VIPA System 200V

CP | Manual

HB97E_CP | RE_240-1EA20 | Rev. 12/42

October 2012

For permission to reproduce or distribute, please contact:
VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH
Ohmstraße 4, D-91074 Herzogenaurach, Germany
Tel.: +49 (91 32) 744 -0
Fax.: +49 9132 744 1864
EMail: info@vipa.de
http://www.vipa.com

Note

Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. Nevertheless, the authors retain the right to modify the information. This customer document describes all the hardware units and functions known at the present time. Descriptions may be included for units which are not present at the customer site. The exact scope of delivery is described in the respective purchase contract.

CE Conformity Declaration

Hereby, VIPA GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions.

Conformity is indicated by the CE marking affixed to the product.

Conformity Information

For more information regarding CE marking and Declaration of Conformity (DoC), please contact your local VIPA customer service organization.

Trademarks

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

SPEED7 is a registered trademark of profichip GmbH.

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

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

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

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

Information product support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax:+49 9132 744 1204
EMail: documentation@vipa.de

Technical support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telephone: +49 9132 744 1150 (Hotline)
EMail: support@vipa.de

Contents

About this manual

This manual describes the System 200V CP 240-EA20 that are available from VIPA. It contains detailed descriptions of the CP with EnOcean Transceiver module

Overview

Chapter 1: Basics and Assembly

The focus of this chapter is on the introduction of the VIPA System 200V. Here you will find the information required to assemble and wire a controller system consisting of System 200V components.
Besides the dimensions the general technical data of System 200V will be found.

Chapter 2: Hardware description

This chapter contains a description of the construction and the interfaces of the communication processor CP 240 with EnOcean Transceiver module.

Chapter 3: Deployment

Here you will find the deployment of the communication processor CP 240 EnOcean.

Objective and contents

This manual describes the System 200V CP 240-1EA20 from VIPA.
It contains a description of the construction, project implementation and usage.

This manual is part of the documentation package with order number HB97E_CP and relevant for:

Product	Order number	as of state: HW
CP 240 EnOcean	VIPA CP 240-1EA20	01

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.

ℹ

Note!
Supplementary information and useful tips.

Safety information

Applications conforming with specifications

The CP 240 is constructed and produced for:

- all VIPA System 200V components
- 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:

- Hardware modifications to the process control system should only be carried out when the system has been disconnected from power!

- Installation and hardware modification 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 200V. Here you will find the information required to assemble and wire a controller system consisting of System 200V components.

Besides the dimensions the general technical data of System 200V will be found.

Contents

Topic **Page**

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 electrostatic sensitive 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.

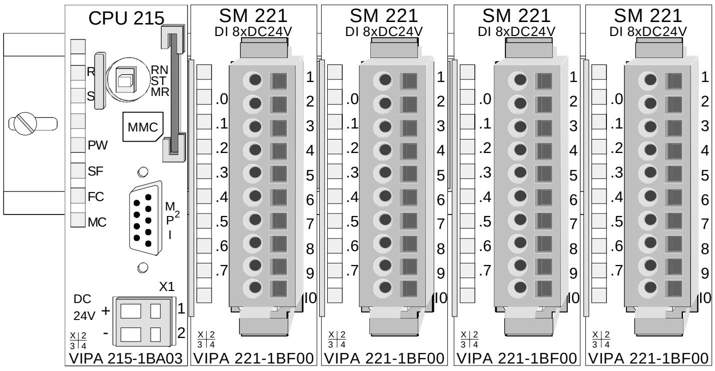
Attention!

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

System conception

Overview

The System 200V is a modular automation system for assembly on a 35mm profile rail. By means of the peripheral modules with 4, 8 and 16 channels this system may properly be adapted matching to your automation tasks.

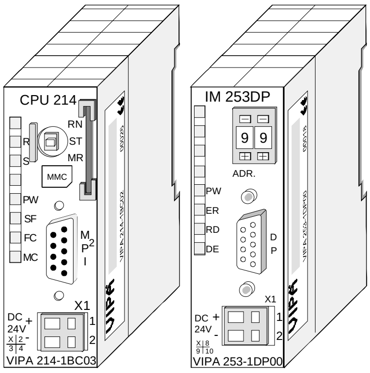


Components

The System 200V consists of the following components:

- *Head modules* like CPU and bus coupler
- *Periphery modules* like I/O, function und communication modules
- *Power supplies*
- *Extension modules*

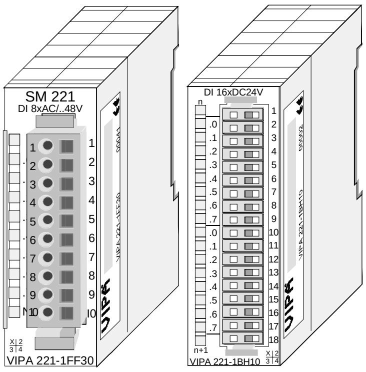
Head modules



With a head module CPU respectively bus interface and DC 24V power supply are integrated to one casing.

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

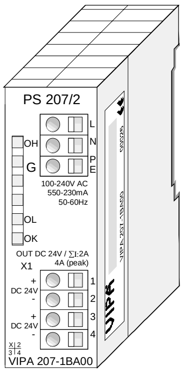
Periphery modules



The modules are direct installed on a 35mm profile rail and connected to the head module by a bus connector, which was mounted on the profile rail before.

Most of the periphery modules are equipped with a 10pin respectively 18pin connector. This connector provides the electrical interface for the signaling and supplies lines of the modules.

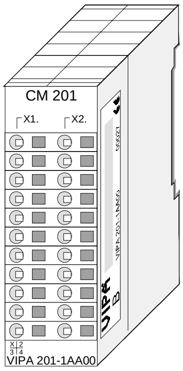
Power supplies



With the System 200V the DC 24V power supply can take place either externally or via a particularly for this developed power supply.

The power supply may be mounted on the profile rail together with the System 200V modules. It has no connector to the backplane bus.

Expansion modules



The expansion modules are complementary modules providing 2- or 3wire connection facilities.

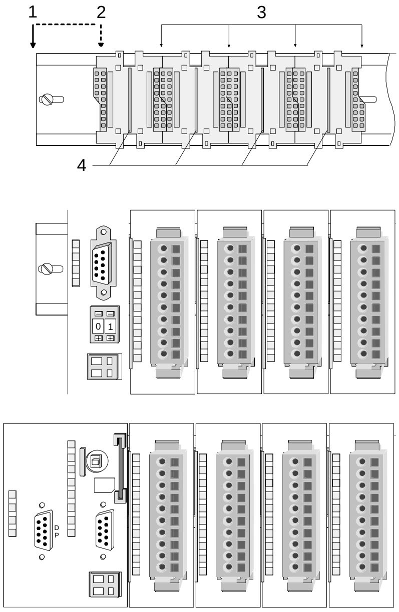
The modules are not connected to the backplane bus.

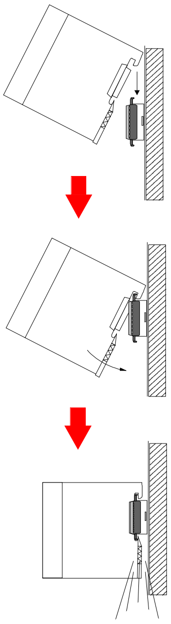
Structure/ dimensions

- Profile rail 35mm
- Dimensions of the basic enclosure:
  1tier width: (HxWxD) in mm: 76x25.4x74 in inches: 3x1x3
  2tier width: (HxWxD) in mm: 76x50.8x74 in inches: 3x2x3

Installation

Please note that you can only install head modules, like the CPU, the PC and couplers at slot 1 or 1 and 2 (for double width modules).





[1]	Head module (double width)
[2]	Head module (single width)
[3]	Periphery module
[4]	Guide rails

Note

Information about the max. number of pluggable modules and the max. current at the backplane bus can be found in the "Technical Data" of the according head module.

Please install modules with a high current consumption directly beside the head module.

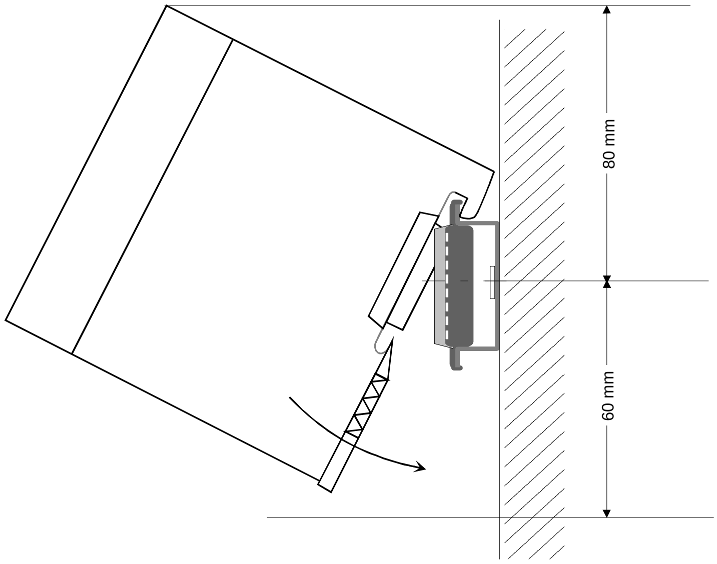
Dimensions

Dimensions Basic enclosure

1tier width (HxWxD) in mm: 76 x 25.4 x 74

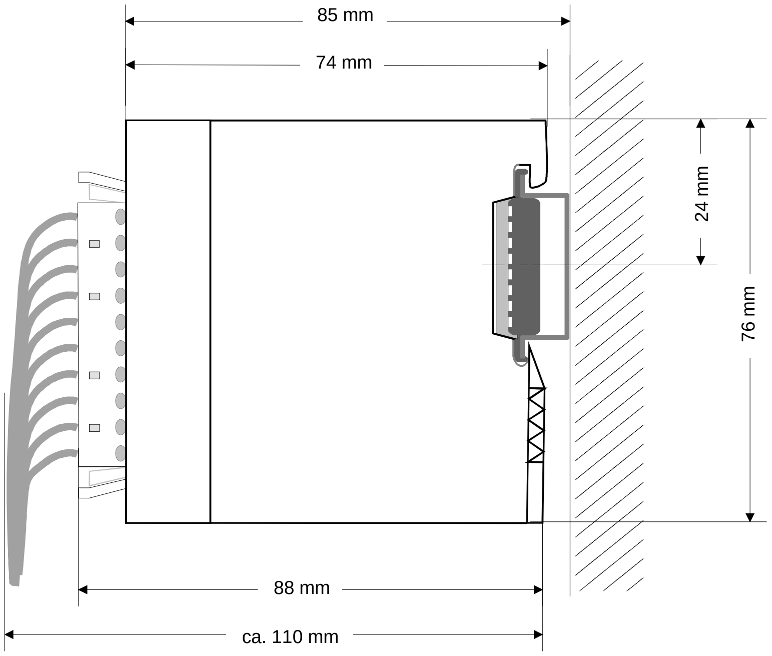
2tier width (HxWxD) in mm: 76 x 50.8 x 74

