VIPA System 200V

CP | Manual

HB97E_CP | RE_240-1FA20 | Rev. 12/42

October 2012

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.

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-FA20 that are available from VIPA. It contains detailed descriptions of the CP 240 M-Bus for communication with energy and excise counters.

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 M-Bus interface.

Chapter 3: Deployment

Here you will find the deployment of the communication processor CP 240 M-Bus.

Objective and contents

This manual describes the System 200V CP 240-1FA20 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 M-Bus	VIPA CP 240-1FA20	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.

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.

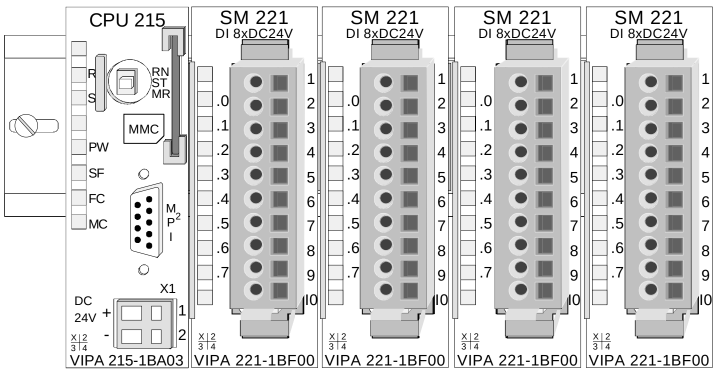
Attention!

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.

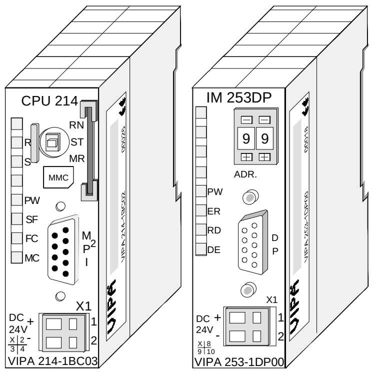


Components

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*

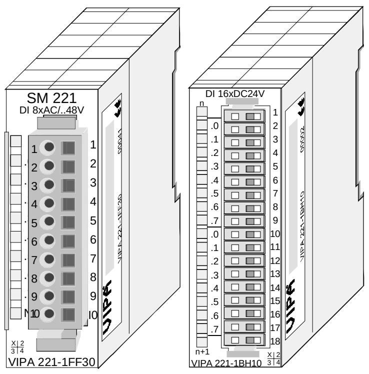
Head modules



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.

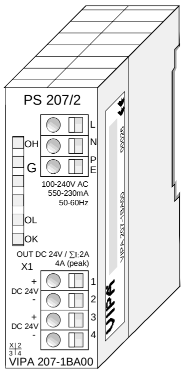
Periphery modules



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.

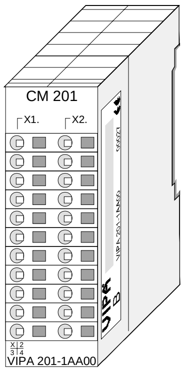
Power supplies



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 back-plane 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

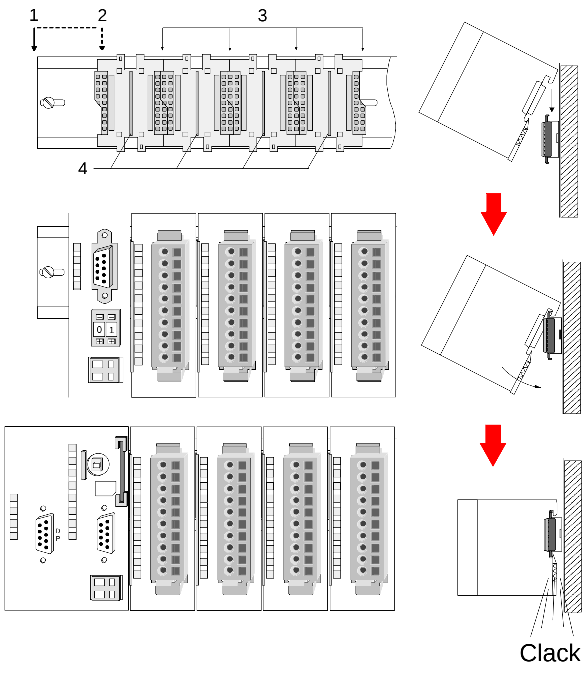
- Profile rail 35mm
- Dimensions of the basic enclosure:

  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

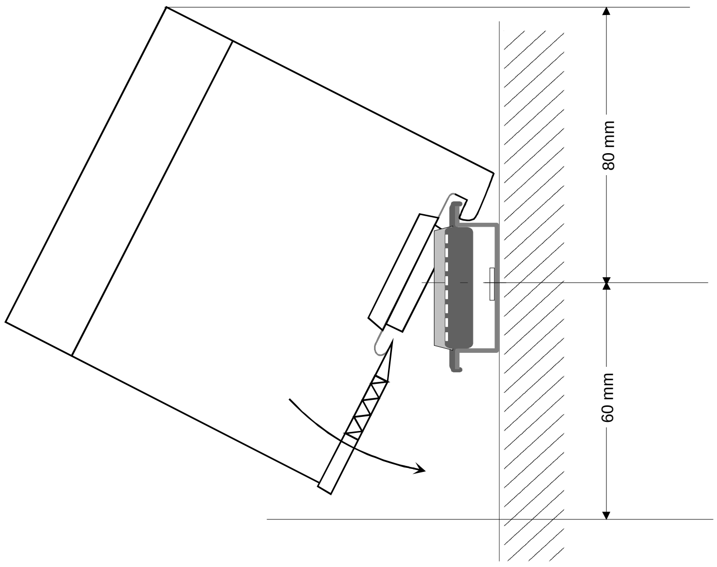
Information about the max. number of pluggable modules and the max. current at the backplane bus can be found in the "Technical Data" of the according head module.

