Handbücher/Manuals

art of automation

VIPA
Gesellschaft für Visualisierung
und Prozessautomatisierung mbH

Ohmstraße 4
D-91074 Herzogenaurach
Tel.: +49-9132-744-0
Fax: +49-9132-744-144
Internet: www.vipa.de
E-Mail: Info@vipa.de

Manual

VIPA System SLIO
IM 053-1CA00

Order No.: VIPA HB300E_IM
Reference: RE_053-1CA00
Rev. 09/38

This manual is part of the documentation package
with order number VIPA HB300E_IM and relevant for:

Product	Order number	as of state: HW	FW
IM 053CAN	VIPA 053-1CA00	01	4.9.1

The information contained in this manual is supplied without warranties. The information is subject to change without notice.

Ohmstraße 4, D-91074 Herzogenaurach,
Tel.: +49 (91 32) 744 -0
Fax.: +49 (91 32) 744-1864
EMail: info@vipa.de
http://www.vipa.de

Hotline: +49 (91 32) 744-1150

Disclaimer of liability

The contents of this manual were verified with respect to the hard- and software.

However, we assume no responsibility for any discrepancies or errors. The information in this manual is verified on a regular basis and any required corrections will be included in subsequent editions.

Suggestions for improvement are always welcome.

Trademarks

VIPA, SLIO and System 300V

are registered trademarks of VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7

is a registered trademark of profichip GmbH

CANopen and CiA

are registered trademarks of CAN in Automation e.V.

SIMATIC, STEP and S7-300

are registered trademarks of Siemens AG.

Any other trademarks referred to in the text are the trademarks of the respective owner and we acknowledge their registration.

About this manual

This manual describes the IM 053-1CA00 bus coupler for CANopen of the System SLIO from VIPA. Here you may find every information for commissioning and operation.

Overview

Chapter 1: Basics and Assembly

The focus of this chapter is on the introduction of the VIPA System SLIO. Here you will find the information required to assemble and wire a controller system consisting of System SLIO components.

Besides the dimensions the general technical data of System SLIO will be found.

Chapter 2: Hardware description

Here the hardware components of the IM 053-1CA00 are more described. You will find the technical data at the end of this chapter.

Chapter 3: Deployment

This chapter contains the description of the IM 053-1CA00 with CANopen.

Besides the fast introduction concerning the project engineering for "experts" you may find an introduction to the telegram structure and the function codes of CANopen.

The chapter is finished by the description of the Emergency Object as well as the Network Management NMT.

Contents

User considerations

Objective and contents

This manual describes the IM 053-1CA00 of the System SLIO from VIPA. It contains a description of the structure, project engineering and deployment.

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
- an overview of the topics for every chapter

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.

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

Note!
Supplementary information and useful tips.

Safety information

Applications conforming with specifications

The System SLIO is constructed and produced for:

- communication and process control
- general control and automation applications
- 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

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

- Modification to the process control system should only be carried out when the system has been disconnected from power!
- Installation and 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!

Chapter 1 Basics and Assembly

Overview

The focus of this chapter is on the introduction of the VIPA System SLIO. Here you will find the information required to assemble and wire a controller system consisting of System SLIO components.

Besides the dimensions the general technical data of System SLIO will be found.

Content

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.

Attention!

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

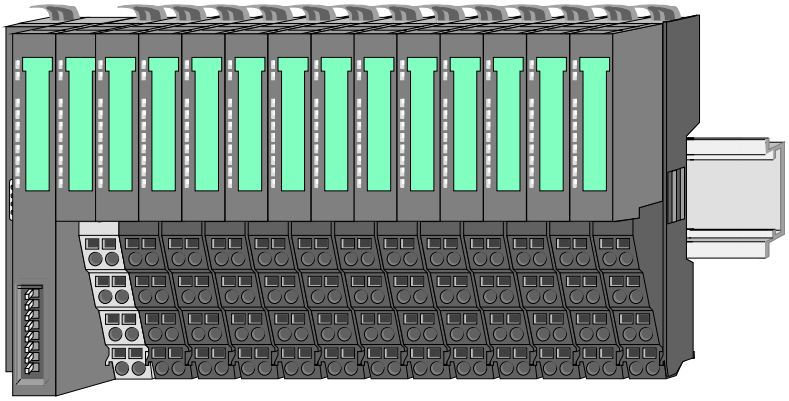
System conception

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.

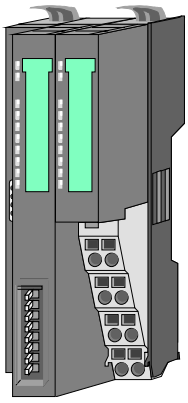


Components

The System SLIO consists of the following components:

- Bus coupler
- Periphery modules
- Power modules
- Accessories

Bus coupler



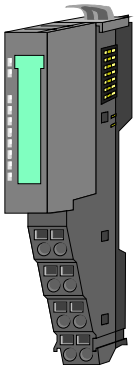
With a bus coupler bus interface and power module is integrated to one casing. With the bus interface you get access to a subordinated bus system.

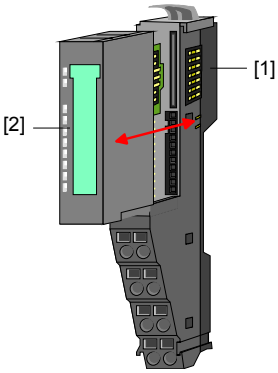
Via the integrated power module for power supply the bus interface is supplied as well as the electronic of the connected periphery modules.

The DC 24 power section supply for the linked periphery modules is established via a further connection.

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.

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

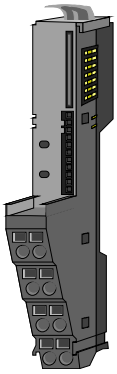




[1] Terminal module

[2] Electronic module

Terminal module

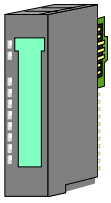


The *terminal module* serves to carry the electronic module, contains the backplane bus with power supply for the electronic, the DC 24V power section supply and the staircase-shaped terminal for wiring.

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 save sliding mechanism.

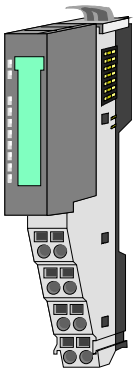
With an error the defective module may be exchanged for a functional module with standing installation.

By an integrated coding only the modules may be plugged, which may be combined.

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.

Power module



In the system SLIO the power supply is established by power modules. These are either integrated to the bus coupler or may be installed between the periphery modules to define isolated areas of the DC 24V power section supply.

For better recognition the color of the power modules are contrasting to the periphery modules.

Accessories

Shield bus carrier

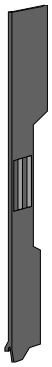
The shield bus carrier serves to carry the shield bus to connect cable shields.

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 adaption to a flat mounting rail you may remove the spacer of the shield bus carrier.

Bus cover



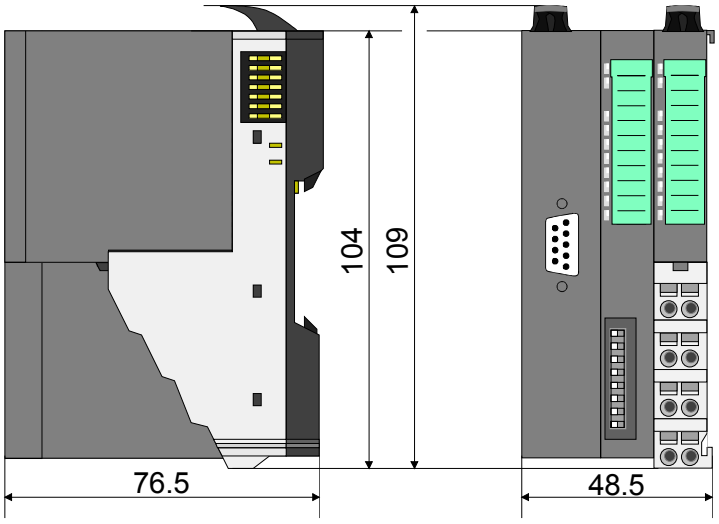
With each bus coupler, to protect the backplane bus connectors, there is a mounted bus cover in the scope of delivery.

You have to remove the bus cover of the bus coupler before mounting a SLIO module.

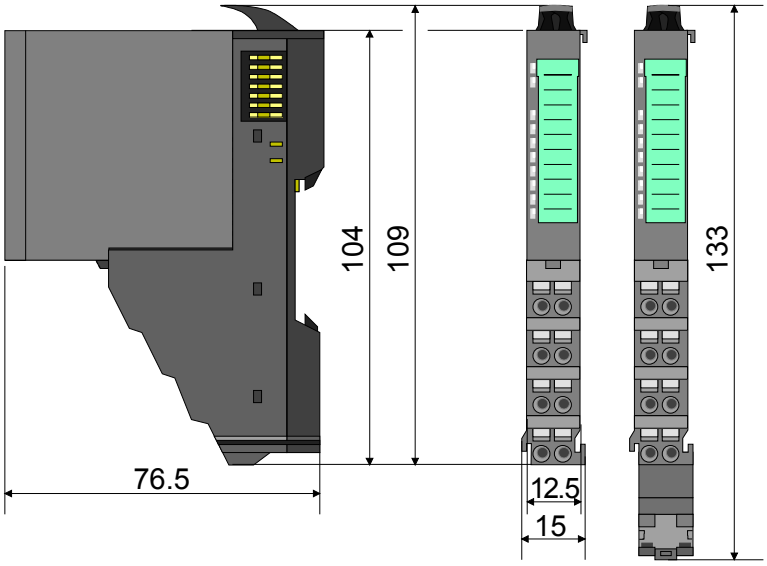
For the protection of the backplane bus connector you should always mount the bus cover at the last module of your system again.

Dimensions

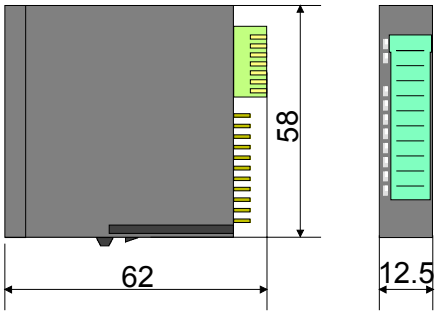
Dimensions bus coupler



Dimensions periphery module



Dimensions electronic module



Dimensions in mm

Installation

Functional principle

Mounting terminal module

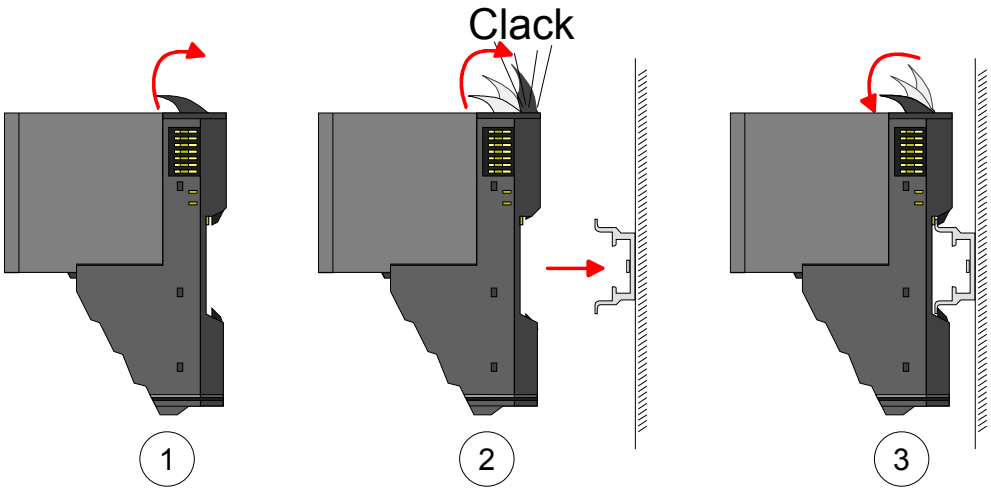
There is a locking lever at the top side of the terminal module. For mounting and de-mounting this locking lever is to turn upwards until this engages audible.

Now the module may be pulled forward.

For mounting plug 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 downwards the locking lever.

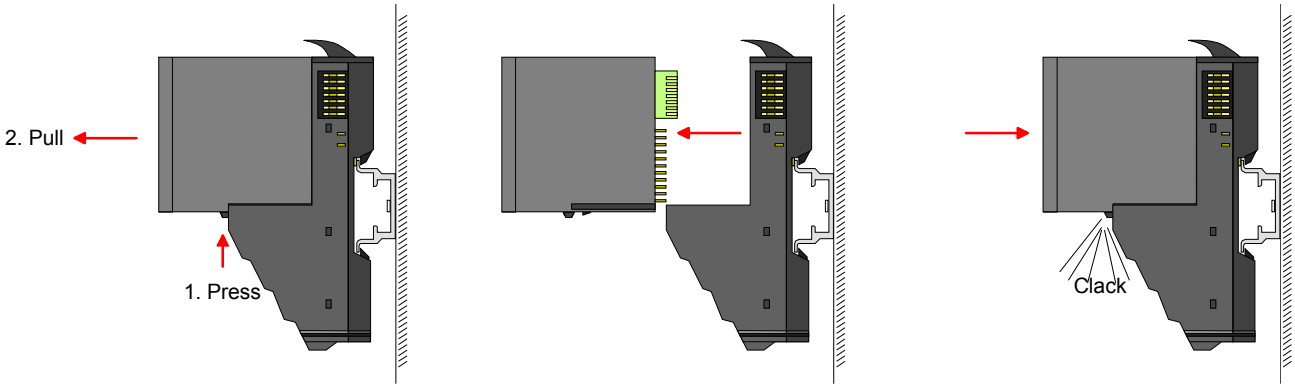
The modules may either separately be mounted to the mounting rail or as block. Here is to be considered that each locking lever is opened.