Installation dimensions

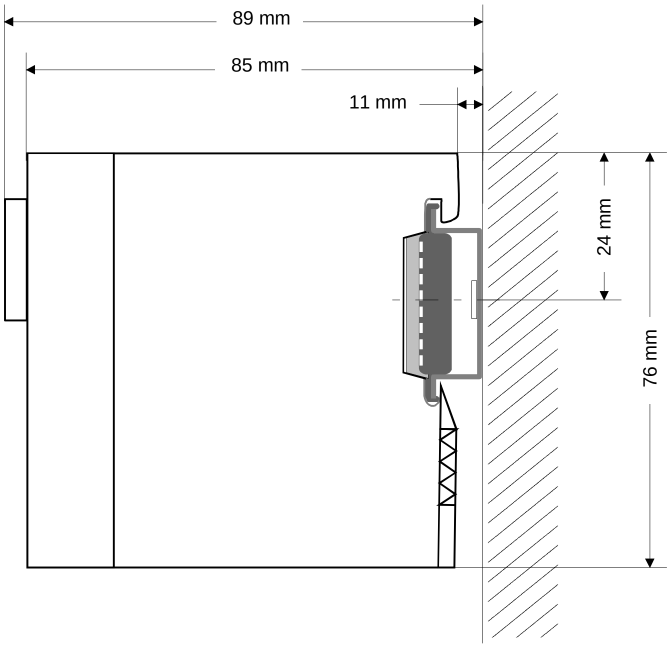


Installed and wired dimensions

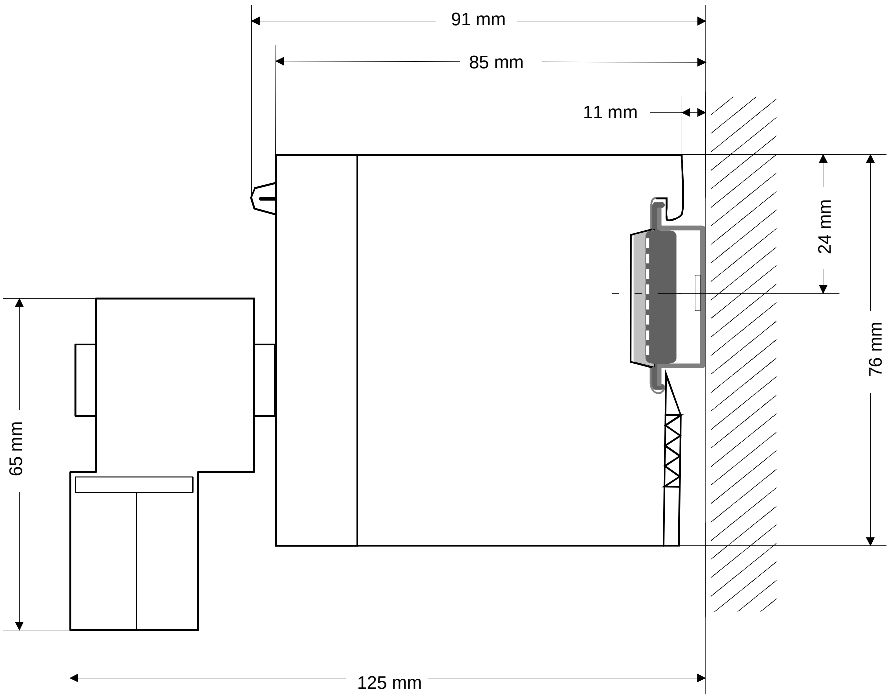
In- / Output modules



Function modules/
Extension modules



CPUs (here with
EasyConn from
VIPA)



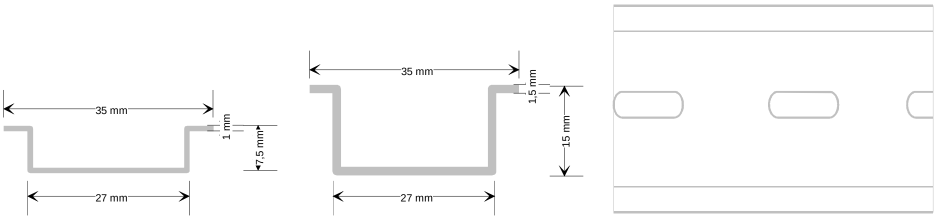
Installation

General

The modules are each installed on a 35mm profile rail and connected via a bus connector. Before installing the module the bus connector is to be placed on the profile rail before.

Profile rail

For installation the following 35mm profile rails may be used:

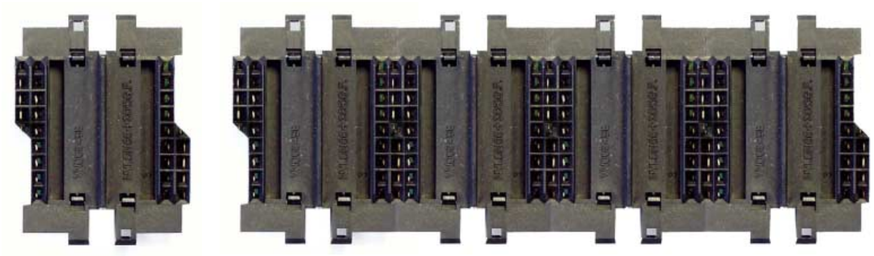


Order number	Label	Description
290-1AF00	35mm profile rail	Length 2000mm, height 15mm
290-1AF30	35mm profile rail	Length 530mm, height 15mm

Bus connector

System 200V modules communicate via a backplane bus connector. The backplane bus connector is isolated and available from VIPA in of 1-, 2-, 4- or 8tier width.

The following figure shows a 1tier connector and a 4tier connector bus:



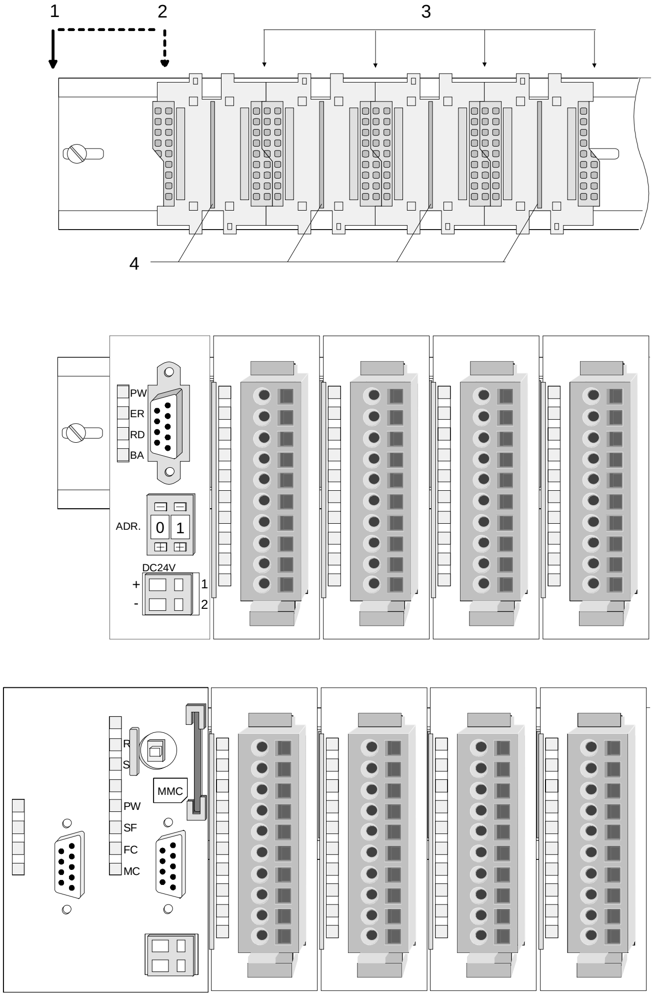
The bus connector is to be placed on the profile rail until it clips in its place and the bus connections look out from the profile rail.

Order number	Label	Description
290-0AA10	Bus connector	1tier
290-0AA20	Bus connector	2tier
290-0AA40	Bus connector	4tier
290-0AA80	Bus connector	8tier

Installation on a profile rail

The following figure shows the installation of a 4tier width bus connector in a profile rail and the slots for the modules.

The different slots are defined by guide rails.

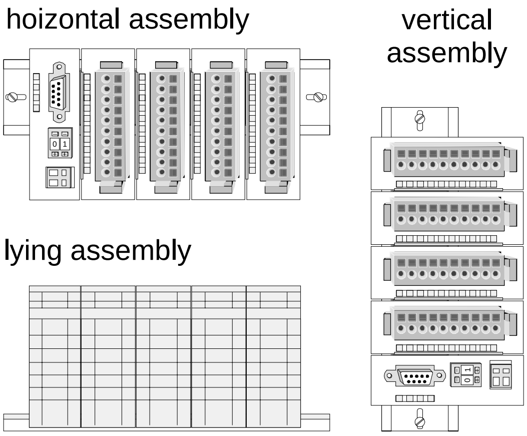


[1] Head module (double width)
[2] Head module (single width)
[3] Peripheral module
[4] Guide rails

Assembly regarding the current consumption

- Use bus connectors as long as possible.
- Sort the modules with a high current consumption right beside the head module. In the service area of www.vipa.com a list of current consumption of every System 200V module can be found.

Assembly possibilities

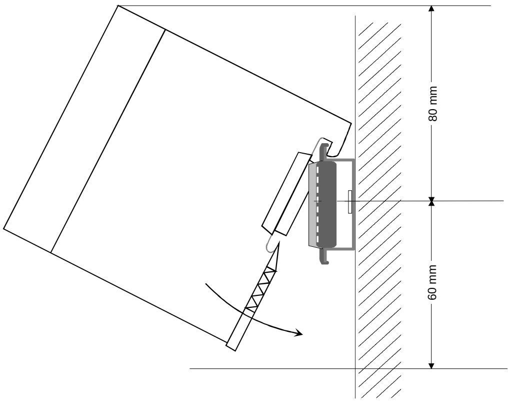


Please regard the allowed environmental temperatures:

- horizontal assembly: from 0 to 60°C
- vertical assembly: from 0 to 40°C
- lying assembly: from 0 to 40°C

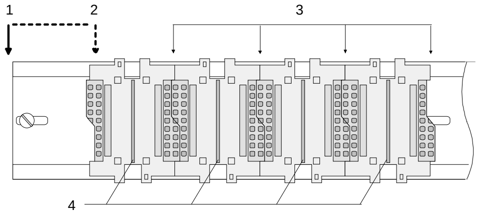
The horizontal assembly always starts at the left side with a head module, then you install the peripheral modules beside to the right.

You may install up to 32 peripheral modules.



Please follow these rules during the assembly!

- Turn off the power supply before you install or remove any modules!
- Make sure that a clearance of at least 60mm exists above and 80mm below the middle of the profile rail.



- Every row must be completed from left to right and it has to start with a head module.

[1] Head module (double width)
[2] Head module (single width)
[3] Peripheral modules
[4] Guide rails

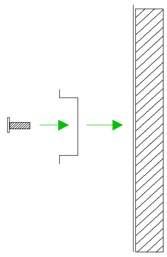
- Modules are to be installed side by side. Gaps are not permitted between the modules since this would interrupt the backplane bus.
- A module is only installed properly and connected electrically when it has clicked into place with an audible click.

Slots after the last module may remain unoccupied.

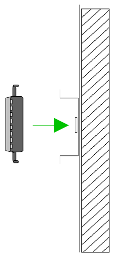
Note!

A maximum of 32 modules can be connected at the back plane bus. Take attention that here the maximum **sum current** of **3.5A** is not exceeded.

