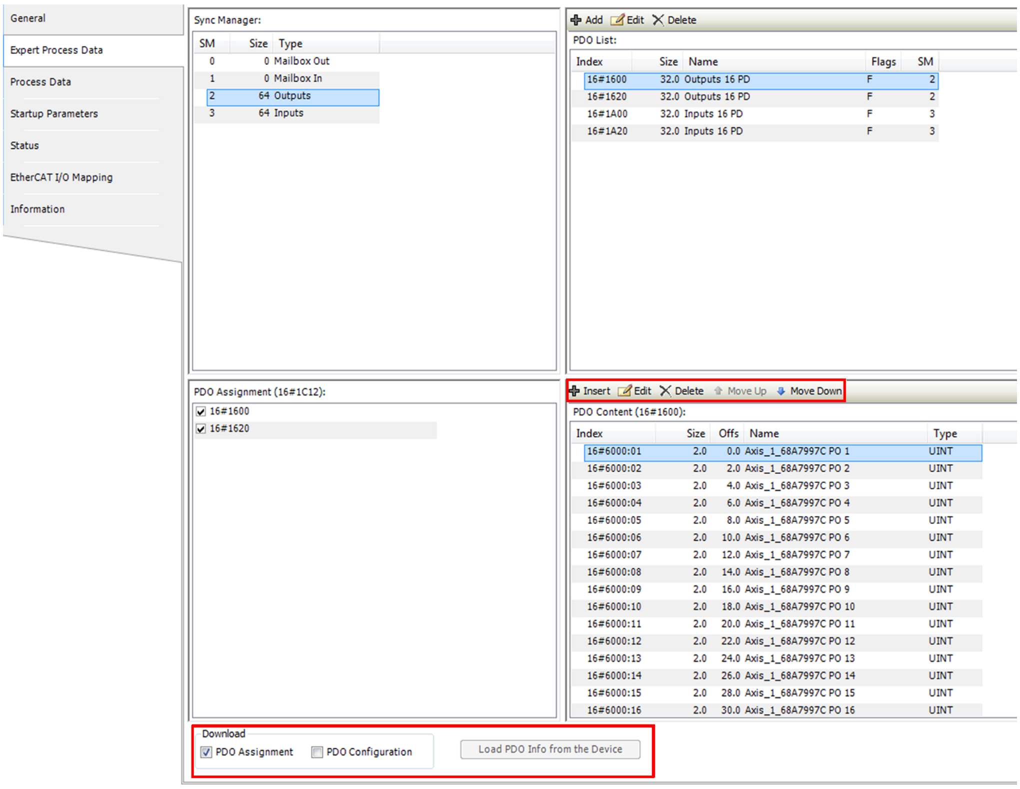
Process data expert settings

An application-specific configuration and assignment of PDOs is useful for gateways and bus couplers. The slaves are configured using a special software of the respective manufacturer.

Under [General], you can access the PDO content for the master. For this purpose, the "Activate Expert Settings" check box must be enabled.

Observe the following notes:

- Adhere to the manufacturer's documentation when configuring the PDOs.
- Instead of configuring PDO: You can adjust the device description file in the IEC Editor. Observe the manufacturer's documentation.
- The configuration of PDOs in the manufacturer's documentation and the configuration in the IEC Editor must match.



22346626827

You can add, remove, or edit PDO objects via menu bar.

To load the configuration data into the slave controller, do the following:

1. Enable the two check boxes "PDO Assignment" and "PDO Configuration".
   ⇨ The data are loaded to the slave controller once you have restarted the master.

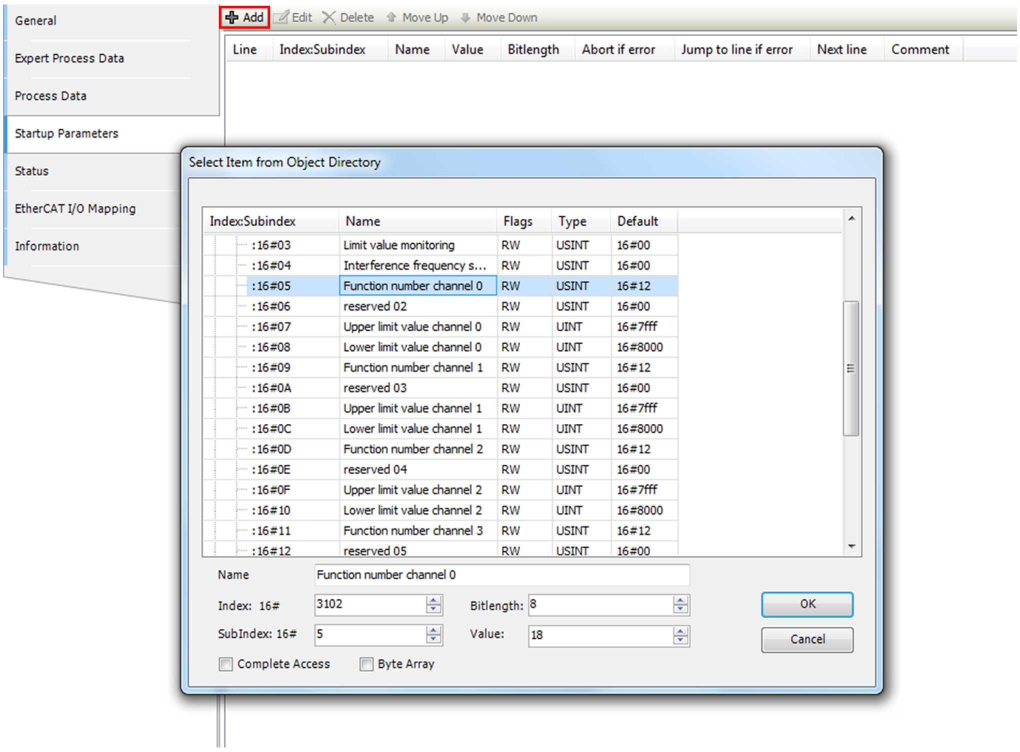
INFORMATION

The [Load PDO Info from the Device] button allows for automatically uploading the configured PDO objects from the slave. This means the PDOs need not be configured manually in the IEC Editor.

25929488/EN – 12/2018

7.8.2 Using startup parameters

Some slaves have configurable parameters (ServiceDataObjects), such as configurable IO modules on a bus coupler. You can select the parameters under [Startup Parameters] menu by clicking "+Add" and adjust them accordingly.



22346628747

Example of adjusting the parameter for the analog input module OAI41C of the MOVI-PLC® I/O system C:

Line	Index:Subindex	Name	Value	Bitlength	Abort if error	Jump to line if error	Next line	Comment
1	16#3104:16#08	Functi...	80	8	☐	☐	0	
2	16#3104:16#07	Functi...	80	8	☐	☐	0	
3	16#3104:16#06	Functi...	80	8	☐	☐	0	
4	16#3104:16#05	Functi...	80	8	☐	☐	0	

22346630667

The current parameters are loaded into the slave next time the network is booted. If the parameters are not adjusted, the MOVI-C® CONTROLLER will load the default values into the slave.

INFORMATION

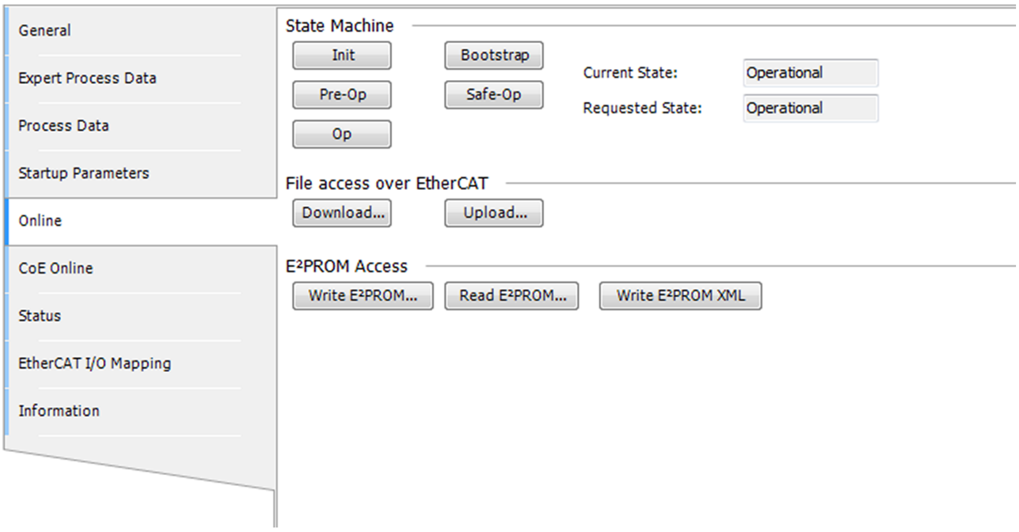
Observe the manufacturer's documentation when selecting and setting the parameters.

7.8.3 Online functions

The expert mode of the slaves must be enabled to being able to use the online functions.

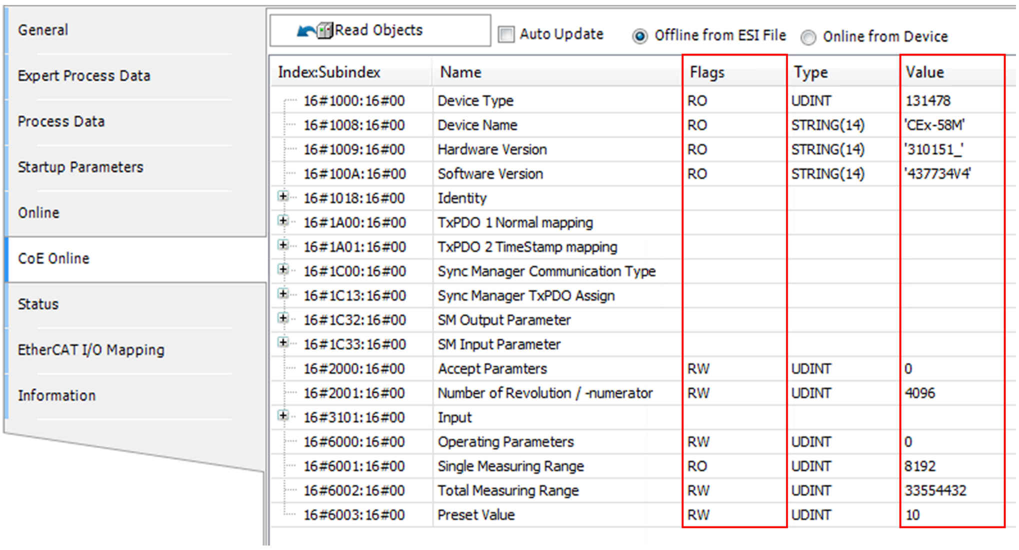
On the "Online" tab, you can perform the following actions:

- The the "State Machine" section, you can set the states of individual EtherCAT® slaves irrespective of the EtherCAT® master. The "Bootstrap" state is not implemented.
- In the "EEPROM Access" section, you can write SII files into the device retrospectively or read them from the device.



22346640779

On the "CoE Online" tab, you can only read (RO) or read/write (RW) the value of the SDOs during runtime. A connection to the IEC project can be made using logical devices.



