

# VIPA System 200V

**CP | Handbuch**

HB97D\_CP | RD\_240-1EA20 | Rev. 12/42

Oktober 2012



## **Copyright © VIPA GmbH. All Rights Reserved.**

Dieses Dokument enthält geschützte Informationen von VIPA und darf außer in Übereinstimmung mit anwendbaren Vereinbarungen weder offengelegt noch benutzt werden.

Dieses Material ist durch Urheberrechtsgesetze geschützt. Ohne schriftliches Einverständnis von VIPA und dem Besitzer dieses Materials darf dieses Material weder reproduziert, verteilt, noch in keiner Form von keiner Einheit (sowohl VIPA-intern als auch -extern) geändert werden, es sei denn in Übereinstimmung mit anwendbaren Vereinbarungen, Verträgen oder Lizenzen.

Zur Genehmigung von Vervielfältigung oder Verteilung wenden Sie sich bitte an:

VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH

Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 (91 32) 744 -0

Fax.: +49 9132 744 1864

E-Mail: [info@vipa.de](mailto:info@vipa.de)

<http://www.vipa.com>

## **Hinweis**

Es wurden alle Anstrengungen unternommen, um sicherzustellen, dass die in diesem Dokument enthaltenen Informationen zum Zeitpunkt der Veröffentlichung vollständig und richtig sind. Das Recht auf Änderungen der Informationen bleibt jedoch vorbehalten.

Die vorliegende Kundendokumentation beschreibt alle heute bekannten Hardware-Einheiten und Funktionen. Es ist möglich, dass Einheiten beschrieben sind, die beim Kunden nicht vorhanden sind. Der genaue Lieferumfang ist im jeweiligen Kaufvertrag beschrieben.

## **EG-Konformitätserklärung**

Hiermit erklärt VIPA GmbH, dass die Produkte und Systeme mit den grundlegenden Anforderungen und den anderen relevanten Vorschriften übereinstimmen.

Die Übereinstimmung ist durch CE-Zeichen gekennzeichnet.

## **Informationen zur Konformitätserklärung**

Für weitere Informationen zur CE-Kennzeichnung und Konformitätserklärung wenden Sie sich bitte an Ihre Landesvertretung der VIPA GmbH.

## **Warenzeichen**

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S und Commander Compact sind eingetragene Warenzeichen der VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 ist ein eingetragenes Warenzeichen der profichip GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300 und S7-400 sind eingetragene Warenzeichen der Siemens AG.

Microsoft und Windows sind eingetragene Warenzeichen von Microsoft Inc., USA.

Portable Document Format (PDF) und Postscript sind eingetragene Warenzeichen von Adobe Systems, Inc.

Alle anderen erwähnten Firmennamen und Logos sowie Marken- oder Produktnamen sind Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer.

## **Dokument-Support**

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Fehler anzeigen oder inhaltliche Fragen zu diesem Dokument stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744 1204

E-Mail: [documentation@vipa.de](mailto:documentation@vipa.de)

## **Technischer Support**

Wenden Sie sich an Ihre Landesvertretung der VIPA GmbH, wenn Sie Probleme mit dem Produkt haben oder Fragen zum Produkt stellen möchten. Ist eine solche Stelle nicht erreichbar, können Sie VIPA über folgenden Kontakt erreichen:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefon: +49 9132 744 1150 (Hotline)

E-Mail: [support@vipa.de](mailto:support@vipa.de)

## Inhaltsverzeichnis

<b>Über dieses Handbuch</b> .....	<b>1</b>
<b>Sicherheitshinweise</b> .....	<b>2</b>
<b>Teil 1 Grundlagen und Montage</b> .....	<b>1-1</b>
Sicherheitshinweis für den Benutzer .....	1-2
Systemvorstellung.....	1-3
Abmessungen .....	1-5
Montage .....	1-7
Demontage und Modultausch.....	1-11
Verdrahtung .....	1-12
Aufbaurichtlinien.....	1-14
Allgemeine Daten.....	1-17
<b>Teil 2 Hardwarebeschreibung</b> .....	<b>2-1</b>
Leistungsmerkmale .....	2-2
Aufbau.....	2-3
Technische Daten .....	2-5
<b>Teil 3 Einsatz</b> .....	<b>3-1</b>
Grundlagen .....	3-2
Schnelleinstieg .....	3-3
GSD und FCs einbinden.....	3-5
Projektierung .....	3-6
Standardhantierungsbausteine.....	3-9
Kommunikationsprinzip .....	3-12
Beispiel zum Einsatz unter EnOcean .....	3-14
Übersicht der EnOcean-Telegramme .....	3-19
Modul ersetzen und IDBase übernehmen .....	3-34



## Über dieses Handbuch

Das Handbuch beschreibt den bei VIPA erhältlichen System 200V CP 240-1EA20. Hier finden Sie Informationen über den Aufbau und die Einbindung des Kommunikationsprozessors CP 240 mit EnOcean Transceiver-Modul.

### Überblick

#### **Teil 1: Grundlagen und Montage**

Kernthema dieses Kapitels ist die Vorstellung des System 200V von VIPA. Hier finden Sie alle Informationen, die für den Aufbau und die Verdrahtung einer Steuerung aus den Komponenten des System 200V erforderlich sind. Neben den Abmessungen sind hier auch die allgemeinen technischen Daten des System 200V aufgeführt.

#### **Teil 2: Hardwarebeschreibung**

In diesem Kapitel finden Sie Informationen über den Aufbau und die Anschlussbelegung des Kommunikationsprozessors CP 240 mit EnOcean Transceiver-Modul.

#### **Teil 3: Einsatz**

Hier finden Sie Informationen über den Einsatz des Kommunikationsprozessors CP 240 EnOcean.

**Zielsetzung und Inhalt**

Das Handbuch beschreibt den CP 240-1EA20 aus dem System 200V von VIPA. Beschrieben wird Aufbau, Projektierung und Anwendung.

Dieses Handbuch ist Bestandteil des Dokumentationspakets mit der Best.-Nr.: HB97D\_CP und gültig für:

Produkt	Best.-Nr.	ab Stand: HW
CP 240 EnOcean	VIPA CP 240-1EA20	01

**Zielgruppe**

Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik.

**Aufbau des Handbuchs**

Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.

**Orientierung im Dokument**

Als Orientierungshilfe stehen im Handbuch zur Verfügung:

- Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs
- Übersicht der beschriebenen Themen am Anfang jedes Kapitels

**Verfügbarkeit**

Das Handbuch ist verfügbar in:

- gedruckter Form auf Papier
- in elektronischer Form als PDF-Datei (Adobe Acrobat Reader)

**Piktogramme Signalwörter**

Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

**Gefahr!**

Unmittelbar drohende oder mögliche Gefahr. Personenschäden sind möglich.

**Achtung!**

Bei Nichtbefolgen sind Sachschäden möglich.

**Hinweis!**

Zusätzliche Informationen und nützliche Tipps

## Sicherheitshinweise

### Bestimmungsgemäße Verwendung

Der CP 240 ist konstruiert und gefertigt für:

- alle VIPA System-200V-Komponenten
- Kommunikation und Prozesskontrolle
- Allgemeine Steuerungs- und Automatisierungsaufgaben
- den industriellen Einsatz
- den Betrieb innerhalb der in den technischen Daten spezifizierten Umgebungsbedingungen
- den Einbau in einen Schaltschrank



### Gefahr!

Das Gerät ist nicht zugelassen für den Einsatz

- in explosionsgefährdeten Umgebungen (EX-Zone)

### Dokumentation

Handbuch zugänglich machen für alle Mitarbeiter in

- Projektierung
- Installation
- Inbetriebnahme
- Betrieb



### Vor Inbetriebnahme und Betrieb der in diesem Handbuch beschriebenen Komponenten unbedingt beachten:

- Hardware-Änderungen am Automatisierungssystem nur im spannungslosen Zustand vornehmen!
- Anschluss und Hardware-Änderung nur durch ausgebildetes Elektro-Fachpersonal
- Nationale Vorschriften und Richtlinien im jeweiligen Verwenderland beachten und einhalten (Installation, Schutzmaßnahmen, EMV ...)

### Entsorgung

Zur Entsorgung des Geräts nationale Vorschriften beachten!





## Teil 1 Grundlagen und Montage

### Übersicht

Kernthema dieses Kapitels ist die Vorstellung des System 200V von VIPA. Hier finden Sie alle Informationen, die für den Aufbau und die Verdrahtung einer Steuerung aus den Komponenten des System 200V erforderlich sind. Neben den Abmessungen sind hier auch die allgemeinen technischen Daten des System 200V aufgeführt.

### Inhalt

Thema	Seite
<b>Teil 1 Grundlagen und Montage</b> .....	<b>1-1</b>
Sicherheitshinweis für den Benutzer.....	1-2
Systemvorstellung .....	1-3
Abmessungen .....	1-5
Montage .....	1-7
Demontage und Modultausch.....	1-11
Verdrahtung .....	1-12
Aufbau Richtlinien.....	1-14
Allgemeine Daten.....	1-17

## Sicherheitshinweis für den Benutzer

### Handhabung elektrostatisch gefährdeter Baugruppen

VIPA-Baugruppen sind mit hochintegrierten Bauelementen in MOS-Technik bestückt. Diese Bauelemente sind hoch empfindlich gegenüber Überspannungen, die z.B. bei elektrostatischer Entladung entstehen.

Zur Kennzeichnung dieser gefährdeten Baugruppen wird nachfolgendes Symbol verwendet:



Das Symbol befindet sich auf Baugruppen, Baugruppenträgern oder auf Verpackungen und weist so auf elektrostatisch gefährdete Baugruppen hin. Elektrostatisch gefährdete Baugruppen können durch Energien und Spannungen zerstört werden, die weit unterhalb der Wahrnehmungsgrenze des Menschen liegen. Hantiert eine Person, die nicht elektrisch entladen ist, mit elektrostatisch gefährdeten Baugruppen, können Spannungen auftreten und zur Beschädigung von Bauelementen führen und so die Funktionsweise der Baugruppen beeinträchtigen oder die Baugruppe unbrauchbar machen. Auf diese Weise beschädigte Baugruppen werden in den wenigsten Fällen sofort als fehlerhaft erkannt. Der Fehler kann sich erst nach längerem Betrieb einstellen.

Durch statische Entladung beschädigte Bauelemente können bei Temperaturänderungen, Erschütterungen oder Lastwechseln zeitweilige Fehler zeigen.

Nur durch konsequente Anwendung von Schutzeinrichtungen und verantwortungsbewusste Beachtung der Handlungsregeln lassen sich Funktionsstörungen und Ausfälle an elektrostatisch gefährdeten Baugruppen wirksam vermeiden.

### Versenden von Baugruppen

Verwenden Sie für den Versand immer die Originalverpackung.

### Messen und Ändern von elektrostatisch gefährdeten Bau- gruppen

Bei Messungen an elektrostatisch gefährdeten Baugruppen sind folgende Dinge zu beachten:

- Potentialfreie Messgeräte sind kurzzeitig zu entladen.
- Verwendete Messgeräte sind zu erden.

Bei Änderungen an elektrostatisch gefährdeten Baugruppen ist darauf zu achten, dass ein geerdeter LötKolben verwendet wird.



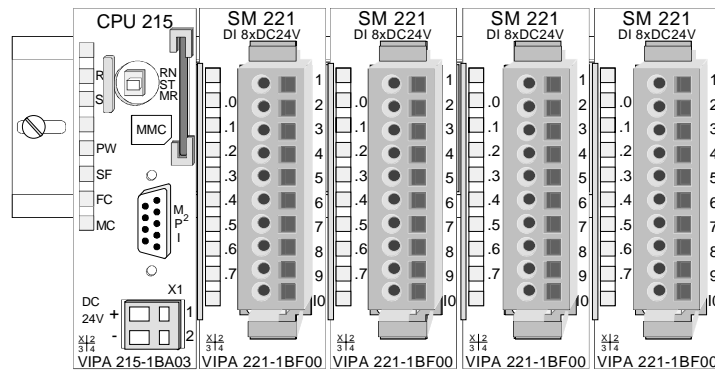
### Achtung!

Bei Arbeiten mit und an elektrostatisch gefährdeten Baugruppen ist auf ausreichende Erdung des Menschen und der Arbeitsmittel zu achten.

# Systemvorstellung

## Übersicht

Das System 200V ist ein modular aufgebautes Automatisierungssystem für die Montage auf einer 35mm Profilschiene. Mittels der Peripherie-Module in 4-, 8- und 16-Kanalausführung können Sie dieses System passgenau an Ihre Automatisierungsaufgaben adaptieren.

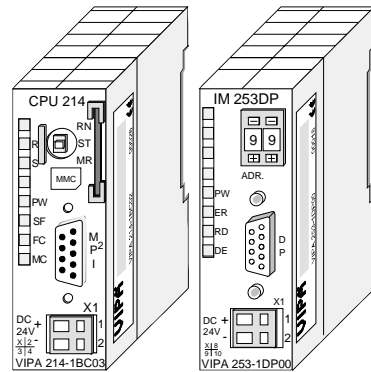


## Komponenten

Das System 200V besteht aus folgenden Komponenten:

- *Kopfmodule* wie CPU und Buskoppler
- *Peripheriemodule* wie I/O-, Funktions- und Kommunikationsmodule
- *Netzteile*
- *Erweiterungsmodule*

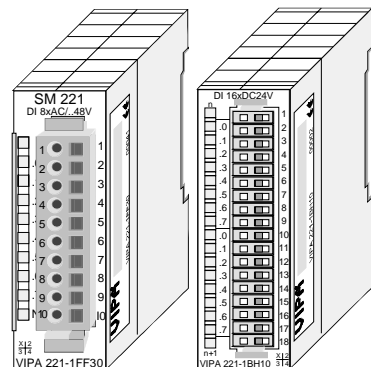
## Kopfmodule



Beim Kopfmodul sind CPU bzw. Bus-Interface und DC 24V Spannungsversorgung in ein Gehäuse integriert.

Über die integrierte Spannungsversorgung werden sowohl CPU bzw. Bus-Interface als auch die Elektronik der angebunden Peripheriemodule versorgt.