Mounting electronic module

For mounting between 2 modules and for the exchange of a defective 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.



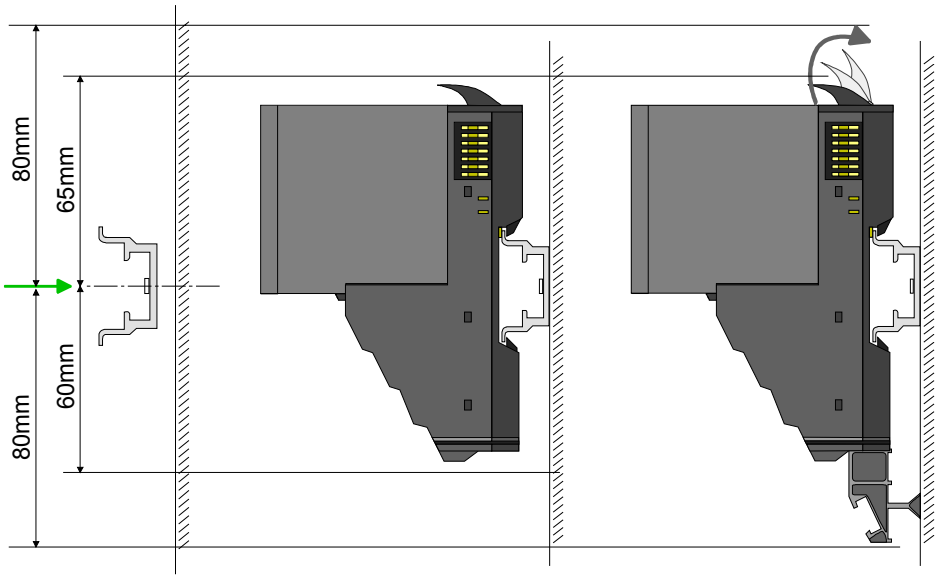
Mounting Proceeding

The modules were directly be mounted to the mounting rail and so connected to the backplane bus and the power supply for the electronic and power section.

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.

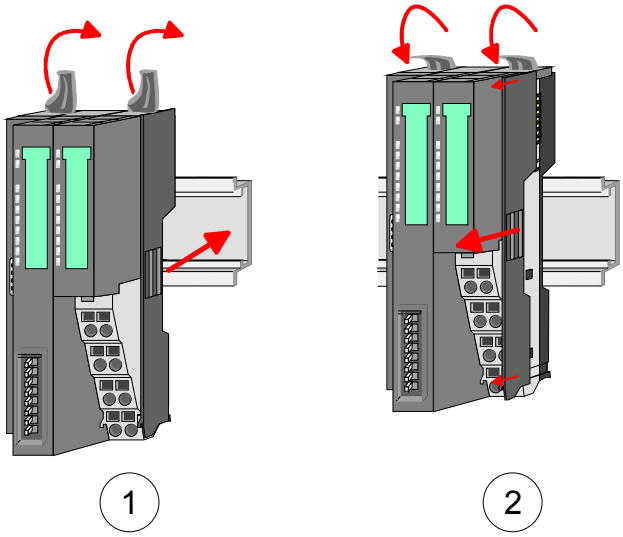
Mounting mounting rail

- 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.



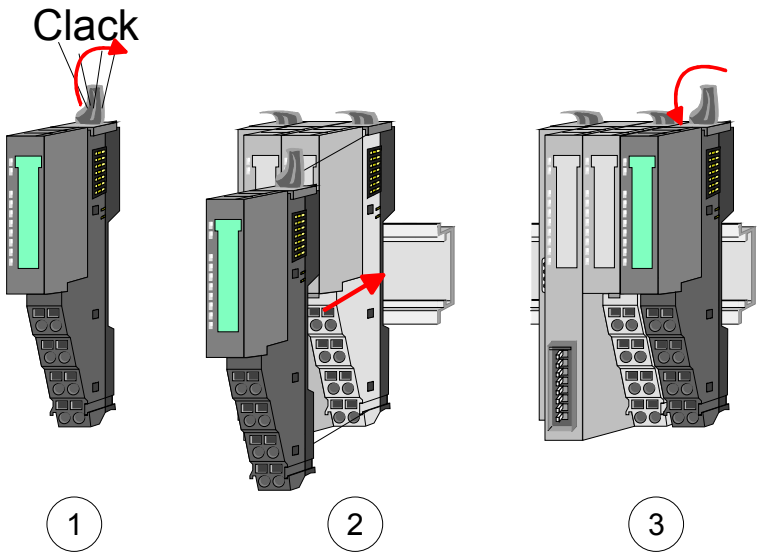
Mounting Head module (e.g. bus coupler)

- Start at the left side with the head module (e.g. bus coupler). For this turn both locking lever upwards, put the head module to the mounting rail and turn both locking lever downwards.
- 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.



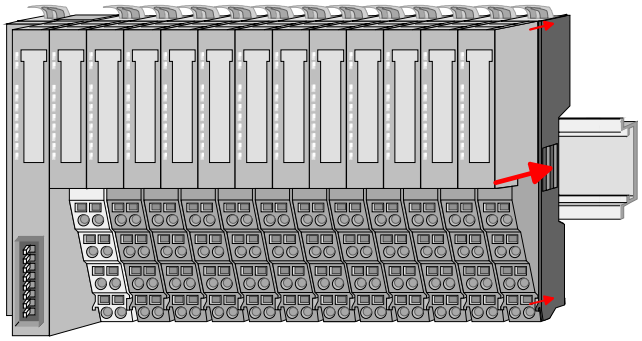
Mounting periphery module

- Mount the periphery modules you want.



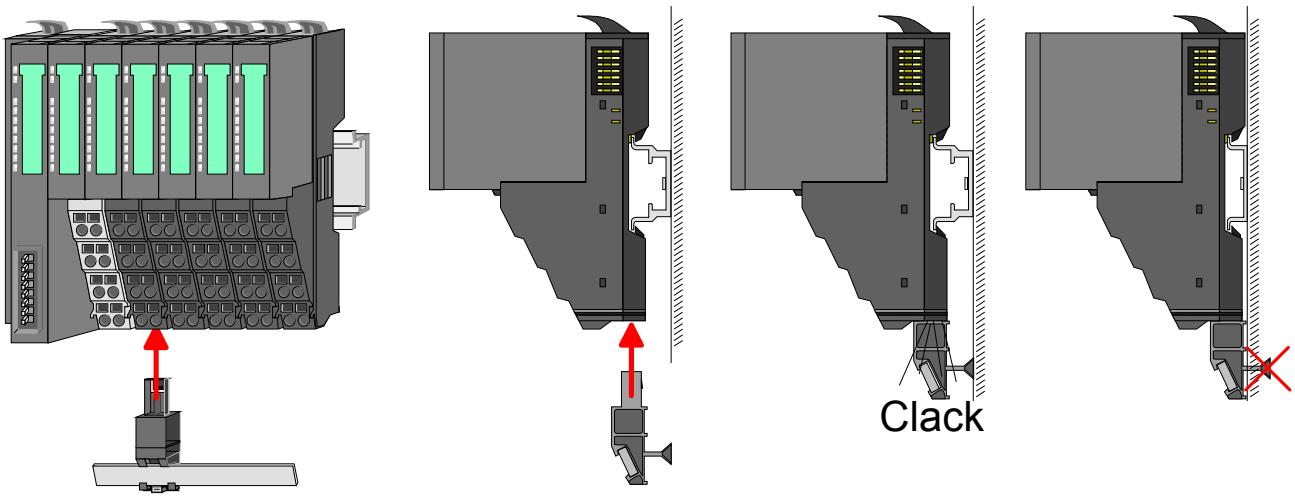
Mounting the bus cover

- After mounting the whole system, to protect the backplane bus connectors the bus cover may now be mounted at the last module



Mounting shield bus carrier

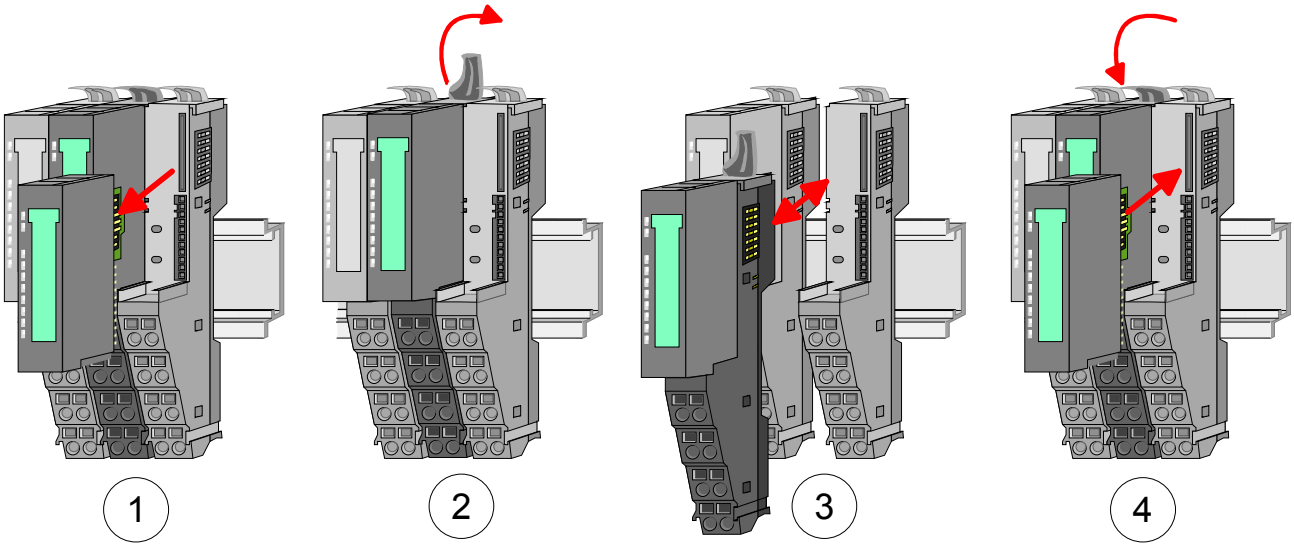
The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields. The shield bus carrier is mounted underneath the terminal of the terminal module. With a flat mounting rail for adaption to a flat mounting rail you may remove the spacer of the shield bus carrier.



Mounting between 2 modules

With the mounting of a SLIO module respectively of a group of SLIO modules between two modules for mounting reasons you have always to remove the electronic module of the just mounted right module. After that it may be plugged again.

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.



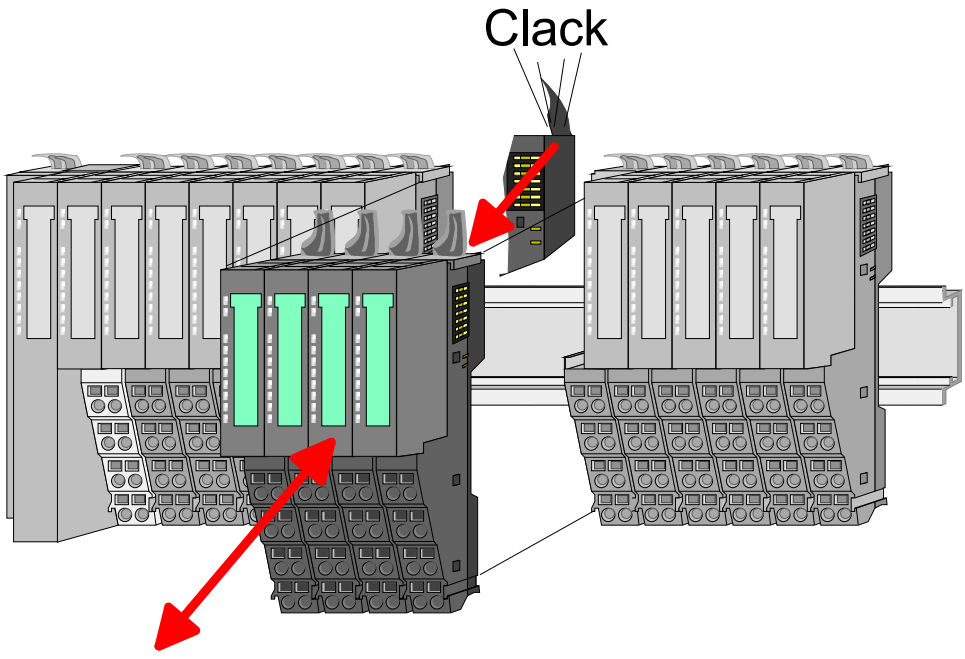
1 module group between 2 modules

With mounting respectively de-mounting of a module group you also have to remove the electronic module of the just mounted right module! After mounting it may be plugged again.

For mounting respectively de-mounting the locking lever of the modules of the block must be turned upwards.

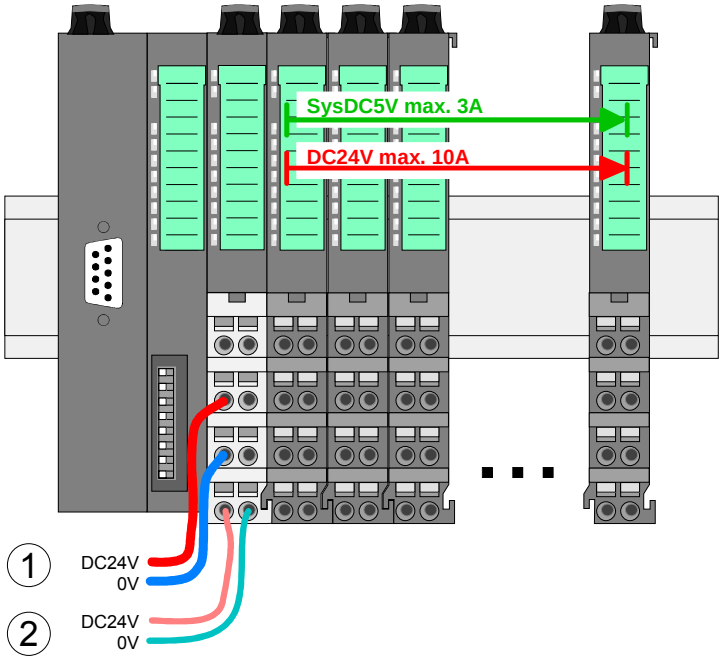
To mount the group of modules put them to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.