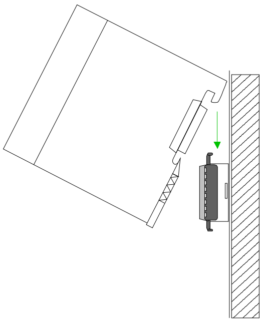
Assembly procedure



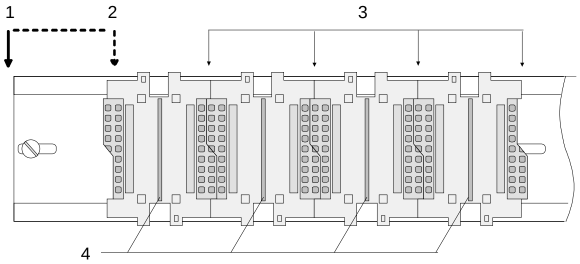
- Install the profile rail. Make sure that a clearance of at least 60mm exists above and 80mm below the middle of the profile rail.



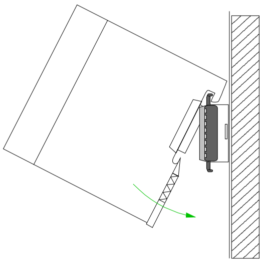
- Press the bus connector into the profile rail until it clips securely into place and the bus-connectors look out from the profile rail. This provides the basis for the installation of your modules.



- Start at the outer left location with the installation of your head module and install the peripheral modules to the right of this.



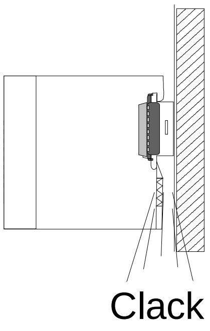
[1]	Head module (double width)
[2]	Head module (single width)
[3]	Peripheral module
[4]	Guide rails



- Insert the module that you are installing into the profile rail at an angle of 45 degrees from the top and rotate the module into place until it clicks into the profile rail with an audible click. The proper connection to the backplane bus can only be guaranteed when the module has properly clicked into place.

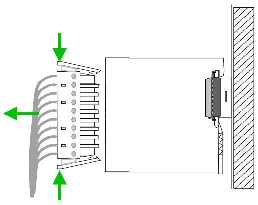
Attention!

Power must be turned off before modules are installed or removed!



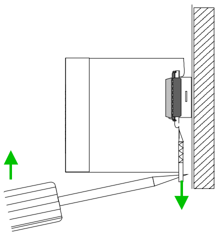
Demounting and module exchange

1



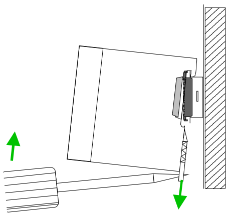
- Remove if exists the wiring to the module, by pressing both locking lever on the connector and pulling the connector.

2



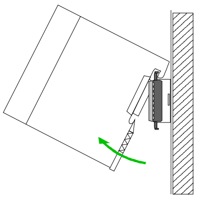
- The casing of the module has a spring loaded clip at the bottom by which the module can be removed.

3



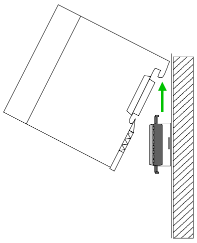
- The clip is unlocked by pressing the screwdriver in an upward direction.

4



- Withdraw the module with a slight rotation to the top.

5



Attention!

Power must be turned off before modules are installed or removed!

Please regard that the backplane bus is interrupted at the point where the module was removed!

Wiring

Overview

Most peripheral modules are equipped with a 10pole or a 18pole connector. This connector provides the electrical interface for the signaling and supply lines of the modules.

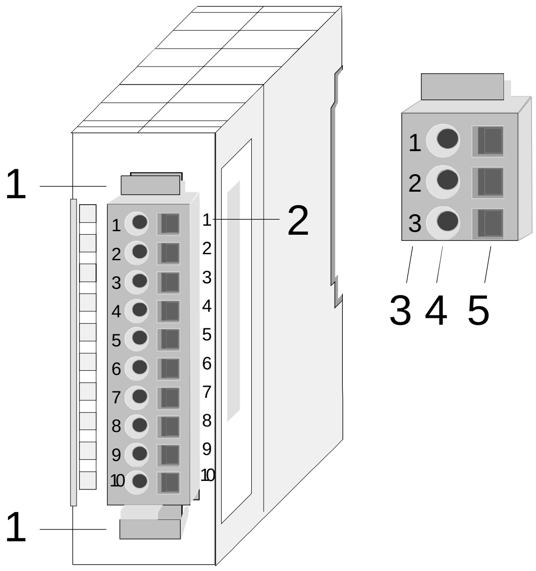
The modules carry spring-clip connectors for interconnections and wiring.

The spring-clip connector technology simplifies the wiring requirements for signaling and power cables.

In contrast to screw terminal connections, spring-clip wiring is vibration proof. The assignment of the terminals is contained in the description of the respective modules.

You may connect conductors with a diameter from 0.08mm^2 up to 2.5mm^2 (max. 1.5mm^2 for 18pole connectors).

The following figure shows a module with a 10pole connector.



[1] Locking lever

[2] Pin no. at the module

[3] Pin no. at the connector

[4] Wiring port

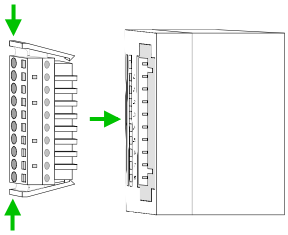
[5] Opening for screwdriver

Note!

The spring-clip is destroyed if you push the screwdriver into the wire port!

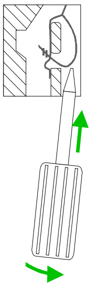
Make sure that you only insert the screwdriver into the square hole of the connector!

Wiring procedure

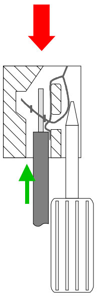


- Install the connector on the module until it locks with an audible click. For this purpose you press the two clips together as shown.
  The connector is now in a permanent position and can easily be wired.

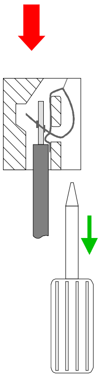
The following section shows the wiring procedure from top view.



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



- Insert the stripped end of the wire into the round opening. You can use wires with a diameter of 0.08mm^2 to 2.5mm^2 (1.5mm^2 for 18pole connectors).



- By removing the screwdriver the wire is connected safely with the plug connector via a spring.

Note!
Wire the power supply connections first followed by the signal cables (inputs and outputs).

Installation guidelines

General

The installation guidelines contain information about the interference free deployment of System 200V systems. 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 200V 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.
  - Please consider luminescent lamps can influence signal lines.
- Create a homogeneous reference potential and ground all electrical operating supplies when possible.
  - Please take care for the targeted employment of the grounding actions. The grounding of the PLC is a protection and functionality activity.
  - Connect installation parts and cabinets with the System 200V 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 200V 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

Structure/ dimensions

- Profile rail 35mm
- Peripheral modules with recessed labelling
- Dimensions of the basic enclosure:
  1tier width: (HxWxD) in mm: 76x25.4x74 in inches: 3x1x3
  2tier width: (HxWxD) in mm: 76x50.8x74 in inches: 3x2x3

Reliability

- Wiring by means of spring pressure connections (CageClamps) at the front-facing connector, core cross-section 0.08 ... 2.5mm^2 or 1.5 mm^2 (18pole plug)
- Complete isolation of the wiring when modules are exchanged
- Every module is isolated from the backplane bus
- ESD/Burst acc. IEC 61000-4-2 / IEC 61000-4-4 (to level 3)
- Shock resistance acc. IEC 60068-2-6 / IEC 60068-2-27 (1G/12G)
- Class of protection IP20

Environmental conditions

- Operating temperature: 0 ... +60°C
- Storage temperature: -25 ... +70°C
- Relative humidity: 5 ... 95% without condensation
- Ventilation by means of a fan is not required

Chapter 2 Hardware description

Overview

This chapter contains a description of the construction and the interfaces of the communication processor CP 240 with EnOcean Transceiver module.

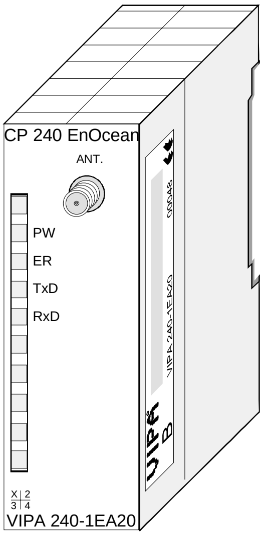
Contents

Topic **Page**

Properties

CP 240 EnOcean
240-1EA20

- CP with EnOcean radio transceiver module
- 16Byte Parameter data
- Voltage supply via back plane bus
- The TCM 120 **T**rans**c**eiver **m**odule works at 868.3MHz

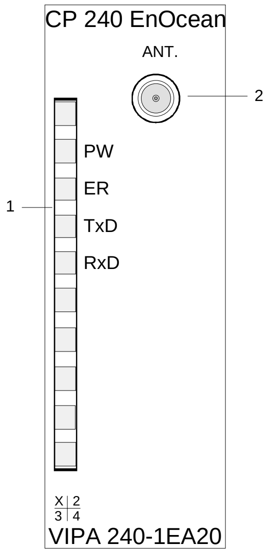


Order data

Type	Order number	Description
CP 240 EnOcean	VIPA 240-1EA20	CP with EnOcean radio transceiver module TCM 120
Portable antenna	VIPA 240-0EA00	Portable antenna with SMA plug
Magnetic socket antenna	VIPA 240-0EA10	Magnetic socket antenna with 150cm cable and SMA plug

Structure

CP 240 EnOcean
240-1EA20



[1] LED Status monitor

[2] SMA antenna jack
with male thread and calyx

Interface

ANT.
1
SMA Antenna

Antennas

The consignment doesn’t include an antenna but you may optional order a portable antenna or a magnetic socket antenna with 150cm cable.

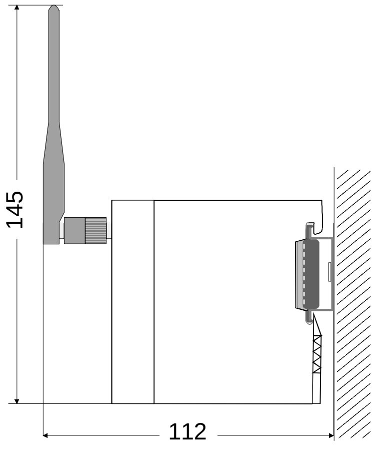
Both antennas are provided with a SMA plug. The coaxial build SMA plug (**s**traight **m**edium **a**daptor) is a miniature HF plug with threaded connector that excels by high HF denseness. In the standard version the plug has a swivel nut with female thread and a pin.

The SMA jack at the CP is with its male thread and the calyx the complement for assembly.

Portable antenna

The portable antenna is a short rod antenna that is mounted without cable directly at the module via the SMA plug.

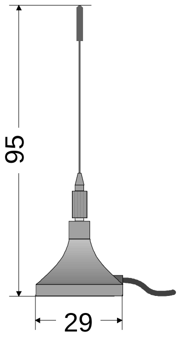
The antenna may be angled and turned into any direction.



all measures in mm

Magnetic socket antenna

The magnetic socket antenna with 150cm cable is convenient for mounting into a cabinet. Due to the magnetic socket you may install the antenna to any steel surface. The connection of the magnetic socket antenna to the CP 240 EnOcean happens via the antenna cable of 150cm with SMA plug.



all measures in mm

Power supply

The communication prozessor receives power via the back plane bus.