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

Dimensions

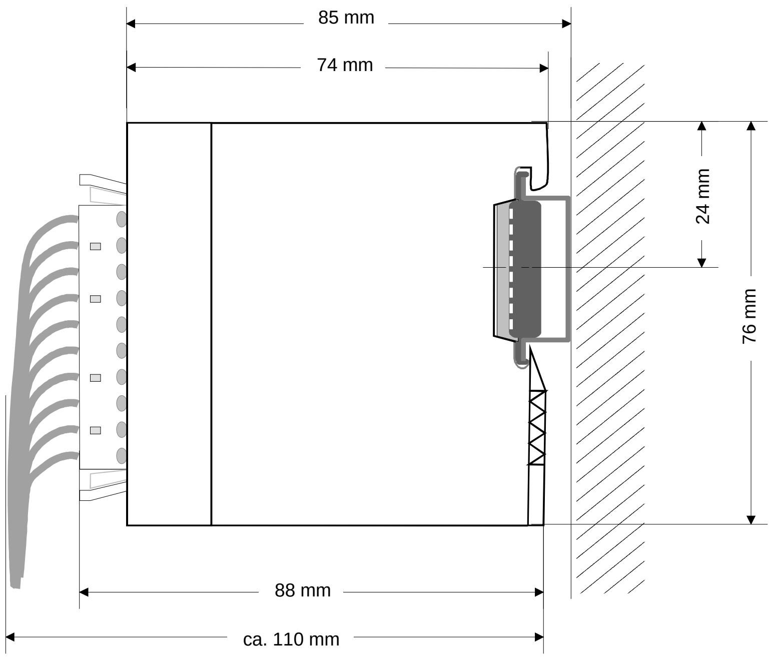
Dimensions Basic enclosure	1tier width (HxWxD) in mm: 76 x 25.4 x 74 2tier width (HxWxD) in mm: 76 x 50.8 x 74

Installation dimensions

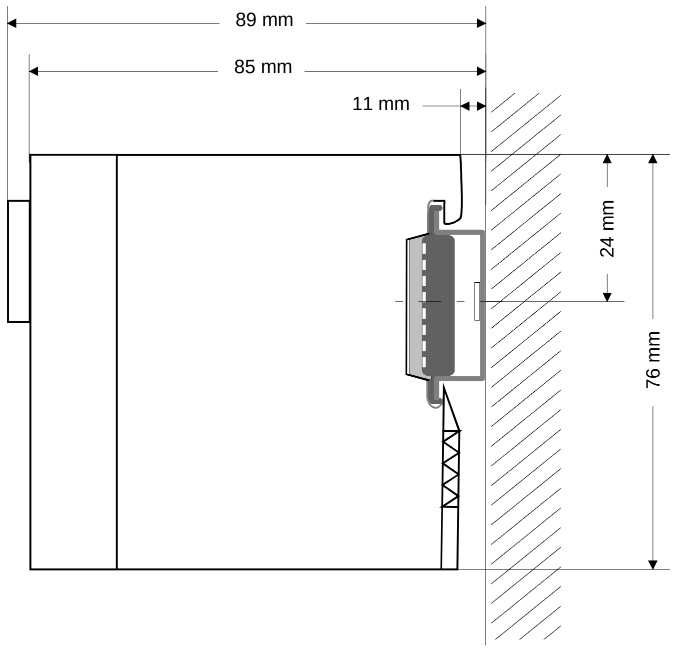


Installed and wired dimensions

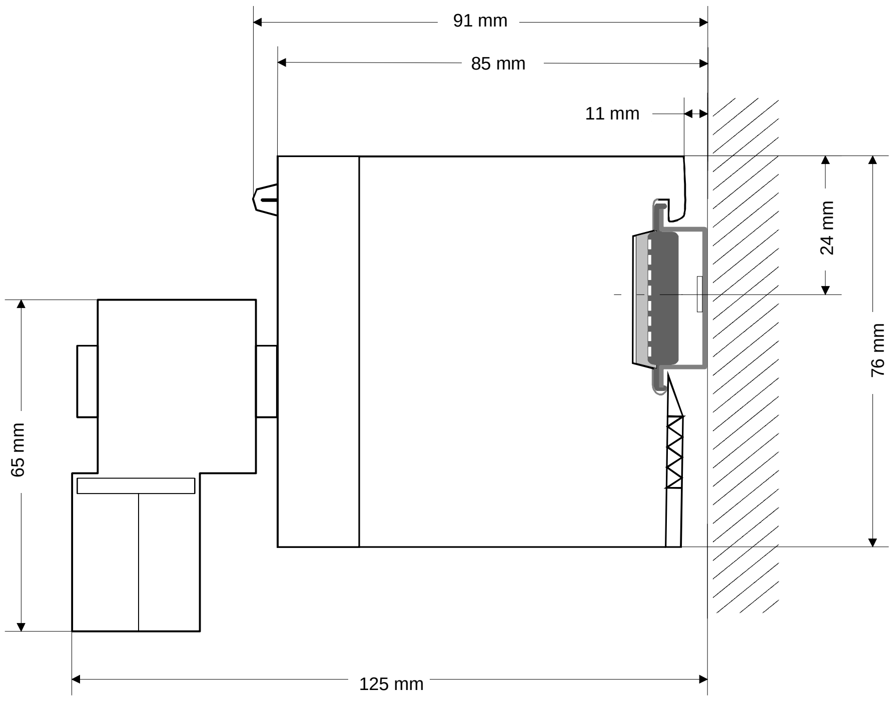
In- / Output modules



Function modules/
Extension modules



CPUs (here with
EasyConn from
VIPA)



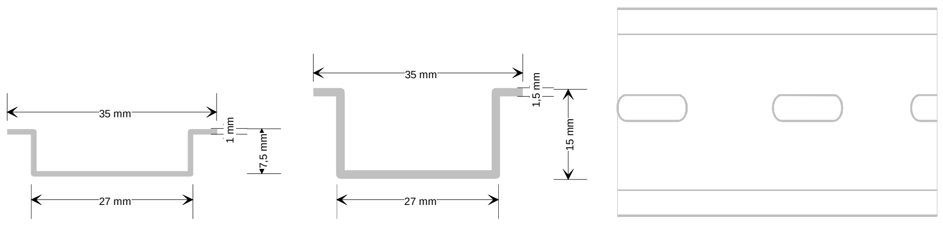
Installation

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:

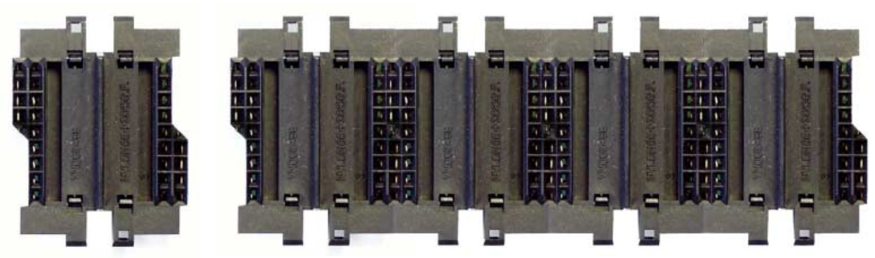


Order number	Label	Description
290-1AF00	35mm profile rail	Length 2000mm, height 15mm
290-1AF30	35mm profile rail	Length 530mm, height 15mm

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:



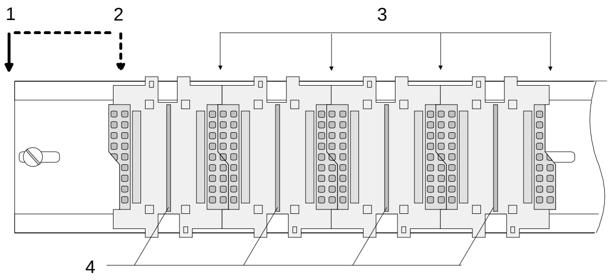
The bus connector is to be placed on the profile rail until it clips in its place and the bus connections look out from the profile rail.