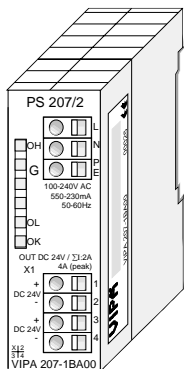
## Peripheriemodule



Die einzelnen Module werden direkt auf eine 35mm-Profilschiene montiert und über Busverbinder, die vorher in die Profilschiene eingelegt werden, an das Kopfmodul gekoppelt.

Die meisten Peripheriemodule besitzen einen 10- bzw. 18poligen Steckverbinder. Über diesen Steckverbinder werden Signal- und Versorgungsleitungen mit den Modulen verbunden.

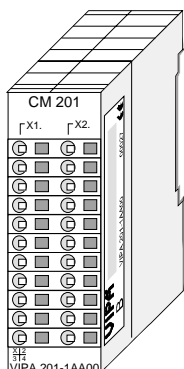
**Netzteile**



Die DC 24V Spannungsversorgung kann im System 200V entweder extern oder über eigens hierfür entwickelte Netzteile erfolgen.

Das Netzteil kann zusammen mit dem System 200V Modulen auf die Profilschiene montiert werden. Es besitzt keine Verbindung zum Rückwandbus.

**Erweiterungs-  
module**



Die Erweiterungsmodule sind unter anderem Ergänzungs-Module für 2- oder 3-Draht Installation.

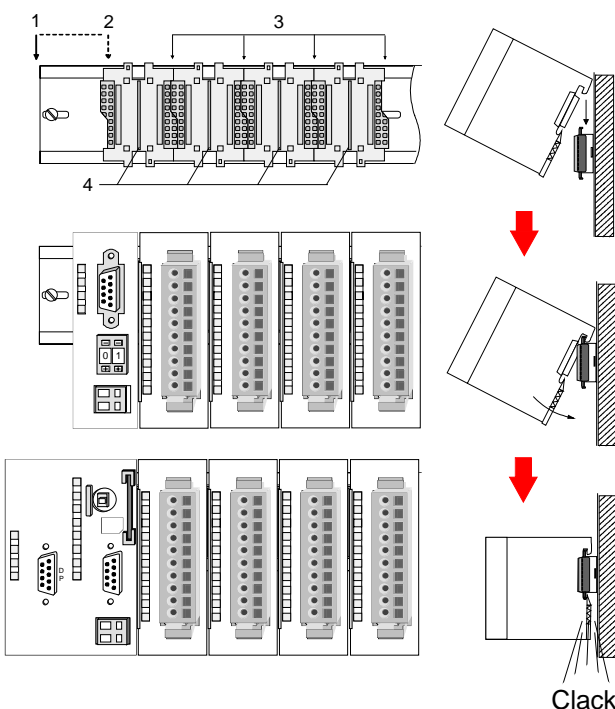
Die Module haben keine Verbindung zum Rückwandbus.

**Aufbau/Maße**

- Profilschiene 35mm
- Maße Grundgehäuse:  
 1fach breit: (HxBxT) in mm: 76x25,4x74 in Zoll: 3x1x3  
 2fach breit: (HxBxT) in mm: 76x50,8x74 in Zoll: 3x2x3

**Montage**

Bitte beachten Sie, dass Sie Kopfmodule nur auf Steckplatz 2 bzw. 1 und 2 (wenn doppelt breit) stecken dürfen.



[1]	Kopfmodul (doppelt breit)
[2]	Kopfmodul (einfach breit)
[3]	Peripheriemodule
[4]	Führungsleisten

**Hinweis**

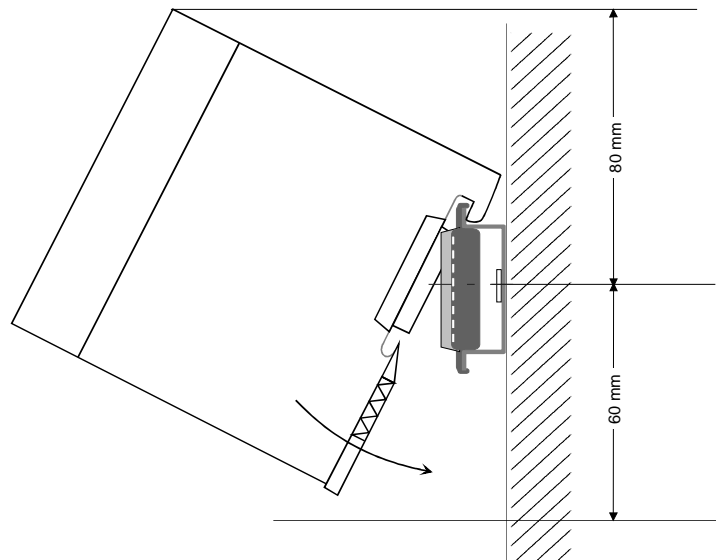
Angaben zur maximalen Anzahl steckbarer Module und zum maximalen Strom am Rückwandbus finden Sie in den "Technischen Daten" des entsprechenden Kopfmoduls.

Bitte montieren Sie Module mit hoher Stromaufnahme direkt neben das Kopfmodul.

## Abmessungen

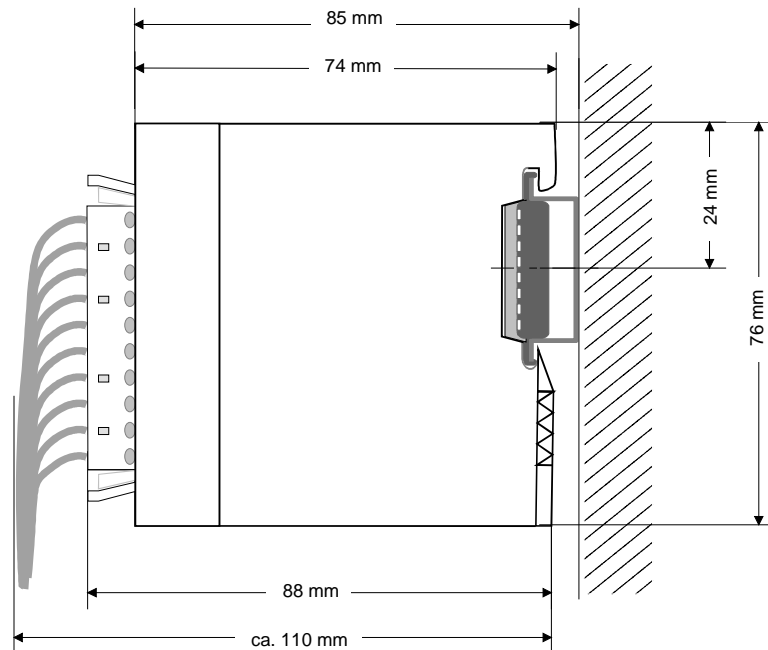
**Maße Grundgehäuse**  
 1fach breit (HxBxT) in mm: 76 x 25,4 x 74  
 2fach breit (HxBxT) in mm: 76 x 50,8 x 74

### Montagemasse

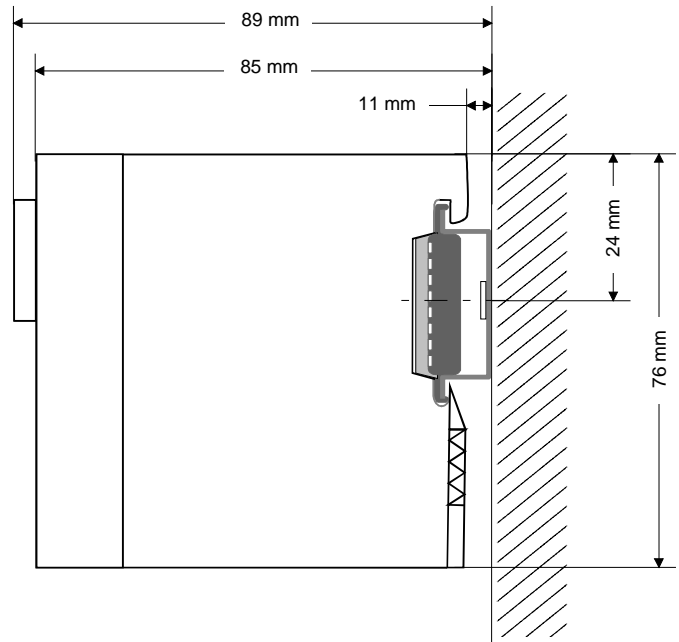


### Maße montiert und verdrahtet

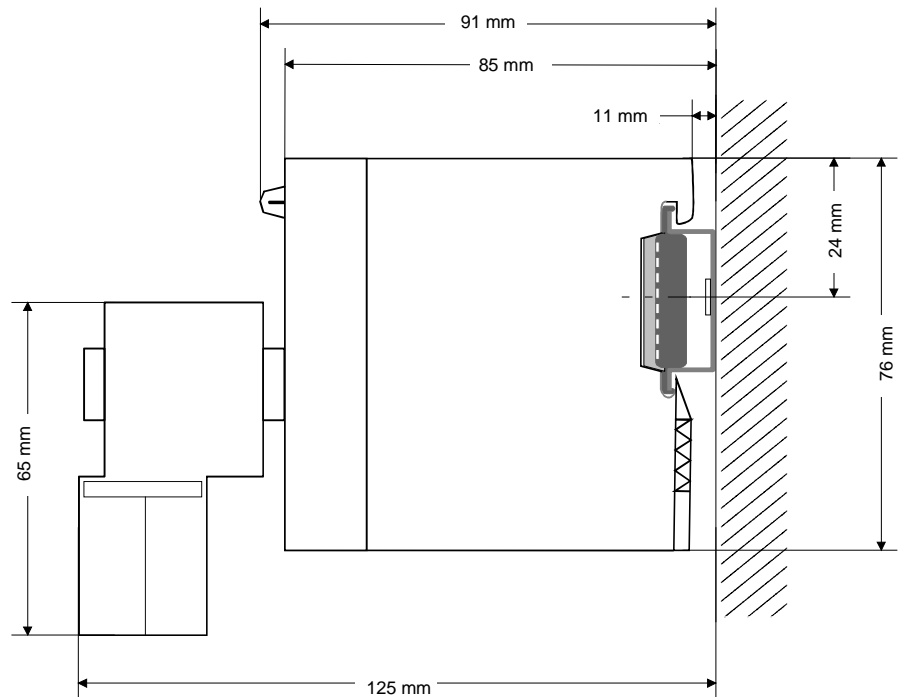
Ein- / Ausgabe-  
 module



Funktionsmodule/  
Erweiterungsmodule



CPUs (hier mit  
VIPA EasyConn)



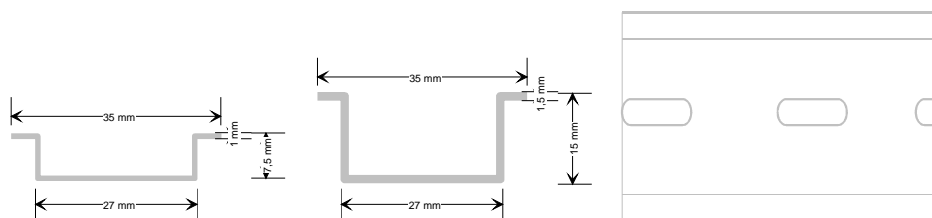
# Montage

## Allgemein

Die einzelnen Module werden direkt auf eine 35mm-Profilschiene montiert und über Rückwandbus-Verbinder verbunden. Vor der Montage ist der Rückwandbus-Verbinder in die Profilschiene einzulegen.

## Profilschiene

Für die Montage können Sie folgende 35mm-Profilschienen verwenden:

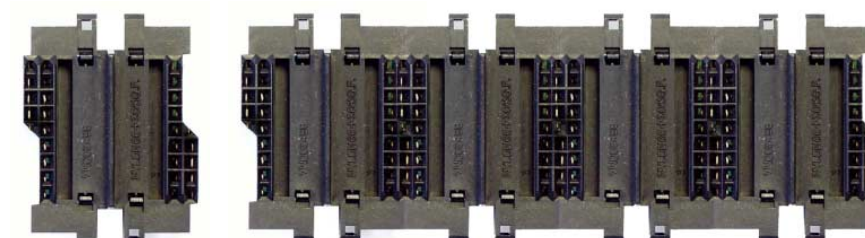


Bestellnummer	Bezeichnung	Beschreibung
290-1AF00	35mm-Profilschiene	Länge 2000mm, Höhe 15mm
290-1AF30	35mm-Profilschiene	Länge 530mm, Höhe 15mm

## Busverbinder

Für die Kommunikation der Module untereinander wird beim System 200V ein Rückwandbus-Verbinder eingesetzt. Die Rückwandbusverbinder sind isoliert und bei VIPA in 1-, 2-, 4- oder 8facher Breite erhältlich.

Nachfolgend sehen Sie einen 1fach und einen 4fach Busverbinder:



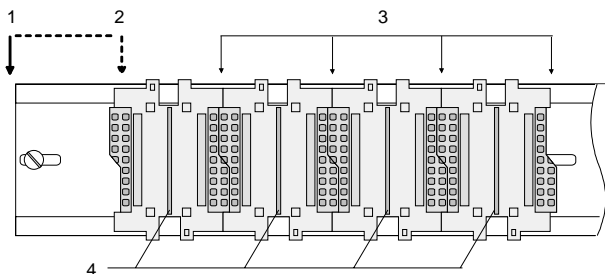
Der Busverbinder wird in die Profilschiene eingelegt, bis dieser sicher einrastet, so dass die Bus-Anschlüsse aus der Profilschiene herauschauen.

