YASKAWA

VIPA System SLIO

CP | 040-1CA00 | Manual

HB300 | CP | 040-1CA00 | en | 18-28

CP 040 - RS422/485

www.vipa.com/en/service-support/manuals

VIPA CONTROLS

VIPA GmbH
Ohmstr. 4
91074 Herzogenaurach
Telephone: 09132-744-0
Fax: 09132-744-1864
Email: info@vipa.com
Internet: www.vipa.com

Table of contents

1 General

1.1 Copyright © VIPA GmbH

All Rights Reserved

For permission to reproduce or distribute, please contact: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

EMail: info@vipa.de

http://www.vipa.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.

CE Conformity Declaration

Hereby, VIPA 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 VIPA customer service organization.

Trademarks

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S and Commander Compact are registered trademarks of VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 is a registered trademark of profichip 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.

Information product support

Contact your local VIPA Customer Service Organization representative if you wish to report errors or questions regarding the contents of this document. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744-1204

EMail: documentation@vipa.de

Technical support

Contact your local VIPA Customer Service Organization representative if you encounter problems with the product or have questions regarding the product. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Tel.: +49 9132 744-1150 (Hotline)

EMail: support@vipa.de

1.2 About this manual

Objective and contents

This manual describes the CP 040-1CA00 of the System SLIO from VIPA. It contains a description of the structure, project engineering and deployment.

Product	Order number	as of state:	
		HW	FW
CP 040 RS422/485	040-1CA00	01	V1.0.1

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.

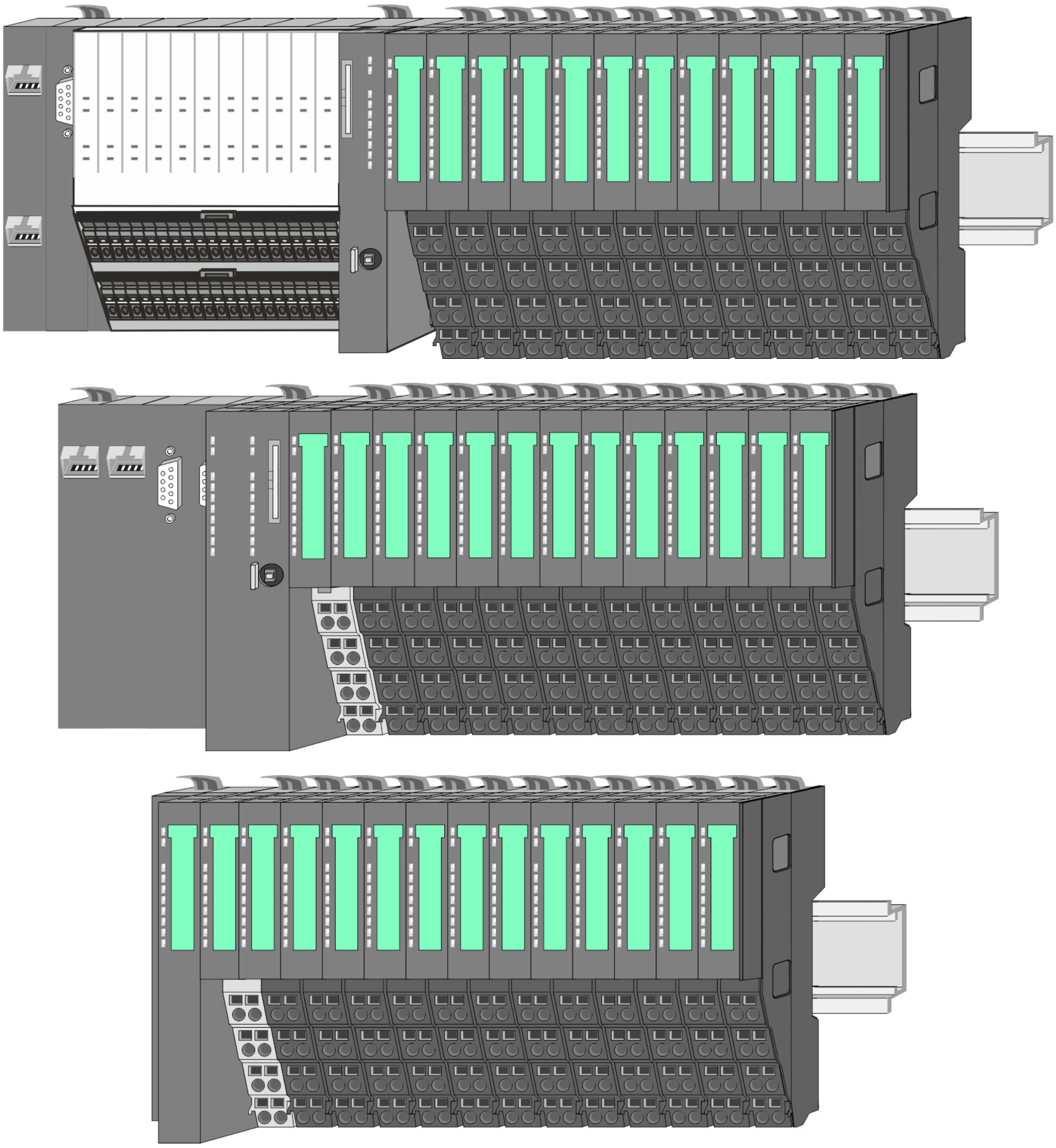
Modifying electrostatic sensitive modules you should only use soldering irons with grounded tips.

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

2.2 System conception

2.2.1 Overview

System SLIO is a modular automation system for assembly on a 35mm mounting rail. By means of the peripheral modules with 2, 4 or 8 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 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.

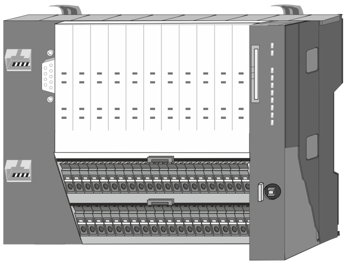


2.2.2 Components

- CPU (head module)
- Bus coupler (head module)
- Line extension
- Periphery modules
- Accessories

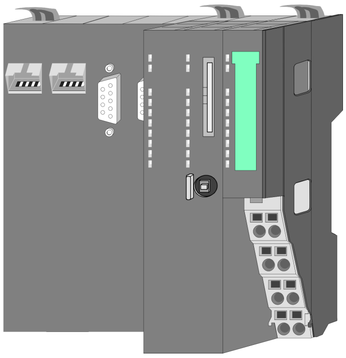
CAUTION!
Only modules of VIPA may be combined. A mixed operation with third-party modules is not allowed!

CPU 01xC



With this CPU 01xC, the CPU 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 supply CPU electronic and the I/O components are power supplied as well as the electronic of the connected periphery modules. To connect the power supply of the I/O components and for DC 24V power supply of via backplane bus connected peripheral 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

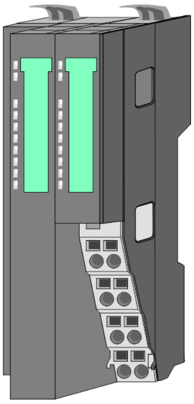


With this CPU 01x, the CPU electronic and power supply are integrated to one casing. As head module, via the integrated power module for power supply, CPU electronic and the electronic of the connected periphery modules are supplied. The DC 24 power section supply for the linked periphery modules is 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!

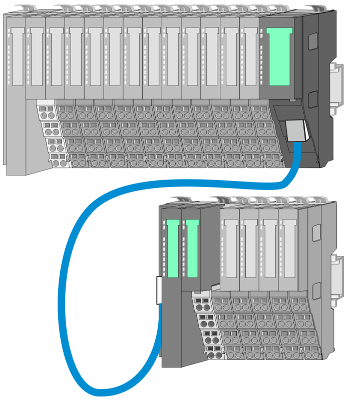
Bus coupler



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 24 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!

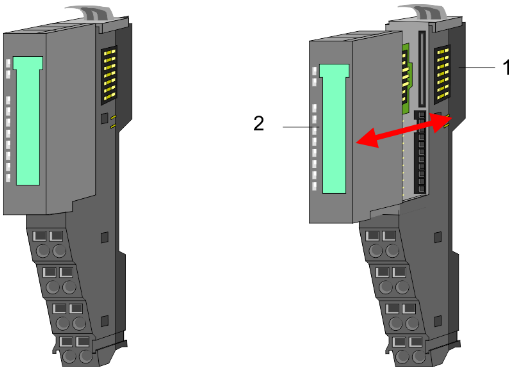
Line extension



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.

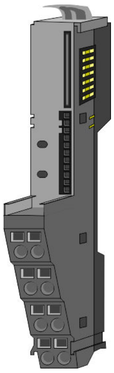
Periphery modules

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



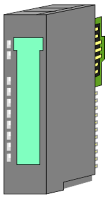
1 Terminal module
2 Electronic module

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 SLIO system may be assembled outside of your switchgear cabinet to be later mounted there as whole system.

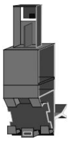
Electronic module



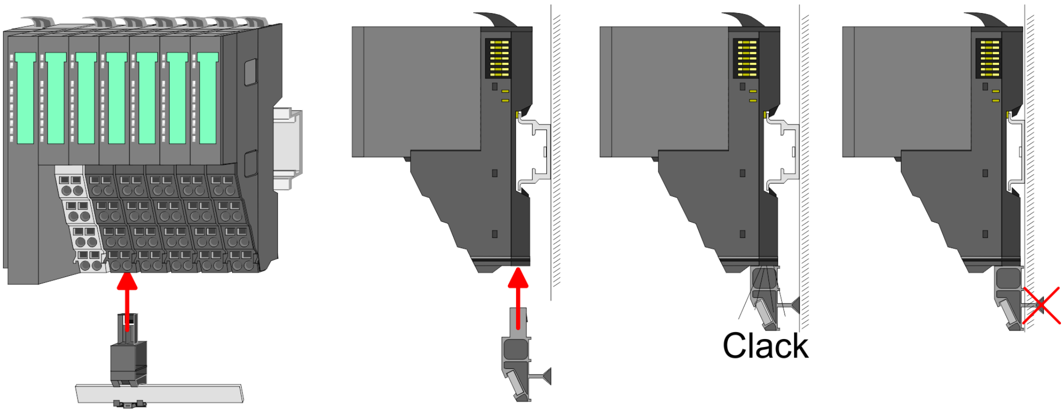
The functionality of a SLIO periphery module is defined by the *electronic* module, which is mounted to the terminal module by a sliding mechanism. With an error the defective module may be exchanged for a functional module with standing installation. At the front side there are LEDs for status indication.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

2.2.3 Accessories

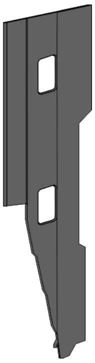
Shield bus carrier



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



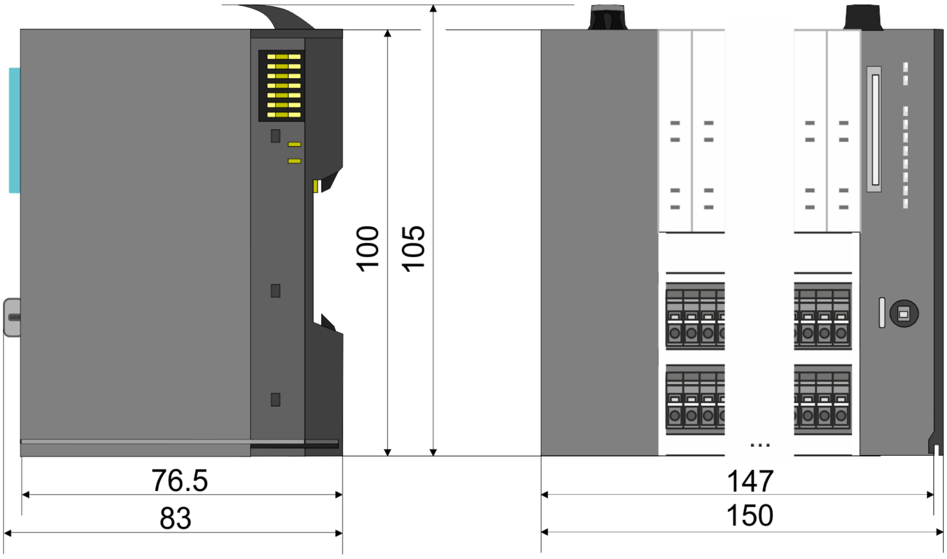
With each head module, to protect the backplane bus connectors, there is a mounted bus cover in the scope of delivery. You have to remove the bus cover of the head module before mounting a System SLIO module. For the protection of the backplane bus connector you always have to mount the bus cover at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

Coding pins

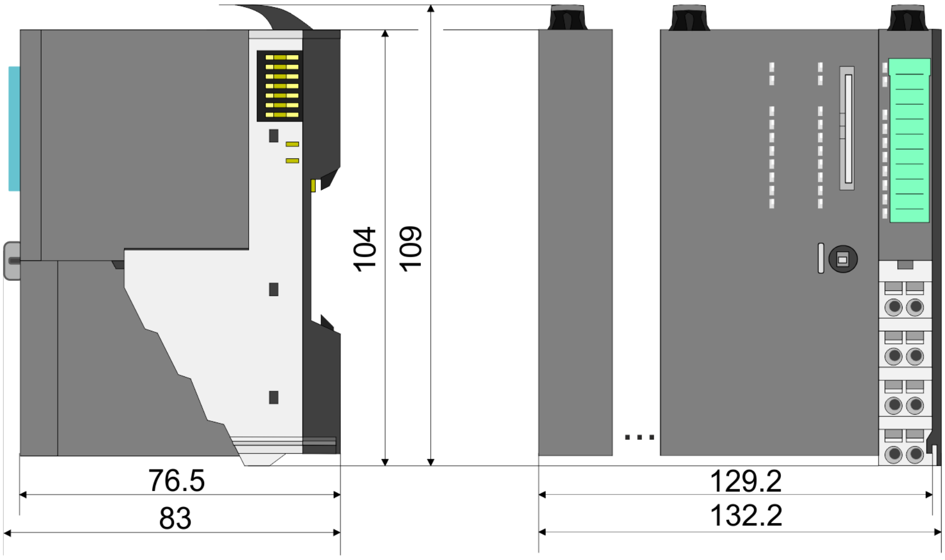
There is the possibility to fix the assignment of electronic and terminal module. Here coding pins (order number 000-0AC00) from VIPA can be used. The coding pin consists of a coding jack and a coding plug. By combining electronic and terminal module with coding pin, the coding jack remains in the electronic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

2.3 Dimensions

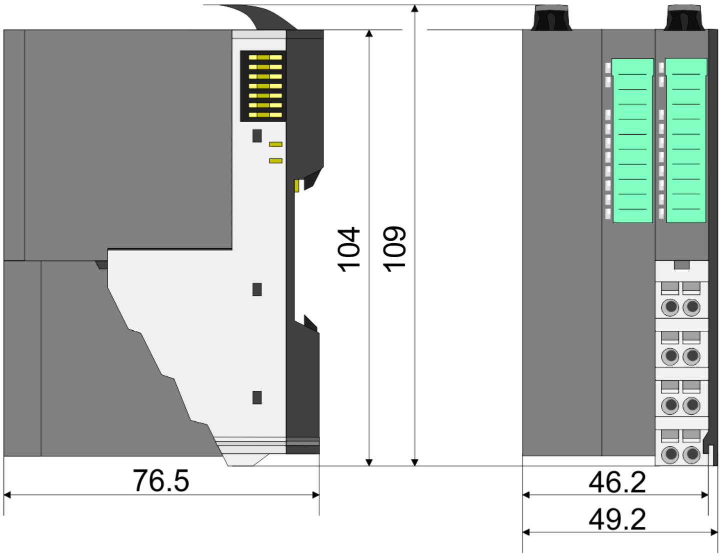
Dimensions CPU 01xC



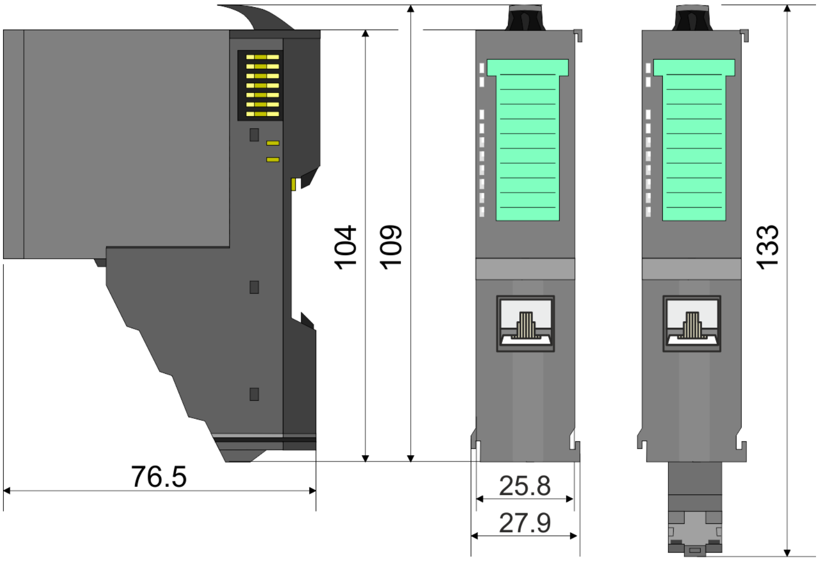
Dimensions CPU 01x



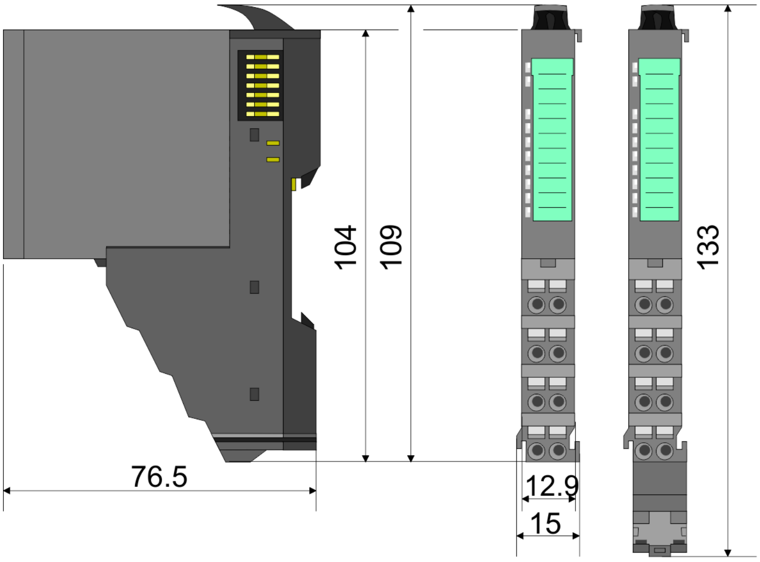
Dimensions bus coupler and line extension slave