After mounting the block turn each locking lever of the modules downwards.



Wiring

Standard wiring



[1] DC 24V Power section supply I/O area

[2] DC 24V Electronic power supply bus coupler and I/O area

Note!

Power section and electronic power section supply are internally protected against higher voltage by fuses. The fuses are within the power module. If one fuse released, its electronic module must be exchanged!

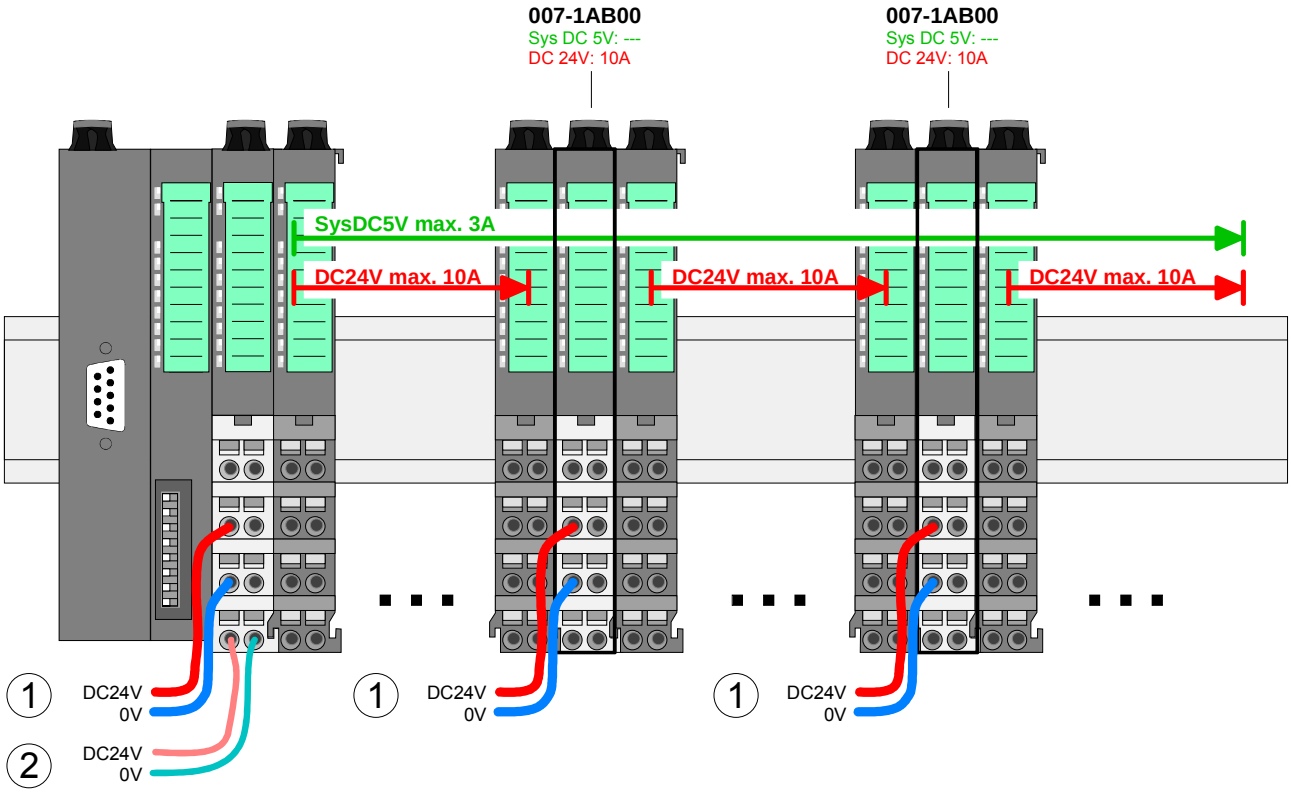
It is recommended to protect the power section supply with a fast 10A fuse and the electronic power supply with a fast 4A fuse.

Definition of isolated groups

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 following figure should illustrate the usage of the power modules.



[1] DC 24V Power section supply I/O area

[2] DC 24V Electronic power supply bus coupler and I/O area

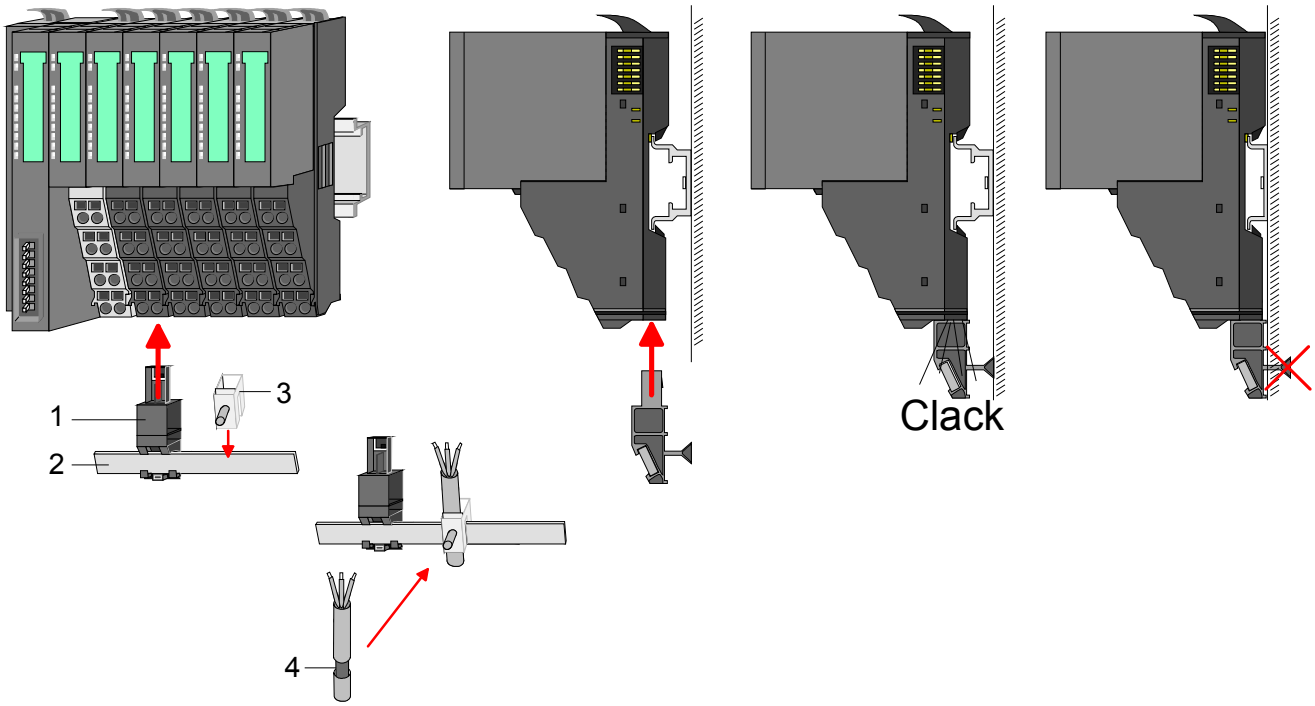
Shield attachment

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.

The shield bus carrier is mounted underneath the terminal of the terminal module. With a flat mounting rail for adaption to a flat mounting rail you may remove the spacer of the shield bus carrier.

After mounting the shield bus carrier with the shield bus, the cables with the accordingly stripped cable screen may be attached and fixed by the shield clamp.



[1] Shield bus carrier
[2] Shield bus
[3] Shield clamp
[4] Cable shield

Installation guidelines

General

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

What means EMC?

Electromagnetic digestibility (EMC) means the ability of an electrical device, to function error free in an electromagnetic environment without being interferenced res. without interferencing the environment.

All System SLIO components are developed for the deployment in hard industrial environments and fulfill 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:

- Fields
- I/O signal conductors
- Bus system
- Current 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.

One differs:

- 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 aluminum parts. Aluminum 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 res. 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 favorable.
  - 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 metalized plug cases for isolated data lines.
- In special use cases you should appoint special EMC actions.
  - Wire all inductivities with erase links, which are not addressed by the System SLIO modules.
  - For lightening cabinets you should prefer incandescent lamps and avoid luminescent lamps.
- 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 is a protection and functionality activity.
  - Connect installation parts and cabinets with the System SLIO in star topology with the isolation/protected earth conductor system. So you avoid ground loops.
  - If potential differences between installation parts and cabinets occur, 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. Hereby 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 res. µA) are transferred
  - foil isolations (static isolations) are used.
- With data lines always use metallic or metalized 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 the System SLIO module and **don't** lay it on there again!

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

General data

Conformity and approval		
Conformity		
CE	73/23/EWG	Low-voltage directive
Approval		
UL	UL 508	Approval for USA and Canada
others		
RoHs	-	Product is unleaded

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	EN 61131-2	-
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	EN 61131-2	0…+60°C
Vertical installation	EN 61131-2	0…+55°C
Air humidity	EN 60068-2-30	RH1 (without condensation, rel. humidity 10…95%)
Pollution	EN 61131-2	Degree of pollution 2
Mechanical		
Oscillation	EN 60068-2-6	1G
Shock	EN 60068-2-27	15G

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

EMC	**Standard**		**Comment**
Emitted interference	EN 61000-6-4		Class A (Industry area)
Noise immunity zone B	EN 61000-6-2		Industry area
		EN 61000-4-2	ESD Degree of severity 3, i.e. 8kV at air discharge, 4kV at contact discharge
		EN 61000-4-3	HF irradiation (casing) 80MHz … 1000MHz, 10V/m 80% AM (1kHz)
		EN 61000-4-6	HF conducted 150kHz … 80MHz, 10V/m 80% AM (1kHz)
		EN 61000-4-4	Burst, degree of severity 3
		EN 61000-4-5	Surge, degree of severity 3

Chapter 2 Hardware description

Overview

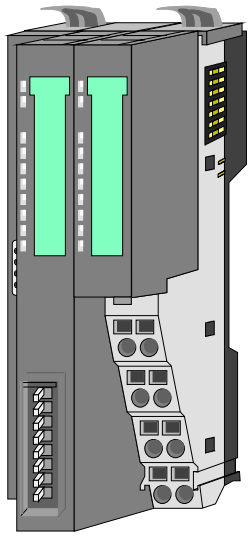
Here the hardware components of the IM 053-1CA00 are more described. You will find the technical data at the end of this chapter.

Content

Properties

Features

- 16 Rx and 16 Tx PDOs
- 2 SDOs
- Support of every transfer rates
- PDO linking
- PDO mapping: fix
- CAN bus address setting via DIP switch.

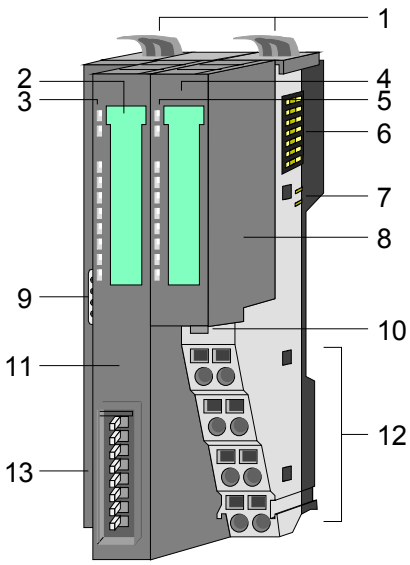


Order data

Type	Order number	Description
IM 053CAN	VIPA 053-1CA00	CAN slave for SLIO

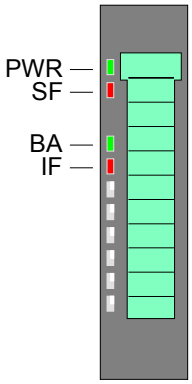
Structure

053-1CA00



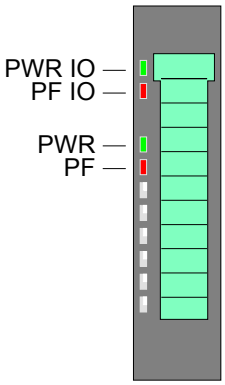
[1] Locking lever terminal module
[2] Labeling strip bus interface
[3] LED status indication bus interface
[4] Labeling strip power module
[5] LED status indication power module
[6] Backplane bus
[7] DC 24V power section supply
[8] Power module
[9] CAN plug bus interface
[10] Unlocking lever power module
[11] Bus interface
[12] Terminal
[13] Address selector

Status indication bus coupler



LED	Color	Description	
PWR	green	●	Bus coupler is power supplied
SF	red	●	Station fault, structure is not corresponding to the configuration
BA	green	●	Operation mode: operational (ready for data exchange)
		☼	Operation mode: pre-operational (waiting for parameters)
IF	red	●	Internal error occurred

Status indication power module

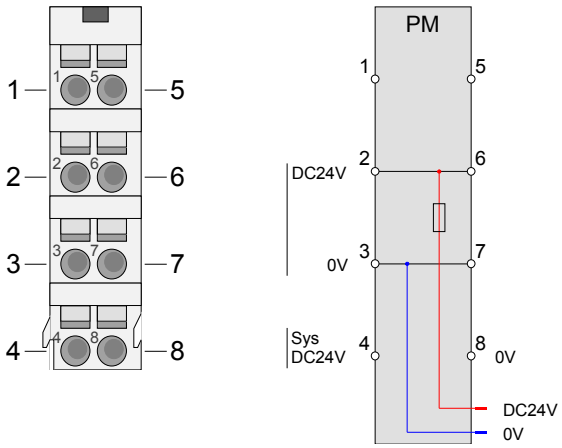


LED	Color	Description	
PWR IO	green	●	Power section supply OK
PF IO	red	●	Fuse power section supply defective (Power fail)
PWR	green	●	Electronic section supply OK
PF	red	●	Fuse electronic section supply defective

on: ● off: ○ blinking with 2Hz: ☼

Terminal

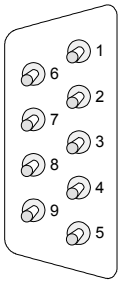
For wires with a core cross-section of 0.08mm^2 up to 2.5mm^2.