Bestellnummer	Bezeichnung	Beschreibung
290-0AA10	Busverbinder	1fach
290-0AA20	Busverbinder	2fach
290-0AA40	Busverbinder	4fach
290-0AA80	Busverbinder	8fach

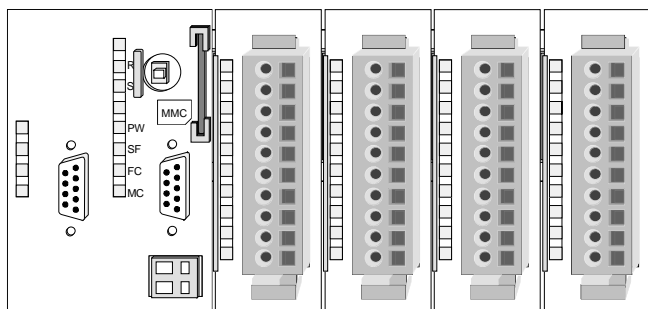
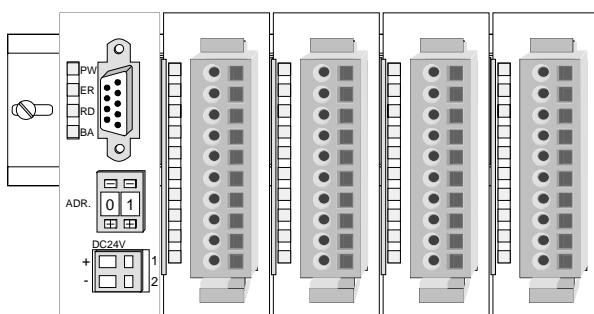
**Montage auf Profilschiene**

Die nachfolgende Skizze zeigt einen 4fach-Busverbinder in einer Profilschiene und die Steckplätze für die Module.

Die einzelnen Modulsteckplätze sind durch Führungsleisten abgegrenzt.



- [1] Kopfmodul (doppelt breit)
- [2] Kopfmodul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten



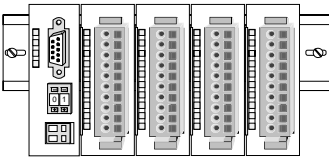
**Montage unter Berücksichtigung der Stromaufnahme**

- Verwenden Sie möglichst lange Busverbinder.
- Ordnen Sie Module mit hohem Stromverbrauch direkt rechts neben Ihrem Kopfmodul an. Im Service-Bereich von [www.vipa.com](http://www.vipa.com) finden Sie alle Stromaufnahmen des System 200V in einer Liste zusammengefasst.

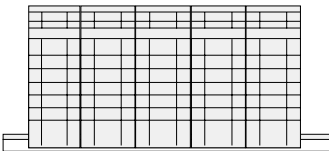


## Montagemöglichkeiten

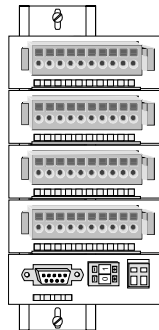
### waagrechter Aufbau



### liegender Aufbau



### senkrechter Aufbau

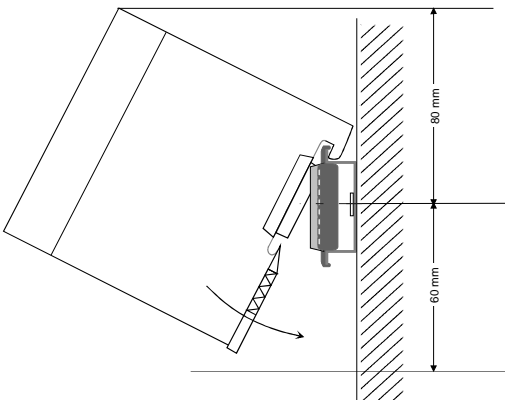


Beachten Sie bitte die hierbei zulässigen Umgebungstemperaturen:

- waagrechter Aufbau: von 0 bis 60°C
- senkrechter Aufbau: von 0 bis 40°C
- liegender Aufbau: von 0 bis 40°C

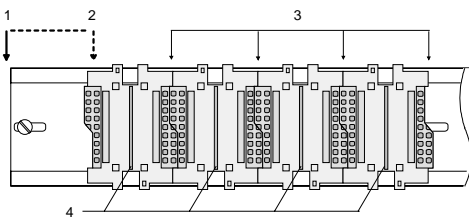
Der waagrechte Aufbau beginnt immer links mit einem Kopfmodul. Rechts daneben sind die Peripherie-Module zu stecken.

Es dürfen bis zu 32 Peripherie-Module gesteckt werden.



### Bitte bei der Montage beachten!

- Schalten Sie die Stromversorgung aus bevor Sie Module stecken bzw. abziehen!
- Halten Sie ab der Mitte der Profilschiene nach oben einen Montageabstand von mindestens 80mm und nach unten von 60mm ein.



- Eine Zeile wird immer von links nach rechts aufgebaut und beginnt immer mit einem Kopfmodul.

- [1] Kopfmodul (doppelt breit)
- [2] Kopfmodul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten

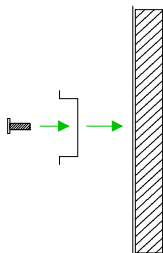
- Module müssen immer direkt nebeneinander gesteckt werden. Lücken sind nicht zulässig, da ansonsten der Rückwandbus unterbrochen ist.
- Ein Modul ist erst dann gesteckt und elektrisch verbunden, wenn es hörbar einrastet.
- Steckplätze rechts nach dem letzten Modul dürfen frei bleiben.



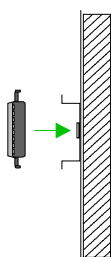
### Hinweis!

Am Rückwandbus dürfen sich maximal 32 Module befinden. Hierbei darf der **Summenstrom** von **3,5A** darf nicht überschritten werden!

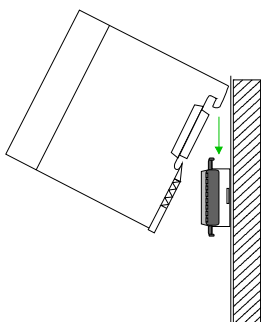
**Montage  
Vorgehensweise**



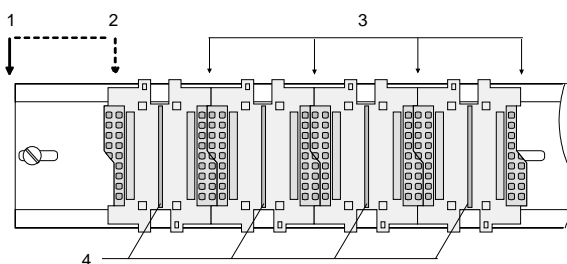
- Montieren Sie die Profilschiene. Bitte beachten Sie, dass Sie ab der Mitte der Profilschiene nach oben einen Modul-Montageabstand von mindestens 80mm und nach unten von 60mm einhalten.



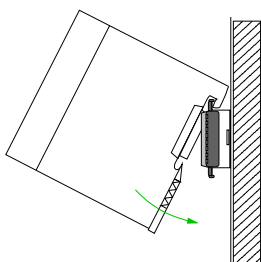
- Drücken Sie den Busverbinder in die Profilschiene, bis dieser sicher einrastet, so dass die Bus-Anschlüsse aus der Profilschiene heraus-schauen. Sie haben nun die Grundlage zur Montage Ihrer Module.



- Beginnen Sie ganz links mit dem Kopfmodul, wie CPU, PC oder Bus-koppler und stecken Sie rechts daneben Ihre Peripherie-Module.



- [1] Kopfmodul (doppelt breit)
- [2] Kopfmodul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten

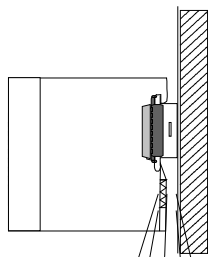


- Setzen Sie das zu steckende Modul von oben in einem Winkel von ca. 45 Grad auf die Profilschiene und drehen Sie das Modul nach unten, bis es hörbar auf der Profilschiene einrastet. Nur bei eingerasteten Modulen ist eine Verbindung zum Rückwandbus sichergestellt.



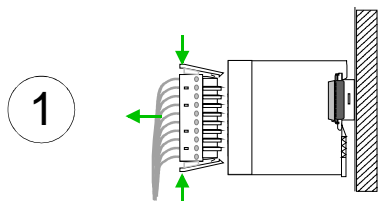
**Achtung!**

Module dürfen nur im spannungslosen Zustand ge-steckt bzw. gezogen werden!

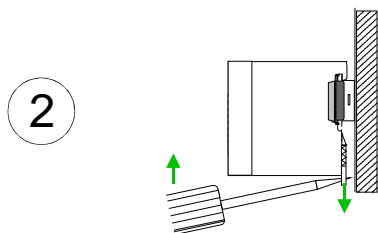


Clack

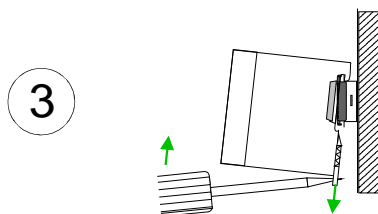
## Demontage und Modultausch



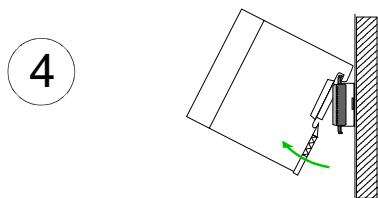
- Entfernen Sie falls vorhanden die Verdrahtung an dem Modul, indem Sie die beiden Verriegelungshebel am Steckverbinder betätigen und den Steckverbinder abziehen.



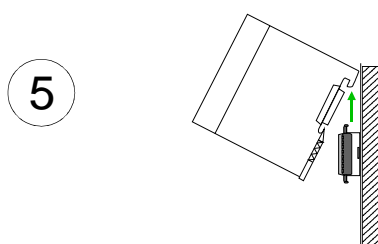
- Zur Demontage des Moduls befindet sich am Gehäuseunterteil eine gefederter Demontageschlitz. Stecken Sie, wie gezeigt, einen Schraubendreher in den Demontageschlitz.



- Entriegeln Sie durch Druck des Schraubendrehers nach oben das Modul.



- Ziehen Sie nun das Modul nach vorn und ziehen Sie das Modul mit einer Drehung nach oben ab.



### Achtung!

Module dürfen nur im spannungslosen Zustand gesteckt bzw. gezogen werden!

Bitte beachten Sie, dass durch die Demontage von Modulen der Rückwandbus an der entsprechenden Stelle unterbrochen wird!

# Verdrahtung

## Übersicht

Die meisten Peripherie-Module besitzen einen 10poligen bzw. 18poligen Steckverbinder. Über diesen Steckverbinder werden Signal- und Versorgungsleitungen mit den Modulen verbunden.

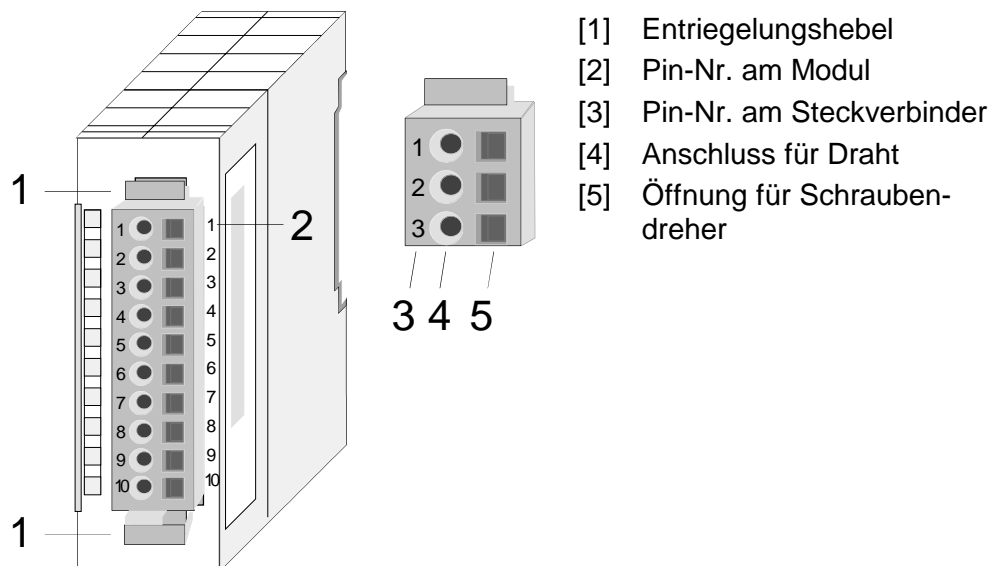
Bei der Verdrahtung werden Steckverbinder mit Federklemmtechnik eingesetzt.

Die Verdrahtung mit Federklemmtechnik ermöglicht einen schnellen und einfachen Anschluss Ihrer Signal- und Versorgungsleitungen.

Im Gegensatz zur Schraubverbindung, ist diese Verbindungsart erschütterungssicher. Die Steckerbelegung der Peripherie-Module finden Sie in der Beschreibung zu den Modulen.

Sie können Drähte mit einem Querschnitt von 0,08mm<sup>2</sup> bis 2,5mm<sup>2</sup> (bis 1,5mm<sup>2</sup> bei 18poligen Steckverbindern) anschließen.

Folgende Abbildung zeigt ein Modul mit einem 10poligen Steckverbinder.

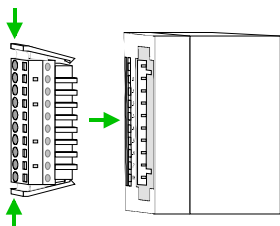


### Hinweis!

Die Federklemme wird zerstört, wenn Sie den Schraubendreher in die Öffnung für die Leitungen stecken!

Drücken Sie den Schraubendreher nur in die rechteckigen Öffnungen des Steckverbinders!

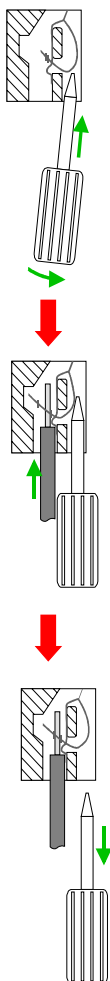
## Verdrahtung Vorgehensweise



- Stecken Sie den Steckverbinder auf das Modul bis dieser hörbar einrastet. Drücken Sie hierzu während des Steckens, wie gezeigt, die beiden Verriegelungsklinken zusammen.

Der Steckverbinder ist nun in einer festen Position und kann leicht verdrahtet werden.

Die nachfolgende Abfolge stellt die Schritte der Verdrahtung in der Draufsicht dar.



- Zum Verdrahten stecken Sie, wie in der Abbildung gezeigt, einen passenden Schraubendreher leicht schräg in die rechteckige Öffnung.
- Zum Öffnen der Kontaktfeder müssen Sie den Schraubendreher in die entgegengesetzte Richtung drücken und halten.

- Führen Sie durch die runde Öffnung Ihren abisolierten Draht ein. Sie können Drähte mit einem Querschnitt von  $0,08\text{mm}^2$  bis  $2,5\text{mm}^2$  (bei 18poligen Steckverbindern bis  $1,5\text{mm}^2$ ) anschließen.