LEDs

The communication processor is provided with 4 LEDs to monitor the operating status. The meaning and the according colors are shown in the following table.

Label	Color	Description
PW	Green	Signalizes a present operating voltage
ER	Red	Signalizes an error by buffer overflow
TxD	Green	transmit data
RxD	Green	receive data

Technical Data

Order number	**240-1EA20**
Type	CP 240, EnOcean
Current consumption/power loss	
Current consumption from backplane bus	120 mA
Power loss	0.75 W
Status information, alarms, diagnostics	
Status display	yes
Interrupts	no
Process alarm	no
Diagnostic interrupt	no
Diagnostic functions	no
Diagnostics information read-out	none
Supply voltage display	yes
Group error display	red LED
Channel error display	none
Functionality Sub-D interfaces	
Type	-
Type of interface	-
Connector	-
Electrically isolated	-
MPI	-
MP²I (MPI/RS232)	-
DP master	-
DP slave	-
Point-to-point interface	-
Point-to-point communication	
PtP communication	-
Interface isolated	-
RS232 interface	-
RS422 interface	-
RS485 interface	-
Connector	SMA antenna socket
Transmission speed, min.	-
Transmission speed, max.	9.6 kbit/s
Cable length, max.	-
Point-to-point protocol	
ASCII protocol	-
STX/ETX protocol	-
3964(R) protocol	-
RK512 protocol	-
USS master protocol	-
Modbus master protocol	-
Modbus slave protocol	-
Special protocols	EnOcean
Datasizes	
Input bytes	16
Output bytes	16
Parameter bytes	16
Diagnostic bytes	0
Housing	
Material	PPE
Mounting	Profile rail 35 mm
Mechanical data	
Dimensions (WxHxD)	25.4 x 76 x 78 mm

Order number	**240-1EA20**
Weight	80 g
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL508 certification	yes

Note!

Please regard that for the usage of the module national guidelines must be kept!

The adherence of these guidelines is incumbent on the user!

Chapter 3 Deployment

Overview

Here you will find the deployment of the communication processor CP 240 EnOcean.

Contents

Topic **Page**

Basics

EnOcean

EnOcean is a battery free radio system that has been developed by the company EnOcean in 2001. Due to the short signal length of 0.5ms and 10mW transmitting power the radio system technique has an energy requirement of 50µWs. For this the system uses the energy of smallest changes of pressure and temperature as power supply for the sensors.

The reach of the sensors is up to 300m out of doors. Additionally, every transmitter gets an unique 32Bit address as ID during manufacturing. The modules are using the internationally accredited SRD frequency band an 869 MHz.

Main points of usage of EnOcean are building automation, industrial production and automotive.

Properties

- Minimal energy requirements
- Support of several transmitters in the immediate environs
- Telegram length 0.5ms
- Transfer reach up to 300m
- Mono- and bi-directional communication
- Easy extensibility

Amplitude modulation

As modulation procedure EnOcean uses the incoherent amplitude - modulation (ASK). The error likeliness is nearly the same compared with the frequency modulation at identical interference signal level. The digital amplitude modulation allows the realization of energy saving transmitters because here only the "1"-Bits are transferred.

Security by means of telegram repetition

The transfer of a data telegram takes about 0.5ms. To enhance data security every telegram is repeated two times within 40ms, whereat the time lag between every repetition is perchancely.

This fast multiple sending allows that many neighborhood transmitters may be working parallel together on one radio frequency with a low error ratio.

IDs for addressing

EnOcean uses IDs for the addressing. An ID is an compound of *ID base* and a freely configurable *bit area*. Since the EnOcean modules are delivered by VIPA with a different *ID base* with extensive projects it is recommended to note all *ID base* of the modules. So on error an module can be replaced and the appropriate *ID base* can be taken.

For this details can be found at "Exchange module and set ID base".

Fast introduction

Overview

The communication processor CP 240 EnOcean enables the process coupling to different destination or source systems based upon the wireless EnOcean communication.

The CP 240 EnOcean is supplied with voltage via the back plane bus. For the internal communication the VIPA FCs are used. For the project engineering of the CP 240 EnOcean together with a CPU 21x in the Siemens SIMATIC Manager, the inclusion of the GSD VIPA_21x.gsd is required. To enable the CP 240 EnOcean to communicate with the CPU, a hardware configuration for the system is always necessary.

A general description for the project engineering of the CP 240 is to be found in "Project engineering".

Approach

Preparation

- Start the Siemens SIMATIC Manager with a new project.
- Include the VIPA_21x.gsd. For this, use a GSD version V. 1.67 or higher.
- Include the block library by extracting *Vipa_Bibliothek_Vxxx.zip* and de-archiving VIPA.ZIP.
- Open the library and transfer the corresponding FCs into your project.

Hardware configuration

Please follow for the hardware configuration the steps described in the manual HB97 - CPU:

- Configure a PROFIBUS-DP master system with the Siemens CPU 315-2DP (6ES7 315-2AF03 V1.2) and create a PROFIBUS subnet.
- Add to the master system the slave system "VIPA_CPU21x" from the hardware catalog. This is listed in the hardware catalog under *PROFIBUS-DP > Additional field devices > I/O > VIPA_System_200V.*
- Assign the address 1 to the slave system. With this, the VIPA CPU identifies the system as central periphery system.
- Within this slave system, you place your modules in the plugged sequence. Start with the CPU at the first plug-in location.
- Then include your System 200V modules and at the correct place the CP 240 EnOcean.
- Parameterize your CP 240 EnOcean.

Parameters

By placing the CP 240 EnOcean in the hardware configuration into the "virtual" PROFIBUS system, the required parameters are automatically created. The parameter area has the following structure:

Byte	Function	Value range	Default parameter
0	reserved		
1	Protocol	E0h: EnOcean	-
2...15	reserved		

You have only to set E0h in Byte 1 as protocol for EnOcean. The other parameters are reserved and not evaluated.

Internal communication

With the help of VIPA-FCs you control the communication between CPU and CP 240. For this, send and receive data have each a reserved 2048Byte buffer which may handle up to 150 telegrams. Together with a CPU 21x the following handling blocks are used:

Label	FCs	Description
SEND	FC0	send block
RECEIVE	FC1	receive block
SYNCHRON_RESET	FC9	reset and synchronization of the CP 240

11Byte telegram for EnOcean communication

Always use telegrams with a length of 11Byte for the communication. At the transmission, the CP 240 EnOcean extends the 11Byte automatically with 2 synchronization bytes and a Checksum to 14Byte res. cuts the 14Byte telegram to 11Byte at reception.

Include GSD and FCs

Project engineering via GSD

The address allocation and he parameterization of the CP 240 happens by means of the Siemens SIMATIC Manager in form of a virtual PROFIBUS system. Since the PROFIBUS interface is software standardized, the inclusion of a GSD file enables the guaranteed functionality of running in the SIMATIC Manager from Siemens at any time. Transfer your project via MPI into CPU.

Include GSD

The following steps are required for the installation of the GSD:

- In the service area of www.vipa.com a GSD file for the System 200V may be found. Load the zip file to your PC.
- Start your un-zip application with a double click on the file and un-zip the files to work directory.
- Copy the GSD file **`VIPA_21X.GSD`** into your GSD directory
  ... \siemens\step7\s7data\gsd
- Start the hardware configurator from Siemens
- Close all projects
- Select **Options** > *Install new GSD-file*
- Set here **VIPA_21X.gsd**

Now the modules of the System 200V from VIPA are integrated into the hardware catalog and may be used.

Installing blocks

The VIPA specific blocks may be found at www.vipa.com as downloadable library at the service area. The library is available as packed zip-file.

If you want to use VIPA specific blocks, you have to import the library into your project.

Retrieve library

Start your un-zip application with a double click on the file Vipa_ Bibliothek_ Vxxx.zip and copy the file vipa.zip to your work directory. It is not necessary to extract this file, too.

To retrieve your library for the SPEED7-CPUs, start the SIMATIC manager from Siemens. Open the dialog window for archive selection via **File** > *Retrieve*. Navigate to your work directory.

Choose VIPA.ZIP and click at [Open].

Select a destination folder where the blocks are to be stored. [OK] starts the extraction.

Open library and transfer blocks to project

After the extraction open the library.

Open your project and copy the necessary blocks from the library into the directory "blocks" of your project.

Now you have access to the VIPA specific blocks via your user application.

Project engineering

General

The address allocation and he parameterization of the directly plugged System 200V modules happens by means of the Siemens SIMATIC Manager in form of a virtual PROFIBUS system. You transfer your project into the CPU serial via the MPI interface or directly via MMC.

Requirements

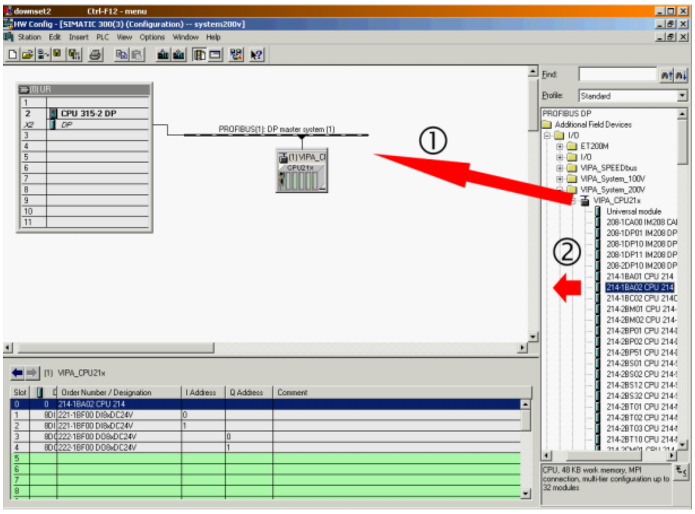
For the project engineering of the CPU a thorough knowledge of the SIMATIC Manager and the hardware configurator from Siemens is required!

For the project engineering the following preconditions must be fulfilled:

- SIMATIC Manager from Siemens is installed at PC res. PG
- GSD files are included into hardware configurator from Siemens
- The project can be transferred into CPU (serial e.g. "Green Cable" or MMC)

Hardware configuration

- Start the hardware configurator from Siemens with a new project and insert a profile rail from the hardware catalog.
- At the first available slot you place the CPU 315-2DP (6ES7 315-2AF03 V1.2) from Siemens.
- If your CPU 21x has an integrated PROFIBUS-DP master, you may now connect it to PROFIBUS and include your DP slaves.
- Create a PROFIBUS subnet (if not present yet).
- Add the system "VIPA_CPU21x" to the subnet. You will find this in the hardware catalog under *PROFIBUS DP* > *Additional field devices* > *IO* > VIPA_System_200V. Assign the **PROFIBUS address 1** to this slave.
- In your configurator, place the CPU 21x, which you are using, **always on the 1. slot** by taking it from the hardware catalog.
- Then you include your System 200V modules in the plugged sequence and your CP 240 at the according place.
- If necessary parameterize your CP 240.
- Save your project.



Slot		Order Number / Designation	I Address	Q Address	Comment
0	0	214-1BA02 CPU 214			
1	8DI	221-1BF00 DI8xDC24V	0		
2	8DI	221-1BF00 DI8xDC24V	1		
3	8DO	222-1BF00 DO8xDC24V		0	
4	8DO	222-1BF00 DO8xDC24V		1	
5					
6					
7					
8					

CPU, 48 KB work memory, MPI connection, multi-tier configuration up to 32 modules

PLC program

For the communication between CPU and CP 240 shown in the text below, the following handling blocks are used:

FC 0	SEND	Data output CPU to CP 240
FC 1	RECEIVE	Receive data from CP 240
FC 9	SYNCHRON_RESET	Synchronization between CPU and CP 240

The handling blocks are available as library and may be integrated into the Siemens SIMATIC Manager like shown above.