Index:Subindex	Name	Flags	Type	Value
16#1000:16#00	Device Type	RO	UDINT	131478
16#1008:16#00	Device Name	RO	STRING(14)	'CEx-58M'
16#1009:16#00	Hardware Version	RO	STRING(14)	'310151_'
16#100A:16#00	Software Version	RO	STRING(14)	'437734V4'
16#1018:16#00	Identity			
16#1A00:16#00	TxPDO 1 Normal mapping			
16#1A01:16#00	TxPDO 2 TimeStamp mapping			
16#1C00:16#00	Sync Manager Communication Type			
16#1C13:16#00	Sync Manager TxPDO Assign			
16#1C32:16#00	SM Output Parameter			
16#1C33:16#00	SM Input Parameter			
16#2000:16#00	Accept Paramters	RW	UDINT	0
16#2001:16#00	Number of Revolution / -numerator	RW	UDINT	4096
16#3101:16#00	Input			
16#6000:16#00	Operating Parameters	RW	UDINT	0
16#6001:16#00	Single Measuring Range	RO	UDINT	8192
16#6002:16#00	Total Measuring Range	RW	UDINT	33554432
16#6003:16#00	Preset Value	RW	UDINT	10

9007221601383691

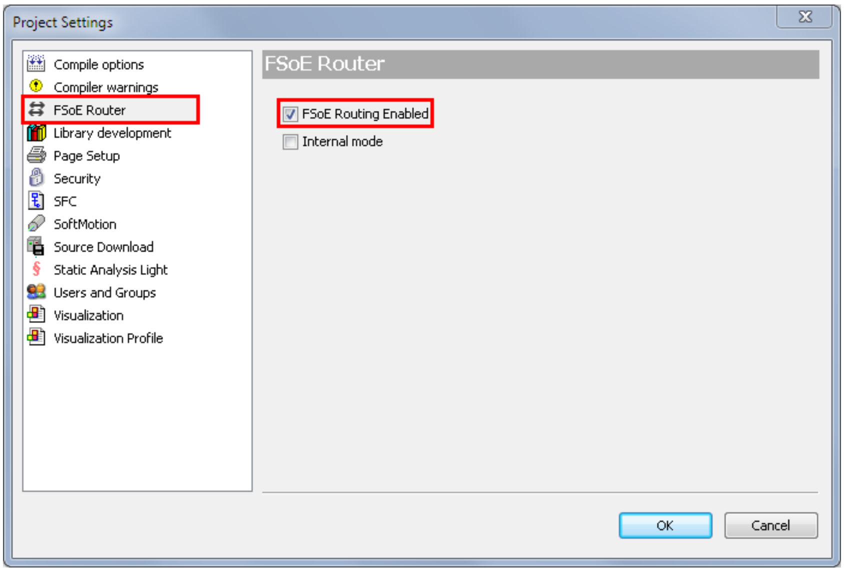
25929488/EN – 12/2018

7.9 Enabling FSoE routing

It is possible to set up routing of a safe FSoE frame on the MOVI-C® CONTROLLER. This is the basis for connecting a 3rd party FSoE master to the SBusPLUS system bus and for using the safe protocol FSoE (Fail Safe over EtherCAT®). The FSoE frame of the EtherCAT® is embedded into the cyclical process data during this process.

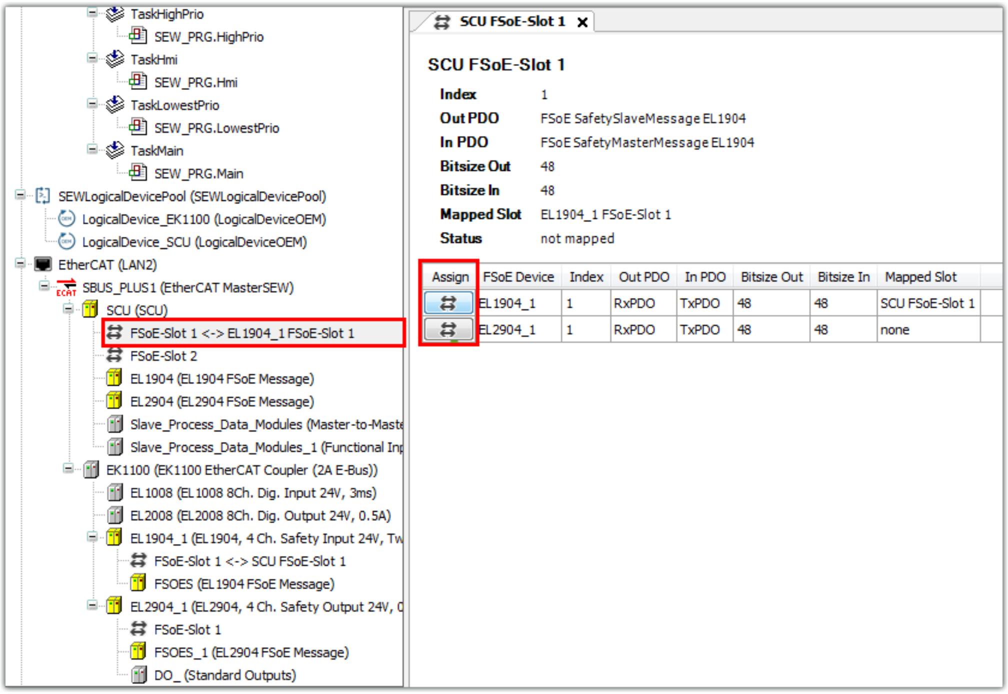
Proceed as follows:

1. In the [Project] > [Project Settings] menu, go to the "FsoE Router" tab to enable the FSoE routing function.



24856435467

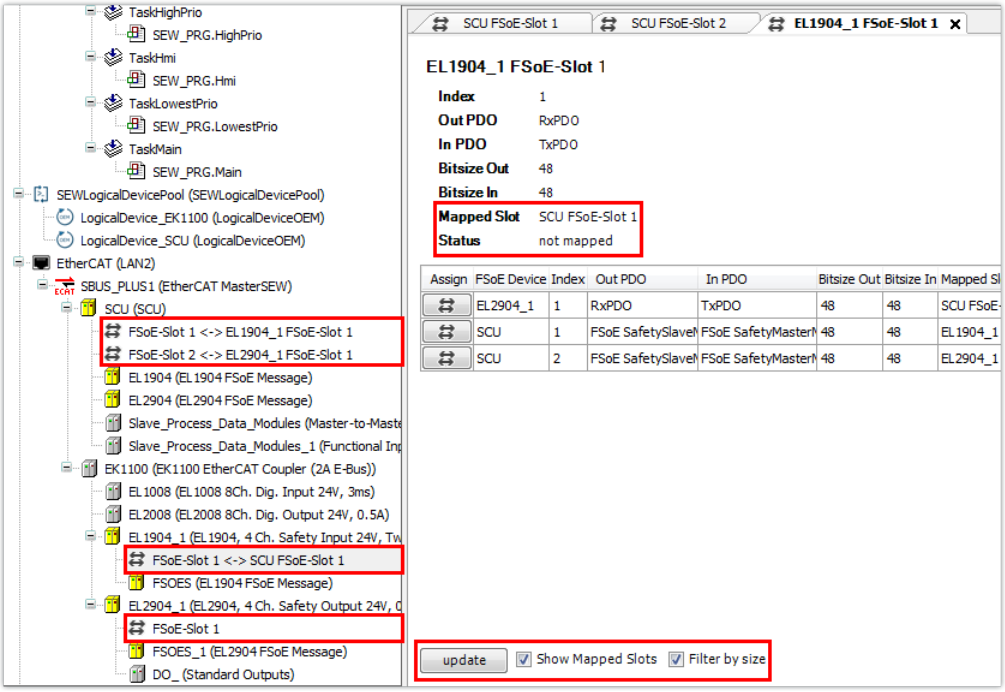
2. Use the associated "Assign" function to connect the FSoE slots of the devices with each other. Observe the correct order of the devices and the associated slots.



24860266891

25929488/EN – 12/2018

3. Check your configuration using "Mapped Slot". The status only changes after the code is translated again. The status of the respective FSoE slots changes to "mapped" when the routing function is used correctly. Check the configured devices and perform a functional test of the components after the network has booted.



24860262539

7.10 Starting MOVI-C® CONTROLLER: booting the network

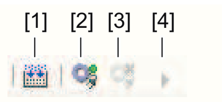
You can now start the MOVI-C® CONTROLLER:

✓ The master and slaves have been configured.

✓ The project planning in the IEC project matches the real system.

1. To compile the IEC project into machine code, press the <F11> key in the IEC Editor.
2. To log in, press the keys <Alt> + <M>.
3. To start the MOVI-C® CONTROLLER, press the <F5> key.

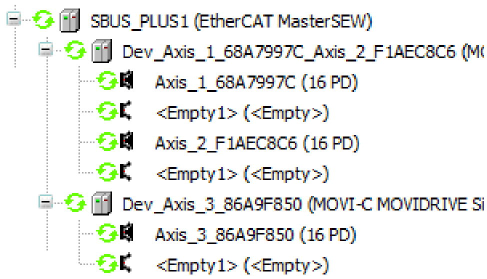
Instead of pressing the keys, you can click the respective symbols in the menu bar of the IEC Editor.



22346632587

[1]	Compile IEC project in machine code
[2]	Log in
[3]	Log off
[4]	Starting MOVI-C® CONTROLLER

When no faults are present in the system, the ESM changes to "operational" ESM state once the network has booted. The devices in the device tree are marked by a green symbol.



22346635019

7.11 Deactivating slaves

Already projected slaves can be deactivated in IEC projects.

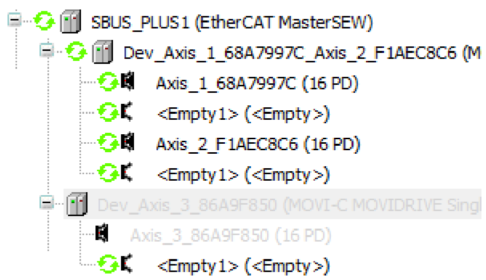
INFORMATION

Deactivated slaves must no longer be integrated in the network in terms of hardware.

Do the following to deactivate the slave:

1. In the IEC Editor, right-click to open the context menu of the slave.
2. Choose "Deactivate Device".

⇨ Deactivated slaves will no longer be taken into account when booting the network next time.

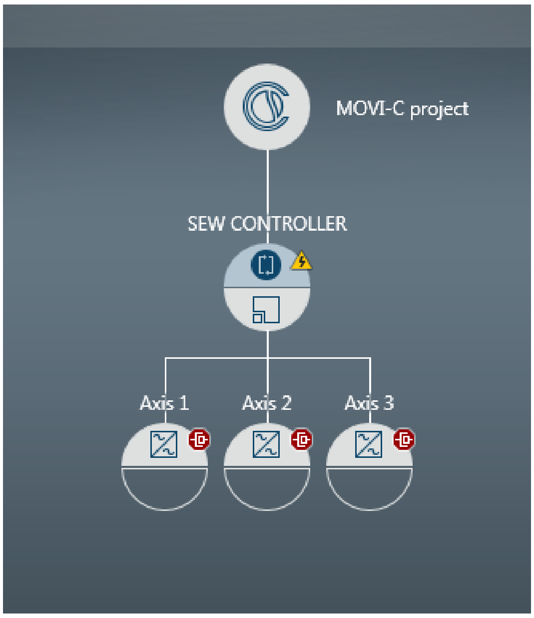


22346638859

25929488/EN – 12/2018

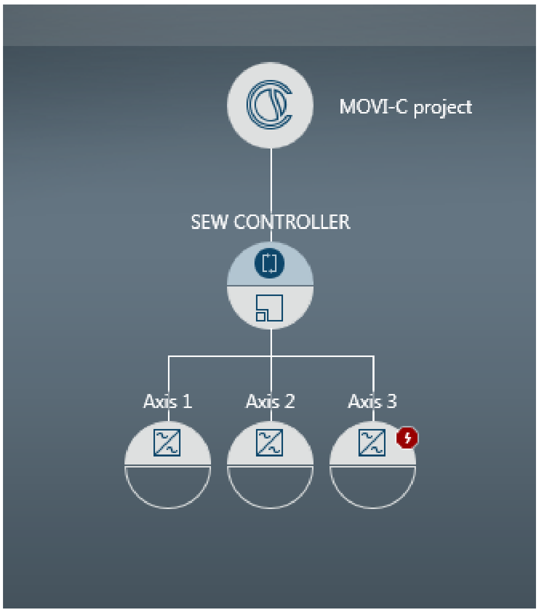
7.12 Updating the network in MOVISUITE®

When configuration state is active, you have to scan the axes again because of the changed address.



22346644619

After the network scan in MOVISUITE®, the lower-level devices are available in "operational" ESM state.



22346646539

INFORMATION

Third-party components are not displayed in the current version of MOVISUITE® .

25929488/EN – 12/2018

8 Diagnostics

Information between SBus^{PLUS} master and connected slaves is exchanged by means of Ethernet frames. The master sends the frames, which cyclically pass the individual slaves. The frame returns to the master during the same cycle due to the ring structure. The data are then processed cyclically and evaluated by the master.

EtherCAT® offers a great number of diagnostics options. These can be divided as follows:

- Information at hardware level
- Information at software level
- Diagnostics through the slave
- Diagnostics through the master

The next chapters describe the most important diagnostics options.

8.1 Integrating the library

The SEW IoDrvEtherCAT library is available when installing the power packages for the MOVI-C® CONTROLLER.

All the objects and variables required for diagnostics are offered in the library.