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

CAN plug bus coupler

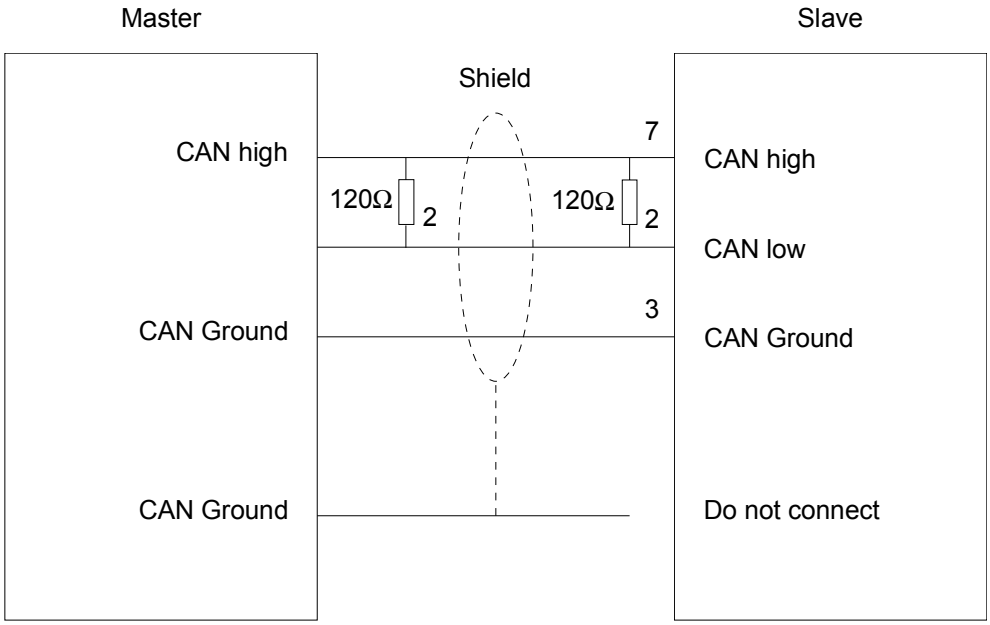


Pin	Assignment
1	not connected
2	CAN low
3	CAN Ground
4	not connected
5	not connected
6	not connected
7	CAN high
8	not connected
9	not connected

Bus wiring

The CAN bus communication medium is a screened three-core cable.

All stations on systems having more than two stations are wired in parallel. This means that the bus cable must be looped from station to station without interruptions.



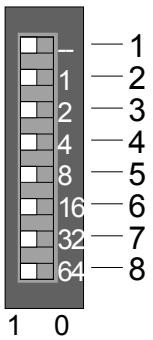
Note!

The end of the bus cable must be terminated with a 120Ω terminating resistor to prevent reflections and the associated communication errors!

Address selector

Valid address may range from 1 to 125. Addresses must be unique on the bus.

The slave address must have been preset before the bus coupler is turned on.



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

Technical Data

Electrical Data	VIPA 053-1CA00
No. of Modules	64
Profiles	DS301 / DS401
Power supply	DC 24V (20.4 ... 28.8V)
Current consumption Electronic power supply Power section supply	 max. 1.9A max. 10A
Current output Electronic power supply Power section supply	 max. 3A max. 10A
max no. of PDOs	16 Tx / 16 Rx PDOs
Data transfer rate	10kbit/s to 1Mbit/s
Connectors/interfaces	9pin D-type (socket)
Status indicator	by means of LEDs located on the front
Combination with peripheral modules	
max. no. of modules max. inputs/outputs	64 (depending on current consumption) 128byte each (128byte = 16 PDOs à 8byte)
Dimensions and weight	
Dimensions (WxHxD) in mm	48.5x109x76.5
Weight	155g

Chapter 3 Deployment

Overview

This chapter contains the description of the IM 053-1CA00 with CANopen.

Besides the fast introduction concerning the project engineering for "experts" you may find an introduction to the telegram structure and the function codes of CANopen.

The chapter is finished by the description of the Emergency Object as well as the Network Management NMT.

Content

Basics CAN

General

CANopen (**C**ontrol **A**rea **N**etwork) is an international standard for open fieldbus systems intended for building, manufacturing and process automation applications that was originally designed for automotive applications.

Due to its extensive error detection facilities, the CAN bus system is regarded as the most secure bus system. It has a residual error probability of less than $4.7x10^{-11}$. Bad messages are flagged and retransmitted automatically.

In contrast to Profibus and Interbus, CAN defines under the CAL-level-7-protocol (CAL=**C**AN **a**pplication **l**ayer) defines various level-7 user profiles for the CAN bus. One standard user profile defined by the CIA (**C**AN **i**n **A**utomation) e.V. is CANopen.

CANopen

CANopen is a user profile for industrial real-time systems, which is currently supported by a large number of manufacturers. CANopen was published under the heading of DS-301 by the CAN in Automation association (CIA). The communication specifications DS-301 define standards for CAN devices. These specifications mean that the equipment supplied by different manufacturers is interchangeable. The compatibility of the equipment is further enhanced by the equipment specification DS-401 that defines standards for the technical data and process data of the equipment. DS-401 contains the standards for digital and analog input/output modules.

CANopen comprises a communication profile that defines the objects that must be used for the transfer of certain data as well as the device profiles that specify the type of data that must be transferred by means of other objects.

The CANopen communication profile is based upon an object directory that is similar to the profile used by Profibus. The communication profile DS-301 defines two standard objects as well as a number of special objects:

- Process data objects (PDO)
  PDOs are used for real-time data transfers
- Service data objects (SDO)
  SDOs provide access to the object directory for read and write operations

Communication medium

CAN is based on a linear bus topology. You can use router nodes to construct a network. The number of devices per network is only limited by the performance of the bus driver modules.

The maximum distance covered by the network is determined by the runtimes of the signals. This means that a data rate of 1Mbit/s limits the network to 40m and 80kbit/s limits the network to 1000m.

The CAN bus communication medium employs a screened three-core cable (optionally a five-core).

The CAN bus operates by means of differential voltages. For this reason it is less sensitive to external interference than a pure voltage or current based interface. The network must be configured as a serial bus, which is terminated by a 120Ω terminating resistor.

Your VIPA CAN bus coupler contains a 9pin socket. You must use this socket to connect the CAN bus coupler as a slave directly to your CAN bus network.

All devices on the network use the same transfer rate.

Due to the bus structure of the network it is possible to connect or disconnect any station without interruption to the system. It is therefore also possible to commission a system in various stages. Extensions to the system do not affect the operational stations. Defective stations or new stations are recognized automatically.

Bus access method

Bus access methods are commonly divided into controlled (deterministic) and uncontrolled (random) bus access systems.

CAN employs a Carrier-Sense Multiple Access (CSMA) method, i.e. all stations have the same right to access the bus as long as the bus is not in use (random bus access).

Data communications is message related and not station related. Every message contains a unique identifier, which also defines the priority of the message. At any instance only one station can occupy the bus for a message.

CAN-Bus access control is performed by means of a collision-free, bit-based arbitration algorithm. Collision-free means that the final winner of the arbitration process does not have to repeat his message. The station with the highest priority is selected automatically when more than one station accesses the bus simultaneously. Any station that is has information to send will delay the transmission if it detects that the bus is occupied.

Fast introduction

Overview

This section is for experienced CANopen user that are already common with CAN. It will be shortly outlined, which messages are necessary for the deployment of the System SLIO with CAN in the start configuration.

Note!

Please regard that this manual prints the hexadecimal numbers in the type for developers "0x".

e.g.: **0x**15AE = 15AE**h**

Adjusting transfer rate and module-ID

Via the address selector you have to adjust a common transfer rate at the bus couplers as well as different node-IDs.

After starting your power supply, you program the transfer rate and the module-ID via 00 at the address selector within 20s.

For details to this see below under "Transfer rate and module ID".

CAN identifier

The CAN identifier for the in-/output data of the System SLIO are generated from the node addresses (1...125):

Kind of data	Default CAN identifier	Kind of data	Default CAN identifier
digital inputs 1 ... 64bit	0x180 + Node address	digital outputs 1 ... 64bit	0x200 + Node address
analog inputs 1 ... 4 words	0x280 + Node address	analog outputs 1 ... 4 Words/Channels	0x300 + Node address
other digital or analog inputs	0x380 + Node address	other digital or analog outputs	0x400 + Node address
	0x480 + Node address		0x500 + Node address
	More identifiers are blocked per default and may be activated via SDO Telegram.		

Digital in-/outputs

The CAN messages with digital input data are represented as follows:

Identifier 0x180+Node address + up to 8byte user data

Identifier 11Bit	**DI 0** 8Bit	**DI 1** 8Bit	**DI 2** 8Bit	**...**	**DI 7** 8Bit

The CAN messages with digital output data are represented as follows:

Identifier 0x200+Node address + up to 8byte user data

Identifier 11Bit	**DO 0** 8Bit	**DO 1** 8Bit	**DO 3** 8Bit	**...**	**DO 7** Bit

Analog in-/outputs

The CAN messages with analog input data are represented as follows::

Identifier 0x280+Node address + up to 4Words user data

Identifier 11Bit	**AI 0** 1Word	**AI 1** 1Word	**AI 2** 1Word	**AI 3** 1Word

The CAN messages with analog output data are represented as follows:

Identifier 0x300+Node address + up to 4Words user data

Identifier 11Bit	**AI 0** 1Word	**AI 1** 1Word	**AI 2** 1Word	**AI 3** 1Word

Node Guarding

For the System SLIO works per default in event-controlled mode (no cyclic DataExchange), a node failure is not always immediately detected. Remedy is the control of the nodes per cyclic state request (Node Guarding).

You request cyclically a state telegram via Remote-Transmit-Request (RTR): the telegram only consists of a 11bit identifier:

Identifier 0x700+Node address

Identifier 11Bit

The System SLIO node answers with a telegram that contains one state byte:

Identifier 0x700+Node address + State byte

Identifier 11Bit	**Status** 8Bit

Bit 0 ... 6: Node state
- 0x7F: Pre-Operational
- 0x05: Operational
- 0x04: Stopped res. Prepared

Bit 7: Toggle bit, toggles after every send

To enable the bus coupler to recognize a network master failure (watchdog function), you still have to set the Guard-Time (Object 0x100C) and the Life-Time-Factor (Object 0x100D) to values≠0.

(reaction time at failure: Guard-Time x Life Time Factor).

Heartbeat

Besides the Node Guarding, the System SLIO CANopen coupler also supports the Heartbeat Mode.

If there is a value set in the index 0x1017 (Heartbeat Producer Time), the device state (Operational, Pre-Operational, ...) is transferred when the Heartbeat-Timer run out by using the COB identifier (0x700+Module-Id):

Identifier 0x700+Node address + State byte

Identifier 11Bit	**Status** 8Bit

The Heartbeat Mode starts automatically as soon as there is a value in index 0x1017 higher 0.

Emergency Object

To send internal device failures to other participants at the CAN bus with a high priority, the SLIO CAN bus coupler supports the Emergency Object.

To activate the emergency telegram, you need the **COB-Identifier** that is fixed after boot-up in the object directory of the variable 0x1014in hexadecimal view: **0x80 + Module-ID.**

The emergency telegram has always a length of 8byte. It consists of:

Identifier 0x80 + Node address + 8byte user data

Identifier 11Bit	**EC0**	**EC1**	**Ereg**	**Inf0**	**Inf1**	**Inf2**	**Inf3**	**Inf4**

Error Code	Meaning	Info 0	Info 1	Info 2	Info 3	Info4
0x0000	Reset Emergency	0x00	0x00	0x00	0x00	0x00
0x8100	Heartbeat Consumer	Node ID	LowByte Timer Value	HighByte Timer Value	0x00	0x00
0x8100	SDO Block Transfer	0xF1	LowByte Index	HighByte Index	SubIndex	0x00
0x8130	Node Guarding Error	LowByte GuardTime	HighByte GuardTime	LifeTime	0x00	0x00
0x8210	PDO not processed due to length error	PDO Number	Wrong length	PDO length	0x00	0x00
0x8220	PDO length exceeded	PDO Number	Wrong length	PDO length	0x00	0x00

Note!

The now described telegrams enable you to start and stop the System SLIO, read inputs, write outputs and control the modules.

In the following, the functions are described in detail.

Transfer rate and module-ID

Overview

There is the possibility to specify *transfer rate* and *module-ID* by means of the address selector.

The settings are permanently stored in an EEPROM and may any time be changed. Per default the bus coupler has a transfer rate of 500kbit/s.

Proceeding

- Turn off the power supply of the bus coupler
- Set the address selector to 0.
- Turn on the power supply of the bus coupler.
  The LEDs SF, BA and IF are blinking. Now for each there is a period of 10s to select *transfer rate* and *module-ID*.

Hinweis!