A more detailed description of the handling blocks is to be found on the following pages. Your PLC program should be build-up with the following structure:

```
OB1:
      CALL  FC     9             //Call Synchron
       ADR       :=0             //1st DW in SEND/EMPF_DB
       TIMER_NR  :=T2            //Delay time Synchron
       ANL       :=M3.0          //Start-up running
       NULL      :=M3.1          //Interim flag
       RESET     :=M3.2          //Execute module reset
       STEUERB_S :=MB2           //Control bits Send_FC
       STEUERB_R :=MB1           //Control bits Receive_FC
      U     M      3.0           //as long as no start-up no
                                   //SEND/RECEIVE processing
      BEB

      CALL  FC     1             //Receive data
       ADR          :=0          //1st DW in SEND/RECEIVE_DB
       _DB          :=DB11       //Receive_DB telegram
       ABD          :=W#16#14    //1st DW receive buffer (DW20)
       ANZ          :=MW10       //Amount of received data
       EMFR         :=M1.0       //Reception ready
       PAFE         :=MB12       //Error byte
       GEEM         :=MW100      //Internal data
       ANZ_INT      :=MW102      //Internal data
       empf_laeuft  :=M1.1       //Internal data
       letzter_block:=M1.2       //Internal data
       fehl_empf    :=M1.3       //Internal data
      U     M      1.0           //Reception ready
      R     M      1.0           //delete reception ready
      CALL  FC     0             //Send data
       ADR          :=0          //1st DW in SEND/RECEIVE_DB
       _DB          :=DB10       //Send_DB telegram
       ABD          :=W#16#14    //1st DW send buffer (DW20)
       ANZ          :=MW14       //Amount of data to send
       FRG          :=M2.0       //Set send ready
       PAFE         :=MB16       //Error byte
       GESE         :=MW104      //Internal data
       ANZ_INT      :=MW106      //Internal data
       ende_kom     :=M2.1       //Internal data
       letzter_block:=M2.2       //Internal data
       senden_laeuft:=M2.3       //Internal data
       fehler_kom   :=M2.4       //Internal data


OB100:
      UN    M      3.0
      S     M      3.0           //Start-up CPU running
```

Transfer project

The data transfer happens via MPI. If your programming device is not provided with a MPI interface you may also use a serial point-to-point transfer from your PC to MPI with the help of the "Green Cable" from VIPA.

The "Green Cable" has the order no. VIPA 950-0KB00 and may only be used with the VIPA CPUs with MP^{2}I interface.

Please regard for this also the hints for the usage of the Green Cable in the basics!

- Connect your PG with the CPU.
- Via **PLC** > *Load to module* in your project engineering tools you transfer the project into the CPU.
- Plug-in a MMC and transfer your user application to the MMC by means of **PLC** > *Copy RAM to ROM*.
- During the write process the "MC"-LED at the CPU is blinking. Due to system reasons a successful write process is announced too early. Please wait until the LED extinguishes.

What is the Green Cable?

The Green Cable is a green connection cable made exclusively for the deployment at VIPA System components.

The Green Cable allows you to:

- transfer project serially from point-to-point
- execute firmware updates of the CPUs and field bus master

Important hints for the deployment of the Green Cable

Non-observance of the following hints may cause damages to the system components.

For damages caused by non-observance of these hints and at incorrect usage, VIPA does not assume liability!

Hints for the operating range

The Green Cable may exclusively be deployed directly at the supposed jacks of the VIPA components (adapter plugs are not permissible). For example you have to pull a plugged MPI cable before connecting a Green Cable.

At this moment the following components supports the Green Cable:

VIPA CPUs with MP^{2}I jack as well as the field bus master from VIPA.

Notes to the lengthening

The lengthening of the Green Cable with another Green Cable res. the combination with other MPI cables is not permissible and causes damages to the connected components!

The Green Cable may only be lengthened with a 1:1 cable (all 9 pins are connected 1:1).

Standard handling blocks

SEND (FC 0)

This FC serves the data output from the CPU to the CP 240. Here you define the send range via the identifiers _DB, ADB and ANZ.

Via the bit FRG the send initialization is set and the data is send. After the data transfer the handling block sets the bit FRG back again.

Declaration	Name	Type	Comment
in	ADR	INT	Logical Address
in	_DB	BLOCK_DB	DB No. of DB containing data to send
in	ABD	WORD	No. of 1. data word to send
in	ANZ	WORD	No of bytes to send
in_out	FRG	BOOL	Start bit of the function
in_out	GESE	WORD	internal use
in_out	ANZ_INT	WORD	internal use
in_out	ENDE_KOMM	BOOL	internal use
in_out	LETZTER_BLOCK	BOOL	internal use
in_out	SENDEN_LAEUFT	BOOL	Status of function
in_out	FEHLER_KOM	BOOL	internal use
out	PAFE	BYTE	Return Code (00=OK)

ADR

Periphery address with which you may call the CP 240. Via the hardware configuration you may set the periphery address.

_DB

Number of the data block, which contains the data to send.

ABD

Word variable that contains the number of the data word from where on the characters for output are stored.

ANZ

Number of the bytes that are to be transferred.

FRG enable send

At FRG = "1" the data defined via _DB, ADB and ANZ are transferred once to the CP addresses by ADR. After the transmission the FRG is set back again. When FRG = "0" at call of the block, it is left immediately!

PAFE

At proper function, all bits of this bit memory byte are "0". At errors an error code is entered. The error setting is self-acknowledging, i.e. after elimination of the error cause, the byte is set back to "0" again. The following errors may occur:

1 = Data block not present

2 = Data block too short

3 = Data block number outside valid range

GESE, ANZ_INT
ENDE_KOM
LETZTER_BLOCK
SENDEN_LAEUFT
FEHLER_KOM

These parameters are internally used. They serve the information exchange between the handling blocks. For the deployment of the SYNCHRON_RESET (FC9) the control bits ENDE_KOM, LETZTER _BLOCK, SENDEN_LAEUFT and FEHLER_KOM must always be stored in a bit memory byte.

RECEIVE (FC 1)

This FC serves the data reception of the CP 240. Here you set the reception range via the identifiers _DB and ADB.

When the output EMFR is set, a new telegram has been read completely. The length of the telegram is stored in ANZ. After the evaluation of the telegram this bit has to be set back by the user, otherwise no further telegram may be taken over by the CPU.

Declaration	Name	Type	Comment
in	ADR	INT	Logical Address
in	_DB	BLOCK_DB	DB No. of DB containing received data
in	ABD	WORD	No. of 1. data word received
out	ANZ	WORD	No of bytes received
out	EMFR	BOOL	1=data received, reset by user
in_out	GEEM	WORD	internal use
in_out	ANZ_INT	WORD	internal use
in_out	EMPF_LAEUFT	BOOL	Status of function
in_out	LETZTER_BLOCK	BOOL	internal use
in_out	FEHLER_EMPF	BOOL	internal use
out	PAFE	BYTE	Return Code (00=OK)

ADR

Periphery address for calling the CP 240. You define the periphery address via the hardware configuration.

_DB

Number of the data block, which contains the data.

ABD

Word variable that contains the number of the data word from where on the received characters are stored.

ANZ

Word variable that contains the amount of received bytes.

EMFR

By setting of EMFR the handling block shows that data has been received. Not until setting back EMFR in the user application new data can be received.

PAFE

At proper function, all bits of this bit memory byte are "0". At errors an error code is entered. The error setting is self-acknowledging, i.e. after elimination of the error cause, the byte is set back to "0" again. The following errors may occur:

1 = Data block not present

2 = Data block too short

3 = Data block number outside valid range

GEEM, ANZ_INT
LETZTER_BLOCK
EMPF_LAEUFT
FEHLER_EMPF

These parameters are internally used. They serve the information exchange between the handling blocks. For the deployment of the SYNCHRON_RESET (FC9) the control bits LETZTER_BLOCK, EMPF_LAEUFT and FEHLER_EMPF must always be stored in a bit memory byte.

SYNCHRON_ RESET Synchronization and reset (FC 9)

The block must be called within the cyclic program section. This function is used to acknowledge the start-up ID of the CP 240 and thus the synchronization between CPU and CP. Furthermore it allows to set back the CP in case of a communication interruption to enable a synchronous start-up.

Note!

A communication with SEND and RECEIVE blocks is only possible when the parameter ANL of the SYNCHRON block has been set in the start-up OB before.

Declaration	Name	Type	Comment
in	ADR	INT	Logical Address
in	TIMER_NR	WORD	No of timer for idle time
in_out	ANL	BOOL	restart progressed
in_out	NULL	BOOL	internal use
in_out	RESET	BOOL	1 = Reset the CP
in_out	STEUERB_S	BYTE	internal use
in_out	STEUERB_R	BYTE	internal use

ADR

Periphery address with which you may call the CP 240. Via the hardware configuration you may set the periphery address.

TIMER_NR

Number of the timer for the delay time.

ANL

With ANL = 1 the handling block is informed that a STOP/START res. NETZ-AUS/NETZ-EIN has been executed at the CPU and now a synchronization is required. After the synchronization, ANL is automatically set back.

NULL

Parameter is used internally.

RESET

RESET = 1 allows you to set back the CP out of your user application.

STEUERB_S

Here you have to set the bit memory byte where the control bits ENDE_KOM, LETZTER_BLOCK, SENDEN_LAEUFT and FEHLER_KOM for the SEND-FC are stored.

STEUERB_R

Here you have to set the bit memory byte where the control bits LETZTER_BLOCK, EMPF_LAEUFT and FEHLER_EMPF for the RECEIVE-FC are stored.

Communication principle

Send and receive data

The CPU writes data via the back plane bus, which is to be sent, into the according data channel. The communication processor enters them into a ring buffer (2048Byte) and transmits them then via EnOcean.

When the communication processor receives data via EnOcean, the data is stored in a ring buffer (2048Byte). The received data may now be read telegram by telegram (11Byte) from the CPU via the data channel .