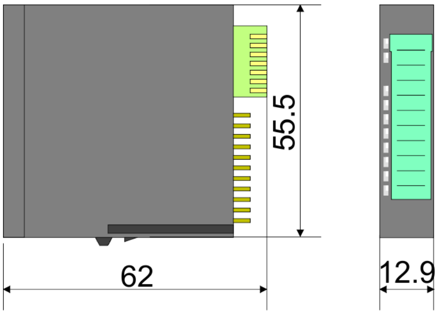
Dimensions line extension master



Dimension periphery module



Dimensions electronic module



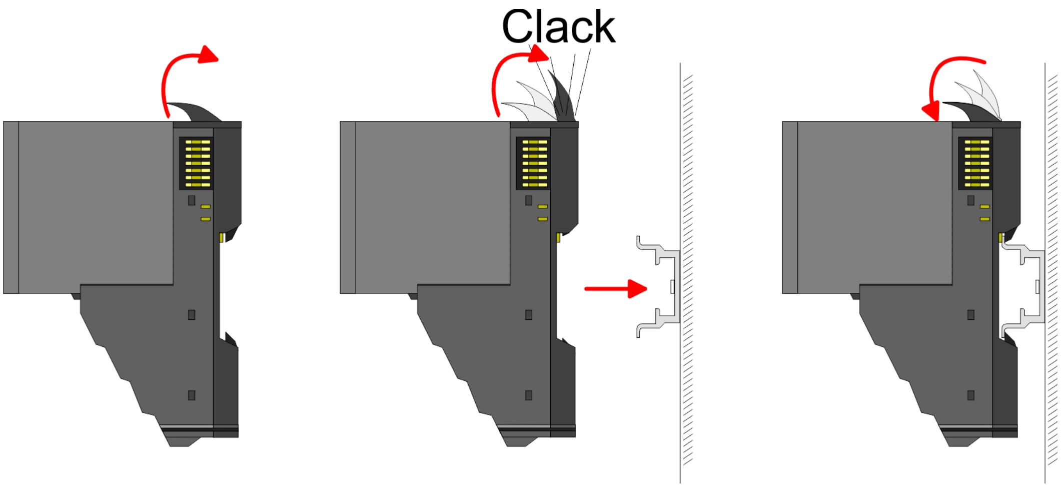
Dimensions in mm

2.4 Mounting periphery modules

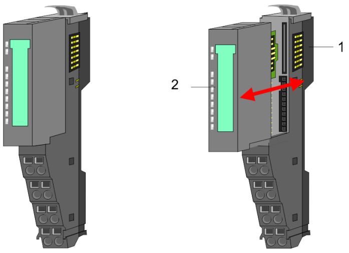
Requirements for UL compliance use

- *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 is a locking lever at the top side of the module. For mounting and demounting this locking lever is to be turned upwards until this engages. For mounting place the module to the module installed before and push the module to the mounting rail guided by the strips at the upper and lower side of the module. The module is fixed to the mounting rail by pushing downward the locking lever. The modules may either separately be mounted to the mounting rail or as block. Here is to be considered that each locking lever is opened. The modules are each installed on a mounting rail. The electronic and power section supply are connected via the backplane bus. Up to 64 modules may be mounted. 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.



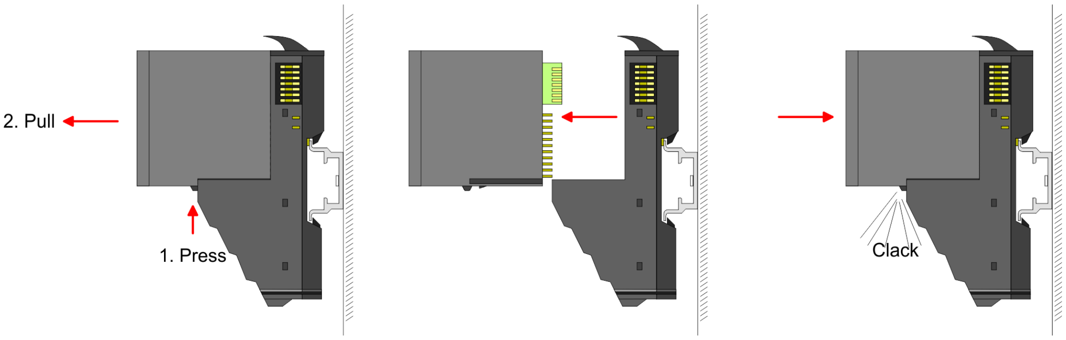
Terminal and electronic module



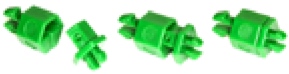
Each periphery module consists of a *terminal* and an *electronic module*.

1 Terminal module
2 Electronic module

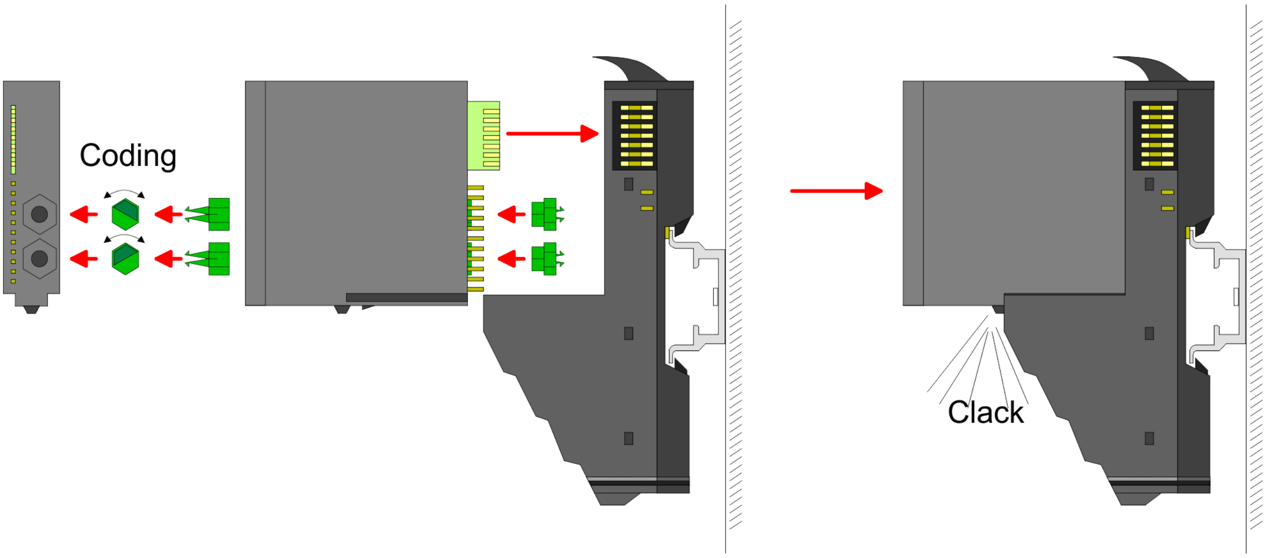
For the exchange of a electronic module, the electronic module may be pulled forward after pressing the unlocking lever at the lower side of the module. For installation plug the electronic module guided by the strips at the lower side until this engages audible to the terminal module.



Coding



There is the possibility to fix the assignment of electronic and terminal module. Here coding pins (order number 000-0AC00) from VIPA 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 electronics module just another electronic module can be plugged with the same encoding.



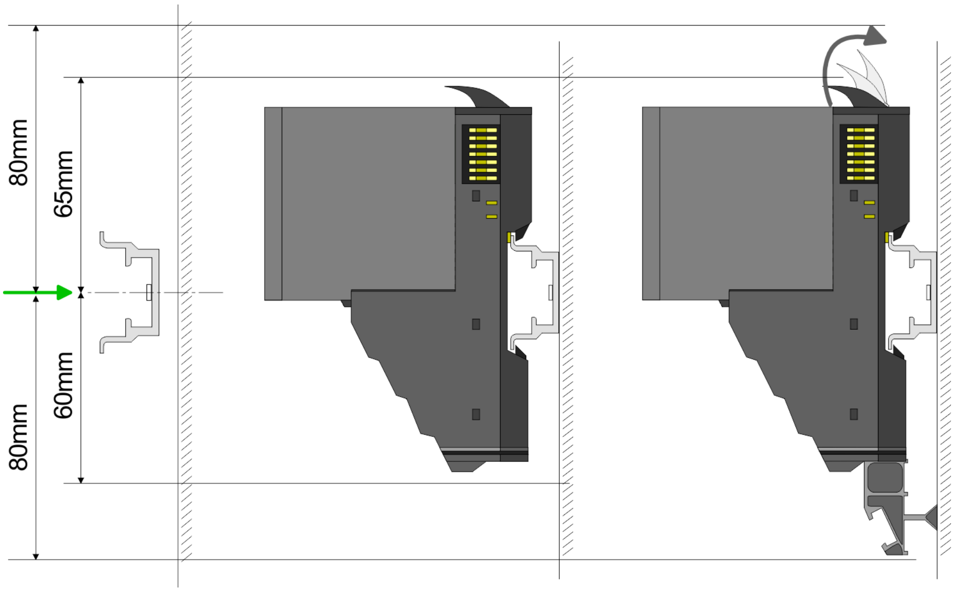
Each electronic module has on its back 2 coding sockets for coding jacks. Due to the characteristics, with the coding jack 6 different positions can be plugged, each. Thus there are 36 possible combinations for coding with the use of both coding sockets.

1. Plug, according to your coding, 2 coding jacks in the coding sockets of your electronic module until they lock
2. Now plug the according coding plugs into the coding jacks.
3. To fix the coding put both the electronic and terminal module together until they lock

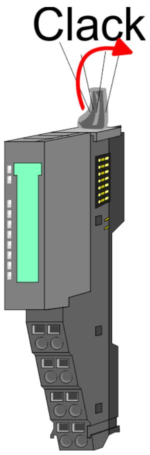
CAUTION!
Please consider that when replacing an already coded electronic module, this is always be replaced by an electronic module with the same coding.

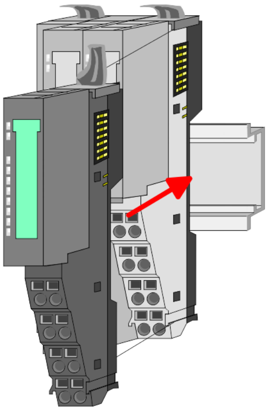
Even with an existing coding on the terminal module, you can plug an electronic module without coding. The user is responsible for the correct usage of the coding pins. VIPA assumes no liability for incorrectly attached electronic modules or for damages which arise due to incorrect coding!

Mounting periphery modules

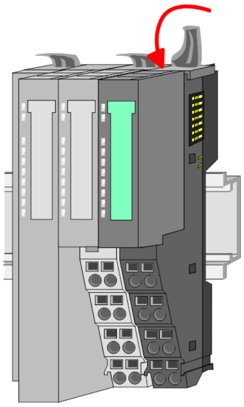


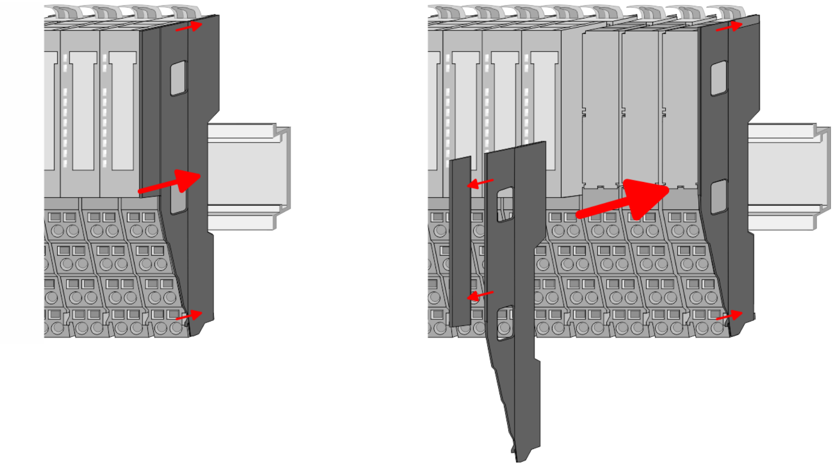
1. Mount the mounting rail! Please consider that a clearance from the middle of the mounting rail of at least 80mm above and 60mm below, respectively 80mm by deployment of shield bus carriers, exist.
2. Mount your head module such as CPU or field bus coupler.
3. Before mounting the periphery modules you have to remove the bus cover at the right side of the Head module by pulling it forward. Keep the cover for later mounting.





4. For mounting turn the locking lever of the module upward until it engages.
5. For mounting place the module to the module installed before and push the module to the mounting rail guided by the strips at the upper and lower side of the module.
6. Turn the locking lever of the periphery module downward, again.





7. 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 periphery modules

Terminal module terminals

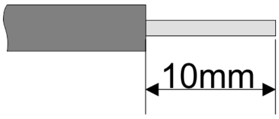
CAUTION!

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!

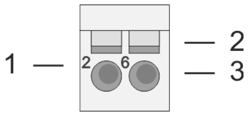
With wiring the terminal modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

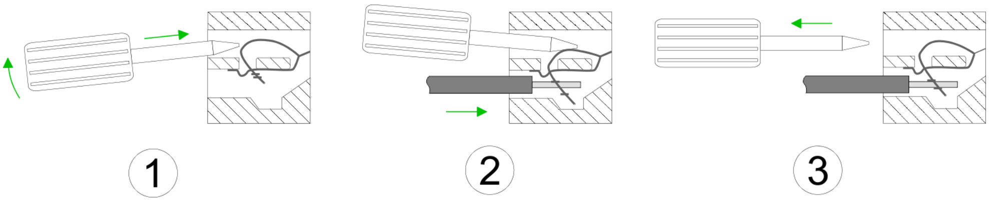


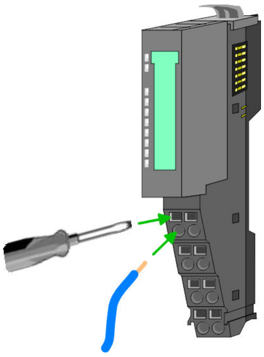
U_{max}	240V AC / 30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm^2 (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



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



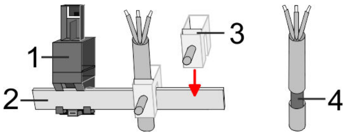


1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.

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

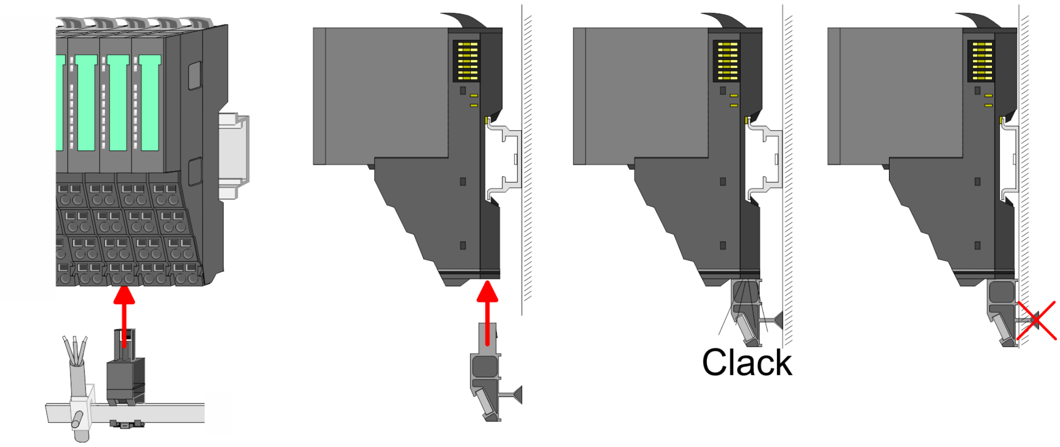


1	Shield bus carrier
2	Shield bus (10mm x 3mm)
3	Shield clamp
4	Cable shield

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 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.