- Durch Entfernen des Schraubendrehers wird der Draht über einen Federkontakt sicher mit dem Steckverbinder verbunden.



### Hinweis!

Verdrahten Sie zuerst die Versorgungsleitungen (Spannungsversorgung) und dann die Signalleitungen (Ein- und Ausgänge)!

## Aufbaurichtlinien

- Allgemeines** Die Aufbaurichtlinien enthalten Informationen über den stör sicheren Aufbau von System 200V Systemen. Es werden die Wege beschrieben, wie Störungen in Ihre Steuerung gelangen können, wie die elektromagnetische Verträglichkeit (EMV), sicher gestellt werden kann und wie bei der Schirmung vorzugehen ist.
- Was bedeutet EMV?** Unter Elektromagnetischer Verträglichkeit (EMV) versteht man die Fähigkeit eines elektrischen Gerätes, in einer vorgegebenen elektromagnetischen Umgebung fehlerfrei zu funktionieren ohne vom Umfeld beeinflusst zu werden bzw. das Umfeld in unzulässiger Weise zu beeinflussen.
- Alle System 200V Komponenten sind für den Einsatz in rauen Industrieumgebungen entwickelt und erfüllen hohe Anforderungen an die EMV. Trotzdem sollten Sie vor der Installation der Komponenten eine EMV-Planung durchführen und mögliche Störquellen in die Betrachtung einbeziehen.
- Mögliche Störeinträge** Elektromagnetische Störungen können sich auf unterschiedlichen Pfaden in Ihre Steuerung einkoppeln:
- Felder
  - E/A-Signalleitungen
  - Bussystem
  - Stromversorgung
  - Schutzleitung
- Je nach Ausbreitungsmedium (leitungsgebunden oder -ungebunden) und Entfernung zur Störquelle gelangen Störungen über unterschiedliche Kopplungsmechanismen in Ihre Steuerung.
- Man unterscheidet:
- galvanische Kopplung
  - kapazitive Kopplung
  - induktive Kopplung
  - Strahlungskopplung

**Grundregeln zur Sicherstellung der EMV**

Häufig genügt zur Sicherstellung der EMV das Einhalten einiger elementarer Regeln. Beachten Sie beim Aufbau der Steuerung deshalb die folgenden Grundregeln.

- Achten Sie bei der Montage Ihrer Komponenten auf eine gut ausgeführte flächenhafte Massung der inaktiven Metallteile.
  - Stellen Sie eine zentrale Verbindung zwischen der Masse und dem Erde/Schutzleitersystem her.
  - Verbinden Sie alle inaktiven Metallteile großflächig und impedanzarm.
  - Verwenden Sie nach Möglichkeit keine Aluminiumteile. Aluminium oxidiert leicht und ist für die Massung deshalb weniger gut geeignet.
- Achten Sie bei der Verdrahtung auf eine ordnungsgemäße Leitungsführung.
  - Teilen Sie die Verkabelung in Leitungsgruppen ein. (Starkstrom, Stromversorgungs-, Signal- und Datenleitungen).
  - Verlegen Sie Starkstromleitungen und Signal- bzw. Datenleitungen immer in getrennten Kanälen oder Bündeln.
  - Führen Sie Signal- und Datenleitungen möglichst eng an Masseflächen (z.B. Tragholme, Metallschienen, Schrankbleche).
- Achten Sie auf die einwandfreie Befestigung der Leitungsschirme.
  - Datenleitungen sind geschirmt zu verlegen.
  - Analogleitungen sind geschirmt zu verlegen. Bei der Übertragung von Signalen mit kleinen Amplituden kann das einseitige Auflegen des Schirms vorteilhaft sein.
  - Legen Sie die Leitungsschirme direkt nach dem Schrankeintritt großflächig auf eine Schirm-/Schutzleiterschiene auf, und befestigen Sie die Schirme mit Kabelschellen.
  - Achten Sie darauf, dass die Schirm-/Schutzleiterschiene impedanzarm mit dem Schrank verbunden ist.
  - Verwenden Sie für geschirmte Datenleitungen metallische oder metallisierte Steckergehäuse.
- Setzen Sie in besonderen Anwendungsfällen spezielle EMV-Maßnahmen ein.
  - Erwägen Sie bei Induktivitäten den Einsatz von Löschgliedern.
  - Beachten Sie, dass bei Einsatz von Leuchtstofflampen sich diese negativ auf Signalleitungen auswirken können.
- Schaffen Sie ein einheitliches Bezugspotential und erden Sie nach Möglichkeit alle elektrischen Betriebsmittel.
  - Achten Sie auf den gezielten Einsatz der Erdungsmaßnahmen. Das Erden der Steuerung dient als Schutz- und Funktionsmaßnahme.
  - Verbinden Sie Anlagenteile und Schränke mit dem System 200V sternförmig mit dem Erde/Schutzleitersystem. Sie vermeiden so die Bildung von Erdschleifen.
  - Verlegen Sie bei Potenzialdifferenzen zwischen Anlagenteilen und Schränken ausreichend dimensionierte Potenzialausgleichsleitungen.

## Schirmung von Leitungen

Elektrische, magnetische oder elektromagnetische Störfelder werden durch eine Schirmung geschwächt; man spricht hier von einer Dämpfung.

Über die mit dem Gehäuse leitend verbundene Schirmschiene werden Störströme auf Kabelschirme zur Erde hin abgeleitet. Hierbei ist darauf zu achten, dass die Verbindung zum Schutzleiter impedanzarm ist, da sonst die Störströme selbst zur Störquelle werden.

Bei der Schirmung von Leitungen ist folgendes zu beachten:

- Verwenden Sie möglichst nur Leitungen mit Schirmgeflecht.
- Die Deckungsdichte des Schirmes sollte mehr als 80% betragen.
- In der Regel sollten Sie die Schirme von Leitungen immer beidseitig auflegen. Nur durch den beidseitigen Anschluss der Schirme erreichen Sie eine gute Störunterdrückung im höheren Frequenzbereich.

Nur im Ausnahmefall kann der Schirm auch einseitig aufgelegt werden. Dann erreichen Sie jedoch nur eine Dämpfung der niedrigen Frequenzen. Eine einseitige Schirmanbindung kann günstiger sein, wenn:

- die Verlegung einer Potenzialausgleichsleitung nicht durchgeführt werden kann
  - Analogsignale (einige mV bzw.  $\mu\text{A}$ ) übertragen werden
  - Folienschirme (statische Schirme) verwendet werden.
- Benutzen Sie bei Datenleitungen für serielle Kopplungen immer metallische oder metallisierte Stecker. Befestigen Sie den Schirm der Datenleitung am Steckergehäuse. Schirm nicht auf den PIN 1 der Steckerleiste auflegen!
  - Bei stationärem Betrieb ist es empfehlenswert, das geschirmte Kabel unterbrechungsfrei abzuisolieren und auf die Schirm-/Schutzleiterschiene aufzulegen.
  - Benutzen Sie zur Befestigung der Schirmgeflechte Kabelschellen aus Metall. Die Schellen müssen den Schirm großflächig umschließen und guten Kontakt ausüben.
  - Legen Sie den Schirm direkt nach Eintritt der Leitung in den Schrank auf eine Schirmschiene auf. Führen Sie den Schirm bis zum System 200V Modul weiter, legen Sie ihn dort jedoch **nicht** erneut auf!



### Bitte bei der Montage beachten!

Bei Potentialdifferenzen zwischen den Erdungspunkten kann über den beidseitig angeschlossenen Schirm ein Ausgleichsstrom fließen.

Abhilfe: Potenzialausgleichsleitung.



## Allgemeine Daten

### Aufbau/Maße

- Profilschiene 35mm
- Peripherie-Module mit seitlich versenkbaaren Beschriftungsstreifen
- Maße Grundgehäuse:  
1fach breit: (HxBxT) in mm: 76x25,4x74 in Zoll: 3x1x3  
2fach breit: (HxBxT) in mm: 76x50,8x74 in Zoll: 3x2x3

### Betriebssicherheit

- Anschluss über Federzugklemmen an Frontstecker, Aderquerschnitt 0,08 ... 2,5mm<sup>2</sup> bzw. 1,5 mm<sup>2</sup> (18-fach Stecker)
- Vollisolierung der Verdrahtung bei Modulwechsel
- Potenzialtrennung aller Module zum Rückwandbus
- ESD/Burst gemäß IEC 61000-4-2 / IEC 61000-4-4 (bis Stufe 3)
- Schockfestigkeit gemäß IEC 60068-2-6 / IEC 60068-2-27 (1G/12G)
- Schutzklasse IP20

### Umgebungsbedingungen

- Betriebstemperatur: 0 ... +60°C
- Lagertemperatur: -25 ... +70°C
- Relative Feuchte: 5 ... 95% ohne Betauung
- Lüfterloser Betrieb



## Teil 2 Hardwarebeschreibung

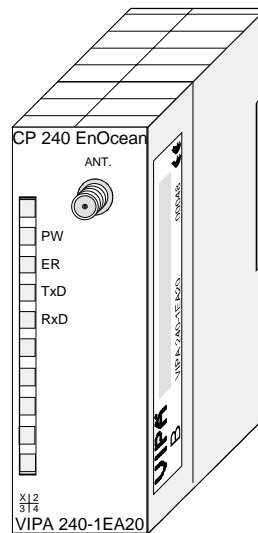
**Übersicht** In diesem Kapitel finden Sie Informationen über den Aufbau und die Anschlussbelegung des Kommunikationsprozessors CP 240 mit EnOcean Transceiver-Modul.

Inhalt	Thema	Seite
	<b>Teil 2 Hardwarebeschreibung</b> .....	<b>2-1</b>
	Leistungsmerkmale .....	2-2
	Aufbau.....	2-3
	Technische Daten .....	2-5

## Leistungsmerkmale

### CP 240 EnOcean 240-1EA20

- CP mit EnOcean Funktransceiver-Modul
- 16Byte Parameterdaten
- Spannungsversorgung über Rückwandbus
- Das TCM 120 Transceiver-Modul arbeitet bei 868,3MHz

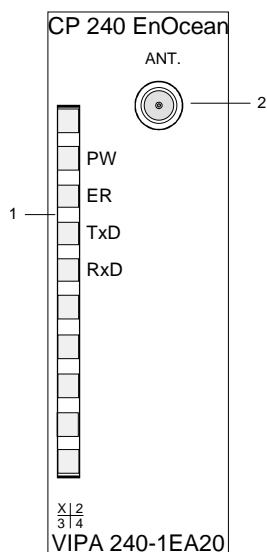


### Bestelldaten

Typ	Bestellnummer	Beschreibung
CP 240 EnOcean	VIPA 240-1EA20	CP mit EnOcean Funktransceiver-Modul TCM 120
Portable Antenne	VIPA 240-0EA00	Portable Antenne mit SMA-Stecker
Magnetfuß Antenne	VIPA 240-0EA10	Magnetfußantenne mit 150cm Kabel und SMA-Stecker

## Aufbau

### CP 240 EnOcean 240-1EA20



- [1] LED Statusanzeigen
- [2] SMA-Antennenbuchse mit Außengewinde und Kelch

### Schnittstelle



### Antennen

Im Lieferumfang ist keine Antenne enthalten. Sie können aber optional eine portable Antenne oder eine Magnetfußantenne mit 150cm Kabel bestellen.

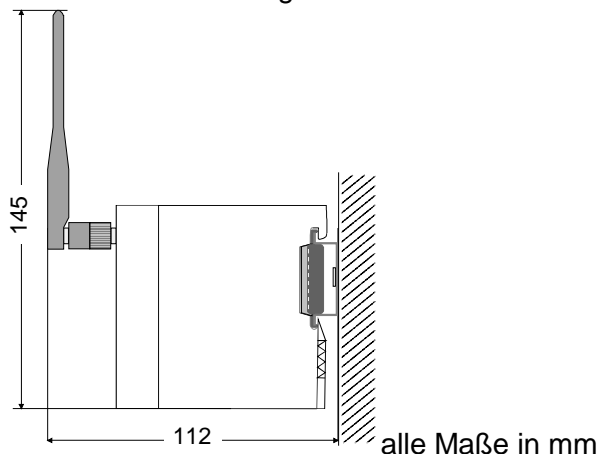
Beide Antennen sind mit einem SMA-Stecker ausgestattet. Der koaxial aufgebaute SMA-Stecker (**stright medium adaptor**) ist ein Miniatur-HF-Stecker mit Gewindeverschluss, der sich durch eine hohe HF-Dichtigkeit auszeichnet. In der Standardversion hat der Stecker eine Überwurfmutter mit Innengewinde und einem Stift.

Die SMA-Buchse, die sich am CP befindet, bildet mit dem Außengewinde und dem Kelch das Gegenstück für die Montage.

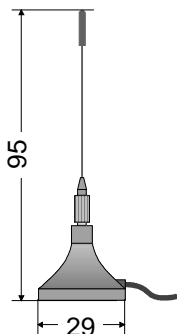
### Portable Antenne

Bei der Portable Antenne handelt es sich um eine kurze Stabantenne, die über den SMA-Stecker ohne Kabel direkt am Modul montiert wird.

Die Antenne kann abgewinkelt und in alle Richtungen gedreht werden.



Magnetfußantenne Die Magnetfußantenne mit 150cm Kabel ist für den Einbau in Schaltschränke geeignet. Aufgrund des Magnetfußes können Sie die Antenne an allen Stahl-Flächen befestigen. Der Anschluss der Magnetfußantennen an den CP 240 EnOcean erfolgt über das 150cm lange Antennenkabel mit SMA-Stecker.



alle Maße in mm

**Spannungsversorgung**

Der Kommunikationsprozessor bezieht seine Versorgungsspannung über den Rückwandbus.