Communication via back plane bus

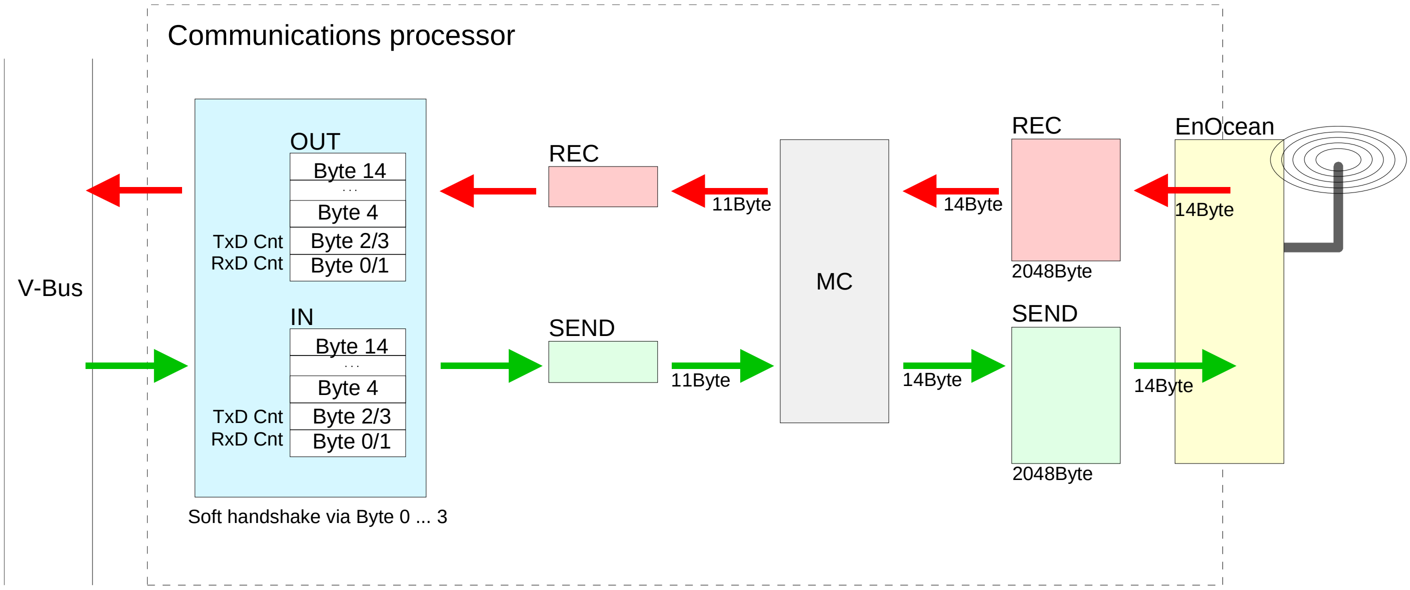
The exchange of received telegrams via back plane bus happens asynchronously. When a complete telegram has been arrived via EnOcean, it is stored in the buffer. The length of the ring buffer limits the maximum number of telegrams. At full buffer new telegrams are ignored.

Out of the telegrams of 14Byte length telegram by telegram 11Byte user data are transferred to the CPU via back plane bus. The first two sync bytes and the Checksum are not handled over.

Tasks of the CPU

A telegram that is to send has to be transferred to the CP 240. This supplements the telegram with the first two sync bytes and the Checksum and handles the telegram on to the send buffer. The CP 240 compiles these blocks in the send buffer and sends it via the EnOcean transceiver as soon as the telegram is complete. Since the data transfer via back plane bus happens asynchronously, a "software handshake" is used between CP 240 and CPU. The registers for the data transfer from the CP 240 have a width of 16Byte. For the handshake, the Bytes 0 to 3 (word 0 and 2) are reserved.

The following picture shall illustrate this:



Software handshake

For the deployment of the CP 240 together with a System 200V CPU VIPA offers handling blocks that enable a comfortable software handshake.

For the deployment of the CP 240 without handling blocks, the following text shows the functionality for transmitting and receiving data with an example.

Example transmitting data w/o handling blocks

An EnOcean telegram contains 11Byte user data. At the transmission the CPU writes for every telegram 11Byte user data into the Bytes 4 to 14 and into Byte 2/3 the length of the telegram (i.e. "11"). The CP 240 receives the data via the back plane bus. To acknowledge the telegram, the CP 240 writes the value "11" (length of the telegram) back to the CPU into Byte 2/3.

At reception of this "11" in Byte 2/3 the CPU sends back a "0" at Byte 2/3. Thereupon the user data in the CP 240 are supplemented to 14Byte with 2 sync bytes at the beginning and the Checksum at the end and stored in the send buffer. After this, the CP responds with a "0" at Byte 2/3. After the CPU received this "0", it may send a new telegram to the CP 240.

The telegrams stored in the send buffer are immediately transmitted via EnOcean.

Example receiving data without handling blocks

Every EnOcean telegram has a size of 14Byte. When the CP 240 receives a telegram this is stored in the receive buffer. For every telegram the 11Byte of user data are handled over to the CPU via the back plane bus into Byte 4 to 14 and the length (i.e. "11") into Byte 0/1. The first two sync bytes and the Checksum are deleted.

The CPU stores the user data and responds with the value "11" at Byte 0/1. The CP acknowledges this with a "0" at Byte 0/1 and thus announces that the transfer has been completed. As soon as new data may be transferred the CPU answers with "0".

With the reception of "0" the CP 240 may send a new telegram to the CPU.

Example for EnOcean deployment

Overview

In the following example an EnOcean communication (send and receive) is build-up. Furthermore the sample illustrates how you may easily establish the control over communication processes by using the handling blocks.

At need you may receive the example project from VIPA.

Requirements

The following components are required for the sample:

1 System 200V consisting of CPU 21x and CP 240 EnOcean

1 switch with EnOcean transmitter

Project engineering tool SIMATIC Manager from Siemens with transmitting cable

Approach

Build-up the System 200V.

Load the example project, if necessary adjust the periphery address and transfer the project into the CPU.

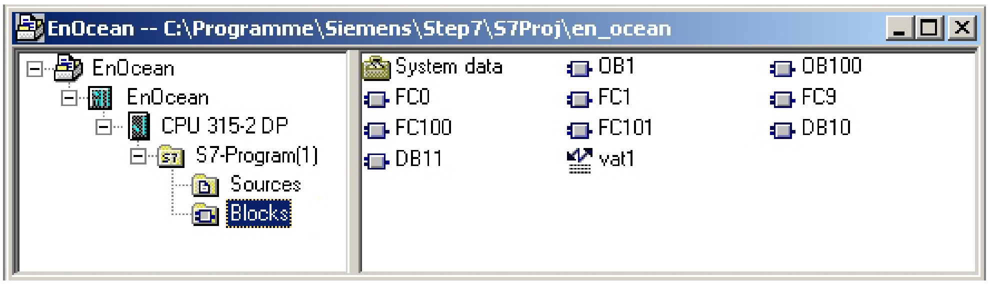
Dearchive the project

Follow these steps in the Siemens SIMATIC Manager:

- Start the Siemens SIMATIC Manager.
- To extract the file Enocean.zip select **File** > *de-archive*.
- Choose the example file Enocean.zip and set "s7proj" as destination directory.
- Open the extracted project.

Project structure

The project already contains the PLC application and the hardware configuration and has the following structure:



Data blocks The example uses the following data blocks:

DB10
Send data block

Addr.	**Label**	**Type**	**Comment**
0.0		STRUCT	
+0.0	Sendefach	STRUCT	Send data block
+0.0	RX_TX_Kennung	BYTE	0B=RX/6B=TX
+1.0	ORG	BYTE	
+2.0	Datenbyte3	BYTE	Data byte 3
+3.0	Datenbyte2	BYTE	Data byte 2
+4.0	Datenbyte1	BYTE	Data byte 1
+5.0	Datenbyte0	BYTE	Data byte 0
+6.0	IDbyte2_3	WORD	ID Byte 2 and 3
+8.0	IDbyte0_1	WORD	ID Byte 0 and 1
+10.0	Status	BYTE	Status
=12.0		END_STRUCT	
+12.0	Reserve	BYTE	
+13.0	SENDEN_LAEUFT	BOOL	Still transmitting
+13.1	LETZTER_BLOCK	BOOL	Last block has been sent
+13.2	FEHL_KOM	BOOL	Error during transmission
+13.3	ENDE_KOM	BOOL	Transfer complete
+14.0	PAFE	BYTE	Parameterization error byte of FC0
+15.0	Res00	BOOL	
+15.1	Res01	BOOL	
+15.2	Res02	BOOL	
+15.3	Res03	BOOL	
+15.4	Res04	BOOL	
+15.5	Res05	BOOL	
+15.6	Res06	BOOL	
+15.7	Senden_start	BOOL	Telegram transmitted completely
+16.0	GESE	WORD	Already sent data
+18.0	ANZ_INT	WORD	Amount of sent data
+20.0	Reserve1	ARRAY[0..50]	
*1.0		BYTE	
=72.0		END_STRUCT	

DB11
Receive data block

Addr.	Label	Type	Comment
0.0		STRUCT	
+0.0	Empfangsfach	STRUCT	Receive data block
+0.0	RX_TX_Kennung	BYTE	0B=RX/6B=TX
+1.0	ORG	BYTE	
+2.0	Datenbyte3	BYTE	Data byte 3
+3.0	Datenbyte2	BYTE	Data byte 2
+4.0	Datenbyte1	BYTE	Data byte 1
+5.0	Datenbyte0	BYTE	Data byte 0
+6.0	IDbyte2_3	WORD	ID byte 2 and 3
+8.0	IDbyte0_1	WORD	ID byte 0 and 1
+10.0	Status	BYTE	Status
=12.0		END_STRUCT	
+12.0	Reserve	BYTE	
+13.0	EMP_LAEUFT	BOOL	Still receiving
+13.1	LETZTER_BLOCK	BOOL	Last block has been received
+13.2	FEHL_EMPF	BOOL	Error during reception
+14.0	PAFE	BYTE	Parameterization error byte of FC1
+15.0	Res00	BOOL	
+15.1	Res01	BOOL	
+15.2	Res02	BOOL	
+15.3	Res03	BOOL	
+15.4	Res04	BOOL	
+15.5	Res05	BOOL	
+15.6	Res06	BOOL	
+15.7	Empfang_fertig	BOOL	Telegram received completely
+16.0	GEEM	WORD	Already received data
+18.0	ANZ_INT	WORD	Amount of received data
+20.0	Reserve1	ARRAY[0..50]	
*1.0		BYTE	
=72.0		END_STRUCT	

PLC program

The example already contains the PLC application and the hardware configuration. The following blocks are used:

OB 1

```
CALL  FC 9                     //Re-boot or Reset
 ADR      :=256                //Address of the module
 TIMER_NR :=T2
 ANL      :=M3.0
 NULL     :=M3.1
 RESET    :=M3.2
 STEUERB_S:=MB4
 STEUERB_R:=MB6
U  M  3.0                      //Reception can only be started
BEB                            //after execution of FC 9 (SYNCHRON_RESET)
CALL FC 100                    //Call of reception-FC
```

OB 100

```
L  0
T  MB 1                        //delete order bit
UN M  3.0                      //Initialize re-boot
S  M  3.0
```

FC 100

```
       CALL  FC 1
        ADR  :=256                                //Address of the module
        _DB  :="EMPFANG_en_ocean"                 //DB with received data
        ABD  :=W#16#0                             //1. DBB received data
        ANZ  :=W#16#B                             //reception length always 11
        EMFR :=M7.0                               //all data received
        PAFE :="EMPFANG_en_ocean".Pafe            //error byte
        GEEM :="EMPFANG_en_ocean".GEEM            //received amount (internal)
        ANZ_INT:="EMPFANG_en_ocean".ANZ_INT       //reception length (internal)
        EMPF_LAEUFT:="EMPFANG_en_ocean".EMP_LAEUFT    //receiving data (internal)
        LETZTER_BLOCK:="EMPFANG_en_ocean".LETZTER_BLOCK    //all data received
        FEHL_EMPF:="EMPFANG_en_ocean".FEHL_EMPF   //Error in the
                                                  //reception routine
       UN    M         7.0                        //no telegram received
       BEB                                        //then end
       R     M         7.0                        //delete reception bit
       L     "EMPFANG_en_ocean".Empfangsfach.IDbyte0_1 //switch ID
       L     W#16#1C7A                            //Please enter here the ID
       ==I                                        //of your switch.
       SPB   e_a2                                 //This can be taken from
       BEA                                        //DB 11.DBW 8

e_a2:  NOP   0
       L     5                                    //ID switch on
       L     "EMPFANG_en_ocean".Empfangsfach.Datenbyte3    //Byte with ID
       SRW   4                                    //ID in Low nipple
       ==I                                        //Proof if switch is pushed
       SPB   ein
       L     7                                    //ID switch of
       ==I                                        //Proof if switch is pushed
       SPB   aus
       BEA

ein:   NOP   0
       S     A         0.0                        //Function on
       BEA

aus:   NOP   0
       R     A         0.0                        //Function off
       BEA
```