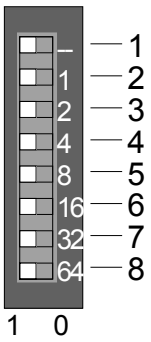
If the address selector is still 0 after 20s, the following values are set:

- Transfer rate: 1Mbit/s
- Module-ID: 0 (not valid)

To change these settings start with the "Proceeding" again.

Programming transfer rate

As long as the LEDs SF, BA and IF are blinking, select the transfer rate with the address selector. There are the following possibilities for setting:

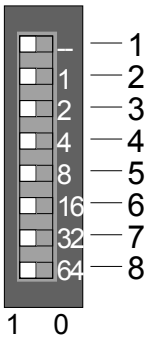


Pos.	Value	Transfer rate kbit/s								
		1000	800	500	250	125	100	50	20	10
1	---	---	---	---	---	---	---	---	---	---
2	1	0	0	1	0	1	0	1	0	1
3	2	0	0	0	1	1	0	0	1	1
4	4	0	0	0	0	0	1	1	1	1
5	8	0	1	0	0	0	0	0	0	0
6	16	0	0	0	0	0	0	0	0	0
7	32	0	0	0	0	0	0	0	0	0
8	64	0	0	0	0	0	0	0	0	0
		25	50	100	250	500	600	1000	2500	5000
		max guarantied bus distance in m								

After 10s the selected transfer rate is stored in the EEPROM and the IF LED gets off.

Programming module-ID

For a further period of 10s, as long as the LEDs SF and BA are blinking, you may set the *module-ID* in a range of 1 ... 125 by means of the address selector. There are the following possibilities for setting:



Pos.	Value	Example State	Address
1	---	---	1+2+32=35 Address: 35
2	1	1	
3	2	1	
4	4	0	
5	8	0	
6	16	0	
7	32	1	
8	64	0	

After 10s the selected module-ID is taken and the bus coupler the bus coupler returns to the normal operating mode (status: "Pre-Operational") with the set values.

Note!

Each module-ID may only exist once at the bus!

The module-ID must be set before the bus coupler is powered on!

Transfer rate selection by SDO-write

You can also modify the CAN transfer rate by means of an SDO-Write operation to the object "0x2001". The entered value is used as the CAN transfer rate when the bus coupler has been RESET. This method is a most convenient when you must change the CAN transfer rate of all the bus couplers of a system from a central CAN terminal. The bus couplers use the programmed transfer rate when the system has been RESET.

Message structure

Identifier

All CANopen messages have the following structure according to CiA DS-301:

Identifier

Byte	Bit 7 ... Bit 0
1	Bit 3 ... Bit 0: most significant 4 bits of the module-ID Bit 7 ... Bit 4: CANopen function code
2	Bit 3 ... Bit 0: data length code (DLC) Bit 4: RTR-Bit: 0: no data (request code) 1: data available Bit 7 ... Bit 5: Least significant 3 bits of the module-ID

Data

Data

Byte	Bit 7 ... Bit 0
3 ... 10	Data

An additional division of the 2 byte identifier into function portion and a module-ID gives the difference between this and a level 2 message. The function determines the type of message (object) and the module-ID addresses the receiver.

CANopen devices exchange data in the form of objects. The CANopen communication profile defines two different object types as well as a number of special objects.

The SLIO CAN bus coupler supports the following objects:

- 16 transmit PDOs (PDO Linking, PDO Mapping)
- 16 receive PDOs (PDO Linking, PDO Mapping)
- 2 standard SDOs
- 1 emergency object
- 1 network management object NMT
- Node Guarding
- Heartbeat

CANopen function codes

Every object is associated with a function code. You can obtain the required function code from the following table:

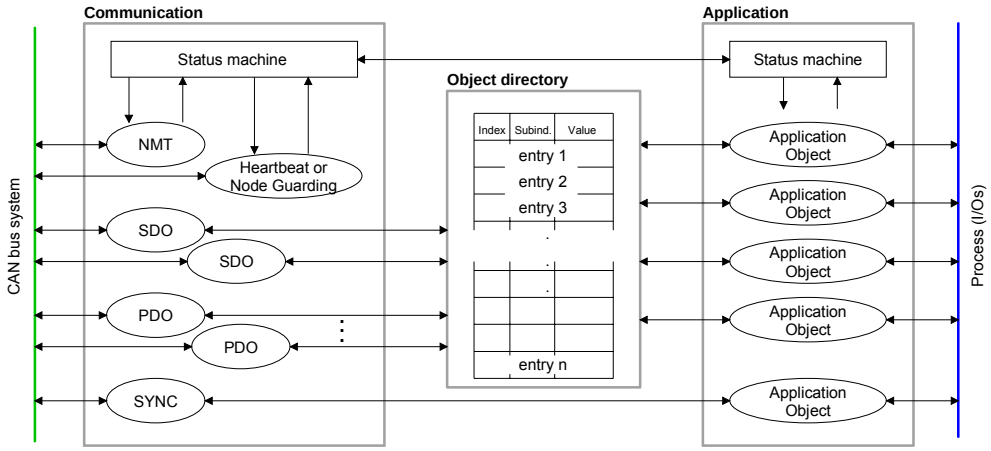
Object	Function code (4 bits)	Receiver	Definition	Function
NMT	0000	Broadcast	CiA DS-301	Network management
EMERGENCY	0001	Master	CiA DS-301	Error message
PDO1S2M	0011	Master, Slave (RTR)	CiA DS-301	Digital input data 1
PDO1M2S	0100	Slave	CiA DS-301	Digital output data 1
SDO1S2M	1011	Master	CiA DS-301	Configuration data
SDO1M2S	1100	Slave	CiA DS-301	Configuration data
Node Guarding	1110	Master, Slave (RTR)	CiA DS-301	Module monitoring
Heartbeat	1110	Master, Slave	Application spec.	Module monitoring

Note!

A detailed description of the structure and the contents of these objects is available in "CiA Communication Profile DS-301 Version 3.0" and "CiA Device Profile for I/O-Modules DS-401 Version 1.4".

Structure of the device model

A CANopen device can be structured as follows:



Communication

Serves the communication data objects and the concerning functionality for data transfer via the CANopen network.

Application

The application data objects contain e.g. in- and output data. In case of an error, an application status machine switches the outputs in a secure state.

The object directory is organized as 2 dimension table. The data is addressed via index and sub-index.

Object directory

This object directory contains all data objects (application data + parameters) that are accessible and that influence the behavior of communication, application and status machines.

PDO

PDO

In many fieldbus systems the whole process image is transferred - mostly more or less cyclically. CANopen is not limited to this communication principle, for CAN supports more possibilities through multi master bus access coordination.

CANopen divides the process data into segments of max. 8byte. These segments are called **p**rocess **d**ata **o**bjects (PDOs). Every PDO represents one CAN telegram and is identified and prioritized via its specific CAN identifier.

For the exchange of process data, the SLIO CAN bus coupler supports 32 PDOs. Every PDO consists of a maximum of 8 data bytes. The transfer of PDOs is not verified by means of acknowledgments since the CAN protocol guarantees the transfer.

There are 16 Tx transmit PDOs for input data and 16 Rx receive PDOs for output data. The PDOs are named seen from the bus coupler:

Receive PDOs (RxPDOs) are received by the bus coupler and contain output data.

Transmit PDOs (TxPDOs) are send by the bus coupler and contain input data.

The assignment of the PDOs to input or output data occurs automatically.

Variable PDO mapping

CANopen predefines the first two PDOs in the device profile. The assignment of the PDOs is fixed in the mapping tables in the object directory. The mapping tables are the cross-reference between the application data in the object directory and the sequence in the PDOs.

The assignment of the PDOs, automatically created by the coupler, are commonly adequate. For special applications, the assignment may be changed. Herefore you have to configure the mapping tables accordingly.

First, you write a 0 to sub-index 0 (deactivates the current mapping configuration). Then you insert the wanted application objects into sub-index 1...8. Finally you parameterize the number of now valid entries in sub-index 0 and the coupler checks the entries for their consistency.

PDO identifier COB-ID

The most important communication parameter of a PDOs is the CAN identifier (also called "Communication Object Identifier", COB-ID). It serves the identification of the data and sets the priority of bus access.

For every CAN data telegram only one sending node may exist (producer). Due to the ability of CAN to send all messages per broadcast procedure, however, a telegram may be received by several bus participants at the same time (consumer). Therefore, one node may deliver its input information to different bus stations similarly - without needing the pass through a logical bus master.

The System SLIO provides receive and transmit PDOs default identifier in dependence of the node address.

Below follows a list of the COB identifiers for the receive and the transmit PDO transfer that are pre-set after boot-up.

The transmission type in the object directory (indices 0x1400-0x140F and 0x1800-0x180F, sub-index 0x02) is preset to asynchronous, event controlled (= 0xFF). The EVENT-timer (value * 1ms) can be used to transmit the PDOs cyclically.

Send:	0x180 + module-ID: PDO1S2M digital	(acc. DS-301)
	0x280 + module-ID: PDO2S2M analog	
	0x380 + module-ID: PDO3S2M digital or analog	
	0x480 + module-ID: PDO4S2M	

Receive:	0x200 + module-ID: PDO1M2S digital	(acc. DS-301)
	0x300 + module-ID: PDO2M2S analog	
	0x400 + module-ID: PDO3M2S digital or analog	
	0x500 + module-ID: PDO4M2S	

Note!

The PDOs 5 ... 16 are blocked per default and must be enabled by SDO telegrams. More information concerning this may be found in the object directory 1400 ... 1410 / 1800 ... 1810.

PDO linking

If the Consumer-Producer model of the CANopen PDOs shall be used for direct data transfer between nodes (without master), you have to adjust the identifier distribution accordingly, so that the TxPDO identifier of the producer is identical with the RxPDO identifier of the consumer:

This procedure is called PDO linking. this enables for example the simple installation of electronic gearing where several slave axis are listening to the actual value in TxPDO of the master axis.

PDO Communication types

CANopen supports the following possibilities for the process data transfer:

- Event triggered
- Polled
- Synchronized

Event triggered

The "event" is the alteration of an input value, the data is send immediately after value change. The event control makes the best use of the bus width for not the whole process image is send but only the changed values. At the same time, a short reaction time is achieved, because there is no need to wait for a master request.

Polled

PDOs may also be polled via data request telegrams (remote frames) to give you the opportunity to e.g. send the input process image of event triggered inputs to the bus without input change for example a monitoring or diagnosis device included during runtime.

The SLIO CANopen bus coupler supports the query of PDOs via remote frames - for this can, due to the hardware, not be granted for all CANopen devices, this communication type is only partially recommended.

Synchronized

It is not only convenient for drive applications to synchronize the input information request and the output setting. For this purpose, CANopen provides the SYNC object, a CAN telegram with high priority and no user data which receipt is used by the synchronized nodes as trigger for reading of the inputs res. writing of the outputs.

PDO transmission type

The parameter "PDO transmission type" fixes how the sending of the PDOs is initialized and what to do with received ones:

Transmission Type	Cyclical	Acyclical	Synchronous	Asynchronous
0		x	x	
1-240	x		x	
254,255				x

Synchronous

The transmission type 0 is only wise for RxPDOs: the PDO is analyzed at receipt of the next SYNC telegram.

At transmission type 1-240, the PDO is send res. expected cyclically: after every "n^{th}" SYNC (n=1...240). For the transmission type may not only be combined within the network but also with a bus, you may thus e.g. adjust a fast cycle for digital inputs (n=1), while data of the analog inputs is transferred in a slower cycle (e.g. n=10). The cycle time (SYNC rate) may be monitored (Object 0x1006), at SYNC failure, the coupler sets its outputs in error state.

Asynchronous

The transmission types 254 + 255 are asynchronous or also event triggered. The transmission type 254 provides an event defined by the manufacturer, at 255 it is fixed by the device profile.

When choosing the event triggered PDO communication you should keep in mind that in certain circumstances there may occur a lot of events similarly. This may cause according delay times for sending PDOs with lower priority values.

You should also avoid to block the bus by assigning a high PDO priority to an often alternating input ("babbling idiot").

Inhibit time

Via the parameter "inhibit time" a "send filter" may be activated that does not lengthen the reaction time of the relatively first input alteration but that is active for the following changes.

The inhibit time (send delay time) describes the min. time span that has to pass between the sending of two identical telegrams.

When you use the inhibit time, you may ascertain the max. bus load and for this the latent time in the "worst case".

SDO

SDO

The **S**ervice **D**ata **O**bject (SDO) serves the read or write access to the object directory. The CAL layer 7 protocol gives you the specification of the Multiplexed-Domain-Transfer-Protocol that is used by the SDOs. This protocol allows you to transfer data of any length because where appropriate, messages are distributed to several CAN messages with the same identifier (segment building).

The first CAN message of the SDO contain process information in 4 of the 8 bytes. For access to object directory entries with up to 4byte length, one single CAN message is sufficient. The following segments of the SDO contain up to 7byte user data. The last byte contains an end sign. A SDO is delivered with acknowledgement, i.e. every reception of a message is receipted.

The COB identifiers for read and write access are:

- Receive-SDO1: 0x600 + Module-ID
- Transmit-SDO1: 0x580 + Module-ID

Note!

A detailed description of the SDO telegrams is to find in the DS-301 norm from CiA.

In the following only the error messages are described that are generated at wrong parameterization.

