YASKAWA

VIPA System SLIO

IM | 053-1EC01 | Manual

HB300 | IM | 053-1EC01 | en | 21-10

Interface module EtherCAT - IM 053EC

YASKAWA Europe GmbH
Ohmstraße 4
91074 Herzogenaurach
Tel.: +49 9132 744 0
Fax: +49 9132 744 186
Email: info@yaskawa.eu.com
Internet: www.yaskawa.eu.com

053-1EC01_000_IM 053EC,6,EN - © 2021

Table of contents

1 General

1.1 Copyright © YASKAWA Europe GmbH

All Rights Reserved

For permission to reproduce or distribute, please contact: YASKAWA Europe GmbH, European Headquarters, Hauptstraße 185, 65760 Eschborn, Germany

Tel.: +49 6196 569 300

Fax.: +49 6196 569 398

Email: info@yaskawa.eu.com

Internet: www.yaskawa.eu.com

Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. Nevertheless, the authors retain the right to modify the information.

This customer document describes all the hardware units and functions known at the present time. Descriptions may be included for units which are not present at the customer site. The exact scope of delivery is described in the respective purchase contract.

EC conformity declaration

Hereby, YASKAWA Europe GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions. Conformity is indicated by the CE marking affixed to the product.

Conformity Information

For more information regarding CE marking and Declaration of Conformity (DoC), please contact your local representative of YASKAWA Europe GmbH.

Trademarks

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S and Commander Compact are registered trademarks of YASKAWA Europe GmbH.

SPEED7 is a registered trademark of YASKAWA Europe GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300, S7-400 and S7-1500 are registered trademarks of Siemens AG.

Microsoft and Windows are registered trademarks of Microsoft Inc., USA.

Portable Document Format (PDF) and Postscript are registered trademarks of Adobe Systems, Inc.

All other trademarks, logos and service or product marks specified herein are owned by their respective companies.

Document support

Contact your local representative of YASKAWA Europe GmbH if you have errors or questions regarding the content of this document. If such a location is not available, you can reach YASKAWA Europe GmbH via the following contact:

YASKAWA Europe GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Fax: +49 9132 744 29 1204

Email: Documentation.HER@yaskawa.eu.com

Technical support

Contact your local representative of YASKAWA Europe GmbH if you encounter problems or have questions regarding the product. If such a location is not available, you can reach the YASKAWA customer service via the following contact:

YASKAWA Europe GmbH,
European Headquarters, Hauptstraße 185, 65760 Eschborn, Germany

Tel.: +49 6196 569 500 (hotline)

Email: support@yaskawa.eu.com

1.2 About this manual

Objective and contents

This manual describes the IM 053-1EC01 of the VIPA System SLIO. It contains a description of the structure, project engineering and deployment.

Product	Order number	as of state:	
		HW	FW
IM 053EC	053-1EC01	01	V2.0.0

Target audience

The manual is targeted at users who have a background in automation technology.

Structure of the manual

The manual consists of chapters. Every chapter provides a self-contained description of a specific topic.

Guide to the document

The following guides are available in the manual:

- An overall table of contents at the beginning of the manual
- References with page numbers

Availability

The manual is available in:

- printed form, on paper
- in electronic form as PDF-file (Adobe Acrobat Reader)

Icons Headings

Important passages in the text are highlighted by following icons and headings:

> ⚠ **DANGER!**
> Immediate or likely danger. Personal injury is possible.

CAUTION!
Damages to property is likely if these warnings are not heeded.

Supplementary information and useful tips.

1.3 Safety information

Applications conforming with specifications

The system is constructed and produced for:

- communication and process control
- general control and automation tasks
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle

DANGER!
This device is not certified for applications in

– in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

CAUTION!
The following conditions must be met before using or commissioning the components described in this manual:

– Hardware modifications to the process control system should only be carried out when the system has been disconnected from power!
– Installation and hardware modifications only by properly trained personnel.
– The national rules and regulations of the respective country must be satisfied (installation, safety, EMC ...)

Disposal

National rules and regulations apply to the disposal of the unit!

2 Basics and mounting

2.1 Safety information for users

Handling of electrostatic sensitive modules

VIPA modules make use of highly integrated components in MOS-Technology. These components are extremely sensitive to over-voltages that can occur during electrostatic discharges. The following symbol is attached to modules that can be destroyed by electrostatic discharges.

The Symbol is located on the module, the module rack or on packing material and it indicates the presence of electrostatic sensitive equipment. It is possible that electrostatic sensitive equipment is destroyed by energies and voltages that are far less than the human threshold of perception. These voltages can occur where persons do not discharge themselves before handling electrostatic sensitive modules and they can damage components thereby, causing the module to become inoperable or unusable. Modules that have been damaged by electrostatic discharges can fail after a temperature change, mechanical shock or changes in the electrical load. Only the consequent implementation of protection devices and meticulous attention to the applicable rules and regulations for handling the respective equipment can prevent failures of electrostatic sensitive modules.

Shipping of modules

Modules must be shipped in the original packing material.

Measurements and alterations on electrostatic sensitive modules

When you are conducting measurements on electrostatic sensitive modules you should take the following precautions:

- Floating instruments must be discharged before use.
- Instruments must be grounded.

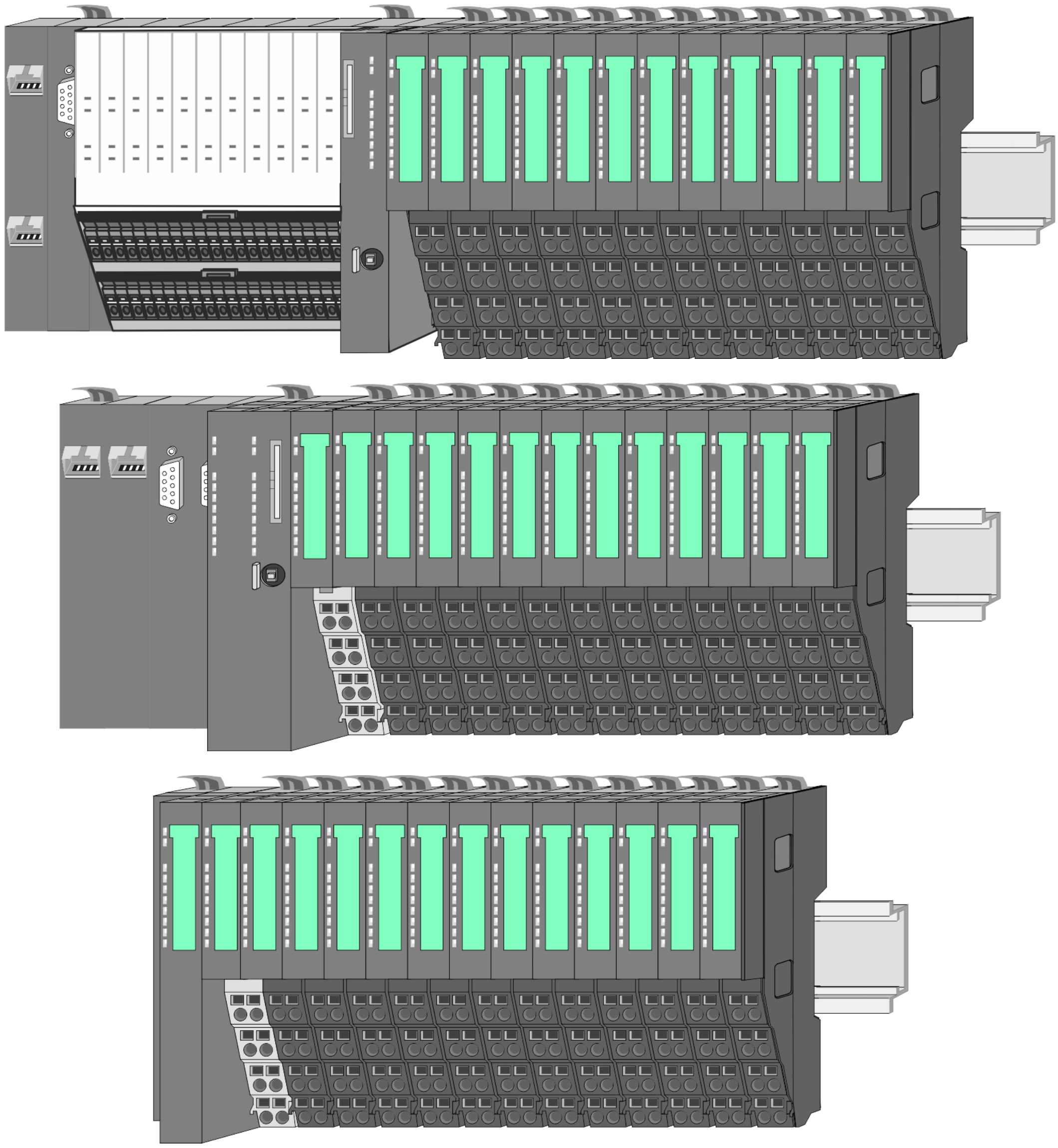
Modifying electrostatic sensitive modules you should only use soldering irons with grounded tips.

> **CAUTION!**
> Personnel and instruments should be grounded when working on electrostatic sensitive modules.

2.2 System conception

2.2.1 Overview

The System SLIO is a modular automation system for assembly on a 35mm mounting rail. By means of the periphery modules with 2, 4, 8 and 16 channels this system may properly be adapted matching to your automation tasks. The wiring complexity is low, because the supply of the DC 24V power section supply is integrated to the backplane bus and defective modules may be replaced with standing wiring. By deployment of the power modules in contrasting colors within the system, further isolated areas may be defined for the DC 24V power section supply, respectively the electronic power supply may be extended with 2A.

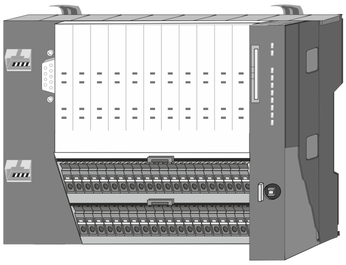


2.2.2 Components

- CPU (head module)
- Bus coupler (head module)
- Line extension
- 8x periphery modules
- 16x periphery modules
- Accessories

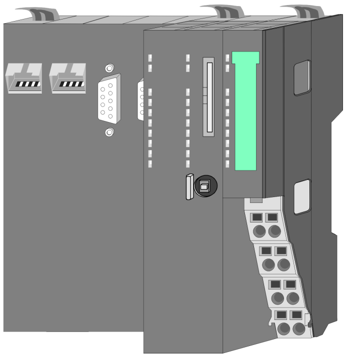
> ⚠ **CAUTION!**
> Only VIPA modules may be combined. A mixed operation with third-party modules is not allowed!

CPU 01xC



With the CPU 01xC electronic, input/output components and power supply are integrated to one casing. In addition, up to 64 periphery modules of the System SLIO can be connected to the backplane bus. As head module via the integrated power module for power supply CPU electronic and the I/O components are supplied as well as the electronic of the periphery modules, which are connected via backplane bus. To connect the power supply of the I/O components and for DC 24V power section supply of via backplane bus connected periphery modules, the CPU has removable connectors. By installing of up to 64 periphery modules at the backplane bus, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

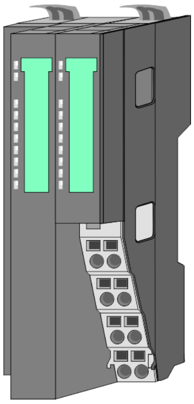
CPU 01x



With this CPU 01x, CPU electronic and power supply are integrated to one casing. As head module, via the integrated power module for power supply, CPU electronic and the electronic of the connected periphery modules are supplied. The DC 24V power section supply for the linked periphery modules is established via a further connection of the power module. By installing of up to 64 periphery modules at the backplane bus, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

> ⚠ **CAUTION!**
> CPU part and power module may not be separated!
>
> Here you may only exchange the electronic module!

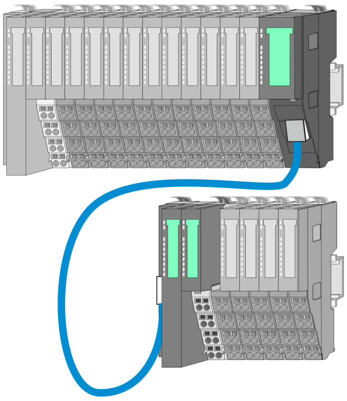
Bus coupler



With a bus coupler bus interface and power module is integrated to one casing. With the bus interface you get access to a subordinated bus system. As head module, via the integrated power module for power supply, bus interface and the electronic of the connected periphery modules are supplied. The DC 24V power section supply for the linked periphery modules is established via a further connection of the power module. By installing of up to 64 periphery modules at the bus coupler, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

> **CAUTION!**
>
> Bus interface and power module may not be separated!
>
> Here you may only exchange the electronic module!

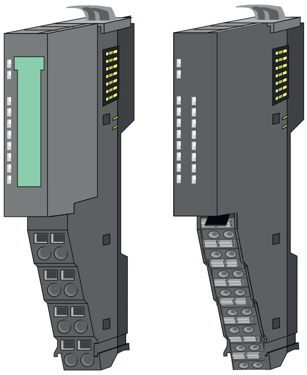
Line extension



In the System SLIO there is the possibility to place up to 64 modules in on line. By means of the line extension you can divide this line into several lines. Here you have to place a line extension master at each end of a line and the subsequent line has to start with a line extension slave. Master and slave are to be connected via a special connecting cable. In this way, you can divide a line on up to 5 lines. For each line extension the maximum number of pluggable modules at the System SLIO bus is decreased by 1. To use the line extension no special configuration is required.

> *Please note that some modules do not support line extensions due to the system. For more information, see the ‘System SLIO - Compatibility List’ at www.vipa.com*

Periphery modules

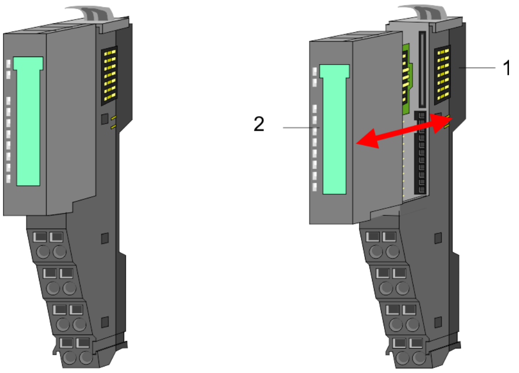


The periphery modules are available in the following 2 versions, whereby of each the electronic part can be replaced with standing wiring:

- 8x periphery module for a maximum of 8 channels.
- 16x periphery module for a maximum of 16 channels.

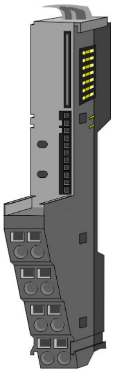
8x periphery modules

Each 8x periphery module consists of a *terminal* and an *electronic module*.



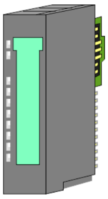
1 Terminal module
2 Electronic module

Terminal module



The *terminal* module serves to carry the electronic module, contains the backplane bus with power supply for the electronic, the DC 24V power section supply and the staircase-shaped terminal for wiring. Additionally the terminal module has a locking system for fixing at a mounting rail. By means of this locking system your system may be assembled outside of your switchgear cabinet to be later mounted there as whole system.

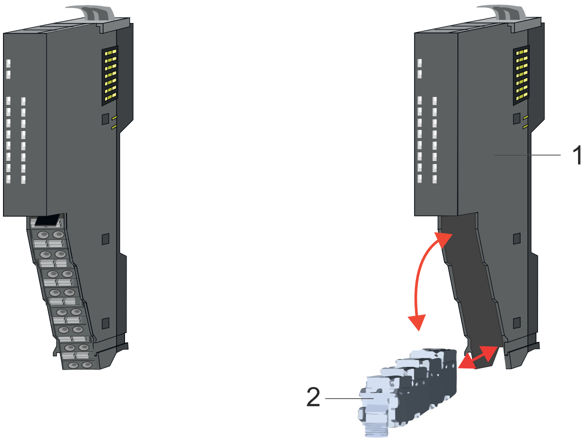
Electronic module



The functionality of a periphery module is defined by the *electronic module*, which is mounted to the terminal module by a sliding mechanism. With an error the defective electronic module may be exchanged for a functional module with standing installation. At the front side there are LEDs for status indication. For simple wiring each module shows corresponding connection information at the front and at the side.

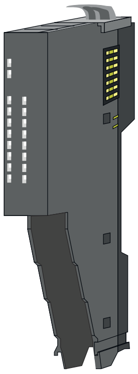
16x periphery modules

Each 16x periphery module consists of an *electronic unit* and a *terminal block*.



1 Electronic unit
2 Terminal block

Electronic unit



The functionality of a 16x periphery module is defined via the terminal block, which is connected to the *electronic unit* via a secure flap mechanism. In the case of an error you can exchange the defective electronic unit for a functional unit with standing wiring. At the front side there are LEDs for status indication. For easy wiring each electronic unit shows corresponding connection information at the side. The electronic unit provides the slot for the terminal block for the wiring and contains the backplane bus with power supply for the electronic and the connection to the DC 24V power section supply. Additionally the electronic unit has a locking system for fixing it at a mounting rail. By means of this locking system your system may be assembled outside of your switchgear cabinet to be later mounted there as whole system.

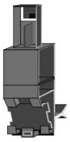
Terminal block

The *terminal block* provides the electrical interface for the signalling and supplies lines of the module. When mounting the terminal block, it is attached to the bottom of the electronic unit and turned towards the electronic unit until it clicks into place. With the wiring a "push-in" spring-clip technique is used. This allows a quick and easy connection of your signal and supply lines. The clamping off takes place by means of a screwdriver.

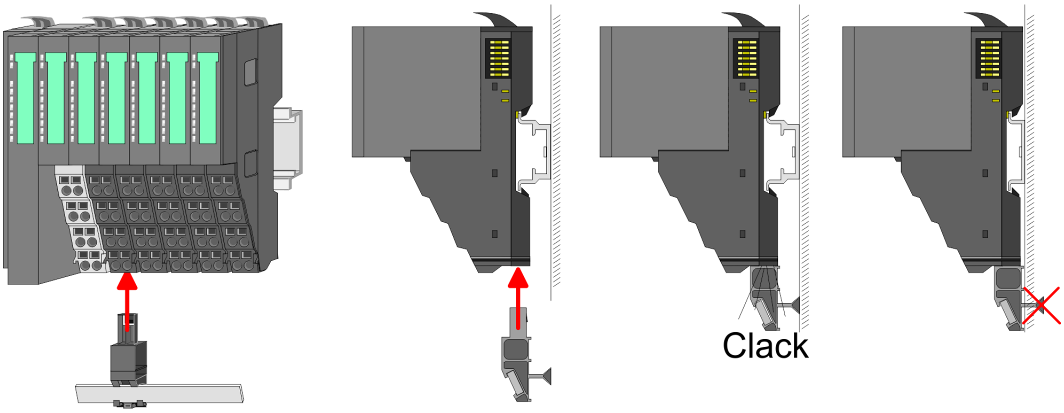
2.2.3 Accessories

Shield bus carrier

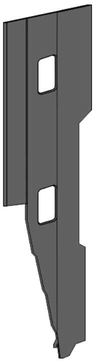
> *Please note that a shield bus carrier cannot be mounted on a 16x periphery module!*



The shield bus carrier (order no.: 000-0AB00) serves to carry the shield bus (10mm x 3mm) to connect cable shields. Shield bus carriers, shield bus and shield fixings are not in the scope of delivery. They are only available as accessories. The shield bus carrier is mounted underneath the terminal of the terminal module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.



Bus cover



With each head module, to protect the backplane bus connectors, there is a mounted bus cover in the scope of delivery. You have to remove the bus cover of the head module before mounting a System SLIO module. For the protection of the backplane bus connector you always have to mount the bus cover at the last module of your system again. The bus cover has the order no. 000-0AA00.

Coding pins

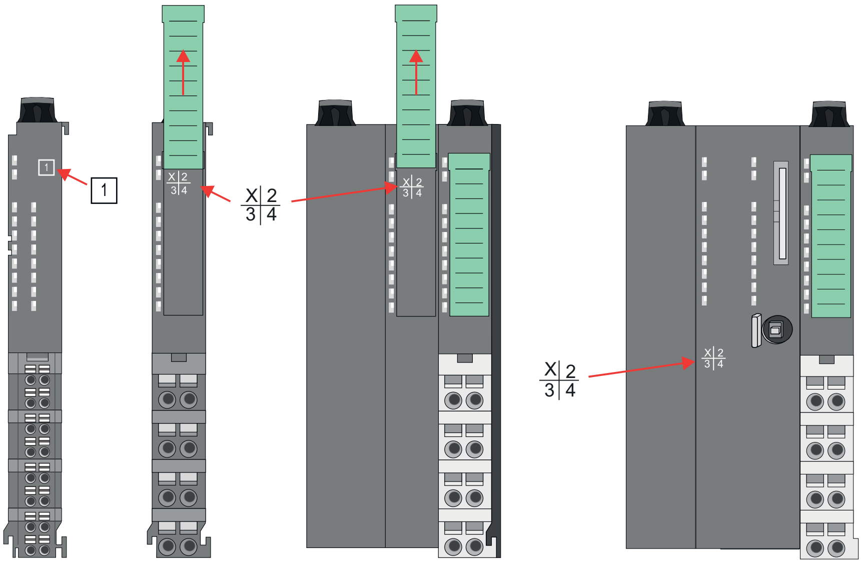
Please note that a coding pin cannot be installed on a 16x periphery module! Here you have to make sure that the associated terminal block is plugged again when the electronics unit is replaced.

There is the possibility to fix the assignment of electronic and terminal module. Here VIPA coding pins (order number 000-0AC00) can be used. The coding pin consists of a coding jack and a coding plug. By combining electronic and terminal module with coding pin, the coding jack remains in the electronic module and the coding plug in the terminal module. This ensures that after replacing the electronic module just another electronic module can be plugged with the same encoding.

2.2.4 Hardware revision

Hardware revision on the front

- The hardware revision version is printed on every System SLIO module.
- Since a System SLIO 8x periphery module consists of a terminal and electronic module, you will find a hardware revision on each of them.
- Authoritative for the hardware revision of a System SLIO module is the hardware revision of the electronic module. This is always located under the labeling strip of the corresponding electronic module.
- On modules without labeling strip, such as CPUs and 16x periphery modules, the hardware revision is printed on the front. The following figure shows the 2 variants for the representation of the hardware revision.



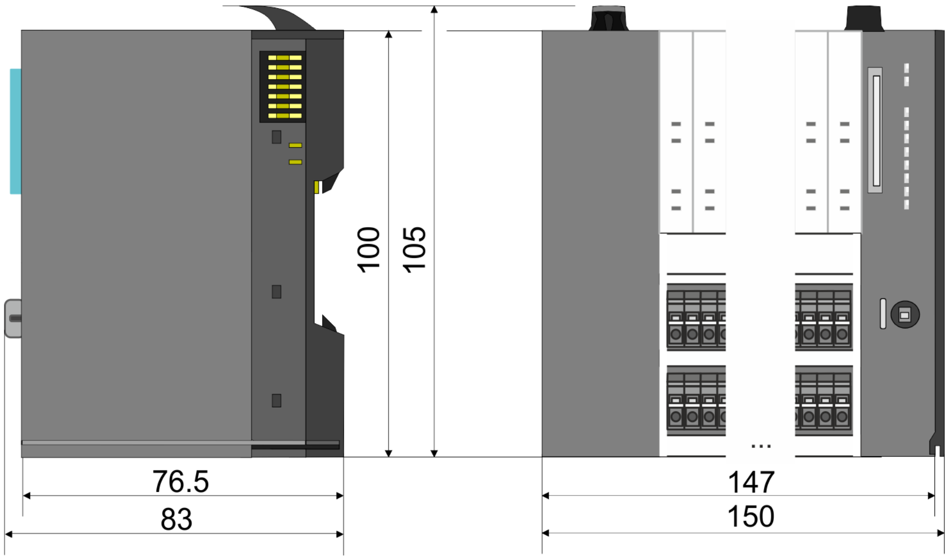
1 is marked with *'X'*.
At 16x periphery modules the hardware revision 1 is directly printed on the front.

Hardware revision via web server

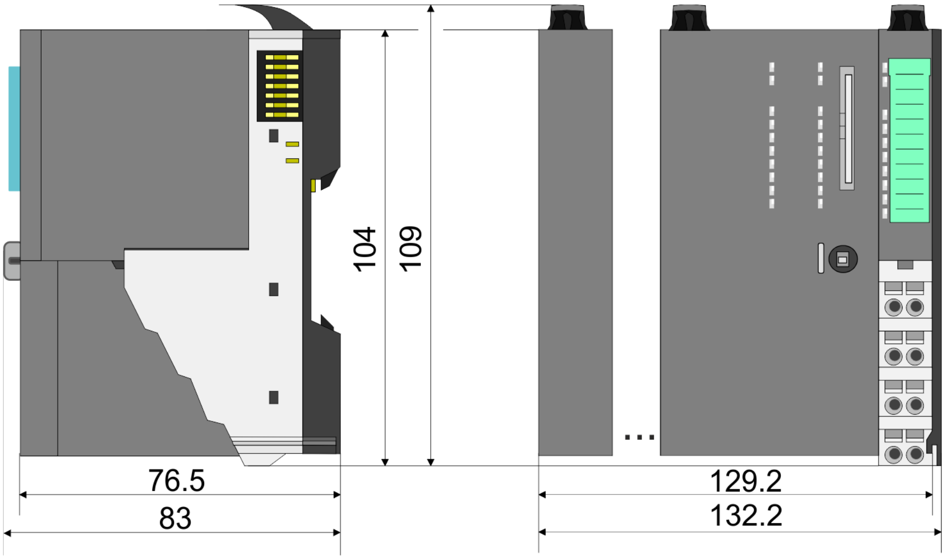
On the CPUs and some bus couplers, you can check the hardware revision *'HW Revision'* via the integrated web server.