**LEDs**

Der Kommunikationsprozessor besitzt 4 LEDs zur Anzeige des Betriebszustands. Die Bedeutung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bez.	Farbe	Bedeutung
PW	Grün	Signalisiert eine anliegende Betriebsspannung
ER	Rot	Signalisiert einen Fehler durch Pufferüberlauf
TxD	Grün	Daten senden (transmit data)
RxD	Grün	Daten empfangen (receive data)

## Technische Daten

<b>Artikelnummer</b>	<b>240-1EA20</b>
Bezeichnung	CP 240, EnOcean
<b>Stromaufnahme/Verlustleistung</b>	
Stromaufnahme aus Rückwandbus	120 mA
Verlustleistung	0,75 W
<b>Status, Alarm, Diagnosen</b>	
Statusanzeige	ja
Alarmer	nein
Prozessalarm	nein
Diagnosealarm	nein
Diagnosefunktion	nein
Diagnoseinformation auslesbar	keine
Versorgungsspannungsanzeige	ja
Sammelfehleranzeige	rote LED
Kanalfehleranzeige	keine
<b>Funktionalität Sub-D Schnittstellen</b>	
Bezeichnung	-
Physik	-
Anschluss	-
Potenzialgetrennt	-
MPI	-
MP?I (MPI/RS232)	-
DP-Master	-
DP-Slave	-
Punkt-zu-Punkt-Kopplung	-
<b>Point-to-Point Kommunikation</b>	
PtP-Kommunikation	-
Schnittstelle potentialgetrennt	-
Schnittstelle RS232	-
Schnittstelle RS422	-
Schnittstelle RS485	-
Anschluss	SMA-Buchse für Antenne
Übertragungsgeschwindigkeit, min.	-
Übertragungsgeschwindigkeit, max.	9,6 kbit/s
Leitungslänge, max.	-
<b>Point-to-Point Protokolle</b>	
Protokoll ASCII	-
Protokoll STX/ETX	-
Protokoll 3964(R)	-
Protokoll RK512	-
Protokoll USS Master	-
Protokoll Modbus Master	-
Protokoll Modbus Slave	-
Spezielle Protokolle	EnOcean
<b>Datengrößen</b>	
Eingangsbytes	16
Ausgangsbytes	16
Parameterbytes	16
Diagnosebytes	0
<b>Gehäuse</b>	
Material	PPE
Befestigung	Profilschiene 35mm
<b>Mechanische Daten</b>	
Abmessungen (BxHxT)	25,4 x 76 x 78 mm

<b>Artikelnummer</b>	<b>240-1EA20</b>
Gewicht	80 g
<b>Umgebungsbedingungen</b>	
Betriebstemperatur	0 °C bis 60 °C
Lagertemperatur	-25 °C bis 70 °C
<b>Zertifizierungen</b>	
Zertifizierung nach UL508	ja

**Hinweis!**

Bitte beachten Sie, dass für den Einsatz des Moduls nationale Richtlinien eingehalten werden müssen!

Die Einhaltung dieser Richtlinien obliegt dem Benutzer!



## Teil 3 Einsatz

**Übersicht** Hier finden Sie Informationen über den Einsatz des Kommunikationsprozessors CP 240 EnOcean.

Inhalt	Thema	Seite
	<b>Teil 3 Einsatz</b> .....	<b>3-1</b>
	Grundlagen .....	3-2
	Schnelleinstieg .....	3-3
	GSD und FCs einbinden.....	3-5
	Projektierung .....	3-6
	Standardhantierungsbausteine.....	3-9
	Kommunikationsprinzip .....	3-12
	Beispiel zum Einsatz unter EnOcean.....	3-14
	Übersicht der EnOcean-Telegramme .....	3-19
	Modul ersetzen und IDBase übernehmen.....	3-34

## Grundlagen

### EnOcean

EnOcean ist ein batterieloses Funksystem, das im Jahre 2001 von der Firma EnOcean entwickelt wurde. Aufgrund der kurzen Signaldauer von 0,5ms und 10mW Sendeleistung hat die Funkübertragungstechnik einen Energiebedarf von 50µWs. Hierbei nutzt das System die Energie aus kleinsten Veränderungen von Druck oder Temperatur zur Stromversorgung der Sensoren.

Die Reichweite der Sensoren beträgt bis zu 300m im Freien. Jeder Sender erhält zudem bereits bei der Herstellung eine eindeutige 32Bit Adresse als ID. Die Module nutzen das international zugelassene SRD-Frequenzband bei 869 MHz.

Einsatzschwerpunkte von EnOcean sind Gebäudeautomation, industrielle Produktion und Automobiltechnik.

### Eigenschaften

- Minimale Energieanforderungen
- Unterstützung mehrerer Sender in nächster Umgebung
- Telegrammdauer 0,5ms
- Übertragungreichweite bis zu 300m
- Uni- und bidirektionale Kommunikation
- Einfache Erweiterbarkeit

### Amplitudenmodulation

Als Modulationsverfahren kommt bei EnOcean die inkohärente Amplitudenmodulation (ASK) zum Einsatz. Ihre Fehlerwahrscheinlichkeit ist gegenüber der Frequenzmodulation bei gleichem Störsignalpegel in etwa gleichwertig. Die Digitale Amplitudenmodulation gestattet die Realisierung energiesparender Sender, da hier nur die "1"-Bits übertragen werden.

### Sicherheit durch Telegrammwiederholung

Die Übertragung eines Datentelegramms dauert ca. 0,5ms. Zur Erhöhung der Datensicherheit wird jedes Telegramm innerhalb von 40ms zweimal wiederholt, wobei der zeitliche Abstand zwischen jeder Wiederholung zufällig gewählt wird.

Diese schnelle Mehrfachaussendung ermöglicht, dass viele benachbarte Sender parallel auf einer gemeinsamen Funkfrequenz mit niedriger Fehlerquote arbeiten können.

### IDs zur Adressierung

EnOcean verwendet zur Adressierung IDs. Eine ID setzt sich zusammen aus einer *IDBase* und einem frei konfigurierbaren *Bitbereich*. Da die EnOcean-Module von VIPA mit einer unterschiedlichen *IDBase* ausgeliefert werden, empfiehlt es sich bei umfangreichen Projekten die *IDBase* aller Module zu notieren. Somit können Sie im Fehlerfall ein Modul ersetzen und die entsprechende *IDBase* übernehmen.

Näheres hierzu finden Sie unter "Modul ersetzen und *IDBase* übernehmen".

## Schnelleinstieg

### Übersicht

Der Kommunikationsprozessor CP 240 EnOcean ermöglicht die Prozess- ankopplung an verschiedene Ziel- oder Quell-Systeme auf Basis der draht- losen EnOcean-Kommunikation.

Der CP 240 EnOcean wird über den Rückwandbus mit Spannung versorgt. Zur internen Kommunikation sind VIPA FCs zu verwenden. Für die Projektierung des CP 240 EnOcean in Verbindung mit einer CPU 21x im Siemens SIMATIC Manager, ist die Einbindung der GSD VIPA\_21x.gsd erforderlich. Damit der CP 240 EnOcean mit der CPU kommunizieren kann, ist für das System immer eine Hardware-Konfiguration durchzuführen.

Eine allgemeine Beschreibung zur Projektierung des CP 240 finden Sie unter "Projektierung".

### Vorgehensweise

#### Vorbereitung

- Starten Sie den Siemens SIMATIC Manager mit einem neuen Projekt.
- Binden Sie die VIPA\_21x.gsd ein. Verwenden Sie hierbei eine GSD- Version ab V. 1.67.
- Binden Sie die Bausteinbibliothek ein, indem Sie die Vipa\_Bibliothek\_ Vxxx.zip entpacken und die Datei VIPA.ZIP dearchivieren.
- Öffnen Sie die Bibliothek und übertragen Sie die gewünschten FCs in Ihr Projekt.

#### Hardware- Konfiguration

Für die Hardwarekonfiguration verfahren Sie auf die gleiche Weise wie im Handbuch HB97 - CPU beschrieben:

- Projektieren Sie ein PROFIBUS-DP-Mastersystem mit der Siemens CPU 315-2DP (6ES7 315-2AF03 V1.2) und legen Sie ein PROFIBUS- Subnetz an.
- Binden Sie an das Master-System aus dem Hardware-Katalog das Slave-System "VIPA\_CPU21x" an. Sie finden das Slave-System im Hardware-Katalog unter *PROFIBUS-DP > Weitere Feldgeräte > I/O > VIPA\_System\_200V*.
- Geben Sie dem Slave-System die Adresse 1. Hiermit identifiziert die VIPA CPU das System als zentrales Peripherie-System.
- Platzieren Sie in diesem Slave-System in der gesteckten Reihenfolge Ihre Module. Beginnen sie mit der CPU auf dem 1. Steckplatz.
- Binden Sie danach Ihre System 200V Module und an der entsprechenden Stelle Ihren CP 240 EnOcean ein.
- Parametrieren Sie Ihren CP 240 EnOcean.

**Parameter**

Durch Platzieren des CP 240 EnOcean in der Hardware-Konfiguration im "virtuellen" PROFIBUS-System werden automatisch die erforderlichen Parameter angelegt. Der Parameterbereich hat folgenden Aufbau:

Byte	Funktion	Wertebereich	Defaultparameter
0	reserviert		
1	Protokoll	E0h: EnOcean	-
2...15	reserviert		

Hier ist lediglich im Byte 1 als Protokoll E0h für EnOcean anzugeben. Die restlichen Parameter sind reserviert und werden nicht ausgewertet.

**Interne Kommunikation**

Mit VIPA-FCs steuern Sie die Kommunikation zwischen CPU und CP 240. Hierbei steht für Sende- und Empfangsdaten je ein 2048Byte großer Puffer zur Verfügung, der maximal 150 Telegramme verwalten kann. In Verbindung mit einer CPU 21x kommen folgende Hantierungsbausteine zum Einsatz:

Name	FCs	Kurzbeschreibung
SEND	FC0	Sende-Baustein
RECEIVE	FC1	Receive-Baustein
SYNCHRON_RESET	FC9	Reset und Synchronisation des CP 240

**11Byte Telegramm für EnOcean-Kommunikation**

Verwenden Sie für die Kommunikation immer Telegramme mit einer Länge von 11Byte. Beim Senden werden im CP 240 EnOcean die 11Byte automatisch mit 2 Synchronisations-Bytes und einer Checksumme auf 14Byte ergänzt bzw. beim Empfang das 14Byte große Telegramm auf 11Byte beschnitten.

## GSD und FCs einbinden

<b>Projektierung über GSD</b>	Adresszuordnung und die Parametrierung des CP 240 erfolgt im Siemens SIMATIC Manager in Form eines virtuellen PROFIBUS-Systems. Da die PROFIBUS-Schnittstelle softwareseitig standardisiert ist, können wir auf diesem Weg gewährleisten, dass über die Einbindung einer GSD-Datei die Funktionalität in Verbindung mit dem SIMATIC Manager von Siemens jederzeit gegeben ist. Ihr Projekt übertragen Sie über MPI in die CPU.
GSD einbinden	<p>Folgende Schritte sind zur Installation der GSD erforderlich:</p> <ul style="list-style-type: none"><li>• Im Service-Bereich von <a href="http://www.vipa.com">www.vipa.com</a> finden Sie die GSD-Datei für das System 200V. Laden Sie die zip-Datei auf Ihren PC.</li><li>• Starten Sie mit einem Doppelklick auf die Datei Ihr Unzip-Programm und entpacken Sie die Daten in Ihr Arbeitsverzeichnis.</li><li>• Kopieren Sie die GSD-Datei <b>VIPA_21X.GSD</b> in Ihr GSD-Verzeichnis ... \siemens\step7\s7data\gsd</li><li>• Starten Sie den Hardware-Konfigurator von Siemens</li><li>• Schließen Sie alle Projekte</li><li>• Gehen Sie auf <b>Extras</b> &gt; <i>Neue GSD-Datei installieren</i></li><li>• Geben Sie hier <b>VIPA_21X.gsd</b> an</li></ul> <p>Die Module des System 200V von VIPA sind jetzt im Hardwarekatalog integriert und können projiziert werden.</p>
<b>Bausteine installieren</b>	<p>Die VIPA-spezifischen Bausteine finden Sie im Service-Bereich auf <a href="http://www.vipa.com">www.vipa.com</a> als Bibliothek zum Download. Die Bibliothek liegt als gepackte zip-Datei vor.</p> <p>Sobald Sie VIPA-spezifische Bausteine verwenden möchten, sind diese in Ihr Projekt zu importieren.</p>
Bibliothek dearchivieren	<p>Starten Sie mit einem Doppelklick auf die Datei <i>Vipa_Bibliothek_Vxxx.zip</i> Ihr Unzip-Programm und kopieren Sie die Datei <i>vipa.zip</i> in Ihr Arbeitsverzeichnis. Es ist nicht erforderlich diese Datei weiter zu entpacken.</p> <p>Zur Dearchivierung Ihrer Bibliothek für die SPEED7-CPU's starten Sie den SIMATIC Manager von Siemens. Über <b>Datei</b> &gt; <i>Dearchivieren</i> öffnen Sie ein Dialogfenster zur Auswahl des Archivs. Navigieren Sie in Ihr Arbeitsverzeichnis.</p> <p>Wählen Sie <i>VIPA.ZIP</i> an und klicken Sie auf [Öffnen].</p> <p>Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind. Mit [OK] startet der Entpackvorgang.</p>
Bibliothek öffnen und Bausteine in Projekt übertragen	<p>Öffnen Sie die Bibliothek nach dem Entpackvorgang.</p> <p>Öffnen Sie Ihr Projekt und kopieren Sie die erforderlichen Bausteine aus der Bibliothek in das Verzeichnis "Bausteine" Ihres Projekts.</p> <p>Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die VIPA-spezifischen Bausteine.</p>

# Projektierung

## Allgemein

Die Adresszuordnung und die Parametrierung der direkt gesteckten System 200V Module erfolgt im SIMATIC Manager von Siemens in Form eines virtuellen PROFIBUS-Systems. Ihr Projekt übertragen Sie seriell über die MPI-Schnittstelle oder über MMC in Ihre CPU.