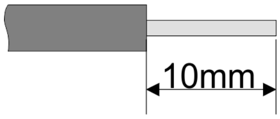
3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

2.6 Wiring power modules

Terminal module terminals

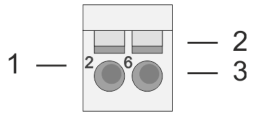
Power modules are either integrated to the head module or may be installed between the periphery modules. With power modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw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

Data

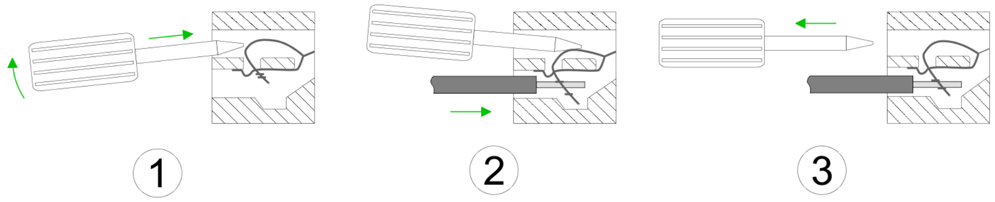


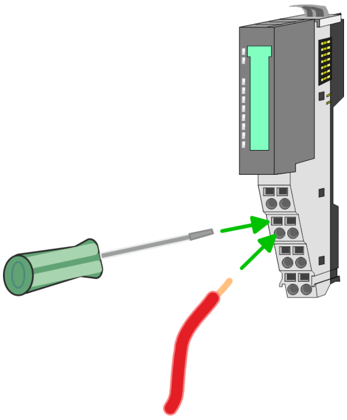
U_{max}	30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm^2 (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



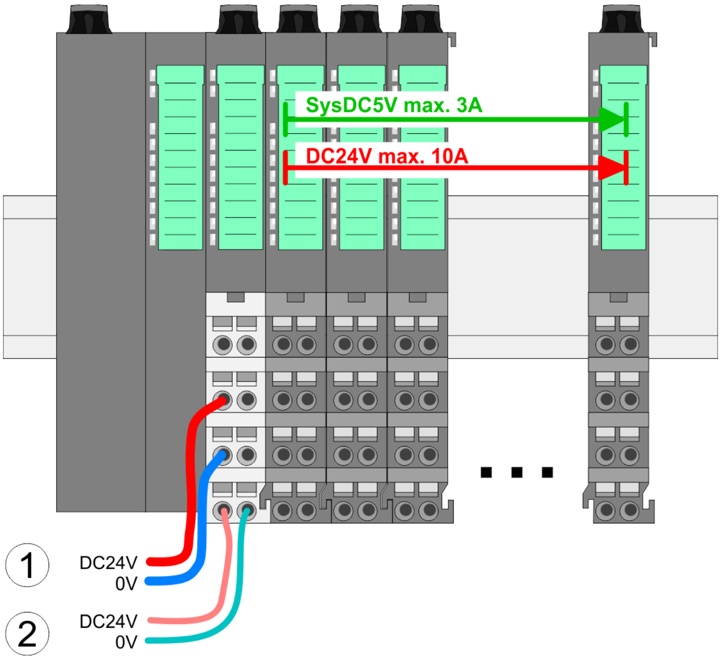
1 Pin number at the connector
2 Opening for screwdriver
3 Connection hole for wire





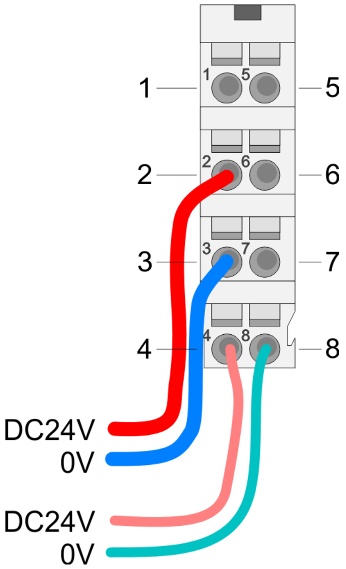
1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm^2 up to 1.5mm^2
3. By removing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

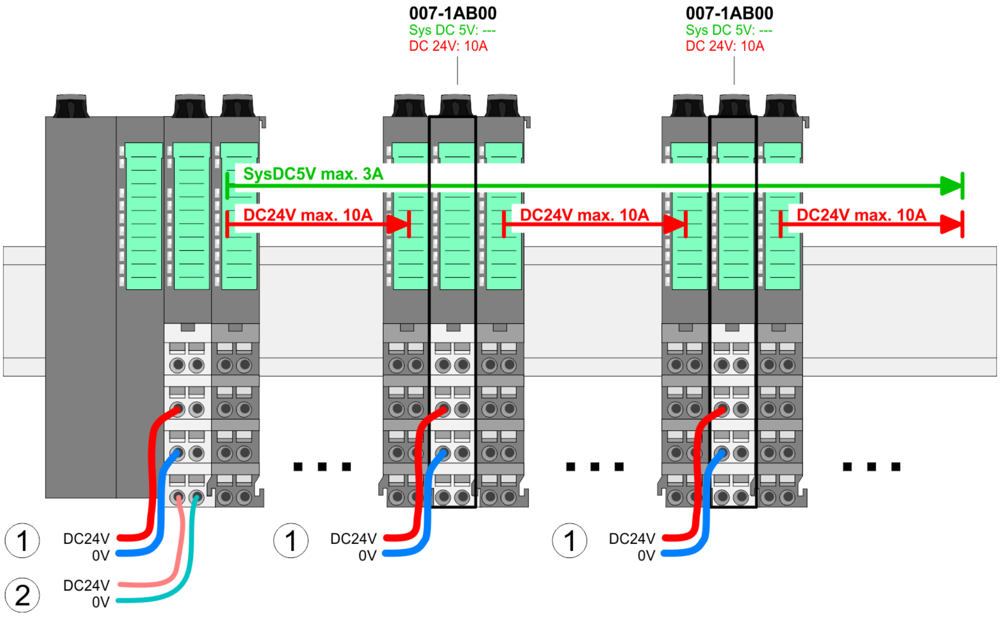
State of the electronic power supply via LEDs

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.

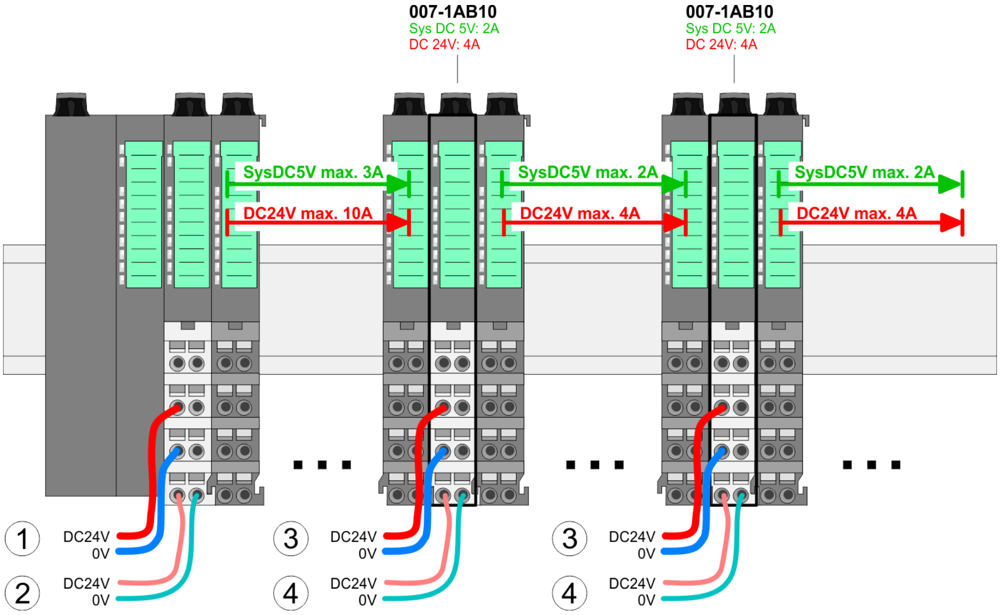
Deployment of the power modules

- If the 10A for the power section supply is no longer sufficient, you may use the power module from VIPA 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

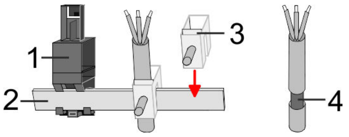


Power module 007-1AB10



(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

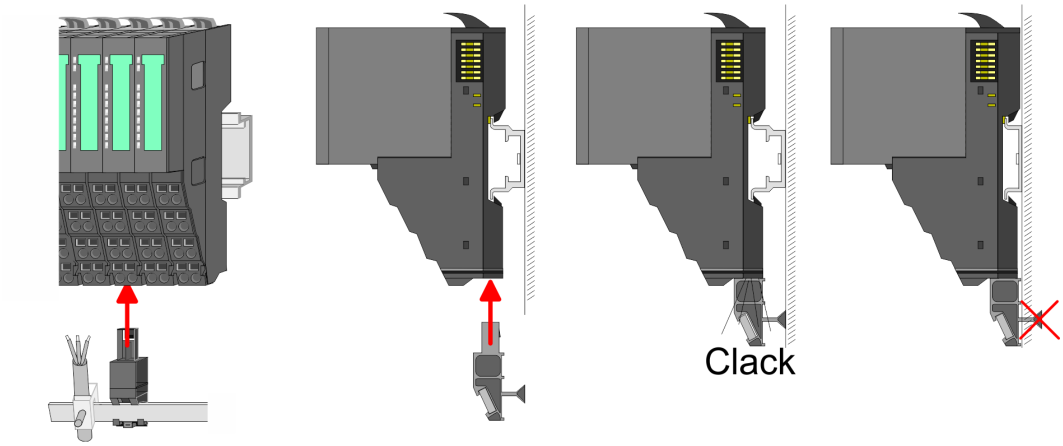


1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

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 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.



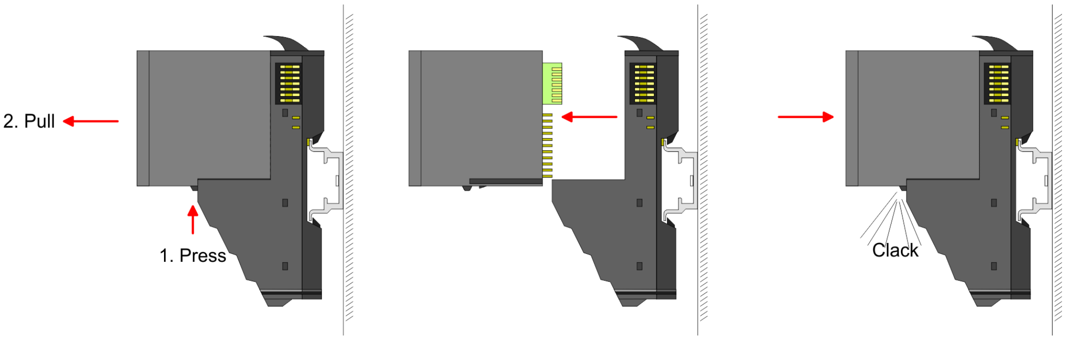
3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

2.7 Demounting periphery modules

Proceeding

Exchange of an electronic module

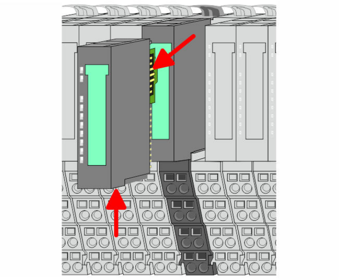
1. Power-off your system.



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.

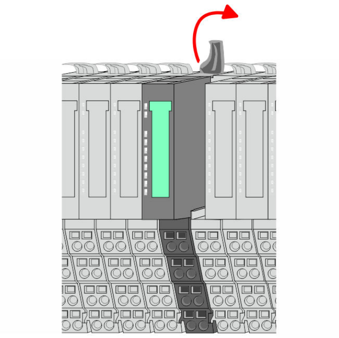
   ⇨ Now you can bring your system back into operation.

Exchange of a periphery module

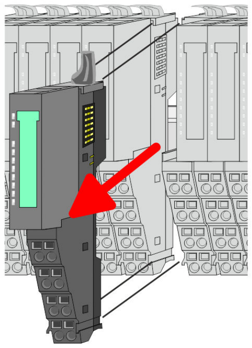


1. Power-off your system.
2. Remove if exists the wiring of the module.
3. *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

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

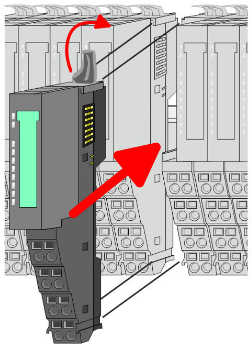


4. Turn the locking lever of the module to be exchanged upwards.



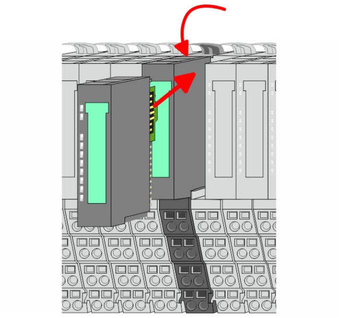
5. Pull the module.

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.

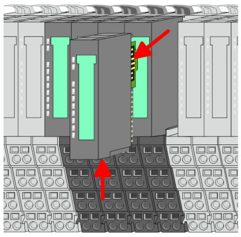
Exchange of a module group

1. Power-off your system.

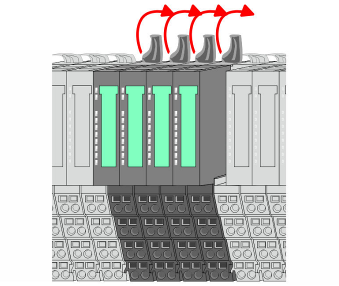
2. Remove if exists the wiring of the module group.

3.

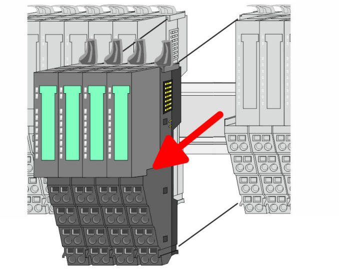
> *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*



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

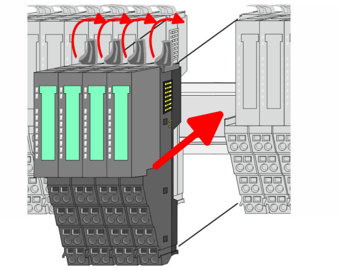


4. Turn all the locking lever of the module group to be exchanged upwards.



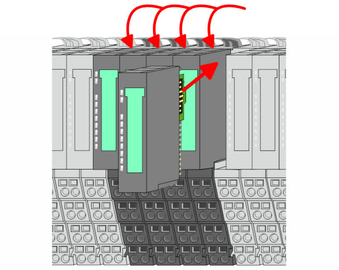
5. Pull the module group forward.

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.

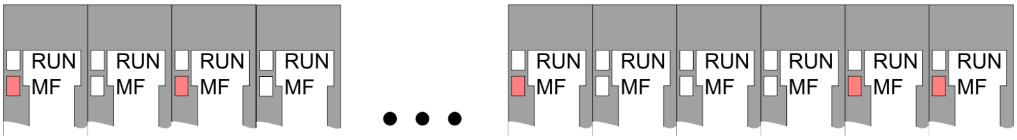
2.8 Trouble shooting - LEDs

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

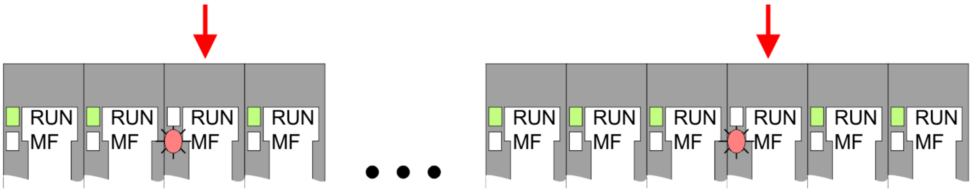


Behaviour: After PowerON the RUN LED of each module is off and the MF LED of each module is sporadically on.

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. ↳ *Chapter 2.6 'Wiring power modules' on page 21*

Error in configuration

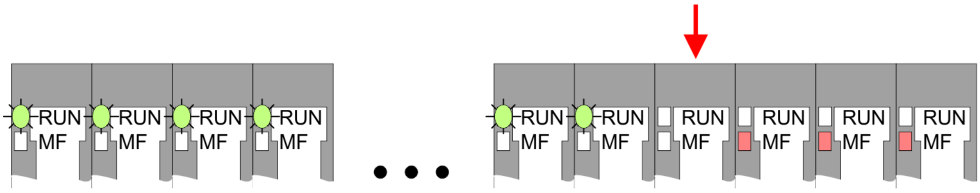


Behaviour: After PowerON the MF LED of one module respectively more modules flashes. The RUN LED remains off.

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.

2.9 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 components of VIPA 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

2.10 General data

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 *

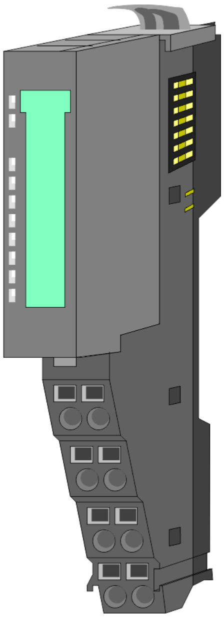
*) Due to the high-energetic single pulses with Surge an appropriate external protective circuit with lightning protection elements like conductors for lightning and overvoltage is necessary.