2.3 Dimensions

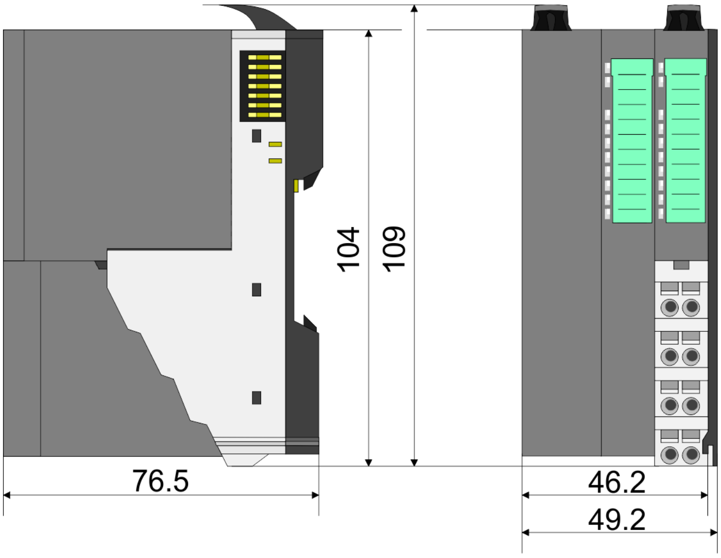
Dimensions CPU 01xC



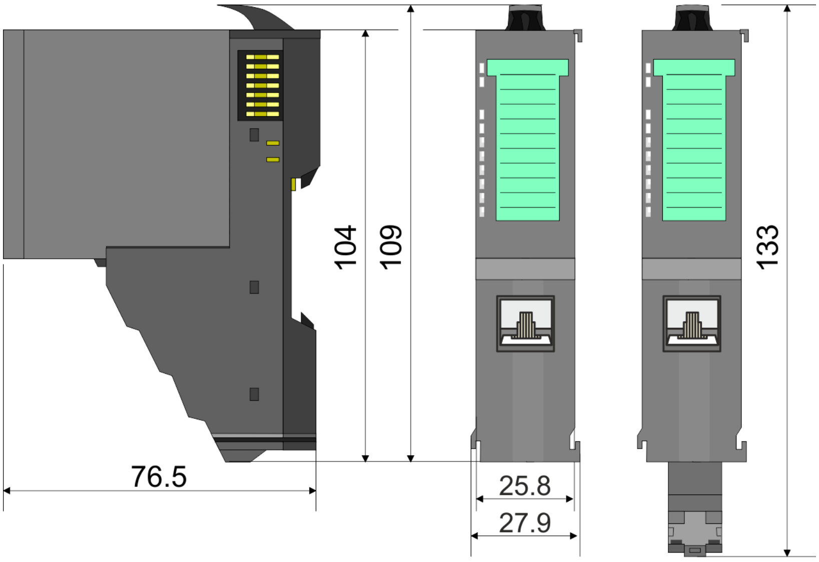
Dimensions CPU 01x



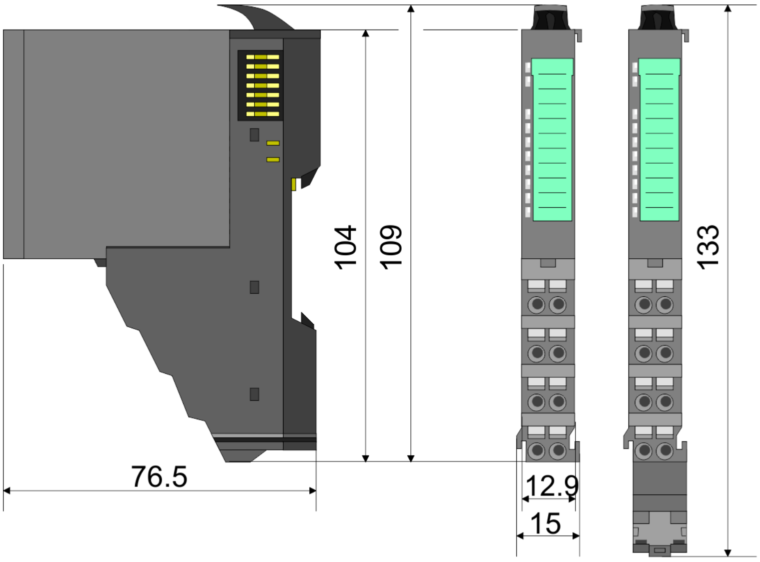
Dimensions bus coupler and line extension slave



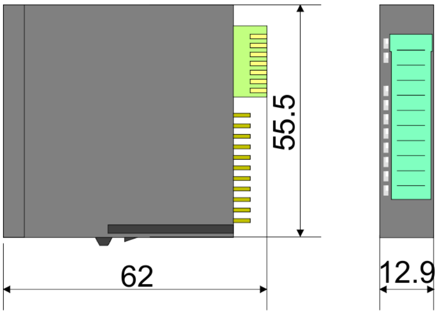
Dimensions line extension master



Dimensions 8x periphery module

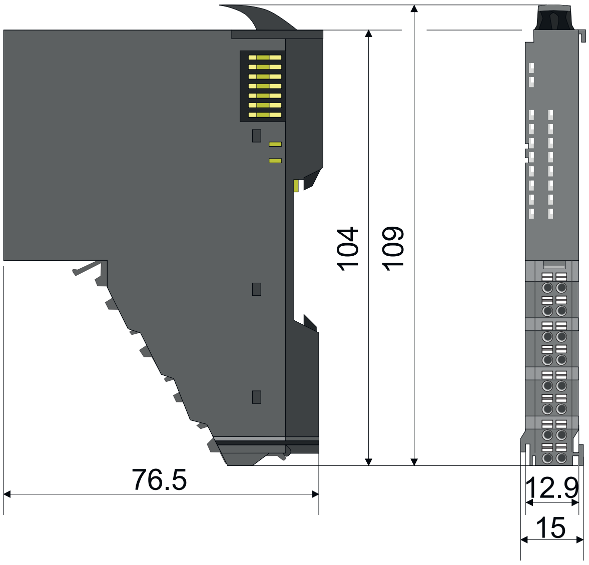


Dimensions electronic module



Dimensions in mm

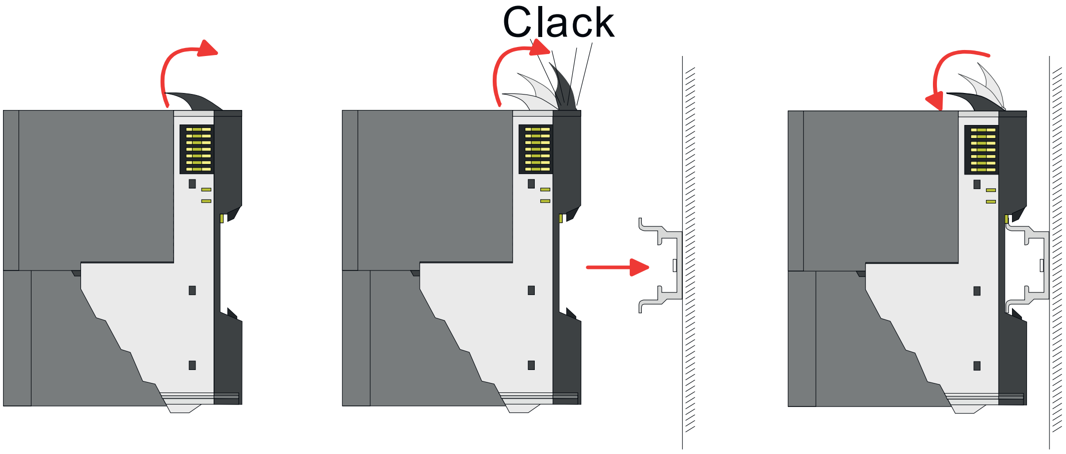
Dimensions 16x periphery module



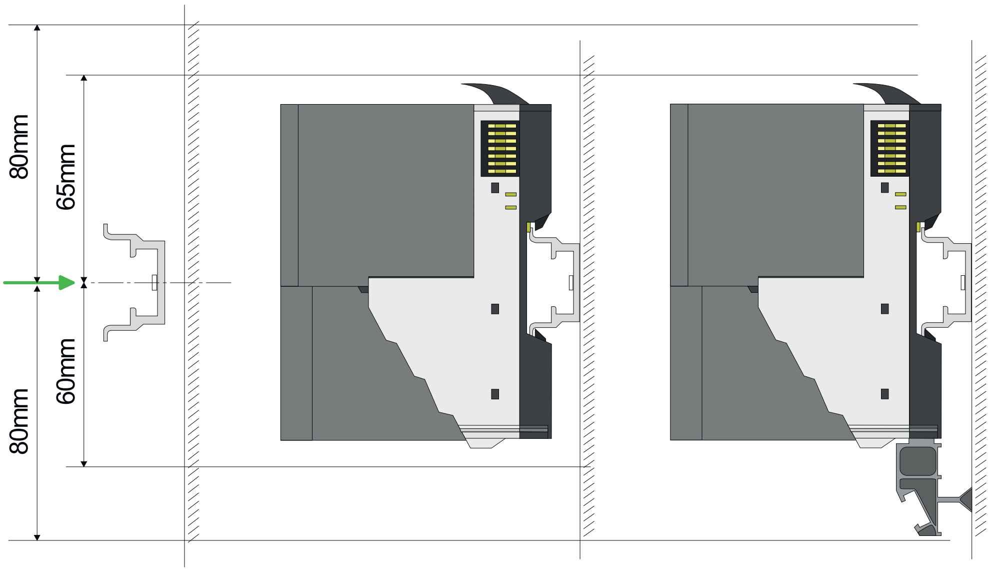
2.4 Mounting bus coupler

> ***Requirements for UL compliance use***
> - *Use for power supply exclusively SELV/PELV power supplies.*
> - *The System SLIO must be installed and operated in a housing according to IEC 61010-1 9.3.2 c).*

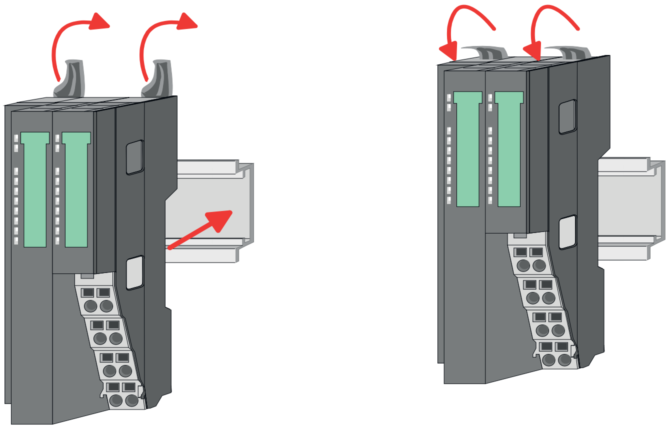
There are locking lever at the top side of the bus coupler. For mounting and demounting these locking lever are to be turned upwards until these engage. Place the bus coupler at the mounting rail. The bus coupler is fixed to the mounting rail by pushing downward the locking levers. The bus coupler is directly mounted at a mounting rail. Up to 64 modules may be mounted. The electronic and power section supply are connected via the backplane bus. Please consider here that the sum current of the electronic power supply does not exceed the maximum value of 3A. By means of the power module 007-1AB10 the current of the electronic power supply may be expanded accordingly.



Proceeding



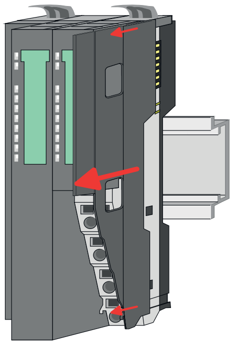
1. Mount the mounting rail! Please consider that a clearance from the middle of the mounting rail of at least 80mm above and 60mm below, respectively 80mm by deployment of shield bus carriers, exist.



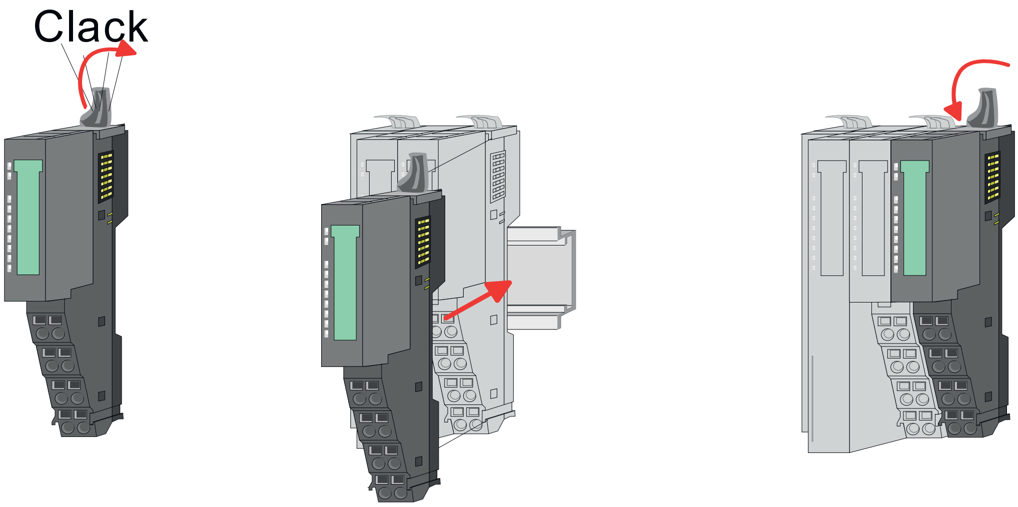
2. Turn the locking lever upwards, place the bus coupler at the mounting rail and turn the lever downward.

Mounting periphery modules

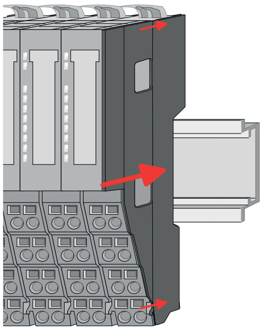
The procedure is identical for 8x and 16x periphery modules.

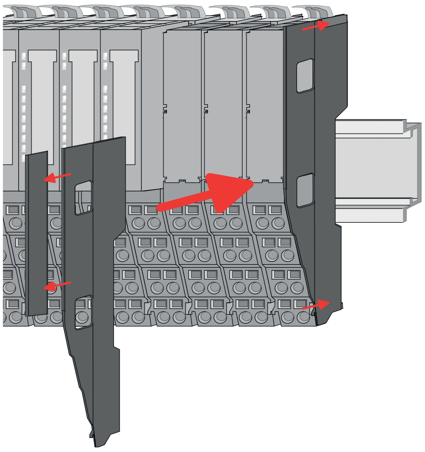


1. Before mounting the periphery modules you have to remove the bus cover at the right side of the bus coupler by pulling it forward. Keep the cover for later mounting.



2. Mount the periphery modules you want.





3. After mounting the whole system, to protect the backplane bus connectors at the last module you have to mount the bus cover, now. If the last module is a clamp module, for adaptation the upper part of the bus cover is to be removed.

2.5 Wiring

CAUTION!
Consider temperature for external cables!
Cables may experience temperature increase due to system heat dissipation. Thus the cabling specification must be chosen 5°C above ambient temperature!

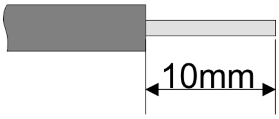
CAUTION!
Separate insulation areas!
The system is specified for SELV/PELV environment. Devices, which are attached to the system must meet theses specifications. Installation and cable routing other than SELV/PELV specification must be separated from the system’s equipment!

2.5.1 Wiring bus coupler

Terminal module terminals

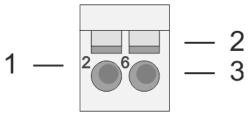
The System SLIO bus coupler have a power module integrated. Terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

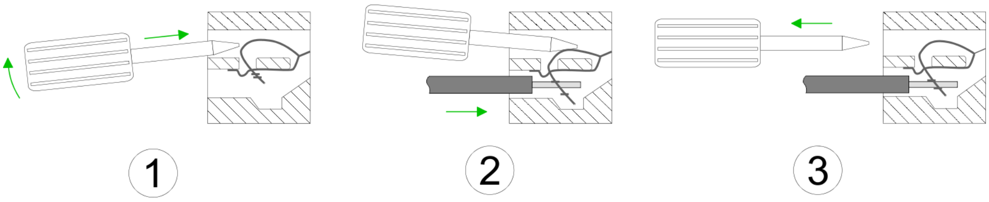


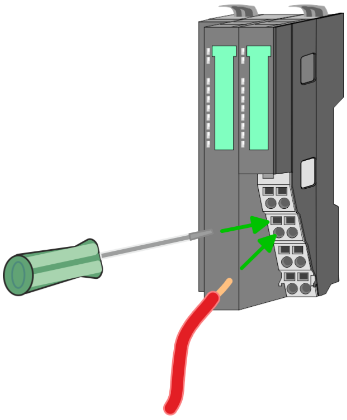
U_{max}	30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm^2 (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



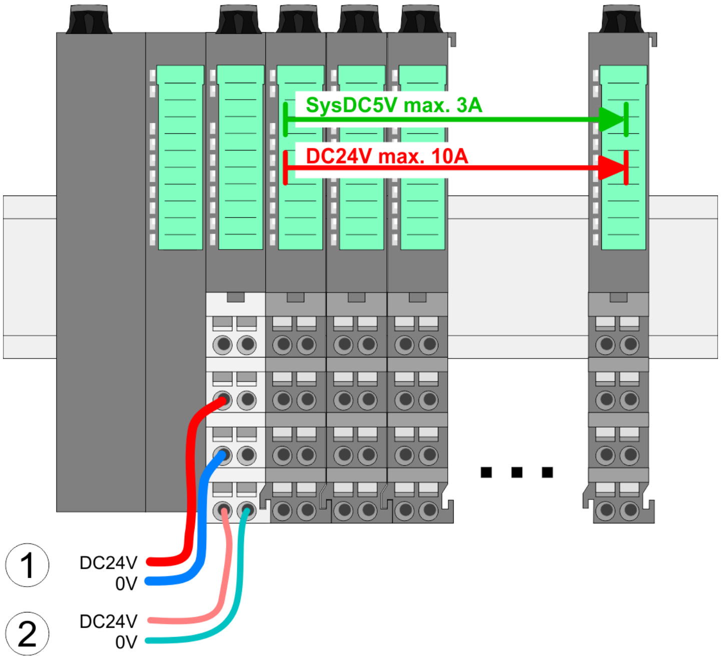
1 Pin number at the connector
2 Opening for screwdriver
3 Connection hole for wire





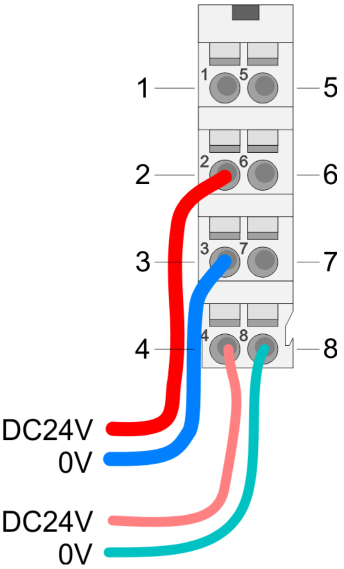
1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm^2 up to 1.5mm^2
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

Standard wiring



(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area

PM - Power module



For wires with a core cross-section of 0.08mm^2 up to 1.5mm^2.

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

CAUTION!
Since the power section supply is not internally protected, it is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected by a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!

The electronic power section supply is internally protected against higher voltage by fuse. The fuse is within the power module. If the fuse releases, its electronic module must be exchanged!

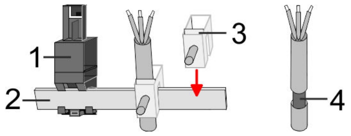
Fusing

- The power section supply is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected with a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!
- It is recommended to externally protect the electronic power supply for bus coupler and I/O area with a 2A fuse (fast) respectively by a line circuit breaker 2A characteristics Z.
- The electronic power supply for the I/O area of the power module 007-1AB10 should also be externally protected with a 1A fuse (fast) respectively by a line circuit breaker 1A characteristics Z.

State of the electronic power supply via LEDs

After PowerON of the System SLIO the LEDs RUN respectively MF get on so far as the sum current does not exceed 3A. With a sum current greater than 3A the LEDs may not be activated. Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

Shield attachment

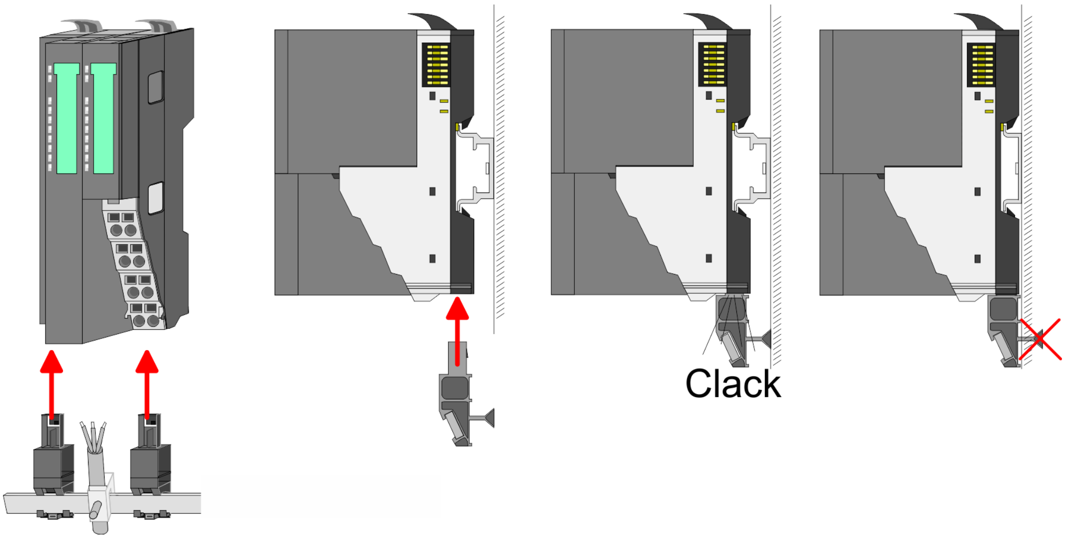


1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.

2. Put your shield bus into the shield bus carrier.



3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

2.5.2 Wiring 8x periphery modules

Terminal module terminals

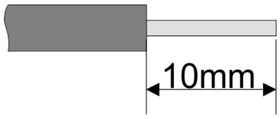
CAUTION!

Do not connect hazardous voltages!

If this is not explicitly stated in the corresponding module description, hazardous voltages are not allowed to be connected to the corresponding terminal module!

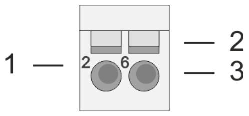
With wiring the terminal modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

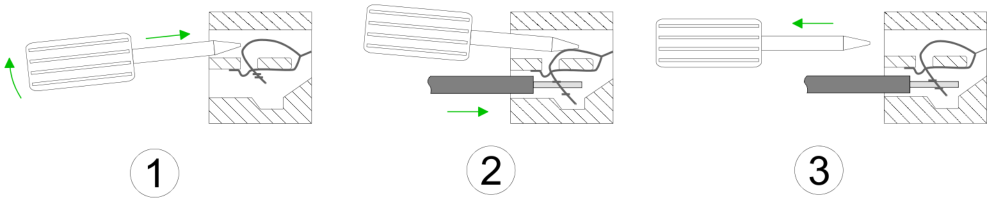


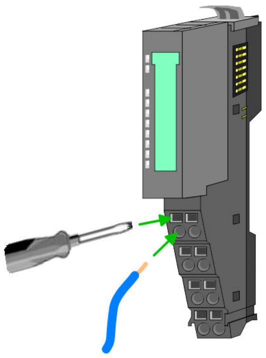
U_{max}	240V AC / 30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm^2 (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



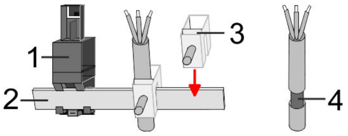
1 Pin number at the connector
2 Opening for screwdriver
3 Connection hole for wire





1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm^2 up to 1.5mm^2
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

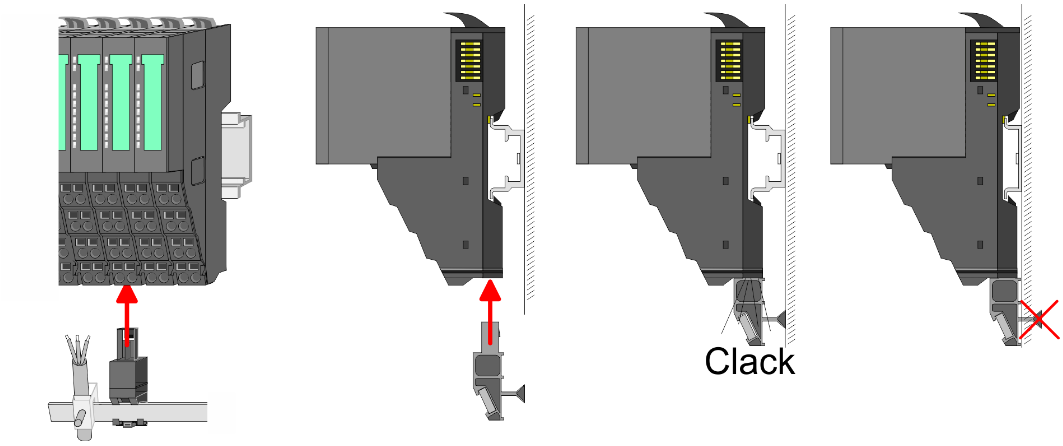
Shield attachment



1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO 8x periphery module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.
2. Put your shield bus into the shield bus carrier.



3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

2.5.3 Wiring 16x periphery modules

Terminal block connectors

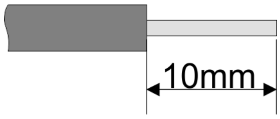
CAUTION!

Do not connect hazardous voltages!

If this is not explicitly stated in the corresponding module description, hazardous voltages are not allowed to be connected to the corresponding terminal block!

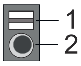
The 16x periphery module has a removable terminal block for wiring. With the wiring of the terminal block a "push-in" spring-clip technique is used. This allows a quick and easy connection of your signal and supply lines. The clamping off takes place by means of a screwdriver.

Data



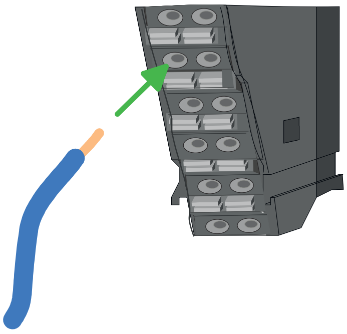
U_{max}	30V DC
I_{max}	10A
Cross section solid wire	0,25 ... 0,75mm^2
Cross section with ferrule	0,14 ... 0,75mm^2
AWG	24 ... 16
Stripping length	10mm

Wiring procedure



1 Release area
2 Connection hole for wire

Insert wire

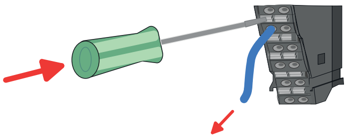


The wiring happens without a tool.

1. Determine according to the casing labelling the connection position.
2. Insert through the round connection hole of the according contact your prepared wire until it stops, so that it is fixed.

   ⇨ By pushing the contact spring opens, thus ensuring the necessary contact pressure.

Remove wire



The wire is to be removed by means of a screwdriver with 2.5mm blade width.

1. Press with your screwdriver vertically at the release button.

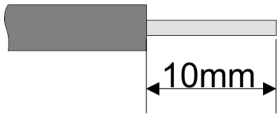
   ⇨ The contact spring releases the wire.
2. Pull the wire from the round hole.