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

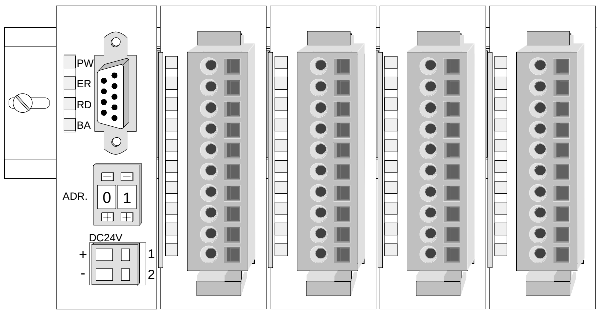
Installation on a profile rail

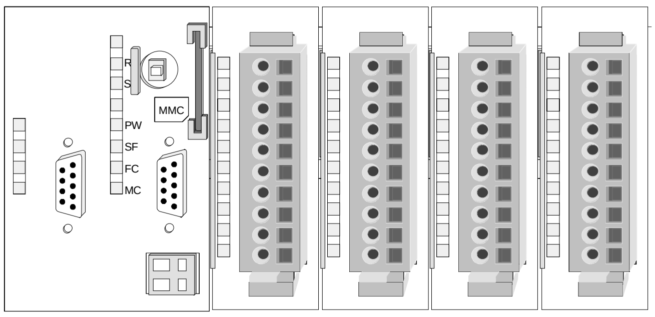
The following figure shows the installation of a 4tier width bus connector in a profile rail and the slots for the modules.

The different slots are defined by guide rails.



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

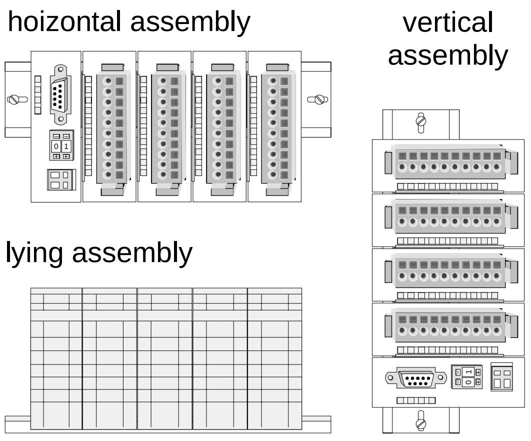




Assembly regarding the current consumption

- 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

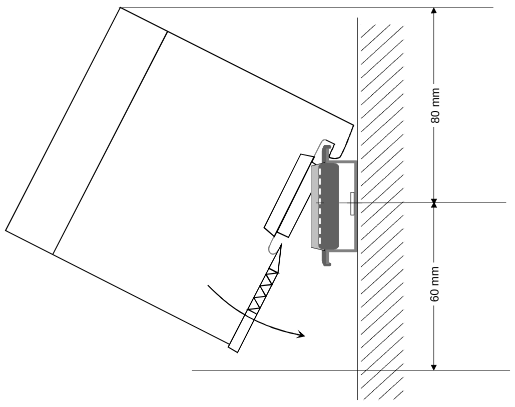


Please regard the allowed environmental temperatures:

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

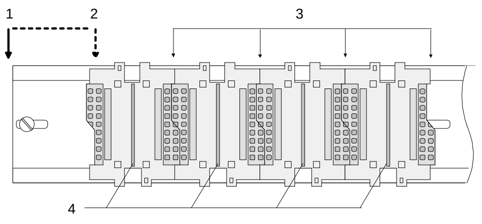
The horizontal assembly always starts at the left side with a head module, then you install the peripheral modules beside to the right.

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.

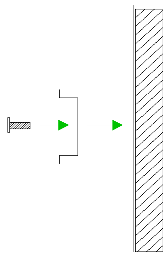
[1]	Head module (double width)
[2]	Head module (single width)
[3]	Peripheral modules
[4]	Guide rails

- 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.

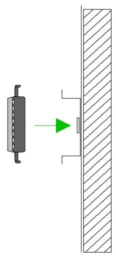
Slots after the last module may remain unoccupied.

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.

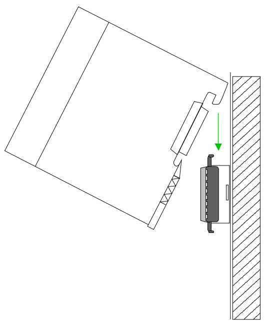
Assembly procedure



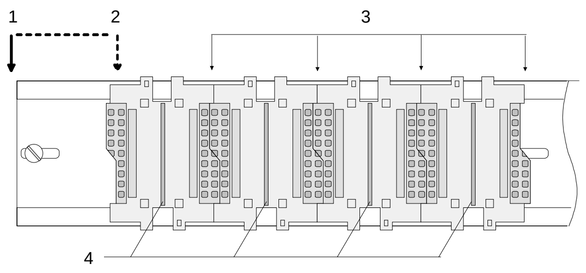
- Install the profile rail. Make sure that a clearance of at least 60mm exists above and 80mm below the middle of the profile rail.



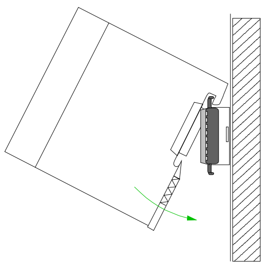
- Press the bus connector into the profile rail until it clips securely into place and the bus-connectors look out from the profile rail. This provides the basis for the installation of your modules.