8.2 Connector status

The following driver settings must be set both for the master and for all slaves of the network:

Connector Status PLC in stop	
MOVI-C CONTROLLER LAN2 *	ModuleWorking
EtherCAT®	ModuleEnabled, DriveFound, ModuleFound, ModuleConfigured

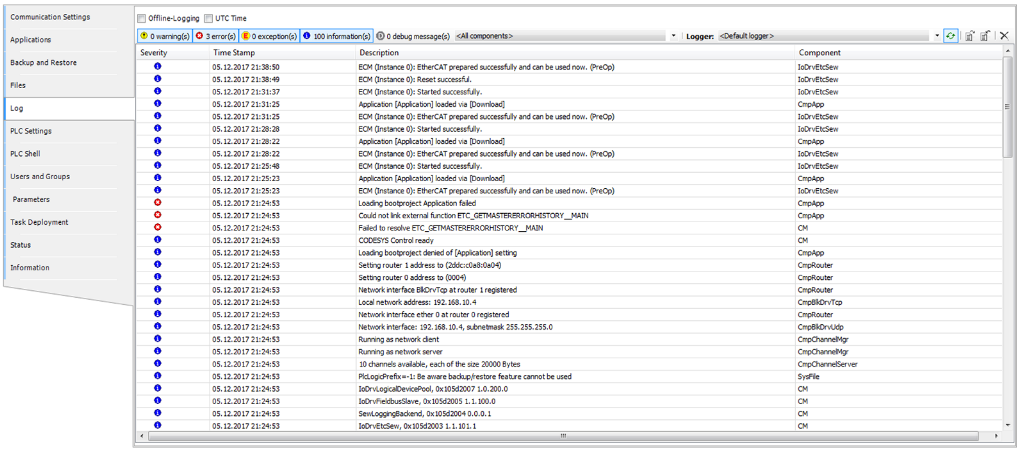
Connector Status PLC in run	
MOVI-C CONTROLLER LAN2 *	ModuleWorking
EtherCAT®	ModuleWorking
* Is only displayed for the master.	

You find the settings for the master and the slaves under the [Status] menu item.

If the driver settings differ, you have to adjust the project to the current firmware of the MOVI-C® CONTROLLER. For more information, refer to the MOVISUITE® manual.

25929488/EN – 12/2018

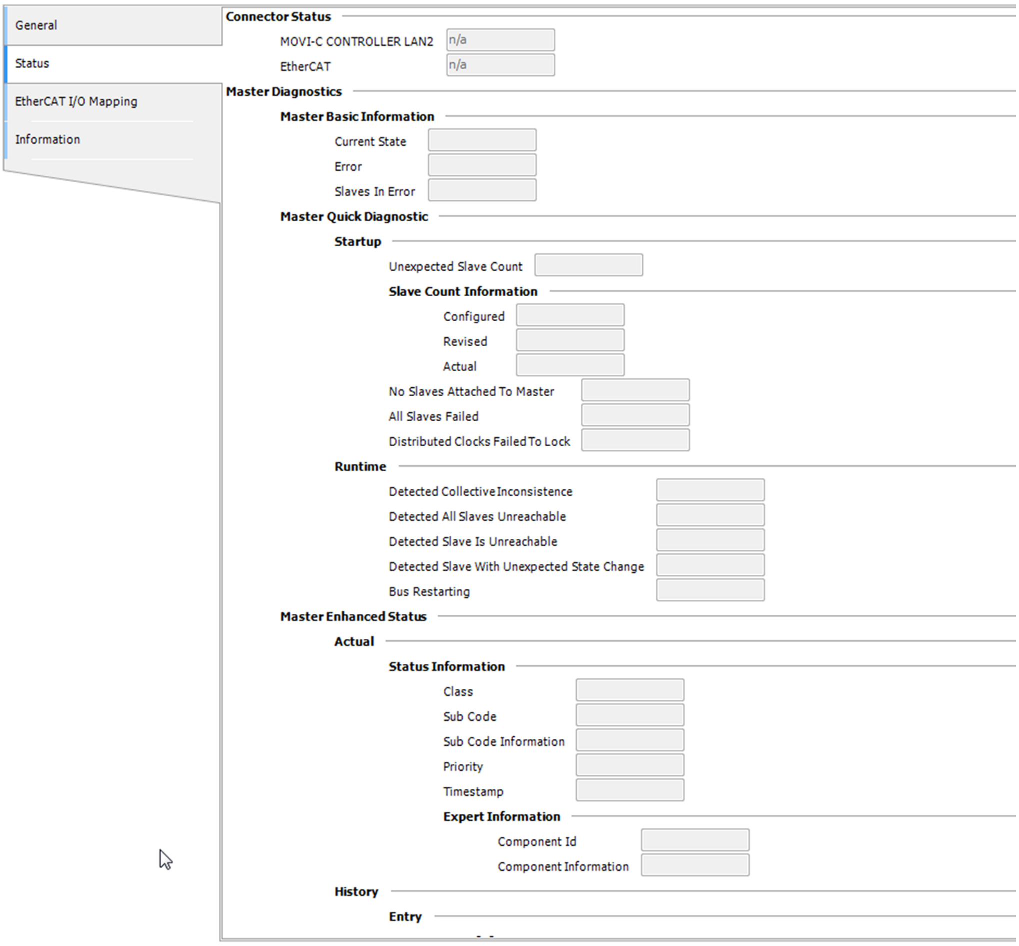
The log file of the MOVI-C® CONTROLLER provides more detailed information.



22346648459

8.3 Master diagnostics

All information available for the master is listed in the "Master Diagnostics" section under the [Status] menu item.



22354856843

The IEC project additionally offers quick diagnostics of the ESM state of the stations. The following table shows the possible states of the master in the network:

Master	ESM state	Description
	Operational	System without faults.
	Pre-operational	After PLC reset.
	Operational	The length of the configured process data differs from the hardware setup; at least one slave is not in "operational" ESM state.
	Init	Deviation between the number of projected devices and hardware setup.

8.3.1 Master Basic Information

Basic information about the ESM state of the master is available in the IEC project under "Current State".

The master can be in one of the following states:

ESM state 8	Operational
ESM state 4	Safe operational
ESM state 2	Pre-operational
ESM state 1	Init

The ESM state of the master is at the same time the target state of the slave in the connected network.

For information on whether a network error is present and how many slaves are affected by the error, refer to the following:

Error	Global error flag: Indicates a network error.
Slaves In Error	Number of invalid slaves in the network.

25929488/EN – 12/2018

8.3.2 Master Quick Diagnostic

The "Master Quick Diagnostic" structure provides possible causes for an error determined by the master.

Startup

The "Startup" structure provides causes of errors after failed bus startup:

Unexpected Slave Count	Unexpected Slave Count = true The number of slaves detected on the bus differs from the number of configured slaves.
Slave Count Information	The structure contains information abut the number of configured slaves detected on the bus.
Configured	Number of configured slaves.
Revised	Number of configured slaves minus optional slaves that are not present.
Actual	Number of slaves detected on the bus.
No Slaves Attached To Master	No slaves connected to the bus.
All Slaves Failed	All Slaves Failed = true Bus startup failed for all slaves. No slave has reached the ESM state requested by the master.
Distributed Clocks Failed To Lock	Distributed Clocks Failed To Lock = true The master failed to synchronize the distributed clocks (DC) of the slaves.

Runtime

The "Runtime" structure provides possible error causes during ongoing bus operation:

Detected Collective Inconsistence	Detected Collective Inconsistence = true At least one slave is not in the bus state specified by the master.
Detected All Slaves Unreachable	Detected All Slaves Unreachable = true The master does not reach a slave on the bus.
Detected Slave Is Unreachable	Contains a diagnostics flag for each configured slave. Detected Slave Is Unreachable = true The master cannot reach the corresponding slave on the bus.
Detected Slave With Unexpected State Change	Contains a diagnostics flag for each configured slave. Detected Slave With Unexpected State Change = true The master has detected a bus state for the corresponding slave that differs from the specified bus state.
Bus Restarting	Bus Restarting = true Automatic restart of all slaves.

25929488/EN – 12/2018

8.3.3 Master Enhanced Status

The "Master Enhanced Status" structure provides expert information about the current internal state of the master.

The "Actual" structure provides information about the current internal state of the master:

Actual

Status Information	The structure provides information about the current internal state of the master.
Class	Classifies the internal state of the master: • 0x08: no error • 0x04: Error
Subcode	Detailed classification of the master state.
Subcode Information	Further information on the subcode of the master state.
Priority	Priority of the master state.
Timestamp	The timestamp indicates the time when the current state of the master has been set.
Expert Information	Structure with expert information on a component involved in an error state.
Component ID	Internal ID of the software component.
Component Information	Further component-specific diagnostics information.

The structures "History" and "Entry" provide information abut the latest error states of the master.

25929488/EN – 12/2018

8.3.4 Master Enhanced Network Diagnostic

The "Master Enhanced Network Diagnostic" structure provides information on the network state and the events that occurred in the network.

Bus Load

The "Bus Load" structure provides information on the current bus load in the network.

Cyclic Traffic	Traffic of the cyclic communication channel in [bytes/second].
Acyclic Traffic	Traffic of the acyclic communication channel in [bytes/second].

Network Diagnostic Counter

The "Network Diagnostic Counter" structure combines the diagnostic counters for events that occurred in the network.

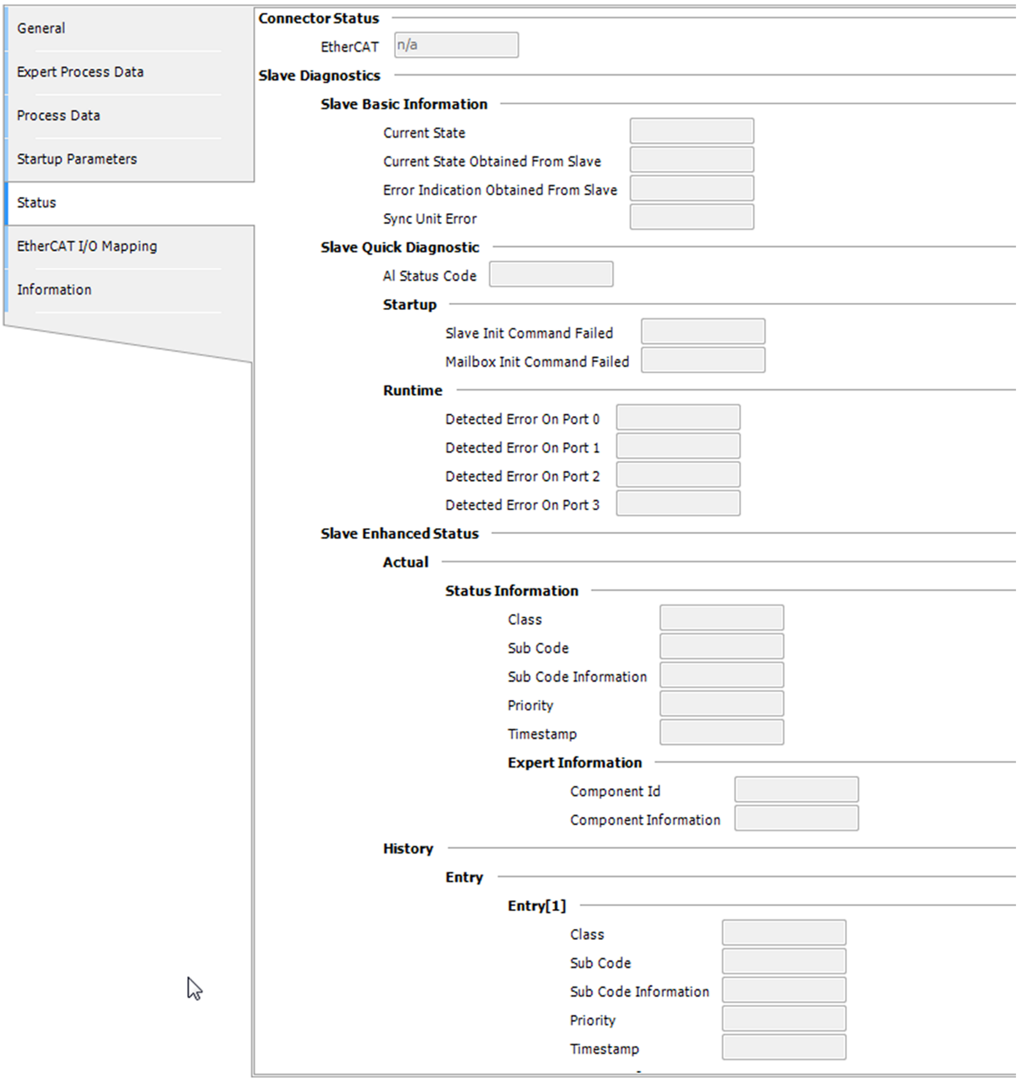
Detected Collective Inconsistence	This counter is incremented when the collective state differs from the expected value.
Detected All Slaves Unreachable	This counter is incremented when no slave can be reached anymore.
Detected Slave is Unreachable	This counter is incremented when a slave was detected as being unreachable.
Detected Slave Is Reachable Again	This counter is incremented when an unreachable slave can be reached again.
Detected Slave With Unexpected State Change	This counter is incremented when an unexpected ESM state change of a slave was detected.
Bus Restarts	This counter is incremented when the complete bus was restarted for re-integration due to activated DCs.
Successful Bus Restarts	This counter is incremented when a bus was restarted successfully.
Failed Bus Restarts	This counter is incremented when a bus was restarted with an error.
Started Reintegrations	This counter is incremented when re-integration was started for a slave.
Successful Reintegrations	This counter is incremented when a started re-integration was completed successfully.
Failed Reintegrations	This counter is incremented when a started re-integration was completed with an error.
Failed Error Counter Detection	This counter is incremented when the error counters of a slave could not be read.
Detected Acyclic Frame Receive Timeouts	This counter is incremented when a "receive timeout" for an acyclic frame was detected.
Released Acyclic Frame Receive Timeouts	This counter is incremented when an acyclic frame was received again after a timeout.
Detected Cyclic Frame Receive Timeouts	This counter is incremented when a "receive timeout" was detected for a cyclic frame.