3 Hardware description

3.1 Properties

Features

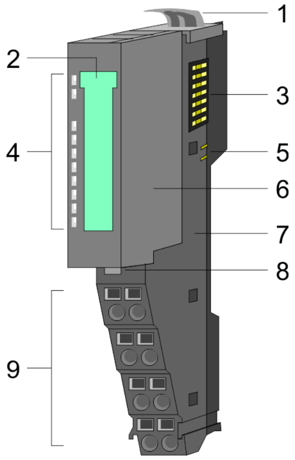
- RS422/485 interface
  (isolated to back plane bus)
- Transfer rate 150bit/s up to 115.2kbit/s
- Serial bus connection
  - full-duplex (RS422 four-wire operation)
  - half-duplex (RS485 two-wire operation)
- Protocols
  - ASCII
  - STX/ETX
  - 3964(R)
  - Modbus (master/slave with ASCII and RTU short & long) with a telegram length of 250byte
- Up to 250 telegrams (1024byte receive and send buffer)
- Character delay time ZVZ parameterizable in ms steps
- Configured by parameter data



Order data

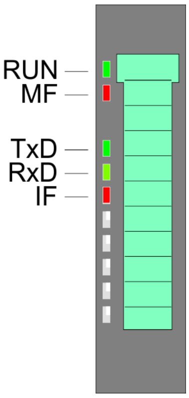
Type	Order number	Description
CP 040 RS422/485	040-1CA00	Communication processor, RS422/485, isolated, ASCII, STX/ETX, 3964(R), Modbus master/slave short/long

3.2 Structure



1 Locking lever terminal module
2 Labeling strip
3 Backplane bus
4 LED status indication
5 DC 24V power section supply
6 Electronic module
7 Terminal module
8 Locking lever electronic module
9 Terminal

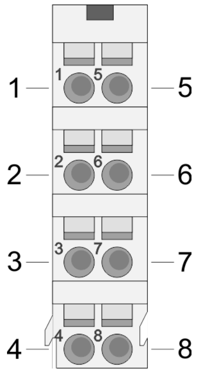
Status indication

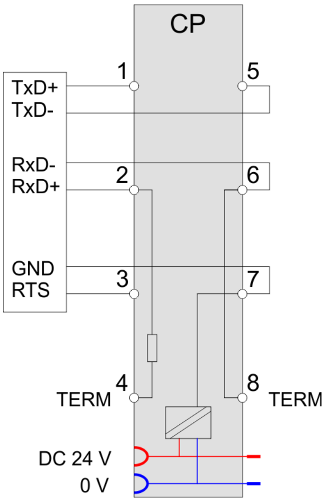


LED		Description
RUN 🟩 green	MF 🟥 red	
🟩	□	Bus communication is OK Module status is OK
🟩	🟥	Bus communication is OK Module status reports an error
□	🟥	Bus communication is not possible Module status reports an error
□	□	Error at bus power supply
X	◩ 2Hz	Error in configuration ☞ *Chapter 2.8 ‘Trouble shooting - LEDs’ on page 28*
TxD	🟩 green	Transmit data
RxD	🟩 green	Receive data
IF	◩ 2Hz	Modbus: internal error Other protocols: error indicator for open circuit lines, overflow, parity or framing errors
not relevant: X		

Terminal

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





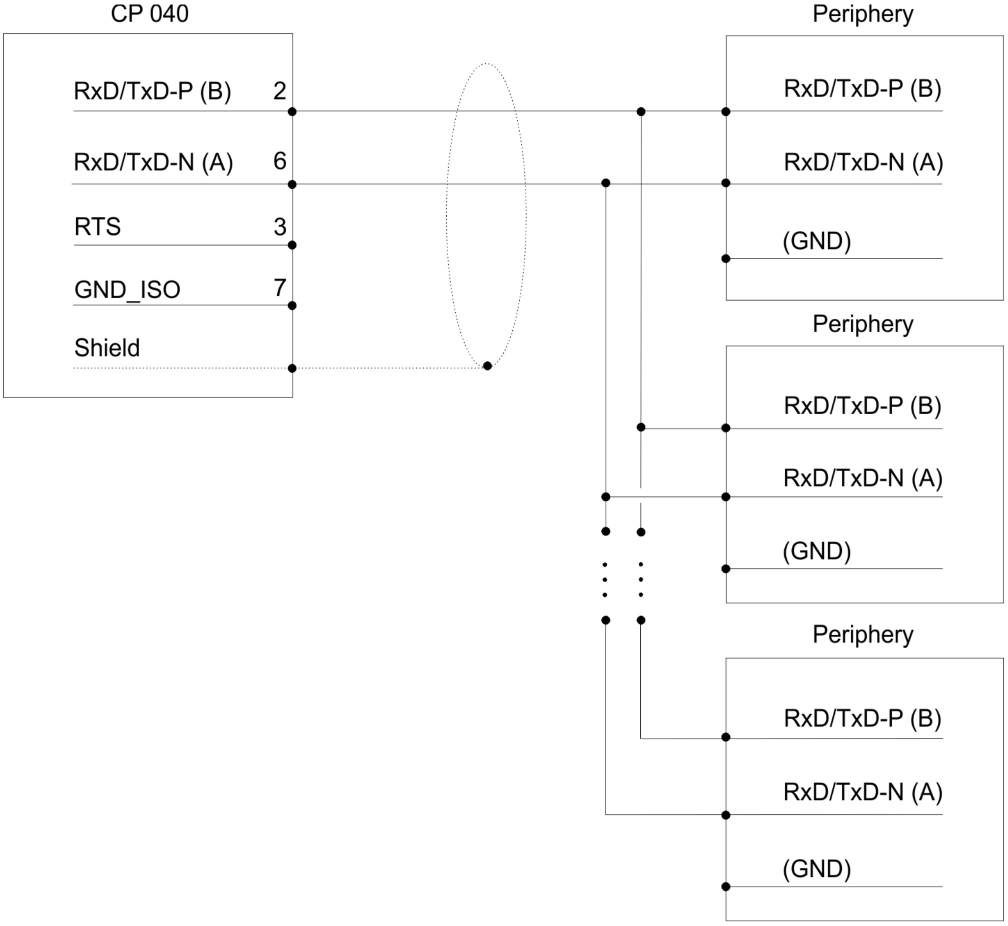
Pos.	Function	Type	Description
1	TxD-P (B)	O	Send data (RS422)
2	RxD-P (B) TxD/RxD-P (B)	I O/I	Receive data (RS422) Send-/Receive data (RS485)
3	RTS	O	Request to send (RS485) RTS at logic "1": CP ready to send RTS at logic "0": CP is not sending
4	TERM	I	Terminating resistor *
5	TxD-N (A)	O	Send data (RS422)
6	RxD-N (A) TxD/RxD-N (A)	I O/I	Receive data (RS422) Send-/Receive data (RS485)
7	GND_ISO	O	Signal ground (isolated)
8	TERM	I	Terminating resistor *
*) A bridge between the two TERM inputs activates a terminal resistance of 120Ω on the receiver side between RxD-P (Pin 2) and RxD-N (Pin 6).			

I: Input, O: Output

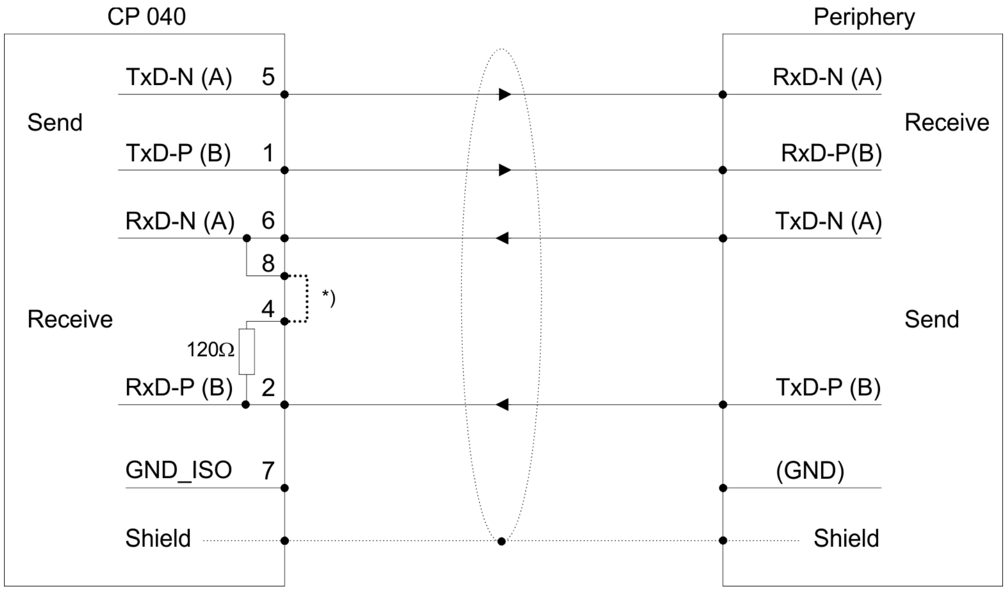
RS422/485

- Logical conditions as voltage difference between 2 twisted lines
- Serial bus connection
  - full-duplex (RS422 four-wire operation)
  - half-duplex (RS485 two-wire operation)
- Line length: 250m at 115.2kbit/s ... 1200m at 19.2kbit/s
- Data transfer rate up to 115.2kbit/s

RS485 connection



RS422 connection

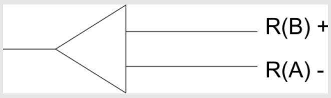
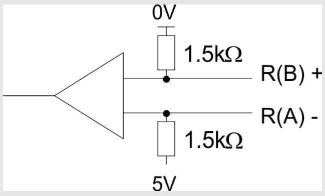
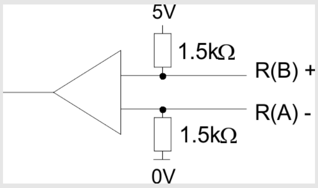


**) A bridge between the two TERM inputs activates a terminal resistance of 120Ω on the receiver side between RxD-P (Pin 2) and RxD-N (Pin 6).*

Defined static voltage levels by parameters

For a connection with minimum reflections and the wire-break recognition at RS422/485 operation, the lines may be preset with defined static voltage levels.

At the CP interface the wiring of the receiver is realized as follows:

Parameter	Description	Wiring of the receiver
None (00h)	No preassignment of the receiving lines. This setting only makes sense with bus-capable special drivers.	
Signal R(A) 5V (Break evaluation) Signal R(B) 0V (01h)	With this preassignment break detection is possible at fullduplex operation (RS422).	
Signal R(A) 0V Signal R(B) 5V (02h)	This preassignment corresponds to the idle state (no sender is activated) at halfduplex operation at RS485. Here wire-break recognition is not possible.	

3.3 Technical data

Order no.	**040-1CA00**
Type	CP 040 RS422/485
Module ID	0E41 1700
Current consumption/power loss	
Current consumption from backplane bus	125 mA
Current consumption from load voltage L+ (without load)	10 mA
Power loss	1 W
Status information, alarms, diagnostics	
Status display	yes
Interrupts	yes, parameterizable
Process alarm	no
Diagnostic interrupt	yes, parameterizable
Diagnostic functions	yes, parameterizable
Diagnostics information read-out	possible
Supply voltage display	green LED
Group error display	red LED
Channel error display	red LED
Point-to-point communication	
PtP communication	✓
Interface isolated	✓
RS232 interface	-
RS422 interface	✓
RS485 interface	✓
Connector	Terminal module
Transmission speed, min.	150 bit/s
Transmission speed, max.	115.2 kbit/s
Cable length, max.	1200 m
Point-to-point protocol	
ASCII protocol	✓
STX/ETX protocol	✓
3964(R) protocol	✓
RK512 protocol	-
USS master protocol	-
Modbus master protocol	✓
Modbus slave protocol	✓
Special protocols	-

Order no.	040-1CA00
Datasizes	
Input bytes	8 / 20 / 60
Output bytes	8 / 20 / 60
Parameter bytes	23
Diagnostic bytes	20
Housing	
Material	PPE / PPE GF10
Mounting	Profile rail 35 mm
Mechanical data	
Dimensions (WxHxD)	12.9 mm x 109 mm x 76.5 mm
Net weight	59 g
Weight including accessories	59 g
Gross weight	74 g
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL certification	yes
KC certification	yes

3.3.1 Technical data protocols

ASCII	
Telegram length	max. 1024 byte
Baud rate	150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 76800, 109700, 115200 Baud
Character delay time ZVZ	0 ... 65535 in ms steps (with 0 triple character time is used)
Flow control	none, hardware, XON/XOFF
Number of telegrams to buffer	max. 250
End recognition of a telegram	after character delay time ZVZ
STX/ETX	
Telegram length	max. 1024 byte
Baud rate	150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 76800, 109700, 115200 Baud

Character delay time TMO	0 ... 65535 in ms steps (with 0 triple character time is used)
Flow control	none, hardware, XON/XOFF
Number of telegrams to buffer	max. 250
End recognition of a telegram	by parameterized end character
Number of start characters	0 ... 2 (characters parameterizable)
Number of end characters	0 ... 2 (characters parameterizable)
3964, 3964R	
Telegram length	max. 1024 byte
Baud rate	150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 76800, 109700, 115200 Baud
Block proof sign	only 3964R
Priority	low/high
Character delay time ZVZ	0 ... 255 in 20ms steps (with 0 triple character time is used)
Acknowledgment delay time QVZ	0 ... 255 in 20ms steps (with 0 triple character time is used)
Number of connection attempts	0 ... 255
Number of transfer attempts	1 ... 255
Modbus	
Telegram length	max. 258 byte
Addressable range	each 1024 byte
Baud rate	150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 76800, 109700, 115200 Baud
Mode	Master ASCII, Master RTU, Slave ASCII short, Slave RTU short, Slave ASCII long, Slave, RTU long
Address	1 ... 255
Delay time	automatically, 1 ... 60000 ms

4 Deployment

4.1 Fast introduction

Overview

The communication processor 040-1CA00 enables the serial process connection to different destination or source systems. Here the CP is used as peripheral module and power supplied by the back plane bus.

Parameter

For the parameterization you may send parameter data to the CP that are differently assigned depending on the chosen protocol. More about the parameter assignment may be found in Chapter "Serial communication protocols". ↳ *Chapter 5.1 'Overview' on page 57*

Protocols

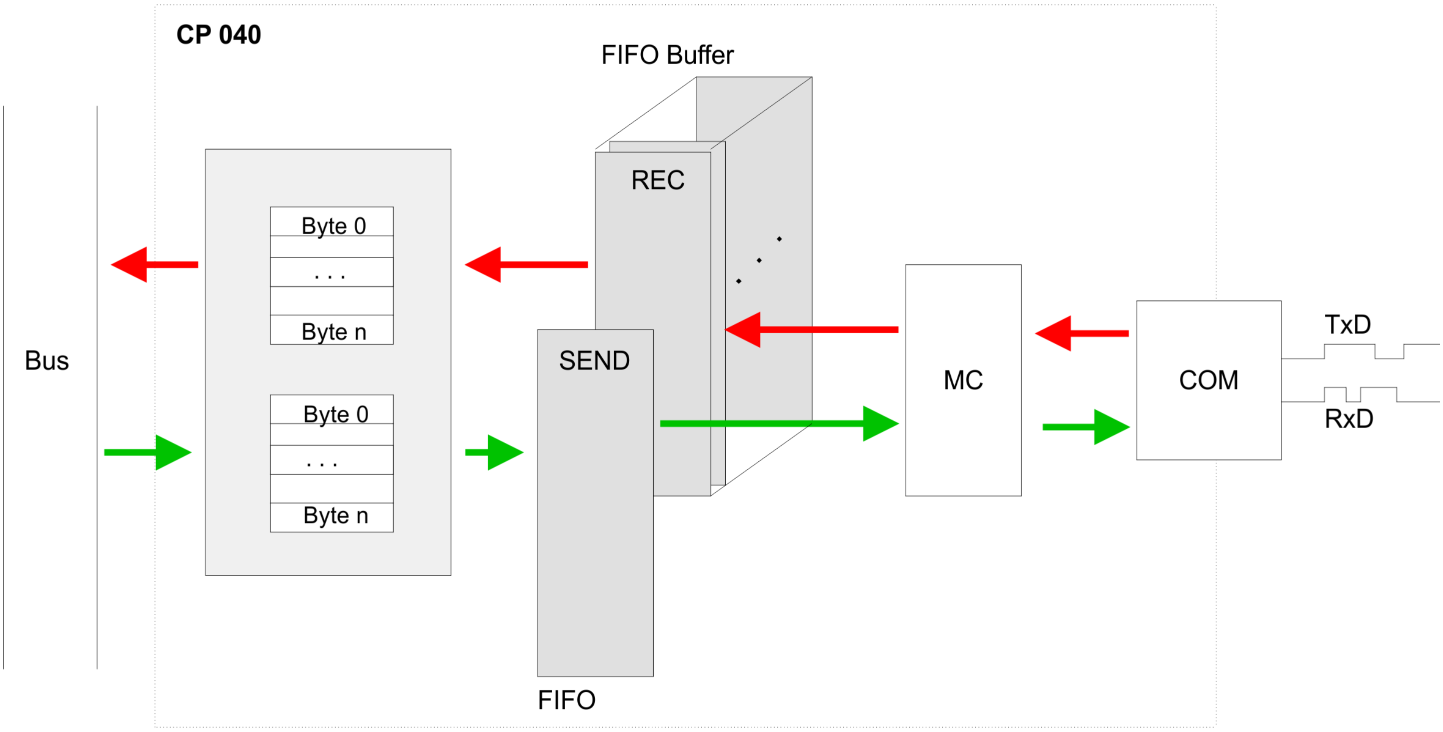
- ASCII
- STX/ETX
- 3964(R)
- Modbus (master, slave)

Communication

When you send data, which are written by a host system via the back plane bus to the corresponding output area, to the send buffer, these are sent by the interface.

If the communication processor receives data from its interface, the data are stored in a circular buffer and transmitted via the back plane bus to the input area of the host system.

Please consider that the size of the I/O area and thus also of the telegram at the back plane bus depends on the host system. On the following pages the IO area and the communication via the back plane bus are more described.



4.2 In-/Output area

Overview

Depending on the host system the CP uses for each input and output the following number of bytes in the address area.

-
- PROFIBUS: 8byte, 20byte or 60byte selectable
- PROFINET: 20byte or 60byte selectable
- CANopen: 8byte
- EtherCAT: 60byte
- DeviceNET: 60byte
- ModbusTCP: 60byte

At CPU, PROFIBUS and PROFINET the input respectively output area is embedded to the corresponding address area.

IX - Index for access via CANopen. With s = Subindex the corresponding byte is addressed.

SX - Subindex for access via EtherCAT with Index 6000h/7000h + EtherCAT-Slot

More can be found in the according manual of your bus coupler.

Input area

Addr.	Name	Bytes	Function	IX = 5450h	SX
+0	CP_IN_STS	1	Status byte	s = 1	01h
+1	CP_IN_1	1	Input byte 1	s = 2	02h
+2	CP_IN_2	1	Input byte 2	s = 3	03h
...	...	...	...	...	...
+n-1	CP_IN_n-1	1	Input byte n-1	s = m	mh

CP_IN_STS

This parameter contains information about the fragmentation of the data in the receive buffer.

CP_IN_x

The content of these data depends on the structure of the data in the receive buffer. For more information, see the following pages.

Output area

Addr.	Name	Bytes	Function	IX = 5650h	SX
+0	CP_OUT_CTRL	1	Control byte	s = 1	01h
+1	CP_OUT_1	1	Output byte 1	s = 2	02h
+2	CP_OUT_2	1	Output byte 2	s = 3	03h
...	...	...	...	...	...
+n-1	CP_OUT_n-1	1	Output byte n-1	s = m	mh

CP_OUT_CTRL

Here you can control the data transfer by means of appropriate commands.

CP_OUT_x

The content of these data depends on the structure of data in the send buffer. For more information, see the following pages.

4.3 Principal communication via back plane bus

4.3.1 Sending data

When sending from the host, the output data are entered in the output area and by means of the *Control-Byte* transferred to the CP.

The CP responds every telegram with an acknowledgement, by copying bit 3...0 of byte 0 of the output area to bit 7...4 of byte 0 of the input area or sending back a *status message* via this byte.

Depending on the length of data the telegram is to be transferred to the CP as one fragment or with multiple fragments. With the fragmented transmission, each fragment is acknowledged by the CP.