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

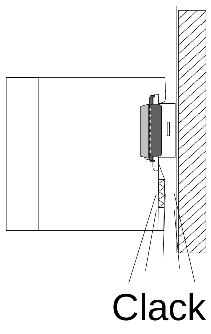


[1]	Head module (double width)
[2]	Head module (single width)
[3]	Peripheral module
[4]	Guide rails



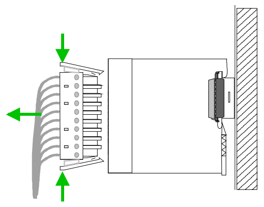
- Insert the module that you are installing into the profile rail at an angle of 45 degrees from the top and rotate the module into place until it clicks into the profile rail with an audible click. The proper connection to the backplane bus can only be guaranteed when the module has properly clicked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

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



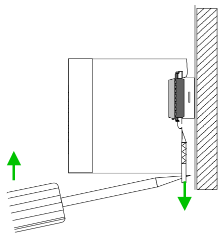
Demounting and module exchange

1



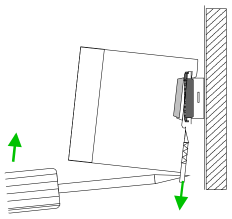
- Remove if exists the wiring to the module, by pressing both locking lever on the connector and pulling the connector.

2



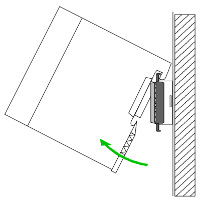
- The casing of the module has a spring loaded clip at the bottom by which the module can be removed.

3



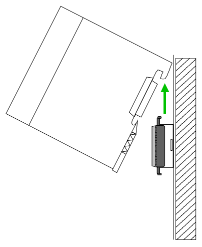
- The clip is unlocked by pressing the screwdriver in an upward direction.

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.

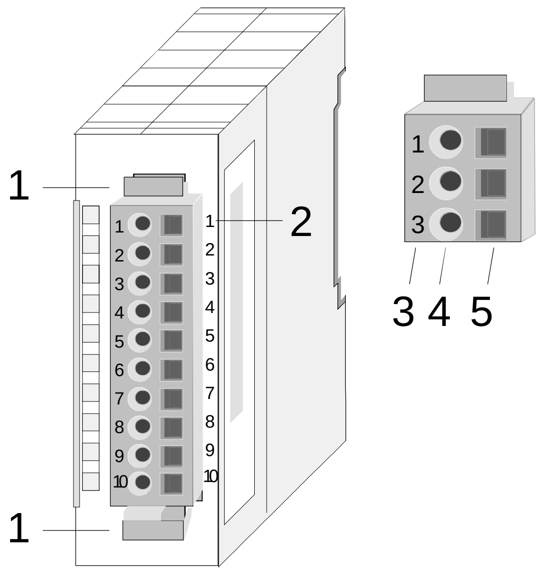
The modules carry spring-clip connectors for interconnections and wiring.

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.



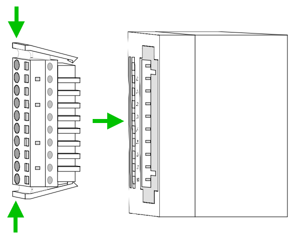
[1]	Locking lever
[2]	Pin no. at the module
[3]	Pin no. at the connector
[4]	Wiring port
[5]	Opening for screwdriver

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!

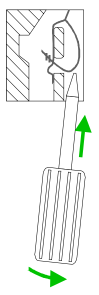
Wiring procedure



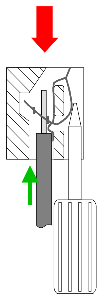
- Install the connector on the module until it locks with an audible click. For this purpose you press the two clips together as shown.

  The connector is now in a permanent position and can easily be wired.

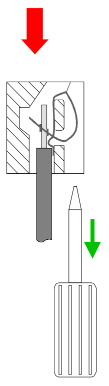
The following section shows the wiring procedure from top view.



- 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 M-Bus interface.

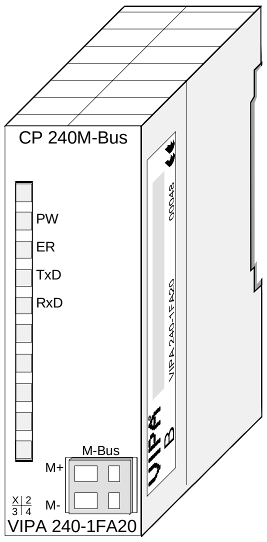
Contents

Topic **Page**

Properties

CP 240 M-Bus
240-1FA20

- Voltage supply via back plane bus
- Up to 6 slaves may be connected
- Standardized bus system acc. DIN EN 1434-3

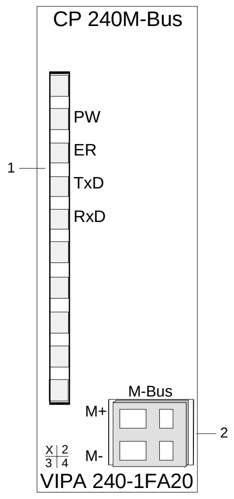


Order data

Type	Order No.	Description
CP 240 M-Bus	VPA 240-1FA20	CP with M-Bus interface

Structure

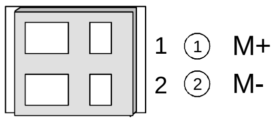
CP 240 M-Bus
240-1FA20



[1] LED Status monitor
[2] M-Bus interface

Interface

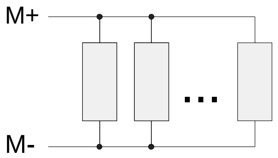
M-Bus



M-Bus interface

The labels M+ and M- serve the distinction of the M-Bus wires. The polarization is irrelevant for M-Bus installations.

Since the CP gets its voltage supply via the back plane bus and thus supplies the connected M-Bus modules, up to 6 slaves may be connected. The slaves must be connected in parallel.



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	**240-1FA20**
Type	CP 240, M-Bus
Current consumption/power loss	
Current consumption from backplane bus	300 mA
Power loss	1.5 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	-
Transmission speed, min.	300 bit/s
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	M-Bus master
Datasizes	
Input bytes	16
Output bytes	16
Parameter bytes	16
Diagnostic bytes	0
Housing	
Material	PPE / PA 6.6
Mounting	Profile rail 35 mm
Mechanical data	
Dimensions (WxHxD)	25.4 x 76 x 78 mm

Order number	**240-1FA20**
Weight	80 g
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL508 certification	yes

Chapter 3 Deployment

Overview

Here you will find the deployment of the communication processor CP 240 M-Bus.

Contents

Topic **Page**

Basics

M-Bus

The M-Bus system (**M**etering **Bus**) is an European standardized (DIN EN 1434-3) two-wire field bus for the excise data capturing. At this the data transfer happens serial via a polarity inversion secure two-wire core from slave systems (measuring devices) to a master system.