25929488/EN – 12/2018

Released Cyclic Frame Receive Timeouts	This counter is incremented when a cyclic frame was received again after a timeout.
Detected Send Frame Errors	This counter is incremented when a send error was detected in the hardware abstraction layer (HAL).
Released Send Frame Errors	This counter is incremented when a pending send error was canceled in the hardware abstraction layer (HAL).
Detected Output Image Begin Read Errors	This counter is incremented when an error is detected at the start of a read operation on the output image.
Detected Output Image End Read Error	This counter is incremented when an error is detected at the end of a read operation on the output image.
Detected Input Image Begin Write Errors	This counter is incremented when an error is detected at the start of a read operation on the input image.
Detected Input Image End Write Errors	This counter is incremented when an error is detected at the end of a read operation on the input image.
Released Output Image Begin Read Errors	This counter is incremented when an error present at the start of a read operation was canceled on the output image.
Released Output Image End Read Error	This counter is incremented when an error present at the end of a read operation was canceled on the output image.
Released Input Image Begin Write Errors	This counter is incremented when an error present at the start of a write operation was canceled on the input image.
Released Input Image End Write Errors	This counter is incremented when an error present at the end of a write operation was canceled on the input image.
Detected Sync Loss	This counter is incremented when a loss of synchronization between master and slave was detected.
Reestablished Sync	This counter is incremented when a lost synchronization is re-established between master and slave.
Detected Slave Sync Errors	This counter is incremented when a "SyncError" flag set in a flag is detected again.
Released Slave Sync Errors	This counter is incremented when a "SyncError" flag set in a slave was canceled.

8.4 Slave diagnostics

You find all the information of a slave for the network under "Slave Diagnostics" of the [Status] menu item.



22354860427

The IEC project additionally offers quick diagnostics of the state of the stations. The following table shows the possible states of the slaves in the network:

Slave	Description
	Fault-free slave in "operational" ESM state.
	The slave is not in "operational" ESM state.
	The slave signals a fault; observe AI status code of the slave.

8.4.1 Basic Slave Information

Basic information about the ESM state of a salve is available in the IEC project under "Current State".

The slave can be in one of the following states:

ESM state 8	Operational
ESM state 4	Safe operational
ESM state 2	Pre-operational
ESM state 1	Init

Additional important notes provide further information:

Current State Obtained From Slave	Current State Obtained From Slave = true The master could determine the current bus state.
Error Indication Obtained From Slave	Error Indication Obtained From Slave = true The slave indicates an error via the AL status register.
Sync Unit Error	Sync Unit Error = true The process data received by the slave are invalid. The slave is part of a sync unit. The process data of this sync unit was not received correctly.

8.4.2 Slave Quick Diagnostic

The "Slave Quick Diagnostic" structure provides possible causes provided by the affected slave for errors that occur.

Startup

The "Startup" structure provides causes of errors after failed bus startup:

Slave Init Command Failed	Slave Init Command Failed = true An initialization command required for bus startup could not be executed properly.
Mailbox Init Command Failed	Mailbox Init Command Failed = true A mailbox initialization command required for bus startup could not be executed properly.

Runtime

The "Runtime" structure provides possible error causes during ongoing bus operation:

Detected Error On Port 0	Detected Error On Port 0 = true Error detected at port 0 of the slave, see "Port Error Counter", chapter "Slave Enhanced Network Diagnostic" (→ 🗎 64).
Detected Error On Port 1	Detected Error On Port 1 = true Error detected at port 1 of the slave, see "Port Error Counter", chapter "Slave Enhanced Network Diagnostic" (→ 🗎 64).
Detected Error On Port 2	Detected Error On Port 2 = true Error detected at port 2 of the slave, see "Port Error Counter", chapter "Slave Enhanced Network Diagnostic" (→ 🗎 64).
Detected Error On Port 3	Detected Error On Port 3 = true Error detected at port 3 of the slave, see "Port Error Counter", chapter "Slave Enhanced Network Diagnostic" (→ 🗎 64).

AL Status Code

To diagnose the software in the slave, the master provides the AL status code. A distinction is made between errors that occur during startup and errors that occur during operation. The following table lists the AL status codes and provides information on how to remedy the error:

Code	Description	Operating phase	Measure
0x0000	No Error	-	No measures required.
0x0001	Unspecified Error	Startup	Restart the slave.
0x0002	No Memory	• Startup • Runtime	Restart the slave.
0x0003	Invalid Device Configuration	Startup	• Check whether the proper device description file is being used. • Check the configuration.

25929488/EN – 12/2018

Code	Description	Operating phase	Measure
0x0011	Invalid Request State Change	Startup	• Restart the master. • Check whether the ESM state change has been performed properly.
0x0012	Unknown Request State	Startup	Restart the master.
0x0013	Bootstrap Not Supported	Startup	The change of state to "start routine" ESM state is not permitted.
0x0014	No Valid Firmware	Startup	Check whether the proper device description file is being used.
0x0015	Invalid Mailbox Configuration Boot (is not used with SEW devices)	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x0016	Invalid Mailbox Configuration Preop	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x0017	Invalid Sync Manager Configuration (is not used with SEW devices)	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x0018	No Valid Inputs Available	Runtime	• Check whether the proper device description file is being used. • Check the configuration.
0x0019	No Valid Outputs (is not used with SEW devices)	Runtime	• Check whether the proper device description file is being used. • Check the configuration.
0x001A	Synchronization Error	Runtime	• Restart the master. • Restart the slave. • Reset the supply voltage. • Replace the device.

Code	Description	Operating phase	Measure
0x001B	Sync Manager Watchdog	Runtime	• Restart the master. • Restart the slave. • Reset the supply voltage. • Replace the device.
0x001C	Invalid Sync Manager Types (is not used with SEW devices)	Runtime	• Check whether the proper device description file is being used. • Check the configuration.
0x001D	Invalid output configuration	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x001E	Invalid input configuration	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x001F	Invalid watchdog configuration	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x0020	Slaves Need Cold Start (is not used with SEW devices)	Runtime	• Reset the supply voltage. • Restart the master.
0x0021	Slaves Need INIT (is not used with SEW devices)	Runtime	• Reset the supply voltage. • Restart the master.
0x0022	Slaves Need PREOP (is not used with SEW devices)	Runtime	Restart the master.
0x0023	Slaves Need SFAOP (is not used with SEW devices)	Runtime	Restart the master.
0x0024	Invalid Input Mapping (is not used with SEW devices)	Startup	• Check whether the proper device description file is being used. • Check the configuration.

25929488/EN – 12/2018

Code	Description	Operating phase	Measure
0x0025	Invalid Output Mapping (is not used with SEW devices)	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x0026	Inconsistent Settings (is not used with SEW devices)	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x0027	Free Run Not Supported (is not used with SEW devices)	Startup	Activate synchronization through distributed clocks.
0x0028	Synchronization Not Supported (is not used with SEW devices)	Startup	Deactivate synchronization through distributed clocks.
0x0029	Free Run Needs 3 Buffer Mode	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x002A	Background Watchdog (is not used with SEW devices)	Runtime	• Restart the master. • Restart the slave.
0x002B	No Valid Inputs and Outputs	Runtime	• Check whether the proper device description file is being used. • Check the configuration.
0x002C	Fatal Sync Error	Runtime	• Reset the supply voltage. • Restart the master. • Restart the slave.
0x002D	No Sync Error	Runtime	• Reset the supply voltage. • Restart the master. • Restart the slave.
0x0030	Invalid DC SYNC configuration	Startup	• Check whether the proper device description file is being used. • Check the configuration.

Code	Description	Operating phase	Measure
0x0031	Invalid DC Latch Configuration (is not used with SEW devices)	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x0032	PLL Error (is not used with SEW devices)	Startup	• Reset the supply voltage. • Restart the master. • Restart the slave.
0x0033	Invalid DC IO Error (is not used with SEW devices)	Runtime	• Restart the master. • Restart the slave.
0x0034	Invalid DC Timeout Error (is not used with SEW devices)	Runtime	• Restart the master. • Restart the slave.
0x0035	DC Invalid Sync Cycle Time (is not used with SEW devices)	Startup	Check the master configuration: Synchronization through DC.
0x0036	DC Sync 0 Cycle Time	Startup	Check the master configuration: "Task Cycle".
0x0037	DC Sync 1 Cycle Time (is not used with SEW devices)	Startup	Check the master configuration: "Task Cycle".
0x0041	MBX_AOE (is not used with SEW devices)	Startup	Not implemented.
0x0042	MBX_EOE (is not used with SEW devices)	Startup	Not implemented.
0x0043	MBX_COE (is not used with SEW devices)	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x0044	MBX_FOE (is not used with SEW devices)	Startup	Not implemented.
0x0045	MBX_SOE (is not used with SEW devices)	Startup	Not implemented.

25929488/EN – 12/2018

Code	Description	Operating phase	Measure
0x004F	MBX_VOE (is not used with SEW devices)	Startup	• Check whether the proper device description file is being used. • Check the configuration.
0x0050	EEPROM No Access (is not used with SEW devices)	Startup	• Restart the slave. • Replace the device.
0x0051	EEPROM Error (is not used with SEW devices)	Startup	• Restart the slave. • Replace the device.
0x0060	Slave Restarted Locally (is not used with SEW devices)	Runtime	• Reset the supply voltage. • Restart the master.
0x8000 to 0xFFFF	Vendor Specific (is not used with SEW devices)	• Runtime • Startup	See manufacturer's documentation.

8.4.3 Slave Enhanced Status

The "Slave Enhanced Status" structure provides expert information about the internal state of a slave.

Actual

The "Actual" structure provides information about the current internal state of the master:

Status Information	The structure provides information about the internal state of the slave.
Class	Classifies the internal state of the slave: • 0x08: no error • 0x04: error
Subcode	Detailed classification of the slave state.
Subcode Information	Further information on the subcode of the slave state.
Priority	Priority of the slave state.
Timestamp	The timestamp indicates the time when the current state of the slave has been set.
Expert Information	Structure with expert information on a component involved in an error state.
Component ID	Internal ID of the software component.
Component Information	Further component-specific diagnostics information.

History/entry

The structures "history" and "entry" are a combination of the last error states of the slave.

8.4.4 Slave Enhanced Network Diagnostic

The "Slave Enhanced Network Diagnostic" structure provides information on the network state and the events that occurred in the network. All the information in the structure is provided by a slave.

Port Error Counter

The "Port Error Counter" structure combines the counters of a slave port.

Lost Link Counter	This counter is incremented when a slave detects an interrupted physical connection.
Physical Layer	This counter is incremented when a slave detects an error when receiving an Ethernet frame.
Auto Forwarder Original	Number of detected invalid Ethernet frames (invalid frame counter).
Auto Forwarder Forwarded	Number of invalid Ethernet frames detected by a previous slave (forwarded Rx error counter).
EtherCAT® Processing Unit	Number of invalid Ethernet frames detected by the EtherCAT® processing unit.
Port 0	The structure combines the error counters of slave port 0.
Port 1	The structure combines the error counters of slave port 1.
Port 2	The structure combines the error counters of slave port 2.
Port 3	The structure combines the error counters of slave port 3.

PDI Diagnostics

The "PDI Diagnostics" structure combines the counters for the process data interface (PDI) of the slave. PDI is the interface between slave controller and application controller.

PDI Error Counter	Number of physical error detected by the PDI.
PDI Error Code	Error code that describes the last error detected by the PDI.
PDI Watchdog Counter	Watchdog counter for monitoring the PDI.
Process Data Watchdog Counter	Counter for activated SyncManager watchdogs. This corresponds to the process data timeout, which means the absence of OUT process data.

8.5 LED Diagnostics

The "Link/Activity" LED on ports with plug-in connections provide information for initial diagnosis.

"Link/Activity" LED	Connected	Active
On	Yes	No
Flashing	Yes	Yes
Off	No	Not available

The "run" LED on the slave allows for evaluating the current ESM state.