2.5.4 Wiring power modules

Terminal module terminals

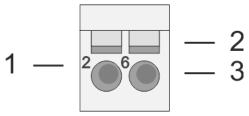
Power modules are either integrated to the head module or may be installed between the periphery modules. With power modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

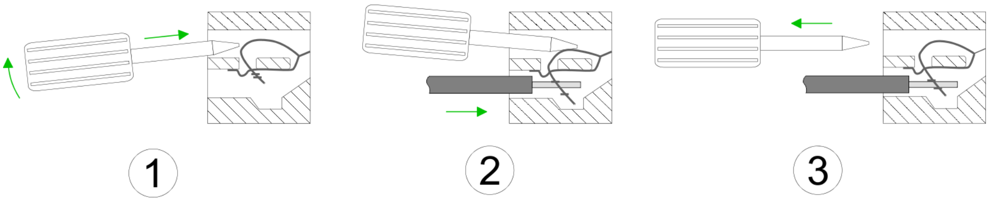


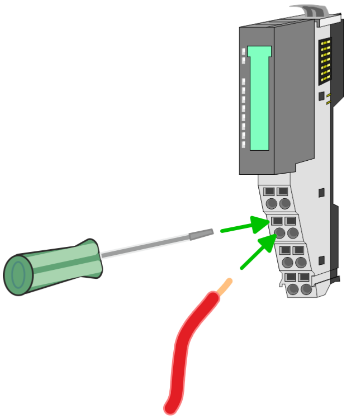
U_{max}	30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm² (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



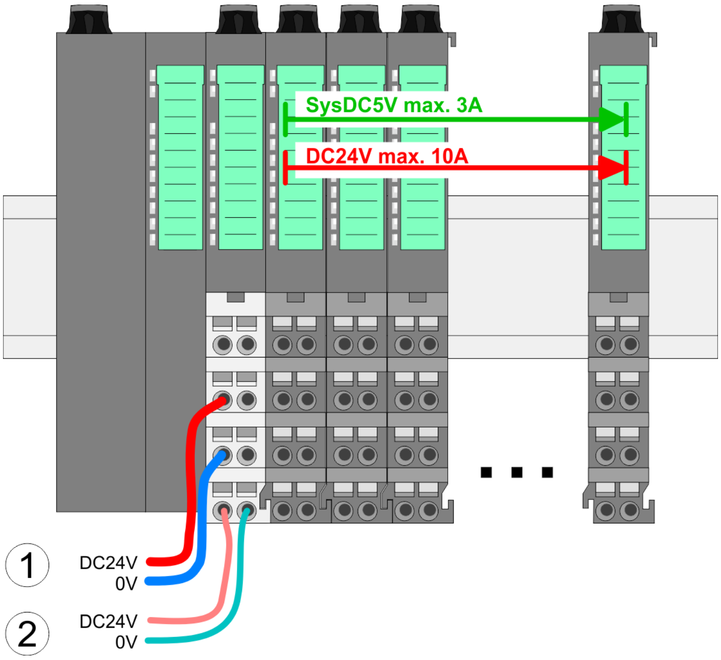
1 Pin number at the connector
2 Opening for screwdriver
3 Connection hole for wire





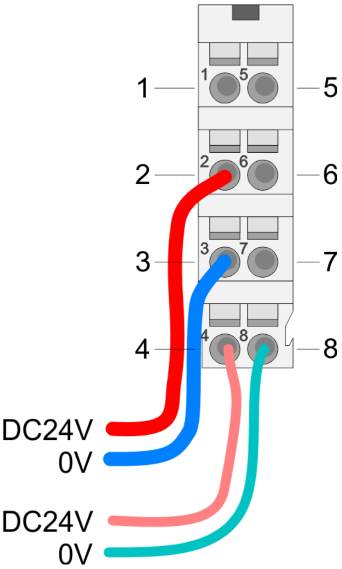
1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm^2 up to 1.5mm^2
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

Standard wiring



(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area

PM - Power module



For wires with a core cross-section of 0.08mm^2 up to 1.5mm^2.

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

CAUTION!
Since the power section supply is not internally protected, it is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected by a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!

The electronic power section supply is internally protected against higher voltage by fuse. The fuse is within the power module. If the fuse releases, its electronic module must be exchanged!

Fusing

- The power section supply is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected with a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!
- It is recommended to externally protect the electronic power supply for head modules and I/O area with a 2A fuse (fast) respectively by a line circuit breaker 2A characteristics Z.
- The electronic power supply for the I/O area of the power module 007-1AB10 should also be externally protected with a 1A fuse (fast) respectively by a line circuit breaker 1A characteristics Z.

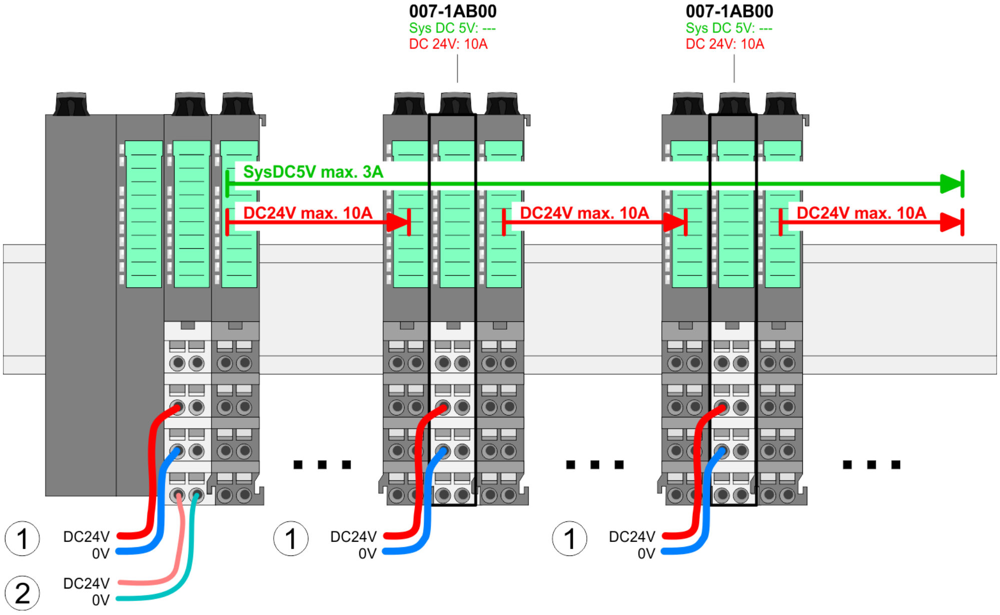
State of the electronic power supply via LEDs

After PowerON of the System SLIO the LEDs RUN respectively MF get on so far as the sum current does not exceed 3A. With a sum current greater than 3A the LEDs may not be activated. Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

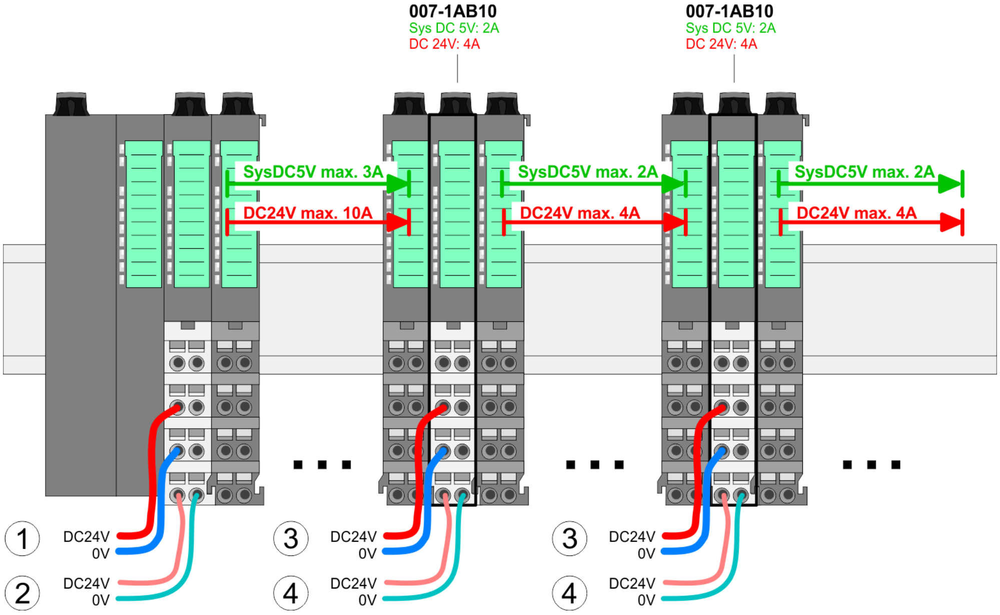
Deployment of the power modules

- If the 10A for the power section supply is no longer sufficient, you may use the power module with the order number 007-1AB00. So you have also the possibility to define isolated groups.
- The power module with the order number 007-1AB10 is to be used if the 3A for the electronic power supply at the backplane bus is no longer sufficient. Additionally you get an isolated group for the DC 24V power section supply with max. 4A.
- By placing the power module 007-1AB10 at the following backplane bus modules may be placed with a sum current of max. 2A. Afterwards a power module is to be placed again. To secure the power supply, the power modules may be mixed used.

Power module 007-1AB00

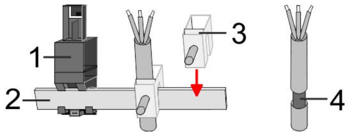


Power module 007-1AB10



(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area
(3) DC 24V for power section supply I/O area (max. 4A)
(4) DC 24V for electronic power supply I/O area

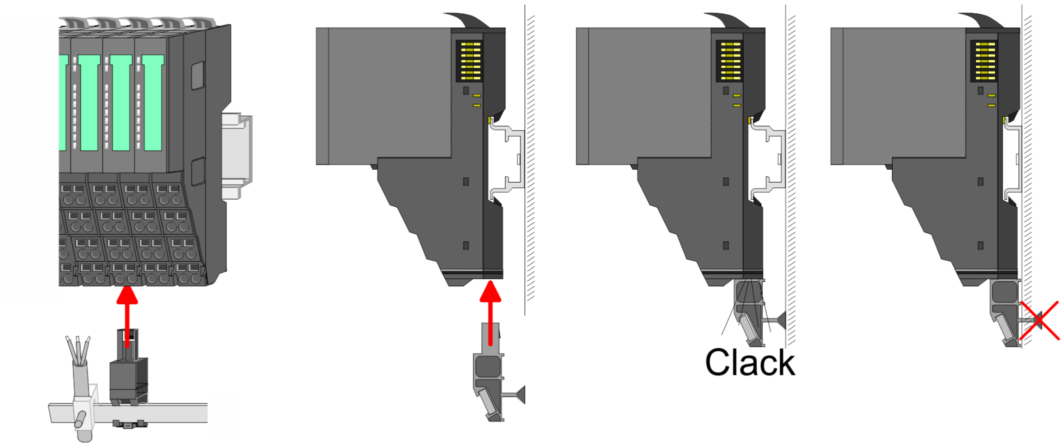
Shield attachment



1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO 8x periphery module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.
2. Put your shield bus into the shield bus carrier.



3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

2.6 Demounting

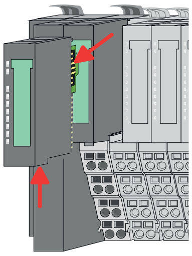
2.6.1 Demounting bus coupler

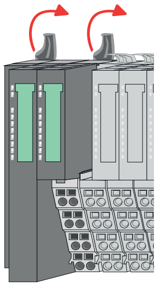
Proceeding

CAUTION!
Bus interface and power module may not be separated! Here you may only exchange the electronic module!

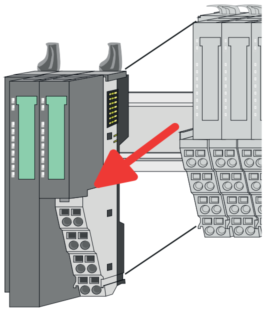
1. Power-off your system.
2. Remove if exists the wiring of the bus coupler.
3. *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

   Press the unlocking lever at the lower side of the just mounted right module near the bus coupler and pull it forward.



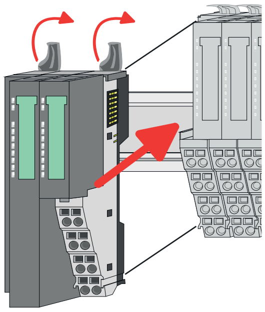


4. Turn all the locking lever of the bus coupler to be exchanged upwards.



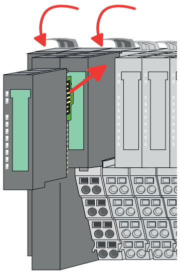
5. Pull the bus coupler forward.

6. For mounting turn all the locking lever of the bus coupler to be exchanged upwards.



7. To mount the bus coupler put it to the left periphery module and push it, guided by the stripes, to the mounting rail.

8. Turn all the locking lever downward, again.



9. Plug again the electronic module, which you have removed before.

10. Wire your bus coupler.

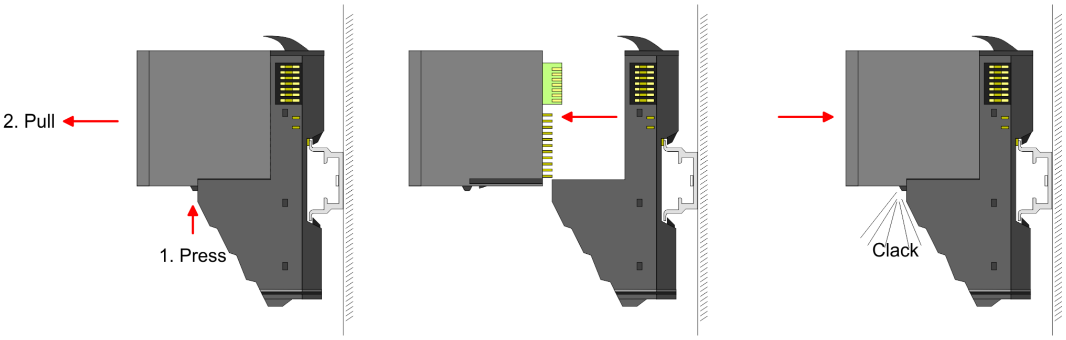
⇨ Now you can bring your system back into operation.

2.6.2 Demounting 8x periphery modules

Proceeding

Exchange of an electronic module

1. Power-off your system.

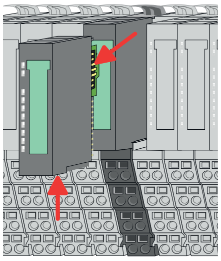


2. For the exchange of a electronic module, the electronic module may be pulled forward after pressing the unlocking lever at the lower side of the module.
3. For installation plug the new electronic module guided by the strips at the lower side until this engages to the terminal module.

⇨ Now you can bring your system back into operation.

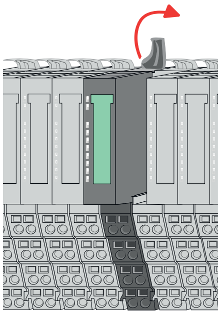
Exchange of a periphery module

1. Power-off your system.
2. Remove if exists the wiring of the module.
3. *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

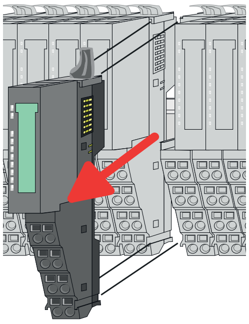


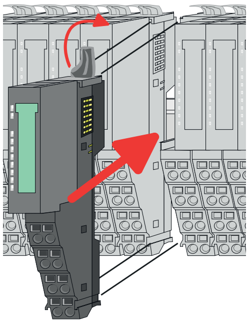
Press the unlocking lever at the lower side of the just mounted right module and pull it forward.

4. Turn the locking lever of the module to be exchanged upwards.



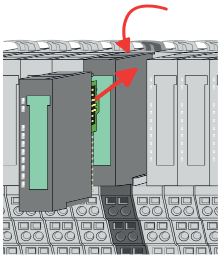
5. Pull the module.
6. For mounting turn the locking lever of the module to be mounted upwards.





7. To mount the module put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.

8. Turn the locking lever downward, again.



9. Plug again the electronic module, which you have removed before.

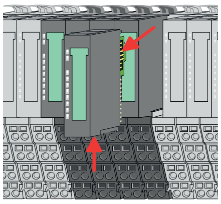
10. Wire your module.

⇨ Now you can bring your system back into operation.

Exchange of a module group

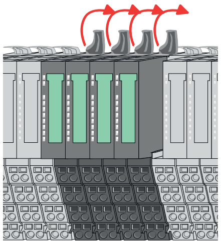
1. Power-off your system.

2. Remove if exists the wiring of the module group.

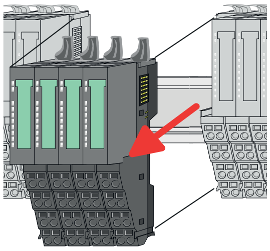


3. *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

Press the unlocking lever at the lower side of the just mounted right module near the module group and pull it forward.

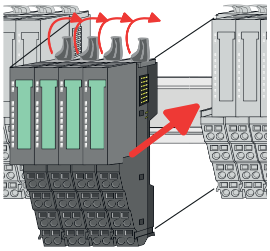


4. Turn all the locking lever of the module group to be exchanged upwards.



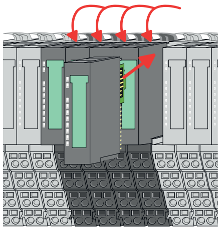
5. Pull the module group forward.

6. For mounting turn all the locking lever of the module group to be mounted upwards.



7. To mount the module group put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.

8. Turn all the locking lever downward, again.



9. Plug again the electronic module, which you have removed before.

10. Wire your module group.

⇨ Now you can bring your system back into operation.

2.6.3 Demounting 16x periphery modules

Proceeding

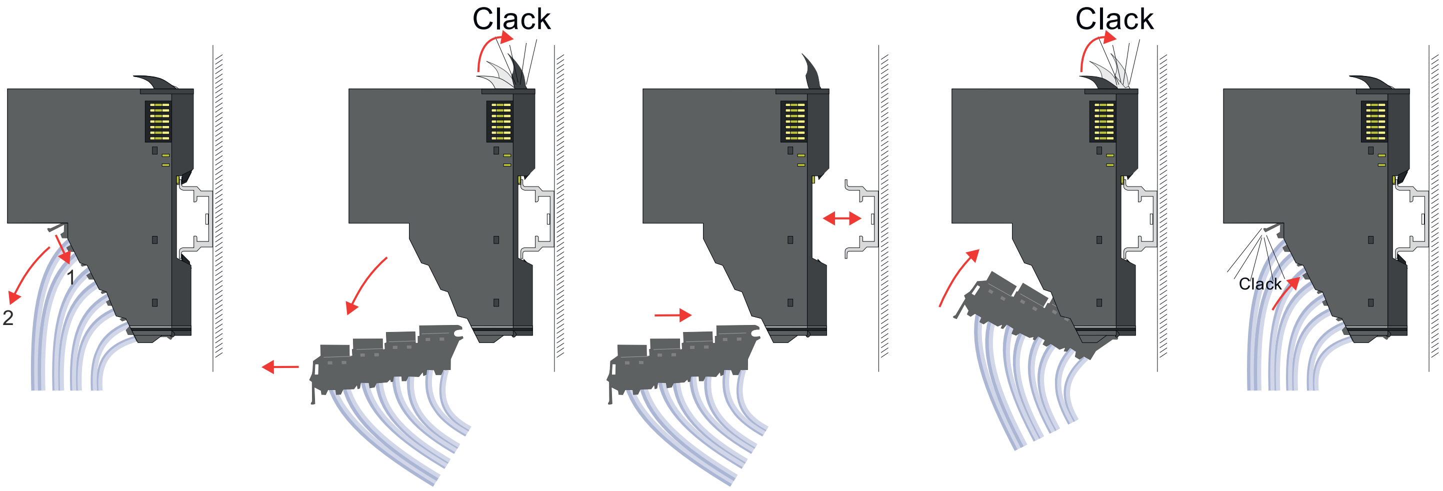
Exchange of an electronic unit

1. Power-off your system.

2. To replace an electronic unit, you can push down and pull off the terminal block after releasing the lock.

   To mount the terminal block, place it horizontally on the lower side of the electronic unit and push it towards the electronic unit until it clicks into place.

   ⇨ Now you can bring your system back into operation.



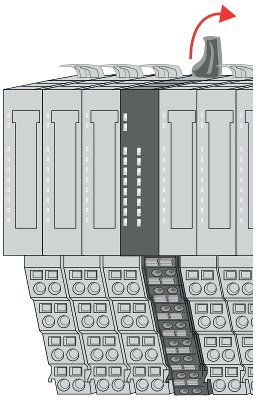
Exchange of a 16x periphery module

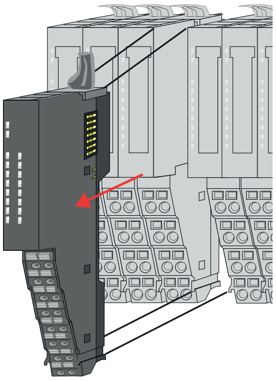
1. Power-off your system.

2. Remove if exists the wiring of the module respectively the wired terminal block.

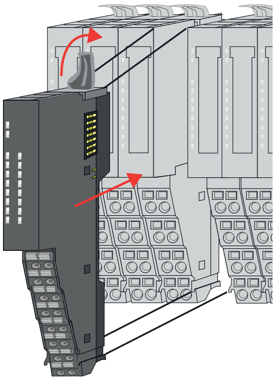
3. *In contrast to 8x periphery modules, you can directly demount and mount 16x periphery modules.*

   Turn the locking lever of the module to be exchanged upwards.

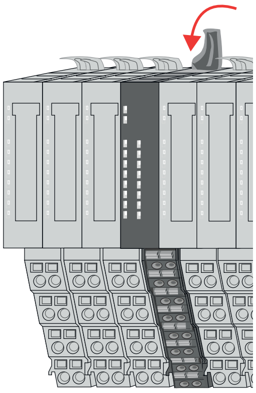




4. Pull the module.
5. For mounting turn the locking lever of the module to be mounted upwards.



6. To mount the module put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.



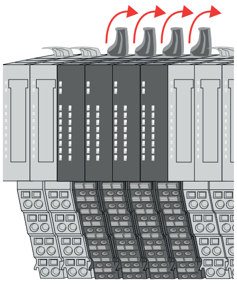
7. Turn the locking lever downward, again.
8. Wire your module respectively plug the wired terminal block again.

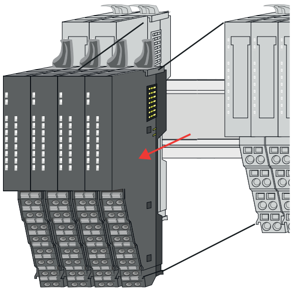
⇨ Now you can bring your system back into operation.

Exchange of a module group

1. Power-off your system.
2. Remove if exists the wiring of the module group respectively the wired terminal blocks.
3. *In contrast to 8x periphery modules, you can directly demount and mount 16x periphery modules.*

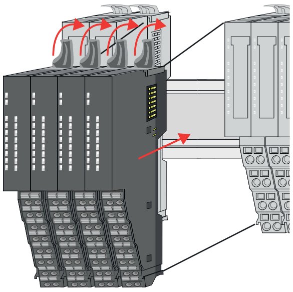
Turn all the locking lever of the module group to be exchanged upwards.



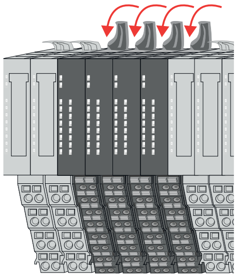


4. Pull the module group forward.

5. For mounting turn all the locking lever of the module group to be mounted upwards.



6. To mount the module group put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.



7. Turn all the locking lever downward, again.

8. Wire your module group respectively plug the wired terminal blocks again.

⇨ Now you can bring your system back into operation.

2.6.4 Easy Maintenance

Overview

Easy Maintenance means the support for adding and removing modules during operation without having to restart the system. There is the following distinction:

- *Easy Maintenance* at *active* bus connection
- *Easy Maintenance* at *passive* bus connection

↳ *Chap. 4.5.6 ‘Easy Maintenance’ page 68*

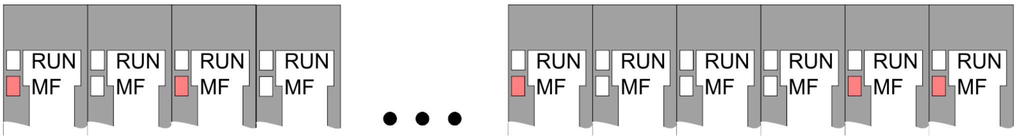
2.7 Trouble shooting - LEDs

General

Each module has the LEDs RUN and MF on its front side. Errors or incorrect modules may be located by means of these LEDs.

In the following illustrations flashing LEDs are marked by ☼.

Sum current of the electronic power supply exceeded

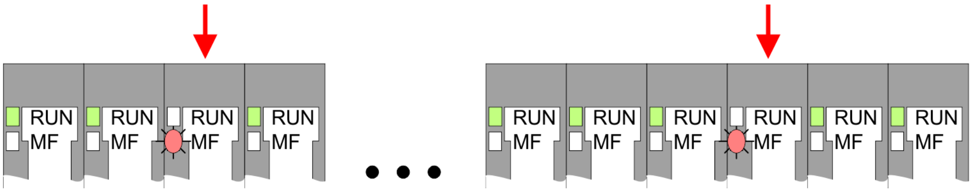


Behaviour: After PowerON the RUN LED of each module is off and the MF LED of each module is sporadically on.

Reason: The maximum current for the electronic power supply is exceeded.

Remedy: As soon as the sum current of the electronic power supply is exceeded, always place the power module 007-1AB10. ➪ *Chap. 2.5.4 'Wiring power modules' page 26*

Error in configuration

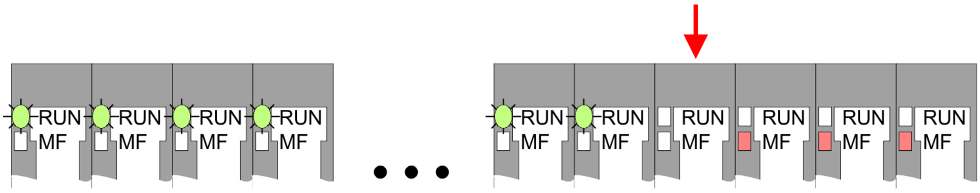


Behaviour: After PowerON the MF LED of one module respectively more modules flashes. The RUN LED remains off.

Reason: At this position a module is placed, which does not correspond to the configured module.

Remedy: Match configuration and hardware structure.

Module failure



Behaviour: After PowerON all of the RUN LEDs up to the defective module are flashing. With all following modules the MF LED is on and the RUN LED is off.

Reason: The module on the right of the flashing modules is defective.

Remedy: Replace the defective module.

2.8 Installation guidelines

General

The installation guidelines contain information about the interference free deployment of a PLC system. There is the description of the ways, interference may occur in your PLC, how you can make sure the electromagnetic compatibility (EMC), and how you manage the isolation.

What does EMC mean?

Electromagnetic compatibility (EMC) means the ability of an electrical device, to function error free in an electromagnetic environment without being interfered respectively without interfering the environment.

The VIPA components are developed for the deployment in industrial environments and meets high demands on the EMC. Nevertheless you should project an EMC planning before installing the components and take conceivable interference causes into account.

Possible interference causes

Electromagnetic interferences may interfere your control via different ways:

- Electromagnetic fields (RF coupling)
- Magnetic fields with power frequency
- Bus system
- Power supply
- Protected earth conductor

Depending on the spreading medium (lead bound or lead free) and the distance to the interference cause, interferences to your control occur by means of different coupling mechanisms.