FC 101

```
L     B#16#6B                                     //allocate send data
T     "SEND_en_ocean".Empfangsfach.RX_TX_Kennung    //send ID
L     B#16#5                                      //ORG-ID
T     "SEND_en_ocean".Empfangsfach.ORG
L     B#16#2
T     "SEND_en_ocean".Empfangsfach.Datenbyte3
L     0
T     "SEND_en_ocean".Empfangsfach.Datenbyte2
T     "SEND_en_ocean".Empfangsfach.Datenbyte1
T     "SEND_en_ocean".Empfangsfach.Datenbyte0
T     "SEND_en_ocean".Empfangsfach.IDbyte2_3
L     W#16#3267                                   //Only the last 7Bit
T     "SEND_en_ocean".Empfangsfach.IDbyte0_1        //are relevant for addr.
L     6                                           //and are "ORed" in the
T     "SEND_en_ocean".Empfangsfach.Status           //CP 240 with the here
                                                  //stored ID-Base
CALL  FC 0
 ADR          :=256
 _DB          :="SEND_en_ocean"
 ABD          :=W#16#0                         //send from data byte 0 on
 ANZ          :=W#16#B                         //always 11Byte
 PAFE         :="SEND_en_ocean".Pafe
 FRG          :="SEND_en_ocean".Senden_start
 GESE         :="SEND_en_ocean".GEEM
 ANZ_INT      :="SEND_en_ocean".ANZ_INT
 ENDE_KOM     :="SEND_en_ocean".ENDE_KOM
 LETZTER_BLOCK:="SEND_en_ocean".LETZTER_BLOCK
 SENDEN_LAEUFT:="SEND_en_ocean".SENDEN_LAEUFT
 FEHLER_KOM   :="SEND_en_ocean".FEHL_KOM
```

Overview of the EnOcean telegrams

General structure

The following table shows the general structure of an EnOcean telegram. Send and receive telegrams have the same structure. They only differ in the ID.

Bit 7 Bit 0

0xA5	These bytes are automatically created at transmission and hidden at reception.
0x5A	
0x0B 0x6B	0x0B: ID for reception telegram 0x06: ID for send telegram
ORG	See table *supported ORG formats*
DataBytes3	Data from a sensor res. to an actuator
DataBytes2	
DataBytes1	
DataBytes0	
IDBytes3*	ID of the transceiver module. With SET_IDBASE you may alter the ID up to 10 times.
IDBytes2*	
IDBytes1*	
IDBytes0*	
Status	Status information of the according sensor
Checksum	Is automatically created at transmission and hidden at reception.

*) During transmission the actual ID base of the module replaces the ID base in the telegram.

General

At the following pages all telegrams are listed that are supported by the CP 240 EnOcean. This description has been taken directly out of the documentation by courtesy of EnOcean.

Note!

Please regard that the first two synchronization bytes and the Checksum of received telegrams are not stored in the CP 240. At transmission, the 11Byte user data are automatically supplemented with these bytes to a total size of 14Byte.

Description of ORG field

The TX_TELEGRAM and RX_TELEGRAM telegrams have the same structure. The only difference is that a TX_TELEGRAM is identified by "3” in H_SEQ instead of "0” for an RX_TELEGRAM.

ORG	Description	RRT / TRT Acronym
0x05	Telegram from a PTM switch module received (original or repeated message)	RPS
0x06	1 byte data telegram from a STM sensor module received (original or repeated message)	1BS
0x07	4 byte data telegram from a STM sensor module received (original or repeated message)	4BS
0x08	Telegram from a CTM module received (original or repeated message)	HRC
0x0A	6byte Modem Telegram (original or repeated)	6DT
0x0B	Modem Acknowledge Telegram	MDA

Serial command encoding for RPS, 1BS, 4BS, HRC

Bit 7 Bit 0

0xA5
0x5A
0x0B (RX_TELEGRAM) 0x6B(TX_TELEGRAM)
ORG
DataBytes3
DataBytes2
DataBytes1
DataBytes0
IDBytes3
IDBytes2
IDBytes1
IDBytes0
Status
ChkSum

DataBytes2= DataBytes1= DataBytes0= 0x00

for RPS,1BS, HRC

Serial command encoding for 6DT

Bit 7 Bit 0

0xA5
0x5A
0x0B (RX_TELEGRAM) 0x6B(TX_TELEGRAM)
0x0A
DataBytes5
DataBytes4
DataBytes3
DataBytes2
DataBytes1
DataBytes0
Address1
Address0
Status
ChkSum

Serial command encoding for MDA

Bit 7 Bit 0

0xA5
0x5A
0x0B (RX_TELEGRAM) 0x6B(TX_TELEGRAM)
0x0B
0xXX
0xXX
0xXX
0xXX
Address1
Address0
0xXX
0xXX
Status
ChkSum

Description of STATUS field

If ORG = 0x05 (Telegram from a PTM switch module)

7			**0**
Reserved	**T21**	**NU**	**RP_COUNTER**

Reserved	(2 bit)	Do not care
T21	(1 bit)	T21=0 → PTM type 1, T21=1 → PTM type 2

Note: In transmission the TCM 120 always sets T21=1
→ it is only possible to transmit PTM type 2 telegrams!

NU	(1 bit)	NU=1 → N-message, NU=0 → U-message.
RP_COUNTER	(4 bit) =0..15	Repeater level: 0 is original message

IMPORTANT NOTE

Within toggle switch applications using the RCM 120 or TCM 120 serial receiver mode in combination with the TCM 110 repeater module, please ensure that no serial command interpretation error may occur at the connected control unit. A toggle signal means that the same telegram (from e.g. PTM 100, PTM 200 or STM 100) is sent for switching something on and off. If e.g. the light is switched on by means of a RCM 120 receiving the I-button telegram from a PTM 100, the repeated telegram (delay <100ms) may switch off the light again. It is therefore mandatory to interpret the RP_COUNTER field as described in the RCM 120 User Manual. If a repeated telegram (RP_COUNTER>0) is received it has to be verified if the same telegram with a lower RP_COUNTER state has already been received in the previous 100 ms. In this case the repeated message has to be discarded.

PTM Type 1

PTM switch modules of Type 1 (e.g. PTM 100) do not support interpretation of operating more than one rocker at the same time:

N-message received → Only one pushbutton was pressed.

U-message received → No pushbutton was pressed when activating the energy generator, or more than one pushbutton was pressed.

PTM Type 2

PTM switch modules of Type 2 allow interpretation of operating two buttons simultaneously:

N-message received → Only one or two pushbuttons have been pressed.

U-message received → No pushbutton was pressed when activating the energy generator, or more than two pushbuttons have been pressed.

Note for telegrams from PTM 100 piezo transmitters:

Due to the mechanical hysteresis of the piezo energy bow, in most rocker switch device implementations, pressing the rocker sends an N-message and releasing the rocker sends a U-message!

If ORG = 0x06, 0x07, 0x08 or 0x0A:

7	**0**
Reserved	**RP_COUNTER**

Reserved	(4 bit)	Do not care
RP_COUNTER	(4 bit)	Repeater level: 0 original message 1 repeated message

Description of DATA_BYTE 3..0

If ORG = 0x05 and NU = 1 (N-message from a PTM switch module):

DATA_BYTE2..0	always = 0
DATA_BYTE3	as follows:

7					**0**
RID	**UD**	**PR**	**SRID**	**SUD**	**SA**

RID	(2 bit)	Rocker ID, from left (A) to right (D): 0, 1, 2 and 3 (decimal)
UD	(1 bit)	UD=1 → O-button, UD=0 → I-button

PR	(1 bit)	PR=1 → energy bow pressed
		PR=0 → energy bow released

SRID	(2 bit)	Second Rocker ID, from left to right: 0, 1, 2 and 3
SUD	(1 bit)	(Second) SUD=1 → O-button, SUD=0 → I-button
SA	(1 bit)	SA=1 → Second action (2 buttons pressed simultaneously), SA=0 → No second action

If ORG = 0x05 and NU = 0 (U-message from a PTM switch module):

DATA_BYTE2..0	always = 0
DATA_BYTE3	as follows:

7		**0**
BUTTONS	**PR**	**Reserved**

BUTTONS	(3 bit)	Number of simultaneously pressed buttons, as follows:

PTM 100	PTM200
0 = 0 Buttons	0 = 0 Button
1 = 2 Buttons	1 = not possible
2 = 3 Buttons	2 = not possible
3 = 4 Buttons	3 = 3 or 4 buttons
4 = 5 Buttons	4 = not possible
5 = 6 Buttons	5 = not possible
6 = 7 Buttons	6 = not possible
7 = 8 Buttons	7 = not possible

PR	(1 bit)	PR = 1 → energy bow pressed
		PR = 0 → energy bow released

Reserved	(4 bit)	for future use

If ORG = 0x06 (Telegram from a 1 Byte STM sensor):

DATA_BYTE2..0	always = 0
DATA_BYTE3	Sensor data byte.

If ORG = 0x07 (Telegram from a 4 Byte STM sensor):

DATA_BYTE3	Value of third sensor analog input
DATA_BYTE2	Value of second sensor analog input
DATA_BYTE1	Value of first sensor analog input
DATA_BYTE0	Sensor digital inputs as follows:

7				0
Reserved	**DI_3**	**DI_2**	**DI_1**	**DI_0**

If ORG = 0x08 (Telegram from a CTM module set into HRC operation):

DATA_BYTE2..0	always = 0
DATA_BYTE3	as follows:

7				0
RID	**UD**	**PR**	**SR**	**Reserved**

RID	(2 bit)	Rocker ID, from left (A) to right (D): 0, 1, 2 and 3
UD	(1 bit)	UD=1 → O-button, UD=0 → I-button
PR	(1 bit)	PR=1 → Button pushed, PR=0 → Button released
SR	(1 bit)	SR=1 → Store, SR=0 → Recall (see note)
Reserved	(3 bit)	for future use

Note: The SR bit is used only when the lower 3 bits from ID_BYTE0 = B'111' (scene switch), and RID ≠ 0 (indicates that the memory buttons M0-M6 are operated in the handheld remote control).

If ORG = 0x0A (Modem telegram):

Please note the different structure of modem telegrams with 6 data bytes and 2 address bytes for the ID of the receiving modem. See A.1.1.

Command Telegrams and Messages

INF_INIT

After a power-on, a hardware reset or a RESET command the TCM informs the user through several of these telegrams about the current status. The messages have the general syntax as shown. The information contained by the bytes marked as X should be decoded according to ASCII code.