## Voraussetzung

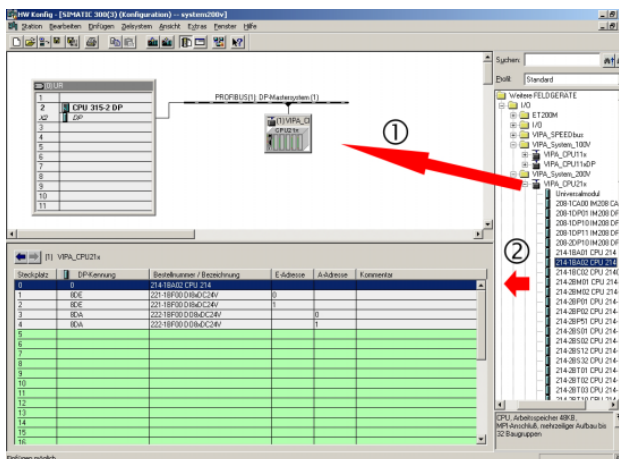
Für die Projektierung der CPU werden fundierte Kenntnisse im Umgang mit dem SIMATIC Manager und dem Hardware-Konfigurator von Siemens vorausgesetzt!

Folgende Voraussetzungen müssen für die Projektierung erfüllt sein:

- SIMATIC Manager von Siemens auf PC bzw. PG installiert
- GSD-Dateien in Hardware-Konfigurator von Siemens eingebunden
- Projekt kann in CPU übertragen werden (seriell z.B. "Green Cable" oder MMC)

## Hardware-Konfiguration

- Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt und fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
- Platzieren Sie auf dem ersten möglichen Steckplatz die CPU 315-2DP (6ES7 315-2AF03 V1.2) von Siemens.
- Sofern Ihre CPU 21x einen PROFIBUS-DP-Master integriert hat, können Sie diesen jetzt mit PROFIBUS vernetzen und Ihre DP-Slaves anbinden.
- Erzeugen Sie ein PROFIBUS-Subnetz (falls noch nicht vorhanden).
- Hängen Sie an das Subnetz das System "VIPA\_CPU21x". Sie finden dies im Hardware-Katalog unter *PROFIBUS DP > Weitere Feldgeräte > IO > VIPA\_System\_200V*. Geben Sie diesem Slave die **PROFIBUS-Adresse 1**.
- Platzieren Sie in Ihrem Konfigurator **immer auf dem 1. Steckplatz** die CPU 21x, die Sie einsetzen, indem Sie diese dem Hardware-Katalog entnehmen.
- Binden Sie danach Ihre System 200V Module in der gesteckten Reihenfolge und an der entsprechenden Stelle Ihren CP 240 ein.
- Parametrieren Sie ggf. Ihren CP 240.
- Sichern Sie Ihr Projekt.



**SPS-Programm**

Für die nachfolgend gezeigte Kommunikation zwischen CPU und CP 240 kommen folgende Hantierungsbausteine zum Einsatz:

FC 0	SEND	Datenausgabe CPU an CP 240
FC 1	RECEIVE	Datenempfang vom CP 240
FC 9	SYNCHRON_RESET	Synchronisation zwischen CPU und CP 240

Die Hantierungsbausteine sind als Bibliothek verfügbar und können, wie weiter oben gezeigt, im Siemens SIMATIC Manager eingebunden werden.

Eine nähere Beschreibung der Hantierungsbausteine finden Sie auf den Folgeseiten. Ihr SPS-Programm sollte nach folgender Struktur aufgebaut sein:

```

OB1:
CALL FC      9                //Synchron aufrufen
  ADR        :=0              //1. DW im SEND/EMPF_DB
  TIMER_NR   :=T2             //Wartezeit Synchron
  ANL        :=M3.0           //Anlauf erfolgt
  NULL       :=M3.1           //Zwischenmerker
  RESET      :=M3.2           //Baugruppenreset ausführen
  STEUERB_S  :=MB2            //Steuerbits Sende_FC
  STEUERB_R  :=MB1            //Steuerbits Receive_FC
  U          M      3.0       //solange Anlauf keine
                                //SEND/RECEIVE Bearbeitung
  BEB

CALL FC      1                //Receive Daten
  ADR        :=0              //1. DW im SEND/EMPF_DB
  _DB        :=DB11           //Empfang_DB Telegramm
  ABD        :=W#16#14        //1. DW Empfangspuffer (DW20)
  ANZ        :=MW10           //Anzahl empfangener Daten
  EMFR       :=M1.0           //Empfang fertig
  PAFE       :=MB12           //Fehlerbyte
  GEEM       :=MW100          //Interne Daten
  ANZ_INT    :=MW102          //Interne Daten
  empf_laeuft :=M1.1          //Interne Daten
  letzter_block:=M1.2         //Interne Daten
  fehl_empf  :=M1.3          //Interne Daten
  U          M      1.0       //Empfang fertig
  R          M      1.0       //loesche Empfang fertig
CALL FC      0                //Sende Daten
  ADR        :=0              //1. DW im SEND/EMPF_DB
  _DB        :=DB10           //Sende_DB Telegramm
  ABD        :=W#16#14        //1. DW Sendepuffer (DW20)
  ANZ        :=MW14           //Anzahl zu sendender Daten
  FRG        :=M2.0           //Senden fertig angeben
  PAFE       :=MB16           //Fehlerbyte
  GESE       :=MW104          //Interne Daten
  ANZ_INT    :=MW106          //Interne Daten
  ende_kom   :=M2.1          //Interne Daten
  letzter_block:=M2.2         //Interne Daten
  senden_laeuft:=M2.3         //Interne Daten
  fehler_kom :=M2.4          //Interne Daten

OB100:
  UN        M      3.0
  S         M      3.0       //Anlauf der CPU erfolgt

```

## Projekt übertragen

Die Datenübertragung erfolgt über MPI. Sollte Ihr Programmiergerät keine MPI-Schnittstelle besitzen, können Sie für eine serielle Punkt-zu-Punkt-Übertragung von Ihrem PC an MPI das "Green Cable" von VIPA verwenden.

Das "Green Cable" hat die Best.-Nr. VIPA 950-0KB00 und darf nur bei den VIPA CPUs mit MP<sup>2</sup>I-Schnittstelle eingesetzt werden.

Bitte beachten Sie hierzu die Hinweise zum Green Cable in den Grundlagen!

- Verbinden Sie Ihr PG mit der CPU.
- Mit **Zielsystem** > *Laden in Baugruppe* in Ihrem Projektierool übertragen Sie Ihr Projekt in die CPU.
- Stecken Sie eine MMC und übertragen Sie mit **Zielsystem** > *RAM nach ROM kopieren* Ihr Anwenderprogramm auf die MMC.
- Während des Schreibvorgangs blinkt die "MC"-LED auf der CPU. Systembedingt wird zu früh ein erfolgter Schreibvorgang gemeldet. Der Schreibvorgang ist erst beendet, wenn die LED erlischt.

## Was ist das Green Cable ?



Das Green Cable ist ein grünes Verbindungskabel, das ausschließlich zum Einsatz an VIPA System-Komponenten konfektioniert ist.

Mit dem Green Cable können Sie:

- Projekte Punkt-zu-Punkt seriell übertragen
- Firmware-Updates der CPUs und Feldbus-Master durchführen



### Wichtige Hinweise zum Einsatz des Green Cable

Bei Nichtbeachtung der nachfolgenden Hinweise können Schäden an den System-Komponenten entstehen.

Für Schäden, die aufgrund der Nichtbeachtung dieser Hinweise und bei unsachgemäßem Einsatz entstehen, übernimmt die VIPA keinerlei Haftung!



### Hinweis zum Einsatzbereich

Das Green Cable darf ausschließlich direkt an den hierfür vorgesehenen Buchsen der VIPA-Komponenten betrieben werden (Zwischenstecker sind nicht zulässig). Beispielsweise ist vor dem Stecken des Green Cable ein gestecktes MPI-Kabel zu entfernen.

Zurzeit unterstützen folgende Komponenten das Green Cable:

VIPA CPUs mit MP<sup>2</sup>I-Buchse sowie die Feldbus-Master von VIPA.



### Hinweis zur Verlängerung

Die Verlängerung des Green Cable mit einem weiteren Green Cable bzw. die Kombination mit weiteren MPI-Kabeln ist nicht zulässig und führt zur Beschädigung der angeschlossenen Komponenten!

Das Green Cable darf nur mit einem 1:1 Kabel (alle 9 Pin 1:1 verbunden) verlängert werden.



## Standardhantierungsbausteine

### SEND (FC 0)

Dieser FC dient zur Datenausgabe von der CPU an den CP 240. Hierbei legen Sie über die Bezeichner `_DB`, `ADB` und `ANZ` den Sendebereich fest.

Über das Bit `FRG` wird der Sendeanstoß gesetzt und die Daten werden gesendet. Nach dem Übertragen der Daten setzt der Hantierungsbaustein das Bit `FRG` wieder zurück.

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** Peripherieadresse unter der der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.

**\_DB** Nummer des Datenbausteins, der die zu sendenden Daten beinhaltet.

**ABD** Wortvariable, welche die Nummer des Datenworts enthält, ab dem die auszugebenden Zeichen abgelegt sind.

**ANZ** Anzahl der Bytes, die zu übertragen sind.

**FRG**  
**Sendefreigabe** Bei `FRG = "1"` werden die über `_DB`, `ADB` und `ANZ` definierten Daten einmalig an den über `ADR` adressierten CP übertragen. Nach der Übertragung wird `FRG` wieder zurückgesetzt. Ist beim Aufruf `FRG = "0"`, wird der Baustein sofort wieder verlassen!

**PAFE** Alle Bits dieses Merker-Bytes sind bei richtiger Funktion "0". Bei Fehlfunktion wird ein Fehlercode eingetragen. Die Fehlerangabe ist selbstquittierend, d.h. nach Beseitigung der Fehlerursache wird das Byte wieder "0" gesetzt. Folgende Fehler sind möglich:

1 = Datenbaustein nicht vorhanden

2 = Datenbaustein zu kurz

3 = Datenbausteinnummer nicht im gültigen Bereich

**GESE, ANZ\_INT**  
**ENDE\_KOM**  
**LETZTER\_BLOCK**  
**SENDEN\_LAEUFT**  
**FEHLER\_KOM** Diese Parameter werden intern verwendet. Sie dienen dem Informationsaustausch zwischen den Hantierungsbausteinen. Für den Einsatz des `SYNCHRON_RESET` (FC9) sind die Steuerbits `ENDE_KOM`, `LETZTER_BLOCK`, `SENDEN_LAEUFT` und `FEHLER_KOM` immer in einem Merker-Byte abzulegen.

**RECEIVE (FC 1)**

Dieser FC dient zum Datenempfang vom CP 240. Hierbei legen Sie über die Bezeichner `_DB` und `ADB` den Empfangsbereich fest.

Ist der Ausgang `EMFR` gesetzt, so ist ein neues Telegramm komplett eingelesen worden. Die Länge des eingelesenen Telegramms wird in `ANZ` abgelegt. Nach der Auswertung des Telegramms ist dieses Bit vom Anwender zurückzusetzen, da ansonsten kein weiteres Telegramm in der CPU übernommen werden kann.

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** Peripherieadresse unter der der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.

**\_DB** Nummer des Datenbaustein, der die empfangenen Daten beinhaltet.

**ABD** Wortvariable, welche die Nummer des Datenworts enthält, ab dem die empfangenen Zeichen abgelegt sind.

**ANZ** Wortvariable, welche die Anzahl der Bytes enthält, die empfangen wurden.

**EMFR** Durch Setzen des `EMFR` zeigt der Hantierungsbaustein an, dass Daten empfangen wurden. Erst durch Rücksetzen von `EMFR` im Anwenderprogramm können weitere Daten empfangen werden.

**PAFE** Alle Bits dieses Merker-Bytes sind bei richtiger Funktion "0". Bei Fehlfunktion wird ein Fehlercode eingetragen. Die Fehlerangabe ist selbstquittierend, d.h. nach Beseitigung der Fehlerursache wird das Byte wieder "0" gesetzt. Folgende Fehler sind möglich:  
 1 = Datenbaustein nicht vorhanden  
 2 = Datenbaustein zu kurz  
 3 = Datenbausteinnummer nicht im gültigen Bereich

**GEEM, ANZ\_INT, LETZTER\_BLOCK, EMPF\_LAEUFT, FEHLER\_EMPF** Diese Parameter werden intern verwendet. Sie dienen dem Informationsaustausch zwischen den Hantierungsbausteinen. Für den Einsatz des `SYNCHRON_RESET` (FC9) sind die Steuerbits `LETZTER_BLOCK`, `EMPF_LAEUFT` und `FEHLER_EMPF` immer in einem Merker-Byte abzulegen.

## SYNCHRON\_ RESET

Synchronisation und Rücksetzen (FC 9)

Der Baustein ist im zyklischen Programmteil aufzurufen. Mit dieser Funktion wird die Anlaufkennung des CP 240 quittiert, und so die Synchronisation zwischen CPU und CP hergestellt. Weiterhin kann bei einer Kommunikationsunterbrechung der CP rückgesetzt werden und so ein synchroner Anlauf erfolgen.



### Hinweis!

Eine Kommunikation mit SEND- und RECEIVE-Bausteinen ist nur möglich, wenn zuvor im Anlauf-OB der Parameter ANL des SYNCHRON-Bausteins gesetzt wurde.

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** Peripherieadresse unter der der CP 240 anzusprechen ist. Über die Hardware-Konfiguration bestimmen Sie die Peripherieadresse.

**TIMER\_NR** Nummer des Timers für die Wartezeit.

**ANL** Mit ANL = 1 wird dem Hantierungsbaustein mitgeteilt, dass an der CPU STOP/START bzw. NETZ-AUS/NETZ-EIN erfolgt ist und nun eine Synchronisation erfolgen muss. Nach der Synchronisation wird ANL automatisch zurückgesetzt.

**NULL** Parameter wird intern verwendet.

**RESET** Mit RESET = 1 können Sie den CP aus Ihrem Anwenderprogramm zurücksetzen.

**STEUERB\_S** Hier ist das Merkerbyte anzugeben, in dem die Steuerbits ENDE\_KOM, LETZTER\_BLOCK, SENDEN\_LAEUFT und FEHLER\_KOM für den SEND-FC abgelegt sind.

**STEUERB\_R** Hier ist das Merkerbyte anzugeben, in dem die Steuerbits LETZTER\_BLOCK, EMPF\_LAEUFT und FEHLER\_EMPF für den RECEIVE-FC abgelegt sind.

## Kommunikationsprinzip

### Daten senden und empfangen

Zu sendende Daten werden von der CPU über den Rückwandbus in den entsprechenden Datenkanal geschrieben. Der Kommunikationsprozessor trägt diese in einem Ringpuffer (2048Byte) ein und gibt sie von dort über EnOcean aus.