There are:

- galvanic coupling
- capacitive coupling
- inductive coupling
- radiant coupling

Basic rules for EMC

In the most times it is enough to take care of some elementary rules to guarantee the EMC. Please regard the following basic rules when installing your PLC.

- Take care of a correct area-wide grounding of the inactive metal parts when installing your components.
  - Install a central connection between the ground and the protected earth conductor system.
  - Connect all inactive metal extensive and impedance-low.
  - Please try not to use aluminium parts. Aluminium is easily oxidizing and is therefore less suitable for grounding.
- When cabling, take care of the correct line routing.
  - Organize your cabling in line groups (high voltage, current supply, signal and data lines).
  - Always lay your high voltage lines and signal respectively data lines in separate channels or bundles.
  - Route the signal and data lines as near as possible beside ground areas (e.g. suspension bars, metal rails, tin cabinet).

- Proof the correct fixing of the lead isolation.
  - Data lines must be laid isolated.
  - Analog lines must be laid isolated. When transmitting signals with small amplitudes the one sided laying of the isolation may be favourable.
  - Lay the line isolation extensively on an isolation/protected earth conductor rail directly after the cabinet entry and fix the isolation with cable clamps.
  - Make sure that the isolation/protected earth conductor rail is connected impedance-low with the cabinet.
  - Use metallic or metallised plug cases for isolated data lines.
- In special use cases you should appoint special EMC actions.
  - Consider to wire all inductivities with erase links.
  - Please consider luminescent lamps can influence signal lines.
- Create a homogeneous reference potential and ground all electrical operating supplies when possible.
  - Please take care for the targeted employment of the grounding actions. The grounding of the PLC serves for protection and functionality activity.
  - Connect installation parts and cabinets with your PLC in star topology with the isolation/protected earth conductor system. So you avoid ground loops.
  - If there are potential differences between installation parts and cabinets, lay sufficiently dimensioned potential compensation lines.

Isolation of conductors

Electrical, magnetically and electromagnetic interference fields are weakened by means of an isolation, one talks of absorption. Via the isolation rail, that is connected conductive with the rack, interference currents are shunt via cable isolation to the ground. Here you have to make sure, that the connection to the protected earth conductor is impedance-low, because otherwise the interference currents may appear as interference cause.

When isolating cables you have to regard the following:

- If possible, use only cables with isolation tangle.
- The hiding power of the isolation should be higher than 80%.
- Normally you should always lay the isolation of cables on both sides. Only by means of the both-sided connection of the isolation you achieve high quality interference suppression in the higher frequency area. Only as exception you may also lay the isolation one-sided. Then you only achieve the absorption of the lower frequencies. A one-sided isolation connection may be convenient, if:
  - the conduction of a potential compensating line is not possible.
  - analog signals (some mV respectively µA) are transferred.
  - foil isolations (static isolations) are used.
- With data lines always use metallic or metallised plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!
- At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.
- To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.
- Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to your PLC and don't lay it on there again!

CAUTION!
Please regard at installation!
At potential differences between the grounding points, there may be a compensation current via the isolation connected at both sides.

Remedy: Potential compensation line

2.9 General data for the System SLIO

Conformity and approval		
Conformity		
CE	2014/35/EU	Low-voltage directive
	2014/30/EU	EMC directive
Approval		
UL	-	Refer to Technical data
Others		
RoHS	2011/65/EU	Restriction of the use of certain hazardous substances in electrical and electronic equipment

Protection of persons and device protection		
Type of protection	-	IP20
Electrical isolation		
to the field bus	-	electrically isolated
to the process level	-	electrically isolated
Insulation resistance	-	-
Insulation voltage to reference earth		
Inputs / outputs	-	AC / DC 50V, test voltage AC 500V
Protective measures	-	against short circuit

Environmental conditions to EN 61131-2		
Climatic		
Storage / transport	EN 60068-2-14	-25…+70°C
Operation		
Horizontal installation hanging	EN 61131-2	0…+60°C
Horizontal installation lying	EN 61131-2	0…+55°C
Vertical installation	EN 61131-2	0…+50°C
Air humidity	EN 60068-2-30	RH1 (without condensation, rel. humidity 10…95%)
Pollution	EN 61131-2	Degree of pollution 2
Installation altitude max.	-	2000m
Mechanical		
Oscillation	EN 60068-2-6	1g, 9Hz ... 150Hz
Shock	EN 60068-2-27	15g, 11ms

Mounting conditions		
Mounting place	-	In the control cabinet
Mounting position	-	Horizontal and vertical

EMC	Standard		Comment
Emitted interference	EN 61000-6-4		Class A (Industrial area)
Noise immunity zone B	EN 61000-6-2		Industrial area
		EN 61000-4-2	ESD 8kV at air discharge (degree of severity 3), 4kV at contact discharge (degree of severity 2)
		EN 61000-4-3	HF field immunity (casing) 80MHz … 1000MHz, 10V/m, 80% AM (1kHz) 1.4GHz ... 2.0GHz, 3V/m, 80% AM (1kHz) 2GHz ... 2.7GHz, 1V/m, 80% AM (1kHz)
		EN 61000-4-6	HF conducted 150kHz … 80MHz, 10V, 80% AM (1kHz)
		EN 61000-4-4	Burst, degree of severity 3
		EN 61000-4-5	Surge, degree of severity 3[1]

1) Due to the high-energetic single pulses with Surge an appropriate external protective circuit with lightning protection elements like conductors for lightning and overvoltage is necessary.

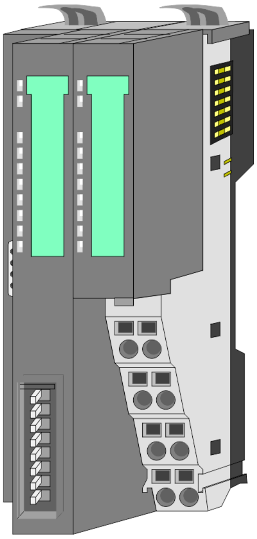
3 Hardware description

3.1 Properties

053-1EC01

The EtherCAT coupler IM 053EC serves for an easy connection of decentralized periphery to EtherCAT. EtherCAT offers real time Ethernet technology on I/O level.

- EtherCAT coupler for max. 64 peripheral modules
- Ethernet-based field bus system with high real-time capability
- Supports the application profile
  - CoE (CANopen over EtherCAT)
  - EoE (Ethernet over EtherCAT)
  - FoE (File-Access over EtherCAT)
- Supports safety modules (FSoE)
- Online configuration via master system
- Various diagnostics functions
- RJ45 jack 100BaseTX
- Operation Modes
  - FreeRun
  - SyncManager-Event
  - Distributed Clock
- HotConnect
  - Station Alias address → via SII parameterizable
  - Explicit Device ID → via dip switch
- Easy Maintenance
- Display of substitute values possible
- Integrated Web server
- LEDs for status display

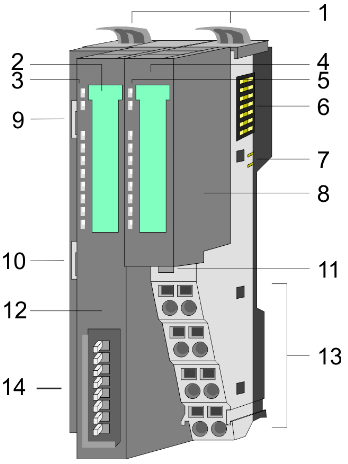


Ordering data

Type	Order number	Description
IM 053EC	053-1EC01	EtherCAT coupler for System SLIO

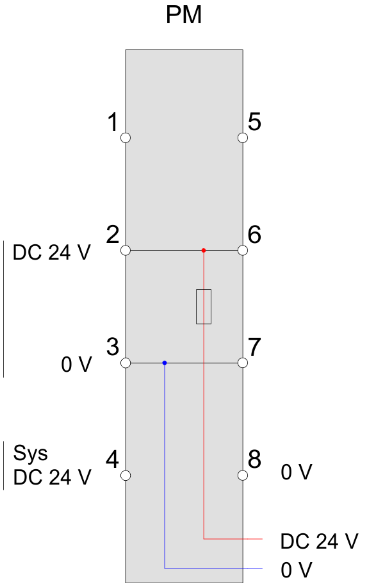
3.2 Structure

053-1EC01

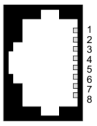


1 Locking lever terminal module
2 Labeling strip bus interface
3 LED status indication bus interface
4 Labelling strip power module
5 LED status indication power module
6 Backplane bus
7 DC 24V power section supply
8 Power module
9 EtherCAT RJ45 bus interface "IN"
10 EtherCAT RJ45 bus interface “OUT"
11 Unlocking lever power module
12 Bus interface
13 Terminal power module
14 Address switch

3.2.1 Interfaces



2x RJ45
EtherCAT

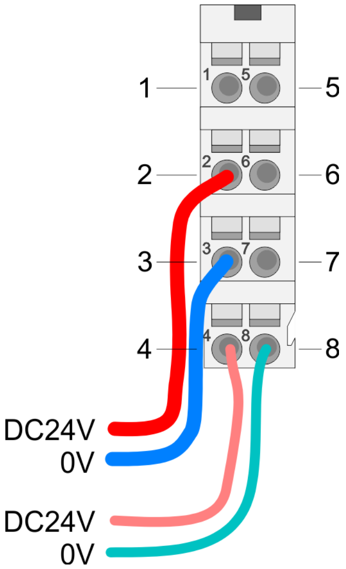


① Transmit +
② Transmit -
③ Receive +
④ -
⑤ -
⑥ Receive -
⑦ -
⑧ -

CAUTION!
Bus interface and power module of the bus coupler may not be separated!

Here you may only exchange the electronic module!

PM - Power module



For wires with a core cross-section of 0.08mm^2 up to 1.5mm^2.

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

Interface for EtherCAT communication

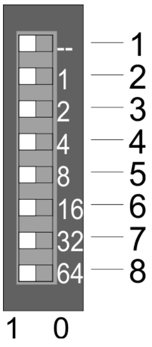
8pin RJ45 jacks:

- An EtherCAT network always consists of a master and an various number of EtherCAT salves (coupler).
- Each EtherCAT slave has an "IN" and "OUT" RJ45 jack. The arriving EtherCAT cable from the direction of the master is to be connected to the "IN" (Port 0) jack. The "OUT" (Port 1) jack is to be connected to the succeeding EtherCAT station. With the respective last EtherCAT station the "OUT" jack remains free.
- EtherCAT uses Ethernet as transfer medium. Standard CAT5 cables are used. Here distances of about 100m between 2 stations are possible.
- Only EtherCAT components may be used in an EtherCAT network. For topologies, which depart from the line structure, the corresponding EtherCAT components are necessary.
- Hubs may not be used.

Possible telegram loss during network interruption

Currently, the EtherCAT slave requires longer (about 1s) to detect that a pulled EtherCAT cable is reconnected. As a result, a slave in the network continues its transmission process too early and the telegrams are lost.

3.2.2 Address switch

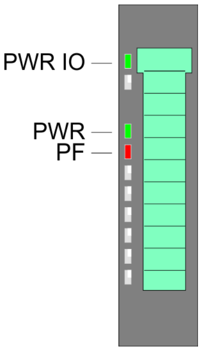


To uniquely identify the EtherCAT slave, this must be assigned a *Device ID*. This address may only be assigned once at the bus. For the assignment of an HotConnect address via Explicit Device ID ☞ *Chap. 4.2 'Setting the HotConnect address' page 53*

Pos.	Value	Example	
		State	Address
1	not used	---	1+2+32=35 Address: 35
2	1	1	
3	2	1	
4	4	0	
5	8	0	
6	16	0	
7	32	1	
8	64	0	

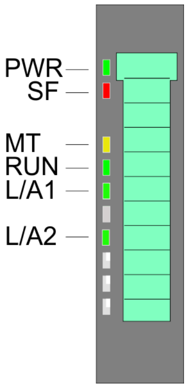
3.2.3 LEDs

LEDs power module



PWR IO 🟩 green	PWR 🟩 green	PF 🟥 red	Description
🟩	X	□	Power section supply OK
🟩	🟩	□	Electronic section supply OK
X	X	🟥	Fuse electronic section supply defective
not relevant: X			

Status indication bus interface



LED		Description
PWR	🟩 green	Bus interface is power supplied
SF	◩ red	■ 1 flash per s: With state change due to an error e.g. a module is removed ■ 2 flashes per s: EtherCAT-Timeout (watchdog) e.g. EtherCAT master is missing ■ Blinking with 2.5Hz: Error in configuration ■ Blinks alternately with MT at firmware update
MT	◩ yellow	Blinks alternately with SF at firmware update
RUN	□	Off: Bus coupler in state Init
	🟩 green	Bus coupler in state Operational
	◩ green	■ Blinking with 2.5Hz: Bus coupler in state Pre-Operational ■ SingleFlash: Bus coupler in state Safe-Operational
L/A1	□	Off: no communication to the preceding station
	◩ green	Blinking with 10Hz: Preceding EtherCAT station is connected
L/A2	□	Off: no communication to the succeeding station
	◩ green	Blinking with 10Hz: Succeeding EtherCAT station is connected

3.3 Address area

Input area

Max. number of possible data are 20byte. ↻ *Chap. 4.5.2 'Accessing the I/O area in the master system' page 56*

Address	Bytes	Function
+0	4	Hardware interrupt counter: number of hardware interrupts
+4	4	Diagnostics interrupt counter: number of diagnostics interrupts
+8	8	Diagnostic module status
+16	1	*Newest Message* available
+17	1	Alignment
+18	1	Error Sync Manager Out
+19	1	Error Sync Manager In

Output area

No byte of the output area is used by the EtherCAT coupler.

3.4 Technical data

Order no.	**053-1EC01**
Type	IM 053EC
Module ID	-
Technical data power supply	
Power supply (rated value)	DC 24 V
Power supply (permitted range)	DC 20.4...28.8 V
Reverse polarity protection	✓
Current consumption (no-load operation)	95 mA
Current consumption (rated value)	0.95 A
Inrush current	3.9 A
I^2t	0.14 A^2s
Max. current drain at backplane bus	3 A
Max. current drain load supply	10 A
Power loss	3 W
Status information, alarms, diagnostics	
Status display	yes
Interrupts	yes, parameterizable
Process alarm	yes, parameterizable
Diagnostic interrupt	yes, parameterizable
Diagnostic functions	yes, parameterizable
Diagnostics information read-out	possible
Supply voltage display	green LED
Service Indicator	yellow LED
Group error display	red SF LED
Channel error display	none
Hardware configuration	
Racks, max.	1
Modules per rack, max.	64
Number of digital modules, max.	64
Number of analog modules, max.	64
Communication	
Fieldbus	EtherCAT
Type of interface	Ethernet 100 MBit
Connector	2 x RJ45
Topology	Line
Electrically isolated	✓

Order no.	**053-1EC01**
Number of participants, max.	65535
Node addresses	-
Transmission speed, min.	100 Mbit/s
Transmission speed, max.	100 Mbit/s
Address range inputs, max.	1024 Byte
Address range outputs, max.	1024 Byte
Number of TxPDOs, max.	-
Number of RxPDOs, max.	-
Supported profile	-
Supported transfer cycle	-
Cyclic data size per node	-
Max. Number of nodes	-
Supported communication method	-
Supported command "Cyclic"	-
Supported command "Event driven"	-
Supported command "Message"	-
Datasizes	
Input bytes	20
Output bytes	-
Parameter bytes	3
Diagnostic bytes	-
Housing	
Material	PPE / PPE GF10
Mounting	Profile rail 35 mm
Mechanical data	
Dimensions (WxHxD)	48.5 mm x 109 mm x 76.5 mm
Net weight	160 g
Weight including accessories	160 g
Gross weight	175 g
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL certification	yes
KC certification	yes

4 Deployment

4.1 Basics EtherCAT

4.1.1 General

Field buses were established for many years in the automation technology. Since higher speeds are required but the technical limits of this technology have already been reached, new solutions needed to be found.

At least in theory, the Ethernet, which is familiar to all of us from the office world, is fast with its 100Mbit/s speed, which is available everywhere today. However, these networks do not offer real-time capability due to the kind of cabling that they use and the rules governing access rights. This effect was corrected with EtherCAT®.

EtherCAT®

- For EtherCAT® is valid: EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- EtherCAT means Ethernet for Controller and Automation Technology. It was originally developed by Beckhoff Automation GmbH and is now supported and further developed by the EtherCAT Technology Group (ETG). ETG is the word biggest international user and producer connection for industrial Ethernet
- EtherCAT is an open Ethernet based field bus system, which is standardized at the IEC.
- As open field bus system EtherCAT matches the user profile for the part of industrial real-time systems.
- In opposition to the normal Ethernet communication at EtherCAT the data exchange of I/O data takes place during the frame passes the coupler with 100Mbit/s in full-duplex. Since in this way a frame to send and receive direction reaches the data of many stations, EtherCAT has a rate of user data of over 90%.
- The EtherCAT protocol, which is optimized for process data, is directly transported with the Ethernet frame. This again can consist of several sub-frames, which serve for a storage area of the process image.

Transfer medium

EtherCAT uses Ethernet as transfer medium Standard CAT5 cables are used. Here distances of about 100m between 2 stations are possible.

Only EtherCAT components may be used in an EtherCAT network. For topologies, which depart from the line structure, the corresponding EtherCAT components are necessary. Hubs may not be used.

Communication principle

At EtherCAT the master sends a telegram to the first station. The station takes its data from the current data stream, inserts its answer data and sends the frame to the succeeding station. Here the frame is handled with the same way.

When the frame has reached the last station this recognizes that no further is connected and sends the frame back to the master. Here the telegram is sent through every station via the other pair of leads (full-duplex). Due to the plug sequence and the use of the full-duplex technology EtherCAT represents a logical ring.

EtherCAT State Machine

Via the EtherCAT State Machine the state of the EtherCAT members is controlled.

Object dictionary (SDOs)

In the object directory the parameter, diagnostics, interrupt or other data are listed, which may be written or read via EtherCAT. The object directory may be accessed by the SDO information service. Additionally the object directory may be found in the device master file.

Process data (PDOs)

The EtherCAT data link layer is optimized for the fast transfer of process data. Here it is specified how the process data of the device are assigned to the EtherCAT process data and how the application of the device is synchronized to the EtherCAT cycle. The mapping of the process data happens by PDO mapping and by Sync-Manager-PDO-Assign objects. These describe, which objects of the object directory are transferred as object data via EtherCAT. The cycle time to transfer the process data via EtherCAT and how this is synchronized for the transfer is specified with the Sync-Manager-Communication objects.

Emergencies

Via Emergencies diagnostics, process events and errors at state change of the State Machine may be transferred.

Status messages, which show the current state of the device, should directly be transferred within the process data.

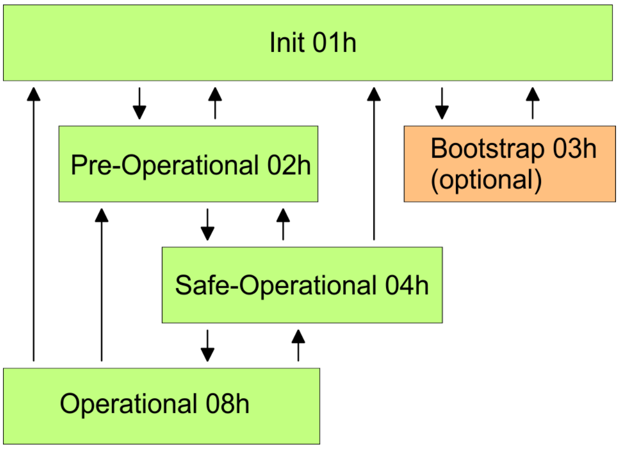
Distributed clocks (DC)

Due to the run time of an EtherCAT frame at the bus, the outputs of the EtherCAT slave stations are activated at different times and the inputs are read at different times. For an isochronous access to the process data EtherCAT provides the "distributed clock" functionality. In EtherCAT "**D**istributed **C**locks" means a logical combination of "clocks", which are located in the EtherCAT devices With this there is the possibility to locally provide a synchronized time in each bus device. With DC current output values are activated in equal time on the slave stations, and the input values read at precisely this time. This moment is called *Sync* signal. If an EtherCAT device supports the DC, it has its own clock. After PowerON this first locally works, based on an own pulse generator. By selecting an EtherCAT slave station, which has to provide the reference time, the distributed clocks can be synchronized. This *reference clock* so represents the system time.

4.1.2 EtherCAT State Machine

States

In each EtherCAT communication device a *state machine* is implemented. For each state there is defined which communication service is active via EtherCAT. The state machine of the slave station is controlled by the state machine of the EtherCAT master.



Init - 01h

After power-on the EtherCAT members are in state *Init*. There is neither mailbox nor process data communication possible. The EtherCAT master initializes the SyncManager channels 0 and 1 for the mailbox communication.

Pre-Operational (Pre-Op) - 02h

The EtherCAT master initializes the SyncManager channels for process data (starting with SyncManager channel 2), the FMMU channels and the PDO mapping respectively the SyncManager PDO assignment. Further in this state the settings for process data transfer and the module-specific parameters, which deviate from the default values are transferred. During the transition from *Init* to *Pre-Op* the EtherCAT slave checks whether the mailbox was correctly initialized. In the state *Pre-Op* mailbox communication and Ethernet over EtherCAT (EoE) are possible but the process data communication is blocked.

Safe-Operational (Safe-Op) - 04h

In *Safe-Op* the input data are cyclically updated but the outputs are de-activated. With the transition from *Pre-Op* to *Safe-Op* the EtherCAT slave checks if the SyncManager channels for process data communication are correct. Before it acknowledges the state change, the EtherCAT slave copies current input data to the corresponding DP RAM areas of the EtherCAT slave controller. In the state *Safe-Op* mailbox and process data communication is possible.

Operational (Op) - 08h

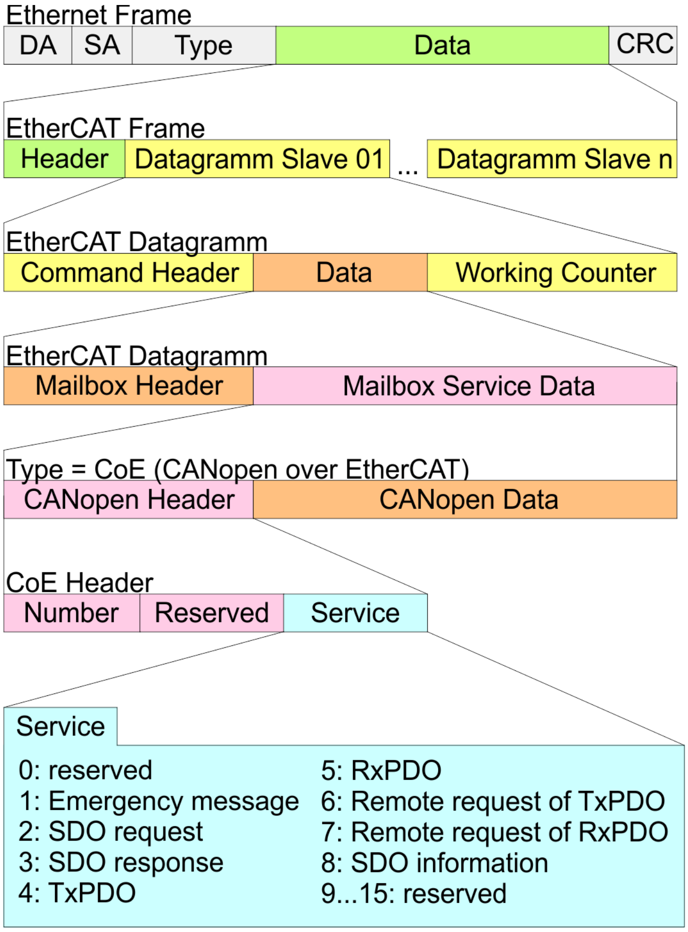
In the state *Op* the input data are cyclically updated and the EtherCAT master sends output data to the EtherCAT slave. The EtherCAT slave copies the output data of the master to its outputs and return input data to the EtherCAT master. In this state process data and mailbox communication is possible.

Bootstrap - option (Boot) - 03h

In state *Boot* the firmware of an EtherCAT slave may be updated via the EtherCAT master. This state may only be reached via Init. In the state *Boot* is mailbox communication via the protocol File-Access over EtherCAT (FoE) possible. Other mailbox and process data communications are de-activated.

4.1.3 CoE - CANopen over Ethernet

CoE means CANopen over EtherCAT. With CANopen you get a standard user interface, which makes a simplified system structure possible with most different devices. With CoE the device parameters may comfortably be accessed and data were may be read or written at the same time. Real-time data may be read by PDOs an the parametrization happens by SDOs. Further there are emergency objects available.



DA	Destination address
SA	Source address
CRC	Checksum

4.1.4 EoE - Ethernet over EtherCAT

EoE means Ethernet over EtherCAT.

- Automatically activated when switching from Init to Pre-Operational (Pre-Op).
- Is deactivated automatically when switching from Pre-Operational (Pre-Op) to Init.

4.1.5 ESI files

From YASKAWA there are ESI files for the EtherCAT coupler available. These files may be found at the *'Service / Support ➔ Downloads ➔ Config files'* of www.vipa.com. Please install the ESI files in your configuration tool. Details on the installation of the ESI files are available from the manual supplied with your configuration tool. For configuration in your configuration tool every System SLIO module may the found in the ESI files as XML data.

4.2 Setting the *HotConnect* address

To uniquely identify the EtherCAT slave, this must be assigned a so called *HotConnect* address. This address may only be assigned once at the bus. You have the following options for setting the *HotConnect* address:

- Explicit Device ID: Address switch on the module - default setting.
- Configured Station Alias: Value which is to be entered in the SII (EEPROM) of the slave station. SSI means **S**lave **I**nformation **I**nterface.

By default, with *HotConnect* the slave station is addressed via the *Explicit Device ID*. To change the addressing type, you must specify the corresponding ESC register in the *HotConnect* settings in the configuration tool:

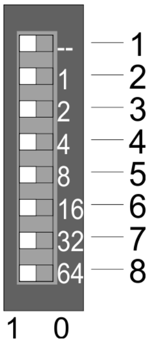
- Explicit Device ID: ESC register 0x0134 (default)
- Configured Station Alias: ESC register 0x0012

4.2.1 *HotConnect* via *Explicit Device ID*

Proceeding

The specification of the *HotConnect* address via *Explicit Device ID* happens as follows:

1. Switch off the power supply of the bus coupler.



2. Set the address at the address switch.

Pos.	Value	Example	
		State	Address
1	not used	—	1+2+32=35 Address: 35
2	1	1	
3	2	1	
4	4	0	
5	8	0	
6	16	0	
7	32	1	
8	64	0	

3. Switch on the power supply of the bus coupler.

   ⇨ The value set on the address switch is used as *Explicit Device ID*.

If addressing via address switch is not activated, you can activate it via your configuration tool. To do this, enter the ESC register address 0x0134 in your configuration tool.

Information on how to enter an ESC register address in your configuration tool can be found in the associated manual.

For example, in the EtherCAT Manager you must open the ‘Advanced options’ of the slave station in your configuration, activate "Check identification" and enter the ESC register address 0x0134 at "Select local address".