Bit 7 *Bit 0*

0xA5
0x5A
0x8B
0x89
X
X
X
X
X
X
X
X
X
ChkSum

In total there are 15 telegrams:

0xA5 0x5A 0x8B 0x89	" ”
0xA5 0x5A 0x8B 0x89	"EnOcean”
0xA5 0x5A 0x8B 0x89	"TCM120”
0xA5 0x5A 0x8B 0x89	"Version”
0xA5 0x5A 0x8B 0x89	Version number in ASCII
0xA5 0x5A 0x8B 0x89	*"0x40” (9600 baud)*
0xA5 0x5A 0x8B 0x89	"Bdrate”
0xA5 0x5A 0x8B 0x89	"Modem”
0xA5 0x5A 0x8B 0x89	"ON” or "OFF”
0xA5 0x5A 0x8B 0x89	"RxID”
0xA5 0x5A 0x8B 0x89	modem ID in ASCII
0xA5 0x5A 0x8B 0x89	"Mode”
0xA5 0x5A 0x8B 0x89	"Run”
0xA5 0x5A 0x8B 0x89	"PrgMem”
0xA5 0x5A 0x8B 0x89	"OK” or "CORRUPT”

OK

Standard message used to confirm that an action was performed correctly by the TCM.

Bit 7 *Bit 0*

0xA5
0x5A
0x8B
0x58
X
X
X
X
X
X
X
X
X
ChkSum

ERR

Standard error message response if after a TCT command the operation could not be carried out successfully by the TCM.

Bit 7 Bit 0

0xA5
0x5A
0x8B
0x19
X
X
X
X
X
X
X
X
X
ChkSum

RD_IDBASE

When this command is sent to the TCM, the base ID range number is retrieved though an INF_IDBASE telegram.

Bit 7 Bit 0

0xA5
0x5A
0xAB
0x58
X
X
X
X
X
X
X
X
X
ChkSum

SET_IDBASE

With this command the user can rewrite its ID range base number. The most significant ID byte is IDBaseByte3**.** The information of the 25 most significant bits is stored in EEPROM.

The allowed ID range is from 0xFF800000 to 0xFFFFFFFF.

Bit 7 Bit 0

0xA5
0x5A
0xAB
0x18
IDBaseByte3
IDBaseByte2
IDBaseByte1
IDBaseByte0
X
X
X
X
X
ChkSum

32							**0**	
25 most significant bits	**0**	**0**	**0**	**0**	**0**	**0**	**0**	ID range base

This command can only be used a maximum number of 10 times. After successfully ID range reprogramming, the TCM answers with an OK telegram. If reprogramming was not successful, the TCM answers sending an ERR telegram if the maximum number of 10 times is exceeded or an ERR_IDRANGE telegram if the ID range base is not within the allowed range.

INF_IDBASE

This message informs the user about the ID range base number.

Bit 7 Bit 0

0xA5	
0x5A	
0x8B	
0x98	
IDBaseByte3	IDBaseByte3 is the most significant byte.
IDBaseByte2	
IDBaseByte1	
IDBaseByte0	
X	
X	
X	
X	
X	
ChkSum	

SET_RX_SENSITIVITY

This command is used to set the TCM radio sensitivity.

In LOW radio sensitivity, signals from remote transmitters are not detected by the TCM receiver. This feature is useful when only information from transmitters in the vicinity should be processed. An OK confirmation telegram is generated after TCM sensitivity has been changed.

Bit 7 Bit 0

0xA5
0x5A
0xAB
0x08
Sensitivity
X
X
X
X
X
X
X
X
ChkSum

Sensitivity=0x00 Low sensitivity

Sensitivity=0x01 High sensitivity

RD_RX_SENSITIVITY

This command is sent to the TCM to retrieve the current radio sensitivity mode (HIGH or LOW). This information is sent via an INF_RX_ SENSITIVITY command.

Bit 7 Bit 0

0xA5
0x5A
0xAB
0x48
X
X
X
X
X
X
X
X
X
ChkSum

INF_RX_SENSITIVITY

This message informs the user about the current TCM radio sensitivity.

Bit 7 Bit 0

0xA5
0x5A
0x8B
0x88
Sensitivity
X
X
X
X
X
X
X
X
ChkSum

Sensitivity= 0x00 Low sensitivity

Sensitivity= 0x01 High sensitivity

SLEEP

If the TCM receives the SLEEP command, it works in an energy-saving mode. The TCM will not wake up before a hardware reset is made or a WAKE telegram is sent via the serial interface.

Bit 7 Bit 0

0xA5
0x5A
0xAB
0x09
X
X
X
X
X
X
X
X
X
ChkSum

WAKE

If the TCM receives the WAKE command, it wakes up from sleep mode. In contrast to all other telegrams this telegram is only one byte long.

Bit 7 Bit 0

0xAA

RESET

Performs a reset of the TCM micro controller. When the TCM is ready to operate again, it sends an ASCII message (INF_INIT) containing the current settings.

Bit 7 *Bit 0*

0xA5
0x5A
0xAB
0x0A
X
X
X
X
X
X
X
X
X
ChkSum

MODEM_ON

Activates TCM modem functionality and sets the modem ID. An OK confirmation telegram is generated. The modem ID is the ID at which the TCM receives messages of type 6DT. The modem ID and modem status (ON/OFF) is stored in EEPROM. The modem ID range is from 0x0001 to 0xFFFF. IF 0x0000 is provided as modem ID, the modem is activated with the ID previously stored in EEPROM.

Bit 7 Bit 0

0xA5
0x5A
0xAB
0x28
Modem ID (MSB)
Modem ID (LSB)
X
X
X
X
X
X
X
ChkSum

MODEM_OFF

Deactivates TCM modem functionality. When this command has been sent, an OK command should be received, confirming that the modem status is OFF. The modem ID is not erased.

Bit 7 Bit 0

0xA5
0x5A
0xAB
0x2A
X
X
X
X
X
X
X
X
X
ChkSum

RD_MODEM_ STATUS

This command requests the TCM to send information about its current modem current status. The requested information is reported to the user through an INF_MODEM_STATUS telegram.

Bit 7 Bit 0

0xA5
0x5A
0xAB
0x68
X
X
X
X
X
X
X
X
X
ChkSum

INF_MODEM_ STATUS

Informs the user about the TCM current modem status. The information provided is the following: Modem status (ON or OFF) and modem ID stored.

Modem state=0x01, modem ON

Modem state=0x00, modem OFF

Modem ID MSB= most significant modem ID byte.

Modem ID LSB=least significant modem ID byte.

Bit 7 Bit 0
0xA5
0x5A
0x8B
0xA8
Modem status
Modem ID MSB
Modem ID LSB
X
X
X
X
X
X
ChkSum

RD_SW_VER

This command requests the TCM to send its current software version number. This information is provided via an INF_SW_VER telegram by the TCM.

Bit 7 Bit 0
0xA5
0x5A
0xAB
0x4B
X
X
X
X
X
X
X
X
X
ChkSum

INF_SW_VER

Informs the user about the current software version of the TCM.

Bit 7 Bit 0

0xA5
0x5A
0x8B
0x8C
TCM SW Version Pos.1
TCM SW Version Pos.2
TCM SW Version Pos.3
TCM SW Version Pos.4
X
X
X
X
X
ChkSum

Example: Version 1.0.1.16
TCM SW Version Pos.1 = 1
TCM SW Version Pos.2 = 0
TCM SW Version Pos.3 = 1
TCM SW Version Pos.4 =16

ERR_MODEM_NOTWANTEDACK

When a 6DT modem telegram has been sent, the TCM waits for a modem acknowledge (MDA) telegram. This error message is generated if an MDA with the right modem ID is received after the timeout (100ms) or if there is more than one MDA received.

Bit 7 Bit 0

0xA5
0x5A
0x8B
0x28
X
X
X
X
X
X
X
X
X
ChkSum

ERR_MODEM_ NOTACK

When a 6DT modem telegram has been sent, the TCM waits for a modem acknowledge (MDA) telegram. This error message is generated if no acknowledge was received before the timeout (100ms).

Bit 7 Bit 0

0xA5
0x5A
0x8B
0x29
X
X
X
X
X
X
X
X
X
ChkSum

ERR_MODEM_ DUP_ID

When the TCM receives an original (not repeated) MDA telegram with the same modem ID as its own, it sends this message through the serial port and informs that at least 2 TCMs have the same modem ID. This is not necessarily a problem and may even be intended. On the other hand it may also indicate that there is another installation/building in the vicinity where the same modem ID is in use.

Bit 7 *Bit 0*

0xA5
0x5A
0x8B
0x0C
X
X
X
X
X
X
X
X
X
ChkSum

ERR_SYNTAX

This telegram is sent automatically through the serial port after the TCM has detected a syntax error in a TCT telegram.

Errors can occur in the H_SEQ, LENGTH, ORG or CHKSUM fields/bytes.

Bit 7 Bit 0

0xA5
0x5A
0x8B
Field
X
X
X
X
X
X
X
X
X
ChkSum

Field code:
H_SEQ=0x08 ORG=0x0B
LENGTH=0x09 CHKSUM=0x0A

ERR_TX_ IDRANGE

When a radio telegram intended to be sent has an ID number outside the ID range, this error message is generated. The radio telegram is not delivered.

Bit 7 Bit 0
0xA5
0x5A
0x8B
0x22
X
X
X
X
X
X
X
X
X
ChkSum

ERR_ IDRANGE

This message is generated when the user tries to change the ID range base using the SET_IDBASE command to a value outside the allowed range from 0xFF800000 to 0xFFFFFFFF.

Bit 7 Bit 0
0xA5
0x5A
0x8B
0x1A
X
X
X
X
X
X
X
X
X
ChkSum

Exchange module and set ID base

Overview

Since the ID base of every module is different, you have in case of replacement the option to change the ID base of a module for up to 10 times by means of a SET_IDBASE telegram. Consequently the newly adjustment of the actuators to the replacement module is no longer necessary. After successful transfer of the ID base you have either to reboot your CPU or reset it via FC 9.

Please regard that only the upper 25 bits are taken over as ID base. The remaining 7 bits you may specify via your user application during runtime and herewith address multiple actuators.

ID base request

With RD_IDBASE the current ID base of your module may be requested.

RD_IDBASE

0xAB	ID for transmission telegram
0x58	ORG ID for RD_IDBASE
X	Irrelevant
...	...
X	Irrelevant

INF_IDBASE

RD_IDBASE reports back the current ID base of the module in form on an INF_IDBASE telegram. The telegram has the following structure:

0x8B	ID for reception telegram
0x98	ORG-ID for INF_IDBASE
IDBaseByte3	Byte 3 current ID base
IDBaseByte2	Byte 2 current ID base
IDBaseByte1	Byte 1 current ID base
IDBaseByte0	Byte 0 current ID base (Bit 6...0 irrelevant)
X	irrelevant
...	...
X	irrelevant

SET_IDBASE

In case of replacement send a SET_IDBASE telegram with the following structure from your CPU to the module (transceiver). Use the address of the module you want to replace as new ID base:

0xAB	ID for transmission telegram
0x18	ORG ID for SET_IDBASE
IDBaseByte3	Byte 3 new ID base
IDBaseByte2	Byte 2 new ID base
IDBaseByte1	Byte 1 new ID base
IDBaseByte0	Byte 0 new ID base (Bit 6...0 irrelevant)
X	irrelevant
...	...
X	irrelevant

Possible respond telegram

0x8B **0x58**	←	**OK** – ID base was set

To take over the ID base during runtime you have to execute a reset via FC 9. Otherwise the new ID base is available after a reboot of the CPU.

In case of an error you receive one of the following messages. Here the old ID base remains valid.

0x8B **0x19**	←	**ERR** More than 10 ID base changes are not permitted
0x8B **0x1A**	←	**ERR_IDRANGE** ID base is outside the valid range (0xFF800000 ... 0xFFFFFFFF).

Control your ID settings and send the telegram once more.