SDO error codes

Code	Error
0x05030000	Toggle bit not alternated
0x05040000	SDO protocol timed out
0x05040001	Client/server command specifier not valid or unknown
0x05040002	Invalid block size (block mode only)
0x05040003	Invalid sequence number (block mode only)
0x05040004	CRC error (block mode only)
0x05040005	Out of memory
0x06010000	Unsupported access to an object
0x06010001	Attempt to read a write only object
0x06010002	Attempt to write a read only object
0x06020000	Object does not exist in the object dictionary
0x06040041	Object cannot be mapped to the PDO
0x06040042	The number and length of the objects to be mapped would exceed PDO length
0x06040043	General parameter incompatibility reason
0x06040047	General internal incompatibility in the device
0x06060000	Access failed due to an hardware error
0x06070010	Data type does not match, length of service parameter does not match
0x06070012	Data type does not match, length of service parameter too high
0x06070013	Data type does not match, length of service parameter too low
0x06090011	Sub-index does not exist
0x06090030	Value range of parameter exceeded (only for write access)
0x06090031	Value of parameter written too high
0x06090032	Value of parameter written too low
0x06090036	Maximum value is less than minimum value
0x08000000	general error
0x08000020	Data cannot be transferred or stored to the application
0x08000021	Data cannot be transferred or stored to the application because of local control
0x08000022	Data cannot be transferred or stored to the application because of the present device state
0x08000023	Object directory dynamic generation fails or no object directory is present (e.g. object directory is generated from file and generation fails because of an file error)

Object directory

Structure

The CANopen object directory contains all relevant CANopen objects for the bus coupler. Every entry in the object directory is marked by a 16bit index.

If an object exists of several components (e.g. object type Array or Record), the components are marked via an 8bit sub-index.

The object name describes its function. The data type attribute specifies the data type of the entry.

The access attribute defines, if the entry may only be read, only be written or read and written.

The object directory is divided into the following 3 parts:

Communication specific profile area (0x1000 – 0x1FFF)

This area contains the description of all relevant parameters for the communication.

0x1000 – 0x1018	General communication specific parameters (e.g. device name)
0x1400 – 0x140F	Communication parameters (e.g. identifier) of the receive PDOs
0x1600 – 0x160F	Mapping parameters of the receive PDOs The mapping parameters contain the cross-references to the application objects that are mapped into the PDOs and the data width of the depending object.
0x1800 – 0x180F 0x1A00 – 0x1A0F	Communication and mapping parameters of the transmit PDOs

Manufacturer specific profile area (0x2000 – 0x5FFF)

Here you may find the manufacturer specific entries like e.g. PDO Control, CAN transfer rate (transfer rate after RESET) etc.

Standardized device profile area (0x6000 – 0x9FFF)

This area contains the objects for the device profile acc. DS-401.

Note!

For the CiA norms are exclusively available in English, we adapted the object tables. Some entries are described below the according tables.

Object directory overview

Index	Content of Object
0x1000	Device type
0x1001	Error register
0x1003	Error store
0x1004	Number of PDOs
0x1005	SYNC identifier
0x1006	SYNC interval
0x1007	Synchronous Window Length
0x1008	Device name
0x1009	Hardware version
0x100A	Software version
0x100B	Node number
0x100C	Guard time
0x100D	Life time factor
0x100E	Node Guarding Identifier
0x1010	Save parameter
0x1011	Load parameter
0x1014	Emergency COB-ID
0x1016	Heartbeat consumer time
0x1017	Heartbeat producer time
0x1018	Device identification
0x1027	Module list
0x1029	Error behavior
0x1400 - 0x140F	Communication parameter for receive PDOs (RxPDO, Master to Slave)
0x1600 - 0x160F	Mapping parameter for receive PDOs (RxPDO)
0x1800 - 0x180F	Communication parameter for transmit PDOs (TxPDO, Slave to Master)
0x1A00 - 0x1A0F	Mapping parameter for transmit PDOs (TxPDO)
0x2001	CAN transfer rate
0x2400	PDO Control
0x3100 - 0x31FF	Module Parameterization
0x3200	Access to record set bus coupler
0x3201 - 0x3240	Access to record set function modules
0x5000	Digital-Input-2/4-bit Array (see DS 401)
0x5002	Polarity Digital-Input-2/4-bit Array (see DS 401)
0x5200	Digital-Output-2/4-bit Array (see DS 401)
0x5202	Polarity Digital-Output-2/4-bit Array (see DS 401)
0x5206	Fault Mode Digital-Output-2/4-bit Array (see DS 401)
0x5207	Fault State Digital-Output-2/4-bit Array (see DS 401)
0x6000	Digital-Input-8-bit Array (see DS 401)
0x6002	Polarity Digital-Input-8-bit Array (see DS 401)
0x6200	Digital-Output-8-bit Array (see DS 401)
0x6202	Polarity Digital-Output-8-bit Array (see DS 401)
0x6206	Fault Mode Digital-Output-8-bit Array (see DS 401)
0x6207	Fault State Digital-Output-8-bit Array (see DS 401)
0x6401	Analog-Input Array (see DS 401)
0x6411	Analog-Output Array (see DS 401)
0x6421	Analog-Input Interrupt Trigger Array (see DS 401)
0x6422	Analog-Input Interrupt Source Array (see DS 401)
0x6423	Analog-Input Interrupt Enable (see DS 401)
0x6424	Analog-Input Interrupt Upper Limit Array (see DS 401)
0x6425	Analog-Input Interrupt Lower Limit Array (see DS 401)
0x6426	Analog-Input Interrupt Delta Limit Array (see DS 401)
0x6443	Fault Mode Analog-Output Array (see DS 401)
0x6444	Fault State Analog-Output Array (see DS 401)

Device Type

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1000	0	Device Type	Unsigned32	ro	N	0x00050191	Statement of device type

The 32bit value is divided into two 16bit fields:

MSB	LSB
Additional information device	**Profile number**
0000 0000 0000 wxyz (bit)	401dec=0x0191

The "additional information" contains data related to the signal types of the I/O device:

z=1 → digital inputs

y=1 → digital outputs

x=1 → analog inputs

w=1 → analog outputs

Error register

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1001	0	Error Register	Unsigned8	ro	Y	0x00	Error register

Bit7							**Bit0**
ManSpec	reserved	reserved	Comm.	reserved	reserved	reserved	Generic

ManSpec.: Manufacturer specific error, specified in object 0x1003.

Comm.: Communication error (overrun CAN)

Generic: A not more precisely specified error occurred (flag is set at every error message)

Error store

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1003	0	Predefined error field (error store)	Unsigned8	ro	N	0x00	Object 0x1003 contains a description of the error that has occurred in the device - sub-index 0 has the number of error states stored
	1	Actual error	Unsigned32	ro	N		Last error state to have occurred
	...	...	...	...	...	...	...
	254		Unsigned32	ro	N		A maximum of 254 error states

The "predefined error field" is divided into two 16bit fields:

MSB	LSB
Additional information	**Error code**

The additional code contains the error trigger (see emergency object) and thereby a detailed error description.

New errors are always saved at sub-index 1, all the other sub-indices being appropriately incremented.

By writing a "0" to sub-index 0, the whole error memory is cleared. If there has not been an error since PowerOn, then object 0x1003 exists only of sub-index 0 with entry "0".

Via reset or PowerCycle, the error memory is cleared.

Number of PDOs

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1004	0	Number of PDOs supported	Unsigned32	ro	N	0x000A000A	Number of PDOs supported
	1	Number of synchronous PDOs supported	Unsigned32	ro	N	0x000A000A	Number of synchronous PDOs supported
	2	Number of asynchronous PDOs supported	Unsigned32	ro	N	0x000A000A	Number of asynchronous PDOs supported

The 32bit value is divided into two 16bit fields:

MSB	LSB
Number of receive (Rx)PDOs supported	**Number of send (Tx)PDOs supported**

SYNC identifier

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1005	0	COB-Id sync message	Unsigned32	ro	N	0x80000080	Identifier of the SYNC message

The lower 11bit of the 32bit value contain the identifier (0x80=128dez), while the MSBit indicates whether the device receives the SYNC telegram (1) or not (0).

Attention: In contrast to the PDO identifiers, the MSB being set indicates that this identifier is relevant for the node.

SYNC interval

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1006	0	Communication cycle period	Unsigned32	rw	N	0x00000000	Maximum length of the SYNC interval in µs.

If a value other than zero is entered here, the coupler goes into error state if no SYNC telegram is received within the set time during synchronous PDO operation.

Synchronous Window Length

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1007	0	Synchronous window length	Unsigned32	rw	N	0x00000000	Contains the length of time window for synchronous PDOs in µs.

Device name

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1008	0	Manufacturer device name	Visible string	ro	N		Device name of the bus coupler

VIPA IM 053 1CA00 = VIPA CANopen slave IM 053-1CA00

Since the returned value is longer than 4byte, the segmented SDO protocol is used for transmission.

Hardware version

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1009	0	Manufacturer Hardware version	Visible string	ro	N		Hardware version number of bus coupler

VIPA IM 053 1CA00 = 1.00

Since the returned value is longer than 4byte, the segmented SDO protocol is used for transmission.

Software version

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100A	0	Manufacturer Software version	Visible string	ro	N		Software version number CANopen software

VIPA IM 053 1CA00 = 1.xx

Since the returned value is longer than 4byte, the segmented SDO protocol is used for transmission.

Node number

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100B	0	Node ID	Unsigned32	ro	N	0x00000000	Node number

The node number is supported for reasons of compatibility.

Guard time

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100C	0	Guard time [ms]	Unsigned16	rw	N	0x0000	Interval between two guard telegrams. Is set by the NMT master or configuration tool.

Life time factor

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100D	0	Life time factor	Unsigned8	rw	N	0x00	Life time factor x guard time = life time (watchdog for life guarding)

If a guarding telegram is not received within the life time, the node enters the error state. If the life time factor and/or guard time =0, the node does not carry out any life guarding, but can itself be monitored by the master (node guarding).

Guarding identifier

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100E	0	COB-ID Guarding Protocol	Unsigned32	ro	N	0x000007xy, xy = node ID	Identifier of the guarding protocol

Save parameters

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1010	0	Store Parameter	Unsigned8	ro	N	0x01	Number of store Options
	1	Store all parameters	Unsigned32	ro	rw	0x01	Stores all (storable) Parameters

By writing the string "save" in ASCII code (hex code: 0x65766173) into sub-index 1, the current parameters are placed into non-volatile storage (byte sequence at the bus incl. SDO protocol: 0x23 0x10 0x10 0x01 0x73 0x61 0x76 0x65).

If successful, the storage process is confirmed by the corresponding TxSDO (0x60 in the first byte).

Note!

For the bus coupler is not able to send or receive CAN telegrams during the storage procedure, storage is only possible when the node is in pre-operational state.

It is recommended to set the complete net to the pre-operational state before storing data to avoid a buffer overrun.

Load default values

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1011	0	Restore parameters	Unsigned8	ro	N	0x01	Number of reset options
	1	Restore all parameters	Unsigned32	rw	N	0x01	Resets all parameters to their default values

By writing the string "load" in ASCII code (hex code: 0x64616F6C) into sub-index 1, all parameters are set back to default values (delivery state) **at next start-up (reset)** (byte sequence at the bus incl. SDO protocol: 0x23 0x11 0x10 0x01 0x6C 0x6F 0x61 0x64).

This activates the default identifiers for the PDOs.

Emergency COB-ID

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1014	0	COB-ID Emergency	Unsigned32	ro	N	0x00000080 + Node_ID	Identifier of the emergency telegram

Consumer heartbeat time

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1016	0	Consumer heartbeat time	Unsigned8	ro	N	0x05	Number of entries
	1		Unsigned32	rw	N	0x00000000	Consumer heartb. time 1
	2		Unsigned32	rw	N	0x00000000	Consumer heartb. time 2
	3		Unsigned32	rw	N	0x00000000	Consumer heartb. time 3
	4		Unsigned32	rw	N	0x00000000	Consumer heartb. time 4
	5		Unsigned32	rw	N	0x00000000	Consumer heartb. time 5

Structure of the "Consumer Heartbeat Time" entry:

Bits	31-24	23-16	15-0
Value	Reserved	Node-ID	Heartbeat time
Encoded as	Unsigned8	Unsigned8	Unsigned16

As soon as you try to configure a consumer heartbeat time unequal zero for the same node-ID, the node interrupts the SDO download and throws the error code 0604 0043hex.

Producer heartbeat time

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1017	0	Producer heartbeat time	Unsigned16	rw	N	0x0000	Defines the cycle time of heartbeat in ms

Identity Object

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1018	0	Identity Object	Unsigned8	ro	N	0x04	Contains general information about the device (number of entries)
	1	Vendor ID	Unsigned32	ro	N	0xAFFEAFFE	Vendor ID
	2	Product Code	Unsigned32	ro	N	*	Product Code
	3	Revision Number	Unsigned32	ro	N		Revision Number
	4	Serial Number	Unsigned32	ro	N		Serial Number

*) Default value Product Code 053-1CA00: 0x0531CA00

Modular Devices

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1027	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Unsigned16	ro	N		Identification number of Module 1
	...	...	...	...	...	...	...
	N	Module N	Unsigned16	ro	N		Identification number of Module N

The *Identification number* corresponds to the first 4 digits of the *module ID*. The *module ID* may be found at the technical data of the corresponding SLIO module.

Error Behavior

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1029	0	Error behavior	Unsigned8	ro	N	0x02	Number of Error Classes
	1	Communication Error	Unsigned8	ro	N	0x00	Communication Error
	2	Manufacturer specific error	Unsigned8	ro	N	0x00	Manufacturer specific error

As soon as a device failure is detected in "operational" state, the module should automatically change into the "pre-operational" state.

If e.g. an "Error behavior" is implemented, the module may be configured that its going into STOP at errors.

The following error classes may be monitored:

0 = pre-operational

1 = no state change

2 = stopped

3 = reset after 2 seconds

Communication parameter RxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1400	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000200 + NODE_ID	COB-ID RxPDO1
	2	Transmis-sion type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Sub-index 1 (COB-ID): The lower 11bit of the 32bit value (bits 0-10) contain the CAN identifier, the MSBit (bit 31) shows if the PDO is active (1) or not(0), bit 30 shows if a RTR access to this PDO is permitted (0) or not (1).