4.2.2 *HotConnect* via *Configured Station Alias*

Proceeding

The specification of the *HotConnect* address via *Configured Station Alias* happens as follows:

1. In your configuration tool, connect to the corresponding slave station. This must be in the INIT state.
2. Adjust the *Configured Station Alias* in the SII (EEPROM).
3. Perform a power cycle.

   Power 0 ⇆ 1 ⏻

   ⇨ The value set in the SII is used as *Configured Station Alias*.

> *Information on how to enter an ESC register and a Configured Station Alias address in your configuration tool can be found in the associated manual.*
>
> *For example, in the EtherCAT Manager you must open the 'Advanced options' of the slave station in your configuration, activate 'Check identification' and enter the ESC register address 0x0012 at 'Select local address'. In this case, the Configured Station Alias address must be specified in 'Diagnostics mode' via 'EEPROM' and in 'Group' via 'Identification value'.*

4.3 Operating Mode - Normal/Enhanced mode

> *Starting with firmware version V2.0.1 and ESI version V5.0, the IM 053-1EC01 supports FSoE safety modules. For this, switching to Enhanced mode is required. More information on the use of the VIPA FSoE safety modules can be found in the corresponding manual.*

Operating modes

You can specify the corresponding operating mode via the object 0x3000: 04. After setting the operating mode, the EtherCAT coupler reboots in the set operating mode. The IM 053-1EC01 has the following operating modes.

- Normal mode (default setting)
  - Setting via object 0x3000: 04 = 0x00
  - Backwards compatible with the previous version of the EtherCAT coupler
- Enhanced Mode
  - Setting via object 0x3000: 04 = 0x01
  - The structure of the module-related objects is changed.
    The index is calculated as follows:
    $\text{Index}_{\text{Enhanced}} = \text{Index} + (16 \cdot \textit{EtherCAT-Slot})$
  - FSoE safety modules are supported.

> *Please note that when replacing an EtherCAT Coupler in Enhanced mode, the new EtherCAT coupler must also be in Enhanced mode. Switching to the Enhanced mode must be done outside the plant, since appropriate default settings are required when commissioning the new EtherCAT coupler, otherwise the EtherCAT coupler does not change to the state PreOp.*

4.4 Synchronization modes

Overview

There are 3 modes of synchronization available to transfer the process data in EtherCAT:

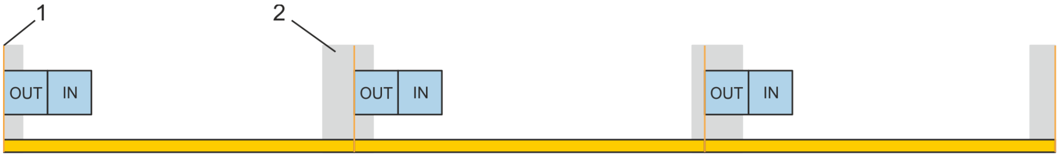
- Synchron with SyncManager event (default)
- Free Run
- Synchron with distributed clocks (DC)

The configuration tools usually include the synchronization modes *Free Run* and *Synchron with SyncManager event* as the operating mode *'Free Run / SM Synchron'*.

Synchron with SyncManager event (default)

By default, the IM 053EC works with the synchronization mode *Synchron with SyncManager event*.

With this mode of synchronization, a SyncManager event (interrupt) is triggered when an EtherCAT frame is received. This interrupt then triggers the processing of the process data.

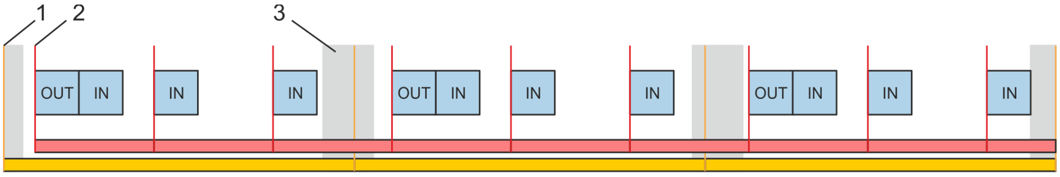


1 EtherCAT frame
2 Jitter, depends on the master implementation

Free Run

To use the *Free Run* synchronization mode, in your configuration tool, you must set the synchronization mode to *Free Run / SM-Synchron* and write 0 to the objects 0x1C32:01 and 0x1C33:01 in the *PreOp* state. The object can also be written by an Init command. More may be found in the manual of your configuration tool.

In the synchronization mode *Free Run*, the processing of the process data is controlled by an internal slave timer.

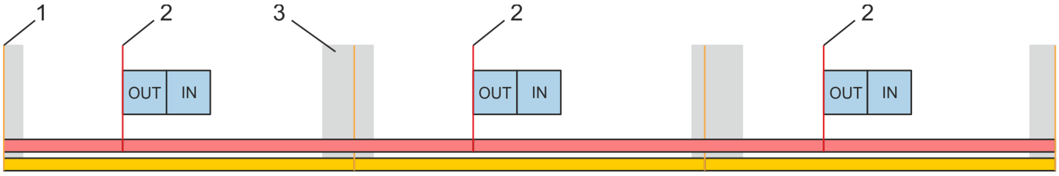


1 EtherCAT frame
2 Local timer event (main loop), depending on the slave implementation
3 Jitter, depends on the master implementation

Synchron with distributed clocks (DC)

To use the synchronization mode *Synchron with distributed clocks (DC)*, in your configuration tool you have to select the synchronization mode *Synchron with DC-Sync0*. More may be found in the manual of your configuration tool.

With this mode of synchronization, after receiving the EtherCAT frame, the process input or output data is read or written in sync with the *DC sync event*. The *DC sync event* is an interrupt which is triggered simultaneously on all DC capable slaves on the bus.



1 EtherCAT frame
2 DC sync event (DC Sync0)
3 Jitter, depends on the master implementation

In order to achieve isochronous mode, the synchronization mode Synchronous with distributed clocks (DC) does not perform any hardware retries on the backplane bus with process data telegrams. If errors are detected when reading or writing the process data on the backplane bus, the slave does not switch to PreOp. A module failure is detected by the NodeMonitoring.

With the synchronization mode Synchron with distributed clocks (DC), no System SLIO CP 040 may exist on the backplane bus.

4.5 Accessing the System SLIO

4.5.1 General

Calculation formula of the max. required cycle time in µs so that the coupler safely switches to the state *Operational (Op)*:

Max. required cycle time (in µs) = (1.83 * number of modules) + (0.33 * Σ of input and output bytes) + 60

Overview

In the following you will find the description of accessing the following System SLIO areas via EtherCAT:

- I/O area in the master system
- I/O area
- Parameter data
- Module information
- Diagnostics data

Information concerning the allocation of these areas may be found in the description of the corresponding System SLIO module.

4.5.2 Accessing the I/O area in the master system

The EtherCAT slave has a variable process data mapping. The process data are assigned to individual PDO mapping objects from the object dictionary, which can be activated or deactivated as required. The following PDO mapping objects are available:

Mapping objects	Process data	parameterized	Default
0x1AFF	Standard Diagnostic Data ■ Hardware Interrupt Counter ■ Diagnostic Interrupt Counter	no	activated
0x1B00	Extended Diagnostic Data ■ Diagnostic Module Status ■ *New Message* available	yes	activated
0x1B01	Sync Error Status ■ Sync Manager Out ■ Sync Manager In	yes	deactivated

The activation / deactivation takes place in the configuration tool. This automatically adjusts the SDO init commands accordingly.

4.5.3 Accessing the I/O area

The I/O area may be accessed via the following objects:

- PDO
- SDO

EtherCAT-Slot

Please consider the System SLIO power and clamp modules do not have any module ID. These may not be recognized by the EtherCAT coupler and so are not listed respectively considered during slot allocation.

Further within EtherCAT the slots are designated as ‘EtherCAT-Slot’. The counting always begins with 0.

Please note that in operation mode Enhanced Mode the index of the module-related objects is to be incremented accordingly.

The index is calculated as follows:

$Index_{Enhanced}$ = Index + (16 · EtherCAT-Slot)

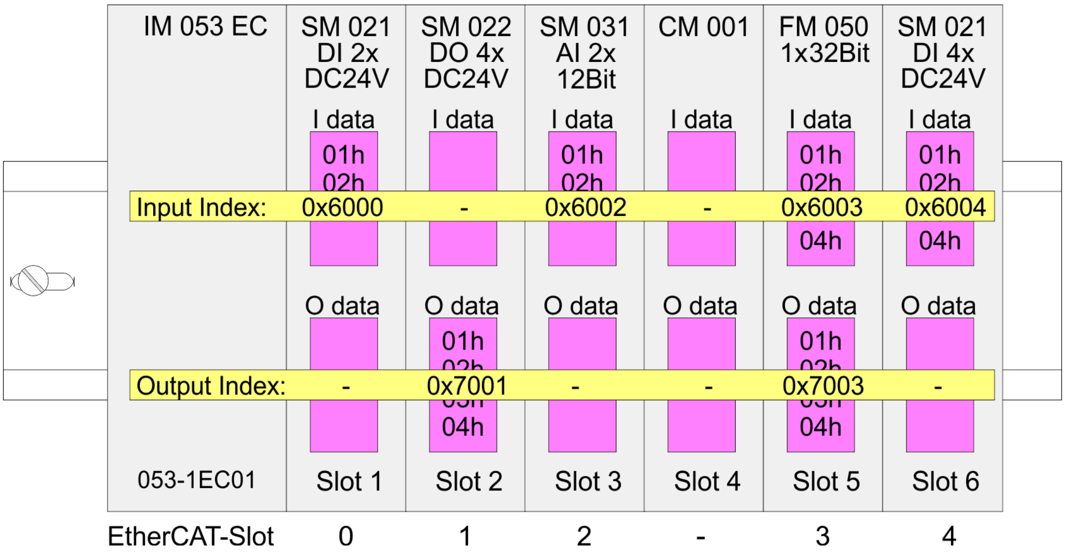
➪ Chap. 4.3 ‘Operating Mode - Normal/Enhanced mode’ page 54

4.5.3.1 Access via PDO

As soon as the EtherCAT coupler changes to the state *Safe-Operational*, the input data may be read. In the state *Operational* the output data may also be written. Information concerning the process data transfer by means of PDO may be found in the manual of your EtherCAT master. Additionally the in/output data are mapped on SDO objects.

4.5.3.2 Access via SDO

By means of SDO access the in/output data of the object directory may be read. The following figure shows how the in/output data are mapped on the SDO objects in operation mode *Normal Mode*:



Input data 0x6000 + EtherCAT-Slot

Index	Module	Subindex
0x6000	SM 021 DI 2x DC 24V	01h, 02h
0x6002	SM 031 AI 2x 12Bit	01h, 02h
0x6003	FM 050 1x32Bit	01h, 02h, 03h, 04h
0x6004	SM 021 DI 4x DC 24V	01h, 02h, 03h, 04h

Output data 0x7000 + EtherCAT-Slot

Index	Module	Subindex
0x7001	SM 022 DO 4x DC 24V	01h, 02h, 03h, 04h
0x7003	FM 050 1x32Bit	01h, 02h, 03h, 04h

Read input data

Index:Sub	Name	Type	Offset	Access	Default	Description
0x6000... 0x603F:00	Input data	U8	0	ro	60	Number of subindexes of the input data for the corresponding EtherCAT-Slot.
0x6000... 0x603F:01		U32	16	ro		Input data (see module description)
0x6000... 0x603F:02		U32		ro		
0x6000... 0x603F: ...	...	...		ro		
☞ *'Explanation of the elements' page 78*						

Please note that in operation mode Enhanced Mode the index of the module-related objects is to be incremented accordingly.

The index is calculated as follows:

$Index_{Enhanced}$ = Index + (16 · EtherCAT-Slot)

☞ *Chap. 4.3 'Operating Mode - Normal/Enhanced mode' page 54*

- With this object the input data of a System SLIO module may be read.
- Here the addressing of the EtherCAT-Slot via index (0x6000 + EtherCAT-Slot) takes place.
- The corresponding input data may be accessed via subindexes.
  – The allocation of the subindexes may be found in the description of the corresponding module.

Read output data

Index:Sub	Name	Type	Offset	Access	Default	Description
0x7000... 0x703F:00	Output data	U8	0	ro	60	Number of subindexes of the output data for the corresponding EtherCAT-Slot.
0x7000... 0x703F:01		U32	16	ro		Output data (see module description)
0x7000... 0x703F:02		U32		ro		
0x7000... 0x703F: ...	...	...		ro		
✎ *'Explanation of the elements' page 78*						

Please note that in operation mode Enhanced Mode the index of the module-related objects is to be incremented accordingly.

The index is calculated as follows:

$\text{Index}_{\text{Enhanced}}$ = Index + (16 · EtherCAT-Slot)

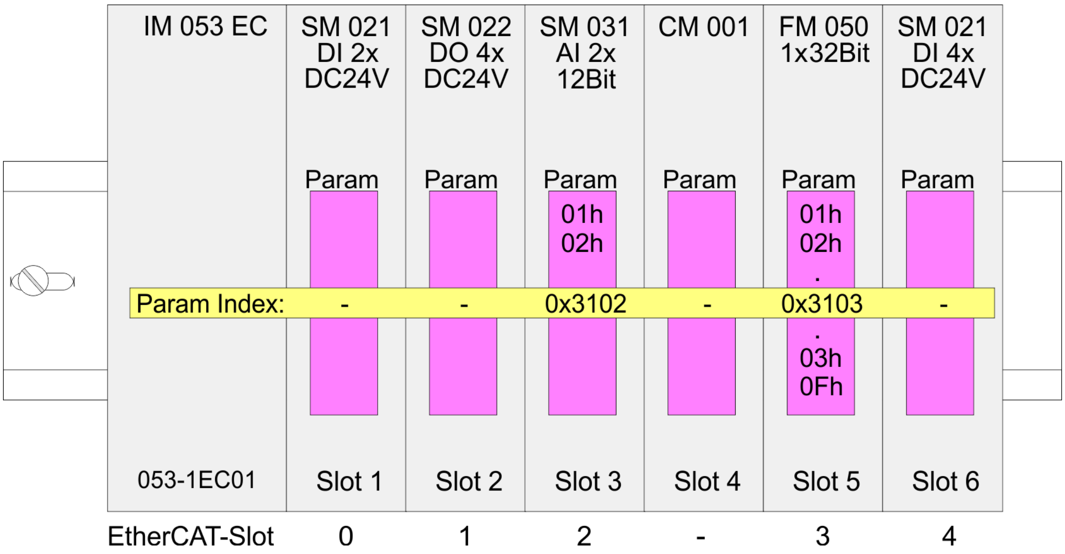
✎ *Chap. 4.3 'Operating Mode - Normal/Enhanced mode' page 54*

- With this object the output data of a System SLIO may be read.
- Here the addressing via index (0x7000 + EtherCAT-Slot) takes place.
- The corresponding output data may be accessed via subindexes.
  - The allocation of the subindexes may be found in the description of the corresponding module.

Information concerning the write access to outputs may be found in the manual of your EtherCAT master.

4.5.4 Accessing parameter data

The following figure shows how the parameter data are mapped on the SDO objects in operation mode *Normal Mode*:



Parameter data 0x3100 + EtherCAT-Slot

Index	Module	Subindex
0x3102	SM 031 AI 2x 12Bit	01h, 02h
0x3103	FM 050 1x32Bit	01h ... 0Fh

Accessing the parameters

For module parameterization SDO transfer takes place. Here the EtherCAT-Slot is addressed via *index*. Via *subindexes* the corresponding parameters may be accessed. The allocation of the subindexes may be found in the description of the corresponding System SLIO module.

Parameter SLIO Module

Index:Sub	Name	Type	Offset	Access	Default	Description
0x3100... 0x313F:00	Parameter	U8	0	ro	Number of parameters	Access to the parameters of the System SLIO modules. Only modules with parameters are mapped.
0x3100... 0x313F:01	Param1	U8	16	rw		Module parameter data (see module description)
0x3100... 0x313F:02	Param2	U8	32	rw		
0x3100... 0x313F: ...	...					

Modules with no interrupt capability or whose diagnostic interrupt is deactivated can not send an interrupt message on an error in parameterization. Possible parameterization errors may be detected by means of a diagnostic request after the parameterization.

> *Please note that in operation mode Enhanced Mode the index of the module-related objects is to be incremented accordingly.*
>
> *The index is calculated as follows:*
>
> *$\text{Index}_{\text{Enhanced}}$ = Index + (16 · EtherCAT-Slot)*
>
> *➪ Chap. 4.3 ‘Operating Mode - Normal/Enhanced mode’ page 54*

4.5.5 FMM - Free module mapping

4.5.5.1 Overview

FMM

- With FMM you can use EtherCAT slave stations with different hardware variants without adapting your user program. You only have to adapt the FMM configuration in the slave station when configuring the hardware variants. Here you have the following possibilities:
  - Modules from the target configuration can be divided in any order to the slots of the actual configuration.
  - Modules from the target configuration may be missing in the actual configuration.
  - Individual slots of the target configuration can be deactivated, on which modules are located in the actual configuration.
- By default, FMM is disabled. You can activate FMM by means of a CoE init command.
- If FMM is activated and configured correctly, the system reacts as follows:
  - During start-up, no target/actual difference of the hardware is diagnosed.
  - Output data of missing modules are ignored and not output.
  - Input data of missing modules are set to 0.
- Target configuration
  - The *target configuration* is the hardware configuration as the superset of all available hardware variants. It serves as a template for the configuration of hardware variants
- Actual configuration
  - The *actual configuration* corresponds to the physical structure of a hardware variant.

Steps target configuration

1. Configure your system with a hardware configuration as target configuration and create your user program. The target configuration represents a superset of all available hardware variants.
2. Open the EtherCAT configuration tool in your project.
3. Activate the parameter "Download slot configuration" for the slave station.
   ⇨ The module IDs of the target configuration are written to the CoE object 0xF030.
4. Adjust the CoE init command 0x3000:02 accordingly so that FMM is activated.
5. Close the EtherCAT configuration tool and save, translate and transfer your project to your CPU.

FMM configuration for hardware variant

To configure a hardware variant, a FMM configuration must be done.

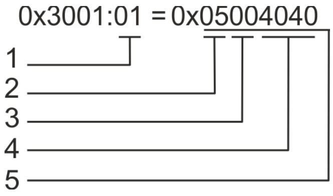
1. To do this, connect to the corresponding EtherCAT slave station.
2. Put the EtherCAT slave station in the *Pre-Op* state.
   ⇨ Now you can perform the FMM configuration.
3. Use the CoE object 0x3001 to adapt the FMM configuration to the hardware variant. *➪ Chap. 4.5.5.2 ‘FMM configuration’ page 63*

4. Activate FMM via the CoE object 0x3000:02, so that the FMM data is stored retentive

⇨ Your system is now ready for operation. An additional adaptation of your PLC program is not required.

4.5.5.2 FMM configuration

The FMM configuration is made with the CoE object 0x3001. The CoE object 0x3001 consists of 64 subindexes, on each of which a FMM value with 4bytes can be written. Here, the subindex represents the slot *Slot*$_{target}$ of the target configuration. For the FMM configuration, you must describe the corresponding subindex with an FMM value for each occupied slot of the target configuration. It has the following structure:



1 Subindex or Slot$_{target}$
2 Mapping or Slot$_{actual}$
3 00h (fix)
4 I/O$_{target}$
5 FMM

Slot$_{target}$ - The subindex of object 0x3001 represents the slot from the target configuration. The FMM configuration always refers to the slot of the target configuration.

Slot$_{actual}$ - Slot of the actual configuration on which the module of the target configuration is located.

Subindex - The subindex for the FMM configuration results from the *Slot*$_{target}$. This value is identical when configuring hardware variants.

Mapping - *Mapping* corresponds to the hex value of *Slot*$_{actual}$ i.e. the slot of the actual configuration on which the module of the target configuration is located.

If a module from the target configuration is missing, for *Mapping* the value FFh for "virtual module" is to be used.

If modules of the target configuration are to be ignored, the value 00h must be used for *Mapping*. In this way, gaps can be projected.

00h - This value is fixed.

I/O$_{target}$ - Number of input and output bytes of the target configuration. This value is identical when configuring hardware variants.

FMM - The value for *FMM* consists of *Mapping* & 00 & *I/O*$_{target}$.

Behaviour of a virtual module:

- The input area always has the value 0, regardless of its size.
- The writing to the output area has no effect.

– *For modules with variable IO size, you have to specify the size to which the module was configured in the hardware configuration.*

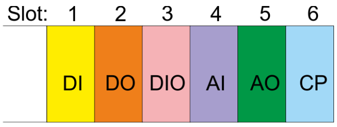
4.5.5.3 Examples

4.5.5.3.1 Target configuration

Sample application

The *target configuration* serves as a template for the configuration of hardware variants. It represents a superset of all available hardware variants.

The determination of the FMM values is shown with the following sample application.



Slot$_{target}$	**Module**	**Input**	**Output**	**I/O$_{target}$**
1	DI	1byte	-	0100h
2	DO	-	1byte	0001h
3	DIO	1byte	1byte	0101h
4	AI	8byte	-	0800h
5	AO	-	8byte	0008h
6	CP	60byte	60byte	3C3Ch

Slot$_{target}$ - The FMM configuration always refers to the slot of the target configuration.

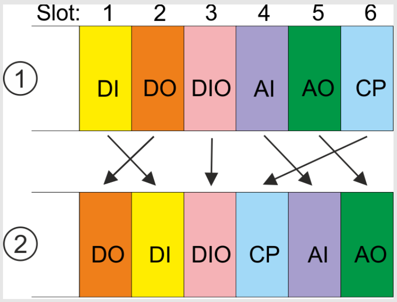
Subindex - The subindex for the FMM configuration results from the *Slot$_{target}$*. This value is identical when configuring hardware variants.

I/O$_{target}$ - Number of input and output bytes of the target configuration. This value is identical when configuring hardware variants.

4.5.5.3.2 Examples of hardware variants

Based on the target configuration, the following examples show how to determine the FMM values for the hardware variants.

Variant 1: Same type and number of modules but reversed slots

(1): Target configuration **(2): Actual configuration**	**$\text{Slot}_{\text{target}}$**	**Sub-index**	**$\text{Slot}_{\text{actual}}$**	**Mapping**	**$\text{I/O}_{\text{target}}$**	**FMM**
 Slot: 1 2 3 4 5 6 1 DI DO DIO AI AO CP 2 DO DI DIO CP AI AO	1	01	2	**02**h	0100h	0x**02**000100
	2	02	1	**01**h	0001h	0x**01**000001
	3	03	3	**03**h	0101h	0x**03**000101
	4	04	5	**05**h	0800h	0x**05**000800
	5	05	6	**06**h	0008h	0x**06**000008
	6	06	4	**04**h	3C3Ch	0x**04**003C3C

Determination of *Mapping* values:

- Subindex 01: The module of $\mathit{Slot}_{\mathit{target}}$ = 1 is in the actual configuration at $\mathit{Slot}_{\mathit{actual}}$ = 2 → Mapping = 02h
- Subindex 02: The module of $\mathit{Slot}_{\mathit{target}}$ = 2 is in the actual configuration at $\mathit{Slot}_{\mathit{actual}}$ = 1 → Mapping = 01h
- Subindex 03: The module of $\mathit{Slot}_{\mathit{target}}$ = 3 is in the actual configuration at $\mathit{Slot}_{\mathit{actual}}$ = 3 → Mapping = 03h
- Subindex 04: The module of $\mathit{Slot}_{\mathit{target}}$ = 4 is in the actual configuration at $\mathit{Slot}_{\mathit{actual}}$ = 5 → Mapping = 05h
- Subindex 05: The module of $\mathit{Slot}_{\mathit{target}}$ = 5 is in the actual configuration at $\mathit{Slot}_{\mathit{actual}}$ = 6 → Mapping = 06h
- Subindex 06: The module of $\mathit{Slot}_{\mathit{target}}$ = 6 is in the actual configuration at $\mathit{Slot}_{\mathit{actual}}$ = 4 → Mapping = 04h

$\text{Slot}_{\text{target}}$ - The FMM configuration always refers to the slot of the target configuration.

Subindex - The subindex for the FMM configuration results from the $\mathit{Slot}_{\mathit{target}}$. This value is identical when configuring hardware variants.

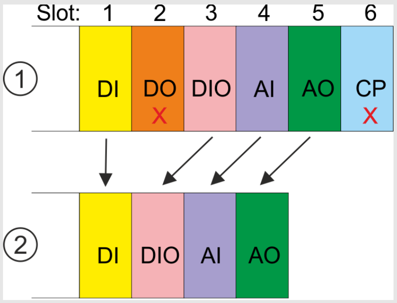
$\text{Slot}_{\text{actual}}$ - Slot of the actual configuration on which the module of the target configuration is located.

Mapping - For variant 1 *Mapping* corresponds to the hex value of $\mathit{Slot}_{\mathit{actual}}$ i.e. the slot of the actual configuration on which the module of the target configuration is located. *Mapping* must be adapted when configuring the hardware variant.

$\text{I/O}_{\text{target}}$ - Number of input and output bytes of the target configuration. This value is identical when configuring hardware variants.

FMM - The value for *FMM* consists of *Mapping* & 00 & $\mathit{I/O}_{\mathit{target}}$.

Variant 2: Reversed slots and modules from the actual configuration are ignored

(1): Target configuration (2): Actual configuration	Slot$_{target}$	Sub-index	Slot$_{actual}$	Mapping	I/O$_{target}$	FMM
 Slot: 1 2 3 4 5 6 1 DI DO X DIO AI AO CP X 2 DI DIO AI AO	1	01	1	**01**h	0100h	0x**01**000100
	2	02	-	**FF**h	0001h	0x**FF**000001
	3	03	2	**02**h	0101h	0x**02**000101
	4	04	3	**03**h	0800h	0x**03**000800
	5	05	4	**04**h	0008h	0x**04**000008
	6	06	-	**FF**h	3C3Ch	0x**FF**003C3C

Determination of *Mapping* values:

- Subindex 01: The module of *Slot$_{target}$* = 1 is in the actual configuration at *Slot$_{actual}$* = 1 → Mapping = 01h
- Subindex 02: The module of *Slot$_{target}$* = 2 is not available in the actual configuration → Mapping = FFh
- Subindex 03: The module of *Slot$_{target}$* = 3 is in the actual configuration at *Slot$_{actual}$* = 2 → Mapping = 02h
- Subindex 04: The module of *Slot$_{target}$* = 4 is in the actual configuration at *Slot$_{actual}$* = 3 → Mapping = 03h
- Subindex 05: The module of *Slot$_{target}$* = 5 is in the actual configuration at *Slot$_{actual}$* = 4 → Mapping = 04h
- Subindex 06: The module of *Slot$_{target}$* = 6 is not available in the actual configuration → Mapping = FFh

Slot$_{target}$ - The FMM configuration always refers to the slot of the target configuration.

Subindex - The subindex for the FMM configuration results from the *Slot$_{target}$*. This value is identical when configuring hardware variants.