Principle of the communication *without fragmentation*

Host system			CP	
Byte	**Function**		**Byte**	**Function**
0	Control-Byte	▶		
1	Telegram-Info-Byte			
2	Length High-Byte			
3	Length Low-Byte			
4...n-1	User data Byte 0...n-5			
		◀	0	Acknowledgement / Status
with n = number of used bytes in the address area (IO-Size)				

Control-Byte

Bit 3...0	■ 8h: Idle state - no data available ■ Ah: Start transfer without fragmentation ■ Bh: Execute a reset on the CP
Bit 7...4	Reserved for receipt

Telegram-Info-Byte

00h (fix) when data are sent.

Length

Length of user data for serial communication in byte.

User data

Enter here the user data for the serial communication.

Acknowledgement Status

Bit 3...0	Reserved for receipt
Bit 7...4	■ 8h: Acknowledgement: Idle state ■ Ah: Acknowledgement: Data received without fragmentation ■ Ch: Status: Reset was executed on the CP ■ Dh: Status: The entered length is not valid ■ Eh: Status: Error in CP communication – there is no response of the other station

Principle of communication *with fragmentation*

With the fragmented communication the number of user data and a part of the user data are already transferred with the 1. telegram (header), followed by the fragment telegrams. The CP responds every telegram with an acknowledgement, by copying bit 3...0 of byte 0 of the output area to bit 7...4 of byte 0 of the input area or sending back a *status message* via this byte.

Sequence

- Write 1. telegram
- Write fragments
- Write last fragment

Calculating the number of fragments

$$Number\ Fragments = \frac{Length+3}{IO\_Size-1}$$

Write 1. telegram (Header)

Host system			CP	
Byte	**Function**		**Byte**	**Function**
0	Control-Byte	▶		
1	Telegram-Info-Byte			
2	Length high byte			
3	Length low byte			
4...n-1	User data byte 0 ...n-5			
		◀	0	Acknowledgement / Status

with n = number of used bytes in the address area (IO-Size)

Control-Byte

Bit 3...0	■ 8h: Idle state - no data available ■ 9h: Start transfer with fragmentation ■ Ah: Transfer last fragment ■ Bh: Execute a reset on the CP
Bit 7...4	Reserved for receipt

Telegram-Info-Byte

00h (fix) when data are sent.

Length

Length of user data for serial communication in byte.

User data

Enter here the user data for the serial communication.

Acknowledgement Status

Bit 3...0	Reserved for receipt
Bit 7...4	■ 8h: Acknowledgement: Idle state ■ 9h: Acknowledgement: Fragmented transfer started ■ Ah: Acknowledgement: Data received without fragmentation ■ Ch: Status: Reset was executed on the CP ■ Dh: Status: The entered length is not valid ■ Eh: Status: Error in CP communication – there is no response of the other station

Write fragments

Host system			CP	
Byte	**Function**		**Byte**	**Function**
0	Control-Byte	▶		
1...n-1	User data			
		◀	0	Acknowledgement / Status

with n = number of used bytes in the address area (IO-Size)

Control-Byte

Bit 3...0	■ 0h...7h: Fragment number ■ 8h: Idle state - no data available ■ Bh: Execute a reset on the CP
Bit 7...4	Reserved for receipt

User data

Enter here the user data for the serial communication.

Acknowledgement Status

Bit 3...0	Reserved for receipt
Bit 7...4	■ 0h...7h: Acknowledgement: Fragment number ■ 8h: Acknowledgement: Idle state ■ Ch: Status: Reset was executed on the CP ■ Dh: Status: The entered length is not valid ■ Eh: Status: Error in CP communication – there is no response of the other station

Write last fragment

Host system			CP	
Byte	**Function**		**Byte**	**Function**
0	Control-Byte	▶		
1...n-1	User data			

Host system			CP	
Byte	Function		Byte	Function
		◀	0	Acknowledgement / Status
with n = number of used bytes in the address area (IO-Size)				

Control-Byte

Bit 3...0	■ 8h: Idle state - no data available ■ Ah: Transfer last fragment ■ Bh: Execute a reset on the CP
Bit 7...4	Reserved for receipt

User data

Enter here the user data for the serial communication.

Acknowledgement Status

Bit 3...0	Reserved for receipt
Bit 7...4	■ 8h: Acknowledgement: Idle state ■ Ah: Acknowledgement: Last fragment received ■ Ch: Status: Reset was executed on the CP ■ Dh: Status: The entered length is not valid ■ Eh: Status: Error in CP communication – there is no response of the other station

4.3.2 Receiving data

When receiving data from the CP, the data are automatically transferred to the input area of the host system.

Depending on the length of the received data, the telegram is transferred to the host system as one fragment or with multiple fragments.

The fragmented transfer is started by copying bit 3 ... 0 of byte 0 of the input area to bit 7 ... 4 of byte 0 of the output area. Possible errors during the transfer may be found in RetVal.

Principle of communication *without fragmentation*

Host system			CP	
Byte	Function		Byte	Function
		▶	0	Info-Byte
			1	Telegram-Info-Byte
			2	Length high byte
			3	Length low byte
			[4]	Offset high byte
			[5]	Offset low byte
			6	RetVal high byte
			7	RetVal low byte

Host system			CP	
Byte	Function		Byte	Function
			8...n-1	User data
0	Acknowledgement	◀	0	
with n = number of used bytes in the address area (IO-Size)				

Info-Byte

Bit 3...0	■ 8h: Idle state - no data available ■ 9h: Data are transferred with fragmentation ■ Ah: Data are transferred without fragmentation
Bit 7...4	Reserved for sending

Telegram-Info-Byte

00h:	The telegram does not contain any additional offset information.
04h:	The telegram contains additional offset data, which are located as word after *Length*. With this offset the position of the user data in the input area is defined.

Length

Length of user data for serial communication in byte plus 2bytes for RetVal.

Length 2byte: only RetVal without user data.

Offset

If the *Telegram-Info-Byte* is 04h, an additional offset is entered. Otherwise there is no *Offset* in the telegram.

RetVal

0517h:	*Length* is not valid (*Length* = 0 or *Length* > 1024)
080Ah:	A free receive buffer is not available.
080Ch:	Character with error received (character frame or parity error)

User data

Here the received user data of the serial communication may be found.

Acknowledgement

After you have processed accordingly the data in your master system, you have to acknowledge the receipt to the CP (also RetVal telegrams without user data). Only then he can provide new received data.

Bit 3...0	Reserved for sending
Bit 7...4	8h: Acknowledgement: Idle state Ah: Acknowledgement: Input area free for new data Bh: Command: Execute a reset on the CP

Principle of communication *with fragmentation*

Host system			CP	
Byte	Function		Byte	Function
		◀	0	Info-Byte
			1	Telegram-Info-Byte
			2	Length high byte
			3	Length low byte
			[4]	Offset high byte
			[5]	Offset low byte
			6...n-1	User data
with n = number of used bytes in the address area (IO-Size)				

After the data are processed in the host system, you have to send an acknowledge to the CP, by copying bit 3...0 of byte 0 of the input area to bit 7...4 of byte 0 of the output area. Only then the CP can send further data.

0	Acknowledgement	▶	0	

Calculating the number of fragments

$$Number\ \ Fragments = \frac{Length+7}{IO\_Size-1}$$

Info-Byte

Bit 3...0	8h: Idle state - no data available 9h: Data were transferred with fragmentation Ah: Data were transferred without fragmentation
Bit 7...4	Reserved for sending

Telegram-Info-Byte

00h:	The telegram does not contain any additional offset information.
04h:	The telegram contains additional offset data, which are located as word after *Length*. With this offset the position of the user data in the input area is defined.

Length

Length of user data in byte plus 2 bytes for RetVal.

Offset

If the *Telegram-Info-Byte* is 04h, an additional offset is entered. Otherwise there is no Offset in the telegram. Calculating the Offset with fragmented transfer:

Data_Offset = (Fragment_counter + 1) × (IO_Size-1) -7 + Offset

- Data_Offset:
  - Offset of the data in the input area
- Fragment_counter:
  - Absolute number of fragments
- IO_Size:
  - Number of used bytes in the address area
- Offset:
  - Offset value in the telegram

User data

Here the received user data of the serial communication may be found.

Acknowledgement

Bit 3...0	Reserved for sending
Bit 7...4	■ 8h: Acknowledgement: Idle state ■ Ah: Acknowledgement: input area free for new data ■ Bh: Command: Execute a reset on the CP

4.3.3 Examples

Send data *without fragmentation*

IO-Size = 60byte, length = 40byte

Host system			CP	
Byte	**Function**		**Byte**	**Function**
0	0Ah Command	▶		
1	00h Telegram-Info			
2	00h Length high byte			
3	28h Length low byte			
4...43	User data byte 0...39			
44...59	is not used			
		◀	0	A0h Acknowledgement

Send data *with fragmentation*

IO-Size = 16byte, length = 50byte

Header Host system			CP	
Byte	**Function**		**Byte**	**Function**
0	09h Command	▶		
1	00h Telegram-Info			
2	00h Length high byte			
3	28h Length low byte			

Header Host system			CP	
Byte	**Function**		**Byte**	**Function**
4...15	User data byte 0...11			
		◀	0	90h Acknowledgement

1. Fragment Host system			CP	
Byte	**Function**		**Byte**	**Function**
0	00h Fragment	▶		
1...15	User data byte 12...26			
		◀	0	00h Acknowledgement

2. Fragment Host system			CP	
Byte	**Function**		**Byte**	**Function**
0	01h Fragment	▶		
1...15	User data byte 27...41			
		◀	0	10h Acknowledgement

Last fragment Host system			CP	
Byte	**Function**		**Byte**	**Function**
0	0Ah Command	▶		
1...8	User data byte 42...49			
11...15	is not used			
		◀	0	A0h Acknowledgement

Receive data *without fragmentation*

IO-Size = 60byte, Length = 40byte

Host system			CP	
Byte	**Function**		**Byte**	**Function**
		◀	0	0Ah Fragment-Info
			1	00h Telegram-Info-Byte
			2	00h Length high byte

Host system			CP	
Byte	Function		Byte	Function
			3	2Ah Length low byte + 2byte
			4	00h Return Value high byte
			5	00h Return Value low byte
			6...45	User data byte 0...39
			46...59	is not used
0	A0h Acknowledgement	▶	0	

Receive data *with fragmentation*

IO-Size = 16byte, Length = 40byte

Header Host system			CP	
Byte	Function		Byte	Function
		◀	0	09h Fragment-Info
			1	00h Telegram-Info-Byte
			2	00h Length high byte
			3	2Ah Length low byte + 2byte
			4	00h Return Value high byte
			5	00h Return Value low byte
			6...15	User data byte 0...9
0	90h Acknowledgement	▶	0	

1. Fragment Host system			CP	
Byte	Function		Byte	Function
		◀	0	00h Fragment-Info
			1...15	User data byte 10...24
0	00h Acknowledgement	▶	0	

Last fragment Host system			CP	
Byte	Function		Byte	Function
		◀	0	0Ah Fragment-Info
			1...15	User data byte 25...39
0	A0h Acknowledgement	▶	0	

4.4 Communication via handling blocks

Communication

For the processing of the connecting jobs at PLC side a user program is necessary in the CPU.

With a System SLIO CPU use for communication FB 65 SEND_RECV.

The following VIPA specific blocks are used for communication between CPU, CP and a communication partner:

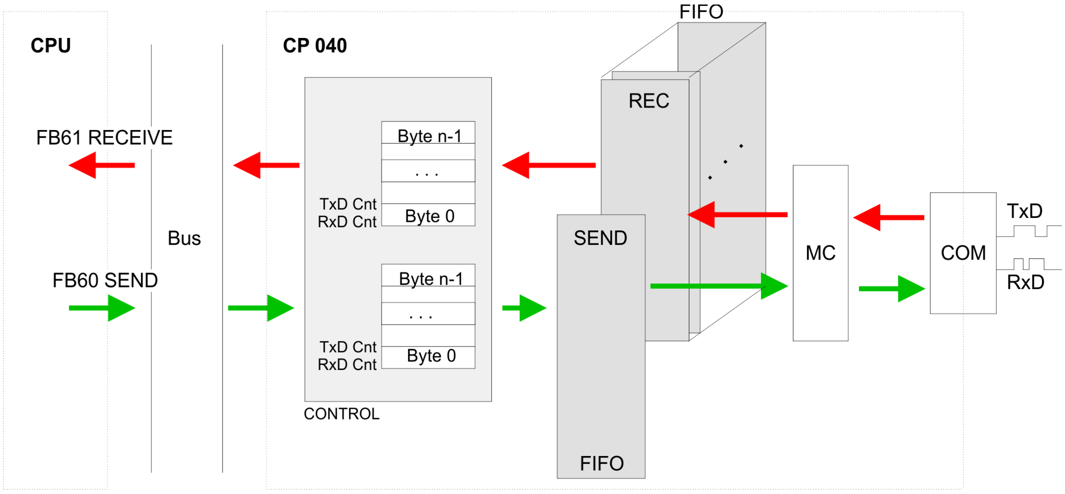
Block	Symbol	Comment
FB 60	SEND	Block for data to be sent to a communication partner.
FB 61	RECEIVE	Block for data receipt from a communication partner.
FB 65	SEND_RECV	Block for data sent and data receive with a communication partner (e.g. System SLIO CPU).

4.4.1 Overview

Communication principle

- By a cyclic call of FB 60 SEND and FB 61 RECEIVE or FB 65 CP040_COM data may be cyclically sent and received by the CP.
- On the CP the transmission of the communication protocols to the communication partner takes place, which may be configured by the hardware configuration.
- A telegram to be sent is divided into blocks in the CPU depending on the IO size and transferred via the data channel to the CP In the CP these blocks are assembled in the send buffer, and when the telegram is complete, the telegram is sent by the serial interface.
- The exchange of received telegrams via the backplane bus is asynchronous.
- If a complete telegram was received via the serial interface, it is stored in a 1024byte ring buffer. From the length of the still free ring buffer the maximum length of a telegram results.
- Depending upon the parametrization up to 250 telegrams can be buffered, whereby their overall length may not exceed 1024.
- If the buffer is full, arriving telegrams are rejected.
- A complete telegram is divided into blocks, depending on the parametrized IO size, and transferred to the backplane bus.

- The data blocks must be assembled in the CPU.
- Since the data exchange via the backplane bus runs asynchronously, a software handshake is used between the CP and the CPU. For this, both handling blocks have the common CONTROL parameter. The same flag byte is to be used for this parameter.



FIFO	Ring buffer max. 250 telegrams 1024byte
CONTROL	Software handshake via CONTROL block

For recognizing a signal change a minimum pulse time is necessary. The decisive factors are CPU cycle time, the refresh time on the CP and the response time of the communication partner.

4.4.2 VIPA Lib

More information about the usage of these blocks can be found in the manual "Serial Communication - SW90GS0MA" at www.vipa.com in the "Service/Support" area at ‘Manuals ➔ VIPA Lib’.

4.5 Diagnostic data

Overview

Via the parameterization you may activate a diagnostic interrupt for the module. With a diagnostic interrupt the module serves for diagnostic data for diagnostic interrupt$_{incoming}$. As soon as the reason for releasing a diagnostic interrupt is no longer present, the diagnostic interrupt$_{going}$ automatically takes place.

Within this time window (1. diagnostic interrupt$_{incoming}$ until last diagnostic interrupt$_{going}$) the MF-LED of the module is on.

DS - Record set for access via CPU, PROFIBUS and PROFINET. The access happens by DS 01h. Additionally the first 4 bytes may be accessed by DS 00h.

IX - Index for access via CANopen. The access happens by IX 2F01h. Additionally the first 4 bytes may be accessed by IX 2F00h.

SX - Subindex for access via EtherCAT with Index 5005h.

More can be found in the according manual of your bus coupler.

Name	Bytes	Function	Default	DS	IX	SX
ERR_A	1	Diagnostic	00h	01h	2F01h	02h
MODTYP	1	Module information	1Ch			03h
ERR_C	1	reserved	00h			04h
ERR_D	1	Diagnostic	00h			05h
CHTYP	1	Channel type	60h			06h
NUMBIT	1	Number diagnostic bits per channel	08h			07h
NUMCH	1	Number channels of the module	01h			08h
CHERR	1	Channel error	01h			09h
CH0ERR	1	Channel-specific error	01h			0Ah
CH1ERR...CH7ERR	7	reserved	00h			0Bh ... 11h
DIAG_US	4	µs ticker	00h			13h

ERR_A Diagnostic

Byte	Bit 7 ... 0
0	■ Bit 0: set at module failure ■ Bit 1: set at internal error ■ Bit 2: set at external error (wire break possible only with RS422) ■ Bit 3: reserved ■ Bit 4: set at missing external power supply ■ Bit 5, 6: reserved ■ Bit 7: set at error in parameterization

MODTYP Modul information

Byte	Bit 7 ... 0
0	- Bit 3 ... 0: Module class – 1100b: CP - Bit 4: set at channel information present - Bit 7 ... 5: reserved

ERR_D Diagnostic

Byte	Bit 7 ... 0
0	- Bit 2 ... 0: reserved - Bit 3: set at internal diagnostics buffer overflow - Bit 4: set at internal communication error - Bit 7 ... 5: reserved

CHTYP Channel type

Byte	Bit 7 ... 0
0	- Bit 6 ... 0: Channel type – 60h: Communication processor - Bit 7: reserved

NUMBIT Diagnostic bits

Byte	Bit 7 ... 0
0	Number of diagnostic bits of the module per channel (here 08h)

NUMCH Channels

Byte	Bit 7 ... 0
0	Number of channels of the module (here 01h)

CHERR Channel error

Byte	Bit 7 ... 0
0	- Bit 0: set at error in channel group 0 - Bit 7 ... 1: reserved

CH0ERR

Byte	Bit 7 ... 0
0	- Bit 3 ... 0: reserved - Bit 4: Wire break (only possible with RS422) - Bit 7 ... 5: reserved

CH1ERR ... CH7ERR

Byte	Bit 7 ... 0
0	Bit 7 ... 0: reserved

DIAG_US µs ticker

Byte	Bit 7 ... 0
0...3	Value of the µs ticker at the moment of the diagnostic

µs ticker

In the SLIO module there is a timer (µs ticker). With PowerON the timer starts counting with 0. After 2^{32}-1µs the timer starts with 0 again.