The sub-index 2 contains the transmission type.

Communication parameter RxPDO2

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1401	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000300 + NODE_ID	COB-ID RxPDO2
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO3

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1402	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000400 + NODE_ID	COB-ID RxPDO3
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO4

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1403	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000500 + NODE_ID	COB-ID RxPDO4
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO5

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1404	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO5
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO6

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1405	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO6
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO7

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1406	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO7
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO8

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1407	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO8
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO9

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1408	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO9
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO10

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1409	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD10
	2	transm. type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO11

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140A	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD11
	2	Transmis-sion type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO12

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140B	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD12
	2	Transmis-sion type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO13

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140C	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD13
	2	Transmis-sion type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO14

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140D	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD14
	2	Transmis-sion type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO15

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140E	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD15
	2	Transmis-sion type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Communication parameter RxPDO16

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140F	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD16
	2	Transmis-sion type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Mapping RxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1600	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the first receive PDO; sub-index 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x62000108	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x62000208	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped	Unsigned32	rw	N	0x62000808	(2 byte index, 1 byte sub-index, 1 byte bit-width)

The first receive PDO (RxPDO1) is per default for the digital outputs. Depending on the number of the inserted outputs, the needed length of the PDO is calculated and mapped into the according objects.

For the digital outputs are organized in bytes, the length of the PDO can be directly seen in sub-index 0.

If the mapping is changed, the entry in sub-index 0 has to be adjusted accordingly.

Mapping RxPDO2

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1601	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the second receive PDO; sub-index 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x64110110	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x64110210	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)

The 2. receive PDO (RxPDO2) is per default for the analog outputs. Depending on the number of the inserted outputs, the needed length of the PDO is calculated and the according objects are mapped.

For the digital outputs are organized in words, the length of the PDO can be directly seen in sub-index 0.

If the mapping is changed, the entry in sub-index 0 has to be adjusted accordingly.

Mapping RxPDO3-RxPDO16

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1602 - 0x160F	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the 3. to 10. receive PDO; sub-index 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)

The receive PDOs 3 to 16 (RxPDO3-16) get an automatic default mapping via the coupler depending from the connected terminals. The procedure is described under "PDO mapping".

Communication parameter TxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1800	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter of the first transmit PDO, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000180 + NODE_ID	COB-ID TxPDO1
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Sub-index 1 (COB-ID): The lower 11bit of the 32bit value (bits 0-10) contain the CAN identifier, the MSBit (bit 31) shows if the PDO is active (1) or not (0), bit 30 shows if a RTR access to this PDO is permitted (0) or not (1). The sub-index 2 contains the transmission type, sub-index 3 the repetition delay time between two equal PDOs. If an event timer exists with a value unequal 0, the PDO is transmitted when the timer exceeds.

If a inhibit timer exists, the event is delayed for this time.

Communication parameter TxPDO2

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1801	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter of the second transmit PDO, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000280 + NODE_ID	COB-ID TxPDO2
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO3

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1802	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 3. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000380 + NODE_ID	COB-ID TxPDO3
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO4

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1803	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 4. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000480 + NODE_ID	COB-ID TxPDO4
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO5

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1804	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 5. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO5
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO6

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1805	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 6. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO6
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO7

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1806	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 7. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO7
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO8

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1807	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 8. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO8
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO9

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1808	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 9. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO9
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO10

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1809	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 10. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO10
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO11

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180A	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 11. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO11
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO12

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180B	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 12. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO12
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO13

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180C	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 13. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO13
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO14

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180D	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 14. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO14
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO15

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180E	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 15. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO15
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Communication parameter TxPDO16

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180F	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 16. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO16
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Mapping TxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A00	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the first transmit PDO; sub-index 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x60000108	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x60000208	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped object	Unsigned32	rw	N	0x60000808	(2 byte index, 1 byte sub-index, 1 byte bit-width)

continue ...

... continue Mapping TxPD01

The first send PDO (TxPDO1) is per default for digital inputs. Depending on the number of the inserted inputs, the needed length of the PDO is calculated and the according objects are mapped.

For the digital inputs are organized in bytes, the length of the PDO can be directly seen in sub-index 0.

If the mapping is changed, the entry in sub-index 0 has to be adjusted accordingly.

Mapping TxPDO2

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A01	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the second transmit PDO; sub-index 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x64010110	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x64010210	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)

The 2.send PDO (RxPDO2) is per default for the analog inputs. Depending on the number of the inserted outputs, the needed length of the PDO is calculated and the according objects are mapped.

For the digital outputs are organized in words, the length of the PDO can be directly seen in sub-index 0.

If the mapping is changed, the entry in sub-index 0 has to be adjusted accordingly.

Mapping TxPDO3-TxPDO16

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A02 - 0x1A0F	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the 3. to 10 th transmit PDO; sub-index 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped object	Unsigned32	rw	N	0x000000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)

The send PDOs 3 to 16 (RxPDO3-16) get an automatic default mapping via the coupler depending from the connected terminals. The procedure is described under "PDO mapping".

CAN transfer rate

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2001	0	CAN transfer rate	Unsigned8	rw	N	0x01	Setting CAN transfer rate

This index entry writes a new transfer rate into the EEPROM.

At the next start-up (reset) the CAN coupler starts with the new transfer rate.

Value	CAN transfer rate
"00"	1Mbit/s
"01"	500kbit/s
"02"	250kbit/s
"03"	125kbit/s
"04"	100kbit/s
"05"	50kbit/s
"06"	20kbit/s
"07"	10kbit/s
"08"	800kbit/s

PDO control

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2400	0	Number of Elements	Unsigned8	ro	N	0x10	Time control for RxPDOs
	1	RxPDO1	Unsigned16	rw	N	0x0000	Timer value [ms]
	2	RxPDO2	Unsigned16	rw	N	0x0000	Timer value [ms]
	...	...	...	...	...	...	...
	16	RxPDO16	Unsigned16	rw	N	0x0000	Timer value [ms]

The control starts as soon as the timer is unequal 0. Every received RxPDO resets the timer. When the timer has been expired, the CAN coupler switches into the state "pre-operational" and sends an emergency telegram.

1. Module Parameter

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x3100	0	Number of Elements	Unsigned8	ro	N	depending on the number of configurable modules	Number of configurable modules 0x00 : no module available 0xXX : XX number of configurable modules available
	1	1. Prm 1. Module	Unsigned8	wo	N	depending on the compo-nents fitted	1. Parameter byte of the 1. configurable module
	2	1. Prm 2. Module	Unsigned8	wo	N	depending on the compo-nents fitted	1. Parameter byte of the 2. configurable module
	...	...	...	...	...	...	...
	64	1. Prm 64. Module	Unsigned8	wo	N	depending on the compo-nents fitted	1. Parameter byte of the 64. configurable module

The number of configurable modules may be determined with index 0x3100 and subindex 0.

Via subindex 1 the 1. parameter byte of the 1. configurable module may be accessed. Please consider that this value does not correspond to the slot number but to the position within the configurable modules.

The 1. parameter byte of the 1. configurable module is assigned to subindex 1, the 1. parameter byte of the 2. module is assigned to subindex 2, and so on.

2. ... 65. Module Parameter

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x3101	1	2. Prm 1. Module	Unsigned8	wo	N	depending on the components fitted	2. Parameter byte of the 1. configurable module
	2	2. Prm 2. Module	Unsigned8	wo	N	depending on the components fitted	2. Parameter byte of the 2. configurable module
	...	...	...	...	...	...	...
	64	2. Prm 64. Module	Unsigned8	wo	N	depending on the components fitted	2. Parameter byte of the 64. configurable module
0x3102	1	3. Prm 1. Module	Unsigned8	wo	N	depending on the components fitted	3. Parameter byte of the 1. configurable module
	2	3. Prm 2. Module	Unsigned8	wo	N	depending on the components fitted	3. Parameter byte of the 2. configurable module
	...	...	...	...	...	...	...
	64	3. Prm 64. Module	Unsigned8	wo	N	depending on the components fitted	3. Parameter byte of the 64. configurable module
...	...	...	...	...	...	...	...
0x3140	1	65. Prm 1. Module	Unsigned8	wo	N	depending on the components fitted	65. Parameter byte of the 1. configurable module
	2	65. Prm 2. Module	Unsigned8	wo	N	depending on the components fitted	65. Parameter byte of the 2. configurable module
	...	...	...	...	...	...	...
	64	65. Prm 64. Module	Unsigned8	wo	N	depending on the components fitted	65. Parameter byte of the 64. configurable module

The 2. ... 65. parameters of one module may be accessed via index 0x3101 ... 0x3140. Here the 1. ... 64. module may be defined via subindex 1 ... 64.

Write Parameters

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x31FF	0	Number of Elements	Unsigned8	ro	N	depending on the number of configurable modules	Number of configurable modules 0x00 : no module available 0xXX : XX number of configurable modules available
	1	Prm 1. Module	Unsigned8	wo	N	0x00 ... 0xFF	Writes parameters to 1. module with any write access
	2	Prm 2. Module	Unsigned8	wo	N	0x00 ... 0xFF	Writes parameters to 2. module with any write access
	...	...	...	...	...	...	...
	64	Prm 64. Module	Unsigned8	wo	N	0x00 ... 0xFF	Writes parameters to 64. module with any write access

The number of configurable modules may be determined with index 0x31FF and subindex 0.

The activation of the preset parameters via index 0x3100 ... 0x3140 happens by any write access to index 0x31FF. Here the module 1 to 64 is defined by subindex 1 ... 64.

Example

Your System SLIO consists of the following configurable modules:

- 031-1BB30
- 032-1BB30

The parameters may be accessed with the following indexes:

Change parameters

Index	Subindex	
	1	2
	031-1BB30	032-1BB30
0x3100	Function channel 0	reserved
0x3101	Function channel 1	Short circuit recognition
0x3102	reserved	Function Channel 0
0x3103	reserved	Function Channel 1

Write parameters

Index	Subindex	
	1	2
	031-1BB30	032-1BB30
0x31FF	Write parameters	Write parameters

Access to record set bus coupler

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x3200	0x00	Number of Elements	Unsigned8	ro	N	0xFF	Number of record sets
	0x50	Device name	Visible string	ro	N		Device name
	0x51	HW Rev.	Visible string	ro	N		Hardware revision
	0x52	SW Rev.	Visible string	ro	N		Software revision
	0x53	Serial No.	Unsigned32	ro	N		Serial number
	0x54	FPGA vers.	Unsigned16	ro	N		FPGA version

With index 0x3200 the bus coupler may be accessed. Here the record set may be defined with the corresponding sub-index.

Access to record set function modules

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x3201 ... 0x3240	0x00	Number of Elements	Unsigned8	ro	N	0xFF	Number of record sets of the modules at slot 1 ... 64
	0x01	Diag RS1	Octet string	r	N		Diagnostics record set 1
	0x02	Diag RS0	Octet string	r	N		Diagnostics record set 0
	0x01	Param RS1	Octet string	w	N		Parameter record set 1
	0x02	Param RS0	Octet string	w	N		Parameter record set 0
	0x50	Device name	Visible string	ro	N		Device name
	0x51	HW Rev.	Visible string	ro	N		Hardware revision
	0x52	SW Rev.	Visible string	ro	N		Software revision
	0x53	Serial No.	Unsigned32	ro	N		Serial number
	0x54	FPGA vers.	Unsigned16	ro	N		FPGA version
	0x7D	RS 0...N	Octet string	rw	N		Record set 0...N
	0x7E	RS 0	Octet string	rw	N		Record set 0
	0x7F	RS 1	Octet string	rw	N		Record set 1
	0x80	RS 128	Octet string	rw	N		Record set 128
	0x81	RS 129	Octet string	rw	N		Record set 129
	...	...	...	...	...		...
	0xAF	RS 175	Octet string	rw	N		Record set 175

The record sets of the assigned function modules at slot 1...64 may be accessed by the index 0x3201...0x3240.

Here the record set may be defined with the corresponding sub-index.

Note!

Please regard with a write access to sub-index 0x01/0x02 the record set 1/0 of the parameter data is written but with a read access the record set 1/0 of the diagnostics data is read!

2/4bit Digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5000	0x00	2/4bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1. input block	Unsigned8	ro	Y		1. digital input block
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned8	ro	Y		64. digital input block

2/4bit Polarity Digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5002	0x00	2/4bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1. input block	Unsigned8	rw	N	0x00	1. polarity digital input block
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned8	rw	N	0x00	64. polarity digital input block

Individual inversion of the input channels

1 = Input inverted

0 = Input not inverted

2/4bit Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5200	0x00	2/4bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	Y		1. digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	Y		64. digital output block

2/4bit Change Polarity Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5202	0x00	2/4bit digital ouput block	Unsigned8	ro	N	Depending on the compo-nents fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0x00	1. polarity digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0x00	64. polarity digital output block

Individual inversion of the output channels

1 = Output inverted

0 = Output not inverted

2/4bit Error Mode Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5206	0x00	2/4bit digital output block	Unsigned8	ro	N	Depending on the compo-nents fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0xFF	1. error mode digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0xFF	64. error mode digital output block

By means of this object you may define whether in an event of an error an output channel takes a defined value, which may be defined with object 0x5207.

1 = take value defined at object 0x5207

0 = on error fix output value

2/4bit Error Value Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5207	0x00	2/4bit digital output block	Unsigned8	ro	N	Depending on the compo-nents fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0x00	1. error value digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0x00	64. error value digital output block

With an activated error mode, in an event of an error the value preset here is taken.