"Run" LED	ESM state
Off	Init
Flashing	Pre-operational
Flashes once	Safe operational
On	Operational

The "error" LED indicates the state of a connected slave.

"Error" LED	Slave state
Off	No error
Flashing	Configuration error
Flashes once	General runtime error
On	Watchdog expired (fieldbus timeout)

Further diagnostics is possible using the "AL Status Code", see chapter "AL Status Code" (→ 🗎 58).

25929488/EN – 12/2018

8.6 Substitute value configuration, master

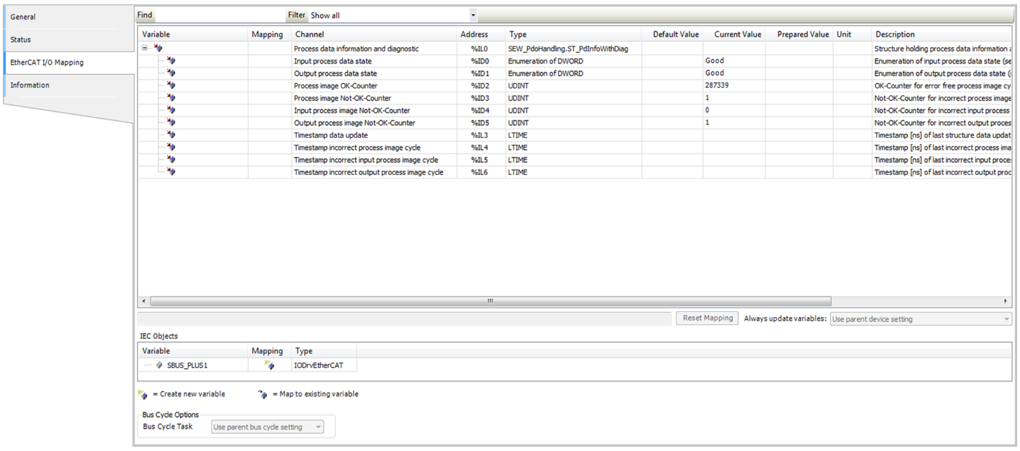
An error tolerance of 10 frames is implemented in the master by default. The error tolerance cannot be configured using the IEC Editor.

Procedure:

- Faulty or corrupt frames are discarded.
- Subsequent process data errors are counted per sync unit. The counter is set to zero when a cycle was error-free.
- When the tolerated value is exceeded, all slaves in the sync unit enter error state with code 0x4400 (Slave Status Code).
- When the tolerated value is exceeded in all sync units, also the master enters error state with code 0x43xx (Master Status Code). The global error bit is set to "true".

Note that the above described processes are not performed true to cycle. A time of 1-2 ms can pass between an exceeded counter and setting of the state.

To obtain information that is true to cycle, you can monitor the process data status in the master module under "EtherCAT® IO Mapping". The slaves are combined to sync units.



22346650379

25929488/EN – 12/2018

8.7 Substitute value configuration, slave

MOVIDRIVE® C

You can configure substitute values in FCB10 mode (interpolated position control).

To do so, proceed as follows:

1. In the MOVISUITE® parameter tree, activate substitute value configuration under the menu items [PO Configuration] > [Basic Settings].

This means that a substitute value is calculated if current values are absent. The substitute value is based on the current speed. This process is performed cyclically and is supported until fieldbus timeout. The timeout is set to 100 ms by default.

8.8 Resetting error counters

To reset error counters, define the following variables in the IEC project:

```
VAR
    result: SEW_ETC.E_EtherCAT_DiagResult;
    get_result: BOOL;
    start_reset_master_error: BOOL;
    start_reset_slave_error: BOOL;
    start_reset_slave_error_single: BOOL;
    start_reset: BOOL;
END_VAR
```

22841105675

To reset the "NetworkDiagnosticCounter" of the master, call and evaluate the following methods:

```
IF start_reset_master_error
    THEN
    result:=SBUS_PLUS1.ResetNetworkDiagnosticCounters();
    start_reset:=TRUE;
END_IF

IF (result= SEW_ETC.E_EtherCAT_DiagResult.Success) AND start_reset=TRUE
    THEN
    start_reset_master_error:=FALSE;
    get_result:=SBUS_PLUS1.ResetNetworkDiagnosticCountersGetResult(xError=> , eError=> );

    IF get_result=TRUE
        THEN
        start_reset:=FALSE;
    END_IF
END_IF
```

22841147659

25929488/EN – 12/2018

To reset all "SlaveErrorCounters", call and evaluate the following methods:

```
IF start_reset_slave_error
    THEN
    result:=SBUS_PLUS1.ResetAllSlaveErrorCounter();
    start_reset:=TRUE;
END_IF

IF (result= SEW_ETC.E_EtherCAT_DiagResult.Success)AND start_reset=TRUE
    THEN
    start_reset_slave_error:=FALSE;
    get_result:=SBUS_PLUS1.ResetAllSlaveErrorCounterGetResult(xError=> , eError=> );

    IF get_result=TRUE
        THEN
        start_reset:=FALSE;
    END_IF
END_IF
```

22841151243

To reset a "SlaveErrorCounter", call and evaluate the following methods:

```
IF start_reset_slave_error_single
    THEN
    result:=Dev_Axis1_Axis2.ResetSlaveErrorCounter();
    start_reset:=TRUE;
END_IF

IF (result= SEW_ETC.E_EtherCAT_DiagResult.Success) AND start_reset=TRUE
    THEN
    start_reset_slave_error_single:=FALSE;
    get_result:=Dev_Axis1_Axis2.ResetSlaveErrorCounterGetResult(xError=> , eError=> );

    IF get_result=TRUE
        THEN
        start_reset:=FALSE;
    END_IF
END_IF
```

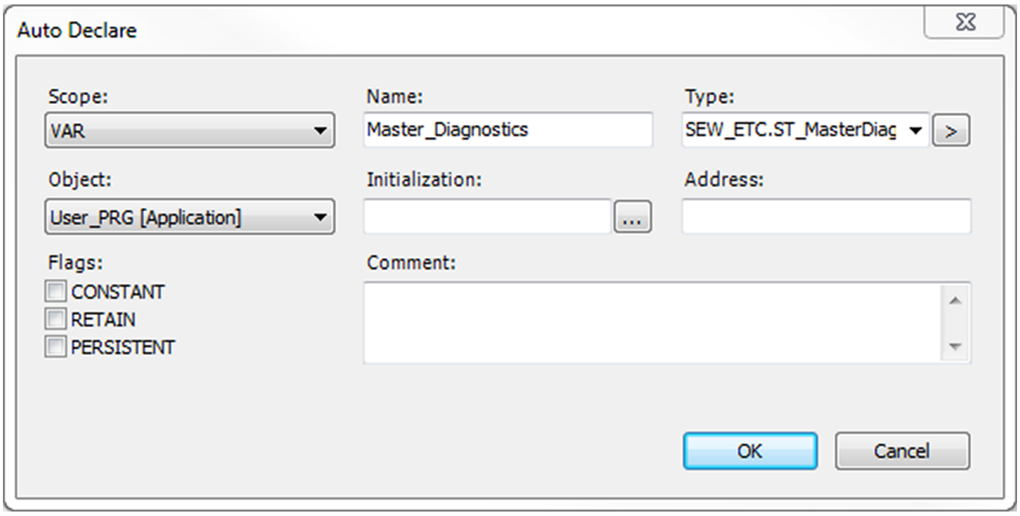
22841154827

25929488/EN – 12/2018

8.9 Using diagnostics information in the IEC Editor

8.9.1 Master

If you want to use the diagnostics information of the master in the IEC Editor, you have to create a variable of the type "SEW_ETC.ST_MasterDiagnostics" in the PLC program.

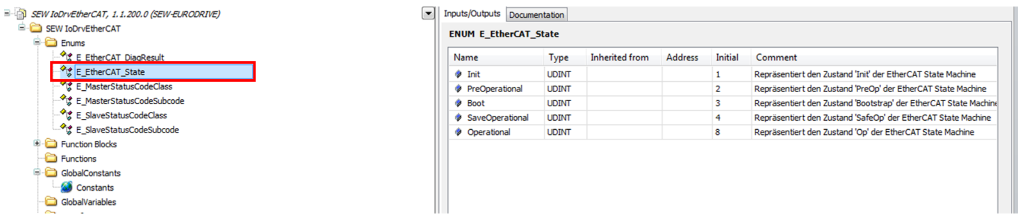


22354927499

The following structure is available after having created the variable:

22354931083

Refer to the Library Manager for an overview of all items of this structure.



22354934667

Example of reading the state machine of the master:

1. Define the variables for master diagnostics as well as the state machine.

```
VAR
    Master_Diagnostics: SEW_ETC.ST_MasterDiagnostics;
    Current_State: SEW_ETC.E_EtherCAT_State;
END_VAR
```

22347402507

2. Call the "GetDiagnostics()" method of the master and link this method with the variable you have created.

⇨ You can read the state of the master from this structure.

```
Master_Diagnostics:=SBUS_PLUS1.GetDiagnostics();
Current_State:= Master_Diagnostics.stMasterBasicInformation.eCurrentState;
```

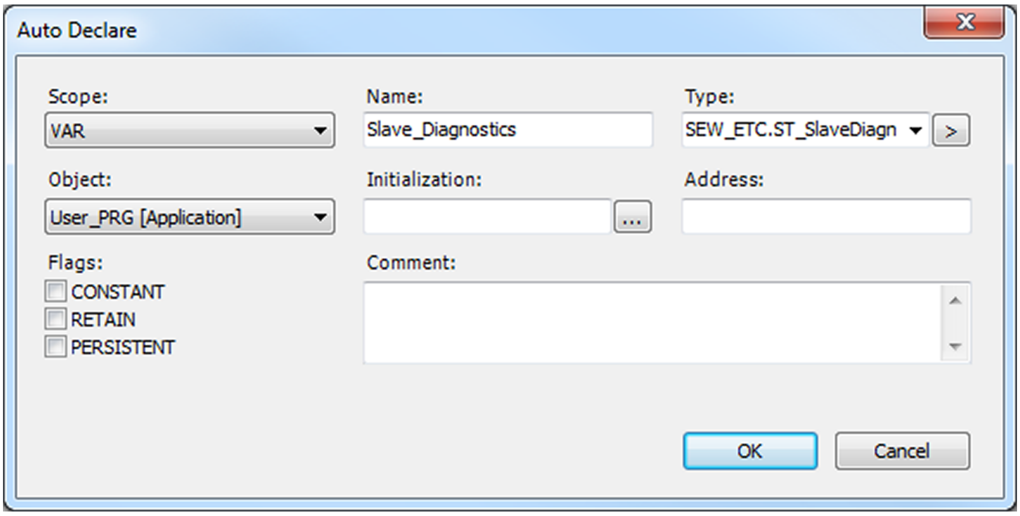
22347280139

```
Master_Diagnostics:=SBUS_PLUS1.GetDiagnostics();
Current_State[Operationa ▸]:= Master_Diagnostics.stMasterBasicInformation.eCurrentState[Operationa ▸];
```

22347283723

8.9.2 Slave

If you want to use the diagnostics information of the slave in the IEC Editor, you have to create a variable of the type "SEW_ETC.ST_SlaveDiagnostics" in the PLC program.



22354938251

The following structure is available after having created the variable:

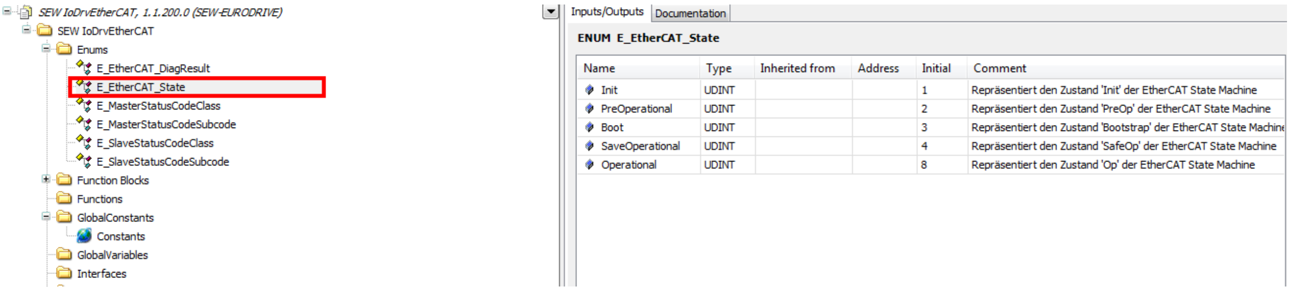
Slave_Diagnostics	SEW_ETC.ST_SlaveDiagnostics
stSlaveBasicInformation	ST_SlaveBasicInformation
eCurrentState	E_ETHERCAT_STATE
xCurrentStateObtainedFromSlave	BOOL
xErrorIndicationObtainedFromSlave	BOOL
xSyncUnitError	BOOL
stSlaveQuickDiagnostic	ST_SlaveQuickDiagnostic
uiAlStatusCode	UINT
stStartup	ST_SlaveStartup
stRuntime	ST_SlaveRuntime
stSlaveEnhancedStatus	ST_SlaveEnhancedStatus
stActual	ST_SlaveActual
stHistory	ST_SlaveHistory
stSlaveEnhancedNetworkDiagnostic	ST_SlaveEnhancedNetworkDiagnostic
stPortErrorCounter	ST_PortErrorCounter
stPdiDiagnostic	ST_PdiDiagnostic
stProcessDataDiagnostic	ST_ProcessDataDiagnostic

22354967435

25929488/EN – 12/2018

SEW EURODRIVE

Refer to the Library Manager for an overview of all items of this structure.