The M-Bus has been developed in Germany by Prof. Dr. Horst Ziegler (Uni Paderborn) together with the companies Techem and Texas Instruments.

The main advantage of the M-Bus technique is its high flexibility. Due to the standardization you may easily combine devices of several manufacturers at one bus. Also the inclusion of emitter counter is possible via special M-Bus adapters. Up to 250 counters may be connected at one bus.

Properties

- Polarity inversion secure standardized bus system acc. DIN EN 1434-3
- Short-circuit resistant M-Bus interface
- Current, gas, water and heat counter can be integrated
- Data is read electronically
- Connection of up to 250 counter at one bus
- Counter are individually addressable via unique addresses
- Remote reading possible in dense read intervals
- Extraction of statistic data as base for net optimization
- No special requirements for bus cables or cabling topologies
- Striking distance up to several km

Transfer principle

The data transfer from master to slaves happens via the modulation of the voltage supply. Here at 36V account for the state "1" and 24V for state "0".

The slave system responds the master via the modulation (increase) of its current consumption. Herat 1.5mA account for the state "1" and 11-20mA for state "0".

Due to the voltage modulation and thus the present M-Bus voltage of 24V the terminal equipment can be supplied with the necessary operating voltage.

Fast introduction

Overview

The communication processor CP 240 M-Bus allows the process coupling to different destination or source systems based upon the M-Bus communication.

The CP 240 M-Bus is power supplied via the back plane bus. For the internal communication the VIPA FCs are used. For the project engineering of the CP 240 M-Bus 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 M-Bus to communicate with the CPU, the system always requires a hardware configuration.

A general description of the project engineering of the CP 240 is to be found in chapter "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 M-Bus.
- Parameterize your CP 240 M-Bus.

Parameters

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

Byte	Function	Values	Default parameter
0	reserved		
1	Protocol	F0h: M-Bus	-
2 ... 15	reserved		

You have only to set F0h in Byte 1 as protocol for M-Bus. 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 M-Bus. For this, send and receive data have each a reserved 2048Byte buffer. Together with a CPU 21x the following handling blocks are used:

Name	FCs	Short description
SEND	FC0	Send block
RECEIVE	FC1	Receive block
SYNCHRON_RESET	FC9	Reset and synchronize the CP 240

Telegram structure

For M-Bus telegrams that are send from the CPU to the CP 240, you must prefix every telegram with one byte, which contains the baud rate. This procedure allows you to communicate with different bus participants during runtime by using different baud rates.

As a countermove the CP 240 announces back the state of the M-Bus data transfer via this byte (0: OK - valid answer, ≠0:error).

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

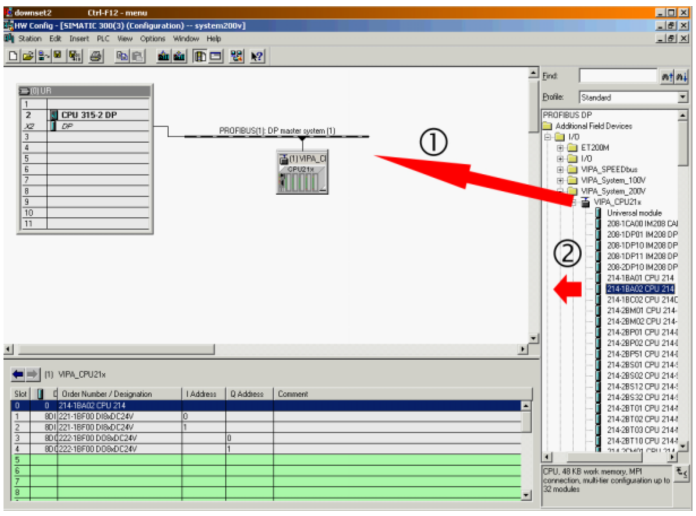
For the project engineering of the CPU a thorough knowledge of the SIMATIC Manager and the hardware configurator from Siemens is required!

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.



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

(1) VIPA_CPU21x

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					

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 via M-Bus.

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

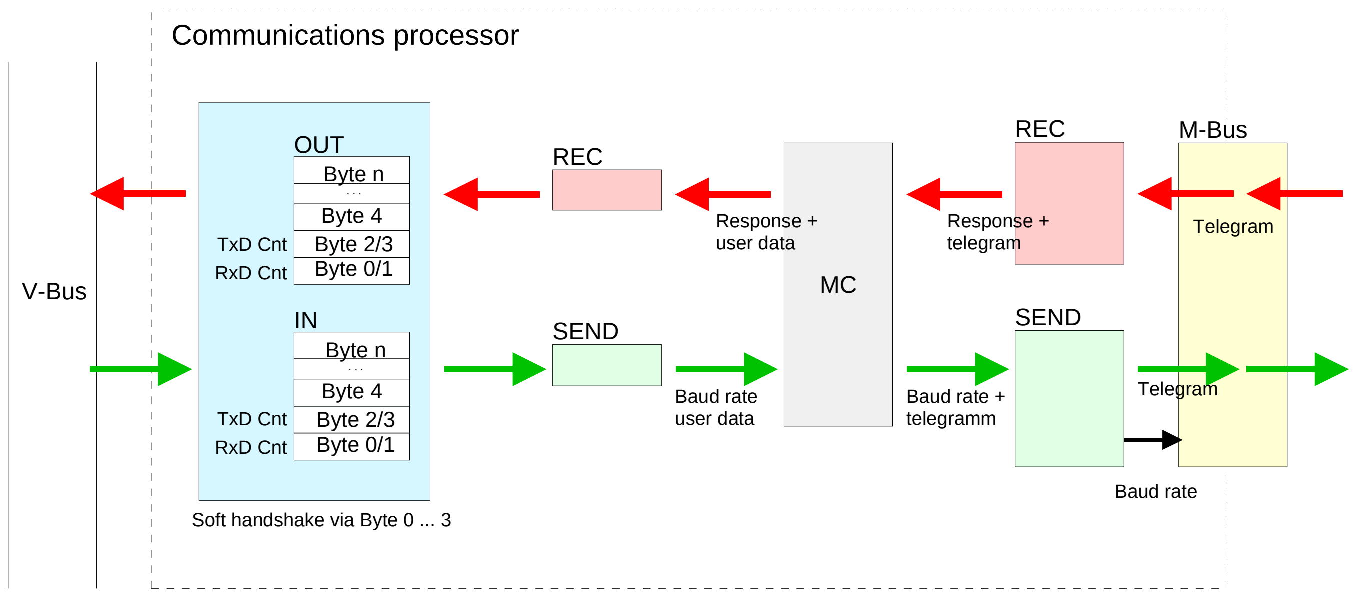
Communication via back plane bus

The exchange of received telegrams via the back plane bus happens asynchronously. When a complete telegram has been arrived via M-Bus, it is stored in the buffer. The user data are extracted from the M-Bus telegram and transferred to the CPU via back plane bus.