5 Serial communication protocols

5.1 Overview

Serial transfer of a character

The simplest type of information exchange between two stations is the point-to-point link. Here the CP serves for the interface between a host system and a communication partner. The data are serially transferred. During the serial data transfer the individual bits of one byte of an information are transferred after another in a fixed order.

Character frame

At bi-directional data transfer it is differentiated between *full-duplex* and half-duplex operation. At *half-duplex* operation at one time data may be sent or received. A simultaneous data exchange is only possible at *full- duplex* operation. Each character to be transferred is preceded by a synchronizing pulse as *start bit*. The end of the transferred character is formed by the *stop bit*. Beside the start and stop bit there are further parameterizable agreements between the communication partners necessary for serial data transfer.

This character frame consists of the following elements:

- Transfer speed (Baud rate)
- Character and acknowledgement delay time
- Parity
- Number of data bits
- Number of stop bits

Protocols

The CP serves for an automatic serial data transfer. To do this the CP is equipped with a driver for the corresponding protocols.

The following protocols are described:

- ASCII
- STX/ETX
- 3964(R)
- Modbus (Master, Slave)

5.2 ASCII

5.2.1 Basics ASCII

Mode of operation

ASCII data communication is a simple kind of data exchange that may be compared to a multicast/broadcast function. Individual telegrams are separated by means of character delay time (ZVZ). Within this time the transmitter must have sent its telegram to the receiver. A telegram is only passed on to the host system if this was received completely. The receiving station must acknowledge the receipt of the telegram within the "time delay after command" (ZNA) or command window that was defined in the sending station. These time stamps may be used to establish a simple serial communication link. Since during ASCII transmission apart from the usage of the parity bit no further step takes place for data protection, the data transfer is very efficient however not secured. With the parity the inversion of one bit within a character may be secured. If two or more bits of a character are inverted, this error may no longer be detected.

5.2.2 Parameter data of ASCII

Parameters

DS - Record set for access via CPU, PROFIBUS and PROFINET

IX - Index for access via CANopen

SX - Subindex for access via EtherCAT with Index 3100h + EtherCAT-Slot

More can be found in the according manual of your bus coupler.

Name	Bytes	Function	Default	DS	IX	SX
PII_L	1	Length process image input data [1]	[2]	02h	3100h	01h
PIQ_L	1	Length process image output data [1]	[2]	02h	3101h	02h
DIAG_EN	1	Diagnostic interrupt [1]	00h	00h	3102h	03h
BAUD	1	Baud rate	00h	80h	3103h	04h
PROTOCOL	1	Protocol	01h	80h	3104h	05h
OPTION3	1	Character frame	13h	80h	3105h	06h
OPTION4, 5	2	ZNA 0 ... 65535 (in ms)	0	80h	3106h ... 3107h	07h ... 08h
OPTION6, 7	2	ZVZ 0 ... 65535 (in ms)	250	80h	3108h ... 3109h	09h ... 0Ah
OPTION8	1	Number Receive buffer	1	80h	310Ah	0Bh
OPTION9...14	6	reserved	00h	80h	310Bh ... 3110h	0Ch ... 11h
OPTION15	1	Operating mode	00h	80h	3111h	12h
OPTION16	1	Line assignment	00h	80h	3112h	13h

1) This record set may only be transferred at STOP state.

2) Value depends on the host system.

DIAG_EN: Diagnostic interrupt

Here you activate respectively deactivate the diagnostic function.

Byte	Bit 7 ... 0
0	■ Range of values: – 00h: deactivate – 40h: activate

- Default: 00h

BAUD: Transfer rate

Speed of the data transfer in bit/s (baud). There are the following range of values; other values are not permitted.

Range of values:

Hex	Baud	Hex	Baud	Hex	Baud
00	9600	06	2400	0C	38400
01	150	07	4800	0D	57600
02	300	08	7200	0F	76800
03	600	09	9600	0E	115200
04	1200	0A	14400	10	109700
05	1800	0B	19200		

- Default: 00h (9600Baud)

PROTOCOL

Protocol, which is to be used. This setting influences the structure. For the ASCII protocol enter 01h.

OPTION3: Character frame

Byte	Bit 7 ... 0
0	■ Bit 1, 0: Data bits – 00b = 5 Data bits – 01b = 6 Data bits – 10b = 7 Data bits – 11b = 8 Data bits ■ Bit 3, 2: Parity – 00b = none – 01b = odd – 10b = even – 11b = even ■ Bit 5, 4: Stop bits – 01b = 1 – 10b = 1.5 – 11b = 2 ■ Bit 7, 6: Flow control – 00b = none – 10b = hardware – 11b = XON/XOFF

- *Default: 13h*
  - *(Data bits: 8, Parity: none, Stop bit: 1, Flow control: none)*

Data bits

Number of bits onto which a character is mapped.

Parity

For the purposes of the parity check, the information bits are expanded by the parity bit. The value of the parity bit ("0" or "1") completes the value of all the bits to obtain a prearranged state. If the parity was not specified, the parity bit is set to "1" but it is not evaluated.

Stop bits

The stop bits are appended to each character and signify the end of the character.

Flow control

This is a mechanism that synchronizes the data transfer when the transmitting station sends the data faster than it can be processed by the receiving station. Flow control can be hardware- or software-based (XON/XOFF). Hardware flow control employs the RTS and CTS lines and these must therefore be wired accordingly. Software flow control employs the control characters XON=11h and XOFF=13h. Please remember that your data must not contain these control characters.

OPTION4, 5: ZNA

The delay time that must expire before a command is executed. The ZNA is specified in ms.

Option4: ZNA (High byte)

Option5: ZNA (Low byte)

Range of values: 0 ... 65535

- Default: 0

OPTION6, 7: ZVZ

The character delay time defines the maximum time that may expire between two characters of a single telegram during the reception of the telegram. The ZVZ is specified in ms.

When the ZVZ=0 the character delay time (ZVZ) will be calculated automatically (about double character time).

Option6: ZVZ (High byte)

Option7: ZVZ (Low byte)

Range of values: 0 ... 65535

- Default: 250

OPTION8: Number of receive buffers

Defines the number of receive buffers. When only 1 receive buffer is available no more data can be received while the receive buffer is occupied. The received data can be redirected into an unused receive buffer when you chain up to a maximum of 250 receive buffers.

Range of values: 1 ... 250

- Default: 1

OPTION15: Operating mode

Via the Operating mode you may specify if the interface is operated in half-duplex (RS485) or full-duplex (RS422) operation.

> *At half-duplex parameterization with RS485 software data flow control is not possible.*

Value	Description
00h	Half-duplex - Two-wire operation (RS485) Data is exchanged between the communication partners but only in one direction at a time. In half-duplex operation, therefore, at any one time data is being either sent or received.
01h	Full-duplex - Four-wire operation (RS422) Data is exchanged between the communication partners in both directions simultaneously. In full-duplex operation, therefore, data may be sent and received at the same time. Each communication partner must operate simultaneously a receipt line.

Range of values:

00h: half-duplex

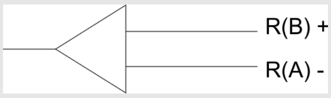
01h: full-duplex

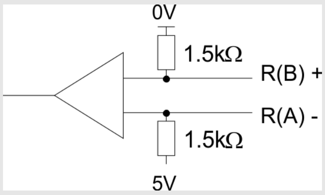
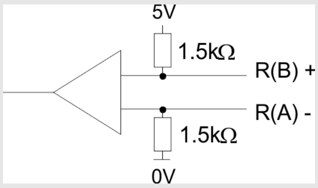
- Default: 00h

OPTION16: Line assignment

For a connection with minimum reflections and the break evaluation at RS422/485 operation, the lines may be preset with defined static voltage levels.

At the CP interface the wiring of the receiver is realized as follows:

Parameters	Description	Wiring of the receiver
00h (Default)	None No pre-assignment of the receiving lines. This setting only makes sense with bus-capable special drivers.	

Parameters	Description	Wiring of the receiver
01h	Signal R(A) 5V (Break evaluation) Signal R(B) 0V With this pre-assignment break detection is with RS422 possible at full-duplex operation.	
02h	Signal R(A) 0V Signal R(B) 5V This pre-assignment corresponds to the idle state (no sender is activated) at half-duplex operation with RS485. Wire-break recognition is not possible, here.	

Range of values:

00h: none

01h: R(A) 5Volt R(B) 0Volt

02h: R(A) 0Volt R(B) 5Volt

- Default: 00h

5.3 STX/ETX

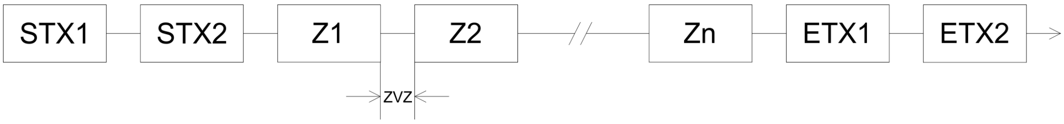
5.3.1 Basics STX/ETX

Mode of operation

STX/ETX is a simple protocol employing header and trailer. The STX/ETX procedure is suitable for the transfer of ASCII characters (20h…7Fh). It does not use block checks. Any data transferred from the periphery must be preceded by an STX (Start of Text) followed by the data characters. An ETX (End of Text) must be inserted as the terminating character. The effective data, which includes all the characters between STX and ETX, are transferred to the host system when the ETX has been received. When data is sent any user data is handed to the CP where it is enclosed with an STX start character and an ETX termination character and transferred to the communication partner.

Telegram structure

You may define up to 2 start and end characters. It is also possible to specify a ZNA for the sending station.



5.3.2 Parameter data of STX/ETX

Parameters

DS - Record set for access via CPU, PROFIBUS and PROFINET

IX - Index for access via CANopen

SX - Subindex for access via EtherCAT with Index 3100h + EtherCAT-Slot

More can be found in the according manual of your bus coupler.

Name	Bytes	Function	Default	DS	IX	SX
PII_L	1	Length process image input data [1]	[2]	02h	3100h	01h
PIQ_L	1	Length process image output data [1]	[2]	02h	3101h	02h
DIAG_EN	1	Diagnostic interrupt [1]	00h	00h	3102h	03h
BAUD	1	Baud rate	00h	80h	3103h	04h
PROTOCOL	1	Protocol	02h	80h	3104h	05h
OPTION3	1	Character frame	13h	80h	3105h	06h
OPTION4, 5	2	ZNA 0 ... 65535 (in ms)	0	80h	3106h ... 3107h	07h ... 08h
OPTION6, 7	2	TMO 0 ... 65535 (in ms)	250	80h	3108h ... 3109h	09h ... 0Ah
OPTION8	1	Number Start identification	1	80h	310Ah	0Bh
OPTION9	1	Start identification 1	2	80h	310Bh	0Ch
OPTION10	1	Start identification 2	0	80h	310Ch	0Dh
OPTION11	1	Number End identification	1	80h	310Dh	0Eh
OPTION12	1	End identification 1	3	80h	310Eh	0Fh
OPTION13	1	End identification 2	0	80h	310Fh	10h

Name	Bytes	Function	Default	DS	IX	SX
OPTION14	1	reserved	00h	80h	3110h	11h
OPTION15	1	Operating mode	00h	80h	3111h	12h
OPTION16	1	Line assignment	00h	80h	3112h	13h
1) This record set may only be transferred at STOP state.						
2) Value depends on the host system.						

DIAG_EN: Diagnostic interrupt

Here you activate respectively deactivate the diagnostic function.

Byte	Bit 7 ... 0
0	■ Range of values: – 00h: deactivate – 40h: activate

- Default: 00h

BAUD: Transfer rate

Speed of the data transfer in bit/s (baud). There are the following range of values; other values are not permitted.

Range of values:

Hex	Baud	Hex	Baud	Hex	Baud
00	9600	06	2400	0C	38400
01	150	07	4800	0D	57600
02	300	08	7200	0F	76800
03	600	09	9600	0E	115200
04	1200	0A	14400	10	109700
05	1800	0B	19200		

- Default: 00h (9600Baud)

PROTOCOL

Protocol, which is to be used. This setting influences the structure. For the STX/ETX protocol enter 02h.

OPTION3: Character frame

Byte	Bit 7 ... 0
0	■ Bit 1, 0: Data bits – 00b = 5 Data bits – 01b = 6 Data bits – 10b = 7 Data bits – 11b = 8 Data bits ■ Bit 3, 2: Parity – 00b = none – 01b = odd – 10b = even – 11b = even ■ Bit 5, 4: Stop bits – 01b = 1 – 10b = 1.5 – 11b = 2 ■ Bit 7, 6: Flow control – 00b = none – 10b = hardware – 11b = XON/XOFF

- *Default: 13h*
  - *(Data bits: 8, Parity: none, Stop bit: 1, Flow control: none)*

Data bits

Number of bits onto which a character is mapped.

Parity

For the purposes of the parity check, the information bits are expanded by the parity bit. The value of the parity bit ("0" or "1") completes the value of all the bits to obtain a pre-arranged state. If the parity was not specified, the parity bit is set to "1" but it is not evaluated.

Stop bits

The stop bits are appended to each character and signify the end of the character.

Flow control

This is a mechanism that synchronizes the data transfer when the transmitting station sends the data faster than it can be processed by the receiving station. Flow control can be hardware- or software-based (XON/XOFF). Hardware flow control employs the RTS and CTS lines and these must therefore be wired accordingly. Software flow control employs the control characters XON=11h and XOFF=13h. Please remember that your data must not contain these control characters.

OPTION4, 5: ZNA

The delay time that must expire before a command is executed. The ZNA is specified in ms.

Option4: ZNA (High byte)

Option5: ZNA (Low byte)

Range of values: 0 ... 65535

- Default: 0

OPTION6, 7: TMO

With TMO the maximum permissible time interval between 2 telegrams is defined. TMO is specified in ms.

Option6: TMO (High-Byte)

Option7: TMO (Low-Byte)

Range of values: 0 ... 65535

- Default: 250

OPTION8: Number start identifications

You may select 1 or 2 start identifications. When you select "1" as number of start identifications, the contents of the 2. start identification is ignored.

Range of values: 0 ... 2

- Default: 1

OPTION9: 10: Start identifications 1, 2

The ASCII value of the start character that precedes a telegram to signify the start of a data transfer. You may select 1 or 2 start characters. When you are using 2 start characters you have to specify "2" at *Number start identifications.*

Start identification 1, 2: Range: 0 ... 255

- Start identification 1: Default: 3
- Start identification 2: Default: 0

OPTION11: Number end identifications

You may select 1 or 2 end identifications. When you select "1" as number of end identifications, the contents of the 2. end identification is ignored.

Range of values: 0 ... 2

- Default: 1

OPTION12, 13: End identifications 1, 2

The ASCII value of the end character that precedes a telegram to signify the end of a data transfer. You may select 1 or 2 end characters. When you are using 2 end characters you have to specify "2" at *Number end identifications*.

End identification 1, 2: Range: 0 ... 255

- End identification 1: Default: 3
- End identification 2: Default: 0

OPTION15: Operating mode

Via the Operating mode you may specify if the interface is operated in half-duplex (RS485) or full-duplex (RS422) operation.

At half-duplex parameterization with RS485 software data flow control is not possible.

Value	Description
00h	Half-duplex - Two-wire operation (RS485) Data is exchanged between the communication partners but only in one direction at a time. In half-duplex operation, therefore, at any one time data is being either sent or received.
01h	Full-duplex - Four-wire operation (RS422) Data is exchanged between the communication partners in both directions simultaneously. In full-duplex operation, therefore, data may be sent and received at the same time. Each communication partner must operate simultaneously a receipt line.

Range of values:

00h: half-duplex

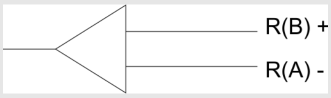
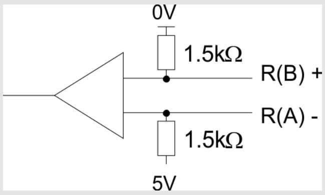
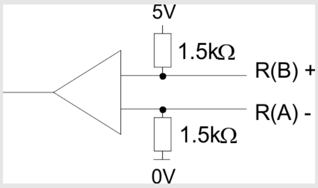
01h: full-duplex

- Default: 00h

OPTION16: Line assignment

For a connection with minimum reflections and the break evaluation at RS422/485 operation, the lines may be preset with defined static voltage levels.

At the CP interface the wiring of the receiver is realized as follows:

Parameters	Description	Wiring of the receiver
00h (Default)	None No pre-assignment of the receiving lines. This setting only makes sense with bus-capable special drivers.	
01h	Signal R(A) 5V (Break evaluation) Signal R(B) 0V With this pre-assignment break detection is with RS422 possible at full-duplex operation.	
02h	Signal R(A) 0V Signal R(B) 5V This pre-assignment corresponds to the idle state (no sender is activated) at half-duplex operation with RS485. Wire-break recognition is not possible, here.	

Range of values:

00h: none

01h: R(A) 5Volt R(B) 0Volt

02h: R(A) 0Volt R(B) 5Volt

- Default: 00h

5.4 3964(R)

5.4.1 Basics 3964(R)

Mode of operation

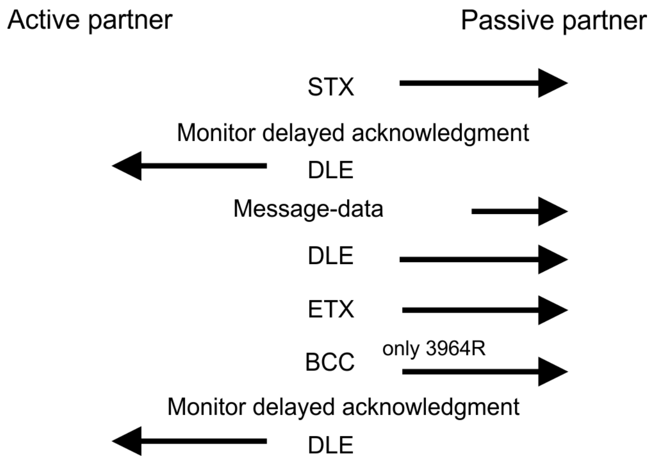
The 3964(R) procedure controls the data transfer of a point-to-point link between the CP and a communication partner. The procedure adds control characters to the telegram data during data transfer. These control characters may be used by the communication partner to verify the complete and error free receipt.