22354971019

Example of reading the state machine of the slave:

1. Define the variables for slave diagnostics as well as the state machine.

```
VAR
    Slave_Diagnostics: SEW_ETC.ST_SlaveDiagnostics;
    Current_State: SEW_ETC.E_EtherCAT_State;
END_VAR
```

22347413259

2. Call the "GetDiagnostics()" method of the master and link this method with the variable you have created.

⇨ You can read the state of the slave from this structure.

```
Slave_Diagnostics:= Dev_Axis1_Axis2.GetDiagnostics();
Current_State:= Slave_Diagnostics.stSlaveBasicInformation.eCurrentState;
```

22347406091

```
Slave_Diagnostics:= Dev_Axis1_Axis2.GetDiagnostics();
Current_State[Operationa]:= Slave_Diagnostics.stSlaveBasicInformation.eCurrentState[Operationa];
```

22347409675

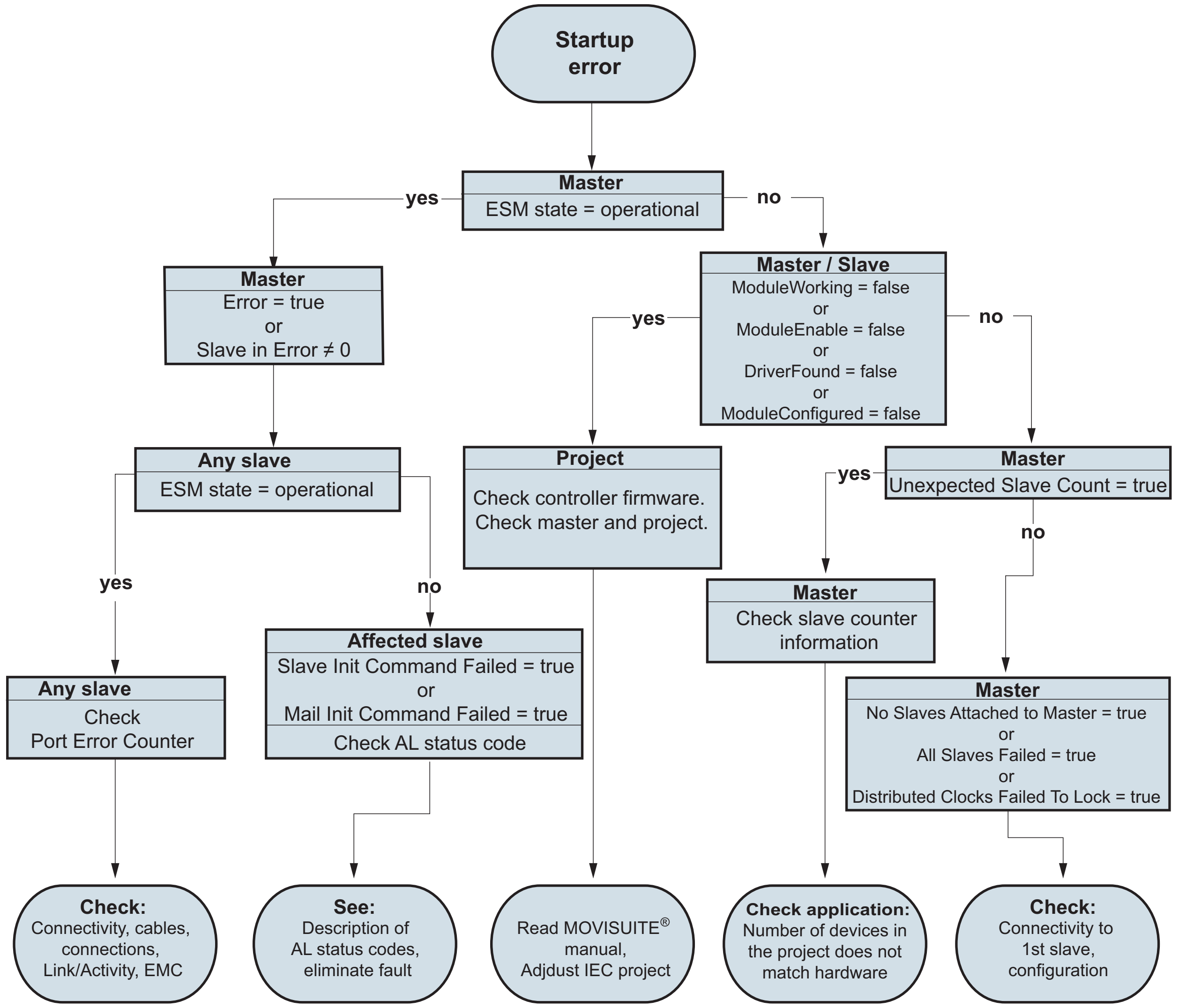
8.10 Recommended procedure for diagnostics

It is useful to divide the error causes into several categories to allow for fast and accurate diagnostics:

- Software errors, for example:
  - Slave not configured correctly in the IEC Editor: e.g. configuration of basic functions or of process data.
  - Faulty processing of process data.
  - Synchronization loss.
- Hardware errors, for example:
  - Connection is interrupted between individual stations in the network, e.g. caused by a broken cable or loose contact.
  - EMC problems in the plant.
  - Malfunctions or voltage failure in the slaves.

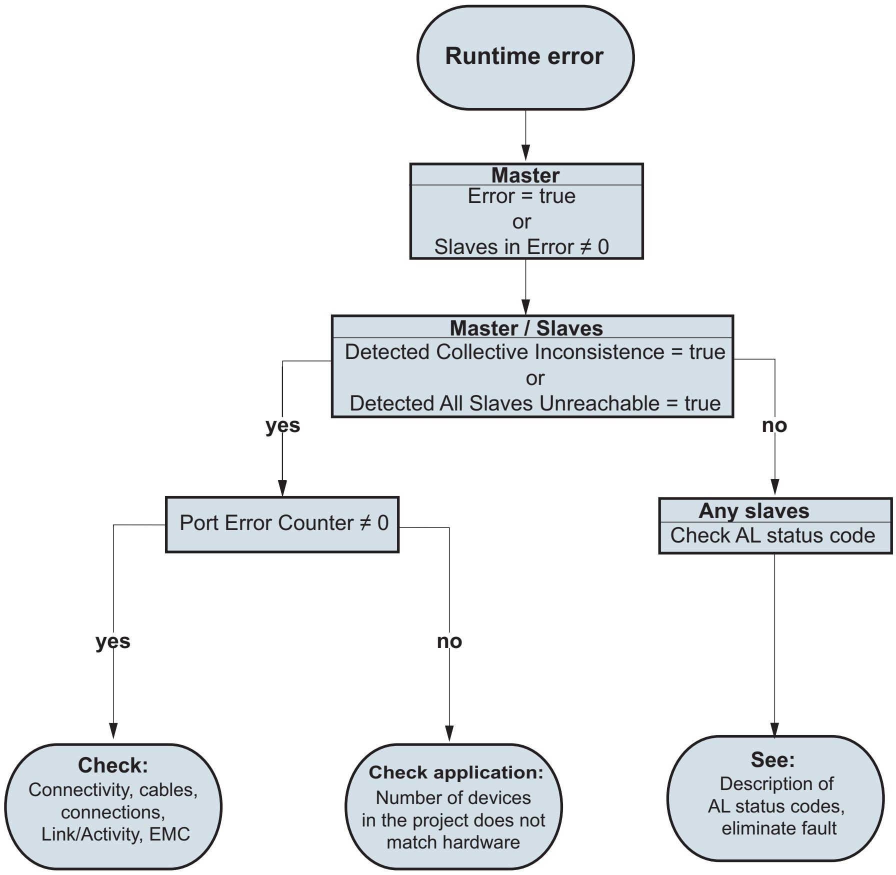
25929488/EN – 12/2018

The following diagram provides support in troubleshooting. A distinction is made between errors that occurred during startup and errors that occurred during runtime:



22302520843

25929488/EN – 12/2018



22302518411

9 Application examples

The following basic functions can be implemented with the current software state:

Startup and engineering using MOVISUITE®

- Easy and convenient operation with MOVISUITE® standard

Auto addressing of slaves

- Quick startup
- In the event of a fault: components can be replaced quickly

SEW configuration state

- Modular structure possible during startup phase
- Enhanced engineering during startup phase

Synchronization of axes using distributed clocks

- Interpolated control types are possible
- Reduced jitter during process data processing in the slaves

Deactivation of slaves in the IEC project

- Only minor adjustments of the project necessary if the hardware setup changes

Scheduling mode

- The dynamic properties of the stations in the system can be adjusted
- The number of stations in the system can be adjusted

Using optional slaves

- Project creation for series machines
- Using unique device identification increases safety when replacing devices or cables

FSoE routing

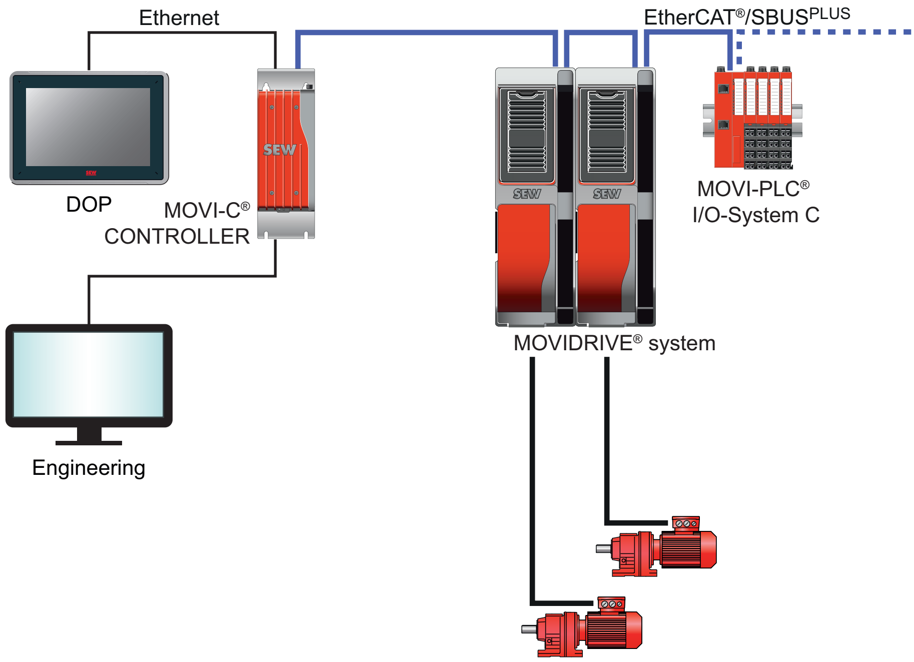
- Integrating a 3rd party FSoE master allows for FSoE communication

The following chapters give an overview of typical application examples and basic functions used.