Tasks of the CPU

A telegram that is to send has to be transferred to the CP 240. This recognizes the telegram type due to the length definition, supplements it with the according telegram bytes and handles the telegram on to the send buffer. The CP 240 compiles these blocks in the send buffer and sends it via M-Bus with the specified baud rate 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 you a series of standard handling blocks that provide the software handshake comfortable and easy.

At deployment of the CP 240 without handling blocks, the functionality is elucidated with an example of data send and receive.

Example SEND data

For example, a telegram with 30Byte length is to send. Please regard that the CP 240 takes the 1st byte of the telegram as baud rate. The CPU writes the first 12Byte user data of the telegram into the Bytes 4 to 15. Byte 2/3 contain the telegram length, i.e. "30". The CP 240 receives the data via the back plane bus and copies the 12Byte user data into the send buffer. For the acknowledgement of the telegram the CP 240 writes the value "30" back to Byte 2/3 (length of the telegram).

At reception of the "30", the CPU can send further 12Byte user data to Byte 4 to 15 and the rest length of the telegram ("18" Byte) to Byte 2/3 to the CP 240. Again, this stores the user data in the send buffer and sends back the length information ("18") in Byte 2/3 to the CPU.

The CPU receives the "18" and sends the remaining 6Byte user data in the Bytes 4 to 9 and the according rest length ("6") in Byte 2/3 to the CP 240. The user data is stored in the send buffer and the value "6" is send back to the CPU via Byte 2/3.

The CPU receives the "6" and sends back a "0" via Byte 2/3. The CP 240 now initializes the sending of the telegram via the M-Bus interface. After data transfer is completed, the CP 240 sends back a "0" to the CPU via Byte 2/3.

At reception of the "0", the CPU is able to send a new telegram to the CP 240.

Example RECEIVE data

For example, the CP 240 received a telegram with 18Byte user data via M-Bus. Out of this telegram the first 11Byte user data and the prefixed respond byte are taken over into the Bytes 4 to 15 of the receive buffer and the length of the telegram (i.e. "18") into Byte 0/1. The data is transferred to the CPU via the back plane bus. The CPU stores the 12Byte user data and sends back the length value "18" to the CP 240.

At reception of the "18", the CP 240 writes the remaining 7Byte user data into Byte 4 to 10 of the receive buffer and the length ("7") of the transferred user data into Byte 0/1. The CPU stores the user data and announces back to the CP 240 the value "7" into Byte 0/1.

At reception of the "7", the CP 240 sends back the value "0" into Byte 0/1, which means telegram complete, to the CPU. The CPU responds a "0" into Byte 0/1 to the CP 240.

Receiving "0" the CP 240 may send another telegram to the CPU.

Overview of M-Bus telegrams

Overview

M-Bus differentiates the following 4 telegrams:

- Single character
  The single character serves the acknowledgment of correctly received telegrams (syntax and check sum)
- Short frame
  For a short frame telegram you always have to define 3 bytes. The M-Bus takes them as telegrams of the CP 240 to a slave, e.g.:
  - SND_NKE: initialize counter
  - REQ_UD2: request counter data
- Control frame
  The control frame requires 4 defined bytes. Herewith you may send M-Bus control commands like e.g.:
  - set baud rate of the slave
  - execute slave reset
- Long frame
  The long frame contains the user data and has consequently a variable length. Here the user data may be sent in both directions, e.g.:
  - send user data to the slave
  - select slave via secondary address
  - set date and time of a slave
  - select data area to read from

Please regard that every M-Bus telegram has to be prefixed by one byte that specifies the baud rate to use. At error-free reception, the Byte 0 of the receive DB contains 00h. A value <> 00h indicates an error.

At the transfer of a M-Bus telegram to the CP 240, the telegram type is automatically detected; the data is automatically included into the according telegram structure and put out via M-Bus. With the call of the VIPA handling blocks exclusively the following data (marked green) must be transferred, depending on the telegram. Here the sum of these bytes has to be set as length value.

Single charact.	**Short frame**	**Control frame**	**Long frame**
Baud rate	Baud rate	Baud rate	Baud rate
E5h	10h Start	68h Start	68h Start
	C field	L field = 3	L field
	A field	L field = 3	L field
	Check sum	68h Start	68h Start
	Stop 16h	C field	C field
		A field	A field
		CI field	CI field
		Check sum	User Data (0...252Byte)
		Stop 16h	
			Check sum
			Stop 16h
Length for handling block: 2	Length for handling block: 3	Length for handling block: 4	Length for handling block: 5...n

Baud rate

Every M-Bus telegram has to be prefixed with one byte that specifies the baud rate. The following baud rates are supported:

Hex value	Baud
B8h	300
BBh	2400
BDh	9600

If none of the mentioned values is entered in the 1st byte, 2400Baud are automatically used.

C field

Via the C field the function of a telegram is defined. It enables also to differentiate between the call and response direction on connection level.

Depending on the direction, the C field has the following structure:

Send	0	1	FCB	FCV	F3	F2	F1	F0
Receive	0	0	ACD	DFC	F3	F2	F1	F0

Functions

With M-Bus, the following functions are defined:

Name	**C field binary**	**C field hex**	**Telegram**	**Description**
SND_NKE	0100 0000	40	short frame	This causes an initialization of the slaves (terminal equipment) and correlates a deletion of the FCB bits and an acknowledgment by the single character E5h.
SND_UD	01*F*1 0011	53/73	long / control frame	With that user data may be send to slaves.
REQ_UD2	01*F*1 1011	5B/7B	short frame	This function summons a slave to answer with data class 2 (e.g. counter values). If the slave has no such data it responds with a single character. Otherwise it sends a RSP_UD. At a defective transfer an answer lacks.
REQ_UD1	01*F*1 1010	5A/7A	short frame	This allows you to summon a slave to answer with data class 1 (e.g. alarm protocols). If the slave has no such data it responds with a single character. Otherwise it sends a RSP_UD. At a defective transfer an answer lacks.
RSP_UD	00*AD* 1000	08/18/28/38	long / control frame	Data transfer after request (slave respond)

F: FCB bit, A: ACD bit, D: DFC bit

FCB bit

The FCB bit alternates at successful communication. An unchanging FCB summons the terminal device to send the last telegram again. The lack of an answer from a slave is accounted after 330 bit times plus 50ms. The master firstly assumes that an error occurred in the connection level. It repeats the transfer of the same telegram for up to two times. If there is still no answer from the slave, the bus receives a pause of 33 bit times.