8bit digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6000	0x00	8bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1. input block	Unsigned8	ro	Y		1. digital input block
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned8	ro	Y		64. digital input block

8bit polarity digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6002	0x00	8bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1. input block	Unsigned8	rw	N	0x00	1. polarity digital input block
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned8	rw	N	0x00	64. polarity digital input block

Individual inverting of input polarity:

1 = input inverted
0 = input not inverted

8bit digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6200	0x00	8bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	Y		1. digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	Y		64. digital output block

8bit change polarity digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6202	0x00	8bit digital output block	Unsigned8	ro	N	Depending on the compo-nents fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0x00	1. polarity digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0x00	64. polarity digital output block

Individual inverting of input channels:

1 = input inverted
0 = input not inverted

8bit error mode digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6206	0x00	8bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0xFF	1. error mode digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0xFF	64. error mode digital output block

This object indicates whether an output is set to a pre-defined error value (set in object 0x6207) in case of an internal device failure.

1 = overtake the value from object 0x6207
0 = keep output value in case of error

8bit error value digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6207	0x00	8bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0x00	1. error value digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0x00	64. error value digital output block

Presupposed that the error mode is active, device failures set the output to the value configured by this object.

Analog inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6401	0x00	2byte input block	Unsigned8	ro	N	depending on the compo-nents fitted	Number of available analog inputs
	0x01	1. input channel	Unsigned16	ro	Y		1. analog input channel
	...	...	...	...	...	...	...
	0xFE	254. input channel	Unsigned16	ro	Y		254. analog input channel

Analog outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6411	0x00	2byte output block	Unsigned8	ro	N	depending on the compo-nents fitted	Number of available analog outputs
	0x01	1.output channel	Unsigned16	ro	Y		1. analog output channel
	...	...	...	...	...	...	...
	0xFE	254. output channel	Unsigned16	ro	Y		254. analog output channel

Analog input interrupt trigger selection

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6421	0x00	Number of Inputs	Unsigned8	ro	N	depending on the compo-nents fitted	Number of available analog inputs
	0x01	Trigger 1. input channel	Unsigned8	rw	N	0x07	Input interrupt trigger for 1. analog input channel
	...	...	...	...	...	...	...
	0xFE	Trigger 254. input channel	Unsigned8	rw	N	0x07	Input interrupt trigger for 254. analog input channel

This object determines which events shall cause an interrupt for a specific channel. Bits set in the list below refer to the interrupt trigger.

Bit no.	Interrupt trigger
0	Upper limit exceeded 6424
1	Input below lower limit 6425
2	Input changed by more than negative delta 6426
3 to 7	Reserved

Analog input interrupt source

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6422	0x00	Number of Interrupt	Unsigned8	ro	N	0x01	Number of interrupt source bank
	0x01	Interrupt source bank	Unsigned32	ro	N	0x00000000	Interrupt source bank 1
	0x02	Interrupt source bank	Unsigned32	ro	N	0x00000000	Interrupt source bank 2

This object defines the channel that is responsible for the Interrupt. Bits set refer to the number of the channel that caused the Interrupt. The bits are automatically reset, after they have been read by a SDO or send by a PDO.

1 = Interrupt produced

0 = Interrupt not produced

Event driven analog inputs

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6423	0x00	Global interrupt enable	Boolean	rw	N	FALSE ("0")	Activates the event-driven transmission of PDOs with analog inputs

Although the analog inputs are -acc. to CANopen - per default set to the transmission type 255 (event triggered) in the TxPDO2, the "event" (the alteration of an input value) is suppressed by the event control in object 0x6423 in order to prevent the bus from being swamped with analog signals.

Before activation, it is convenient to parameterize the transmission behavior of the analog PDOs:

- inhibit time (object 0x1800ff, sub-index 3)
- limit value monitoring (objects 0x6424 + 0x6425)
- delta function (object 0x6426)

Upper limit value analog inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6424	0x00	Number of Inputs	Unsigned8	ro	N	depending on the compo-nents fitted	Number of available analog inputs
	0x01	Upper limit 1. input channel	Unsigned32	rw	N	0x00000000	Upper limit value for 1. analog input channel
	...	...	...	...	...	...	...
	0xFE	Upper limit 254. input channel	Unsigned32	rw	N	0x00000000	Upper limit value for 254. analog input channel

Values unequal to zero are activating the upper limit value for this channel. A PDO is then transmitted when the upper limit value is exceeded. In addition, the event trigger has to be active (object 0x6423). The data format corresponds to that of the analog inputs.

Lower limit value analog inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6425	0x00	Number of Inputs	Unsigned8	ro	N	depending on the compo-nents fitted	Number of available analog inputs
	0x01	Lower limit 1. input channel	Unsigned32	rw	N	0x00000000	Lower limit value for 1. analog input channel
	...	...	...	...	...	...	...
	0xFE	Lower limit 254. input channel	Unsigned32	rw	N	0x00000000	Lower limit value for 254. analog input channel

Values unequal to zero are activating the lower limit value for this channel. A PDO is then transmitted when the lower limit value is underrun. In addition, the event trigger has to be active (object 0x6423). The data format corresponds to that of the analog inputs.

Delta function

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6426	0x00	Number of Inputs	Unsigned8	ro	N	depending on the compo-nents fitted	Number of available analog inputs
	0x01	Delta value 1. input channel	Unsigned32	rw	N	0x00000002	Delta value for 1. analog input channel
	...	...	...	...	...	...	...
	0xFE	Delta value 254. input channel	Unsigned32	rw	N	0x00000002	Delta value for 254. analog input channel

Values unequal to zero are activating the delta function for this channel. A PDO is then transmitted when the value has been changed for more than the delta value since the last transmission. In addition, the event trigger has to be active (object 0x6423). The data format corresponds to that of the analog inputs (The delta function accepts only positive values).

Analog output error mode

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6443	0x00	Analog output block	Unsigned8	ro	N	Depending on the components fitted	Number of available analog outputs
	0x01	1. analog output block	Unsigned8	rw	N	0xFF	1. error mode analog output block
	...	...	...	...	...	...	...
	0xFE	254. analog output block	Unsigned8	rw	N	0xFF	254. error mode analog output block

This object indicates whether an output is set to a pre-defined error value (set in object 0x6444) in case of an internal device failure.

0 = current value

1 = set to error value 0x6444

Analog output error value

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6444	0x00	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available analog output blocks
	0x01	1. analog block	Unsigned16	rw	N	0x0000	1. analog output block
	...	...	...	...	...	...	...
	0xFE	254. analog block	Unsigned16	rw	N	0x0000	254. analog output block

Presupposed that the corresponding error (0x6443) is active, device failures set the output to the value configured by this object.

SDO Abort Codes

0x05030000 //Toggle bit not alternated
0x05040000 //SDO protocol timed out
0x05040001 //Client/server command specifier not valid or unknown
0x05040002 //Invalid block size (block mode only)
0x05040003 //Invalid sequence number (block mode only)
0x05040004 //CRC error (block mode only)
0x05040005 //Out of memory
0x06010000 //Unsupported access to an object
0x06010001 //Attempt to read a write only object
0x06010002 //Attempt to write a read only object
0x06020000 //Object does not exist in the object dictionary
0x06040041 //Object cannot be mapped to the PDO
0x06040042 //The number and length of the objects to be mapped would exceed PDO length
0x06040043 //General parameter incompatibility reason
0x06040047 //General internal incompatibility in the device
0x06060000 //Access failed due to an hardware error
0x06070010 //Data type does not match, length of service parameter does not match
0x06070012 //Data type does not match, length of service parameter too high
0x06070013 //Data type does not match, length of service parameter too low
0x06090011 //Sub-index does not exist
0x06090030 //Value range of parameter exceeded (only for write access)
0x06090031 //Value of parameter written too high
0x06090032 //Value of parameter written too low
0x06090036 //Maximum value is less than minimum value
0x08000000 //general error
0x08000020 //Data cannot be transferred or stored to the application
0x08000021 //Data cannot be transferred or stored to the application because of local control
0x08000022 //Data cannot be transferred or stored to the application because of the present device state
0x08000023 //Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated
from file and generation fails because of an file error)

Emergency Object

Overview

The SLIO CAN bus coupler is provided with the emergency object to notify other devices connected to the CANopen bus about internal error events or CAN-Bus errors. It has a high priority and gives you important information about the states of device and network.

Note!

We strongly recommend to analyze the emergence object - it is an important information pool!

Telegram structure

The emergency telegram has always a length of 8byte. It starts with 2byte error code followed by the 1byte error register and closes with 5byte additional code.

Error code low byte	Error code high byte	ErrorRegister Index 0x1001	Info 0	Info 1	Info 2	Info 3	Info 4

Error messages

Error Code	Meaning	Info 0	Info 1	Info 2	Info 3	Info4
0x0000	Reset Emergency					
0x1000	PDO Control	0xFF	0x10	PDO Number	LowByte Timer Value	HighByte Timer Value
0x8100	Heartbeat Consumer	Node ID	LowByte Timer Value	HighByte Timer Value	0x00	0x00
0x8100	SDO Block Transfer	0xF1	LowByte Index	HighByte Index	SubIndex	0x00
0x8130	Node Guarding Error	LowByte GuardTime	HighByte GuardTime	LifeTime	0x00	0x00
0x8210	PDO not processed due to length error	PDO Number	Wrong length	PDO length	0x00	0x00
0x8220	PDO length exceeded	PDO Number	Wrong length	PDO length	0x00	0x00

Network management

Übersicht

Network management (NMT) provides the global services specifications for network supervision and management. This includes the login and logout of the different network devices, the supervision of these devices as well as the processing of exceptions.

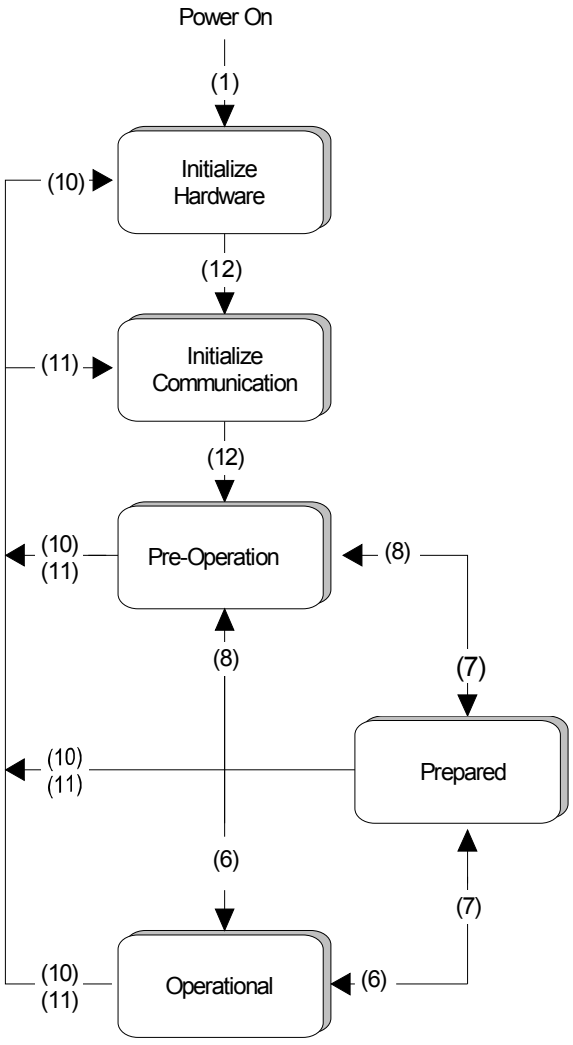
NMT service messages have the COB identifier 0x0000. An additional module-ID is not required. The length is always 2 data bytes.

The 1. data byte contains the NMT command specifier: **CS**.

The 2. data byte contains the module-ID (0x00 for broadcast command).

States

The following picture shows an overview over all CANopen status changes and the corresponding NMT command specifiers:



(1): The initialization state is reached automatically after start-up.

(6): "Start_Remote_Node" (CS: 0x01)
Starts the module, releases outputs and starts the PDO transmission.

(7): "Stop_Remote_Node" (CS: 0x02)
Outputs are switching into error state, SDO and PDO are switched off.

(8): "Enter_Pre-operational_State" (CS:0x80)
Stops PDO transmission, SDO still active.

(10): "Reset_Node" (CS:0x81)
Executes reset. All objects are set back to PowerOn defaults.

(11): "Reset_Communication" (CS:0x82)
Executes reset of the communication functions. Objects 0x1000 - 0x1FFF are set back to PowerOn defaults.

(12): After initialization the state "pre-operational is automatically reached - here the boot-up message is send.

Node Guarding

The bus coupler also supports the Node Guarding object as defined by CANopen to ensure that other devices on the bus are supervised properly.

Node Guarding operation is started when the first guard requests (RTR) is received from the master. The respective COB identifier is permanently set to 0x700 + module-ID at variable 0x100E in the object directory. If the coupler does not receive a guard request message from the master within the "guard time" (object 0x100C) when the node guarding mode is active the module assumes that the master is not operating properly. When the time determined by the product of "guard time" (0x100C) and "life-time factor" (0x100D) has expired, the module will automatically assume the status "pre-operational".

When either the "guard time" (object 0x100C) or the "life-time factor" (0x100D) has been set to zero by an SDO download from the master, the expiry of the guard time is not monitored and the module remains in its current operating mode.

Heartbeat

The VIPA CAN coupler also supports the Heartbeat Mode in addition to Node Guarding.

When a value is entered into index 0x1017 (Heartbeat Producer Time) then the device status (Operational, Pre-Operational,...) of the bus coupler is transferred by means of the COB identifier (0x700+module-ID) when the heartbeat timer expires.

The Heartbeat Mode starts automatically as soon as the index 1017h contains a value that is larger than 0.