25929488/EN – 12/2018

9.1 Operation without synchronized axes

Application examples	Functions used
• Materials handling equipment • Conveyors • Agitators and mixers	• Startup and engineering with MOVISUITE® • Auto addressing of slaves • SEW configuration state

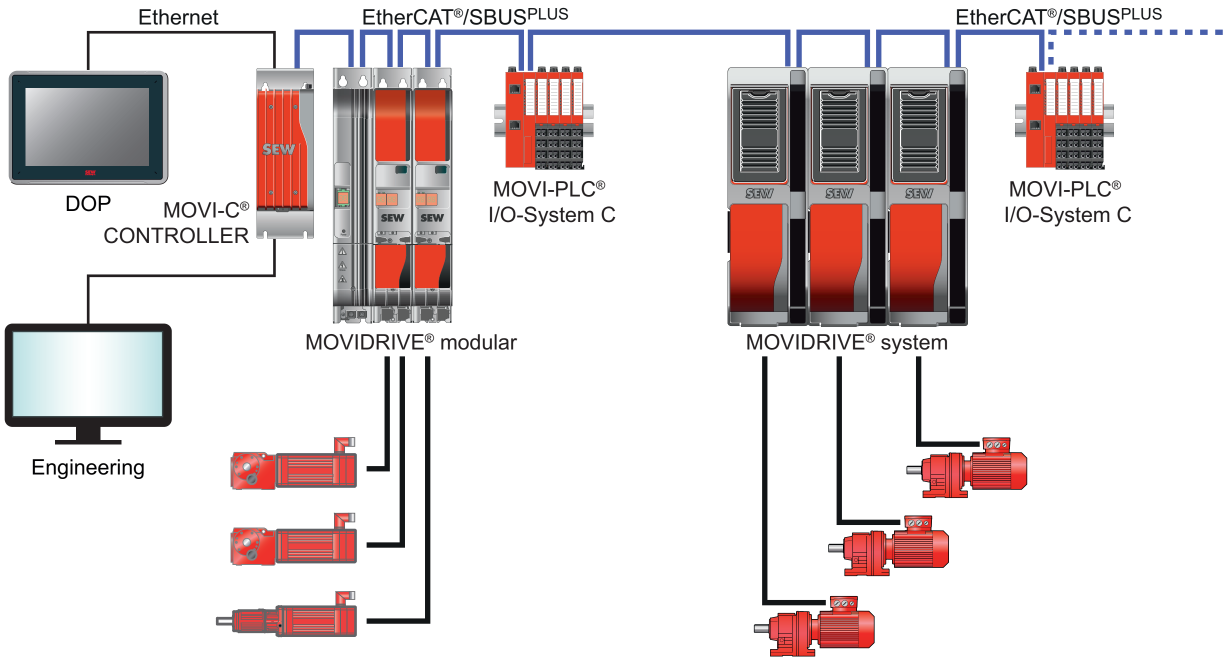


22819385227

25929488/EN – 12/2018

9.2 Operation with synchronized axes

Application examples	Functions used
• Packaging machines • Handling tasks • Robotics applications • Molding machines	• Startup and engineering with MOVISUITE® • Auto addressing of slaves • SEW configuration state • Synchronization through distributed clocks

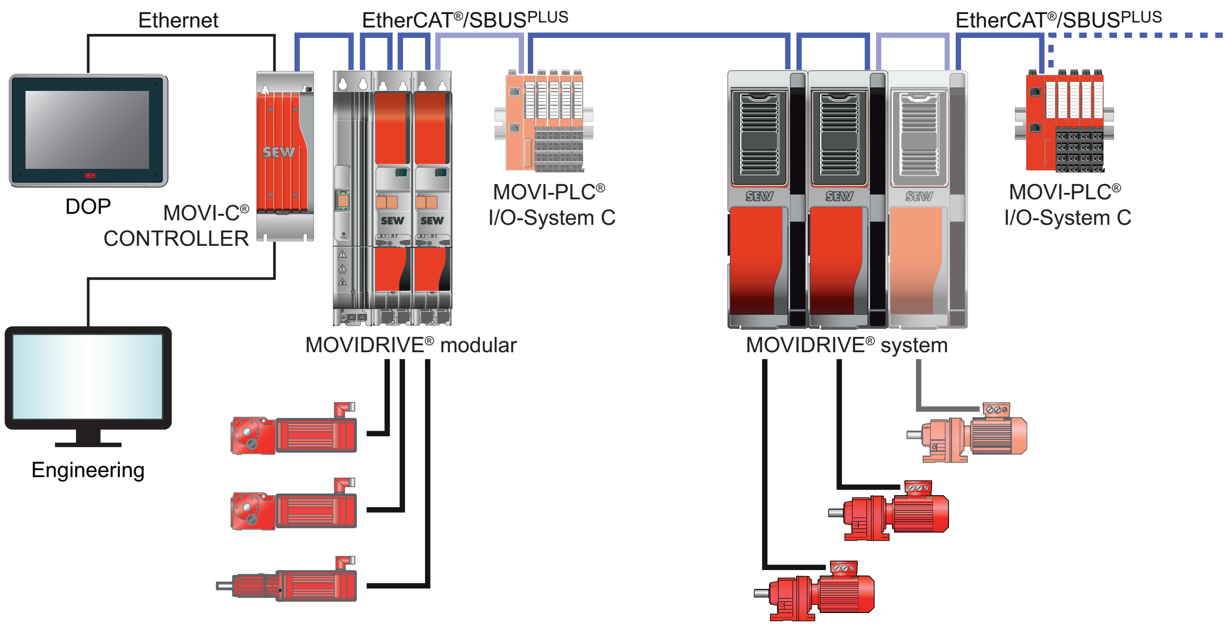


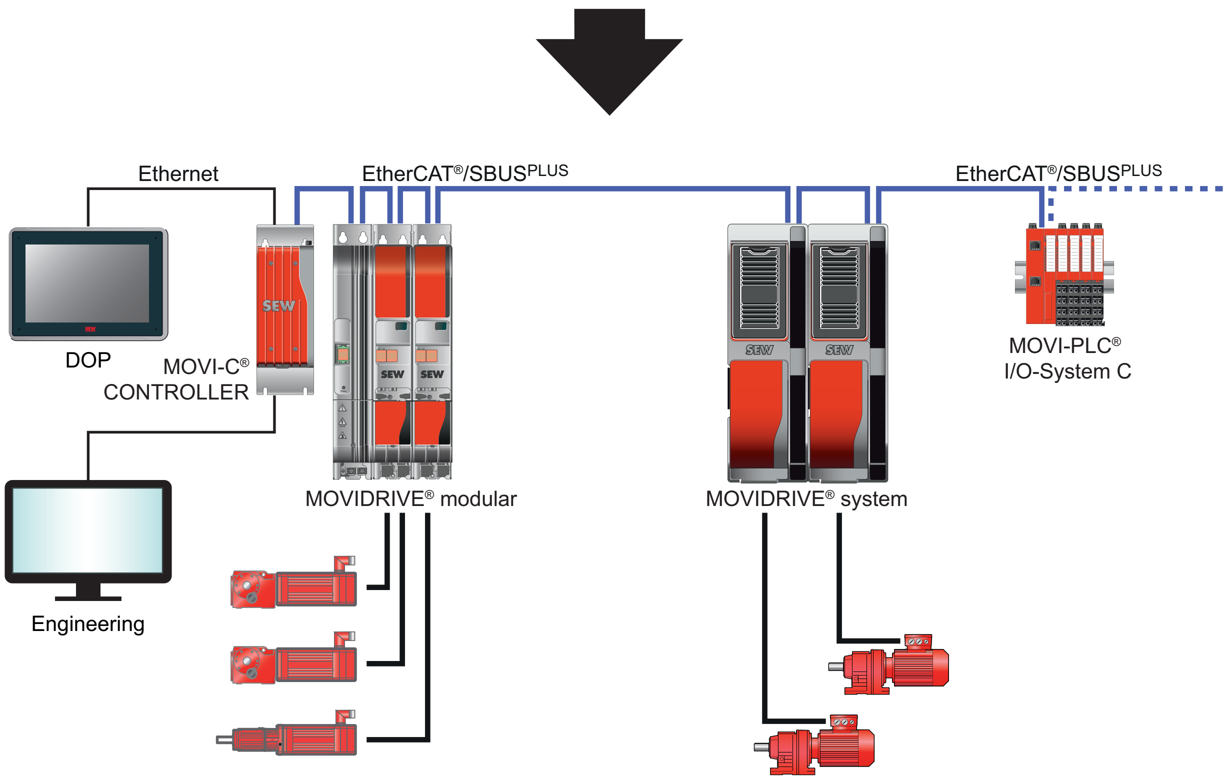
22819378059

25929488/EN – 12/2018

9.3 Operation with variable project adjustment of network stations

Application examples	Functions used
• Series machines with different stations per system	• Startup and engineering with MOVISUITE® • Auto addressing of slaves • SEW configuration state • Synchronization of distributed clocks • Deactivation of slaves in the IEC project



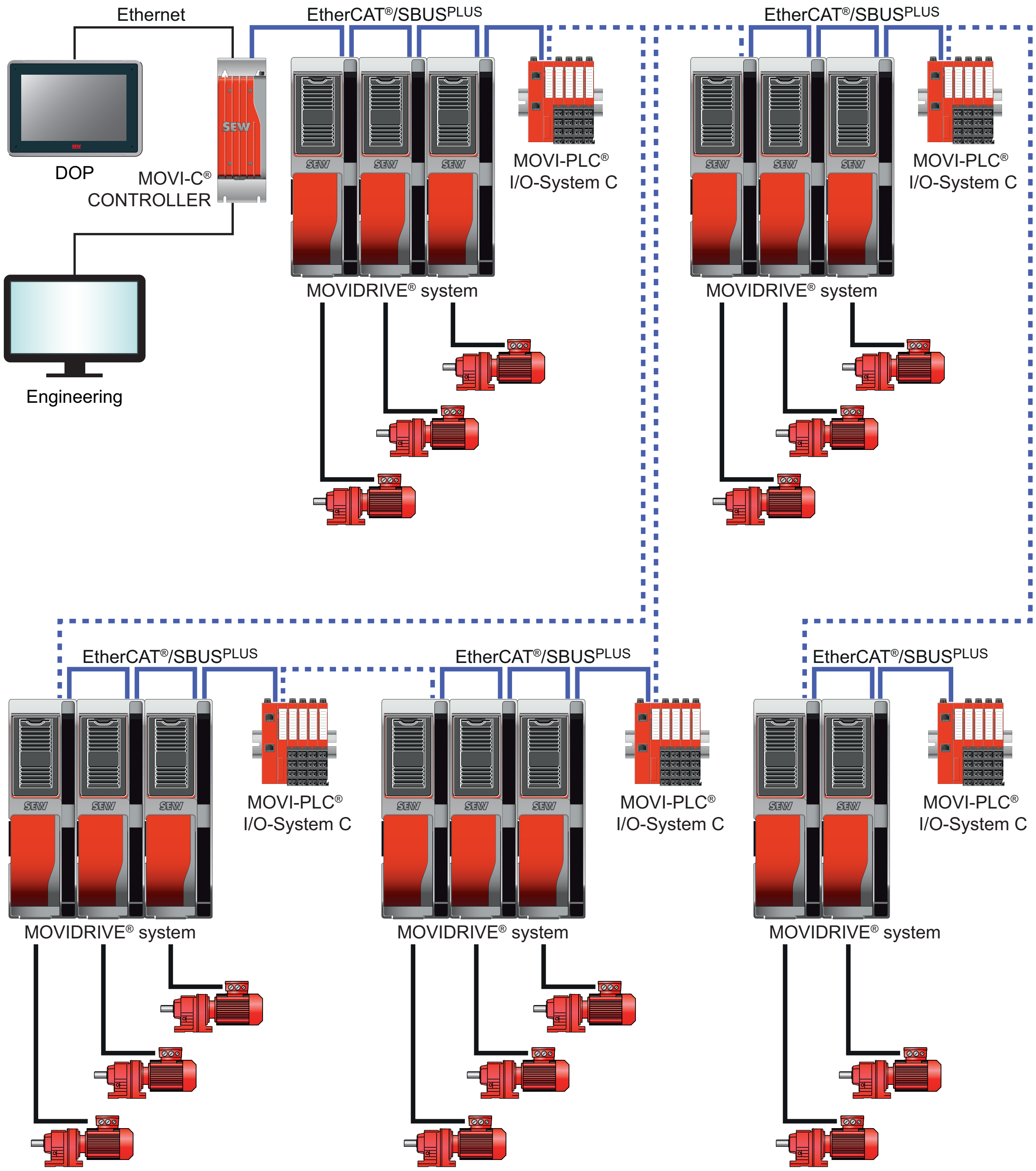


22819381643

25929488/EN – 12/2018

9.4 Operation with a large number of network stations

Application examples	Functions used
• Sprinklers • Special systems	• Startup and engineering with MOVISUITE® • Auto addressing of slaves • SEW configuration state • Changeover to scheduling mode