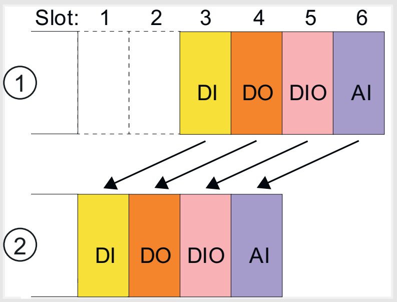
Slot$_{actual}$ - Slot of the actual configuration on which the module of the target configuration is located.

Mapping - For variant 2 *Mapping* corresponds to the hex value of *Slot$_{actual}$* i.e. the slot of the actual configuration on which the module of the target configuration is located. If a module from the target configuration is missing, for *Mapping* the value FFh for "virtual module" is to be used.

I/O$_{target}$ - Number of input and output bytes of the target configuration. This value is identical when configuring hardware variants.

FMM - The value for *FMM* consists of *Mapping* & 00 & *I/O$_{target}$*.

Variant 3: Gaps in the target configuration

(1): Target configuration **(2): Actual configuration**	**Slot$_{target}$**	**Sub-index**	**Slot$_{actual}$**	**Mapping**	**I/O$_{target}$**	**FMM**
	empty	01	1	**00**h	0	0
	empty	02	2	**00**h	0	0
	3	03	3	**01**h	0100h	0x**01**000100
	4	04	4	**02**h	0001h	0x**02**000001
	5	05	5	**03**h	0101h	0x**03**000101
	6	06	6	**04**h	0800h	0x**04**000800

Determination of *Mapping* values:

- Subindex 01: The module of *Slot$_{target}$* = 1 is ignored in the actual configuration → Mapping = 00h
- Subindex 02: The module of *Slot$_{target}$* = 2 is ignored in the actual configuration → Mapping = 00h
- Subindex 03: The module of *Slot$_{target}$* = 3 is in the actual configuration at *Slot$_{actual}$* = 1 → Mapping = 01h
- Subindex 04: The module of *Slot$_{target}$* = 4 is in the actual configuration at *Slot$_{actual}$* = 2 → Mapping = 02h
- Subindex 05: The module of *Slot$_{target}$* = 5 is in the actual configuration at *Slot$_{actual}$* = 3 → Mapping = 03h
- Subindex 06: The module of *Slot$_{target}$* = 6 is in the actual configuration at *Slot$_{actual}$* = 4 → Mapping = 04h

Slot$_{target}$ - The FMM configuration always refers to the slot of the target configuration.

Subindex - The subindex for the FMM configuration results from the *Slot$_{target}$*. This value is identical when configuring hardware variants.

Slot$_{actual}$ - Slot of the actual configuration on which the module of the target configuration is located.

Mapping - For variant 3 *Mapping* corresponds to the hex value of *Slot$_{actual}$* i.e. the slot of the actual configuration on which the module of the target configuration is located. If modules of the target configuration are to be ignored, the value 00h must be used for *Mapping*.

I/O$_{target}$ - Number of input and output bytes of the target configuration. This value is identical when configuring hardware variants.

FMM - The value for *FMM* consists of *Mapping* & 00 & *I/O$_{target}$*.

The presence of gaps in the System SLIO is not allowed! But you can place modules and define them via the configuration as empty slot for the target hardware configuration.

4.5.6 Easy Maintenance

Overview

Easy Maintenance means the support for adding and removing System SLIO modules during operation without having to restart the system. There are the following behaviors:

- *Easy Maintenance* at *active* bus connection
  - The bus cable is plugged and the IM 053-1EC01 is active on the bus.
  - As soon as a module is removed, the IM 053-1EC01 detects this, switches automatically to *PreOp* state and stops the process data communication.

    By re-plugging the module or a compatible module after a State-Change-Request, the change to the state *Op*takes place.
  - As soon as a module is added to the existing modules, the IM 053-1EC01 detects this and automatically refreshes its configuration.
- *Easy Maintenance* at *passive* bus connection
  - The bus cable is removed.
  - As soon as a module is removed, the IM 053-1EC01 detects this and automatically refreshes its configuration.
  - As soon as a module is added to the existing modules, the IM 053-1EC01 detects this and automatically refreshes its configuration.

Please always replace only one module in your configuration.

4.5.6.1 Examples

4.5.6.1.1 Module is removed with active bus connection

If a module is removed during an active bus connection, the IM 053-1EC01 behaves as follows:

1. The IM 053-1EC01 changes to the *PreOp* state.
2. The process data communication is ended.
3. Entries are made in the history object of the IM 053-1EC01:
   - Module was removed
   - State change
4. The System SLIO backplane bus is refreshed.

   ⇨ In the state *PreOp*:
   - The SF LED blinks.
   - The RUN LED blinks.

4.5.6.1.2 Compatible module is added again when the bus connection is active

If a compatible module is inserted into the slot of a previously removed module during an active bus connection, the IM 053-1EC01 behaves as follows:

1. Entries are made in the history object of the IM 053-1EC01:
   - Module was plugged.
2. The System SLIO backplane bus is refreshed.
3. The coupler can only be brought to the state *Op* by means of a State-Change-Request.

   ⇨ In the state *Op*:
   - The SF LED turns off.
   - The RUN LED lights up.
   - The process data communication is restarted.

4.5.6.1.3 Incompatible module is added again when the bus connection is active

With activated *Download slot configuration*

If an incompatible module is inserted into the slot of a previously removed module during an active bus connection, the IM 053-1EC01 behaves as follows:

- The System SLIO backplane bus is refreshed.

  ⇨ Since a difference between different modules of the same I/O length is detected, the SF and RUN LEDs continue to flash.

With deactivated *Download slot configuration*

If an incompatible module is inserted into the slot of a previously removed module during an active bus connection, the IM 053-1EC01 behaves as follows:

- The System SLIO backplane bus is refreshed.

  ⇨ Since no difference is detected between different modules of the same I/O length, the SF LED goes off and the RUN LED continues to blink.

4.5.6.1.4 Not configured modules are added to the existing modules when the bus connection is active

If one or more modules, which are not configured, are added to the existing modules during an active bus connection, the IM 053-1EC01 behaves as follows:

1. The IM 053-1EC01 changes to the *PreOp* state.
2. The System SLIO backplane bus is refreshed.
   ⇨ In the state *PreOp*:
   - The SF LED blinks.
   - The RUN LED blinks.

4.5.6.1.5 Modules are added or removed when the bus plug is removed

If modules are added or removed, when the bus plug is removed, the IM 053-1EC01 behaves as follows:

The modification is detected automatically and the configuration is refreshed.
⇨
- The SF LED remains off.
- The RUN LED remains unchanged.

4.5.7 Access to module information

Module information

Index:Sub	Name	Type	Offset	Access	Default	Description
0x4100:00	Module Information	U8	0	ro	8	Module Identity Information IM 053EC
0x4100:01	Module Name	VSTRG	16	ro	0	Module Name
0x4100:02	Module ID	U32	256	ro	0	Module ID
0x4100:03	Serial Number	VSTRG	288	ro	0	Serial number of the module
0x4100:04	Product Version	VSTRG	448	ro	0	Product version of the module
0x4100:05	Hardware Version	VSTRG	520	ro	0	Hardware version of the module
0x4100:06	Software Version	VSTRG	536	ro	0	Software version of the module
0x4100:07	FPGA Version	U16	632	ro	0	FPGA version
0x4100:08	Mx File	VSTRG	648	ro	0	Mx file of the coupler
⇘ *'Explanation of the elements' page 78*						

When accessing the module information of IM 053EC the corresponding module is addressed via the index (0x4100 + EtherCAT-Slot). Subindexes give you access to the corresponding module information.

> *Please note that in operation mode Enhanced Mode the index of the module-related objects is to be incremented accordingly.*
>
> *The index is calculated as follows:*
>
> $Index_{Enhanced} = Index + (16 \cdot EtherCAT\text{-}Slot)$
>
> ⇘ *Chap. 4.3 'Operating Mode - Normal/Enhanced mode' page 54*

4.5.8 Accessing diagnostics data

Hardware and diagnostic interrupt data of System SLIO modules with interrupt capability were automatically sent by an emergency telegram if the interrupt is activated by parameterization. There is also the possibility to request diagnostics data via SDO.

Alarm Status

The Alarm Status contains in each case a counter for hardware and diagnostic interrupts for interrupt signaling. These counter are input data of the EtherCAT coupler and were transferred together with the process data.

Diagnostic

Index:Sub	Name	Type	Offset	Access	Default	Description
0xF100:00	Diagnostic	U8	0	ro	3	
0xF100:01	Hardware interrupt counter	U32	16	ro	0x00000000	Counter for hardware interrupt
0xF100:02	Diagnostic interrupt counter	U32	48	ro	0x00000000	Counter for diagnostic interrupt
0xF100:03	Diagnostic Module Status	U64	80	ro	0x00000000	64 bit array, each bit represents a pending (but acknowledged) diagnostic alarm
↳ *'Explanation of the elements' page 78*						

- With de-activated *Auto-Acknowledge* of the EtherCAT coupler the corresponding counter is set to 1 until you acknowledge this accordingly. For this you have to write any value you like to subindex 0x06 under the accordingly assigned index.
- With activated *Auto-Acknowledge* here you will find the number of hardware and diagnostics interrupts, which happened since the last interrupt reset. To reset the corresponding counter you have to write any value you like to the subindex 0x06 under the accordingly assigned index.

There are the following index assignments:

- Write to 0x06 of index 0x5000:
  - Reset of counter hardware interrupt
- Write to 0x06 of index 0x5002:
  - Reset of counter diagnostic interrupt

Hardware interrupt data

Index:Sub	Name	Type	Offset	Access	Default	Description
0x5000:00	Hardware interrupt data	U8	0	ro	6	Current hardware interrupt data
0x5000:01	Slot number	U8	16	ro	0x00	EtherCAT-Slot of the module which reports an interrupt
0x5000:02	Hardware interrupt data 00	U8	24	ro	0x00	Hardware interrupt data (see module description)
0x5000:03	Hardware interrupt data 01	U8	32	ro	0x00	
0x5000:04	Hardware interrupt data 02	U8	40	ro	0x00	
0x5000:05	Hardware interrupt data 03	U8	48	ro	0x00	
0x5000:06	Acknowledge	U8	56	rw	0x00	Any value resets the hardware interrupt counter and acknowledges if necessary the interrupt.
➭ *'Explanation of the elements' page 78*						

- If the object 0xF100 indicates a hardware interrupt here the current hardware interrupt data may be accessed. The allocation of the hardware interrupt data may be found in the description of the corresponding module.
- With de-activated *Auto-Acknowledge* of the EtherCAT coupler the *hardware interrupt counter* of object 0xF100 is reset and the hardware interrupt is acknowledged. For this you have to write any value you like to subindex 0x06 of index 0x5000.

Diagnostic Interrupt Data (byte 1 ... 4)

Index:Sub	Name	Type	Offset	Access	Default	Description
0x5002:00	Diagnostic Interrupt Data	U8	0	ro	6	Current diagnostic interrupt data
0x5002:01	Slot number	U8	16	ro	0x00	EtherCAT-Slot of the module which reports an interrupt.
0x5002:02	Diagnostic data 00	U8	24	ro	0x00	Byte 1 ... 4 of the diagnostic data (see module description)
0x5002:03	Diagnostic data 01	U8	32	ro	0x00	
0x5002:04	Diagnostic data 02	U8	40	ro	0x00	
0x5002:05	Diagnostic data 03	U8	48	ro	0x00	
0x5002:06	Acknowledge	U8	56	rw	0x00	Any value resets the diagnostic interrupt counter and acknowledges if necessary the interrupt.
➭ *'Explanation of the elements' page 78*						

- If the object 0xF100 indicates a diagnostic interrupt here the first 4 bytes of the current diagnostics interrupt data may be accessed. The allocation of the diagnostics data may be found in the description of the corresponding module.
- With de-activated *Auto-Acknowledge* of the EtherCAT coupler the *diagnostic interrupt counter* of object 0xF100 is reset and the diagnostic interrupt is acknowledged. For this you have to write any value you like to subindex 0x06 of index 0x5002.

Diagnostic Data (byte 1 ... n)

Index:Sub	Name	Type	Offset	Access	Default	Description
0x5005:00	Diagnostic Data	U8	0	ro	19	
0x5005:01	Slot number	U8	16	rw	0	By reading this index here the EtherCAT-Slot of the module may be found, to which the following diagnostic data belong to. By writing of an EtherCAT-Slot to this index the diagnostic data of the corresponding module may be read.
0x5005:02	Module error	U8	24	ro	0	Diagnostics data of the module (see module description)
0x5005:03	Module type	U8	32	ro	0	
0x5005:04	Fix	U8	40	ro	0	
0x5005:05	Communication/ Process interrupt	U8	48	ro	0	
0x5005:06	Channel type	U8	56	ro	0	
0x5005:07	Number of diagnostic bits	U8	64	ro	0	
0x5005:08	Number of channels	U8	72	ro	0	
0x5005:09	Channel error	U8	80	ro	0	
0x5005:0A	Channel-specific error channel 0	U8	88	ro	0	
0x5005:0B	Channel-specific error channel 1	U8	96	ro	0	
0x5005:0C	Channel-specific error channel 2	U8	104	ro	0	
0x5005:0D	Channel-specific error channel 3	U8	112	ro	0	
0x5005:0E	Channel-specific error channel 4	U8	120	ro	0	
0x5005:0F	Channel-specific error channel 5	U8	128	ro	0	
0x5005:10	Channel-specific error channel 6	U8	136	ro	0	
0x5005:11	Channel-specific error channel 7	U8	144	ro	0	
0x5005:13	-	U8	152	ro	0	
0x5005:14	µs-Ticker	U32	160	ro	0	

‘Explanation of the elements’ page 78

The whole diagnostic data of a module may be accessed with this object. Here you may either access current diagnostic data or the diagnostic data of a module on any EtherCAT-Slot.

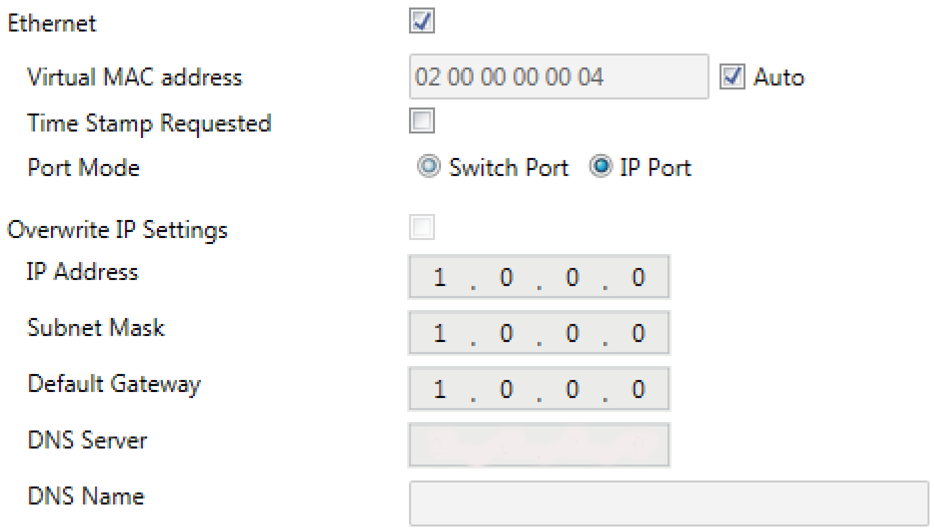
4.6 Accessing the web server via Ethernet

Access via Ethernet (EoE)

On delivery the EtherCAT coupler has no IP address. To be able to access the EtherCAT coupler via EoE, you have to assign data to this IP address. The following options in the SPEED7 EtherCAT Manager are available:

Configuration - slave station - Ethernet (EoE)

- Port Mode:
  - Switch Port (is not supported)
  - IP Port
- Overwrite IP Settings:
  - IP Address
  - Subnet Mask
  - Default Gateway
  - DNS Server (is not supported)
  - DNS Name (is not supported)



Ethernet ☑

Virtual MAC address: 02 00 00 00 00 04 ☑ Auto

Time Stamp Requested □

Port Mode ○ Switch Port ● IP Port

Overwrite IP Settings □

IP Address: 1 . 0 . 0 . 0

Subnet Mask: 1 . 0 . 0 . 0

Default Gateway: 1 . 0 . 0 . 0

DNS Server

DNS Name

4.6.1 Web server

Activate / deactivate web server

The activation is carried out via the Command object 0x2000 (Default: deactivated). ☞ *'Command' page 89*

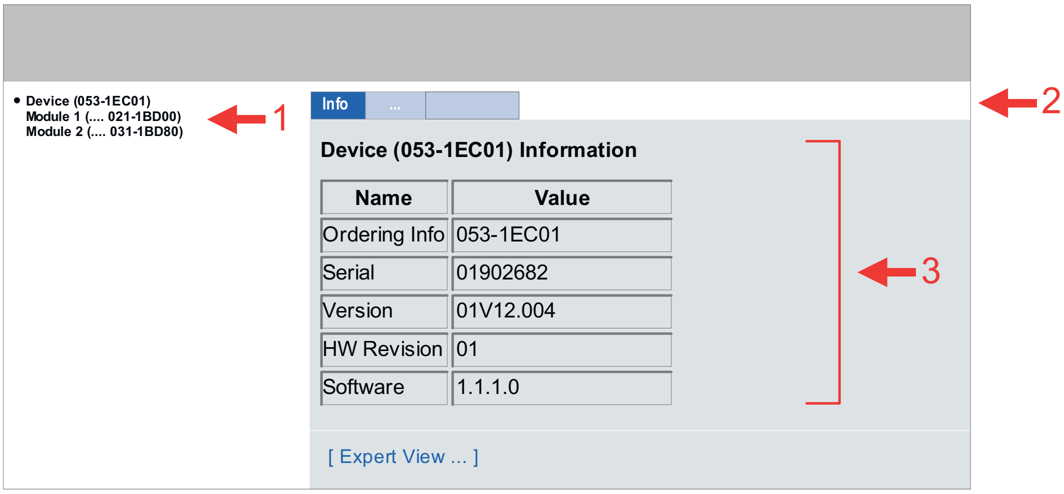
Procedure

- Use *Parameter* (object 0x2000:02) to determine whether the web server is enable ("1") or disable ("0"),
- use *Execute Command* (object 0x2000:01) to issue the command 0x00008001 (Set web server state),
- *Status* (object 0x2000: 03) indicates whether the command was executed successfully or incorrectly.
  - 1: Command executed
  - 2: Command not executed
  - 3: Command unknown
  - 4: Command not executable in this state

The setting is saved remanently.

Structure of the Web page

The Web page is built dynamically and depends on the number of modules, which are connected to the EtherCAT coupler.



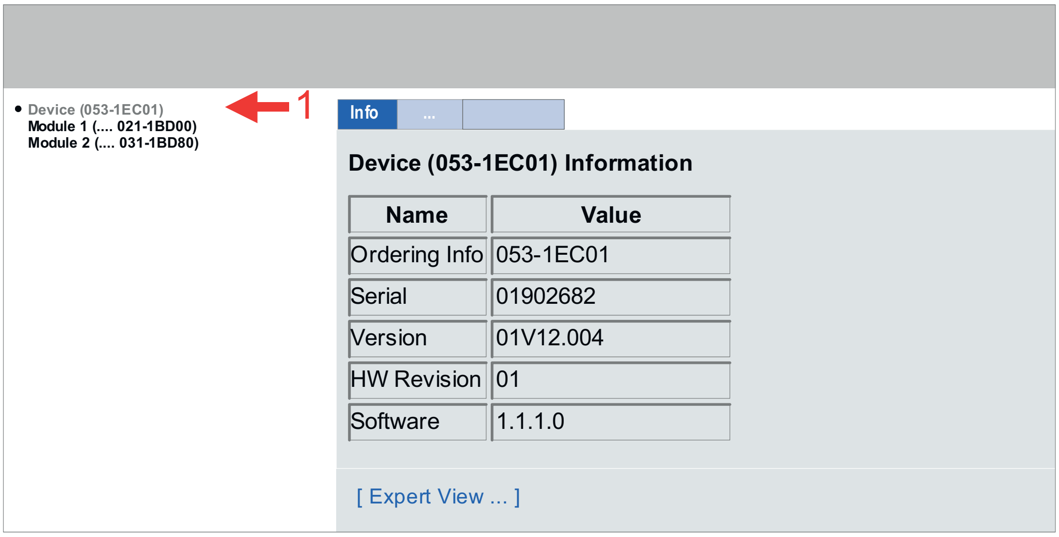
- Device (053-1EC01)
  Module 1 (.... 021-1BD00)
  Module 2 (.... 031-1BD80)

Device (053-1EC01) Information

Name	Value
Ordering Info	053-1EC01
Serial	01902682
Version	01V12.004
HW Revision	01
Software	1.1.1.0

[1] Module list: EtherCAT coupler and System SLIO module in the plugged sequence
[2] Functions for the module, which is selected in the *module list*
[3] Information respectively input field for the according function

Web page with selected EtherCAT coupler



- Device (053-1EC01)
  Module 1 (.... 021-1BD00)
  Module 2 (.... 031-1BD80)

Device (053-1EC01) Information

Name	Value
Ordering Info	053-1EC01
Serial	01902682
Version	01V12.004
HW Revision	01
Software	1.1.1.0

- Info
  - Here order number, serial number and the version of firmware and hardware of the IM 053-1EC01 are listed.
- Data
  - The process data from the IM 053-1EC01 and all modules are displayed here.
- Parameter
  - Here you can see the current parameters of the IM 053-1EC01. ↳ *'Parameter EtherCAT coupler' page 90*
- Diagnosis
  - Configuration State
    Deviations between the setpoint and actual configuration are shown here (OK: no deviation).
  - Diagnosis Entries
    Here all pending diagnostic messages are shown.
  - Diagnosis Buffer
    Here the content of the diagnostics buffer is listed.

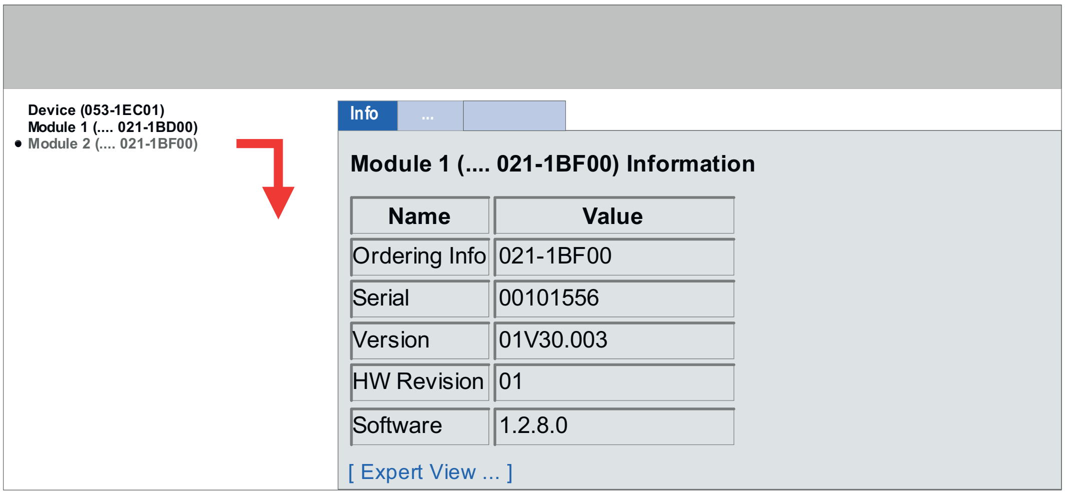
- Security
  – All functions for the writing access to the IM 053-1EC01 can be secured by a password.
- IP
  – Here you can see the current IP address data of the IM 053-1EC01.
- Firmware
  – With this function you can bring in a firmware update.
    The latest firmware versions are to be found in the service area at www.vipa.com. Load the Px000314.pkg file. ↻ *Chap. 4.11 'Firmware update' page 106*

Expert View

[Expert View] takes you to the advanced "Expert View".

Please consider the System SLIO power and clamp modules do not have any module ID. These may not be recognized by the EtherCAT coupler and so are not listed respectively considered during slot allocation.

Web page with selected module



Module 1 (.... 021-1BF00) Information

Name	Value
Ordering Info	021-1BF00
Serial	00101556
Version	01V30.003
HW Revision	01
Software	1.2.8.0

[Expert View ...]

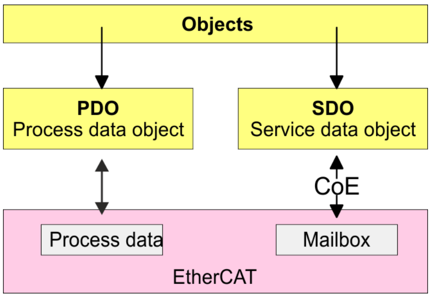
- Info
  – Here product name, order number, serial number, firmware version and hardware state number of the according module are listed.
- Data
  – At Data the states of the inputs respectively outputs are listed.
- Parameter
  – If available the parameter data of the corresponding module may be shown.
- Diagnosis
  – Configuration State
    Here the status of the configuration is displayed.
  – Diagnosis Entries
    All active diagnostic messages are displayed here.
  – Diagnosis Buffer
    Here the content of the diagnostics buffer is listed.

Expert View

[Expert View] takes you to the advanced "Expert View".

4.7 Data transfer via PDO and SDO

Overview



PDO

PDO means **P**rocess **D**ata **O**bject. Process data may be transferred during run-time by means of PDOs. Here the Inputs respectively outputs are directly addressed within the frame. More concerning the process data transfer during run-time may be found in the manual of your EtherCAT master.

SDO

SDO means **S**ervice **D**ata **O**bject. Within a SDO message the parameter data of your system may be accessed. Here the parameters are addressed by according indexes and subindexes. An overview of the objects may be found in the object directory.

4.8 Variable PDO mapping via SDO

Overview

Some modules a variable process image. If your master system supports variable PDO mapping, on modules with variable process image the size of the process image may be preset by a appropriate adjustment of the PDO contents. More information about PDO mapping may be found in the manual of your master system.

Functionality

- The ESI file the modules are defined, which have a variable process image When modules with a variable process image are used, so startup commands are automatically inserted to the startup list of the master.
  - These startup commands are SDO write accesses to the areas 0x16yy and 0x1Ayy (PDO mapping), where the structure of the process data of a module is described.
  - The startup commands are executed with each change from Pre-Op to Safe-Op condition of the appropriate EtherCAT coupler.
- Within your EtherCAT master system / EtherCAT configurator the PDO contents may be removed or added. Here the startup command are automatically adjusted.

With the adjustment of the PDO contents please always consider the data structure of the I/O data of the corresponding module. Information about the data structure of the I/O data of a module may be found in the additional System SLIO manual.

- So that the new configuration can be taken, the master system is to be initialized again, respectively when a EtherCAT configurator is used the new configuration can be loaded into the EtherCAT master.
  - With the change from Pre-Op to Safe-Op the new startup commands were transferred to the EtherCAT coupler and thus the new PDO mapping is adjusted.
- The EtherCAT coupler configures the corresponding module and automatically adjusts the length of the module parameters in the object directory 0x31yy an.

The adjustment of the process image of modules with a variable process image size should not be made by the object 0x31yy, since otherwise only the length in the module is adjusted but EtherCAT coupler and master further use the standard length!