Empfängt der Kommunikationsprozessor Daten über EnOcean, werden die Daten in einem Ringpuffer (2048Byte) abgelegt. Die empfangenen Daten können über den Datenkanal telegrammweise (11Byte) von der CPU gelesen werden.

### Kommunikation über Rückwandbus

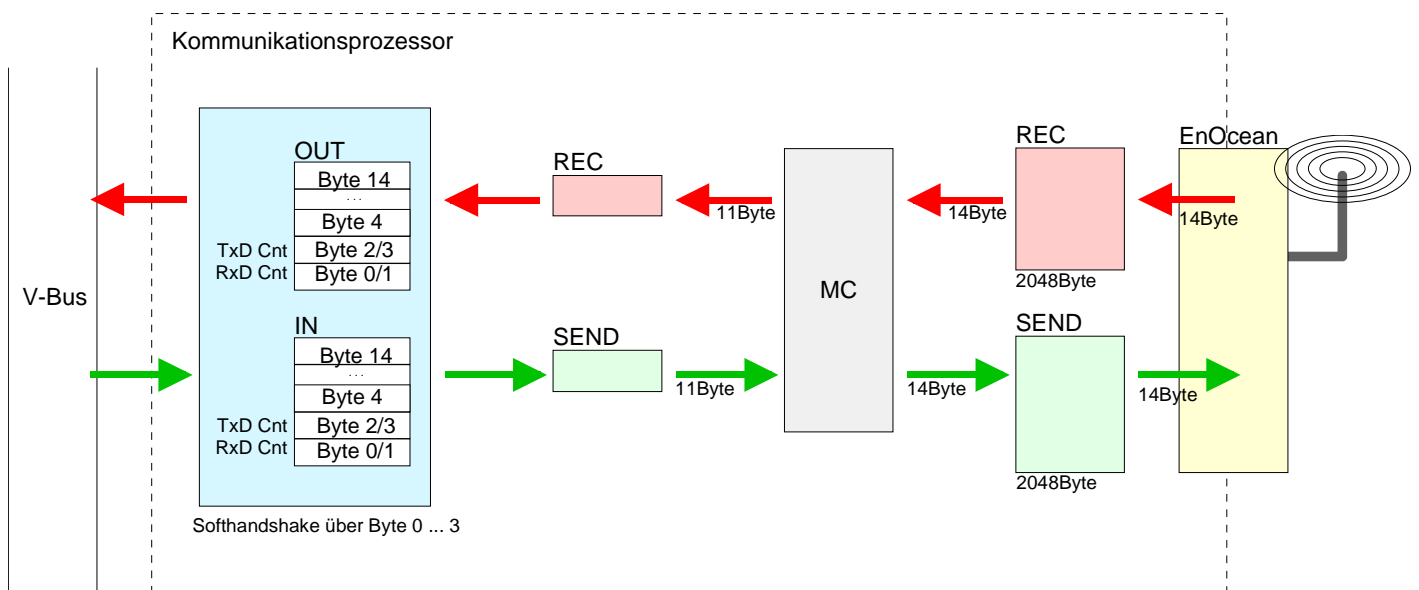
Der Austausch von empfangenen Telegrammen über den Rückwandbus erfolgt asynchron. Ist ein komplettes Telegramm über EnOcean eingetroffen, wird dies im Puffer abgelegt. Aus der Länge des Ringpuffers ergibt sich die maximale Anzahl der Telegramme. Ist der Puffer voll, werden neu ankommende Telegramme verworfen.

Aus den 14Byte großen Telegrammen werden telegrammweise 11Byte Nutzdaten über den Rückwandbus an die CPU übergeben. Die ersten beiden 2 Sync-Bytes und die Checksumme werden nicht weitergereicht.

### Aufgaben der CPU

Ein zu sendendes Telegramm ist an den CP 240 zu übergeben. Dieser ergänzt das Telegramm mit den ersten beiden Sync-Bytes und der Checksumme und reicht das Telegramm an den Sendepuffer weiter. Im CP 240 werden diese Blöcke im Sendepuffer zusammengesetzt und bei Vollständigkeit des Telegramms über den EnOcean-Transceiver gesendet. Da der Datenaustausch über den Rückwandbus asynchron abläuft, wird ein "Software Handshake" zwischen dem CP 240 und der CPU eingesetzt. Die Register für den Datentransfer vom CP 240 sind 16Byte breit. Für den Handshake sind die Bytes 0 bis 3 (Wort 0 und 2) reserviert.

Folgende Abbildung soll dies veranschaulichen:



---

**Software-  
handshake**

Für den Einsatz des CP 240 in Verbindung mit einer System 200V CPU sind bei VIPA Hantierungsbausteine erhältlich, die den Softwarehandshake komfortabel übernehmen.

Bei Einsatz des CP 240 ohne Hantierungsbausteine soll hier die Funktionsweise anhand eines Beispiels für das Senden und Empfangen von Daten erläutert werden.

**Beispiel Daten  
senden ohne  
Hantierungs-  
baustein**

Ein EnOcean-Telegramm besitzt 11Byte Nutzdaten. Beim Senden werden von der CPU je Telegramm 11Byte Nutzdaten in die Bytes 4 bis 14 und in Byte 2/3 die Länge des Telegramms (also "11") geschrieben. Der CP 240 empfängt die Daten über den Rückwandbus. Zur Quittierung des Telegramms schreibt der CP 240 in Byte 2/3 den Wert "11" (Länge des Telegramms) an die CPU zurück.

Beim Empfang der "11" auf Byte 2/3 sendet die CPU eine "0" auf Byte 2/3. Daraufhin werden im CP 240 die Nutzdaten am Anfang mit 2 Sync-Bytes und am Ende mit der Checksum auf 14Byte ergänzt und im Sendepuffer abgelegt. Ist dies erfolgt, antwortet der CP mit einer "0" auf Byte 2/3. Beim Empfang der "0" kann die CPU ein neues Telegramm an den CP 240 senden.

Die im Sendepuffer abgelegten Telegramme werden sofort über EnOcean ausgegeben.

**Beispiel Daten  
empfangen ohne  
Hantierungs-  
baustein**

Jedes EnOcean-Telegramm hat eine Größe von 14Byte. Empfängt der CP 240 ein Telegramm, so wird dies im Empfangspuffer abgelegt. Von jedem Telegramm werden die 11Byte Nutzdaten in Byte 4 bis 14 und die Länge (also "11") in Byte 0/1 über den Rückwandbus an die CPU übergeben. Die ersten beiden Sync-Bytes und die Checksumme werden verworfen.

Die CPU speichert die Nutzdaten und antwortet mit dem Wert "11" auf Byte 0/1. Dies quittiert der CP mit einer "0" auf Byte 0/1 und meldet somit, dass der Transfer abgeschlossen ist. Sobald neue Daten übertragen werden können, antwortet die CPU mit "0".

Mit dem Empfang der "0" kann der CP 240 ein neues Telegramm an die CPU senden.

## Beispiel zum Einsatz unter EnOcean

### Übersicht

In dem nachfolgenden Beispiel wird eine EnOcean-Kommunikation (Senden und Empfangen) aufgebaut. Weiter soll das Beispiel zeigen, wie Sie unter Einsatz der Hantierungsbausteine auf einfache Weise die Kontrolle über die Kommunikationsvorgänge haben.

Bei Bedarf können Sie das Beispielprojekt von VIPA beziehen.

### Voraussetzung

Folgende Komponenten sind für das Beispiel erforderlich:

1 System 200V bestehend aus CPU 21x und CP 240 EnOcean

1 Schalter mit EnOcean-Sender

Projektiertool SIMATIC Manager von Siemens mit Übertragungskabel

### Vorgehensweise

Bauen Sie das System 200V auf.

Laden Sie das Beispielprojekt, passen Sie ggf. die Peripherieadresse an und übertragen Sie Ihr Projekt in die CPU.

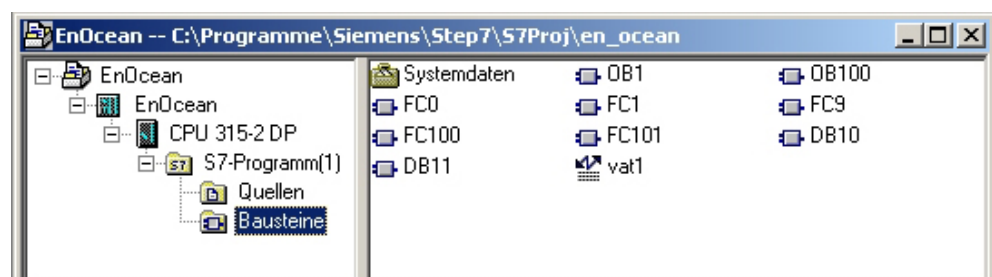
### Projekt dearchivieren

Im Siemens SIMATIC Manager gehen Sie nach folgenden Schritten vor:

- Starten Sie den Siemens SIMATIC Manager.
- Zum Entpacken der Datei EnOcean.zip gehen Sie auf **Datei** > *dearchivieren*.
- Wählen sie die Beispieldatei EnOcean.zip aus und geben Sie als Zielverzeichnis "s7proj" an.
- Öffnen Sie das entpackte Projekt.

### Projekt-Struktur

Das Projekt beinhaltet schon das SPS-Programm und die Hardware-Konfiguration und besitzt folgende Struktur:



**Datenbausteine** In diesem Beispiel werden folgende Datenbausteine verwendet:

DB10  
Sendebaustein

Adr.	Name	Typ	Kommentar
0.0		STRUCT	
+0.0	Sendefach	STRUCT	
+0.0	RX_TX_Kennung	BYTE	0B=RX/6B=TX
+1.0	ORG	BYTE	
+2.0	Datenbyte3	BYTE	Datenbyte 3
+3.0	Datenbyte2	BYTE	Datenbyte 2
+4.0	Datenbyte1	BYTE	Datenbyte 1
+5.0	Datenbyte0	BYTE	Datenbyte 0
+6.0	IDbyte2_3	WORD	ID Byte 2 und 3
+8.0	IDbyte0_1	WORD	ID Byte 0 und 1
+10.0	Status	BYTE	Status
=12.0		END_STRUCT	
+12.0	Reserve	BYTE	
+13.0	SENDEN_LAEUFT	BOOL	Senden läuft noch
+13.1	LETZTER_BLOCK	BOOL	letzter Block wurde gesendet
+13.2	FEHL_KOM	BOOL	Fehler beim Senden aufgetreten
+13.3	ENDE_KOM	BOOL	Übertragung abgeschlossen
+14.0	PAFE	BYTE	Parametrierfehler-Byte des 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	Telegramm komplett gesendet
+16.0	GESE	WORD	schon gesendete Daten
+18.0	ANZ_INT	WORD	Anzahl gesendete Daten
+20.0	Reserve1	ARRAY[0..50]	
*1.0		BYTE	
=72.0		END_STRUCT	

DB11  
Empfangsbaustein

Adr.	Name	Typ	Kommentar
0.0		STRUCT	
+0.0	Empfangsfach	STRUCT	
+0.0	RX_TX_Kennung	BYTE	0B=RX/6B=TX
+1.0	ORG	BYTE	
+2.0	Datenbyte3	BYTE	Datenbyte 3
+3.0	Datenbyte2	BYTE	Datenbyte 2
+4.0	Datenbyte1	BYTE	Datenbyte 1
+5.0	Datenbyte0	BYTE	Datenbyte 0
+6.0	IDbyte2_3	WORD	ID Byte 2 und 3
+8.0	IDbyte0_1	WORD	ID Byte 0 und 1
+10.0	Status	BYTE	Status
=12.0		END_STRUCT	
+12.0	Reserve	BYTE	
+13.0	EMP_LAEUFT	BOOL	Empfangen läuft noch
+13.1	LETZTER_BLOCK	BOOL	letzter Block wurde empfangen
+13.2	FEHL_EMPF	BOOL	Fehler beim Empfang aufgetreten
+14.0	PAFE	BYTE	Parametrierfehler-Byte des 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	Telegramm komplett empfangen
+16.0	GEEM	WORD	schon empfangene Daten
+18.0	ANZ_INT	WORD	Anzahl empfangener Daten
+20.0	Reserve1	ARRAY[0..50]	
*1.0		BYTE	
=72.0		END_STRUCT	



**SPS-Programm** Das Beispiel beinhaltet schon das SPS-Programm und die Hardware-Konfiguration. Hierbei kommen folgende Bausteine zum Einsatz:

```

OB 1      CALL FC 9                //Neustart oder Reset
          ADR      :=256           //Adresse des Moduls
          TIMER_NR :=T2
          ANL      :=M3.0
          NULL     :=M3.1
          RESET    :=M3.2
          STEUERB_S:=MB4
          STEUERB_R:=MB6
          U M 3.0                //Empfang kann erst nach Abarbeitung von FC 9
          BEB      //((SYNCHRON_RESET) gestartet werden
          CALL FC 100           //Aufruf des Empfangs-FC

OB 100   L 0
          T MB 1                //Auftragsbit löschen
          UN M 3.0             //Neustart auslösen
          S M 3.0

FC 100   CALL FC 1
          ADR      :=256           //Adresse des Moduls
          _DB     :="EMPFANG_en_ocean" //DB mit Empfangsdaten
          ABD     :=W#16#0         //1. DBB Empfangsdaten
          ANZ     :=W#16#B        //Empfangslänge immer 11
          EMFR    :=M7.0          //alle Daten empfangen
          PAFE    :="EMPFANG_en_ocean".Pafe //Fehlerbyte
          GEEM    :="EMPFANG_en_ocean".GEEM //Empfangene Anzahl (intern)
          ANZ_INT:= "EMPFANG_en_ocean".ANZ_INT //Empfangslänge (intern)
          EMPF_LAEUFT:= "EMPFANG_en_ocean".EMPF_LAEUFT //Datenempfang läuft (intern)
          LETZTER_BLOCK:= "EMPFANG_en_ocean".LETZTER_BLOCK //alle Daten empfangen
          FEHL_EMPF:= "EMPFANG_en_ocean".FEHL_EMPF //Fehler in der
          //Empfangsroutine
          UN M 7.0                //kein Telegramm empfangen
          BEB      //dann Ende
          R M 7.0                 //Empfangsbit löschen
          L "EMPFANG_en_ocean".Empfangsfach.IDbyte0_1 //Schalterkennung
          L W#16#1C7A             //Bitte hier Kennung Ihres
          ==I                      //Schalters angeben.
          SPB e_a2                //Diese kann dem DB 11.DBW 8
          BEA                      //entnommen werden

e_a2:   NOP 0
          L 5                      //Kennung Schalter ein
          L "EMPFANG_en_ocean".Empfangsfach.Datenbyte3 //Byte mit Kennung
          SRW 4                    //Kennung im Low-Nippel
          ==I                      //Prüfen ob Schalter gedrückt
          SPB ein
          L 7                      //Kennung Schalter aus
          ==I                      //Prüfen ob Schalter gedrückt
          SPB aus
          BEA

ein:    NOP 0
          S A 0.0                  //Funktion ein
          BEA

aus:    NOP 0
          R A 0.0                  //Funktion aus
          BEA

```

## FC 101

```

L      B#16#6B                                     //Sendedaten vorbelegen
T      "SEND_en_ocean".Empfangsfach.RX_TX_Kennung //Kennung senden
L      B#16#5                                       //ORG-Kennung
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                                   //Nur die letzten 7Bit
T      "SEND_en_ocean".Empfangsfach.IDbyte0_1     //sind für Adr. relevant
L      6                                           //und werden im CP 240
T      "SEND_en_ocean".Empfangsfach.Status       //mit der dort abgelegten
                                                //IDBase verodert

CALL FC 0
ADR      :=256
_DB      := "SEND_en_ocean"
ABD      :=W#16#0                                  //ab Datenbyte 0 senden
ANZ      :=W#16#B                                  //immer 11 Byte
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

```

## Übersicht der EnOcean-Telegramme

### Allgemeiner Aufbau

Die nachfolgende Tabelle zeigt den allgemeinen Aufbau eines EnOcean-Telegramms. Sende- und Empfangstelegramme besitzen die gleiche Struktur. Sie unterscheiden sich ausschließlich in der Kennung.