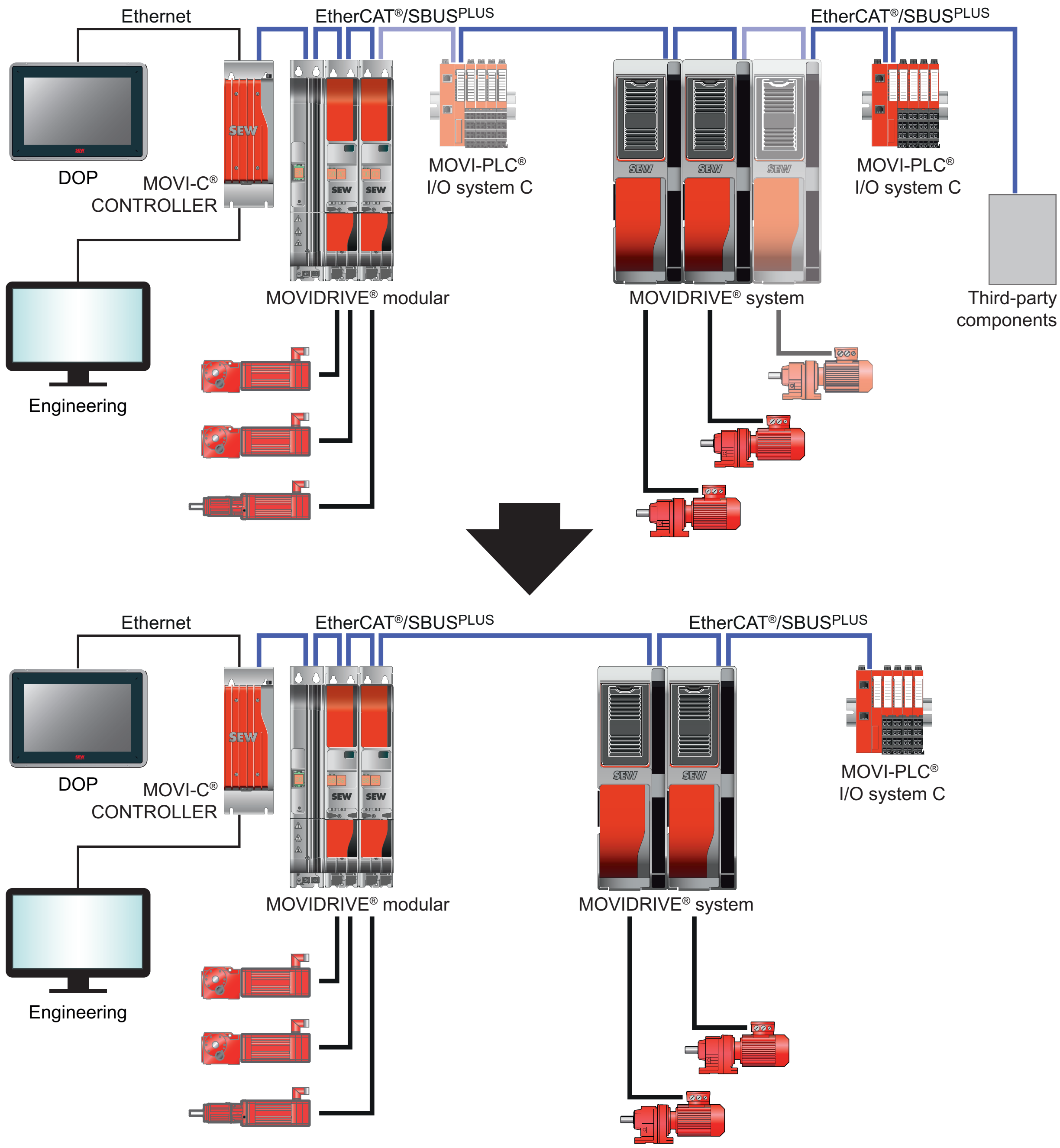


22819374475

25929488/EN – 12/2018

9.5 Operation with variable project adjustment of network stations by configuration of optional slaves

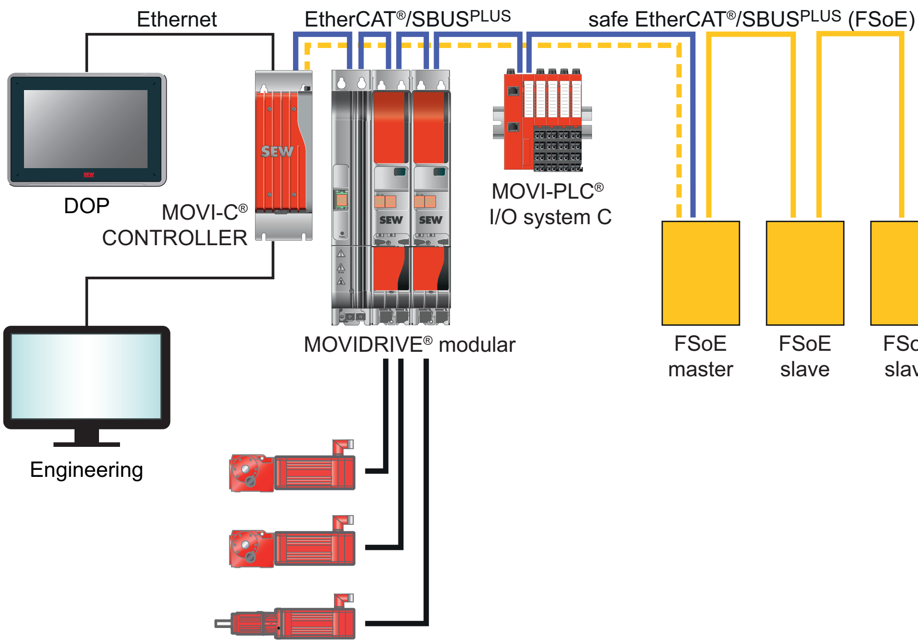
Application examples	Functions used
• Configuration of series machines	• Startup and engineering with MOVISUITE® • Auto addressing of slaves • SEW configuration state • Synchronization of distributed clocks • Using ID selection via DIL switch • Using the station alias ID • Using optional slaves



24847490443

9.6 Operation with functional safety technology

Application examples	Functions used
• Systems with FS requirements	• FSoE routing in the MOVI-C® CONTROLLER



24846808203

10 Network layout

10.1 Maximum number of stations

The table below shows the maximum permitted number of network stations:

	MOVI-C® CONTROLLER ...			
	standard UHX25A	advanced UHX45A	power eco UHX84A	power UHX85A
EtherCAT® stations	128	128	256	256
Frequency inverters	8	16	64	64
Interpolating axes	2	8	16	32
Auxiliary axes	6	8	16	32
Safety slots	8	8	24	24

10.2 Calculation of available useful data length

This chapter informs about how to calculate the available useful data length.

Useful data length available when using LRW

Use	Frame length
Ethernet frame	1518 bytes
Ethernet header + CRC	-18 bytes
EtherCAT® header	-2 bytes
Mailbox polling per LRD	-12 bytes - [≈(S/8)] bytes S = Number of slaves
Distribution of system time and sync monitoring	-36 bytes Length independent of DC
Monitoring of slave state	-14 bytes
Useful data header	-12 bytes
Available useful data	**1424 bytes - [≈(S/8)] bytes** **S = Number of slaves**

Available useful data when using LWR/LRD

Use	Frame length
Ethernet frame	1518 bytes
Ethernet header + CRC	-18 bytes
EtherCAT® header	-2 bytes
Mailbox polling per LRD	-12 bytes - [≈(S/8)] bytes S = Number of slaves

25929488/EN – 12/2018

Use	Frame length
Distribution of system time and sync monitoring	-36 bytes Length independent of DC
Monitoring of slave state	-14 bytes
Useful data header	-24 bytes
Available useful data	**1412 bytes - [≈(S/8)] bytes** **S = Number of slaves**

10.3 Available slots in the IEC project

The following tables provide an overview of the process data length required by a MOVI-C® MOVIDRIVE® system.

Slots for MOVI-C® MOVIDRIVE®

Name in the IEC project	Process data length	
	Bytes in	Bytes out
4 PD	8	8
8 PD	16	16
12 PD	24	24
16 PD	32	32

Safety slots for MOVI-C® MOVIDRIVE®

Name in the IEC project	Process data length	
	Bytes in	Bytes out
Failsafe 10 bytes	10	10
Failsafe 14 bytes	14	14
Failsafe 28 bytes	28	28
Failsafe container PROFIsafe 4/3 bytes	14	14
Failsafe container PROFIsafe 6/5 bytes	14	14

10.4 Process data utilization: Calculation examples

You can calculate the process data utilization of a network using the information provided in chapters "Maximum number of stations" (→ 🗎 81) and "Calculation of available useful data length" (→ 🗎 81). The network utilization must not exceed 100%.

The following two tables show a calculation example each for two different networks:

Example (LRW command)	Quantity		Bytes in	Bytes out
MOVIDRIVE® modular MDA90A single-axis module	10	16 PD	320	320
		2 × PROFsafe 4/3 bytes	28	28
MOVIDRIVE® modular MDD90A double-axis module	10	12 PD each	480	480
		1 x PROFIsafe 6/5 bytes	14	14
MOVI-PLC® I/O system C: 3 × ODI81C 2 × ODO81C	1	8 bytes in for diagnostics 3 bytes in 2 bytes out	11	2
Number of slaves	**21**			
Utilized useful data (sum)	*Comment: With LRW, use the greater value of bytes in und bytes out as the sum.*		**853**	**844**
			853 bytes	
Available useful data	1424 bytes - [≈(21/8)] bytes = 1421 bytes			
Process data utilization	60%			

Example (LWR/LDR command)	Quantity		Bytes in	Bytes out
MOVIDRIVE® modular MDA90A single-axis module	15	16 PD	480	480
		1 x PROFIsafe 6/5 bytes	14	14
MOVIDRIVE® modular MDD90A double-axis module	2	8 PD each	64	64
		1 × PROFIsafe 4/3 bytes	14	14
Hilscher NT-100-RE-DP	1	10 PD	20	20
Number of slaves	**18**			
Utilized useful data (sum)	*Comment: With LWR/LDR, use the sum of bytes in and bytes out.*		**592**	**592**
			1184 bytes*	
Available useful data	1412 bytes - [≈(18/8)] bytes = 1409 bytes			
Process data utilization	84%			

25929488/EN – 12/2018

Index

Icons

A

B

C

D

E

H

L

M

N

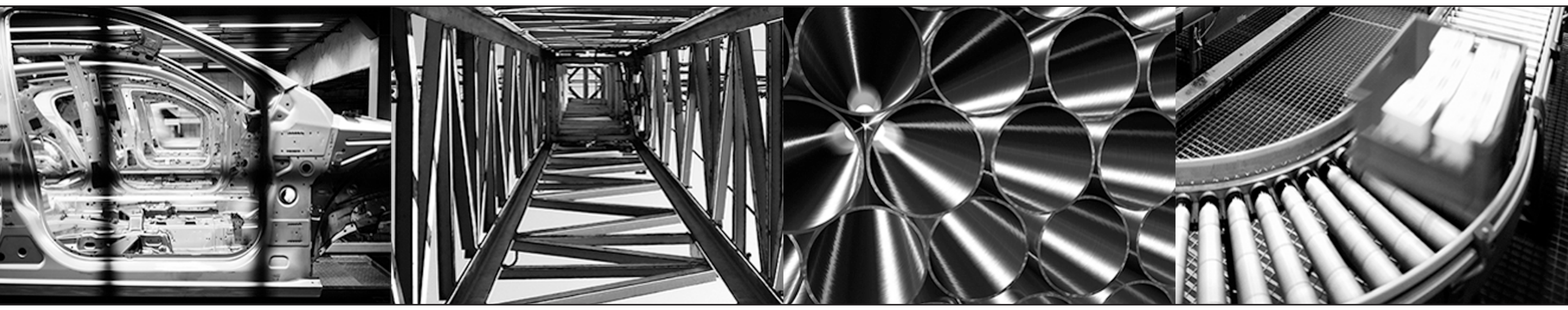
O

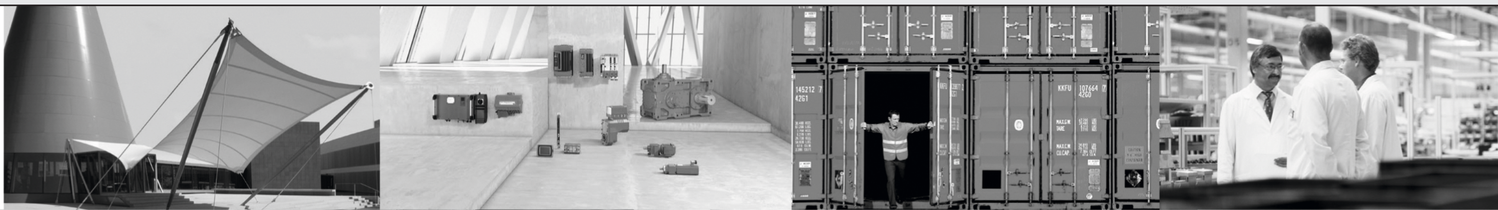
P

R

S

25929488/EN – 12/18






SEW-EURODRIVE
Driving the world

SEW-EURODRIVE GmbH & Co KG
Ernst-Blickle-Str. 42
76646 BRUCHSAL
GERMANY
Tel. +49 7251 75-0
Fax +49 7251 75-1970
sew@sew-eurodrive.com
→ **www.sew-eurodrive.com**