4.9 Object Dictionary

Object overview

Index	Object Dictionary Area
0x0000 ... 0x0FFF	Data Type Area
0x1000 ... 0x1FFF	Communication Area
0x2000 ... 0x5FFF	Manufacturer Specific Area
0x6000 ... 0x6FFF	Input Area
0x7000 ... 0x7FFF	Output Area
0x8000 ... 0x8FFF	Configuration Area
0x9000 ... 0x9FFF	Information Area
0xA000 ... 0xAFFF	Diagnosis Area
0xB000 ... 0xBFFF	Service Transfer Area
0xC000 ... 0xEFFF	Reserved Area
0xF000 ... 0xFFFF	Device Area

EtherCAT-Slot

Please consider the System SLIO power and clamp modules do not have any module ID. These may not be recognized by the EtherCAT coupler and so are not listed respectively considered during slot allocation.

Further within EtherCAT the slots are designated as ‘EtherCAT-Slot’. The counting always begins with 0.

In the following example at the physical slot 2 a clamp module (CM) is connected. Since the clamp module does not have any module ID, this is not recognized by the EtherCAT coupler. So in EtherCAT the next module is mapped to EtherCAT-Slot 1 and so on.

Explanation of the elements

Index:Sub	- Index and subindex
Ux	- Data type UNSIGNEDx
VSTRG	- Data type VISIBLE STRING
Access	- Read, write access (ro: Read access only, rw: Read and write access)
Default	- Default value
Offset	- Please note the offset

Device Type

Index:Sub	Name	Type	Access	Default	Description
0x1000:00	Device type	U32	ro	0x00001389	0x00001389 means MDP
➪ *‘Explanation of the elements’ page 78*					

Error Register

Index:Sub	Name	Type	Access	Default	Description
0x1001:00	Error Register	U8	ro	0	Error Register ■ Bit 0: Generic error – A not more precisely specified error occurred (flag is set at every error message). ■ Bit 1: Current error ■ Bit 2: Voltage error ■ Bit 3: Temperature error ■ Bit 4: Communication error ■ Bit 5: Device profile specific error ■ Bit 6: reserved ■ Bit 7: Manufacturer specific error
↳ *'Explanation of the elements' page 78*					

Device name

Index:Sub	Name	Type	Access	Default	Description
0x1008:00	Device name	VSTRG	ro	053-1EC01	Name of the coupler
↳ *'Explanation of the elements' page 78*					

Hardware version

Index:Sub	Name	Type	Access	Default	Description
0x1009:00	Hardware version	VSTRG	ro		Hardware version of the coupler
↳ *'Explanation of the elements' page 78*					

Software version

Index:Sub	Name	Type	Access	Default	Description
0x100A:00	Software version	VSTRG	ro		Software version of the coupler
↳ *'Explanation of the elements' page 78*					

Identity object

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1018:00	Identity object	U8	0	ro	4	Identity object
0x1018:01	Vendor ID	U32	16	ro	0x0000022B	Vendor ID assigned uniquely by ETG
0x1018:02	Product code	U32	48	ro	0x0531EC01	Product Code assigned uniquely by vendor
0x1018:03	Revision number	U32	80	ro	2	Revision number assigned uniquely by vendor
0x1018:04	Serial number	U32	112	ro	0	Serial number assigned uniquely for this device by vendor
↳ *'Explanation of the elements' page 78*						

Error Setting

Index:Sub	Name	Type	Offset	Access	Default	Description
0x10F1:00		U8	0	ro	0x0002	Error Setting
0x10F1:01		U32	16	rw	0x0001	■ Local Error Reaction – 0 = PDO State slave changes the value of the PDO State Bit (either in the PDO Parameter or if it is part of the process data description there. Slave remains in state *Op*. – 1 = Disable SyncManager (outputs) from PDI side and go to ErrSafeOp
0x10F1:02		U16	48	rw	0x0004	■ Sync Error Counter Limit – Limit set by the master when slave shall change its EtherCAT state to *Safe-Op* with AL Status Code 0x1A. The Slave shall reset an internal error counter when it is synchronized again and confirms the state transition from *Safe-Op* to *Op*.

‘Explanation of the elements’ page 78

Diagnosis History

Index:Sub	Name	Type	Offset	Access	Default	Description
0x10F3:00	Diagnosis History	U8	0	ro	0xFF	Diagnosis History
0x10F3:01	Maximum Messages	U8	16	ro	0xFA	Maximum Messages
0x10F3:02	Newest Message	U8	24	ro	0	Newest Message ■ 0x10F3:02 Reading *Newest Message* and save at user program ■ Bit 0 set again: read again 0x10F3:02 and difference to previously read value is the number of new entries ■ 0x10F3:xx: Read entries accordingly
0x10F3:03	Newest Acknowledged Message	U8	32	rw	0	Newest Acknowledged Message
0x10F3:04	New Message Available	Bool	40	ro	0	New Message Available ■ Overwrite Mode: – 0: newest message was read – 1: newest message was not read ■ Acknowledge Mode: – 0: no unacknowledged message – 1: diagnosis messages are available which can be acknowledged
0x10F3:05	Flags	U16	48	rw	0	Flags
0x10F3:06	Diagnosis message	OCTET-STRING		ro	0	Diagnosis message
...		...		...	...	...
0x10F3:255	Diagnosis message	OCTET-STRING		ro	0	Diagnosis message

➔ *'Explanation of the elements' page 78*

In the *Diagnosis History*, both the diagnostic and process alarms, as information about status changes as well as EtherCAT-specific error messages are entered chronologically.

Output mapping modules

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1600 ... 0x163F:00	RxPDO Map	U8	0	ro/rw[1]	Number of outputs at this slot	Entry only available at slots with output modules.
0x1600... 0x163F:01	Output mapping	U32	16	ro/rw[1]		■ Bit 7...0: length of the mapped objects in bits ■ Bit 15...8: subindex of the mapped object ■ Bit 31...16: index of the mapped object Example: 0x7000:01, 1 > at slot 0 the first output has a length of 1 bit.
0x1600... 0x163F:02	Output mapping	U32		ro/rw[1]		
0x1600... 0x163F: ...	...					

➔ *'Explanation of the elements' page 78*

1) rw only at output modules with variable process output image

Input mapping modules

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1A00 ... 0x1A3F:00	TxPDO map	U8	0	ro/rw[1]	Number of inputs at this slot	Entry only available at slots with input modules.
0x1A00... 0x1A3F:01	Input mapping	U32	16	ro/rw[1]		■ Bit 7...0: Length of the mapped objects in bits ■ Bit 15...8: Subindex of the mapped object ■ Bit 31...16: Index of the mapped object Example: 0x6000:01, 8 > at slot 0 the first input has a length of 8 bit.
0x1A00... 0x1A3F:02	Input mapping	U32		ro/rw[1]		
0x1A00... 0x1A3F: ...	...					

➪ *'Explanation of the elements' page 78*

1) rw only at output modules with variable process input image

Due to the system the mapping should not be changed in the configuration tool, otherwise the process image could be influenced!

Input Mapping coupler

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1AFF:00	Status PDO Mapping	U8	0	ro	4	Mapping of Status PDO ■ Bit 7...0: Length of the mapped objects in bits ■ Bit 15...8: Subindex of the mapped object ■ Bit 31...16: Index of the mapped object
0x1AFF:01	Input Mapping 01	U32	16	ro	0xF1000120	Mapping of hardware interrupt counter
0x1AFF:02	Input Mapping 02	U32	48	ro	0xF1000220	Mapping of diagnostic interrupt counter

➪ *'Explanation of the elements' page 78*

Input Mapping coupler

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1B00:00	Status PDO Mapping	U8	0	ro	2	Mapping of Status PDO ■ Bit 7...0: Length of the mapped objects in bits ■ Bit 15...8: Subindex of the mapped object ■ Bit 31...16: Index of the mapped object
0x1B00:01	Input Mapping 01	U32	16	ro	0xF1000340	Extended Diagnostic Data Diagnostic Module Status
0x1B00:02	Input Mapping 02	U32	48	ro	0x10F30401	*New Message* available

➪ *'Explanation of the elements' page 78*

The PDOs 0x1B00 and 0x1B01 are optional and can be activated or deactivated as required.

- Default:
  - PDO 0x1B00 is activated
  - PDO 0x1B01 is deactivated

Input Mapping coupler

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1B01:00	Status PDO Mapping	U8	0	ro	2	Mapping of Status PDO ■ Bit 7...0: Length of the mapped objects in bits ■ Bit 15...8: Subindex of the mapped object ■ Bit 31...16: Index of the mapped object
0x1B01:01	Input Mapping 01	U32	16	ro	0x1C322001	Sync Error Status Sync Manager Out
0x1B01:02	Input Mapping 02	U32	48	ro	0x1C332001	Sync Error Status Sync Manager In

➪ *'Explanation of the elements' page 78*

The PDOs 0x1B00 and 0x1B01 are optional and can be activated or deactivated as required.

- Default:
  – PDO 0x1B00 is activated
  – PDO 0x1B01 is deactivated

SyncManager Type

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1C00:00	SyncManager Type	U8	0	ro	4	Sync Manager Communication Type
0x1C00:01	SyncManager 1	U8	16	ro	1	1: Mailbox receive (master to slave)
0x1C00:02	SyncManager 2	U8	24	ro	2	2: Mailbox send (slave to master)
0x1C00:03	SyncManager 3	U8	32	ro	3	3: Process data output
0x1C00:04	SyncManager 4	U8	40	ro	4	4: Process data input

➪ *'Explanation of the elements' page 78*

RxPDO Assign

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1C12:00	RxPDO Assign	U8	0	ro	Number of modules with process output data	Sync Manager 2 PDO Assignment (0x16yy)
0x1C12:01	RxPDO Mapping 1	U16	16	ro		PDO Mapping object index of assigned PDO
0x1C12 ...	...	U16	...	ro		...
0x1C12:40	RxPDO Mapping 64	U16		ro		

➪ *'Explanation of the elements' page 78*

TxPDO Assign

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1C13:00	TxPDO Assign	U8	0	ro	Number of input PDOs of coupler and modules	Sync Manager 3 PDO Assignment
0x1C13:01	TxPDO Mapping 1	U16	16	ro		PDO Mapping object index of assigned PDO (0x1Ayy)
0x1C13:02	TxPDO Mapping 2	U16		ro		
0x1C13: ...	...	U16	...	ro		...
0x1C13:40	TxPDO Mapping 64	U16		ro		
➪ *'Explanation of the elements' page 78*						

The EtherCAT coupler also has process input data, which show the alarm status. So there is in the first place the assignment 0x1AFF for the input PDOs of the coupler.

SM output parameter

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1C32:00	SM output parameter	U8	0	ro	32	
0x1C32:01	Synchronization Type	U16	16	rw		- 0x00: Free Run (not synchronized) - 0x01: Synchronous - synchronized with Sync Manager Event - 0x02: DC Sync0 - synchronized with Sync0 Event - 0x03: DC Sync1 - synchronized with Sync1 Event
0x1C32:02	Cycle time	U32	32	rw		- Free Run: Time between two local timer events in ns - Free Run: Time between two local timer events in ns Synchronous SM2 Event: Minimum time between two SM2 events in ns - DC Sync0: Sync0 Cycle Time (Register 0x9A3-0x9A0) in ns
0x1C32:03	Shift time	U32	64	ro		- Time between related event and the associated action (outputs valid hardware) - Shift of output valid equal or greater than 0x1C32:09
0x1C32:04	Synchronization Types supported	U16	96	ro		
0x1C32:05	Minimum cycle time	U32	112	ro		- *Minimum cycle time* supported by the slave (maximum duration time of the local cycle) in ns - It might be necessary to start the Dynamic Cycle Time measurement SI04, Bit 14 and SI08, Bit 0 to get a valid value used in Synchronous or DC Mode
0x1C32:06	Calc and Copy Time	U32	144	ro		Time needed by the application controller to copy the process data from the Sync Manager to the local memory and perform calculations if necessary before the data is sent to the process.
0x1C32:07	Minimum Delay Time	U32	176	ro		Only important for DC Sync0/1 (Synchronization type = 0x02 or 0x03): Minimum Hardware delay time of the slave. Because of software synchronization there could be a distance between the minimum and the maximum delay time.
0x1C32:08	Get Cycle Time	U16	208	ro		- Bit 0: – 0: Measurement of local cycle time stopped – 1: Measurement of local cycle time started If written again, the measured values are reset Used in Synchronous or (DC Mode with variable Cycle Time) - Bit 1: – 0: Reserved – 1: Reset the error counters - Bit 15 … 2: Reserved

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1C32:09	Delay Time	U32	224	ro		Only important for DC Sync0/1 (Synchronization type = 0x02 or 0x03): Hardware delay time of the slave. Time from receiving the trigger (Sync0 or Sync1 Event) to drive output values to the time until they become valid in the process (e.g. electrical signal available).
0x1C32:10	Sync0 Cycle Time	U32	256	rw		Only important for DC Sync0 (Synchronization type = 0x03) and subordinated local cycles: Time between two Sync0 signals if fixed *Sync0 Cycle Time* is needed by the application.
0x1C32:11	SM-Event Missed	U16	288	ro		■ This error counter is incremented when the application expects a SM event but does not receive it in time and as consequence the data cannot be copied any more. ■ Used in DC mode.
0x1C32:12	Cycle Time Too Small	U16	304	ro		■ This error counter is incremented when the cycle time is too small so that the local cycle cannot be completed and input data cannot be provided before the next SM event. ■ Used in Synchronous or DC Mode.
0x1C32:13	Shift Time Too Short	U16	320	ro		■ This error counter is incremented when the time distance between the trigger (Sync0) and the Outputs Valid is too short because of a too short Shift Time or Sync1 Cycle Time. ■ Used in DC Mode.
0x1C32:14	RxPDO Toggle Failed	U16	336	ro		This error counter is incremented when the slave supports the RxPDO Toggle and does not receive new RxPDO data from the master (RxPDO Toggle set to TRUE).
0x1C32:15	Minimum Cycle Distance in ns	U32	352	ro		Used in conjunction with SI 16 to monitor the jitter between two SM events.
0x1C32:16	Maximum Cycle Distance in ns	U32	384	ro		Used in conjunction with SI 15 to monitor the jitter between two SM events.
0x1C32:17	Minimum SM SYNC Distance in ns	U32	416	ro		Used in conjunction with SI 18 to monitor the jitter between the SM event and SYNC0 event in DC-SYNC0 Mode
0x1C32:18	Maximum SM SYNC Distance in ns	U32	448	ro		Used in conjunction with SI 17 to monitor the jitter between the SM event and SYNC0 event in DC-SYNC0 Mode
0x1C32:32	Sync Error	BOOL	480	ro		Shall be supported if SM event Missed or Shift Time Too Short Counter is supported Mappable in TxPDO ■ 0: no Synchronization Error or Sync Error not supported ■ 1: Synchronization Error

‘Explanation of the elements’ page 78

SM input parameter

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1C33:00	SM input parameter	U8	0	ro	32	
0x1C33:01	Synchronization Type	U16	16	rw		■ 0x00: Free Run (not synchronized) ■ 0x01: Synchronous with SM3 Event ■ 0x02: DC Sync0 – synchronized with Sync0 Event ■ 0x03: DC Sync1 – synchronized with Sync1 Event ■ 0x22: Synchronous with SM2 Event
0x1C33:02	Cycle time	U32	32	rw		■ Free Run: Time between two local timer events in ns ■ Free Run: Time between two local timer events in ns Synchronous SM2 Event: Minimum time between two SM2 events in ns ■ DC Sync0: Sync0 Cycle Time (Register 0x9A3-0x9A0) in ns
0x1C33:03	Shift time	U32	64	ro		■ Time between related event and the associated action (inputs latched from hardware) ■ Shift of Input Latch equal or greater than 0x1C33:09
0x1C33:04	Synchronization Types supported	U16	96	ro		
0x1C33:05	Minimum cycle time	U32	112	ro		■ *Minimum cycle time* supported by the slave (maximum duration time of the local cycle) in ns ■ It might be necessary to start the Dynamic Cycle Time measurement SI04, Bit 14 and SI08, Bit 0 to get a valid value. ■ Used in Synchronous or DC Mode
0x1C33:06	Calc and Copy Time	U32	144	ro		Time in ns needed by the application controller to perform calculations on the input values if necessary and to copy the process data from the local memory to the Sync Manager before the data is available for EtherCAT.
0x1C33:07		U32	176	ro		Reserved
0x1C33:08	Get Cycle Time	U16	208	ro		■ Bit 0: – 0: Measurement of local cycle time stopped – 1: Measurement of local cycle time started If written again, the measured values are reset Used in Synchronous or (DC Mode with variable Cycle Time) ■ Bit 1: – 0: Reserved – 1: Reset the error counters ■ Bit 15 … 2: Reserved
0x1C33:09	Delay Time	U32	224	ro		■ Only important for DC Sync1 (Synchronization type = 3), if the Input Latch is started by the Sync1-event ■ Hardware delay time of the slave. ■ Used in DC mode

Index:Sub	Name	Type	Offset	Access	Default	Description
0x1C33:10	Sync0 Cycle Time	U32	256	rw		Only important for DC Sync0 (Synchronization type = 0x03) and subordinated local cycles: Time between two Sync0 signals if fixed Sync0 Cycle Time is needed by the application.
0x1C33:11	SM-Event Missed	U16	288	ro		■ This error counter is incremented when the application expects a SM event but does not receive it in time and as consequence the data cannot be copied any more. ■ Used in DC mode.
0x1C33:12	Cycle Time Too Small	U16	304	ro		■ This error counter is incremented when the cycle time is too small so that the local cycle cannot be completed and input data cannot be provided before the next SM event. ■ Used in Synchronous or DC Mode.
0x1C33:13	Shift Time Too Short	U16	320	ro		■ This error counter is incremented when the time distance between the trigger (Sync0) and the outputs valid is too short because of a too short Shift Time or Sync1 Cycle Time. ■ Used in DC Mode.
0x1C33:14			336			Reserved
...			...			...
0x1C33:18			448			Reserved
0x1C33:32	Sync Error	BOOL	480	ro		Shall be supported if SM event Missed or Shift Time Too Short counter is supported Mappable in TxPDO ■ 0: no Synchronization Error or Sync Error not supported ■ 1: Synchronization Error

‘Explanation of the elements’ page 78

Command

Index:Sub	Name	Type	Offset	Access	Default	Description
0x2000:00	Command	U8	0	ro	3	Command Object (complete access is not possible)
0x2000:01	Execute Command	U32	16	rw	0	Execute Command ■ 0x00000000: Clear all values in object 0x2000 ■ 0x00000001: Reset FMM data ■ 0x00004001: Read web server state – *Parameter* in 0x2000:02 ■ 0x00008001: Set web server state – *Parameter* in 0x2000:02
0x2000:02	Parameter	U32	48	ro	0	Additional parameter for command execution State web server ■ 1: enable ■ 0: disable Will be stored permanently
0x2000:03	Status	U32	80	ro	0	Return value of command execution: ■ 0: initial state ■ 1: command executed ■ 2: command not executed ■ 3: command unknown ■ 4: command not executable in this state

‘Explanation of the elements’ page 78

Parameter EtherCAT coupler

Index:Sub	Name	Type	Offset	Access	Default	Description
0x3000:00	Coupler parameter	U8	0	ro	3	Parameter EtherCAT coupler
0x3000:01	Auto acknowledge	U8	16	rw	1	Auto acknowledge ■ 0x00: deactivated ■ 0x01: activated (default)
0x3000: 02	Free Module Mapping	U8	24	rw	0	Free Module Mapping ■ 0x00: deactivated (default) ■ 0x01: activated
0x3000:03	Default values	U8	32	rw	0	Default values ■ 0x00: deactivated (default) ■ 0x01: activated
0x3000:04	Operation Mode	U8	40	rw	0	Operating mode ■ 0x00: Normal mode (default) ■ 0x01: Enhanced mode
⇘ *'Explanation of the elements' page 78*						

- Auto acknowledge
  - With this object the write access is only possible when the EtherCAT coupler is in *PreOp* or *SafeOp* state. In *Op* state write access is denied.
  - With *Auto acknowledge* = 0 per module the corresponding interrupt counter is increased by 1. Once you acknowledge the interrupt, the interrupt counter is decreased by 1. If there are more modules with a pending interrupt, by the write SDO access always the most recent interrupt is acknowledged (last in - first out).
  - With *Auto acknowledge* = 1 each interrupt is automatically acknowledged by the EtherCAT coupler. In this mode the diagnostics data were always overwritten by new interrupts. Default setting is *Auto acknowledge* = 1. For continuous use you should set *Auto acknowledge* = 1.
- Free Module Mapping
  - With FMM you can use EtherCAT slave stations with different hardware variants without adapting your user program.
  - ⇘ *Chap. 4.5.5 'FMM - Free module mapping' page 62*
- Default values
  - Behavior when BASP (**B**efehls-**A**usgabe-**Sp**erre i.e. command output disable) is activated
  - With *Default values* = 0 and activated BASP all module outputs are switched off.
  - With *Default values* = 1 and activated BASP, the last states of the outputs are retained.

    Please note that this represents a potentially hazardous operating condition. This should only be used for test purposes respectively for troubleshooting.
- Operating mode
  - Operating mode of the EtherCAT coupler
  - In operating mode *Normal mode*, the EtherCAT coupler is backward compatible with the previous version and FSoE safety modules are not supported.
  - In operating mode *Enhanced mode*, the EtherCAT coupler is not backward compatible with the previous version and FSoE safety modules are supported.
  - ⇘ *Chap. 4.3 'Operating Mode - Normal/Enhanced mode' page 54*

Free Module Mapping (FMM) Configuration

Index:Sub	Name	Type	Offset	Access	Default	Description
0x3001:00	Free Module Mapping Configuration	U16	0	ro	0x64	Slot configuration for Free Module Mapping (data will be stored retentive)
0x3001:01	SubIndex 001	U32	16	rw	0	■ Byte 0: Output size of module in byte from max. configuration ■ Byte 1: Input size of module in byte from max. configuration ■ Byte 2: 00 (fix) ■ Byte 3: FMM slot to real configuration – 0: empty slot – 1...64: assigned slot – 255: Virtual module - module does not exist in the actual configuration Subindex = Slot + 1 of max. configuration
...	SubIndex ...	...		...	...	...
↳ *'Explanation of the elements' page 78*						

FMM is the ability to map a real existing module configuration to a custom module configuration. All available real modules can be distributed freely to all available slots. FMM allows the creation of permanent free slots, as well as slots for inserting virtual modules for emulation of non-existent modules.

Parameter SLIO Module

Index:Sub	Name	Type	Offset	Access	Default	Description
0x3100... 0x313F:00	Parameter	U8	0	ro	Number of parameters	Parameter of a SLIO Module
0x3100... 0x313F:01	Param1	U8	16	rw		
0x3100... 0x313F:02	Param2	U8	32	rw		
0x3100... 0x313F: ...	...					
➪ *'Explanation of the elements' page 78*						

With this object the parameters of a System SLIO module may be accessed. Here the addressing by index takes place. Via subindexes the corresponding parameters may be accessed. The allocation of the subindexes may be found in the description of the corresponding module. Also here it is valid that power and clamp modules are not recognized by the EtherCAT coupler and so are not listed and considered during slot allocation.

If the module has parameters it is valid

- Index 0x3100: Access to EtherCAT-Slot 0
- Index 0x3101: Access to EtherCAT-Slot 1
- ...
- Index 0x313F: Access to EtherCAT-Slot 64

> *Please note that in operation mode Enhanced Mode the index of the module-related objects is to be incremented accordingly.*
>
> *The index is calculated as follows:*
>
> $Index_{Enhanced} = Index + (16 \cdot EtherCAT\text{-}Slot)$
>
> ➪ *Chap. 4.3 'Operating Mode - Normal/Enhanced mode' page 54*

In the following example, you have access to the parameters of the module at physical slot 4 via the following index.

Physical slot number	**1**	**2**	**3**	**4**
Module	DI	CM	DO	AI
Index (Operation mode: *Normal Mode*)	*0x3100*[1]	-	*0x3101*[1]	**0x3102**
Index (Operation mode: *Enhanced Mode*)	*0x3100*[1]	-	*0x3110*[1]	**0x3120**
EtherCAT-Slot	0	-	1	**2**

[1] This entry is not listed, because the module has no parameters.

Clear SLIO counter

Index:Sub	Name	Type	Offset	Access	Default	Description
0x4000:00	Clear SLIO counter	U8	0	ro	2	Writing of a value you want to the corresponding subindex clears the counter.
0x4000:01	Clear master counter	U8	16	rw	0	■ Master Counter – Object 0x4001
0x4000:02	Clear module counter	U8	24	rw	0	■ MDL and NDL Counter – Object 0x4002 and 0x4003

☞ *'Explanation of the elements' page 78*

Master counter

Index:Sub	Name	Type	Offset	Access	Default	Description
0x4001:00	Master counter	U8	0	ro	14	Error counter of System SLIO Master
0x4001:01	Expected length error	U16	16	ro	0	
0x4001:02	TimeOut error	U16	32	ro	0	
0x4001:03	StopBit error	U16	48	ro	0	
0x4001:04	FCS error	U16	64	ro	0	
0x4001:05	Telegram length error	U16	80	ro	0	
0x4001:06	Telegram type error	U16	96	ro	0	
0x4001:07	Alarm retry error	U16	112	ro	0	
0x4001:08	Bus idle time error	U16	128	ro	0	
0x4001:09	Wrong node address	U16	144	ro	0	
0x4001:0A	Telegram valid	U16	160	ro	0	
0x4001:0B	Master load	U16	176	ro	0	
0x4001:0C	One Telegram Retry Counter	U16	192	ro	0	
0x4001:0D	Two Telegram Retry Counter	U16	208	ro	0	
0x4001:0E	Three or More Telegram Retry Counter	U16	224	ro	0	

☞ *'Explanation of the elements' page 78*

Module MDL counter

Index:Sub	Name	Type	Offset	Access	Default	Description
0x4002:00	Module MDL counter	U8	0	ro	64	Master Data Line Counter
0x4002:01	Slot 1	U16	16	ro	0	Slot 1
0x4002:02	Slot 2	U16	32	ro	0	Slot 2
0x4002: ...	...	U16	...	ro	0	...
0x4002:40	Slot 64	U16		ro	0	Slot 64
⇨ *'Explanation of the elements' page 78*						

Module NDL counter

Index:Sub	Name	Type	Offset	Access	Default	Description
0x4003:00	Module NDL counter	U8	0	ro	64	Node Data Line Counter
0x4003:01	Slot 1	U16	16	ro	0	Slot 1
0x4003:02	Slot 2	U16	32	ro	0	Slot 2
0x4003: ...	...	U16	...	ro	0	...
0x4003:40	Slot 64	U16		ro	0	Slot 64
⇨ *'Explanation of the elements' page 78*						

Module information

Index:Sub	Name	Type	Offset	Access	Default	Description
0x4100:00	Module Information	U8	0	ro	8	Module Identity Information IM 053EC
0x4100:01	Module Name	VSTRG	16	ro	0	Module Name
0x4100:02	Module ID	U32	256	ro	0	Module ID
0x4100:03	Serial Number	VSTRG	288	ro	0	Serial number of the module
0x4100:04	Product Version	VSTRG	448	ro	0	Product version of the module
0x4100:05	Hardware Version	VSTRG	520	ro	0	Hardware version of the module
0x4100:06	Software Version	VSTRG	536	ro	0	Software version of the module
0x4100:07	FPGA Version	U16	632	ro	0	FPGA version
0x4100:08	Mx File	VSTRG	648	ro	0	Mx file of the coupler
➪ *'Explanation of the elements' page 78*						

When accessing the module information of IM 053EC the corresponding module is addressed via the index (0x4100 + EtherCAT-Slot). Subindexes give you access to the corresponding module information.