The procedure employs the following control characters:

- STX Start of Text
- DLE Data Link Escape
- ETX End of Text
- BCC Block Check Character (only for 3964R)
- NAK Negative Acknowledge

> *When a DLE is transferred as part of the information it is repeated to distinguish between data characters and DLE control characters that are used to establish and to terminate the connection (DLE duplication). The DLE duplication is reversed in the receiving station. The 3964(R) procedure requires that a lower priority is assigned to the communication partner. When communication partners issue simultaneous send commands the station with the lower priority will delay its send command.*

Procedure



You can maximally transfer 250byte per telegram.

Timeout times

The QVZ is monitored between STX and DLE and between BCC and DLE. ZVZ is monitored for the entire period of receiving the telegram. When the QVZ expires after an STX, the STX is repeated. This process is repeated 5 times after which the attempt to establish a connection is terminated by the transmission of a NAK. The same sequence is completed when a NAK or any other character follows an STX. When the QVZ expires after a telegram (following the BCC-byte) or when a character other than DLE is received the attempt to establish the connection and the telegram are repeated. This process is also repeated 5 times after which a NAK is transmitted and the attempt is terminated.

Passive operation

When the procedure driver is expecting a connection request and it receives a character that is not equal to STX it will transmit a NAK. The driver does not respond with an answer to the reception of a NAK. When the ZVZ is exceeded at reception, a NAK is sent and it is waited for a new connection. When the driver is not ready yet at reception of the STX, it sends a NAK.

Block check character (BCC-Byte)

3964R appends a **B**lock **c**heck **c**haracter to safeguard the transmitted data. The BCC-Byte is calculated by means of an XOR function over the entire data of the telegram, including the DLE/ETX. When a BCC-Byte is received that differs from the calculated BCC, a NAK is transmitted instead of the DLE.

Initialization conflict

If two stations should simultaneously attempt to issue a connection request within the QVZ then the station with the lower priority will transmit the DLE and change to receive mode.

Data Link Escape (DLE-character)

The driver duplicates any DLE-character that is contained in a telegram, i.e. the sequence DLE/DLE is sent. During the reception, the duplicated DLEs are saved as a single DLE in the buffer. The telegram always terminates with the sequence DLE/ETX/BCC (only for 3964R).

The control codes :

- 02h = STX
- 03h = ETX
- 10h = DLE
- 15h = NAK

5.4.2 Parameter data of 3964(R)

Parameters

DS - Record set for access via CPU, PROFIBUS and PROFINET

IX - Index for access via CANopen

SX - Subindex for access via EtherCAT with Index 3100h + EtherCAT-Slot

More can be found in the according manual of your bus coupler.

Name	Bytes	Function	Default	DS	IX	SX
PII_L	1	Length process image input data [1]	[2]	02h	3100h	01h
PIQ_L	1	Length process image output data [1]	[2]	02h	3101h	02h
DIAG_EN	1	Diagnostic interrupt [1]	00h	00h	3102h	03h
BAUD	1	Baud rate	00h	80h	3103h	04h
PROTOCOL	1	Protocol	03h	80h	3104h	05h
OPTION3	1	Character frame	13h	80h	3105h	06h
OPTION4	1	ZNA (x 20ms)	0	80h	3106h	07h
OPTION5	1	ZVZ (x 20ms)	10	80h	3107h	08h
OPTION6	1	QVZ (x 20ms)	10	80h	3108h	09h
OPTION7	1	BWZ (x 20ms)	10	80h	3109h	0Ah
OPTION8	1	STX repetitions	5	80h	310Ah	0Bh
OPTION9	1	DBL	6	80h	310Bh	0Ch
OPTION10	1	Priority	0	80h	310Ch	0Dh

Name	Bytes	Function	Default	DS	IX	SX
OPTION11...14	4	reserved	00h	80h	310Dh ... 3110h	0Eh ... 11h
OPTION15	1	Operating mode	00h	80h	3111h	12h
OPTION16	1	Line assignment	00h	80h	3112h	13h
1) This record set may only be transferred at STOP state.						
2) Value depends on the host system.						

DIAG_EN: Diagnostic interrupt

Here you activate respectively deactivate the diagnostic function.

Byte	Bit 7 ... 0
0	■ Range of values: – 00h: deactivate – 40h: activate

- Default: 00h

BAUD: Transfer rate

Speed of the data transfer in bit/s (baud). There are the following range of values; other values are not permitted.

Range of values:

Hex	Baud	Hex	Baud	Hex	Baud
00	9600	06	2400	0C	38400
01	150	07	4800	0D	57600
02	300	08	7200	0F	76800
03	600	09	9600	0E	115200
04	1200	0A	14400	10	109700
05	1800	0B	19200		

- Default: 00h (9600Baud)

PROTOCOL

Protocol, which is to be used. This setting influences the structure.

Range of values: 03h: 3964

Range of values: 04h: 3964R

- Default: 03h

OPTION3: Character frame

Byte	Bit 7 ... 0
0	■ Bit 1, 0: Data bits – 00b = 5 Data bits – 01b = 6 Data bits – 10b = 7 Data bits – 11b = 8 Data bits ■ Bit 3, 2: Parity – 00b = none – 01b = odd – 10b = even – 11b = even ■ Bit 5, 4: Stop bits – 01b = 1 – 10b = 1.5 – 11b = 2 ■ Bit 7, 6: reserved

- Default: 13h
  - (Data bits: 8, Parity: none, Stop bit: 1)

Data bits

Number of bits onto which a character is mapped.

Parity

For the purposes of the parity check, the information bits are expanded by the parity bit. The value of the parity bit ("0" or "1") completes the value of all the bits to obtain a pre-arranged state. If the parity was not specified, the parity bit is set to "1" but it is not evaluated.

Stop bits

The stop bits are appended to each character and signify the end of the character.

OPTION4: ZNA

The delay time that must expire before a command is executed. The ZNA is specified in units of 20ms.

Range of values: 0 ... 255

- Default: 0

OPTION5: ZVZ

The character delay time (ZVZ) defines the maximum time that may expire between two characters of a single telegram during the reception of the telegram. The ZVZ is specified in units of 20ms.

When the ZVZ=0 the character delay time (ZVZ) will be calculated automatically (about double character time).

Range of values: 0 ... 255

- Default: 10

OPTION6: QVZ

The delayed acknowledgment time defines the maximum time for the acknowledgment from the partner when the connection is being established. The QVZ is specified in units of 20ms.

Range of values: 0 ... 255

- Default: 10

OPTION7: BWZ

BWZ is the max. time between acknowledgement of a request telegram (DLE) and STX of the answer telegram. The BWZ is specified in units of 20ms.

Range of values: 0 ... 255

- Default: 10

OPTION8: STX repetitions

Maximum number of allowed attempts for the CP to establish a connection.

Range of values: 0 ... 255

- Default: 5

OPTION9: DBL

With exceeding the block waiting time (BWZ) you can set the maximum number of repetitions for the request telegram by means of the parameter DBL. If these attempts are unsuccessful, the transmission is interrupted.

Range of values: 0 ... 255

- Default: 6

OPTION10: Priority

A communication partner has a high priority when its transmit request supersedes the transmit request of a partner. When the priority is lower, it must take second place after the transmit request of the partner. The priorities of the two partners must be different for the 3964(R) protocol. You may select one of the following settings:

Range of values: 00h: low

Range of values: 01h: high

- Default: 0

OPTION15: Operating mode

Via the Operating mode you may specify if the interface is operated in half-duplex (RS485) or full-duplex (RS422) operation.

> *At half-duplex parameterization with RS485 software data flow control is not possible.*

Value	Description
00h	Half-duplex - Two-wire operation (RS485) Data is exchanged between the communication partners but only in one direction at a time. In half-duplex operation, therefore, at any one time data is being either sent or received.
01h	Full-duplex - Four-wire operation (RS422) Data is exchanged between the communication partners in both directions simultaneously. In full-duplex operation, therefore, data may be sent and received at the same time. Each communication partner must operate simultaneously a receipt line.

Range of values:

00h: half-duplex

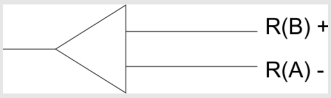
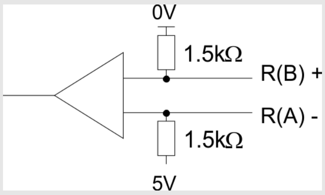
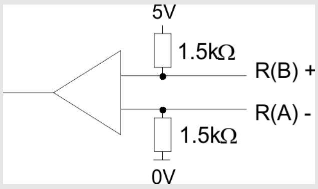
01h: full-duplex

- Default: 00h

OPTION16: Line assignment

For a connection with minimum reflections and the break evaluation at RS422/485 operation, the lines may be preset with defined static voltage levels.

At the CP interface the wiring of the receiver is realized as follows:

Parameters	Description	Wiring of the receiver
00h (Default)	None No pre-assignment of the receiving lines. This setting only makes sense with bus-capable special drivers.	
01h	Signal R(A) 5V (Break evaluation) Signal R(B) 0V With this pre-assignment break detection is with RS422 possible at full-duplex operation.	
02h	Signal R(A) 0V Signal R(B) 5V This pre-assignment corresponds to the idle state (no sender is activated) at half-duplex operation with RS485. Wire-break recognition is not possible, here.	

Range of values:

00h: none

01h: R(A) 5Volt R(B) 0Volt

02h: R(A) 0Volt R(B) 5Volt

- Default: 00h

5.5 Modbus

5.5.1 Basics Modbus

Overview

The Modbus protocol is a communication protocol that defines a hierarchic structure between a master and several slaves. Physically, Modbus transmits via a serial half-duplex connection as point-to-point connection with RS232 or as multi-point connection with RS485.

Master-Slave-Communication

There are no bus conflicts for the master, because the master can only communicate with one slave at a time. After the master requested a telegram, it waits for an answer until an adjustable wait period has expired. During the latency the communication with another slaves is not possible.

Telegram-structure

The request telegrams of the master and the respond telegrams of a slave have the same structure:

Start ID	Slave address	Function code	Data	Flow control	End ID

Broadcast with slave address = 0

A request may be addressed to a certain slave or sent as broadcast telegram to all slaves. For identifying a broadcast telegram, the slave address 0 is set. Only write commands may be sent as broadcast.

ASCII-, RTU Modus

Modbus supports two different transmission modes:

- ASCII mode:
  - Every byte is transferred in 2-character ASCII code. A start and an end ID mark the data. This enables high control at the transmission but needs time.
- RTU mode:
  - Every byte is transferred as character. Thus enables a higher data throughput than the ASCII mode. Instead of start and end ID, RTU uses a watchdog.

The mode selection is made at the parameterization.

5.5.2 Modbus at the CP from VIPA

The CP supports several Modbus operating modes that are described in the following:

Modbus Master

In *Modbus Master* operation you control the communication via your PLC user application in your host system. By means of the Modbus function codes you can access the Modbus slaves with read write functions of the Modbus master. There is the possibility to transfer up to 250byte user data with one telegram.

Modbus Slave short

In *Modbus Slave short* operation the CP communicates with a Modbus Master. Depending on the function code, the CP receives data from the Modbus Master or serves for data for him. The data handling on slave side automatically takes place. This operation mode is especially convenient for the fast transfer of small volumes of data via Modbus.

Modbus Slave long

In *Modbus Slave long* operation only a changed data area, beginning with 0 is transferred from the CP to the host system. If the Modbus master requests data, you have to serve for the relevant data in the CP with an user program. Writing master accesses may not lie outside of the receipt area!

> *Only after all data are present in the CP, the CP sends an answer telegram to the master.*

5.5.3 Parameter data of Modbus

Parameters

DS - Record set for access via CPU, PROFIBUS and PROFINET

IX - Index for access via CANopen

SX - Subindex for access via EtherCAT with Index 3100h + EtherCAT-Slot

More can be found in the according manual of your bus coupler.

Name	Bytes	Function	Default	DS	IX	SX
PII_L	1	Length process image input data [1]	[2]	02h	3100h	01h
PIQ_L	1	Length process image output data [1]	[2]	02h	3101h	02h
DIAG_EN	1	Diagnostic interrupt [1]	00h	00h	3102h	03h
BAUD	1	Baud rate	00h	80h	3103h	04h
PROTOCOL	1	Protocol	0Bh	80h	3104h	05h
OPTION3	1	Character frame	13h	80h	3105h	06h
OPTION4	1	Slave address	1	80h	3106h	07h
OPTION5, 6	2	Delay time	0	80h	3107h ... 3108h	08h ... 09h
OPTION7...14	8	reserved	00h	80h	3109h ... 3110h	0Ah ... 11h
OPTION15	1	Operating mode	00h	80h	3111h	12h
OPTION16	1	Line assignment	00h	80h	3112h	13h
1) This record set may only be transferred at STOP state.						
2) Value depends on the host system.						

DIAG_EN: Diagnostic interrupt

Here you activate respectively deactivate the diagnostic function.

Byte	Bit 7 ... 0
0	■ Range of values: – 00h: deactivate – 40h: activate

- Default: 00h

BAUD: Transfer rate

Speed of the data transfer in bit/s (baud). There are the following range of values; other values are not permitted.

Range of values:

Hex	Baud	Hex	Baud	Hex	Baud
00	9600	06	2400	0C	38400
01	150	07	4800	0D	57600
02	300	08	7200	0F	76800
03	600	09	9600	0E	115200
04	1200	0A	14400	10	109700
05	1800	0B	19200		

- Default: 00h (9600Baud)

PROTOCOL

Protocol, which is to be used. This setting influences the structure.

Range of values with Modbus:

0Ah:	Modbus Master ASCII
0Bh:	Modbus RTU
0Ch:	Modbus Slave ASCII short
0Dh:	Modbus Slave RTU short
1Ch:	Modbus Slave ASCII long
1Dh:	Modbus Slave RTU long

- Default: 0Bh

OPTION3: Character frame

Byte	Bit 7 ... 0
0	■ Bit 1, 0: Data bits – 00b = 5 Data bits – 01b = 6 Data bits – 10b = 7 Data bits – 11b = 8 Data bits ■ Bit 3, 2: Parity – 00b = none – 01b = odd – 10b = even – 11b = even ■ Bit 5, 4: Stop bits – 01b = 1 – 10b = 1,5 – 11b = 2 ■ Bit 7, 6: reserved

- Default: 13h
  - *(Data bits: 8, Parity: none, Stop bit: 1)*

Data bits

Number of bits onto which a character is mapped.

Parity

For the purposes of the parity check, the information bits are expanded by the parity bit. The value of the parity bit ("0" or "1") completes the value of all the bits to obtain a pre-arranged state. If the parity was not specified, the parity bit is set to "1" but it is not evaluated.

Stop bits

The stop bits are appended to each character and signify the end of the character.

OPTION4: Slave address

Enter in the Modbus slave protocol an address for the Modbus slave. By means of this address a Modbus slave may be accessed with the Modbus function codes. With Modbus master this parameter is ignored.

Range of values: 1 ... 255

- Default: 1

OPTION5, 6: Delay time

Here for the Modbus master a delay time in ms is to be preset. With 0 the delay time is evaluated automatically depending on the protocol with the following formula:

Modbus ASCII:

$$50ms + \frac{2926000ms}{Baudrate} \; x \; Bit/s$$

with Baudrate in bit/s

Modbus RTU:

$$50ms + \frac{5190000ms}{Baudrate} \; x \; Bit/s$$

with Baudrate in bit/s

In Modbus slave this parameter is ignored.

Option5: Delay time (high byte)

Option6: Delay time (low byte)

Range of values: 0 ... 60000 in ms

- Default: 0

OPTION15: Operating mode

Via the Operating mode you may specify if the interface is operated in half-duplex (RS485) or full-duplex (RS422) operation.

At half-duplex parameterization with RS485 software data flow control is not possible.

Value	Description
00h	Half-duplex - Two-wire operation (RS485) Data is exchanged between the communication partners but only in one direction at a time. In half-duplex operation, therefore, at any one time data is being either sent or received.
01h	Full-duplex - Four-wire operation (RS422) Data is exchanged between the communication partners in both directions simultaneously. In full-duplex operation, therefore, data may be sent and received at the same time. Each communication partner must operate simultaneously a receipt line.

Range of values:

00h: half-duplex

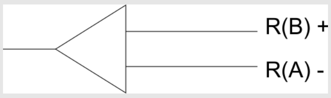
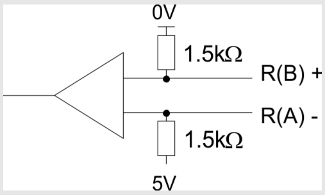
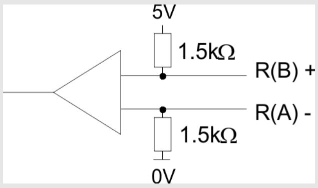
01h: full-duplex

- Default: 00h

OPTION16: Line assignment

For a connection with minimum reflections and the break evaluation at RS422/485 operation, the lines may be preset with defined static voltage levels.

At the CP interface the wiring of the receiver is realized as follows:

Parameters	Description	Wiring of the receiver
00h (Default)	None No pre-assignment of the receiving lines. This setting only makes sense with bus-capable special drivers.	
01h	Signal R(A) 5V (Break evaluation) Signal R(B) 0V With this pre-assignment break detection is with RS422 possible at full-duplex operation.	
02h	Signal R(A) 0V Signal R(B) 5V This pre-assignment corresponds to the idle state (no sender is activated) at half-duplex operation with RS485. Wire-break recognition is not possible, here.	

Range of values:

00h: none

01h: R(A) 5Volt R(B) 0Volt

02h: R(A) 0Volt R(B) 5Volt

- Default: 00h

5.6 Deployment - Modbus

5.6.1 Modbus - Overview

The number of input and output data, dependent on the IO-Size, is parameterizable via GSD file at the 040-1CA00. For the deployment with Modbus a hardware configuration must always be executed.