The same procedure starts when the master receives a defective respond from the slave.

Baud rate	33x	330x
300	110	1100
2400	13.8	137.5
9600	3.4	34.4

Bit time in ms

A field

For the addressing of the slaves the values 1 to 250 are available. Not configured (new) slaves have the primary address 0.

The addresses 254 and 255 are to be used as broadcast address. Using 255, the master sends information to all participants, but does not receive an answer. Using 254, every slave answers with its address. If you are having more than one slave this causes collisions. Therefore the address 254 should exclusively be used for test purposes.

The address 253 shows a secondary addressing. The addresses 251 and 252 are reserved for future extensions.

CI field

The CI field defines the purpose of the transmitted telegram. The data fields are always sent with the lowest value byte first (LSB first).

Slave → Master

CI field	Application	Defined at
70h	Sending of an error state	User group March ´94
71h	Sending of an alarm state	User group March ´94
72h	Respond with variable data structure	EN1434-3
73h	Respond with fix data structure	EN1434-3

Master → Slave

CI field	Application	Defined at
51h	Sending data	EN1434-3
52h	Select slave	User group July ´93
50h	Reset on application level	User group March ´94
54h	Synchronize slave	-
B8h	Set baud rate 300	User group July ´93
B9h	Set baud rate 600	User group July ´93
BAh	Set baud rate 1200	User group July ´93
BBh	Set baud rate 2400	User group July ´93
BCh	Set baud rate 4800	User group July ´93
BDh	Set baud rate 9600	User group July ´93
BEh	Set baud rate 19200	-
BFh	Set baud rate 38400	-
B1h	Output RAM content	Techem suggestion
B2h	Write RAM content	Techem suggestion
B3h	Start calibration test mode	User group July ´93
B4h	Read EEPROM	Techem suggestion
B6h	Start software test	Techem suggestion
90h ... 97h	Reserved	

Check sum

Check sum serves the recognition of transfer and synchronization errors. Whereat the check sum is evaluated over the following data bytes: C field, A field, CI field (if present) and user data (if present).

Examples

The following examples show how a telegram is build-up from predefined data and how the received telegram is stored in the data block.

Send data

Assignment via DB

BBh	Baud rate
7Bh	C field
FEh	A field

→

Telegram via M-Bus

10h	Start
7Bh	C field: REQ_UD2
FEh	A field: PtP Broadcast
79h	Check sum of C and A field
16h	Stop

ANZ for handling block FC0: 3

Baud rate

Please take care to prefix one byte to every M-Bus telegram that specifies the baud rate to use.

Receive data

Telegram via M-Bus

68h	68h Start
00h	L field
03	L field
68h	68h Start
08h	C field: RSP_UD
02h	A field: address 02h
72h	CI field: resp. variable
01h	Data field
02h	
03h	
75h	Check sum of C,A,CI, data
16h	Stop

→

storage in DB:

00h	Response
08h	C field: RSP_UD
02h	A field: address 02h
72h	CI field: response variable length
01h	Data field
02h	
03h	

ANZ for handling block FC1: 4

Response byte

At error-free reception, Byte 0 of the receive-DB contains the value 00h. A value <> 00h indicates an error.

Here the bits have the following allocation:

Bit 0: set if no answer was received

Bit 1: set in the case of short-circuit at the bus

Bit 2... 7: reserved

Example for M-Bus deployment

Overview

In the following example a M-Bus communication (send and receive) is build-up. Furthermore the example shows how you may easily control the communication processes by using the handling blocks.

At need you may receive the example project from VIPA.

Requirements

For the example, the following components are required:

1 System 200V consisting of CPU 21x and CP 240 M-Bus

1 acquisition device with M-Bus interface

Project engineering tool SIMATIC Manager from Siemens with transfer cable

Approach

Build-up the System 200V.

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

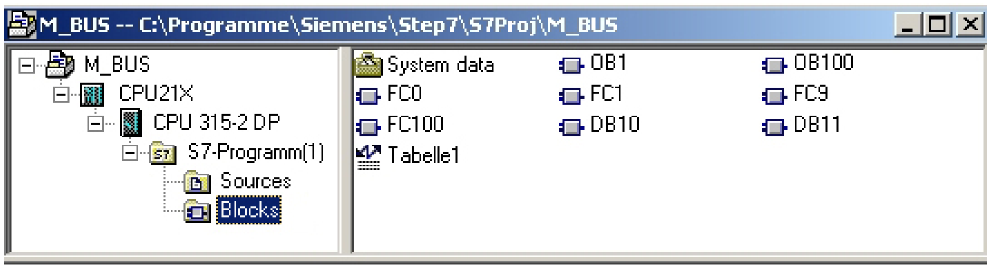
Dearchive the project

Follow these steps in the Siemens SIMATIC Manager:

- Start the Siemens SIMATIC Manager.
- To extract the file MBUS.zip select **File** > *de-archive*.
- Choose the example file MBUS.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.	**Name**	**Type**	**Comment**
0.0		STRUCT	
+0.0	Sendefach	STRUCT	Send data block
+0.0	Baudrate	BYTE	B8h=300, BBh=2400, BDh=9600
+1.0	OK / C-Feld	BYTE	E5h=OK / C field
+2.0	A-Feld	BYTE	A field
+3.0	CI-Feld	BYTE	CI field
+4.0	User-Data Byte 0	BYTE	
...			
+252.0	User-Data Byte 247	BYTE	Transfer complete
+253.0	Reserve	BYTE	
+254.0	Anzahl	WORD	Transmitting length
+256.0	gesendet	WORD	Already sent data
+258.0	Byte_Zaehler	WORD	Transmitting length (internal)
+260.0	Kom_Ende	BOOL	Telegram transmitted completely
+260.1	LB	BOOL	Last block has been sent
+260.2	SL	BOOL	Still transmitting
+260.3	Fehl	BOOL	Error during transmission
+260.4	Senden_Start	BOOL	Start bit
+261.0	PAFE	BYTE	Parameterization error
=262.0		END_STRUCT	