Please note that in operation mode Enhanced Mode the index of the module-related objects is to be incremented accordingly.

The index is calculated as follows:

$Index_{Enhanced}$ = Index + (16 · EtherCAT-Slot)

➪ *Chap. 4.3 'Operating Mode - Normal/Enhanced mode' page 54*

Hardware interrupt data

Index:Sub	Name	Type	Offset	Access	Default	Description
0x5000:00	Hardware interrupt data	U8	0	ro	6	Current hardware interrupt data
0x5000:01	Slot number	U8	16	ro	0x00	EtherCAT-Slot of the module which reports an interrupt
0x5000:02	Hardware interrupt data 00	U8	24	ro	0x00	Hardware interrupt data (see module description)
0x5000:03	Hardware interrupt data 01	U8	32	ro	0x00	
0x5000:04	Hardware interrupt data 02	U8	40	ro	0x00	
0x5000:05	Hardware interrupt data 03	U8	48	ro	0x00	
0x5000:06	Acknowledge	U8	56	rw	0x00	Any value resets the hardware interrupt counter and acknowledges if necessary the interrupt.

☞ *'Explanation of the elements' page 78*

- If the object 0xF100 indicates a hardware interrupt here the current hardware interrupt data may be accessed. The allocation of the hardware interrupt data may be found in the description of the corresponding module.
- With de-activated *Auto-Acknowledge* of the EtherCAT coupler the *hardware interrupt counter* of object 0xF100 is reset and the hardware interrupt is acknowledged. For this you have to write any value you like to subindex 0x06 of index 0x5000.

Diagnostic Interrupt Data (byte 1 ... 4)

Index:Sub	Name	Type	Offset	Access	Default	Description
0x5002:00	Diagnostic Interrupt Data	U8	0	ro	6	Current diagnostic interrupt data
0x5002:01	Slot number	U8	16	ro	0x00	EtherCAT-Slot of the module which reports an interrupt.
0x5002:02	Diagnostic data 00	U8	24	ro	0x00	Byte 1 ... 4 of the diagnostic data (see module description)
0x5002:03	Diagnostic data 01	U8	32	ro	0x00	
0x5002:04	Diagnostic data 02	U8	40	ro	0x00	
0x5002:05	Diagnostic data 03	U8	48	ro	0x00	
0x5002:06	Acknowledge	U8	56	rw	0x00	Any value resets the diagnostic interrupt counter and acknowledges if necessary the interrupt.

☞ *'Explanation of the elements' page 78*

- If the object 0xF100 indicates a diagnostic interrupt here the first 4 bytes of the current diagnostics interrupt data may be accessed. The allocation of the diagnostics data may be found in the description of the corresponding module.
- With de-activated *Auto-Acknowledge* of the EtherCAT coupler the *diagnostic interrupt counter* of object 0xF100 is reset and the diagnostic interrupt is acknowledged. For this you have to write any value you like to subindex 0x06 of index 0x5002.

Diagnostic Data (byte 1 ... n)

Index:Sub	Name	Type	Offset	Access	Default	Description
0x5005:00	Diagnostic Data	U8	0	ro	19	
0x5005:01	Slot number	U8	16	rw	0	By reading this index here the EtherCAT-Slot of the module may be found, to which the following diagnostic data belong to. By writing of an EtherCAT-Slot to this index the diagnostic data of the corresponding module may be read.
0x5005:02	Module error	U8	24	ro	0	Diagnostics data of the module (see module description)
0x5005:03	Module type	U8	32	ro	0	
0x5005:04	Fix	U8	40	ro	0	
0x5005:05	Communication/ Process interrupt	U8	48	ro	0	
0x5005:06	Channel type	U8	56	ro	0	
0x5005:07	Number of diagnostic bits	U8	64	ro	0	
0x5005:08	Number of channels	U8	72	ro	0	
0x5005:09	Channel error	U8	80	ro	0	
0x5005:0A	Channel-specific error channel 0	U8	88	ro	0	
0x5005:0B	Channel-specific error channel 1	U8	96	ro	0	
0x5005:0C	Channel-specific error channel 2	U8	104	ro	0	
0x5005:0D	Channel-specific error channel 3	U8	112	ro	0	
0x5005:0E	Channel-specific error channel 4	U8	120	ro	0	
0x5005:0F	Channel-specific error channel 5	U8	128	ro	0	
0x5005:10	Channel-specific error channel 6	U8	136	ro	0	
0x5005:11	Channel-specific error channel 7	U8	144	ro	0	
0x5005:13	-	U8	152	ro	0	
0x5005:14	µs-Ticker	U32	160	ro	0	

☞ *‘Explanation of the elements’ page 78*

The whole diagnostic data of a module may be accessed with this object. Here you may either access current diagnostic data or the diagnostic data of a module on any EtherCAT-Slot.

Read input data

Index:Sub	Name	Type	Offset	Access	Default	Description
0x6000... 0x603F:00	Input data	U8	0	ro	60	Number of subindexes of the input data for the corresponding EtherCAT-Slot.
0x6000... 0x603F:01		U32	16	ro		Input data (see module description)
0x6000... 0x603F:02		U32		ro		
0x6000... 0x603F: ...	...	...		ro		

➡ *'Explanation of the elements' page 78*

> *Please note that in operation mode Enhanced Mode the index of the module-related objects is to be incremented accordingly.*
>
> *The index is calculated as follows:*
>
> $Index_{Enhanced} = Index + (16 \cdot EtherCAT\text{-}Slot)$
>
> ➡ *Chap. 4.3 'Operating Mode - Normal/Enhanced mode' page 54*

- With this object the input data of a System SLIO module may be read.
- Here the addressing of the EtherCAT-Slot via index (0x6000 + EtherCAT-Slot) takes place.
- The corresponding input data may be accessed via subindexes.
  - The allocation of the subindexes may be found in the description of the corresponding module.

Read output data

Index:Sub	Name	Type	Offset	Access	Default	Description
0x7000... 0x703F:00	Output data	U8	0	ro	60	Number of subindexes of the output data for the corresponding EtherCAT-Slot.
0x7000... 0x703F:01		U32	16	ro		Output data (see module description)
0x7000... 0x703F:02		U32		ro		
0x7000... 0x703F: ...	...	...		ro		

☞ *‘Explanation of the elements’ page 78*

> *Please note that in operation mode Enhanced Mode the index of the module-related objects is to be incremented accordingly.*
>
> *The index is calculated as follows:*
>
> $Index_{Enhanced} = Index + (16 \cdot EtherCAT\text{-}Slot)$
>
> ☞ *Chap. 4.3 ‘Operating Mode - Normal/Enhanced mode’ page 54*

- With this object the output data of a System SLIO may be read.
- Here the addressing via index (0x7000 + EtherCAT-Slot) takes place.
- The corresponding output data may be accessed via subindexes.
  - The allocation of the subindexes may be found in the description of the corresponding module.

> *Information concerning the write access to outputs may be found in the manual of your EtherCAT master.*

Modular device profile

Index:Sub	Name	Type	Offset	Access	Default	Description
0xF000:00	Modular device profile	U8	0	ro	4	Contains information to interpret the objects in function areas of the modules
0xF000:01	Module index distance	U16	16	ro	1	Index distance between two modules
0xF000:02	Maximum number of modules	U16	32	ro	0x40 (64)	Maximum number of modules
0xF000:03	General configuration	U32	48	ro	0	
0xF000:04	General information	U32	80	ro	0	

☞ *‘Explanation of the elements’ page 78*

Module Profile List

Index:Sub	Name	Type	Offset	Access	Default	Description
0xF010:00	Module Profile List	U8	0	ro	0x40 (64)	Module ID of the modules ■ Bit 15...0: Module profile information of the module ■ Bit 31…16: fix (0) Module profile specific information
0xF010:01	Subindex 001	U32	16	ro	0	
0xF010:02	Subindex 002	U32	48	ro	0	
0xF010: ...	...	U32	...	ro		
0xF010:40	Subindex 064	U32		ro	0	
☞ *'Explanation of the elements' page 78*						

Configured Module Ident List

Index:Sub	Name	Type	Offset	Access	Default	Description
0xF030:00	Configured Module Ident List	U8	0	rw	0x40 (64)	Module ID of the configured modules
0xF030:01	Subindex 001	U32	16	rw	0	
0xF030:02	Subindex 002	U32	48	rw	0	
0xF030: ...	...	U32	...	rw		
0xF030:40	Subindex 064	U32		rw	0	
☞ *'Explanation of the elements' page 78*						

Write access to the object 0xF030 is only possible with CompleteAccess. Single access is not allowed!

- When writing to a subindex the entered value is compared with the module ID of the same subindex of index 0xF050. So the configuration may be checked.
- Some configuration tools cannot automatically be write to this index, thus some configuration errors may not be detected.

Detected address list

Index:Sub	Name	Type	Offset	Access	Default	Description
0xF040:00	Detected address list	U8	0	ro	0x40 (64)	Addresses of the modules, which were recognized by the coupler.
0xF040:01	Subindex 001	U16	16	ro		
0xF040:02	Subindex 002	U16	32	ro		
0xF040: ...	...	U16		ro		
0xF040:40	Subindex 064	U16		ro		
☞ *'Explanation of the elements' page 78*						

Detected module list

Index:Sub	Name	Type	Offset	Access	Default	Description
0xF050:00	Detected module list	U8	0	ro	0x40 (64)	Module IDs of the modules, which were recognized by the coupler.
0xF050:01	Subindex 001	U32	16	ro		
0xF050:02	Subindex 002	U32	48	ro		
0xF050: ...	...	U32		ro		
0xF050:40	Subindex 064	U32		ro		
☞ *'Explanation of the elements' page 78*						

Diagnostic

Index:Sub	Name	Type	Offset	Access	Default	Description
0xF100:00	Diagnostic	U8	0	ro	3	
0xF100:01	Hardware interrupt counter	U32	16	ro	0x00000000	Counter for hardware interrupt
0xF100:02	Diagnostic interrupt counter	U32	48	ro	0x00000000	Counter for diagnostic interrupt
0xF100:03	Diagnostic Module Status	U64	80	ro	0x00000000	64 bit array, each bit represents a pending (but acknowledged) diagnostic alarm
☞ *'Explanation of the elements' page 78*						

- With de-activated *Auto-Acknowledge* of the EtherCAT coupler the corresponding counter is set to 1 until you acknowledge this accordingly. For this you have to write any value you like to subindex 0x06 under the accordingly assigned index.
- With activated *Auto-Acknowledge* here you will find the number of hardware and diagnostics interrupts, which happened since the last interrupt reset. To reset the corresponding counter you have to write any value you like to the subindex 0x06 under the accordingly assigned index.

There are the following index assignments:

- Write to 0x06 of index 0x5000:
  - Reset of counter hardware interrupt
- Write to 0x06 of index 0x5002:
  - Reset of counter diagnostic interrupt

4.10 Error handling

4.10.1 Overview

In EtherCAT there are the following two types of error messages:

- Emergency error message
- Standard error message

4.10.2 Emergency error message

Emergency error messages are generated on errors during state changes respectively at hardware and diagnostic interrupts if the interrupt is activated by parameterization. Emergency error messages are released within device-internal mechanisms and sent to the master by means of the mailbox service.

Frame structure

Byte

0	1	2	3	4	5	6	7
EEC		ER	DATA				

EEC - **E**mergency **E**rror **C**odes following ETG specifications:

0xFFxx: Vendor-specific emergency message (VIPA = 0xFF00)

0xA0xx: State Machine transition error, e.g. due to a configuration error (length, address, settings). More information concerning this may be found at ETG.

ER - Error register - State of the State Machine:

01: Init

02: Pre-Op

03: BootStrap

04: Safe-Op

08: Op

DATA - Error data with more information concerning the error

With a hardware interrupt or diagnostic interrupt respectively in DATA the *Manufacturer Specific Error Field* (MEF) may be found. The field has the following structure:

Byte

0	1	2	3	4	5	6	7
EEC		ER	SLOT	TYPE	DIAG		
			MEF				

SLOT - EtherCAT-Slot of the module where is the error from

TYPE - Interrupt type

01: Hardware interrupt

02: Diagnostics interrupt

DIAG - Error code, which is described at corresponding module.

4.10.3 Standard error messages

Emergency error messages are released within device-internal mechanisms and sent to the EtherCAT master. The following standard error messages my occur:

AL Status Code

To get more information about the error the EtherCAT master reads the AL Status Code (AL = application layer). This is allocated in the EtherCAT coupler in register 0x0134.

EtherCAT specific error messages

Code	Description	Current State	Resulting State
0x0000	No error	Any	Current state
0x0001	Unspecified error	Any	Any + E
0x0002	No Memory	Any	Any + E
0x0011	Invalid requested state change	I > S, I > O, P > O, O > B, S > B, P > B	Current state + E
0x0012	Unknown requested state	Any	Current state + E
0x0013	Bootstrap not supported	I > B	I + E
0x0014	No valid firmware	I > P	I + E
0x0015	Invalid mailbox configuration	I > B	I + E
0x0016	Invalid mailbox configuration	I > P	I + E
0x0017	Invalid SyncManager configuration	P > S, S > O	Current state + E
0x0018	No valid inputs available	O, S > O	S + E
0x0019	No valid outputs	O, S > O	S + E
0x001A	Synchronization error	O, S > O	S + E
0x001B	SyncManager watchdog	O, S	S + E
0x001C	Invalid SyncManager Types	O, S, P > S	S + E
0x001D	Invalid Output Configuration	O, S, P > S	S + E
0x001E	Invalid Input Configuration	O, S, P > S	P + E
0x001F	Invalid Watchdog Configuration	O, S, P > S	P + E
0x0020	Coupler needs cold start	Any	Current state + E
0x0021	Coupler needs INIT	B, P, S, O	Current state + E
0x0022	Coupler needs PREOP	S, O	S + E, O + E
0x0023	Coupler needs SAFEOP	O	O + E
0x0024	Invalid Input Mapping	P > S	P + E
0x0025	Invalid Output Mapping	P > S	P + E
0x0026	Inconsistent Settings	P > S	P + E
0x0027	Free-run not supported	P > S	P + E
0x0028	Synchronization not supported	P > S	P + E
0x0029	Free-run needs 3 Buffer Mode	P > S	P + E
0x002A	Background Watchdog	S, O	P + E
0x002B	No Valid Inputs and Outputs	O, S > O	S + E
0x002C	Fatal Sync Error	O	S + E
0x002D	No Sync Error	S > O	S + E
0x0030	Invalid DC SYNC Configuration	O, S > O, P > S	P + E, S + E
0x0031	Invalid DC Latch Configuration	O, S > O, P > S	P + E, S + E

Code	Description	Current State	Resulting State
0x0032	PLL Error	O, S > O	S + E
0x0033	DC Sync IO Error	O, S > O	S + E
0x0034	DC Sync Timeout Error	O, S > O	S + E
0x0035	DC Invalid Sync Cycle Time	P > S	P + E
0x0036	DC Sync0 Cycle Time	P > S	P + E
0x0037	DC Sync1 Cycle Time	P > S	P + E
0x0041	MBX_AOE	B, P, S, O	Current state + E
0x0042	MBX_EOE	B, P, S, O	Current state + E
0x0043	MBX_COE	B, P, S, O	Current state + E
0x0044	MBX_FOE	B, P, S, O	Current state + E
0x0045	MBX_SOE	B, P, S, O	Current state + E
0x004F	MBX_VOE	B, P, S, O	Current state + E
0x0050	EEPROM No Access	Any	Any + E
0x0051	EEPROM Error	Any	Any + E
0x0060	Coupler Restarted Locally	Any	I
< x8000	Reserved		

4.10.3.1 SyncManager watchdog

0x001B

If there is no communication to the master possible e.g. line is interrupted, you will get the error message 0x001B after the SyncMaster watchdog time has been elapsed. The time is to be configured in the master.

- SyncManager watchdog is used for "frame monitoring"
  – If no more frames arrive to the slave in a certain time, it comes to the watchdog and the slave changes to SafeOp with the AlStatusCode 0x001B
- By default, the SyncManager watchdog is always active with a time of 100ms
- Changing in the parameters e.g. via the EtherCAT Manager is changed once and remains valid until new values are received. The values are not stored remanently
- Time from the SyncManager watchdog should always be greater than the set bus cycle time at the EtherCAT Master. Since the SyncManager watchdog is always triggered when a process data frame is received
- If parameterized, substitute values are displayed for the modules

4.10.4 SDO error code

If a SDO request is negative evaluated, an according SDO error code is released. In the following table the possible errors are listed.

Code	Description
0x05030000	Toggle bit not alternated
0x05040000	SDO protocol timed out

Code	Description
0x05040001	Client/server command specifier not valid or unknown
0x05040002	Invalid block size (block mode only)
0x05040003	Invalid sequence number (block mode only)
0x05040004	CRC error (block mode only)
0x05040005	Out of memory
0x06010000	Unsupported access to an object
0x06010001	Attempt to read a write only object
0x06010002	Attempt to write a read only object
0x06020000	Object does not exist in the object dictionary
0x06040041	Object cannot be mapped to the PDO
0x06040042	The number and length of the objects to be mapped would exceed PDO length
0x06040043	General parameter incompatibility reason
0x06040047	General internal incompatibility in the device
0x06060000	Access failed due to an hardware error
0x06070010	Data type does not match, length of service parameter does not match
0x06070012	Data type does not match, length of service parameter too high
0x06070013	Data type does not match, length of service parameter too low
0x06090011	Subindex does not exist
0x06090030	Value range of parameter exceeded (only for write access)
0x06090031	Value of parameter written too high
0x06090032	Value of parameter written too low
0x06090036	Maximum value is less than minimum value
0x08000000	General error
0x08000020	Data cannot be transferred or stored to the application
0x08000021	Data cannot be transferred or stored to the application because of local control
0x08000022	Data cannot be transferred or stored to the application because of the present device state
0x08000023	Object directory dynamic generation fails or no object directory is present (e.g. object directory is generated from file and generation fails because of an file error)

4.11 Firmware update

Current firmware at www.vipa.com

The latest firmware versions are to be found in the service area at www.vipa.com. Load the Px000314.pkg file.

> **CAUTION!**
>
> – When installing a new firmware you have to be extremely careful. Under certain circumstances you may destroy the slave station, for example if the voltage supply is interrupted during transfer or if the firmware file is defective. In this case, please call the YASKAWA Hotline!
>
> – Please regard that the version of the update firmware has to be different from the existing firmware otherwise no update is executed.

Precondition

- There is an Ethernet respectively remote connection between the PC and the VIPA EtherCAT slave station, where a firmware update is to be established.

Proceeding

1. Make a connection from the VIPA slave 053-1EC01 to the EtherCAT configuration tool.
2. Put the slave in state Bootstrap.

> *Refer to possible system properties (when using the EtherCAT Manager, the master must be in the state Init).*

3. Select file Px000314.pkg for download via FoE.
4. Enter the following FoE parameters:

   Filename: Px000314

   Password (hex): 0x0000000

   Timeout (ms): 100000 (if available in the configuration tool)
5. Download the package to your slave.

> – *After the package is completely transferred to the slave, the firmware update starts (recognizable by the alternately flashing SF and MT LEDs).*
>
> – *During this process, the slave must never be disconnected from the power supply.*

6. After the firmware update has been carried out by the slave (SF and MT LEDs stop flashing), a PowerCycle must be performed on the slave.

   ⇨ With this operation the new firmware file is taken.

4.12 Replace EtherCAT slave 053-1EC00 with 053-1EC01

It is not possible to copy modules and their properties / configuration from EtherCAT Slave 053-1EC00 to EtherCAT Slave 053-1EC01 since the EtherCAT slaves come from different ESI files.

When replacing an EtherCAT slave 053-1EC00 by 053-1EC01 an adaptation of your project is required. When replacing the EtherCAT slaves, make sure that all settings and parameters of the 053-1EC00 EtherCAT slave that deviate from the standard setting are transferred to the EtherCAT slave 053-1EC01. Therefore, these settings should be saved appropriately before the replacement. Precondition for replacing the EtherCAT slave 053-1EC00 by 053-1EC01 is that the ESI file for the EtherCAT slave 053-1EC01 is installed in the configuration tool. The replacement takes place in the configuration by deleting the EtherCAT slave 053-1EC00 and adding the EtherCAT slave 053-1EC01. In the following all settings are listed, which should be noted.

Settings

- EtherCAT station address
  - If the station address does not match that of the EtherCAT Slave 053-1EC00, there is a topology error. The EtherCAT slave is not controlled.
- Position of the EtherCAT slave on the bus
  - If the position of the EtherCAT slave on the bus does not match the position of the EtherCAT slave 053-1EC00, there is a topology error. The EtherCAT slave is not controlled.
- Slot assignment of the modules
  - As long as the total IO size of the EtherCAT slave matches the configured size of the master, the EtherCAT slave can switch to Operational (*Op*). If the slot assignment is changed, the wrong process data of the modules is transmitted
- Download the slot configuration
  - If this option was activated for the EtherCAT slave 053-1EC00, then this option should also be activated for the EtherCAT slave 053-1EC01. Please pay attention to an identical slot assignment. If the slot configuration fits between master and 053-1EC01 but deviates from the slot configuration of EtherCAT slave 053-1EC00, this deviation is not recognized!
- PDO mapping of the modules
  - For modules with variable IO size, for this modules the PDO mapping of the 053-1EC01 should be set exactly the same as for the 053-1EC00. Otherwise, the slave does not change to *Safe-Op* or *Op*.
- CoE init commands
  - The CoE init commands at the 053-1EC01 should be made the same as with the 053-1EC00. If there are deviations, the EtherCAT slave can switch to *Op*, but there is a possibility that the EtherCAT slave or its modules will not behave as expected.

- *HotConnect* address
  - If the 053-1EC00 was addressed via a *HotConnect* address, then for the 053-1EC01 this *HotConnect* address is taken to the master configuration. Also the 053-1EC01 must use this *HotConnect* address at the bus.

    For this there are the following possibilities:

    Explicit device ID via address switch:

    If the *HotConnect* address is less than 128, you can set the *HotConnect* address via the address switch on the EtherCAT slave. ↬ *Chap. 4.2 'Setting the HotConnect address' page 53*

    Configured Station Alias:

    If the *HotConnect* address is greater than 128, you can set the *HotConnect* address via *Configured Station Alias* in the EtherCAT configuration. ↬ *Chap. 4.2.2 'HotConnect via Configured Station Alias' page 54*
- Parameter - Advanced options
  - In addition, there are other parametrization options for the EtherCAT Slave as advanced options. These settings, such as *SyncManager Watchdog* or *State Change Timeouts*, are slave-specific settings. These parameters should be checked.

Additional features

The following features were added:

- Free module mapping: FMM
  - When replacing the EtherCAT slaves, please note that FMM is deactivated on the EtherCAT slave 053-1EC01. Otherwise, configuration errors are reported which cause the EtherCAT slave not to change to *Safe-Op* or *Op*.
- Operating modes:
  - The EtherCAT slave 053-1EC00 can only be operated with the *SyncManager-Event* operating mode. The EtherCAT slave 053-1EC01 additionally has the operating modes *Free Run* and *Synchron with Distributed Clocks (DC)*. In order to be able to use these operating modes, additional parametrizations are required. The default operating mode is *SyncManager-Event*. When replacing the EtherCAT slaves, there is nothing to be considered regard to the operating mode.
- Diagnostics:
  - The slave has an additional CoE object (0x10F3) via which a diagnostic history can be shown.

CoE Object Dictionary

The following objects were adapted and have a different meaning. Please also note these deviations in your user program.

053-1EC00	053-1EC01
Object 0x100B ■ Specification of the System SLIO version ■ Calculation from FPGA version of the EtherCAT slave and the modules.	Object was deleted.
Object 0x4100 ■ Information about FPGA or System SLIO version.	Object was deleted or has got another meaning.
Object 0x4101 ■ List of all FPGA versions of all connected modules.	Object was deleted or has got another meaning.
Object 0x4102 ■ List with all firmware versions of all connected modules.	Object was deleted or has got another meaning.
Object 0x4103 ■ List with all serial numbers of all connected modules.	Object was deleted or has got another meaning.
Object 0x4104 ■ List of all Mx-File names of all connected modules.	Object was deleted or has got another meaning.
	Object 0x41xx (new) For each module there is an object 0x41xx, where "xx" corresponds to the module's EtherCAT slot. You can use the subindices to determine all relevant product information about the module (see manual).

Process data

The 053-1EC01 occupies a larger address range of the process input data than the 053-1EC00. Depending on the firmware of the 053-1EC01 used, proceed as follows:

053-1EC01 from version V1.0.3 EtherCAT Manager - Rev. 1	■ Set the S7 addresses for the input bytes to a free address range. In this way you can take the previously used S7 addresses for your modules. If the EtherCAT slave itself is explicitly addressed in the user program, then the addresses in the user program must be adapted at these points.
053-1EC01 from version V1.1.2 EtherCAT Manager - Rev. 2	■ The 053-1EC01 offers variable PDO mapping for its own process data. ■ To achieve the same process data size as for the 053-1EC00, only the PDO 0x1AFF must be activated. The PDOs 0x1B00 and 0x1B01 should be deactivated.

Last step

The changed hardware configuration and (if necessary) the adapted user program must be finally transmitted to the controller. If the system does not behave as expected, use the corresponding diagnostics functions of the controller and the EtherCAT slave.

Appendix

Content

A History of changes

Rev.	Changes
18-06	The manual was created.
18-09	Chapter "Deployment" ■ Description "Synchronization modes" was changed.
18-49	Chapter "Hardware description" ■ Description "Properties" was changed. ■ Description "Address area" was added. Chapter "Deployment" ■ Description "FMM - Free module mapping" was added. ■ Description "Accessing the web server via Ethernet" was added. ■ Description "Object Dictionary" was changed. ■ Description "Replace EtherCAT slave 053-1EC00 with 053-1EC01" was added.
21-10	General ■ CI adjustment was made. Chapter "Basics and Mounting" ■ "System conception" was revised. ■ Description "Hardware revision" was added. ■ Description "Easy Maintenance" was added. Chapter "Deployment" ■ Description "Operating Mode - Normal/Enhanced mode" was added. ■ Description "Accessing the I/O area" was changed. ■ Description "Accessing the parameter data" was changed. ■ Description "FMM - Free module mapping" was changed. ■ Description "Easy Maintenance" was added. ■ Description "Access to module information" was changed. ■ Description "Accessing the web server via Ethernet" was changed. ■ Description "Object Dictionary" was changed.