Requirements for operation

The following components are required for the deployment of the System SLIO Modbus modules:

- Master System consisting of System SLIO with CP 040
- Slave System consisting of System SLIO with CP 040
- Siemens SIMATIC manager respectively WinPLC7 from VIPA
- GSD file
- VIPA handling blocks Fx000011_Vxxx.zip
- Serial connection between both CP

Parameterization

The CP 040 always requires a hardware configuration. For this the inclusion of the VIPA GSD file into the hardware catalog is necessary. The parameterization has the following approach:

- Start the Siemens SIMATIC manager respectively WinPLC7 from VIPA.
- Install the selected GSD-file in the hardware catalog.
- Configure a SLIO system.
- Insert a CP 040 labeled with "Modbus".
- Parameterize the CP 040 to your specifications.
- Transfer your project to the PLC.

PLC application

Except of the "Modbus Slave short", the communication always requires a PLC application. For this the communication happens via handling blocks that you may include into your configuration tool by means of the VIPA library Fx000011_Vxxx.zip. The library is available at the service area of www.vipa.com.

5.6.1.1 Communication options

The following text describes the communication options between Modbus master and Modbus slave with the following combination options:

- CP 040 Modbus Master ⇔ CP 040 Modbus Slave short
- CP 040 Modbus Master ⇔ CP 040 Modbus Slave long

Master ⇔ Slave short

Modbus Master

The communication in master mode happens via data blocks deploying the CP 040 handling blocks FB 60 - SEND and FB 61 - RECEIVE (or FB 65 SEND_RECV). Here you can transfer up to 250byte user data.

Modbus Slave short

The Modbus Slave short mode limits the volume of user data for in- and output to the IO-Size. For this you only need a hardware configuration at the slave section.

Approach

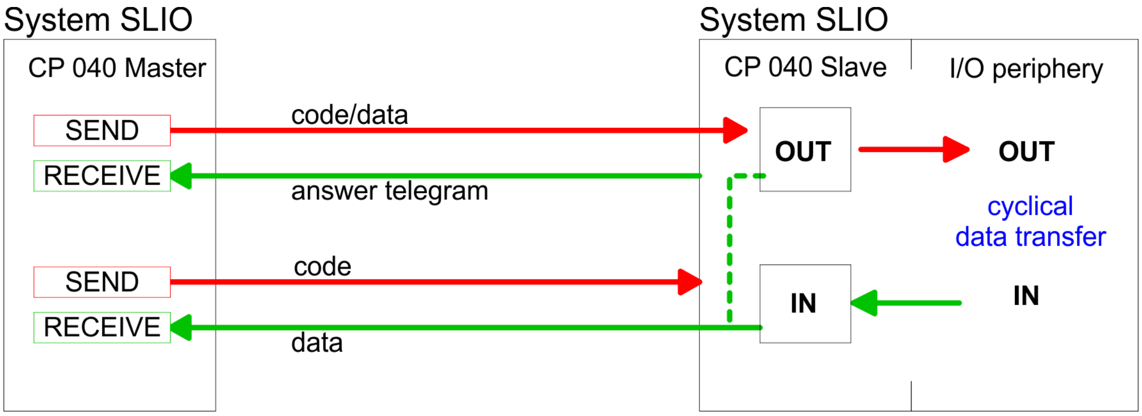
1. Build-up each for the master and slave side a SLIO system, which both contain a CP 040.
2. Connect both systems via the serial interface.
3. Configure the master section.

   The configuration of the CP 040 as Modbus master happens via the hardware configuration. In addition you need a PLC user application for the communication with the following structure:

   OB 100: One-time call of the handling blocks FB 60 - SEND and FB 61 - RECEIVE (or FB 65 SEND_RECV) with all parameters and set R for initialization.

   OB 1: Call of FB 60 - SEND (or FB 65 SEND_RECV) with error evaluation. For this the telegram is to be stored in the send block according to the Modbus rules. Call of FB 61 - RECEIVE with error evaluation. The data are stored in the receive block according to Modbus rules.
4. Configure the slave section.

   The parameterization of the CP 040 happens via the hardware configuration. Enter here the start address for the in- and output area from where on, depending on the IO Size, the input and output data are stored in the CPU.



Master ⇔ Slave long

Modbus Master

The communication in master mode happens via data blocks deploying the CP 040 handling blocks FB 60 - SEND and FB 61 - RECEIVE (or FB 65 SEND_RECV). Here you can transfer up to 250byte user data.

Modbus Slave long

In the Modbus Slave long mode only a changed data area is transferred to the CPU via FB 61 - RECEIVE starting with 0. If the master requests data it has to be made sure that the relevant data are present in the CP. With a FB 60 - SEND call a wanted data area is transferred to the CP starting with 0.

Approach

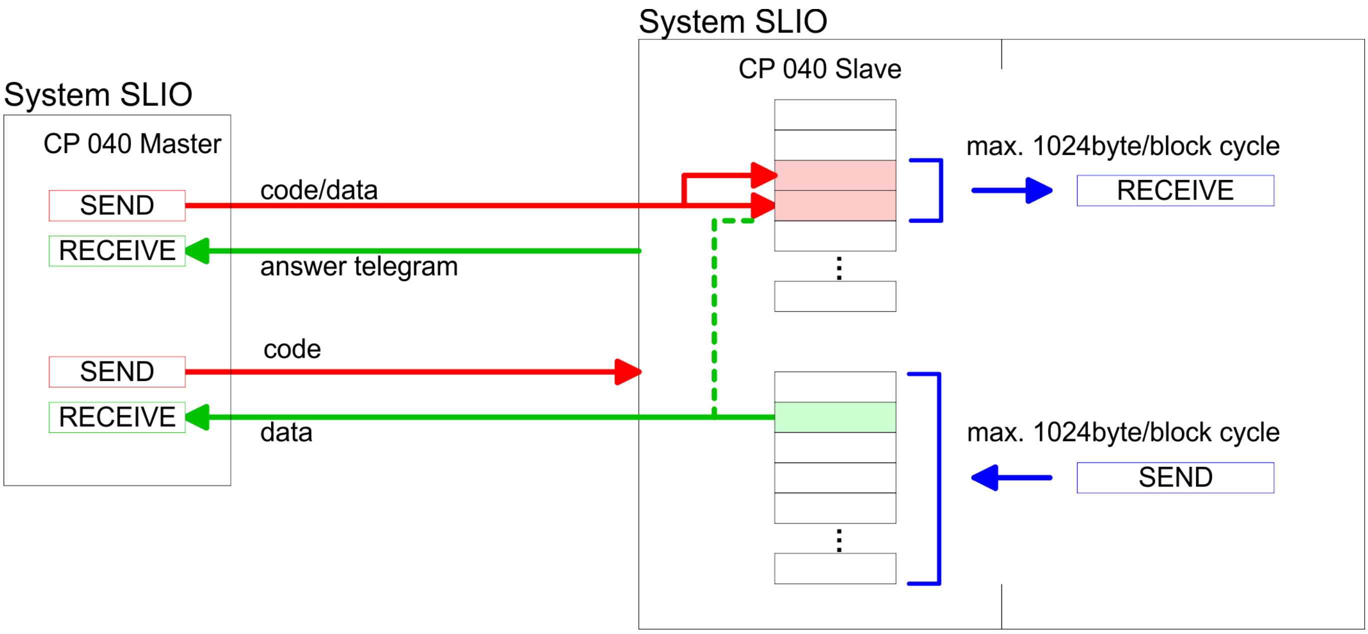
1. Build-up each for the master and slave side a SLIO system, which both contain a CP 040.
2. Connect both systems via the serial interface.
3. Configure the master section.

   The project engineering of the master section happens like shown in the sample above.

4. Configure the slave section. The configuration of the CP 040 as Modbus master happens via the hardware configuration. In addition you need a PLC user application for the communication with the following structure:

   OB 100: One-time call of the handling blocks FB 60 - SEND and FB 61 - RECEIVE (or FB 65 SEND_RECV) with all parameters and set R for initialization.

   OB 1: Call of FB 60 - SEND (or FB 65 SEND_RECV) with error. For this an area starting at 0 is stored in the CP 040 where the master may gain access via Modbus. The FB 61 - RECEIVE with error evaluation allows you to transfer a data area into the CPU. At a data change by the master, only those data are transferred to the CPU where changes occurred.



5.6.2 Modbus - Access to multiple slaves

At deployment of multiple slaves with RS485, there cannot occur bus conflict errors because the master may only communicate with one slave at a time. The master sends a command telegram to the slave specified via the address and waits for a certain time where within the slave may send its respond telegram. During the latency the communication with another slave is not possible. For the communication with multiple slaves every slave needs a SEND data block for the command telegram and a RECEIVE data block for the respond telegram. An application with several slaves would consist of an according amount of data blocks with commands.

These are executed in sequence:

1. slave:

1. Send command telegram to slave address 1. slave
2. Receive respond telegram from slave address 1. slave
3. Evaluate respond telegram

2. slave:

1. Send command telegram to slave address 2. slave
2. Receive respond telegram from slave address 2. slave

3. Evaluate respond telegram

3. slave:

...

A request may be sent to a specified slave or as broadcast telegram to all slaves. To mark a broadcast telegram the slave address is set to 0.

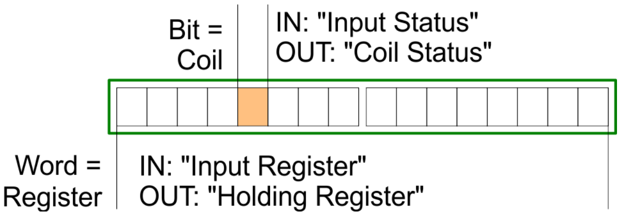
Only write commands may be sent as broadcast.

After a broadcast the master is not waiting for a respond telegram.

5.6.3 Modbus - Function codes

Naming convention

Modbus has some naming conventions:



- Modbus differentiates between bit and word access; bits = "Coils" and words = "Register".
- Bit inputs are referred to as "Input-Status" and bit outputs as "Coil-Status".
- word inputs are referred to as "Input-Register" and word outputs as "Holding-Register".

Range definitions

Normally the access at Modbus happens by means of the ranges 0x, 1x, 3x and 4x.

0x and 1x gives you access to digital bit areas and 3x and 4x to analog word areas.

For the CPs from VIPA is not differentiating digital and analog data, the following assignment is valid:

0x - Bit area for master output data

Access via function code 01h, 05h, 0Fh

1x - Bit area for master input data

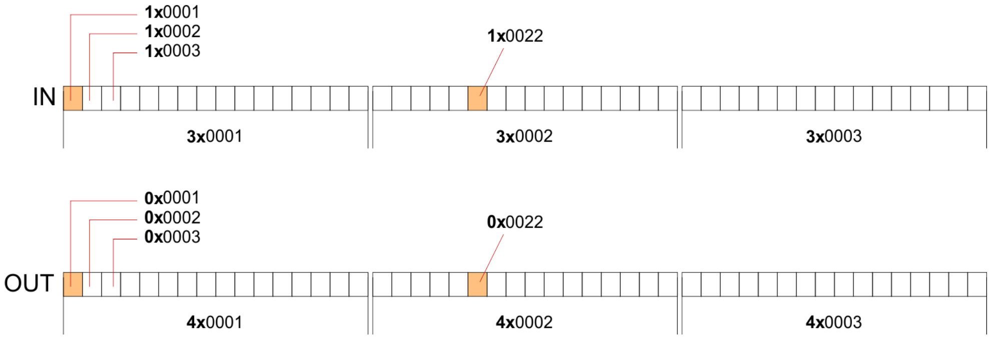
Access via function code 02h

3x - word area for master input data

Access via function code 04h

4x - word area for master output data

Access via function code 03h, 06h, 10h



A description of the function codes follows below.

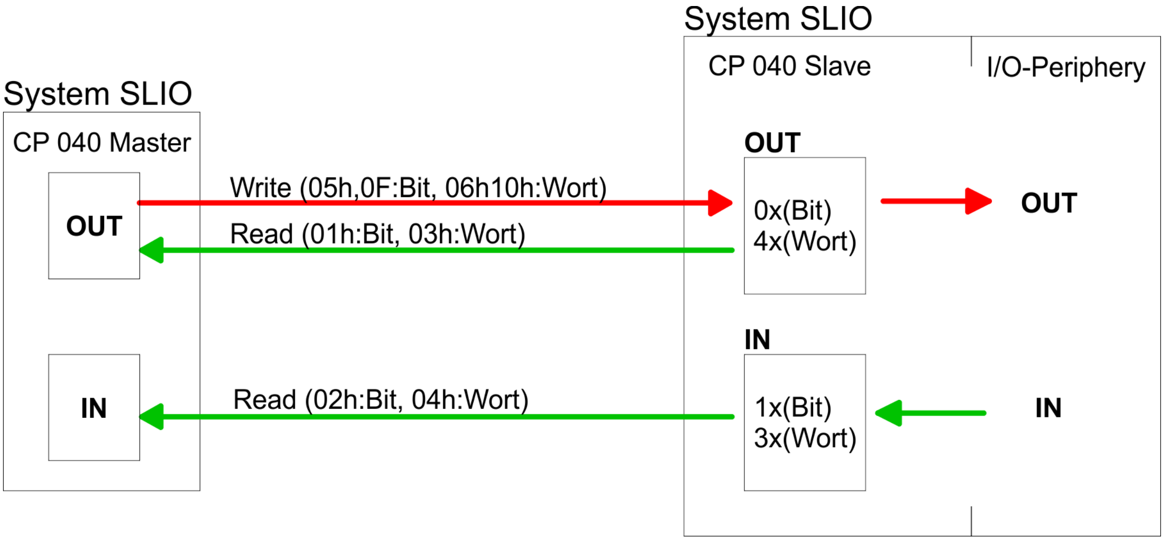
Overview

With the following Modbus function codes a Modbus master can access a Modbus slave: With the following Modbus function codes a Modbus master can access a Modbus slave. The description always takes place from the point of view of the master:

Code	Command	Description
01h	Read n bits	Read n bits of master output area 0x
02h	Read n bits	Read n bits of master input area 1x
03h	Read n words	Read n words of master output area 4x
04h	Read n words	Read n words master input area 3x
05h	Write 1 bit	Write 1 bit to master output area 0x
06h	Write 1 word	Write 1 word to master output area 4x
0Fh	Write n bits	Write n bits to master output area 0x
10h	Write n words	Write n words to master output area 4x

Point of View of "Input" and "Output" data

The description always takes place from the point of view of the master. Here data, which were sent from master to slave, up to their target are designated as "output" data (OUT) and contrary slave data received by the master were designated as "input" data (IN).



Respond of the slave

If the slave announces an error, the function code is send back with an "OR" 80h.

Without an error, the function code is sent back.

Slave answer:	Function code OR 80h	→ Error
	Function code	→ OK

Byte sequence in a word

1 word	
High byte	Low byte

Check sum CRC, RTU, LRC

The shown check sums CRC at RTU and LRC at ASCII mode are automatically added to every telegram. They are not shown in the data block.

Slave address

The Slave address must be the same address as the parameterized Slave address (OPTION4).

Read n bits 01h, 02h

Code 01h: Read n bits of master output area 0x

Code 02h: Read n bits of master input area 1x

Command telegram

Slave address	Function code	Address 1. bit	Number of bits	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Number of read bytes	Data 1. byte	Data 2. byte	...	Check sum CRC/LRC
1byte	1byte	1byte	1byte	1byte		1word
			max. 250byte			

Read n words 03h, 04h

03h: Read n words of master output area 4x

04h: Read n words master input area 3x

Command telegram

Slave address	Function code	Address 1. bit	Number of words	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Number of read bytes	Data 1. word	Data 2. word	...	Check sum CRC/LRC
1byte	1byte	1byte	1word	1word		1word
			max. 125words			

Write 1 bit 05h

Code 05h: Write 1 bit to master output area 0x

A status change is via "Status bit" with following values:

"Status bit" = 0000h → Bit = 0

"Status bit" = FF00h → Bit = 1

Command telegram

Slave address	Function code	Address bit	Status bit	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Address bit	Status bit	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Write 1 word 06h

Code 06h: Write 1 word to master output area 4x

Command telegram

Slave address	Function code	Address word	Value word	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Address word	Value word	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Write n bits 0Fh

Code 0Fh: Write n bits to master output area 0x

Please regard that the number of bits has additionally to be set in byte.

Command telegram

Slave address	Function code	Address 1. bit	Number of bits	Number of bytes	Data 1. byte	Data 2. byte	...	Check sum CRC/LRC
1byte	1byte	1word	1word	1byte	1byte	1byte	1byte	1word
					max. 250byte			

Respond telegram

Slave address	Function code	Address 1. bit	Number of bits	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Write n words 10h

Code 10h: Write n words to master output area 4x

Command telegram

Slave address	Function code	Address 1. word	Number of words	Number of bytes	Data 1. word	Data 2. word	...	Check sum CRC/LRC
1byte	1byte	1word	1word	1byte	1word	1word	1word	1word
					max. 125words			

Respond telegram

Slave address	Function code	Address 1. word	Number of words	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

5.6.4 Modbus - Error messages

Overview

At the communication with Modbus there are 2 error types:

- Master doesn’t receive valid data
- Slave responds with error message

Master doesn’t receive valid data

If the slave doesn’t answer within the specified delay time or if a telegram is defective, the master enters an error message into the receive block in plain text.

The following error messages may occur:

ERROR01 NO DATA	*Error no data* No telegram arrived within the specified delay time.
ERROR02 D LOST	*Error data lost* No data is available because either the receive buffer is full or an error occurred in the receive section.
ERROR03 F OVERF	*Error frame overflow* The telegram end wasn’t recognized or maximum telegram length exceeded.
ERROR04 F INCOM	*Error frame incomplete* Only a part telegram has been received.
ERROR05 F FAULT	*Error frame fault* The check sum of the telegram is faulty.
ERROR06 F START	*Error frame start* The start bit it wrong. this error may only occur with Modbus-ASCII.

Slave answers with error message

If the slave answers with an error, the function code is sent back like shown below, marked as "or" with 80h:

DB11.DBD 0	**DW#16#05900000** with 05 →	**Respond telegram** Slave address 05h
	90 →	Function code 90h (error message, for 10h "or" 80h = 90h)
	0000 →	The rest data is not relevant, for an error has been sent.