Bit 7	Bit 0
0xA5	Diese Bytes werden beim Senden automatisch generiert und beim Empfang ausgeblendet.
0x5A	
0x0B 0x06	0x0B: Kennung für Empfangstelegramm 0x06: Kennung für Sendetelegramm
<b>ORG</b>	Siehe Tabelle <i>unterstützte ORG-Formate</i>
DataBytes3	Daten von einem Sensor bzw. an einen Aktor
DataBytes2	
DataBytes1	
DataBytes0	
IDBytes3*	ID des Transceiver-Moduls. Mit SET_IDBASE können Sie die ID bis zu 10 Mal ändern
IDBytes2*	
IDBytes1*	
IDBytes0*	
Status	Statusinformation des entsprechenden Sensors
Checksum	Wird beim Senden automatisch generiert und beim Empfang ausgeblendet.

\*) Beim Senden wird die ID-Base im Telegramm durch die tatsächliche ID-Base des Moduls ersetzt.

### Allgemein

Auf den Folgeseiten sind alle Telegramme aufgelistet, die vom CP 240 EnOcean unterstützt werden. Diese Beschreibung wurde mit freundlicher Genehmigung der Firma EnOcean in englischer Sprache direkt aus der Dokumentation übernommen.



#### Hinweis!

Bitte beachten Sie, dass im CP 240 bei empfangenen Telegrammen die ersten beiden Synchronisations-Bytes und die Checksumme nicht abgelegt werden. Beim Senden werden die 11Byte Nutzdaten automatisch mit diesen Bytes auf 14Byte ergänzt.

**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

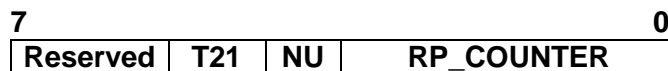
**Serial command encoding for MDA**

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)**



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.

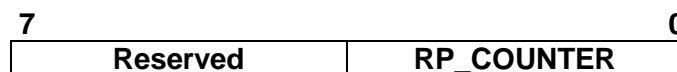
**PTM Type 2**

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:**

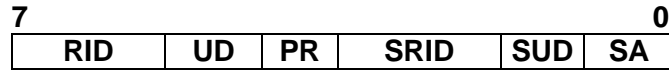


Reserved (4 bit) Do not care  
 RP\_COUNTER (4 bit) Repeater level: 0 original message  
 1 repeated message

**Description of  
DATA\_BYTE 3..0**

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

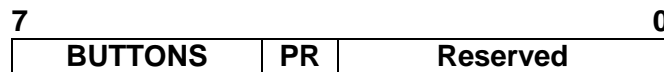
DATA\_BYTE2..0 always = 0  
DATA\_BYTE3 as follows:



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:



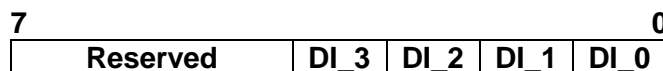
BUTTONS	(3 bit)	Number of simultaneously pressed buttons, as follows: <table style="margin-left: 20px; border-collapse: collapse;"> <tr> <td style="width: 50%;">PTM 100</td> <td style="width: 50%;">PTM200</td> </tr> <tr> <td>0 = 0 Buttons</td> <td>0 = 0 Button</td> </tr> <tr> <td>1 = 2 Buttons</td> <td>1 = not possible</td> </tr> <tr> <td>2 = 3 Buttons</td> <td>2 = not possible</td> </tr> <tr> <td>3 = 4 Buttons</td> <td>3 = 3 or 4 buttons</td> </tr> <tr> <td>4 = 5 Buttons</td> <td>4 = not possible</td> </tr> <tr> <td>5 = 6 Buttons</td> <td>5 = not possible</td> </tr> <tr> <td>6 = 7 Buttons</td> <td>6 = not possible</td> </tr> <tr> <td>7 = 8 Buttons</td> <td>7 = not possible</td> </tr> </table>	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
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:



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

- DATA\_BYTE2..0 always = 0
- DATA\_BYTE3 as follows:



- 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.





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

Bit 7	Bit 0
<b>0xA5</b>	
<b>0x5A</b>	
<b>0x8B</b>	
<b>0x58</b>	
X	
X	
X	
X	
X	
X	
X	
X	
X	
X	
<b>ChkSum</b>	

**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
<b>0xA5</b>	
<b>0x5A</b>	
<b>0x8B</b>	
<b>0x19</b>	
X	
X	
X	
X	
X	
X	
X	
X	
X	
X	
<b>ChkSum</b>	

**RD\_IDBASE** When this command is sent to the TCM, the base ID range number is retrieved though an INF\_IDBASE telegram.

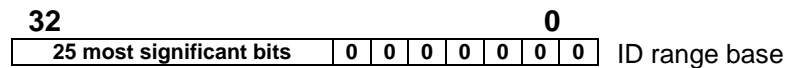
Bit 7	Bit 0
<b>0xA5</b>	
<b>0x5A</b>	
<b>0xAB</b>	
<b>0x58</b>	
X	
X	
X	
X	
X	
X	
X	
X	
X	
<b>ChkSum</b>	

**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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0xAB</i>	
<i>0x18</i>	
<i>IDBaseByte3</i>	
<i>IDBaseByte2</i>	
<i>IDBaseByte1</i>	
<i>IDBaseByte0</i>	
X	
X	
X	
X	
X	
<i>ChkSum</i>	



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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0x8B</i>	
<i>0x98</i>	
<i>IDBaseByte3</i>	
<i>IDBaseByte2</i>	
<i>IDBaseByte1</i>	
<i>IDBaseByte0</i>	
X	
X	
X	
X	
X	
<i>ChkSum</i>	

IDBaseByte3 is the most significant byte.

**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	
		<b>0xA5</b>
		<b>0x5A</b>
		<b>0xAB</b>
		<b>0x08</b>
		<b>Sensitivity</b>
		X
		X
		X
		X
		X
		X
		X
		X
		X
		<b>ChkSum</b>

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 a INF\_RX\_SENSITIVITY command.

Bit 7	Bit 0	
		<b>0xA5</b>
		<b>0x5A</b>
		<b>0xAB</b>
		<b>0x48</b>
		X
		X
		X
		X
		X
		X
		X
		X
		X
		<b>ChkSum</b>

**INF\_RX\_SENSITIVITY** This message informs the user about the current TCM radio sensitivity.

Bit 7	Bit 0	
		<b>0xA5</b>
		<b>0x5A</b>
		<b>0x8B</b>
		<b>0x88</b>
		<b>Sensitivity</b>
		X
		X
		X
		X
		X
		X
		X
		X
		<b>ChkSum</b>

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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0xAB</i>	
<i>0x09</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>ChkSum</i>	

**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
<i>0xAA</i>	

**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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0xAB</i>	
<i>0x0A</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>ChkSum</i>	

**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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0xAB</i>	
<i>0x28</i>	
<i>Modem ID (MSB)</i>	
<i>Modem ID (LSB)</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>ChkSum</i>	

**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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0xAB</i>	
<i>0x2A</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>ChkSum</i>	

**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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0xAB</i>	
<i>0x68</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>X</i>	
<i>ChkSum</i>	

**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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0x8B</i>	
<i>0xA8</i>	
<i>Modem status</i>	
<i>Modem ID MSB</i>	
<i>Modem ID LSB</i>	
X	
X	
X	
X	
X	
X	
X	
<i>ChkSum</i>	

**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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0xAB</i>	
<i>0x4B</i>	
X	
X	
X	
X	
X	
X	
X	
X	
X	
X	
<i>ChkSum</i>	

**INF\_SW\_VER**                      Informs the user about the current software version of the TCM.

Bit 7	Bit 0
<i>0xA5</i>	
<i>0x5A</i>	
<i>0x8B</i>	
<i>0x8C</i>	
<i>TCM SW Version Pos.1</i>	
<i>TCM SW Version Pos.2</i>	
<i>TCM SW Version Pos.3</i>	
<i>TCM SW Version Pos.4</i>	
X	
X	
X	
X	
X	
<i>ChkSum</i>	

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\_NO\_TWANTEDACK**      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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0x8B</i>	
<i>0x28</i>	
X	
X	
X	
X	
X	
X	
X	
X	
X	
<i>ChkSum</i>	

**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
<i>0xA5</i>	
<i>0x5A</i>	
<i>0x8B</i>	
<i>0x29</i>	
X	
X	
X	
X	
X	
X	
X	
X	
<i>ChkSum</i>	

**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	
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	
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
<b>0xA5</b>	
<b>0x5A</b>	
<b>0x8B</b>	
<b>0x22</b>	
X	
X	
X	
X	
X	
X	
X	
X	
X	
X	
X	
<b>ChkSum</b>	

**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
<b>0xA5</b>	
<b>0x5A</b>	
<b>0x8B</b>	
<b>0x1A</b>	
X	
X	
X	
X	
X	
X	
X	
X	
X	
X	
X	
<b>ChkSum</b>	

## Modul ersetzen und IDBase übernehmen

### Übersicht

Da die IDBase jedes Moduls unterschiedlich ist, haben Sie die Möglichkeit im Ersatzfall bis zu 10 Mal die IDBase eines Moduls mit einem SET\_IDBASE-Telegramm zu ändern. Somit entfällt das erneute Abstimmen der Aktoren auf das Ersatz-Modul. Nach erfolgreicher Übertragung der IDBase ist entweder die CPU neu zu starten oder Reset über FC 9 durchzuführen.

Bitte beachten Sie, dass nur die oberen 25 Bits als IDBase übernommen werden. Die restlichen 7 Bits können Sie über Ihr Anwenderprogramm zur Laufzeit angeben und hiermit mehrere Aktoren adressieren.

### IDBase ermitteln

Mit RD\_IDBASE können Sie die aktuelle IDBase Ihres Moduls abfragen.

#### RD\_IDBASE

0xAB	Kennung für Sendetelegramm
<b>0x58</b>	ORG-Kennung für RD_IDBASE
X	Irrelevant
...	...
X	Irrelevant

#### INF\_IDBASE

RD\_IDBASE liefert die aktuelle IDBase des Moduls in Form eines INF\_IDBASE-Telegramms zurück. Das Telegramm hat folgenden Aufbau:

0x8B	Kennung für Empfangstelegramm
<b>0x98</b>	ORG-Kennung für INF_IDBASE
<i>IDBaseByte3</i>	Byte 3 aktuelle IDBase
<i>IDBaseByte2</i>	Byte 2 aktuelle IDBase
<i>IDBaseByte1</i>	Byte 1 aktuelle IDBase
<i>IDBaseByte0</i>	Byte 0 aktuelle IDBase (Bit 6...0 irrelevant)
X	irrelevant
...	...
X	irrelevant

**SET\_IDBASE**

Im Ersatzfall senden Sie ein SET\_IDBASE-Telegramm nach folgender Struktur von Ihrer CPU an das Modul (Transceiver). Verwenden Sie als neue IDBase die Adresse des zu ersetzenden Moduls:

0xAB	Kennung für Sendetelegramm
<b>0x18</b>	ORG-Kennung für SET_IDBASE
<i>IDBaseByte3</i>	Byte 3 neue IDBase
<i>IDBaseByte2</i>	Byte 2 neue IDBase
<i>IDBaseByte1</i>	Byte 1 neue IDBase
<i>IDBaseByte0</i>	Byte 0 neue IDBase (Bit 6...0 irrelevant)
X	irrelevant
...	...
X	irrelevant

Mögliche Antwort-  
Telegramme

0x8B <b>0x58</b>	← OK - IDBase wurde übernommen
---------------------	--------------------------------

Zur Übernahme der IDBase zur Laufzeit, ist Reset über FC 9 durchzuführen. Ansonsten steht ihnen nach einem CPU-Neustart die neue IDBase zur Verfügung.

Im Fehlerfall erhalten Sie eine dieser Meldungen. Hierbei bleibt die alte IDBase erhalten.

0x8B <b>0x19</b>	← <b>ERR</b> Mehr als 10 IDBase-Wechsel sind nicht zulässig
0x8B <b>0x1A</b>	← <b>ERR_IDRANGE</b> IDBase liegt außerhalb des zulässigen Bereichs (0xFF800000 ... 0xFFFFFFFF).

Überprüfen Sie Ihre ID-Angaben und senden Sie das Telegramm erneut.