DB11
RECEIVE data block

Addr.	**Name**	**Type**	**Comment**
0.0		STRUCT	
+0.0	Data	ARRAY [0..100]	
*1.0		Byte	
+102.0	Anzahl	WORD	Amount of received data
+104.0	empfangen	WORD	Already received data
+106.0	Byte_Zaehler	WORD	Amount of received bytes (internal)
+108.0	Empf_laeuft	BOOL	Still receiving
+108.1	LB	BOOL	Last block has been received
+108.2	Fehl	BOOL	Error during reception
+108.3	Reserve	BOOL	
+108.4	Empfang fertig	BOOL	Reception ready
+109.0	PAFE	BYTE	Parameterization error
=110.0		END_STRUCT	

PLC program

The PLC application has the following structure:

OB1

```
          CALL       FC    9              //SYNCHRON_RESET
           ADR       :=256                //Module address
           TIMER_NR  :=T8                 //Delay time for CP
           ANL       :=M3.0               //CPU start complete
           NULL      :=M3.1               //Internal flag
           RESET     :=M3.2               //Initialize reset at CP
           STEUERB_S:=DB10.DBB260         //Control bits for Send
           STEUERB_R:=DB11.DBB108         //Control bits for Receive

          U          M     3.0            //as long as Synchron active
          BEB                             //no processing of CP

          CALL       FC    100            //M-Bus communication
           Adr_CP    :=256                //Module address
           Baud      :=MB100              //Transfer baud rate
           C_Field   :=MB101              //Transfer value C field
           A_Field   :=MB102              //Transfer value A field
           CI_Field  :=MB103              //Transfer value CI field
           Data      :=MB104              //Transfer telegram length
           RET_VAL   :=MW106              //Return value
           Senden_Start:=M99.0            //Start order

          U          M     99.0           //Order running
          BEB

          L          MW    106            //Return of send function
          L          W#16#2000            //Ready without error
          ==I
          SPB        copy

          NOP        0                    //Error evaluation
          BEA

   copy:  L          0                    //Delete ready without error
          T          MW    106
          L          MB    102            //Participant address
          L          20                   //Basic No. for the data-DBs
          +I
          T          MW    50             //Data block no. for data storage
          L          0                    //1st byte to copy
          T          MW    188            //Predefine byte counter

   loo:   L          MW    188            //Load byte counter
          SLW        3                    //x8 is byte address in DB
          T          MD    184            //Save address
          AUF        DB    11             //Open receive buffer
          L          DBB   [MD 184]       //Value from receive buffer
          AUF        DB    [MW 50]
          T          DBB   [MD 184]       //Store in data-DB

          L          MW    188
          +1                              //Increase byte number
          T          MW    188
          L          DB11.DBW  102        //Last byte to copy
          <=I                             //not all bytes copied yet
          SPB        loo                  //then resume
```

OB 100

```
          UN         M     3.0
          S          M     3.0            //Set start-up ID
```

FC 100

This function sends a request to a M-Bus participant and receives the answer. The transmit data has to be entered into DB 10 starting with data byte 4 before calling the function.

```
        UN        #Senden_Start
        BEB
        U         DB11.DBX    108.7          //Waiting for acknowledgment
        SPB   REC
        NOP       0                          //Enter transmit data into send buffer
        L         #Baud                      //1st send byte is baud rate
        T         DB10.DBB    0
        L         #C_Field                   //2nd send byte is C_Field
        T         DB10.DBB    1
        L         #A_Field                   //3rd send byte is A_Field
        T         DB10.DBB    2
        L         #CI_Field                  //4th send byte is CI_Field
        T         DB10.DBB    3              //At Long Frame data must be entered
        NOP   0                              //from user data on before FC call
        SET                                  //
        S         DB10.DBX    260.4          //Set send release
        L         0
        L         #Data                      //Telegram length at Long Frame
        <>I
        +4
        T         DB10.DBW    254            //Telegram length for Long Frame
        SPB       send
        L         0
        L         #CI_Field                  //ID for Control Frame
        <>I
        L         4                          //Telegram length for Control Frame
        SPB       sen1
        L         3                          //Telegram length for Short Frame
sen1:   T         DB10.DBW    254            //Telegram length
send:   CALL      FC    0                    //Block send
         ADR      :=#Adr_CP                  //Module address
         _DB      :=DB10               //DB send buffer
         ABD      :=W#16#0                   //1st data byte to send
         ANZ      :=DB10.DBW254              //Amount of send data
         PAFE     :=DB10.DBB261              //Error byte
         FRG      :=DB10.DBX260.4            //Send release
         GESE     :=DB10.DBW256              //Internal variable
         ANZ_INT  :=DB10.DBW258              //Internal variable
         ENDE_KOM :=DB10.DBX260.0            //Internal variable
         LETZTER_BLOCK:=DB10.DBX260.1        //Internal variable
         SENDEN_LAEUFT:=DB10.DBX260.2        //Internal variable
         FEHLER_KOM   :=DB10.DBX260.3        //Internal variable
        U         DB10.DBX    260.4          //Transmission still running
        BEB                                  //then end
        S         DB11.DBX    108.7          //Waiting for acknowledgment
REC:    NOP       0
        CALL      FC    1
         ADR      :=#Adr_CP                  //Module address
         _DB      :=DB11               //DB receive buffer
         ABD      :=W#16#0                   //1st data byte receive buffer
         ANZ      :=DB11.DBW102              //Amount of received bytes
         EMFR     :=DB11.DBX108.4            //Telegram received completely
         PAFE     :=DB11.DBB109              //Error byte
         GEEM     :=DB11.DBW104              //Internal variable
         ANZ_INT  :=DB11.DBW106              //Internal variable
         EMPF_LAEUFT     :=DB11.DBX108.0 //Internal variable
         LETZTER_BLOCK   :=DB11.DBX108.1 //Internal variable
         FEHL_EMPF       :=DB11.DBX108.2 //Internal variable
        UN        DB11.DBX    108.4          //No new value received yet
        BEB
        R         DB11.DBX  108.4
        R         DB11.DBX  108.7
        R         #Senden_Start
        L         DB11.DBW  102
        L         1                          //If received only 1Byte -> error
        ==I
        SPB       Fehl
        L         W#16#2000                  //After respond reception, delete
Ende:   T         #RET_VAL                   //start bit and return ID to RET_VAL
        BEA
Fehl:   L         DB11.DBB    0              //Received byte
        L         1                          //No response from M-Bus slave
        ==I
        L         W#16#8001                  //Error ID for no response
        SPB       Ende
        L         W#16#80FF                  //Undefined response from CP
        SPA       Ende
```